Molecular Disassembly with RRT-like Algorithms

Juan Cortés, Léonard Jaillet and Thierry Siméon

Abstract— **This paper addresses the problem of computing pathways for a ligand to exit from the active site of a protein. Such problem can be formulated as a mechanical disassembly problem for two articulated objects. Its solution requires searching paths in a constrained high-dimensional configuration-space. Indeed, the ligand passageway inside the protein is often extremely cluttered so that current path planning techniques are unable to solve the disassembly problem in reasonable computing time. The techniques presented in this paper are based on the RRT algorithm. First we discuss some simple and general modifications of the basic algorithm that significantly improve its performance. Then we describe a new variant of the planner that treats ligand and protein degrees of freedom separately. This new algorithm outperforms the basic RRT, particularly for very constrained problems, and is able to handle models with hundreds of degrees of freedom. We analyze the effects of each RRT variant via several examples of different complexity. Although discussions and results of this paper focus on molecular models, the ideas behind the algorithms are general and can be applied to path planners for disassembling articulated mechanical parts.**

I. Introduction

The computational analysis of molecular interactions in biological systems is a key instrument for the understanding of life. In this framework, the present paper focuses on the study of protein-ligand interactions [4]. Most of the computational approaches to this problem address a static view of the molecular recognition. However, several studies tend to show that the ligand access/exit to the protein active site can be very important for the understanding of the biological mechanism [20], [13]. The difficulty is that computing the pathway of a ligand to go out from a deep active site to the surface of a protein (or vice versa) with "classic" molecular modeling methods [24] is too computationally expensive.

For facing the complexity of computing molecular motions, molecules can be modeled as articulated mechanisms [21], [27] and efficient path planning algorithms can be used to explore their conformational changes [3], [2], [8], [10]. In this paper, the protein-ligand exit problem is formulated as a mechanical disassembly problem for articulated objects (see Section II) and an RRT-like algorithm is proposed for finding solution pathways.

The RRT algorithm, introduced in [17], has been widely studied and applied to different types of problems in the last years (see `http://msl.cs.uiuc.edu/rrt/` for a general survey). Numerous variants have been proposed to improve its performance in general cases (e.g. [18], [5], [25], [19], [26], [15]) or for particular applications (e.g. [12], [16], [11]). Section III reminds the principle of the RRT algorithm, discusses some basic improvements and analyzes their effect in the constrained high-dimensional problems addressed in this paper. These basic RRT variants perform reasonably well for moderately complex instances of the protein-ligand disassembly problem. However, they fail in more difficult cases such as the one illustrated in Fig. 1. In this example, the ligand is extremely cluttered in the protein active site and significant motions of flexible parts of the protein are necessary to extract the ligand.

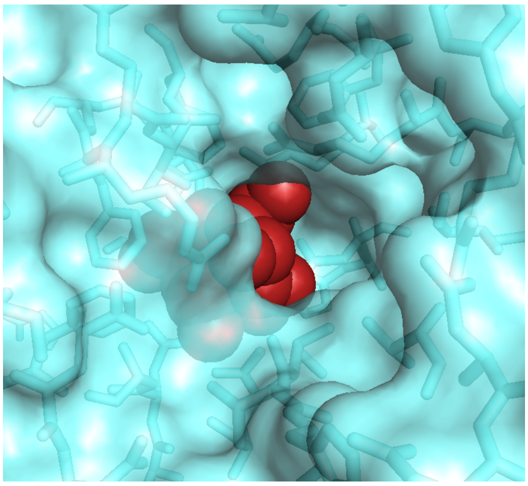

Fig. 1. Model of a ligand (red-colored) in the active site of a protein. The passageway from the active site to the surface of the protein is very narrow. Protein side-chain motions are necessary to permit the ligand exit.

We propose a new RRT variant, called Manhattan-like RRT (ML-RRT), for disassembly path planning of articulated objects. The particularity of this algorithm is that the motions of the different parts are decoupled. Indeed, for the present application, ligand and protein degrees of freedom are treated at different levels. The ligand motion is privileged, while flexible parts of the protein only move if they hinder the ligand progression. This new algorithm presents two advantages with respect to the basic RRT. First, the computing time and its variance are notably reduced. And second, but not less important, the flexible parts that have to move for finding a solution path are automatically identified. Thus, the planner is able to handle models involving hundreds of potential degrees of freedom, avoiding user intervention to select the important ones. The ML-RRT algorithm is presented in Section IV. This section also shows results that reflect the performance gain with respect to the improved RRT discussed in Section III. Several directions for extending ML-RRT to more complex instances of the disassembly problem are mentioned in the concluding section (Section V).

All the authors are with LAAS-CNRS, 7 av. du Colonel-Roche, 31077 Toulouse, France `{jcortes,ljaillet,nic}@laas.fr`

II. Mechanistic Model and Problem Formulation

Within our molecular path planning approach [8], molecules are modeled as articulated mechanisms. Groups of rigidly bonded atoms form the bodies and the articulations between bodies correspond to bond torsions. These torsions are the molecular degrees of freedom. The atoms are represented by spheres. Considering a geometric interpretation of the van der Waals repulsive force, which is the most important contributor (at short distance) to the molecular interaction energy, the spheres associated with non-bonded atoms cannot overlap. Such constraint is a powerful filter usually applied in conformational search methods [9]. This collision-avoidance restriction is the motion constraint imposed to the mechanistic molecular model. Using such a simplified model of molecules, one can formulate the problem of computing the pathway to dissociate a molecular complex as a disassembly problem for articulated mechanisms.

The work presented in this paper focuses on protein-ligand complexes. A protein is a biological macromolecule composed by one or several long polypeptide chains, generally folded in a globular manner[1]. The mechanical model of a polypeptide is composed by a set of kinematic chains, which match the bonds of the chemical representation: the main-chain, or *backbone*, and the *side-chains* of the amino acid residues. In the present work, the protein backbone is considered as a rigid body with (possibly hundreds of) articulated side-chains. The ligand is a relatively small molecule, containing in general a few tens of atoms. We consider fully flexible ligands, which are modeled as free-flying articulated objects.

Figure 2 illustrates the mechanical disassembly problem treated in this paper. By analogy to the molecular disassembly problem, the small moving object is the ligand and the big object is the protein. The articulated parts in the big object are the protein side-chains.

[1] See molecular modeling textbooks (e.g. [24]) for a detailed structural description of proteins.

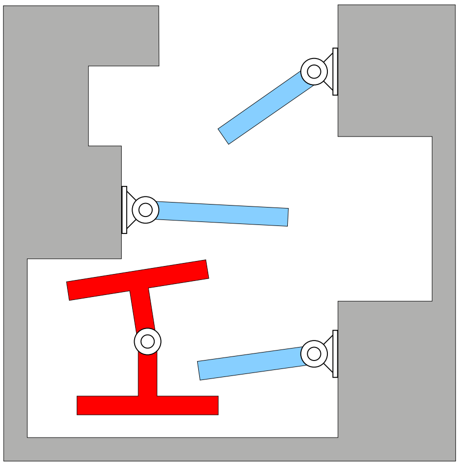

Fig. 2. Disassembly problem for two articulated objects. The problem consists in finding a path to extract the small (red) object from the big one. Both objects have articulated parts.

III. The RRT Algorithm and Some Basic Improvements

The techniques presented in this paper are based on the RRT algorithm. This section first reminds the basic RRT principle and discusses some issues. Then we propose simple modifications and show how they improve the performance of the algorithm through the examples at the end of the section.

A. Basic RRT Algorithm

The basic principle of the RRT algorithm [17] is to incrementally grow a random tree rooted at the initial configuration $\mathbf{q}_{init}$ to explore the reachable configuration-space and find a feasible path connecting $\mathbf{q}_{init}$ to a goal configuration $\mathbf{q}_{goal}$. At each iteration, the tree is expanded toward a randomly sampled configuration $\mathbf{q}_{rand}$. This random sample is used to simultaneously determine the tree node to be expanded and the direction in which it is expanded. Given a distance metric in the configuration-space, the nearest node $\mathbf{q}_{near}$ in the tree to the sample $\mathbf{q}_{rand}$ is selected and an attempt is made to expand $\mathbf{q}_{near}$ in the direction of $\mathbf{q}_{rand}$. For holonomic systems, the expansion procedure can be simply performed by moving on the straight-line segment between $\mathbf{q}_{near}$ and $\mathbf{q}_{rand}$. If the expansion succeeds, a new node $\mathbf{q}_{new}$ and a feasible local path from $\mathbf{q}_{near}$ are generated. The key idea of this expansion strategy is to bias the exploration toward unexplored regions of the space. Hence, the probability that a node will be chosen for an expansion is proportional to the volume of its Voronoi region (i.e. the set of points closer to this node than to the others). Therefore RRTs are biased by large Voronoi regions to rapidly explore before uniformly covering the space.

Different strategies can be adopted for the design of path planners based on the RRT algorithm [18]. One can choose between an unidirectional or a bidirectional exploration strategy. An unidirectional planner develops a single tree from one of the two given configurations, $\mathbf{q}_{init}$ or $\mathbf{q}_{goal}$, until the other configuration is reached, while a bidirectional technique constructs one tree from $\mathbf{q}_{init}$ and another from $\mathbf{q}_{goal}$ until the two trees meet at a point. One can also choose a more or less greedy strategy for the expansion procedure. In the basic RRT algorithm, a single expansion step of fixed distance is performed. In the more greedy *RRT-Connect* variant, the expansion step is iterated while feasibility constraints (e.g. collision avoidance) are satisfied. The strategy selection depends on the nature of the path planning problem to be solved. For the disassembly problems treated in this paper, we use the unidirectional RRT-Connect variant. The choice of a unidirectional exploration seems obvious, since the initial configuration is highly constrained while the goal is in an unconstrained region and can be fuzzily defined (e.g. ligand outside the protein). RRT-Connect is in general more efficient than the single-step version for systems without differential constraints [18].

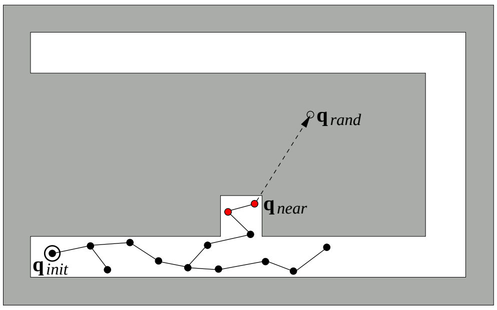

Fig. 3. Illustration of an RRT with exhausted nodes. The (red-colored) nodes in the small side corridor will be frequently selected for expansion, thus hindering the tree development.

B. Metric Sensitivity: A Well-Known Drawback

The weakness of the RRT algorithm is its high sensitivity to the distance metric used to select the nearest neighbor to $\mathbf{q}_{rand}$, $\mathbf{q}_{near}$. Since $\mathbf{q}_{near}$ is the node to be extended, the performance of the algorithm strongly depends on the metric. The ideal metric should consider motion constraints (kinematic constraints, obstacles, ...). However, devising such an ideal metric is at least as difficult as solving the path planning problem itself. Therefore, an Euclidean metric in the configuration-space is normally used. The use of this simple distance metric can however lead to an undesired behavior of the planner, as illustrated in Fig. 3. Some nodes that are close to the border between feasible and unfeasible regions of the configuration-space (i.e. the border of the C-obstacles) are repeatedly selected for futile expansion. We call such nodes *exhausted nodes*, in reference to [5]. The issue of exhausted nodes is particularly important when using the RRT-Connect variant, which tends to generate nodes close to the border.

Another concern related to the use of a simple Euclidean metric in the configuration-space is that, in some cases, particularly for articulated mobile systems, the distance metric is not representative of the path planning problem. Consider for example the manipulator in Fig. 4. Two significantly different robot configurations are at the same distance than two more similar ones (i.e. inducing a smaller motion). Several approaches (see [1] for example) have been proposed to design more sophisticated metrics aiming to find a solution to this issue, usually applying weights to balance the different influence of the degrees of freedom in the system configuration. However, in high-dimensional spaces, it is difficult to find the adequate metric showing a good trade-off between computational complexity and accuracy.

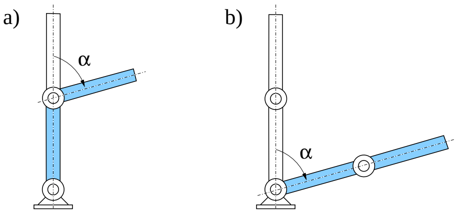

Fig. 4. Two configurations of a 2-DOF manipulator. Considering an Euclidean metric in the configuration-space, both configurations are at the same distance to the reference configuration (white drawings). However, the induced motion is much smaller for (a) than for (b).

Next we discuss simple improvements of the basic RRT algorithm that notably limit the pathological behaviors due to the Euclidean metric.

C. Avoiding Exhausted Nodes

We first describe two different strategies to circumvent the undesired behavior of the RRT algorithm occasioned by exhausted nodes. The first consists in removing from the search tree nodes that are considered as exhausted, as initially proposed in [5]. For each node, the number of consecutive times that its expansion fails is counted. When the counter reaches a given limit number l, the node is considered to be exhausted and it is no longer selected. A similar strategy discussed in [16] uses an expansion failure counter to devise a weighted metric for the nearest node selection.

The second strategy consists in selecting $\mathbf{q}_{near}$ at random among the k nearest neighbors. In this way, a chance for expansion is given to nodes that are not in the tree boundary, and which, in the basic algorithm, will be selected for refining the space coverage rather than to explore uncovered regions. This idea has been applied in other works (e.g. [25]).

The results presented below (see Section III-E) show that the two above strategies improve the performance of the basic RRT against the exhausted-node issue. Since both strategies have a very low computational cost, they can be applied simultaneously.

D. Using a Simplified Task-Adapted Metric

For path planning problems involving many degrees of freedom, we propose to use a simplified distance metric that only considers the subset of the most significant degrees of freedom with respect to the task. The most significant parameters for the disassembly of two articulated objects are generally those defining the relative location of the objects. Thus, if one of the objects is considered to be static, a simplified metric can be designed by only measuring the distance in SE(3) for a reference frame associated with the center of mass of the mobile object (the ligand in our case). The degrees of freedom of the articulated parts in the static object (i.e. the protein side-chains) and the internal degrees of freedom of the mobile object (i.e. the ligand torsions) are neglected within this distance metric.

The interest of considering approximate metrics for nearest-neighbor search in the framework of sampling-based path planning has been recently shown in [22]. It has been empirically demonstrated that, in high-dimensional spaces, the use of a simplified metric provides a considerable computational gain without a significant loss of accuracy. Besides, the results presented below show that an adequate selection of the parameters involved in the metric yields a much better performance of the planner.

E. Empirical Performance Analysis

Figure 5 shows numerical results obtained with the basic RRT algorithm and the different enhancements discusses above. We call l-RRT the variant that removes nodes after l consecutive expansion failures and k-RRT the variant choosing $\mathbf{q}_{near}$ at random among the k nearest neighbors; lk-RRT combines both improvements. The variant m-RRT is a basic RRT algorithm using the simplified task-adapted metric. Finally, lkm-RRT incorporates the three improvements. In our implementation, the values of the parameters l and k are currently assigned empirically: l is a constant and its default value is 10, and k is a variable that is computed as the nearest integer greater than or equal to $n_{nodes}/100$, where n_{nodes} is the current number of nodes in the tree.

The algorithms were tested on three molecular disassembly problems, similar to the one illustrated in Fig. 1, with increasing difficulty that we refer to as `ExA`, `ExB` and `ExC`. These problems involve 68 degrees of freedom, which reflect the flexibility of the ligand and of the 18 side-chains in the active site and the access corridor. The increasing complexity of the problems comes from the geometric encumbering of the passage. Numerical results are averaged over 50 runs on an AMD Opteron 148 processor at 2.6 GHz. The values displayed in the table are the following: *avrT* is the average computing time, *SN* is its variance, *Nn* is the average number of nodes in the computed search trees and *Ns* is the average number of samples required for their construction.

Several conclusions can be extracted from the analysis of the numerical results in Fig. 5. First, concerning the exhausted-node avoidance, both variants l-RRT and k-RRT significantly reduce the number of required samples *Ns* in comparison to the basic RRT. Reducing *Ns* implies reducing the number of operations such as nearest-neighbor search and local path validation. A side effect of these variants is that the number of nodes in the tree increases, resulting in more expensive elementary nearest-neighbor searches. The overall gain ranges between 5 and 7, and the results show that k-RRT is more efficient that l-RRT, particularly when the problem difficulty increases. The results obtained with lk-RRT show that the combination of both strategies further improves the performance of the planner, particularly for the most difficult problem. RRT, l-RRT and k-RRT performed very poorly on `ExC` (results are not reported because most of the runs exceeded a fixed time limit) while lk-RRT got solutions in acceptable computing time. The better performance of m-RRT shows the importance of using an adequate distance metric. Compared to the basic RRT (that considers all the degrees of freedom within the metric) the number of samples is reduced while the number of nodes remains similar. Another reason for the decrease of the computing time with m-RRT is that using a simplified metric reduces the cost of the nearest-neighbor search. When the three variants are combined, the performance of the planner improves remarkably. The computing time is reduced by two orders of magnitude for the moderately complex problem `ExB` and the gain is still higher for `ExC`. Finally note that an important benefit of the lkm-RRT variant is to reduce the performance variability, which is an issue for randomized algorithms [14].

		ExA	**ExB**	**ExC**
RRT	*avrT*	43.6	562.2	
	SN	187.7	1650.2	
	Nn	79	136	
	Ns	11186	86303	
l-RRT	*avrT*	7.2	310.9	
	SN	14.9	881.2	
	Nn	263	4534	
	Ns	3093	61448	
k-RRT	*avrT*	6.1	169.3	
	SN	6.9	911.7	
	Nn	202	270	
	Ns	1119	10259	
lk-RRT	*avrT*	6.1	62.2	2446.1
	SN	6.0	237.6	3862.1
	Nn	227	733	4877
	Ns	1242	8250	59478
m-RRT	*avrT*	4.0	17.2	150.4
	SN	7.4	141.3	325.7
	Nn	104	167	753
	Ns	3436	16202	82160
lkm-RRT	*avrT*	1.7	5.1	72.8
	SN	1.4	6.1	88.1
	Nn	234	457	2730
	Ns	1059	3727	24478

Fig. 5. Numerical results for the RRT algorithm with basic improvements.

IV. Decoupling Part Motions

A. Motivation and Overview

In the examples presented in Section III-E, only a few side-chains that may be directly involved in the ligand access are considered to be flexible. Selecting these side-chains without a priori knowledge about the ligand passageway is not a straightforward task that has to be made manually. The main motivation for the development of the algorithm presented in this section is to avoid this kind of user intervention. The basic principle of this new algorithm is to treat ligand and protein degrees of freedom separately. These two sets of parameters are referred to as $\mathbf{q}^{lig}$ and $\mathbf{q}^{prot}$ respectively. The ligand is treated as an *active robot* while the side-chains are considered as *passive robots* that only move when they obstruct the progression of the ligand. Because the paths computed by the algorithm look like Manhattan paths over these two sets of parameters that change alternatively, we call it *Manhattan-like RRT* (*ML-RRT*).

B. Manhattan-like RRT

The ML-RRT algorithm is schematized in Algorithm 1. At each iteration, the ligand motion is computed first. Following the basic RRT principle, a configuration of the ligand $\mathbf{q}^{lig}_{rand}$ is randomly sampled. The function `SampleLigConf` generates uniform samples considering that the ligand can freely translate and rotate in a box enveloping the active site, and allowing (by default) full rotation of all bond torsions. A near neighbor $\mathbf{q}_{near}$ is selected in the current tree, based on a distance metric in the ligand configuration parameters. We use a simplified metric only involving the ligand's center of mass, as discussed in Section III-D. Note that the function `BestNeighbor` also integrates the other basic improvements discussed in Section III-C. `ExpandLigConf` performs the expansion of the ligand configuration using the greedy strategy of RRT-Connect. This function returns a configuration $\mathbf{q}_{new}$ corresponding to the last valid point in the straight-line segment from $\mathbf{q}_{near}$ toward $\mathbf{q}^{lig}_{rand}$. If

Algorithm 1: Construct_ML-RRT

```
input  : the search-space $\mathbf{S}$, the root $\mathbf{q}_{init}$
output : the tree $\tau$
begin
    $\tau \leftarrow$ InitTree($\mathbf{q}_{init}$);
    while not StopCondition($\tau$) do
        $\mathbf{q}^{lig}_{rand} \leftarrow$ SampleLigConf($\mathbf{S}$);
        $\mathbf{q}_{near} \leftarrow$ BestNeighbor($\tau$, $\mathbf{q}^{lig}_{rand}$);
        ($\mathbf{q}_{new}$,$L_C$) $\leftarrow$ ExpandLigConf($\mathbf{q}_{near}$,$\mathbf{q}^{lig}_{rand}$);
        if not TooSimilar($\mathbf{q}_{near}$, $\mathbf{q}_{new}$) then
            AddNewNode($\tau$, $\mathbf{q}_{new}$);
            AddNewEdge($\tau$, $\mathbf{q}_{near}$, $\mathbf{q}_{new}$);
            $\mathbf{q}_{near} \leftarrow \mathbf{q}_{new}$;
        if $L_C \neq \emptyset$ then
            $\mathbf{q}^{prot}_{rand} \leftarrow$PerturbProtConf($\mathbf{q}_{near}$,$\mathbf{S}$,$L_C$);
            $\mathbf{q}_{new} \leftarrow$ ExpandProtConf($\mathbf{q}_{near}$,$\mathbf{q}^{prot}_{rand}$);
            if not TooSimilar($\mathbf{q}_{near}$, $\mathbf{q}_{new}$) then
                AddNewNode($\tau$, $\mathbf{q}_{new}$);
                AddNewEdge($\tau$, $\mathbf{q}_{near}$, $\mathbf{q}_{new}$);
end
```

the expansion is not negligible, a new node and a new edge are added to the tree. The function ExpandLigConf also analyzes the collision pairs yielding the stop of the expansion process. If ligand atoms collide with side-chain atoms, the list of the involved side-chains L_C is returned. Such information is efficiently obtained by BioCD [23], a collision checker tailored to molecular models, thanks to the use of spatially-adapted hierarchical data structures matching the biochemical structure. The list L_C is used in the second part of the algorithm, which performs the protein motion. A configuration of the protein $\mathbf{q}^{prot}_{rand}$ is generated by randomly sampling the configuration of the involved side-chains in a ball around their current configuration. An attempt is then made to generate a new node by expanding $\mathbf{q}_{near}$ toward $\mathbf{q}^{prot}_{rand}$. Note that, if the previous call to ExpandLigConf has been successful, $\mathbf{q}_{near}$ has been updated in order to contain the new ligand configuration.

The goal of the second part the ML-RRT algorithm is to move the side-chains surrounding the ligand aiming to gain clearance for its passage. Note than, in some cases, the motion of some side-chains close to the ligand can be hindered by other farther side-chains. In such situations, a possible solution is to extend the function ExpandProtConf to return the list of side-chains in contact with the moving ones, and to operate in an iterative manner. The implementation of this idea remains for future work.

		ExA	ExB	ExC	ExA+	ExD
ML-RRT	*avrT*	1.2	1.7	9.6	2.6	159.4
	SN	0.4	0.9	9.0	1.1	83.1
	Nn	176	217	1104	142	4428
	Ns	355	732	5382	244	38965

Fig. 6. Numerical results for the ML-RRT algorithm.

C. Results

The performance of ML-RRT has been analyzed using the same examples and in the same conditions than for the basic RRT variants (see Section III-E). Comparing the values in Fig. 6 with those in Fig. 5, one can observe that ML-RRT significantly outperforms the basic RRT variants[2]. The computing time is reduced for all the examples, the improvement rate increasing with the problem difficulty. Besides, the variance of the computing time is reduced, even for the more complex examples. Figure 6 also reports results obtained with ML-RRT on two other examples, ExA+ and ExD, which could not be solved by the previous RRT variants.

ExA+ concerns the same protein-ligand complex than ExA. However, in this case, all 127 side-chains of the protein model are considered to be flexible (instead of 17 for ExA). Consequently, ExA+ involves 332 potential degrees of freedom. Comparing results for ExA and ExA+, one can observe that the average number of samples and nodes in the trees required to solve the problems are analogous, indicating that the performance of the exploration is not affected by the high number of articulated side-chains. The limited slow down of the computing time for ExA+ is due to the higher cost of operations such as spatial position updating, collision detection and memory management.

ExD corresponds to an even more difficult problem. The protein model also contains a high number of articulated side-chains (more than 600), but the exit pathway is much more constrained and requires an important motion of some of these side-chains to let the ligand exit from the deep active site. The image in Fig. 1 corresponds to this example and shows the initial configuration of the ligand inside the protein active site. Among all the side-chains, the ML-RRT algorithm only made 9 of them move for finding the solution path illustrated in Fig. 7. Note the considerable motion of the side-chain located at the middle-top of the image. This side-chain motion, which is known to be biologically important for the protein-ligand interaction, was automatically identified by the algorithm. Finally, mention that although the ligand and the side-chains move alternately in the path obtained by the ML-RRT algorithm, a randomized path smoothing post-processing is performed in the composite configuration-space of the protein and the ligand, so that simultaneous motions are obtained in the final path.

V. Conclusions and Future Work

We have presented a new RRT-like algorithm, called ML-RRT, for disassembly path planning of articulated objects. Although the paper focuses on molecular disassembly applications, the ideas behind ML-RRT are general and the algorithm can be used in robotic applications [6]. Note that results obtained in [6] for solving the mechanical disassembly problem represented in Fig. 2 show a performance gain of two orders of magnitude in relation to RRT.

[2] Because of the algorithmic design of ML-RRT, the values of *Ns* in Fig. 6 only take into account samples of the ligand configuration.

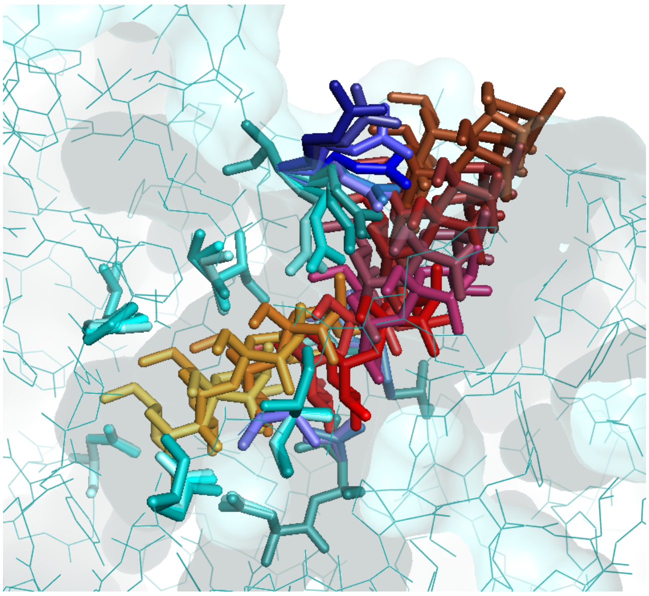

Fig. 7. Solution path for ExD (the initial configuration is shown in Fig. 1). The image shows a transversal cut of the protein active site and the trace of the ligand path. The ligand and the 9 residues with moving side-chains are displayed in stick representation. The configurations of the ligand and the moving side-chains at different moments along the path are colored in red scale and blue scale respectively.

ML-RRT handles models containing hundreds of potential degrees of freedom and only changes the configuration of parts that have to move in the solution path. In the current implementation, a protein side-chain moves if it hinders the progression of the ligand. However, this side-chain motion is not propagated to other surrounding side-chains. As mentioned in Section IV-B, extending ML-RRT to perform such propagation would be rather simple. Although we have still not found a protein-ligand disassembly problem (only involving side-chain motions) making ML-RRT fail, cases involving a concerted motion of several side-chains may exist. Besides, the above mentioned extension may be interesting for other applications. A further extension of ML-RRT involves considering the flexibility of the protein backbone. A first step toward a completely general approach may be to consider protein loop motions. We expect to integrate ML-RRT into the combined molecular-modeling/path-planning approach described in [7], [8].

References

[1] N.M. Amato, O. Bayazit, L. Dale, C. Jones and D. Vallejo, “Choosing Good Distance Metrics and Local Planners for Probabilistic Roadmap Methods”, *IEEE Transactions on Robotics and Automation* 16(4), 2000, pp. 442-447, .

[2] N.M. Amato, K.A. Dill and G. Song, “Using Motion Planning to Map Protein Folding Landscapes and Analyze Folding Kinetics of Known Native Structures”, *Journal of Computational Biology*, vol. 10(3-4), 2003, pp. 239-255.

[3] M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe and C. Varma, “Stochastic Roadmap Simulation: An Efficient Representation and Algorithm for Analyzing Molecular Motion”, *Journal of Computational Biology*, vol. 10(3-4), 2003, pp. 257-281.

[4] H.-J Böhm and G. Schneider, *Protein-Ligand Interactions: From Molecular Recognition to Drug Design*, Wiley-VCH, Weinheim; 2003.

[5] P. Cheng and S.M. LaValle, “Reducing Metric Sensitivity in Randomized Trajectory Design”, *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2001, pp. 43-48.

[6] J. Cortés and T. Siméon, “Disassembly Path Planning for Objects with Articulated Parts”, *IFAC Workshop on Intelligent Assembly and Disassembly*, 2007, submitted.

[7] J. Cortés, T. Siméon, M. Remaud-Siméon and V. Tran, “Geometric Algorithms for the Conformational Analysis of Long Protein Loops”, *Journal of Computational Chemistry*, vol. 25(7), 2004, pp. 956-967.

[8] J. Cortés, T. Siméon, V. Ruiz-deAngulo, D. Guieysse, M. Remaud-Siméon and V. Tran, “A Path Planning Approach for Computing Large-Amplitude Motions of Flexible Molecules”, *Bioinformatics*, vol. 21(Suppl.1), 2005, pp. i116-i125.

[9] M.A. DePristo, P.I.W. de Bakker, S.C. Lovell and T.L. Blundell, “Ab-initio Construction of Polypeptide Fragments: Efficient Generation of Accurate, Representative Ensembles”, *Proteins* 51, 2003, pp. 41-55.

[10] A. Enosh, S.J. Fleishman, N. Ben-Tal and D. Halperin, “Prediction and Simulation of Motion in Pairs of Transmembrane α-Helices”, *Bioinformatics*, vol. 23(2), 2007, pp. e212-e218.

[11] A. Ettlin and H. Bleuler, “Randomised Rough-Terrain Robot Motion Planning”, *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2006, pp. 5798-5803.

[12] E. Ferré and J.-P. Laumond, “An Iterative Diffusion Algorithm for Part Disassembly”, *Proc. IEEE Int. Conf. Robotics and Automation*, 2004, pp. 3149-3154.

[13] D. Guieysse, C. Salagnad, P. Monsan, M. Remaud-Simeon and V. Tran V, “Towards a Novel Explanation of Pseudomonas Cepacia Lipase Enantioselectivity Via Molecular Modelling of the Enantiomer Trajectory Into the Active Site”, *Tetrahedron: Asymmetry*, vol. 14, 2003, pp. 1807-1817.

[14] P. Isto, M. Mantyla and J. Tuominen, “On Addressing the Run-Cost Variance in Randomized Motion Planners”, *Proc. IEEE Int. Conf. Robotics and Automation*, 2003, pp. 2934-2939.

[15] M. Kalisiak and M. van de Panne, “RRT-Blossom: RRT with a Local Flood-fill Behavior”, *Proc. IEEE Int. Conf. Robotics and Automation*, 2006, pp. 1237-1242.

[16] J. Kim, J.M. Esposito and V. Kumar, “An RRT-Based Algorithm for Testing and Validating Multi-Robot Controllers”, *Robotics: Science and Systems I*, S. Thrun, G. Sukhatme, S. Schaal and O. Brock (Eds.), MIT Press, Cambridge, 2005, pp. 249-256.

[17] S.M. LaValle, “Rapidly-Exploring Random Trees: A New Tool for Path Planning”, *TR 98-11*, Computer Science Dept., Iowa State University, 1998.

[18] S.M. LaValle and J.J. Kuffner, “Rapidly-Exploring Random Trees: Progress and Prospects”, *Algorithmic and Computational Robotics: New Directions (WAFR2000)*, B.R. Donald, K.M. Lynch and D. Rus (Eds.), A.K. Peters, Boston, 2001, pp. 293-308.

[19] S.R. Lindemann and S.M. LaValle, “Incrementally Reducing Dispersion by Increasing Voronoi Bias in RRTs”, *Proc. IEEE Int. Conf. Robotics and Automation*, 2004, pp. 3251-3257.

[20] S.K. Lüdemann, R.R. Gabdoulline, V. Lounnas and R.C. Wade, “Substrate Access to Cytochrome P450cam Investigated by Molecular Dynamics Simulations: An Interactive Look at the Underlying Mechanisms”, *Internet Journal of Chemistry*, vol. 4(6), 2001.

[21] D. Parsons and J. Canny, “Geometric Problems in Molecular Biology and Robotics”, *Proc. 2nd Int. Conf. on Intelligent Systems for Molecular Biology*, 1994, pp. 322-330.

[22] E. Plaku and L.E. Kavraki, “Quantitative Analysis of Nearest-Neighbors Search in High-Dimensional Sampling-Based Motion Planning”, *Proc. Workshop on the Algorithmic Foundations of Robotics*, 2006, in press.

[23] V. Ruiz de Angulo, J. Cortés and T. Siméon, “BioCD: An Efficient Algorithm for Self-Collision and Distance Computation between Highly-Articulated Molecular Models”, *in Robotics: Science and Systems I*, S. Thrun, G. Sukhatme, S. Schaal and O. Brock (Eds.), MIT Press, Cambridge, 2005, pp. 6-11.

[24] T. Schlick, *Molecular Modeling and Simulation - An Interdisciplinary Guide*, Springer, New York; 2002.

[25] C. Urmson and R. Simmons, “Approaches for Heuristically Biasing RRT Growth”, *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2003, pp. 1178-1183.

[26] A. Yershova, L. Jaillet, T. Siméon and S.M. LaValle, “Dynamic-Domain RRTs: Efficient Exploration by Controlling the Sampling Domain”, *Proc. IEEE Int. Conf. Robotics and Automation*, 2005, pp. 3867-3872.

[27] M. Zhang and L.E. Kavraki, “A New Method for Fast and Accurate Derivation of Molecular Conformations” *Journal of Chemical Information and Computer Sciences*, vol. 42(1), 2002, pp. 64-70.