# - Milestone Report -
# Improving the Efficiency of RRT in Heterogeneous Environments with Context Sensitivity

15-780 Grad AI
Filipe Militão, Karl Naden, Bernardo Toninho

April 8, 2010

## 1 Introduction

[Where is RRT useful? Needed? Continuous etc. not optimal solution...]

Our project is centered on the observation that the fixed extension length used in the RRT[BV02] family of algorithms may not be adequate for all kinds of environments. In particular, we note that for RRT to be effective, the extension length must be calibrated to the particular environment that a robot is trying to plan in. In Figure [SIMPLER FIGURE?], for example, the narrow passages in the wall means that the extension length must be small. However, this goes against one of the strengths of the RRT algorithm which is that global reasoning about the environment is not necessary. That is, as long as a map can tell us if a given vector is obstacle free, we should not be required to discover or reason about global properties of the world.

**Goal** Our goal is to explore a set of variations in the standard RRT (and ERRT) algorithms to try to offer a better algorithm for these heterogeneous worlds both in a single iteration and re-planning.

**Related Work** Some other papers describing extensions to the RRT algorithm which may be helpful include [Lav98],[LL04],[JYLS05], and [YL09].

[TODO: expand?]

## 2 RRT Algorithms

[ Basic idea of RRT, core algorithm. ]

[ Standard Biased RRT pseudo code. ]

One potential way to address this issue is to allow the length of the expansion distance to change over the course of the (re)planning. Our idea is that associating an extension factor with each point in the random tree provides information that can be used to improve both RRT planning and re-planning. The basic planning algorithm would behave something like this:

1. Choose a random point in the world and find the nearest point in the plan tree

2. Choose an extension distance based on the extension factor of this point
3. if the expansion fails (there was an obstacle), decrease the point's extension factor and go-to (1)
4. if the expansion succeeds, increase the start point's extension factor and set the extension factor of the new node in the tree with the factor of the previous old point.

We hypothesize that this would allow for the RRT algorithm to better handle heterogeneous spaces because it could accelerate across open spaces with larger steps, but navigate through constrained spaces as well. Furthermore, the extension factor information can be used to inform a re-planning procedure. Since the extension factor is a function of how obstructed the world around a point is, we can use this information to approximate the set of the unobstructed regions of the world and further improve our successive plans (i.e. if we select a point in the tree that falls into a large unobstructed region, we probably can afford larger extensions from that point).

### 2.1 Context Sensitivity

#### 2.1.1 DVLRRT

#### 2.1.2 VLRRT

## 3 Evaluating the algorithms

*Testing and performance metrics:* To evaluate the performance of our algorithm, we will implement a test suite (with visualization) that will allow us to measure the performance of our re-planner (in successful plans to the goal and number of steps in a plan) in several types of randomly generated worlds (worlds with uniform distributions of obstacles, non-uniform, etc.), as well as compare its performance to that of ERRT in the same scenarios.

In order to compare our modification, we had to define a set of metrics not only to measure the case of when the goal is reached but also the quality of the search when it is not found.

Next we present our current metrics:

**iterations to goal** number of iterations that the algorithm takes to reach the goal;

**distance travelled** the span/length of the tree

**point density** something on the concentration of the points?

## 4 Preliminary Results

Borked.

## 5 Progress So Far

So far, we have fulfilled the following objectives:

1. *The basic algorithm:* Designing and coding the basic algorithm for a single iteration will be an important piece of the project. Determining the best way to increment and decrement the extension factor will be an interesting challenge. Furthermore, it may well be that the simple algorithm outlined above will not work perfectly. However, we may be able to enhance it with ideas such as associating directions with our extension factor (i.e only increase it when heading in the same or similar direction), by including the expansion multiplier in

the determination of the closest point, or another scheme.

## 6 Future Steps

We plan to continue our project with the following:

- Integrate replanning ERRT.
- improve the algorithm (voronoi?) play with parameters.
- *Applying the gathered information to replanning:* Apart from implementing the waypoints already used in ERRT, this portion of the project entails developing a way of extrapolating from the the several regions that make up the world and finding appropriate ways of using these regions as input to the extension factor changes (as summarized above). Another important part of this component is finding out how to adjust these regions, as the number of planning iterations increases and we obtain more information.
- evaluate in extreme world...?

## References

[BV02] James Bruce and Manuela Veloso. Real-time randomized path planning for robot navigation, 2002.

[JYLS05] Lonard Jaillet, Anna Yershova, Steven M. LaValle, and Thierry Simon. Adaptive tuning of the sampling domain for dynamic-domain rrts, 2005.

[Lav98] Steven M. Lavalle. Rapidly-exploring random trees: A new tool for path planning. Technical report, 1998.

[LL04] Stephen R. Lindemann and Steven M. LaValle. Incrementally reducing dispersion by increasing voronoi bias in rrts, 2004.

[YL09] Anna Yershova and Steven M. Lavalle. Motion planning for highly constrained spaces. In *Robot Motion and Control 2009*. 2009.