Graduated Fidelity Motion Planning

Mihail Pivtoraiko and Alonzo Kelly
Robotics Institute, Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15120

Abstract

This paper presents an approach to differentially constrained robot motion planning and efficient replanning. Satisfaction of differential constraints is guaranteed by the search space which consists of motions that satisfy the constraints by construction. Any systematic replanning algorithm, e.g. D*, can be utilized to search the state lattice to find a motion plan that satisfies the differential constraints, and to repair it efficiently in the event of a change in the environment. Further efficiency is obtained by varying the fidelity of representation of the planning problem. High fidelity is utilized where it matters most, while it is lowered in the areas that do not affect the quality of the plan significantly. The paper presents a method of modifying the fidelity between replans, thereby enabling dynamic flexibility of the search space, while maintaining its compatibility with replanning algorithms. The approach is especially suited for mobile robotics applications in unknown challenging environments. We successfully applied the motion planner on a real robot: the planner featured 10Hz replan rate on minimal computing hardware [2], while satisfying the car-like differential constraints.

In recent years there has been a growing interest in efficient motion replanning. Real mobile robot applications face challenges due to scarce and uncertain perception information. In order to facilitate planning a robot's motion given such challenges, dynamic replanning algorithms were developed [3]. Such algorithms incorporate updated perception information and modify the motion plan accordingly, while reusing previous computation.

This work introduces efficient replanning to motion planning under differential constraints that is based on searching a *state lattice*, a directed cyclic graph that encodes the constraints by construction [2]. Substantial computation is performed off-line to determine the connectivity of edges that represents the differential constraints. This allows fast planning (on-line) by utilizing standard search algorithms in this graph, while naturally satisfying the constraints.

In order to satisfy the differential constraints, relatively high dimensionality of the state lattice may be required. Deterministic search in this setting can be computationally costly. This cost is especially significant in outdoor robotics applications, as they pose complicated planning problems, in particular due to complex environments.

This paper addresses this limitation by managing the complexity of the search through modification of the fidelity of representation. The search space consists of one or more arbitrary regions of different fidelities. Lower fidelity of representation results in faster search, but higher fidelity results in better quality solutions. The approach is closely related to multi-resolution planning [1], but we use the term *graduated fidelity* to emphasize that the quality of representation is expressed not only as the resolution of state discretization, but also – more importantly in this setting – as the connectivity of edges between the vertices in the state lattice. Each region of the search space can be assigned a fidelity arbitrarily, yet practically this choice is guided by the region's relevance for the planning problem and the availability of the environment information. In particular, it is often beneficial to utilize a high fidelity of representation in the immediate vicinity of the moving robot. Our method meets that need by allowing the regions of different fidelity to move or change shape arbitrarily.

The contribution of this work is an improved state lattice search space that consists of regions of different fidelities of representation and allows the regions to move or change shape between replans. This search space remains compatible with standard search algorithms and is capable of producing motion plans that satisfy differential constraints without any post-processing. The state lattice structure is key to enable the reuse of computation which renders the presented algorithm efficient even on modest computing hardware.

Graduated Fidelity Planning

We extend the typical definition of the search graph by assuming that it consists of subgraphs $\mathbf{G_1}, \mathbf{G_2}, \ldots, \mathbf{G_n}$. The arrangement of vertices and edges in each subgraph is assumed to be regular, but this arrangement may be different among subgraphs to reflect the differences in the fidelity of representation. This composite search space is maintained to remain a directed cyclic graph, so that replanning algorithms can be utilized to reuse previous computation while replanning.

We define modifying a subgraph as arbitrarily changing its position (in coordinates that do not affect its connectivity, namely the translational ones) and shape (the extent of its boundary in state space). After a subgraph is modified, some of the vertices near its boundary are set to belong to a different subgraph. Note that the vertices do not move, they simply change ownership from one subgraph to another. For example, as a subgraph changes position, it "gains" the vertices on its leading edge and "loses" vertices on its trailing edge.

After all subgraphs are modified as desired, a replanning procedure needs to be executed to repair the plan. This is the same procedure that repairs the plan due to other changes in the search space (e.g. a perception update). To see why, note that deterministic replanning algorithms reuse computation by storing previously computed costs of vertices in the graph. As a result of modifying the subgraphs, some vertices change their cost due to new connectivity under a different subgraph. It is entirely transparent to the replanning algorithm whether they changed cost due to new edge costs from a perception update, or due to modification of subgraphs. Thus, the only required change to replanning algorithms to enable graduated fidelity is a process to make them aware of the vertices that have new subgraph ownership.

The vertex conversion procedure [2] is executed on each vertex v_b that changes subgraph ownership. If the vertex has not been expanded at all so far, the function returns. Otherwise, we note all vertices that may contain v_b as a predecessor – it is exactly the set of successors of v_b under the edge connectivity of its previous subgraph $\mathbf{G_i}$, denoted as $Succ_i(v_b)$. Further, we remove any back-pointers from these successor vertices v_s to v_b by examining the predecessors of v_s, $Pred(v_s)$. Effectively, we undo the effects of a previous expansion of v_b. Lastly, if this change resulted in a change of cost of any successor vertex, we insert the affected vertices and v_b itself into the priority queue. D* variants can detect this cost change automatically by recomputing the rhs-value. Note that this procedure is likely to cause replanning from the farthest affected vertex to the robot (assuming the search direction from the goal to the robot). Thus, more previous computation is reused if such changes occur closer to the robot.

This procedure is illustrated in Figure 1, using a simplified search space for ease of visualization. In this example, the search space consists of two subgraphs, $\mathbf{G_1}$ (black square vertices) and $\mathbf{G_2}$ (gray circle vertices). Arrows of similar colors are the edges. $\mathbf{G_2}$ is a small subgraph, consisting of six vertices, highlighted with a gray bounding box. The rest of the search space belongs to $\mathbf{G_1}$. In this example, $Succ_1(v_i), \forall v_i \in \mathbf{G_1}$ is 4 nearest neighbors, and $Succ_2(v_j)$, $\forall v_j \in \mathbf{G_2}$ is 8 nearest neighbors. A real implementation of graduated fidelity under differential constraints would utilize the same algorithm, but a more sophisticated connectivity of the subgraphs.

Conclusions and Future Work

In this paper we described an approach to improve the efficiency of motion planning and replanning by varying the fidelity of representation of the planning problem. In addition to leveraging dynamic replanning algorithms, this approach enables dynamic and deliberative flexibility in search space connectivity to boost efficiency. Standard replanning algorithms can be utilized, while the proposed search space design allows both the automatic satisfaction of differential constraints and the adjustment of the search space between replans. Future work includes a further investigation into the state and motion space sampling to further improve planning efficiency.

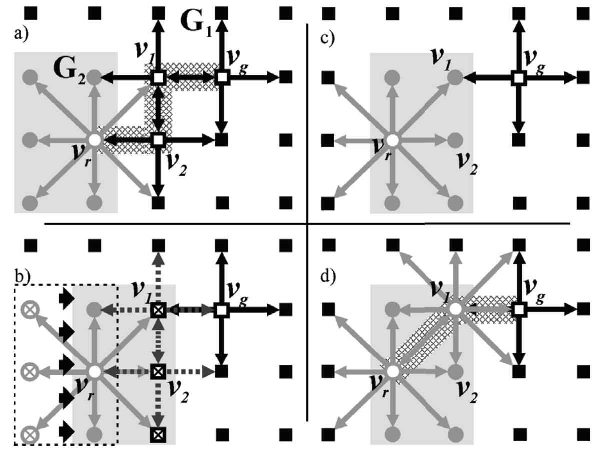

Figure 1: Maintaining the connectivity between two subgraphs of different fidelities of representation. $\mathbf{G_1}$ is a static subgraph (black square vertices), and $\mathbf{G_2}$ (gray circular vertices) moves w.r.t. the former. Arrows are edges. Hollow vertices have been expanded. Subfigure a) shows the initial plan (thick patterned line), as it originates in $\mathbf{G_1}$ and proceeds into $\mathbf{G_2}$; b) as $\mathbf{G_2}$ moves from left to right, the six crossed out vertices change subgraph ownership, and vertex conversion is executed on each of them, which results in undoing the previous expansions of v_1 and v_2; c) shows the completion of moving $\mathbf{G_2}$: the vertices v_1 and v_2 now belong to $\mathbf{G_2}$ and are available for re-expansion, if necessary, when the search algorithm performs replanning; lastly, d) shows the result of replanning in the search space from c), where due to re-expansion of v_1 under $\mathbf{G_2}$ edge connectivity, a new plan is found.

Acknowledgments

This work was supported in part by the NASA Graduate Student Researchers Program Fellowship and the JPL Mars Technology Program.

References

[1] Lavalle, S.M. 2006. *Planning Algorithms*: Cambridge University Press.

[2] Pivtoraiko, M. and Kelly, A. 2008. Differentially Constrained Motion Replanning Using State Lattices with Graduated Fidelity. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*.

[3] Stentz, A. 1995. The Focussed D* Algorithm for Real-Time Replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*: AAAI Press.