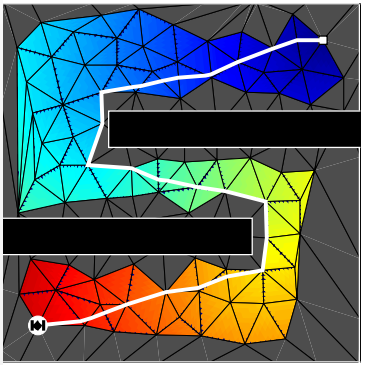
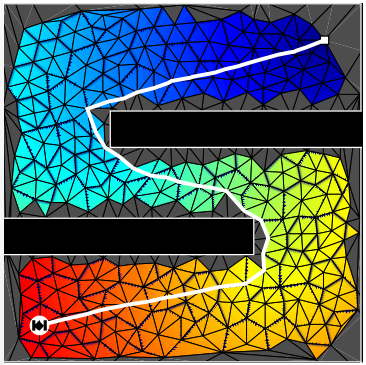
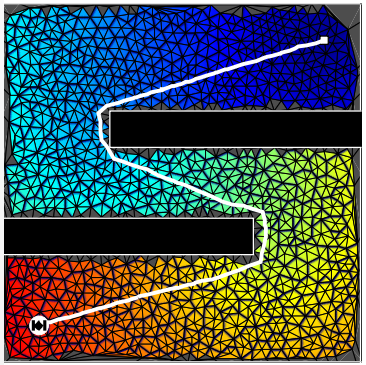
Sampling-based Volumetric Methods for Optimal Feedback Planning

Dmitry Yershov, Michael Otte, Emilio Frazzoli

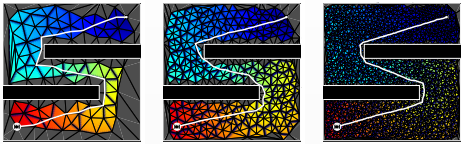
Massachusetts Institute of Technology







General Idea



Algorithm 1 ACIDIC

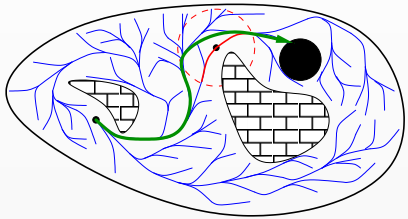
Input: Sets $X_{\text{goal}} \subseteq X_{\text{free}} \subseteq X$

1: **loop**
2: $x_{\text{new}} \leftarrow \text{SamplePoint}(X)$
3: UpdateDelaunay(x_{new})
4: **for all** τ_{new} **do**
5: CollisionCheck(τ_{new})
6: ReplanFMM(x_{new})
7: **yield** feedback $\pi(x)$

- Random and pseudo-random sampling
- Volumetric free-space approximation
- Feedback policy computations
- Asymptotic convergence

Motivation

Path-centric algorithms



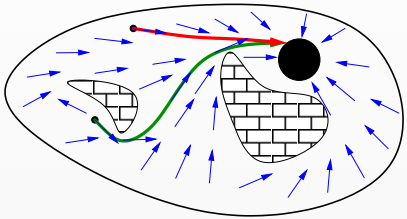
Pros

We are using graphs for planning algorithms because many efficient graph libraries are available

Cons

The main problem with graphs is in artificial restriction of robot movements on a set of zero measure

Policy-centric algorithms



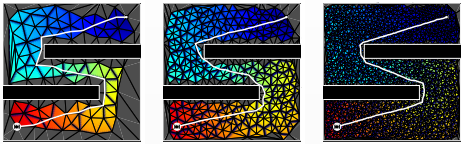
Pros

A feedback policy stabilizes robot motions towards the goal without using path-following controller middleware

Cons

Numerical methods for feedback function computations are currently unavailable, but we are determined to eliminate this problem

General Idea



Algorithm 1 ACIDIC

Input: Sets $X_{\text{goal}} \subseteq X_{\text{free}} \subseteq X$

1: **loop**
2: $x_{\text{new}} \leftarrow \text{SamplePoint}(X)$
3: UpdateDelaunay(x_{new})
4: **for all** τ_{new} **do**
5: CollisionCheck(τ_{new})
6: ReplanFMM(x_{new})
7: **yield** feedback $\pi(x)$

- Random and pseudo-random sampling
- Volumetric free-space approximation
- Feedback policy computations
- Asymptotic convergence

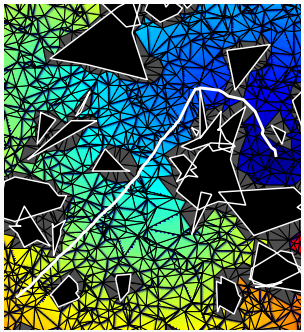
Sampling Strategies

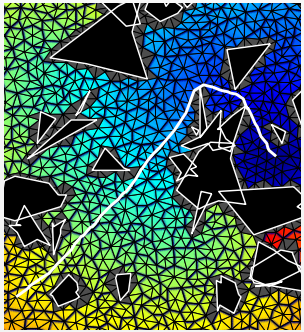
Random Sampling

- Random sampling avoids unfortunate subspace alignment
- Random sampling **does not** avoid the curse of dimensionality
- Finding narrow corridors using random sampling is as hard as using regular grids [LaValle, Branicky, Lindemann, 2004]

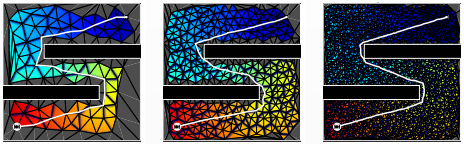
Delaunay Refinement [Ruppert, 1995; Shewchuk, 2010]

- Insert next sample at the center of the largest circumsphere
- Optimal dispersion convergence
- Randomly seeded deterministic sampling algorithm
- Sampling bias towards *regions of interest*





General Idea



Algorithm 1 ACIDIC

Input: Sets $X_{\text{goal}} \subseteq X_{\text{free}} \subseteq X$

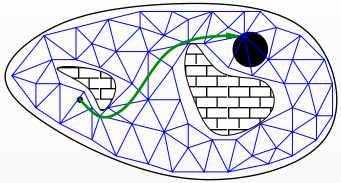
1: **loop**
2: $x_{\text{new}} \leftarrow \text{SamplePoint}(X)$
3: UpdateDelaunay(x_{new})
4: **for all** τ_{new} **do**
5: CollisionCheck(τ_{new})
6: ReplanFMM(x_{new})
7: **yield** feedback $\pi(x)$

- Random and pseudo-random sampling
- Volumetric free-space approximation
- Feedback policy computations
- Asymptotic convergence

Incremental Delaunay Triangulation

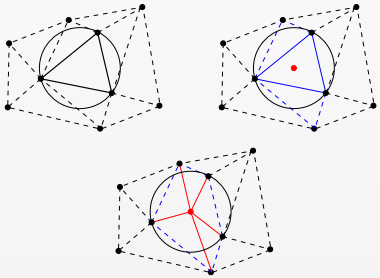
Delaunay Triangulation

- Delaunay triangulation of a sample set is a geometric simplicial complex such that the interior of all simplex circumspheres do not contain any sampled points
- The complexity of Delaunay triangulation is $O(N)$ for N samples in the *general position*, regardless of the dimension number d [Miles, 1974]

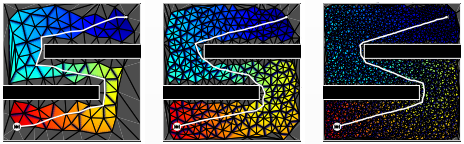


Incremental Delaunay Triangulation

- Start with the initial Delaunay triangulation
- Newly inserted sample always violates empty-circumsphere property
- "Rewire" local simplices to enforce this property
- Computational complexity $O(\log(N))$ (N is the size of the initial sample set)



General Idea



Algorithm 1 ACIDIC

Input: Sets $X_{\text{goal}} \subseteq X_{\text{free}} \subseteq X$

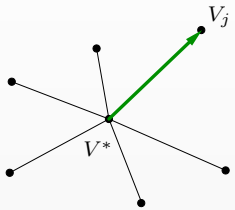
1: **loop**
2: $x_{\text{new}} \leftarrow \text{SamplePoint}(X)$
3: UpdateDelaunay(x_{new})
4: **for all** τ_{new} **do**
5: CollisionCheck(τ_{new})
6: ReplanFMM(x_{new})
7: **yield** feedback $\pi(x)$

- Random and pseudo-random sampling
- Volumetric free-space approximation
- Feedback policy computations
- Asymptotic convergence

Repairing Fast Marching Method

Dijkstra's Algorithm

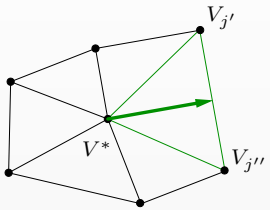
[Dijkstra, 1959]



1: Initialize Q
2: **while** Q is not empty **do**
3: Pop j with least V_j from Q
4: **for all** $i \in \mathcal{N}(j)$ **do**
5: $V^* \leftarrow \|x_i - x_j\| + V_j$
6: **if** $V^* < V_i$ **then**
7: Update V_i

Fast Marching Method

[Sethian, 1999]



1: Initialize Q
2: **while** Q is not empty **do**
3: Pop j with least V_j from Q
4: **for all** $i \in T \in \mathcal{N}(j)$ **do**
5: $V^* \leftarrow$ **minloc**(i, T, V)
6: **if** $V^* < V_i$ **then**
7: Update V_i

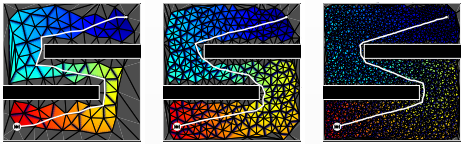
Is it difficult?

Change one line in your favorite graph-search algorithm: Dijkstra, A*, D*, LPA*, ARA*, etc*

What is necessary?

Use your favorite interpolation method to propagate cost-to-go function values

General Idea



Algorithm 1 ACIDIC

Input: Sets $X_{\text{goal}} \subseteq X_{\text{free}} \subseteq X$

1: **loop**
2: $x_{\text{new}} \leftarrow \text{SamplePoint}(X)$
3: UpdateDelaunay(x_{new})
4: **for all** τ_{new} **do**
5: CollisionCheck(τ_{new})
6: ReplanFMM(x_{new})
7: **yield** feedback $\pi(x)$

- Random and pseudo-random sampling
- Volumetric free-space approximation
- Feedback policy computations
- Asymptotic convergence

Convergence Rate and Computational Complexity

Convergence Rate

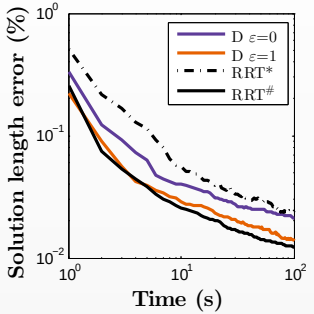
Deterministic sampling	$\mathcal{E} \sim O\big((\frac{1}{N})^{\frac{1}{d}}\big)$
Random sampling	$\mathcal{E} \sim O\big((\frac{\log N}{N})^{\frac{1}{d}}\big)$
Delaunay refinement	$\mathcal{E} \sim O\big((\frac{1}{N})^{\frac{1}{d}}\big)$

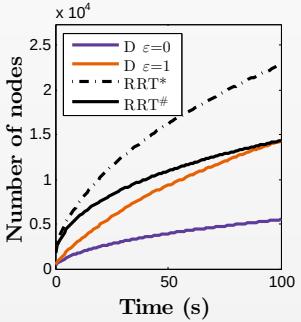
Computational Complexity per Sample

Sampling	$O(1)$
Delaunay refinement	$O(\log(N))$
Sample localization	$O(\log(N))$
Simplex rewiring	$O(1)$
Collision check	$O(1)$
Repairing FMM (amortized)	$O(\log(N))$

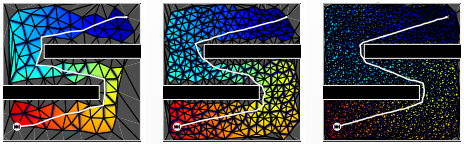
Running Time Complexity

per N samples	$O(N \log(N))$
per digit of accuracy	$O(\mathcal{E}^{-d(1+\delta)}) \quad \forall \delta > 0$





General Idea



Algorithm 1 ACIDIC

Input: Sets $X_{\text{goal}} \subseteq X_{\text{free}} \subseteq X$

1: **loop**
2: $x_{\text{new}} \leftarrow \text{SamplePoint}(X)$
3: UpdateDelaunay(x_{new})
4: **for all** τ_{new} **do**
5: CollisionCheck(τ_{new})
6: ReplanFMM(x_{new})
7: **yield** feedback $\pi(x)$

- Random and pseudo-random sampling
- Volumetric free-space approximation
- Feedback policy computations
- Asymptotic convergence

Simulation Results

http://tinyurl.com/qjnazvr