Geometric Problems in Molecular Biology and Robotics

David Parsons and **John Canny** *
571 Evans Hall
University of California
Berkeley, CA 94720

Abstract

Some of the geometric problems of interest to molecular biologists have macroscopic analogues in the field of robotics. Two examples of such analogies are those between protein docking and model-based perception, and between ring closure and inverse kinematics. Molecular dynamics simulation, too, has much in common with the study of robot dynamics. In this paper we give a brief survey of recent work on these and related problems.

Keywords: protein docking, model-based perception, ring closure, robot kinematics, protein folding, dynamics simulation

Introduction

Structural biology is that branch of biology which studies the properties and functions of proteins and other macromolecules based on analyses of their physical conformations in space. Robotics is an interdisciplinary science concerned with the synthesis of certain aspects of human function and ability to interact with the environment. On the surface, these two fields seem to have little in common. However, below the level of their application domains lies a common denominator: three-dimensional geometry. The central tenet of structural biology is that much of the chemical and biological function of biomolecules can be explained as a function of their geometric conformations. The synthesis of robots and robotic algorithms is also tightly bound with geometric analysis.

With this simple observation in mind, we have been engaged in reviewing the progress made in various geometric problems in the two fields, with an eye to finding similarities in problem structures. Our hope is that the algorithmic tools and techniques which have proven useful in one field may serve as useful suggestions to researchers in the other. This paper summarizes the current state of this comparison effort. It is not intended to be an exhaustive survey of these problems; rather, we merely hope to provide some pointers into the literature by which others may find papers of interest to their own research.

The specific analogies discussed here are between protein docking problems and model-based perception; between certain structure prediction problems and manipulator kinematics; and between dynamics simulations of molecules and of robot systems. In some of these comparisons we find exactly analogous situations; in others, either the problems or the central bottlenecks in their solutions are less similar.

Geometric Matching

In a geometric matching problem we are given two three-dimensional objects and we wish to compare or align them in some way. The comparison is made based on a fixed set of geometric primitives, or features, of the objects. The goal of the comparison is to determine whether some subset of the features of one of the objects bears a geometric similarity with a subset of features of the other. This problem arises in robotics in the context of interpreting sensory data, and in molecular biology it appears as the protein docking problem.

Protein Docking

In the protein docking problem, also called molecular recognition, the objects to be matched are molecules. One of them is generally a protein (the "receptor"), and the other may be another protein, a DNA sequence, or a smaller molecule such as an enzyme inhibitor (the "ligand"). The goal is to determine whether the two can associate, and if so, to predict the geometric structure of their combined complex (Cherfils & Janin 1993). A good solution to this problem would have important applications in the field of rational drug design (Kuntz 1992), (Navia & Murcko 1992).

Protein docking in full generality is a very complex problem, because each molecule may have many thou-

Support for this research was provided by NSF Presidential Young Investigator Grant #IRI-8958577.

sands of internal degrees of freedom. To overcome this, researchers usually simplify the problem by treating the molecules as rigid bodies, thereby reducing the dimension of the search space to 6 (Shoichet & Kuntz 1991). When X-ray crystallographic data is available, this approximation can be justified by comparing the experimental structures of bound complexes with those of their free components (Janin & Chothia 1990); most known associating proteins (but not all) are observed to behave as rigid bodies.

There are two problems which a docking algorithm must address. The first is the geometric problem of computing "reasonable" relative configurations between the receptor and the ligand, and the second is the chemical problem of evaluating the free energies of tentative matches. Our concern in this paper is the first problem, although the second is also far from solved; current theoretical models of energy evaluation are sufficiently inaccurate that they cannot in general distinguish between known complexes and geometrically reasonable "false positives" (Shoichet & Kuntz 1991).

Various different types of features have been used in the matching process. In (Kuhl, Crippen, & Friesen 1984), the features of the ligand are the atoms on its surface, considered as point masses, while the receptor features are "site points", which may be thought of as the centers of pockets in the molecular surface of the receptor into which ligand atoms may be placed. The set of site points constitutes a negative image of the active regions of the receptor surface. Recently, (Lin *et al.* 1994) proposed a representation which supplements such point information with surface normals for better discrimination. In another approach ((Kuntz *et al.* 1982), (Shoichet & Kuntz 1991)), the features used are spheres. This algorithm generates one set of spheres which fill the "ridges" on the surface of the ligand and another set which fill the "grooves" on the receptor surface. (That is, ligand spheres lie on the inside of the ligand, while the receptor's spheres lie on its outside.) Proteins have also been modelled with sets of cubes for docking purposes (Jiang & Kim 1991).

Applying direct search methods in the six-dimensional conformational space $SE(3)$ is usually considered computationally infeasible. Instead, most of the docking algorithms reviewed here employ a combinatorial approach, based on pairwise matchings between individual features from the two molecules. (Kuhl, Crippen, & Friesen 1984) defines the docking problem as that of finding a relative configuration which maximizes the number of pairwise contacts made, where a ligand atom and a receptor site point are considered to be in contact in a given configuration if the distance between them is less than some small fixed tolerance. They point out that there is a straightforward brute-force method for finding such maximal matchings which works by trying all possible matchings and which runs in time $O(m^3n^3 \min(m,n))$, where m is the number of atoms in the ligand and n is the number of receptor site points. This proves that polynomial-time algorithms are possible, but the brute force method is much too slow in practice for problems of reasonable size. They propose instead another algorithm which works by creating a combinatorial graph (the "docking graph") based on internal (intramolecular) pairwise distances. The docking graph has a node for each of the mn possible pairwise feature matches and an edge between two pairs wherever both matches can be simultaneously made. Finding a maximal matching reduces to searching for maximal cliques in this graph. Although clique-finding is known to be an NP-hard problem (Garey & Johnson 1979), the authors argue that their approach is efficient in practice due to the geometric nature of the constraints imposed by the distances.

The DOCK algorithm described in (Kuntz *et al.* 1982) also tries to maximize the number of features matched, although it does not guarantee the global maximum. The algorithm is as follows: first, systematically pair each ligand sphere with each receptor sphere. For each such pairing, choose a second pair which maximizes the number of further matches which could still be made without violating a simple intramolecular distance constraint. Then pick a third pair of spheres which maximizes the remaining possible matches, and continue in this manner until no further matches can be made. Whenever at least 4 matches are made, an orientation is uniquely defined and the match is retained as a candidate for energy evaluation (assuming it passes a handedness test and some other simple geometric filters).

The last algorithm we consider is the "soft docking" procedure of (Jiang & Kim 1991). They do a form of exhaustive search in the six-dimensional conformational space. The set $SO(3)$ of rotations is discretized, and for each discrete rotation a set of translations which cause the cube sets representing the molecules to come into contact is computed. As in the previous two algorithms, the relative configurations which get output from the geometric phase of the algorithm are those which maximize the number of features (cubes in this case) matched.

Extensive tests on real data of these last two algorithms described have been reported in their respective papers ((Kuntz *et al.* 1982) and (Shoichet & Kuntz 1991) for DOCK, (Jiang & Kim 1991) for soft

docking). For the purposes of evaluation, these tests were performed on pairs of molecules for which high-resolution X-ray crystallography data is available for the docked complexes, such as trypsin with bovine pancreatic trypsin inhibitor. In all reported cases, the algorithms found the correct conformations to a high degree of accuracy within 24 hours of computing time. In some cases, this was true even when the rigid molecule models were taken from their unbound X-ray crystal conformations, which lends further credence to the rigid body approximation. Both algorithms, however, also found many false positive conformations, especially when using the unbound molecules as starting points. This limitation reduces the applicability of docking algorithms in drug design and other contexts in which the conformation of the complex is not known a priori. More accurate energy models could eventually solve this.

The primary limitation of the geometric algorithms for molecular recognition developed so far is their computational complexity. For example, we know of no algorithm which runs fast enough to successfully dock two proteins together in any reasonable amount of computing time. One possible source of inspiration for more efficient docking algorithms is the work done in robotics on model-based perception tasks. We give an overview of this work in the next section.

Model-based Object Recognition and Localization

Geometric matching problems related to protein docking arise in the robotics domain of model-based object recognition. The context of this problem is the interpretation of sensory data, which may come from a camera, a range-finding device, tactile sensors, etc. A general statement of the basic problem is: given one or more object models and a scene possibly containing images of one or more of those objects, determine if any of the objects occur in the scene (object recognition) and if so, where (object localization). The piece of this problem which we focus on here is the localization question; we further restrict our attention to those problem domains in which the scene data are three-dimensional, such as range data, tactile sensor data, or data derived stereooptically from two cameras.

Suppose we are given a data image consisting of a finite set of points in three dimensions, some of which may lie on the surface of an object. The goal is to compute a rigid body transformation which aligns as much of the data as possible onto one of the object models known to the program. The object model may be thought of as the ligand, and the scene corresponds to the negative image of the receptor (the site points). The seminal paper in this area is (Grimson & Lozano-Pérez 1984). They introduce the *interpretation tree*, which is a way of organizing the set of possible interpretations of the data. An *interpretation* of an image with respect to one of the object models is a set of matchings of the form (s_i, f_j), where s_i is one of the m sensed points and f_j is a feature of the object model. (Because they deal with polyhedral models, a feature may be a face, an edge, or a vertex; in docking it would be an atom of the ligand.) An interpretation is represented as a path from the root to a leaf in the interpretation tree. Each edge along the path corresponds to one datum-to-feature matching of the interpretation.

The usefulness of this data structure is that it facilitates eliminating large sections of the search space by the application of geometric constraints. Useful constraints on which pairs of matchings are simultaneously feasible derive from the rigid body constraint of the object. For example, the partial interpretation $\{(s_i, f_j), (s_k, f_l)\}$, where f_j and f_l are model vertices, is feasible only if $\|d(s_i, s_k) - d(f_j, f_l)\| < \epsilon$, where $d(\cdot, \cdot)$ denotes Euclidean distance. The constant term ϵ is a way of accounting for errors in sensing or modeling. (Similar constraints can be derived for other feature types; angle constraints involving triples of matchings are also possible.) Note that the graph G of the distance constraints between matchings is exactly the complement of the docking graph defined in (Kuhl, Crippen, & Friesen 1984); therefore, a maximal independent set of vertices in G corresponds to a maximal clique in the docking graph, and hence to an interpretation with the maximum possible number of matches.

The algorithm constructs an interpretation tree in a systematic way by matching each image data point in turn to a feature of the model. The ith level of the tree contains attempted matchings of the ith data point. From a given node N, the subtree corresponding to a new matching (s_i, f_j) is only generated if (s_i, f_j) is compatible with *all* of the matchings already made on the path from the root of the tree to N. This tree-building process continues down for as many levels as there are data points. Although the size of the tree could grow exponentially with the number of data points, in practice the geometry imposes sufficiently many constraints that the tree remains quite small.

In the special case that all the object features are vertices, building the entire tree is not necessary. Since three independent matchings completely determine, up to a reflection, a rigid body transformation, three levels of the tree suffice. At a leaf of the three-level tree, the remaining $m - 3$ data points can be checked by computing and applying the two possible

transformations. This is, in fact, *exactly* the same as the $O(m^3 n^3 \min(m,n))$ brute-force algorithm for protein docking mentioned above (Kuhl, Crippen, & Friesen 1984)! The second docking algorithm from that paper corresponds here to finding maximal interpretations, i.e. deepest leaves in the full interpretation tree. The degree of similarity in the work described in these two papers is striking. That they were both published in the same year suggests that the authors might have found an opportunity to share ideas very useful.

The interpretation tree idea has been applied to many different variations on the theme of object recognition and localization in robotics (for a discussion, see (Faugeras 1993)). Another successful paradigm is called *geometric hashing* (Lamdan & Wolfson 1988). The idea is to invert the natural "imaging" mapping from the set of possible interpretations of the data to the set of sensor readings. The inversion is made possible by a hash table data structure. Conceptually, the sensor data is used as an index to the table, and interpretations are stored at each entry. To make such table lookups possible, the method uses *transformation-invariant* representations of objects. If one chooses a fixed, minimum-sized tuple of object features which can uniquely determine a basis for a coordinate system for the object, all the other features can be described with respect to that basis. The size of a minimal basis depends on both the dimension of the object space and the type of transformation implied by the imaging process. For example, for point features in three dimensions with rigid body transformations, three non-collinear points are necessary and sufficient. In geometric hashing, the hash table is filled with one entry for every feature coordinate value, appropriately quantized, arising from every possible basis tuple. In the entry for a coordinate is stored the basis tuple which gave rise to that coordinate value (as well as the model, in the case that there are more than one). The recognition step, then, given an image, picks a tuple of *image* basis points, computes the coordinates of all the other image points with respect to that basis, and looks in the table under each, tallying "votes" for the favorite basis among those retrieved from the table. Given such a candidate object basis, the transformation which maps the image basis to it is unique and easily computed. This transformation is verified against the remaining scene data; if the verification fails, the process repeats using other tuples of image points.

If no clear favorite emerges from one image basis tuple, others are used and cumulative votes are tallied. Given the image basis points and the elected overall favorite object basis, an interpretation (i.e. a transformation) is uniquely determined.

One advantage of this approach is that the hash table for a set of possible models can be computed *off-line*, and can then be reused on many different images. In a typical industrial manufacturing application there are only a small number of possible objects, so this ability to pre-process the models is very desirable. (Note that an interpretation tree cannot be precomputed, since its branching topology depends on both the model *and* a particular image.) During the on-line recognition step, in the worst case all possible image basis tuples may have to be tried; in this case the precomputation saves nothing. But if many of the points in the image arise from just one of the modeled objects, the first few tuples tried are very likely to yield a good match right away and recognition will be efficient.

The general technique of geometric hashing has found numerous applications in robotics, and often yields efficient and robust algorithms (see for example (Wallack, Canny, & Manocha 1993)). It is also a natural candidate for application to matching problems in molecular biology, as first suggested in (Nussinov & Wolfson 1991). That paper describes a geometric hashing algorithm for detecting structural motifs in proteins. The same approach has also yielded interesting results for the docking problem (Norel *et al.* 1994); a straightforward extension to the technique also solves a generalization of the docking problem, in which the ligand may have up to 3 internal rotational degrees of freedom, centered on a designated "hinge" atom of the ligand (Sandak, Nussinov, & Wolfson 1994).

Structure Prediction

We use the term *structure prediction* to refer to a broad class of problems in the conformational analysis of molecules. The goal of this type of problem is to compute one or more three-dimensional shapes which a molecule may adopt, given a description of the molecule in terms of some lower-level information such as a covalent structural formula or an amino acid sequence. Acceptable 3D structures must not only be viable in a geometric sense (e.g. not have overlapping atoms), but in most applications they must also satisfy some secondary criteria such as minimizing a free-energy function.

Structure prediction can be described as a search problem. The search space is the space of "all possible" three-dimensional structures. Such a conformational space may be parameterized in various different ways. The simplest way is to describe molecular conformations by giving three Cartesian coordinate values for each of the atoms in the molecule. Another parametrization uses "internal variables," such as the ϕ

and ψ dihedral angles along the backbone in the case of a peptide chain. Yet another description of molecular conformation gives the set of pairwise distances among the atoms of the molecule.

The choice of parametrization of a space sets limits on the range of what are considered possible structures. For example, the dihedral angle model holds the bond angles and lengths at fixed values, while the Cartesian model allows them to vary. This extra flexibility comes at a price, however, since the parametrization also affects the efficiency of various search strategies. The dimension of the space for the chosen parametrization, in particular, is an important factor in the difficulty of solving any given problem. It has been recommended that searches for low-energy conformations run initially using a model with as few parameters as possible, and then continue if necessary in a larger search space as a means of refinement (Gō & Scheraga 1970).

A good survey of conformational search techniques is (Howard & Kollman 1988). The easiest exhaustive searching technique to describe is called *grid search*. The idea is to break the search space into a finite partition by trying each of a small set of discrete values for each parameter of the space. For example, with a dihedral-angle parametrization of a chain molecule, a grid search simply varies systematically each dihedral angle by some fixed increment, and examines each conformation thereby obtained. It has been suggested that a dihedral angle step size of 60° is sufficient for hydrocarbon compounds (Lipton & Still 1988), and it may prove to be possible to use even coarser grids. However, unadorned grid search methods which traverse the entire tree of possibilities inevitably run into a combinatorial brick wall, since the number of conformations which they consider is an exponential function of the number of degrees of freedom of the molecular model. Using a coarser grid only reduces the base of the exponent.

Ultimately, any approach which attempts to examine all geometrically possible conformations must fail for larger molecules. The number of such conformations has been estimated at $(1.7)^n$ for a polypeptide backbone, where n is the number of degrees of freedom, or $(1.4)^n$ if only compact conformations are considered (Dill 1985). One simple approach which abandons the goal of examining all possible conformations is the "build-up" procedure (Gibson & Scheraga 1987). In this approach, systematic search is applied to short segments of the molecule, and the molecule is built up by combining only the lowest-energy conformations of these segments.

Two examples of structure prediction problems are the *protein folding* and *ring closure* problems. The long-term goal of protein folding is to predict native protein structures using only knowledge of the amino acid sequence. This ambitious goal has sparked much attention and research (see overview in (Dill 1993)); the best algorithms, however, still run too slowly on large proteins by factors of at least 10^6 for computing a complete and accurate folding. The ring closure problem asks for conformations of cyclic molecules in which the cyclic covalent structure of the molecule is maintained; these molecules are much more modest in size than proteins. Although these two problems are conceptually very similar, they differ in the dimension of the search space by several orders of magnitude, and are therefore amenable to differing approaches. In particular, exhaustive search quickly becomes intractible for the larger problems, and heuristic techniques are necessary.

Ring Closure and Inverse Kinematics

Simply stated, the ring closure problem is to compute conformations of a molecule with a cyclic structure in which the constraints imposed by the bond lengths and angles are respected. Because these are considered fixed quantities, the molecule is parametrized by its dihedral angles.

The seminal paper in this area is (Gō & Scheraga 1970). Its authors were interested in finding valid conformations of a single-loop molecule with n of its dihedral angles considered free (any others are considered parts of a rigid chain). They prove that such a molecule has $n - 6$ degrees of freedom. Thus if the values of all but 6 of the angles are held fixed, the remaining 6 are no longer free but instead are needed to enforce closure of the chain. The paper derives a system of six algebraic equations in six unknowns which describes the chain closure constraint, and proposes an algorithm based on trying different sets of values for different subsets of size $n - 6$ of the free angles, solving the constraint equations for each such partial conformation. They then show how these solutions can be obtained for certain specializations of the problem, such as the special case in which all of the dihedral angles are considered variable (i.e. consecutive angle axes intersect), and the special case of chains with Pauling-Corey geometry. A method for the general case is not developed, and the question of the number of possible solutions of the six equations is not addressed, except in noting that there may be multiple solutions. Using this approach combined with powerful filters to trim the conformation space, (Moult & James 1986) successfully performed exhaustive searches for cyclic molecule models with up to 10 degrees of freedom.

An exact analogy exists between the ring closure

problem and a particular inverse kinematics problem in robotics. Consider a robot arm which consists of seven rigid links connected in series, with a single rotational degree of freedom at each connection (this is the "$6R$ manipulator of general geometry"). The link at one end of this kinematic chain is fixed in space (the base), and the other end is connected to a gripper or tool of some kind (the end effector). Given the angle of each of the six joints of such a robot, the position and orientation of the end effector relative to the base is clearly uniquely determined. The inverse kinematics question asks: For a given desired position and orientation of the end effector, what set(s) of angle values for the joints will achieve this goal? The analogy with ring closure is clear if we replace each link of the robot by a section of a molecular chain with fixed dihedral angles and each joint with a variable dihedral angle. The chain closure condition on this molecule is then just a special case of inverse kinematics, in which the desired pose (position and orientation) of the end effector is the same as that of the base.

The history of work on the $6R$ inverse kinematics problem is about as recent as that of the ring closure problem. Whereas the attention of molecular biologists turned toward higher-dimensional problems and hence away from the algebraic approach initiated by (Gō & Scheraga 1970), robotics researchers pursued algebraic techniques to a quite satisfactory solution. The next two paragraphs sketch the history of this effort.

The earliest published work on $6R$ inverse kinematics is that reported by Donald Pieper in his Ph.D. thesis (Pieper 1968). He developed closed-form solutions for certain special geometries, such as when any three consecutive joint axes intersect in a common point. (Partly because this condition simplifies the analysis of the system so much, many common industrial robots are designed to be in this class; see e.g. (Craig 1989).) Pieper also formulates the general $6R$ problem with a single polynomial in one variable. Solving this polynomial by numerical methods is not possible in practice, since its degree is around 64,000. Subsequently, a non-constructive proof of an upper bound of 32 on the number of possible solutions was given in (Roth, Rastegar, & Scheinman 1974), and (Duffy & Crane 1980) gives a 32^{nd} degree polynomial in the tangent of one of the half-angles, which a numerical method could in principle use to solve for the joint angles. A different constructive approach was reported in (Tsai & Morgan 1985). There the problem is cast as a system of eight second-degree polynomials in eight variables, and a homotopy method (also called a polynomial continuation method) is used to solve this system. (Primrose 1986) later showed that 16 of the solutions to the 32^{nd} degree polynomial of (Duffy & Crane 1980) must have non-zero imaginary parts, proving an upper bound of 16 on the number of real solutions. Subsequently a 16^{th} degree "input-output" polynomial was derived in (Raghavan & Roth 1989). Examples of general $6R$ manipulators and end-effector poses which have 16 distinct real solutions are known (Manseur & Doty 1989); in this sense, 16 is a tight upper bound. (Wampler & Morgan 1991) uses this bound to develop a polynomial continuation method which tracks exactly 16 paths which are guaranteed to converge to all 16 solutions; they report success on an extensive suite of test problems, with most running times around 6 to 8 seconds on an IBM 370 (the slowest took 20 seconds).

Recently, a much more efficient algorithm was presented in (Manocha & Canny 1992). They start with a system of 14 equations presented in (Raghavan & Roth 1989), and simplify it through a series of symbolic precomputations. Then given the parameters (link lengths, twist angles, etc.) of a particular 6R manipulator, their algorithm uses matrix operations and numerical elimination to transform the problem to one of computing the eigenvalues and eigenvectors of a 24 by 24 matrix. This latter problem has been well studied, and efficient and numerically stable solutions are available (e.g. in the package LAPACK, described in (Anderson *et al.* 1992)). The running times reported on some of the same test cases used in (Wampler & Morgan 1991) are 11 *milliseconds* on average (the slowest was 25 milliseconds). In most cases, ordinary double floating-point arithmetic was sufficient to compute the answers to a very high degree of accuracy, making less efficient variable-precision arithmetic unnecessary.

The case of kinematic chains with 6 degrees of freedom is important in robotics, since this is the minimum number necessary for the robot to span a full-rank subset of position-orientation space ($SE(3)$). Indeed, the general $6R$ case was once dubbed the "Mount Everest" of inverse kinematic problems (Freudenstein 1973)! Although kinematic chains with 7 degrees of freedom have been considered (Waldron & Reidy 1986), the complexity of controlling such robots makes them impractical, and hence higher-dimensional inverse kinematics has not received as much attention. In contrast, in molecular biology there is no such intrinsic reason to stop at 6 degrees of freedom. On the contrary, the ability to compute closed-ring conformations for larger rings would be very useful. One obvious approach to this is to apply the original proceedure of (Gō & Scheraga 1970), but using the fast inverse kinematics procedure of (Manocha & Canny 1992) to close the loop. For example, to generate conformations of

a cyclic molecule with 8 free dihedral angles, one could choose 2 of them as basis variables and apply (Manocha & Canny 1992) to solve for the other 6 for each point on a two dimensional grid. But a more interesting direction to pursue is to apply more purely algebraic techniques to compute the $(n-6)$-dimensional variety corresponding to ring closure. The recently developed algorithms for computing solutions to sparse systems of polynomial equations reported in (Emiris & Canny 1993) and (Emiris 1993) may prove useful in this endeavor.

Protein Folding

The protein folding problem is to predict what three-dimensional structure a protein, described by its amino acid sequence, will take. Most proteins of any interest have far too many conformational degrees of freedom for even considering exhaustive search methods, which have exponential running times. However, this may not be such a bad thing. Folded proteins are very compact structures, with mainly hydrophilic side chains on the exterior and clusters of hydrophobic ones inside (McCammon & Harvey 1987). Therefore, searching in those regions of conformational space in which the protein doesn't have these characteristics is destined to be fruitless. Also, proteins in nature have a remarkable ability to fold relatively quickly to their unique thermodynamically minimal conformations; recent evidence suggests that many of the important secondary structural elements of proteins form within the first few milliseconds during folding (Dyson & Wright 1993). This "existence proof" of a fast algorithm strongly suggests that exhaustive search is overkill.

Some protein folding methods attempt to imitate the success of nature directly. One such approach relies on searching a database of known molecular conformations for similar protein fragments. In the earliest form of this approach, a sequence of amino acid residues comprising an unknown protein fragment is given, a matching algorithm compares it to fragments of the proteins in the database, and the geometry of the best matches is used in predicting the structure of the unknown fragment (perhaps as a starting point for further refinement). The matching problem for sequences (i.e. strings over an alphabet of 20 symbols) is quite straightforward and yields easily to dynamic programming methods (see (Taylor & Orengo 1989) and references therein). However, in a sense this solves the wrong problem, since there are examples of pairs of structurally similar proteins which have very different sequences. Since classical sequence alignment methods need at least 25—30% sequence identity to detect homology, a recent trend is towards "sequence-structure" alignment ideas, in which a sequence and a structure are compared directly by using potential energy functions informed by a library of known protein structures; see (Wodak & Rooman 1993) for a review. Such approaches effectively restrict the search of conformational space to certain subsets which contain known motifs; the challenge is to obtain subsets small enough to be tractably searched, and large enough to contain the right answer.

A general approach which holds great promise for the protein folding problem is molecular dynamics simulation. We defer its discussion to the next section, since it is a technique of very broad applicability, not only for molecular simulations but in robotics as well.

Dynamics Simulation

Dynamics simulation is the general approach of predicting the dynamic behavior of physical systems by simulation, based on a mathematical model of the system and its dynamics. It is sufficiently general that it could in principle be used to solve all of the biology problems discussed in this paper so far; it also has a wide range of applications in robotics and computer graphics. The basic algorithm repeatedly makes changes to the system over a sequence of discrete time steps. If the time step chosen is small relative to the period of the highest-frequency motion in the system, such simulations can remain highly accurate even after many steps (assuming the mathematical model of the physics of the system is correct).

For simulations of macroscopic rigid multibody systems (such as robots), the appropriate model is described by classical Newtonian dynamics, which yields a system of equations of motion for the rigid bodies in the physical system. These equations, together with the constraints implied by the topological and kinematic structure of the system, can be used in deriving algorithms both for simulation, which is an important component of off-line robot programming systems, and for controlling actual robots. A good recent text which gives a modern treatment of the mathematical foundations of robot dynamics is (Murray, Li, & Sastry 1994). It turns out that classical dynamics is also sufficient to describe many of the types of processes of functional interest in biomolecular systems as well. (Higher-frequency motions, such as bond stretching, should properly be simulated by quantum dynamics; this topic is beyond the scope of this paper.) An excellent introduction to both the theoretical and algorithmic aspects of molecular dynamics is (McCammon & Harvey 1987).

Because biological applications of dynamics simulations generally involve minimization of an energy func-

tion, potential functions are often used in deriving the force terms of the equations of motion. The most common type of potential function used in molecular dynamics studies is the "molecular mechanics" type. Such a function expresses the potential energy as a sum of terms deriving from the mechanics of simpler systems, such as springs. There are *bonded* terms contributed by the bonds and angles of the covalent structure, and *nonbonded* terms modelling pairwise interactions between atoms not directly bonded. For example, a typical term for a bond length or angle is a Hooke's law spring term, which is a reasonable approximation to bond fluctuations at normal biological temperatures. The energy contribution of changes in dihedral angles can be modelled as a periodic function. The nonbonded terms are especially important in protein folding studies because of the high packing density of proteins.

The number of the bonded terms in the energy function is linear in the number of atoms. There are, however, a quadratic number of nonbonded pairwise interactions between atoms. For reasons of efficiency it is necessary to avoid computing terms for all of these. Since at large distances the nonbonded contributions to total energy asymptotically approach zero, it is customary to truncate beyond a certain distance. (Tasaki, McDonald, & Brady 1993) discusses ways to do this truncation gracefully, as well as the tradeoff between the cutoff distance and the accuracy of the simulation. Most MD systems maintain for each atom a list of the atoms considered close enough to contribute a nonbonded term; this list may or may not get updated periodically, depending on the time scale of the simulation. The naive way to do the update is to reconsider all possible pairs of atoms, which takes time quadratic in the number of atoms. Yet under a certain reasonable "space-filling" assumption about the atoms, the number of immediate neighbors within any fixed radius of an atom is constant (with respect to the total number of atoms). Because of this, a data structure from computational geometry called the "fat" space partition (Overmars 1992) could potentially be employed to reduce the update time to linear.

Given a potential function describing the energy of a system as a function of its state parameters, dynamics simulations proceed in three steps, each of which is applied to each rigid body in the system at every time step:

1. compute the net force and moment acting on the body;
2. compute the linear and angular acceleration of the body due to these forces; and
3. compute the new position of the body for the next time step.

When a potential function is used, the force acting on a point is computed as the negative gradient of the potential energy with respect to the coordinates of that point. The second step above is an application of Newton's and Euler's equations of motion, and the third involves applying a numerical integration technique.

Under the molecular mechanics potential function model, representing a molecule by its atomic coordinates is a natural choice. It makes evaluating the potential function at any given conformation very simple. Also, because this form is explicitly differentiable, the force computation is straightforward. This choice of parametrization is not without problems, however. An interesting point about molecular dynamics is that the highest frequency motions are the small bond length and angle oscillations. This means that the very factor which most limits the size of the time step which can be taken (and hence the total simulation time possible) is also the factor which contributes the least to the dynamic behavior which is generally of interest. This effect can be reduced by imposing constraints on the distances and applying an iterative constraint-satisfaction procedure after each time step (this is the essence of the SHAKE algorithm described in (Ryckaert, Ciccotti, & Berendsen 1977)); however, the computational problems of this approach make it impractical for large molecules.

A promising alternative approach is that of (Jain, Vaidehi, & Rodriguez 1993), which parametrizes the molecule with generalized internal variables (e.g. dihedral angles), thus allowing bond lengths and angles to be constrained directly. The algorithm works not only for chain molecules, but for more general tree-topology molecules as well. They derive a system of equations of motion for the molecule using a Newton-Euler approach, and give an algorithm for its solution which runs in time linear in the number of degrees of freedom. This algorithm is based on a body of previous work by the same authors on the dynamics of robotic systems; see e.g. (Rodriguez, Jain, & Kreutz-Delgado 1992).

Conclusions

In this paper, we have discussed similarities between various problems in molecular biology and robotics. In two of these areas, namely geometric matching and dynamics simulation, interdisciplinary research efforts are already yielding promising results for structural biology (Norel *et al.* 1994), (Jain, Vaidehi, & Rodriguez 1993). We have also discussed the highly analogous situation between the problems of molecular ring closure

and the inverse kinematics of serial manipulators. The methods developed for the latter problem (Manocha & Canny 1992) may well find useful application to the former. Other analogies are not hard to envision; for example, the "distance geometry" problem (Crippen & Havel 1988), which arises in the context of interpreting NMR data, bears a certain resemblance to the forward kinematics problem for parallel manipulators such as the Stewart platform (Murray, Li, & Sastry 1994). It is unclear whether such analogies will ultimately prove to be useful; but if nothing else, looking at one of these problems in one field from the perspective of the other should help to suggest new avenues of research to explore.

References

Anderson, E.; Bai, Z.; Bischof, C.; Demmel, J.; Dongara, J.; Croz, J. D.; Greenbaum, A.; Hammarling, S.; and Sorensen, D. 1992. *LAPACK User's Guide, Release 1.0.* Philadelphia: SIAM.

Cherfils, J., and Janin, J. 1993. Protein docking algorithms: Simulating molecular recognition. *Current Opinion in Structural Biology* 3:265–269.

Craig, J. J. 1989. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley Publishing Company, Inc., second edition.

Crippen, G. M., and Havel, T. F. 1988. *Distance Geometry and Molecular Conformation.* New York: Wiley.

Dill, K. A. 1985. Theory for the folding and stability of globular proteins. *Biochemistry* 24:1501–1509.

Dill, K. A. 1993. Folding proteins: Finding a needle in a haystack. *Current Opinion in Structural Biology* 3:99–103.

Duffy, J., and Crane, C. 1980. A displacement analysis of the general spatial 7-link, $7R$ mechanism. *Mechanism and Machine Theory* 15:153–169.

Dyson, H. J., and Wright, P. E. 1993. Peptide conformation and protein folding. *Current Opinion in Structural Biology* 3:60–65.

Emiris, I., and Canny, J. 1993. A practical method for the sparse resultant. In Bronstein, M., ed., *ACM International Symposium on Symbolic Algebra and Computation*, 183–192.

Emiris, I. 1993. An efficient computation of mixed volume. Technical Report 734, Computer Science Division, U.C. Berkeley, Berkeley, California.

Faugeras, O. 1993. *Three-Dimensional Computer Vision.* Cambridge, MA: The MIT Press.

Freudenstein, F. 1973. Kinematics: Past, present, and future. *Mechanism and Machine Theory* 8(2):151–161.

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness.* San Francisco: Freeman.

Gibson, K. D., and Scheraga, H. A. 1987. Revised algorithms for the build-up procedures for predicting protein conformations by energy minimization. *Journal of Computational Chemistry* 8:826–834.

Gō, N., and Scheraga, H. A. 1970. Ring closure and local conformational deformations of chain molecules. *Macromolecules* 3(2):178–194.

Grimson, W. E. L., and Lozano-Pérez, T. 1984. Model-based recognition and localization from sparse range or tactile data. *The International Journal of Robotics Research* 3(3):3–35.

Howard, A. E., and Kollman, P. A. 1988. An analysis of current methodologies for conformational searching of complex molecules. *Journal of Medicinal Chemistry* 31(9):1669–1675.

Jain, A.; Vaidehi, N.; and Rodriguez, G. 1993. A fast recursive algorithm for molecular dynamics simulation. *Journal of Computational Physics* 106(2):258–268.

Janin, J., and Chothia, C. 1990. The structure of protein-protein recognition sites. *Journal of Biological Chemistry* 265:16027–16030.

Jiang, F., and Kim, S. H. 1991. "Soft Docking": Matching of molecular surface cubes. *Journal of Molecular Biology* 219:79–102.

Kuhl, F. S.; Crippen, G. M.; and Friesen, D. K. 1984. A combinatorial algorithm for calculating ligand binding. *Journal of Computational Chemistry* 5(1):24–34.

Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; and Ferrin, T. E. 1982. A geometric approach to macromolecule–ligand interactions. *Journal of Molecular Biology* 161:269–288.

Kuntz, I. D. 1992. Structure-based strategies for drug design and discovery. *Science* 257:1078–1082.

Lamdan, Y., and Wolfson, H. J. 1988. Geometric hashing: A general and efficient model-based recognition scheme. In *Second IEEE International Conference on Computer Vision*, 238–249.

Lin, S. L.; Norel, R.; Fischer, D.; and Wolfson, H. J. 1994. Molecular surface representations by sparse critical points. *Proteins: Structure, Function, and Genetics* 18:94–101.

Lipton, M., and Still, W. C. 1988. The multiple minimum problem in molecular modeling: Tree searching internal coordinate conformational space. *Journal of Computational Chemistry* 9:343–355.

Manocha, D., and Canny, J. F. 1992. Real time inverse kinematics of general 6R manipulators. In *IEEE Conference on Robotics and Automation*, 383–389.

Manseur, R., and Doty, K. L. 1989. A robot manipulator with 16 real inverse kinematic solution sets. *International Journal of Robotics Research* 8(5):75–79.

McCammon, J. A., and Harvey, S. C. 1987. *Dynamics of Proteins and Nucleic Acids.* Cambridge, England: Cambridge University Press.

Moult, J., and James, M. N. G. 1986. An algorithm for determining the conformation of polypeptide segments in proteins by systematic search. *Proteins: Structure, Function, and Genetics* 1:146–163.

Murray, R. M.; Li, Z.; and Sastry, S. S. 1994. *A Mathematical Introduction to Robotic Manipulation.* CRC Press, Inc.

Navia, M. A., and Murcko, M. A. 1992. Use of structural information in drug design. *Current Opinion in Structural Biology* 2:202–210.

Norel, R.; Fischer, D.; Wolfson, H. J.; and Nussinov, R. 1994. Molecular surface recognition by a computer vision technique. *Protein Engineering* 7(1):39–46.

Nussinov, R., and Wolfson, H. J. 1991. Efficient detection of three-dimensional motifs in biological macromolecutes by computer vision techniques. *Proceedings of the National Academy of Sciences, USA* 88:10495–10499.

Overmars, M. 1992. Point location in fat subdivisions. *Information Processing Letters* 44:261–265.

Pieper, D. 1968. *The Kinematics of Manipulators under Computer Control.* Ph.D. Dissertation, Stanford University, Palo Alto, CA.

Primrose, E. J. F. 1986. On the input-output equation of the general $7R$ mechanism. *Mechanism and Machine Theory* 21:509–510.

Raghavan, M., and Roth, B. 1989. Kinematic analysis of the $6R$ manipulator of general geometry. In *International Symposium on Robotics Research*, 314–320.

Rodriguez, G.; Jain, A.; and Kreutz-Delgado, K. 1992. Spatial operator algebra for multibody system dynamics. *The Journal of the Astronautical Sciences* 40(1):27–50.

Roth, B.; Rastegar, J.; and Scheinman, V. 1974. On the design of computer controlled manipulators. In *On the Theory and Practice of Robots and Manipulators.* New York: Springer-Verlag. 93–113.

Ryckaert, J. P.; Ciccotti, G.; and Berendsen, H. J. C. 1977. Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *Journal of Computational Physics* 23:327–341.

Sandak, B.; Nussinov, R.; and Wolfson, H. J. 1994. 3-D flexible docking of molecules. In *Workshop on Shape and Pattern Matching in Computational Biology.* Seattle, WA: IEEE.

Shoichet, B. K., and Kuntz, I. D. 1991. Protein docking and complementarity. *Journal of Molecular Biology* 221:327–346.

Tasaki, K.; McDonald, S.; and Brady, J. W. 1993. Observations concerning the treatment of long-range interactions in molecular dynamics simulations. *Journal of Computational Chemistry* 14(3):278–284.

Taylor, W. R., and Orengo, C. A. 1989. Protein structure alignment. *Journal of Molecular Biology* 208:1–22.

Tsai, L.-W., and Morgan, A. P. 1985. Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods. *Journal of Mechanisms, Transmissions, and Automation in Design* 107:189–200.

Waldron, K., and Reidy, J. 1986. A study of kinematically redundant manipulator structure. In *IEEE Conference on Robotics and Automation.*

Wallack, A.; Canny, J.; and Manocha, D. 1993. Object localization using crossbeam sensing. In *IEEE Conference on Robotics and Automation*, volume 1, 692–699.

Wampler, C., and Morgan, A. 1991. Soving the $6R$ inverse position problem using a generic-case solution methodology. *Mechanism and Machine Theory* 26(1):91–106.

Wodak, S. J., and Rooman, M. J. 1993. Generating and testing protein folds. *Current Opinion in Structural Biology* 3:247–259.