Geometric and Topological Techniques in Graph Search-based Robot Planning

Subhrajit Bhattacharya

Under the guidance of

Prof. Vijay Kumar

Prof. Maxim Likhachev

University of Pennsylvania

GRASP Laboratory

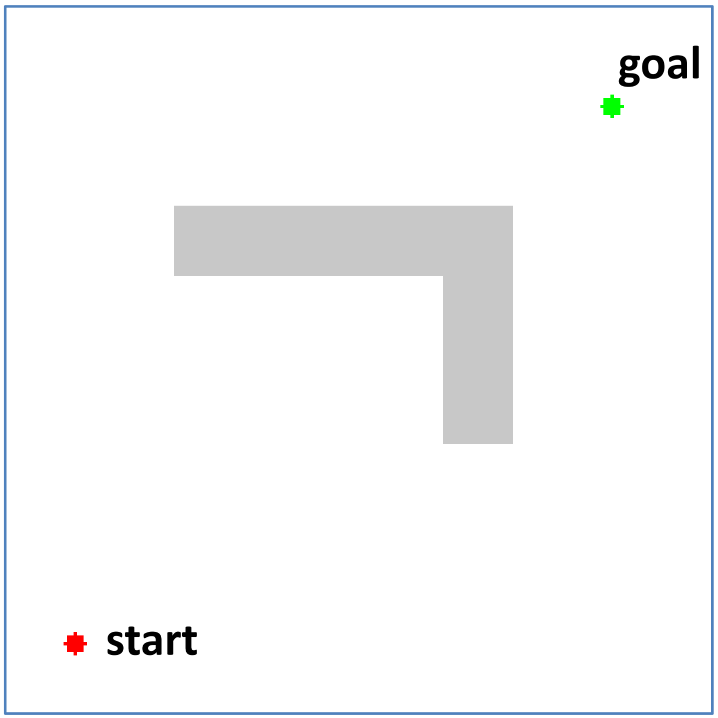
Challenges and Motivation

Graph-search based approaches **Graph construction by discretization of configuration space, and Dijkstra's, A* search**	**Continuous approaches** **Gradient descent, vector fields, solving Geodesic equation, etc.**
Pros: • Fast, efficient (in low dimensions) • Complete • Indifferent to ***non-convexity, holes*** in the environment • Globally ***optimal*** (in the graph) • Works well for ***non-Euclidean metric***	**Pros:** • More suitable for capturing continuous topological and metric features of configuration space • Gradient descent methods scale well with dimensionality.
Cons: • Topological information about the environment largely lost • Optimal path in the graph may not be ***optimal in the original metric space*** • Graph size and search complexity ***increases exponentially*** with dimension of configuration space (e.g. ***multi-robot config. spaces, tasks***).	**Cons:** • Inability to deal with **non-convexity &holes** in config. space • Local minima • Slow convergence, computationally difficult (e.g. solving Geodesic eqn.)

Can we design methods using both techniques to complement each other?

Quick overview of A* search

Algorithm to find least cost path in a graph (e.g. graph created by discretization of an environment)



- **Important feature:** No need to store entire graph in memory from beginning
  – nodes and edges are "created on the fly", incrementally.

Overview of My Work

- Planning with Topological constraints – Homotopy & Homology class constraints
  (AAAI 2010, RSS 2011)
- Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric
  (DARS 2010)
- Transformation for Efficient Optimal Planning in Environments with Non-uniform Metric
- Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow
  (RSS 2010, ICRA 2010)

Search-based Path Planning with Homotopy Class Constraints

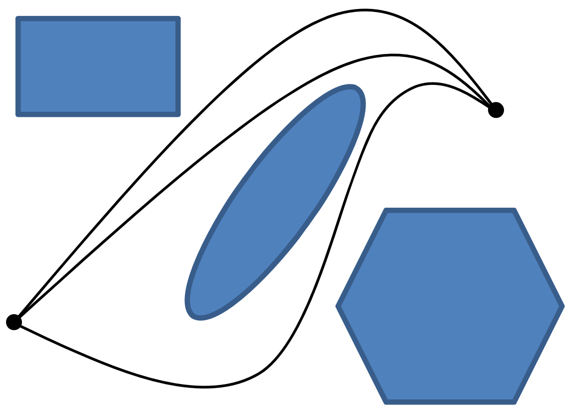
The 2-dimensional case

Bhattacharya, Kumar, Likhachev

The Twenty-Fourth AAAI Conference on Artificial Intelligence **(AAAI 2010)**

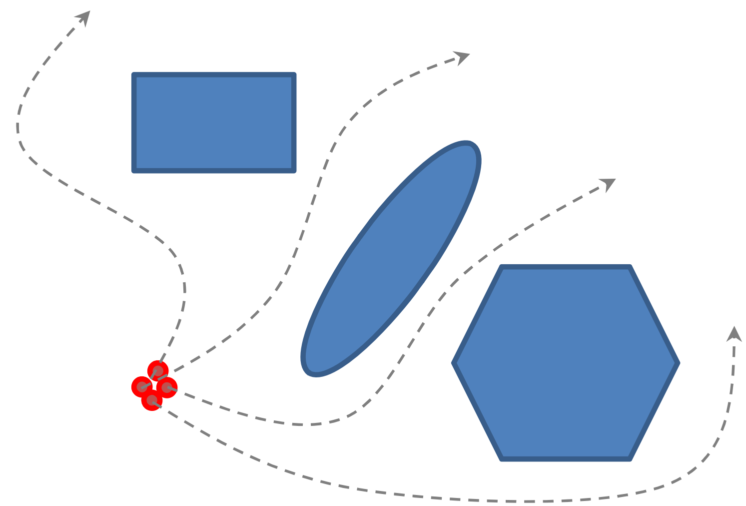
Homotopy Classes

Definition



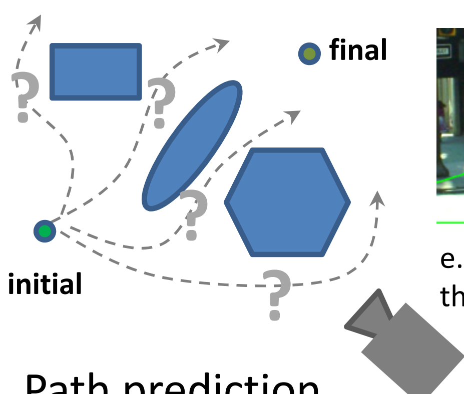
Trajectories in different
homotopy classes

Motivational Example



Deploying multiple agents for:
- Searching/exploring the map
- Pursuing an agent with uncertain paths

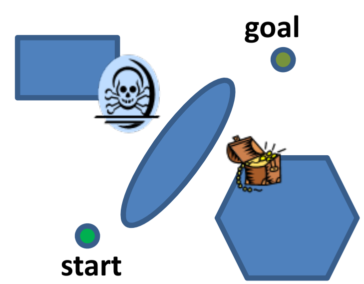
Other applications:



Path prediction

e.g. in tracking dynamic agents through multiple occlusions

J. Shi, et al.



Avoid or visit certain homotopy classes

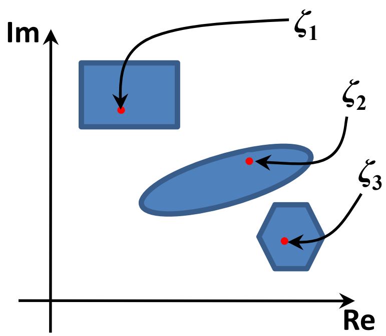
Our Goal

- Plan for optimal cost paths, cost being any **arbitrary cost function** (not necessarily Euclidean distances).
- Derive an **efficient** representation of homotopy classes
- Avoid certain homotopy classes or constrain to certain homotopy classes – ***homotopy class constraints***. *Constrained optimization!*
- Efficiently plan in **arbitrary discretization and graph representation** (Uniform discretization, unstructured discretization , triangulation, visibility graph, etc.)
- To be able to use **any standard graph search algorithm** (Dijkstra's, A*, D*, ARA*, etc.).

Our approach: Exploit theorems from **Complex analysis** – **Cauchy Integral Theorem** and **Residue Theorem**

Basic Concept

(Construction)



Represent the X-Y plane by a **complex plane**
i.e. A point (x,y) is represented as $z = x + iy$

Place "representative points", ζ_i, inside *significant* obstacles

f_0, for example, can be any arbitrary polynomial in z

$$\mathcal{F}(z) = \frac{f_0(z)}{(z-\zeta_1)(z-\zeta_2)\cdots(z-\zeta_N)}$$

Define an *Obstacle Marker function* such that it is **Complex Analytic** everywhere,

except for having **poles (singularities)** at the *representative points*

Complex Analytic Function ≡

Complex Differentiable Functions:

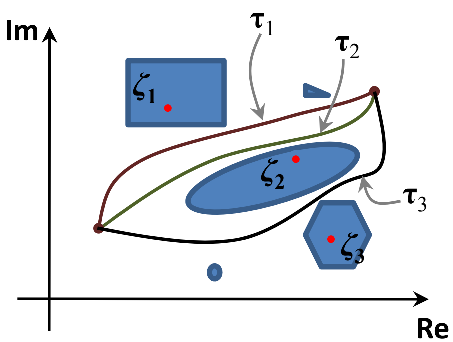
$$\mathcal{F}(z) \equiv \mathcal{F}(x+iy) \equiv u(x,y) + i\,v(x,y)$$

Equivalently, $\boxed{\mathcal{F}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u(x,y) \\ v(x,y) \end{pmatrix}}$

with u, v following certain properties ($\nabla^2 u = \nabla^2 v = 0$) which are guaranteed when x & y are implicitly used within z in construction of $\mathcal{F}$.

Basic Concept
(Properties of Complex Analytic functions)

A direct consequence of **Cauchy Integral Theorem** and **Residue Theorem**



$$\mathcal{F}(z) = \frac{f_0(z)}{(z-\zeta_1)(z-\zeta_2)\cdots(z-\zeta_N)}$$

$$\int_{\boldsymbol{\tau}_1} \mathcal{F}(z)dz = \int_{\boldsymbol{\tau}_2} \mathcal{F}(z)dz \neq \int_{\boldsymbol{\tau}_3} \mathcal{F}(z)dz$$

An alternative (preferred)**:**

$$\mathcal{F}(z) = \begin{bmatrix} \dfrac{f_1(z)}{z-\zeta_1} \\ \dfrac{f_2(z)}{z-\zeta_2} \\ \vdots \\ \dfrac{f_N(z)}{z-\zeta_N} \end{bmatrix}$$

But the singularities lie on the obstacles!!

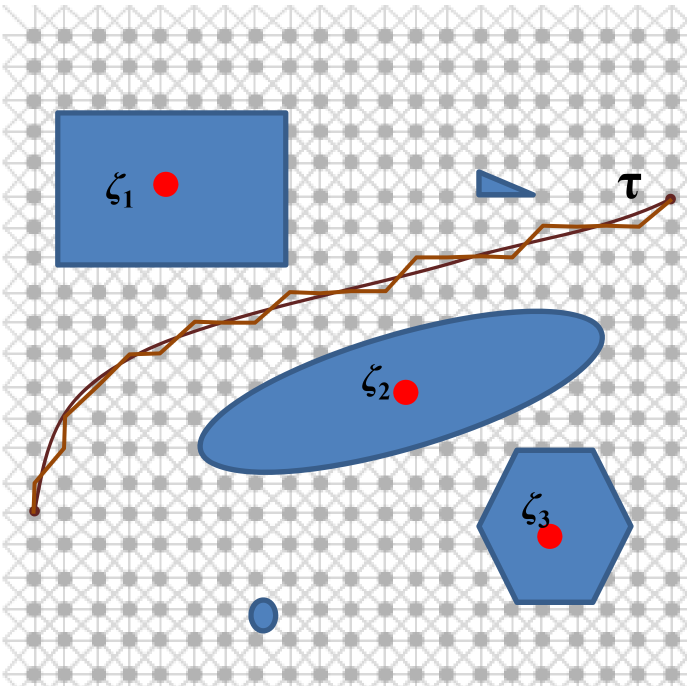
The value of $L(\tau) = \int_\tau \mathcal{F}(z)dz$ uniquely defines the homotopy class of a trajectory τ

Note: L is an additive function of the trajectory:

$L(\tau_1 \cup \tau_2) = L(\tau_1) + L(\tau_2)$

– we will exploit this for graph-search algorithm

Switching to a Discretized Perspective

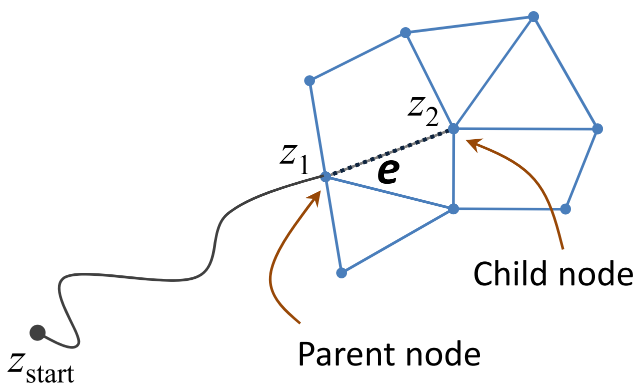


A **trajectory** in a discretized setting, is nothing but a **path in the graph**

$$\int_\tau \mathcal{F}(z)dz = \sum_{\substack{\text{edge } \boldsymbol{e} \\ \text{in path } \boldsymbol{\tau}}} \int_{\boldsymbol{e}} \mathcal{F}(z)dz$$

An **integration along a path** in the graph is nothing but

summation of the values of $L(\boldsymbol{e})$ of the edges $\boldsymbol{e}$ along that path



$$L(z_{\text{start}} \rightarrow z_2) = L(z_{\text{start}} \rightarrow z_2) + L(\boldsymbol{e})$$

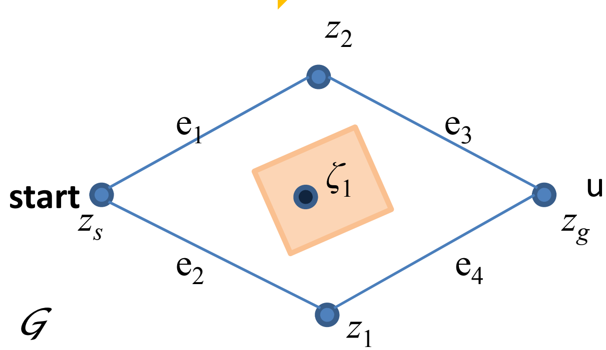
Turns out, $L(\boldsymbol{e})$ can be computed efficiently using a closed-form analytical expression. **(more details in paper)**

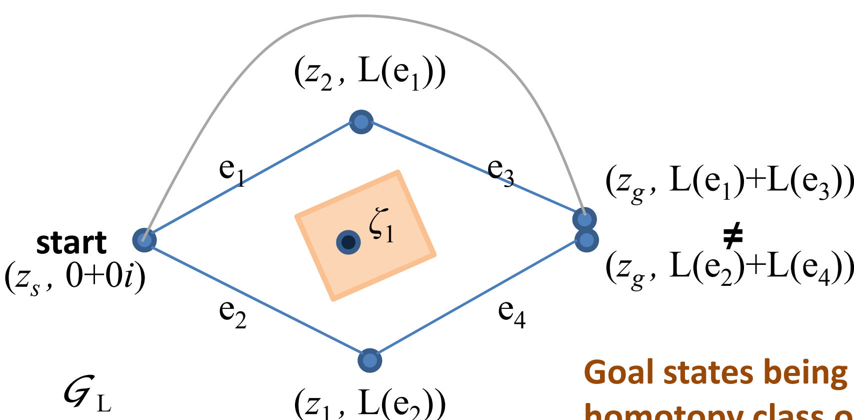
Graph Construction
(The *L*-augmented graph)

Given the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ laid upon the environment, we construct,

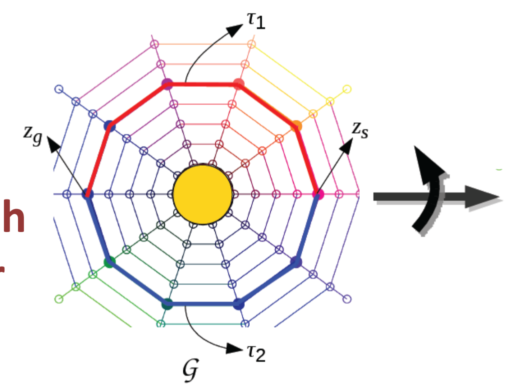
$$\mathcal{G}_L(\mathcal{G}) = \{\mathcal{V}_L, \mathcal{E}_L\}$$

z in $\mathcal{G}$ $\Rightarrow$ $\{z, L(z_s \rightarrow z)\}$ in $\mathcal{G}_L$

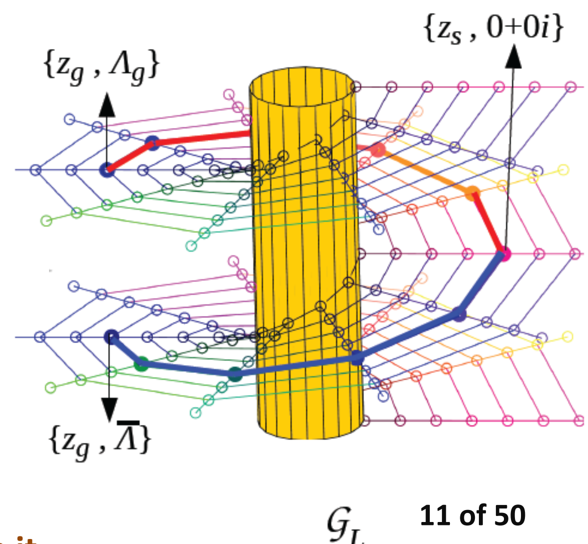




Insight into graph topology:



More details on Graph construction in paper



Goal states being distinguished by homotopy class of path taken to reach it

L-augmented Graph
- Formal definition:

Homotopy Class Constraints:

Set $\mathcal{A} = \{\alpha_1, \alpha_2, \cdots, \alpha_a\}$ denotes the set of *L*-values of allowed homotopy classes

Set $\mathcal{B} = \{\beta_1, \beta_2, \cdots, \beta_b\}$ denotes the set of *L*-values of blocked homotopy classes

$$\mathcal{G}_L(\mathcal{G}) = \{\mathcal{V}_L, \mathcal{E}_L\}$$

where,

1.

$$\mathcal{V}_L = \left\{ \{z, \Lambda\} \middle| \begin{array}{l} z \in \mathcal{V}, \text{ and,} \\ \Lambda \notin \mathcal{B} \text{ (or equivalently, } \Lambda' \in \mathcal{A}) \\ \qquad \text{if } z = z_g, \text{ and,} \\ \Lambda = L(z_s \rightarrow z) \text{ for some trajectory} \\ \qquad z_s \rightarrow z, \text{ from } z_s \text{ to } z \end{array} \right\}$$

2. And, edge $\{\{z, \Lambda\} \rightarrow \{z', \Lambda'\}\}$ is in $\mathcal{E}_L$ for $\{z, \Lambda\} \in \mathcal{V}_L$ and $\{z', \Lambda'\} \in \mathcal{V}_L$, iff
   i. $\{z \rightarrow z'\} \in \mathcal{E}$, and
   ii. $\Lambda' = \Lambda + L(z \rightarrow z')$, where $L(z \rightarrow z')$ is the L-value of the straight line segment joining the adjacent nodes z and z'
3. And, the cost/weight associated with an edge $\{\{z, \Lambda\} \rightarrow \{z', \Lambda'\}\} \in \mathcal{E}_L$ is same as the cost of the edge $\{z \rightarrow z'\} \in \mathcal{E}$.

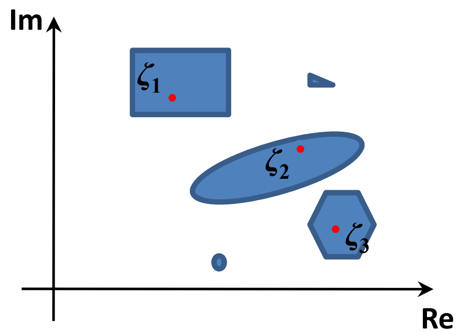
Theoretical guarantee

Theorem 1. *If $\mathcal{P}_L^* = \{\{z_1, L_1\}, \{z_2, L_2\}, \cdots, \{z_P, L_P\}\}$ is an optimal path in $\mathcal{G}_L$, then the path $\mathcal{P}^* = \{z_1, z_2, \cdots, z_P\}$ is an optimal path in the graph $\mathcal{G}$ satisfying the Homotopy class constraints specified by $\mathcal{A}$ and $\mathcal{B}$*

Implementation details

- Small obstacles

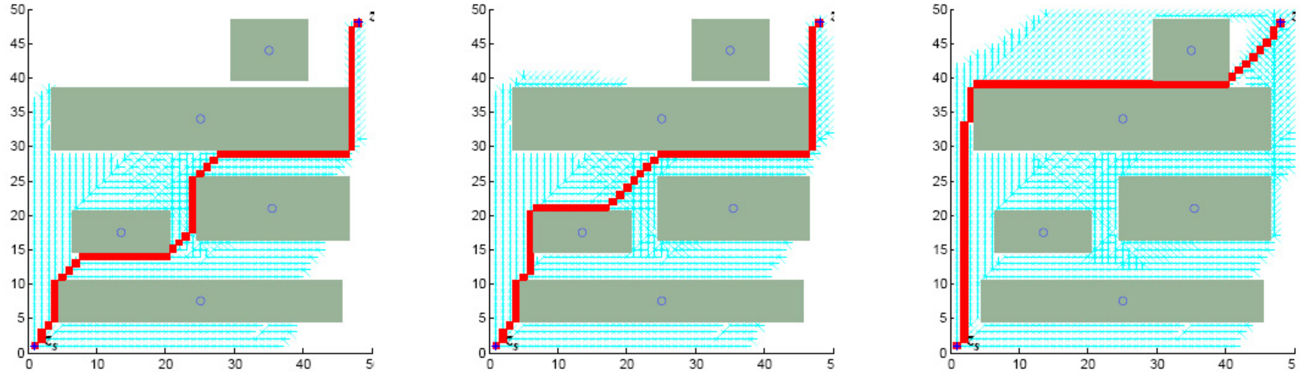
  We can ignore small obstacles or potential noise (incorrect reading from sensor data) by choosing not to put a ζ on an obstacle.



- Single search for finding least cost paths in different homotopy classes

  We can perform a single graph search to achieve this by continued expansion of states.

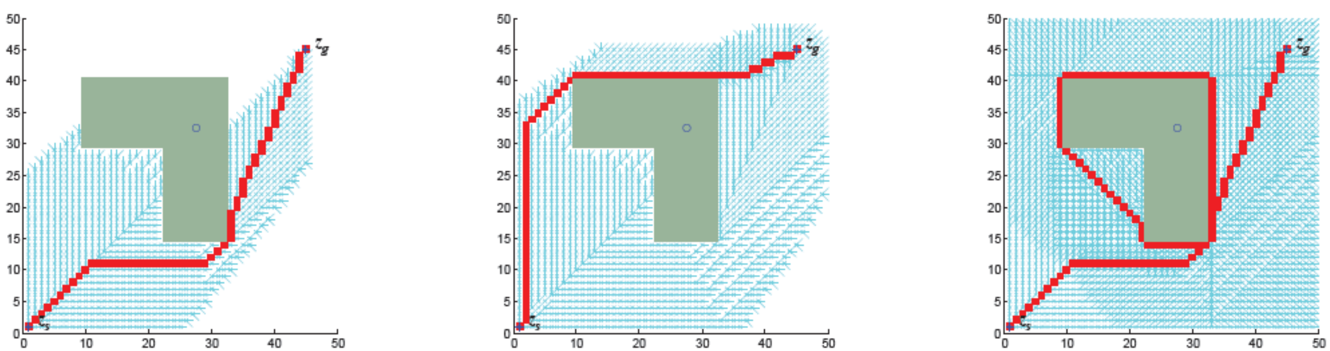
Experimental Results for 8-connected Grid
(Homotopy class exploration)



(a) $\mathcal{B}=\{\}$

(b) $\mathcal{B}=\{-409.76+2557.70i\}$.

(c) $\mathcal{B}=\{-409.76+2557.70i, 567.90+2220.77i\}$.



(a) No homotopy class constraint

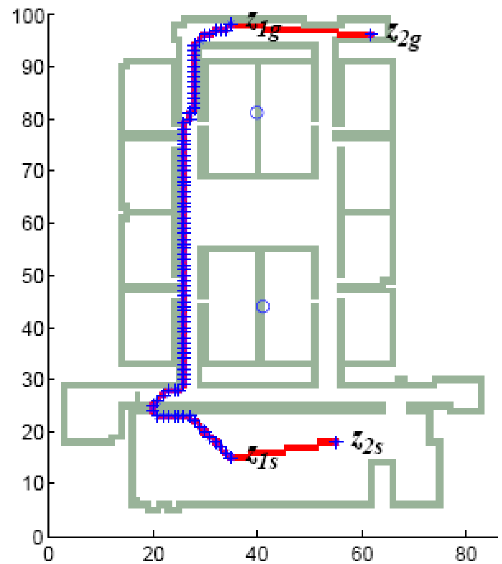
(b) $\mathcal{B}=\{52.15+85.97i\}$.

(c) Non-Jordan curve.

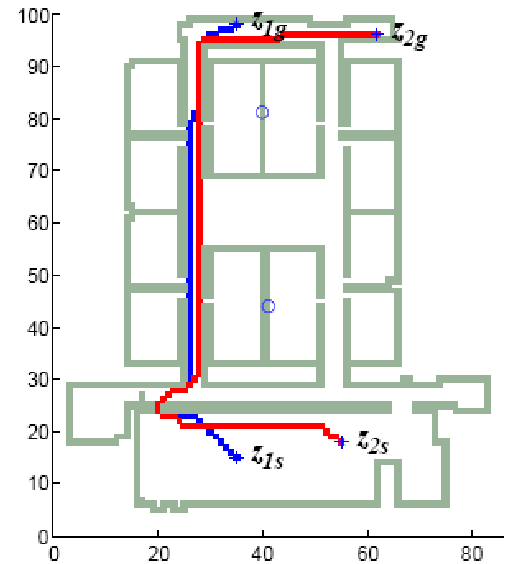
Results

(“Visibility” constraint translates to homotopy class constraint)

(a) Unconstrained plans of two robots



(b) Robot 2 determines L-value of desired hmtp. class



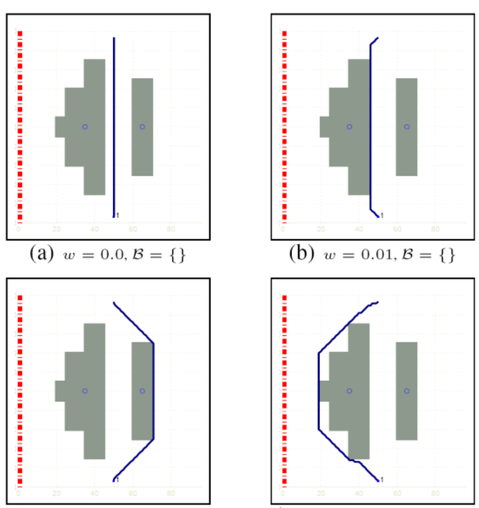
(c) Optimal plan with *visibility constraint* satisfied

More applications

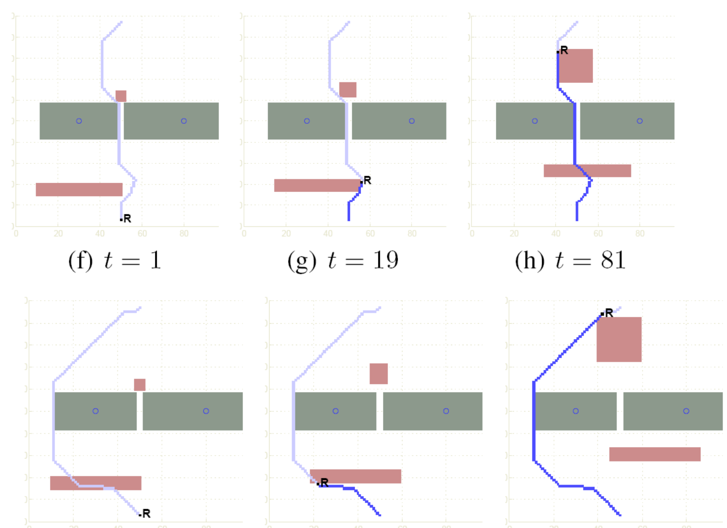
Subhrajit Bhattacharya, Vijay Kumar and Maxim Likhachev (2010)
“Search-based Path Planning with Homotopy Class Constraints”.
In Proceedings of The Twenty-Fourth AAAI Conference on Artificial Intelligence.

Revisit if there is time:

jump to slide

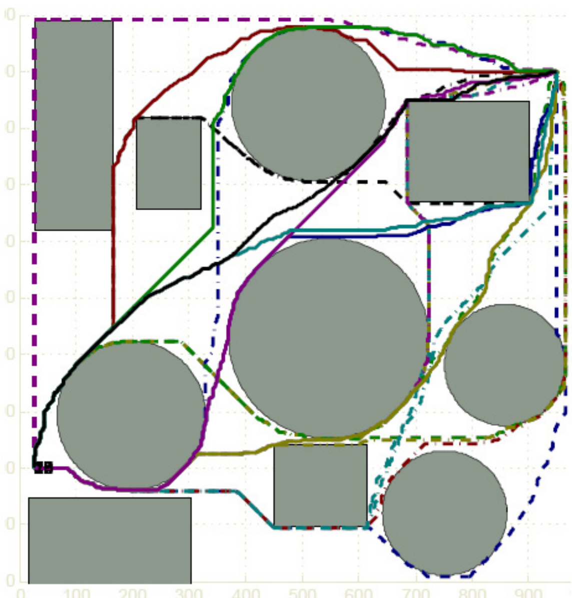


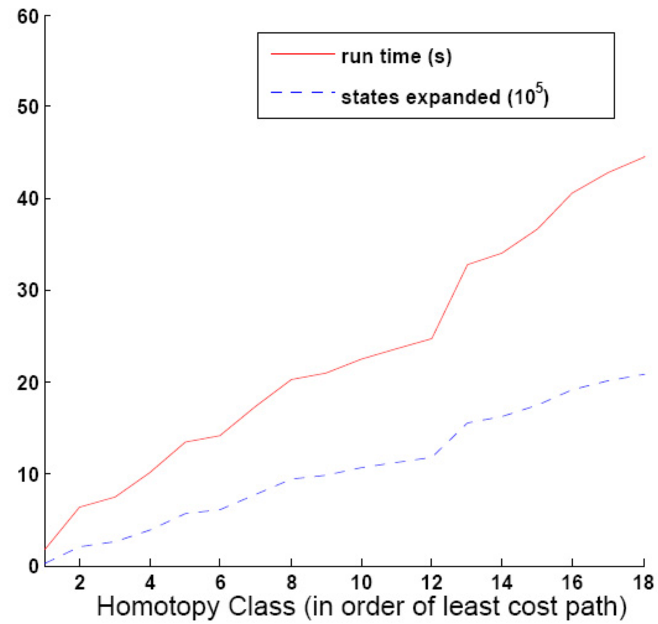
(a) $w = 0.0, \mathcal{B} = \{\}$ (b) $w = 0.01, \mathcal{B} = \{\}$ (c) $w = 0.0, \mathcal{B} = \{-8.41 + 8.41i\}$ (d) $w = 0.01, \mathcal{B} = \{-8.41 + 8.41i\}$



(f) $t = 1$ (g) $t = 19$ (h) $t = 81$ (c) $t = 1$ (d) $t = 30$ (e) $t = 113$

Results
(Demonstrating efficiency and scalability)

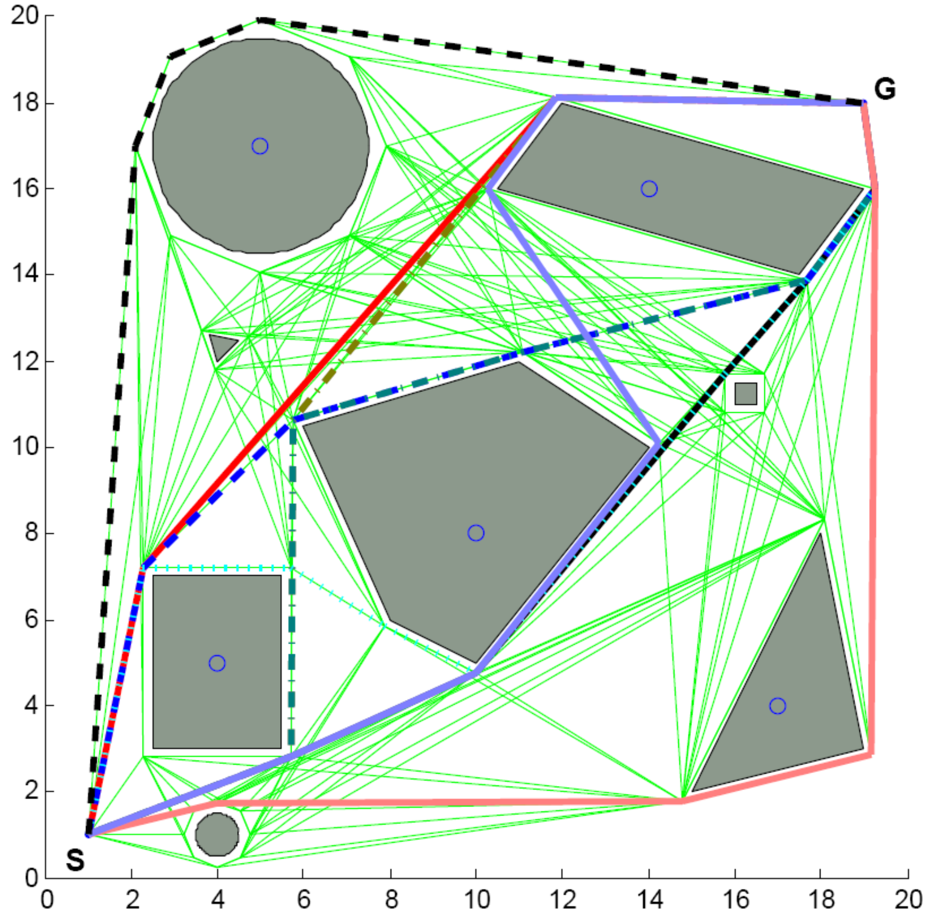




Exploring 20 homotopy classes in a
1000x1000 uniformly discretized environment

Time required for finding all the 20 homotopy classes < 50 seconds

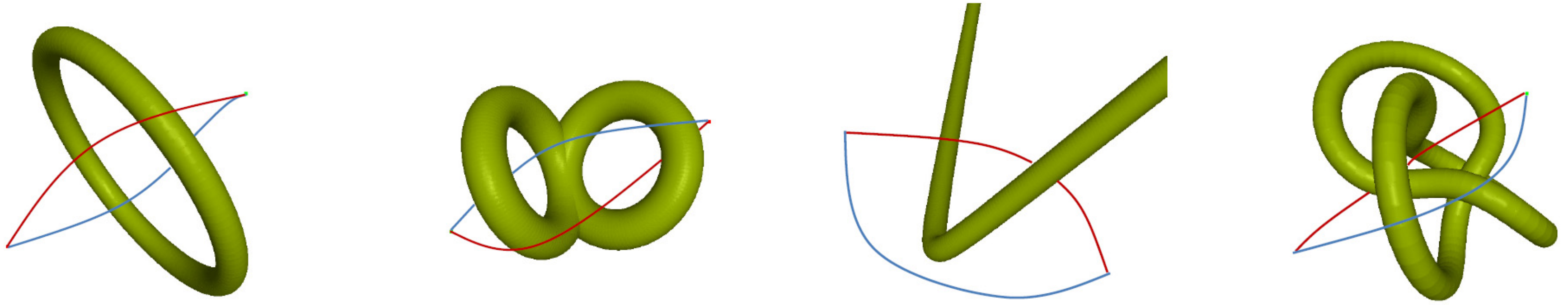
Results
(Implementation on a Visibility Graph)



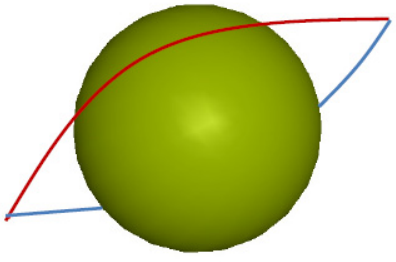
Homotopy Classes in 3 Dimensional Configuration Spaces

Bhattacharya, Likhachev, Kumar

Robotics: Science and Systems **(RSS 2011)** ***[to appear]***

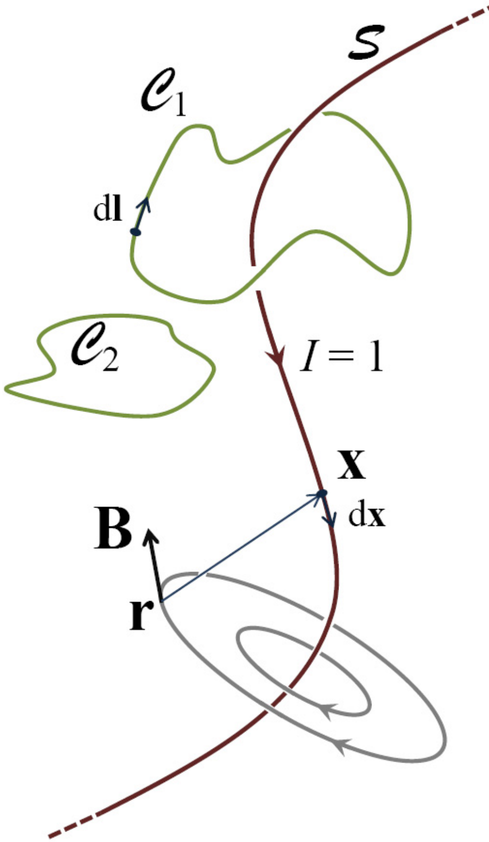


Trajectories in different homotopy classses



Trajectories in same homotopy classs

We exploit theorems from Electromagnetism



Biot-Savart's Law

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_{\mathcal{S}} \frac{(\mathbf{x} - \mathbf{r}) \times \mathrm{d}\mathbf{x}}{\|\mathbf{x} - \mathbf{r}\|^3}$$

Ampere's Law

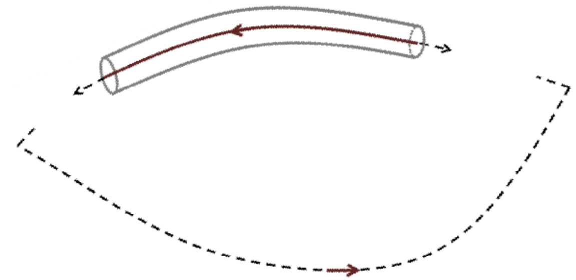
$$\Xi(\mathcal{C}) := \int_{\mathcal{C}} \mathbf{B}(\mathbf{l}) \cdot \mathrm{d}\mathbf{l} = \mu_0 I_{encl}$$

$\mathbf{B}$: Magnetic field vector

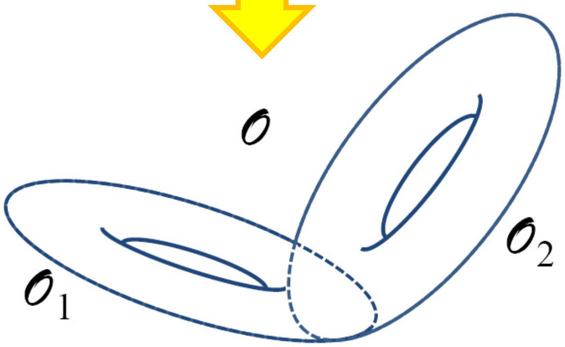
μ_0: Magnetic constant (can be chosen as 1 with proper choice of units)

Constructions on 3D obstacles

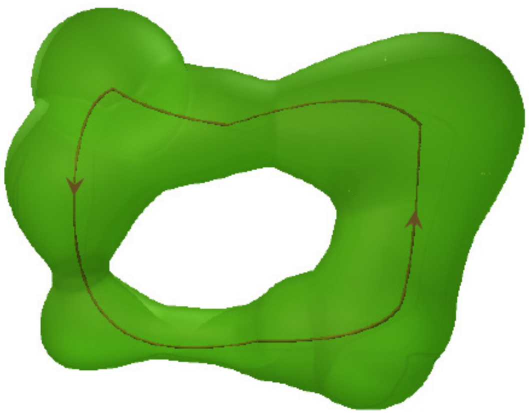
Arbitrary obstacles



Close unbounded obstacles



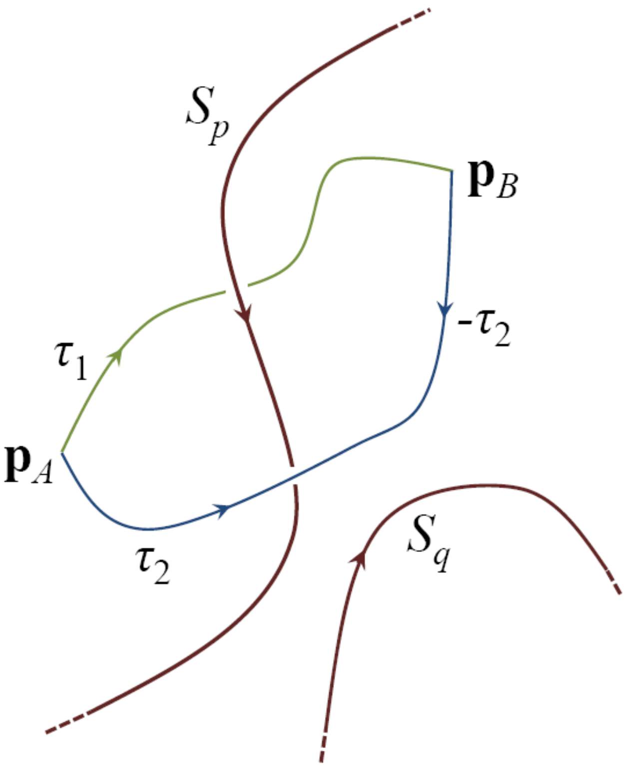
Virtually decompose genus > 1 obstacles



A simple homotopy inducing obstacle (SHIO)

Construct skeleton of a SHIO and model that as a current carrying conductor

Homotopy Signature



Virtual Magnetic Field due to *i*th SHIO

$$\mathbf{B}_i(\mathbf{r}) = \frac{1}{4\pi} \int_{S_i} \frac{(\mathbf{x} - \mathbf{r}) \times \mathrm{d}\mathbf{x}}{\|\mathbf{x} - \mathbf{r}\|^3}$$

h*-signature of trajectory *τ

$$\mathcal{H}(\tau) = [h_1(\tau),\ h_2(\tau),\ \ldots,\ h_M(\tau)]^T$$

where,

$$h_i(\tau) = \int_\tau \mathbf{B}_i(\mathbf{l}) \cdot \mathrm{d}\mathbf{l}$$

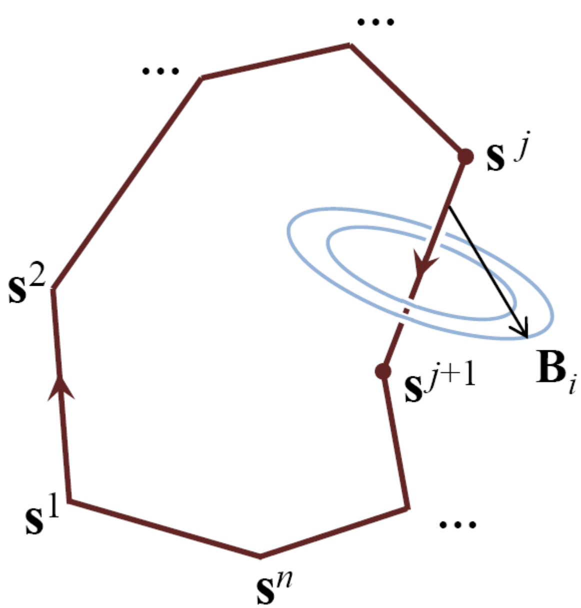
Implication/consequences – coming up in a few slides!

Proposition:

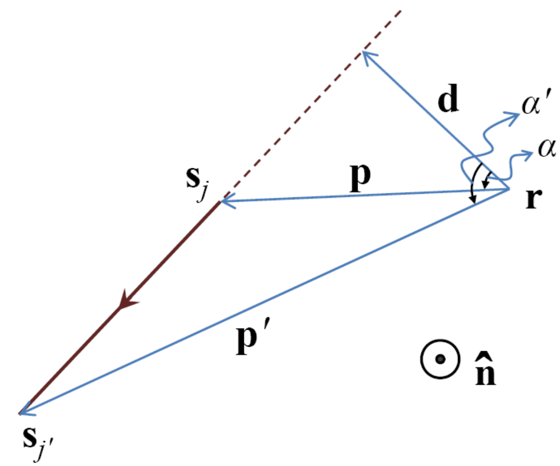
Two trajectories are in same homotopy class $\Rightarrow$ their *h-signatures* are same

$$\begin{aligned} h_i(\tau_1 \cup -\tau_2) &= \int_{\tau_1 \cup -\tau_2} \mathbf{B}_i(\mathbf{l}) \cdot \mathrm{d}\mathbf{l} \\ &= h_i(\tau_1) - h_i(\tau_2) \end{aligned}$$

Analytic computation of Virtual Magnetic Field



Construct skeletons of SHIO such that it is made up of line segments



$$\mathbf{B}_i(\mathbf{r}) = \frac{1}{4\pi}\sum_{j=1}^{n_i} \boldsymbol{\Phi}(\mathbf{s}_i^j, \mathbf{s}_i^{j'}, \mathbf{r})$$

$$\boldsymbol{\Phi}(\mathbf{s}_i^j, \mathbf{s}_i^{j'}, \mathbf{r}) = \frac{1}{\|\mathbf{d}\|^2}\left(\frac{\mathbf{d}\times\mathbf{p}'}{\|\mathbf{p}'\|} - \frac{\mathbf{d}\times\mathbf{p}}{\|\mathbf{p}\|}\right)$$

The virtual magnetic field, $\mathbf{B}_i$, can be computed efficiently using closed form formulae.

H-signature augmented graph

Definition is similar to that of L-augmented graph

$$\mathcal{G}_H = \{\mathcal{V}_H, \mathcal{E}_H\}$$

where,

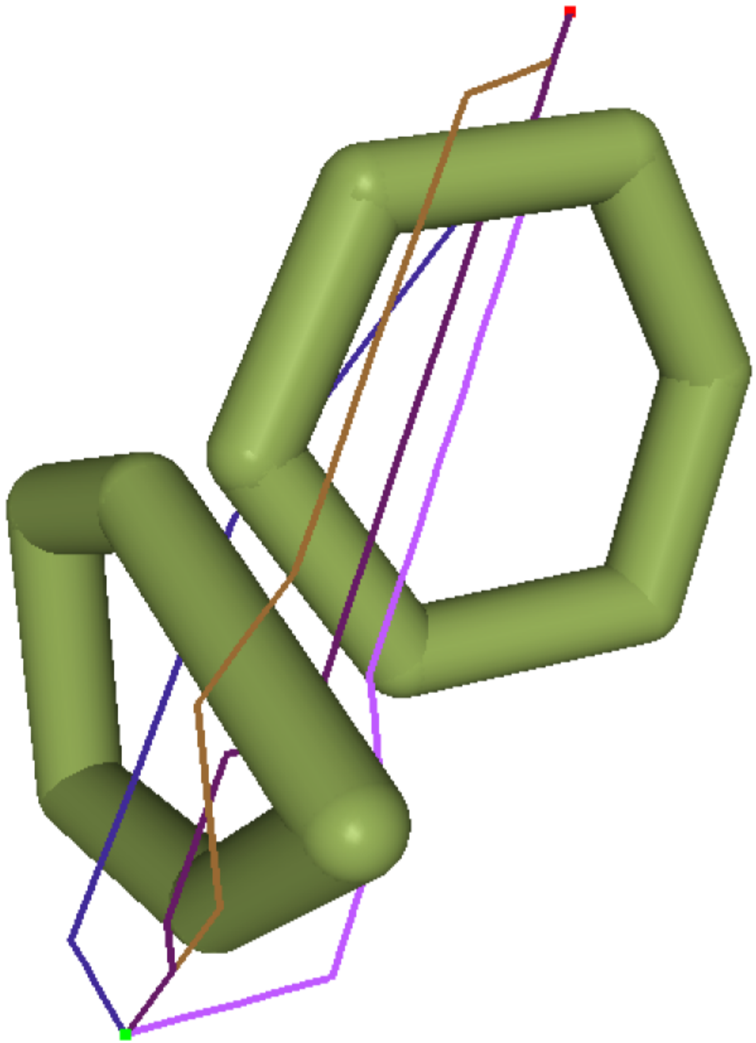
1.

$$\mathcal{V}_H = \left\{ \{\mathrm{v}, \mathrm{h}\} \left| \begin{array}{l} \mathrm{v} \in \mathcal{V}, \text{ and,} \\ \mathrm{h} = \mathcal{H}(\widetilde{\mathrm{v}_s\mathrm{v}}) \text{ for some trajectory} \\ \qquad\qquad \widetilde{\mathrm{v}_s\mathrm{v}} \in \mathcal{P}(\mathrm{v}_s, \mathrm{v}), \text{ and,} \\ \mathrm{h} \in \mathcal{A} \text{ (equivalently, } \mathrm{h} \notin \mathcal{B}) \\ \qquad\qquad\qquad \text{when } \mathrm{v} = \mathrm{v}_g \end{array} \right. \right\}$$

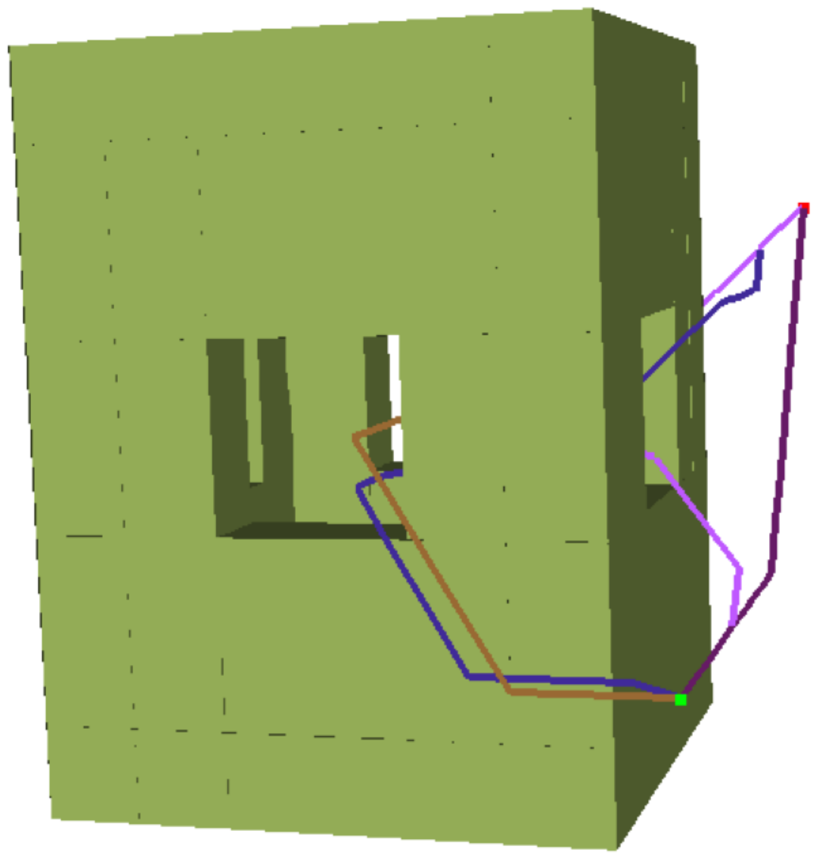
2. An edge $\{\{\mathrm{v}, \mathrm{h}\} \rightarrow \{\mathrm{v}', \mathrm{h}'\}\}$ is in $\mathcal{E}_H$ for $\{\mathrm{v}, \mathrm{h}\} \in \mathcal{V}_H$ and $\{\mathrm{v}', \mathrm{h}'\} \in \mathcal{V}_H$, iff
   i. The edge $\{\mathrm{v} \rightarrow \mathrm{v}'\} \in \mathcal{E}$, and,
   ii. $\mathrm{h}' = \mathrm{h} + \mathcal{H}(\mathrm{v} \rightarrow \mathrm{v}')$, where, $\mathcal{H}(\mathrm{v} \rightarrow \mathrm{v}')$ is the *homotopy signature* of the edge $\{\mathrm{v} \rightarrow \mathrm{v}'\} \in \mathcal{E}$.
3. The cost/weight associated with an edge $\{\{\mathrm{v}, \mathrm{h}\} \rightarrow \{\mathrm{v}', \mathrm{h}'\}\}$ is same as the cost/weight associated with edge $\{\mathrm{v} \rightarrow \mathrm{v}'\} \in \mathcal{E}$.

Same theorem for optimality holds.

Results

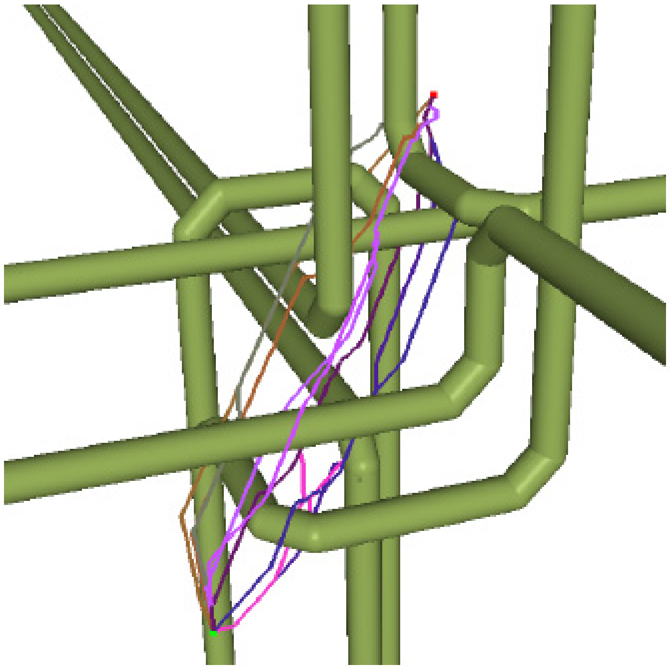


Exploration of 4 homotopy classes in presence of 2 SHIOs

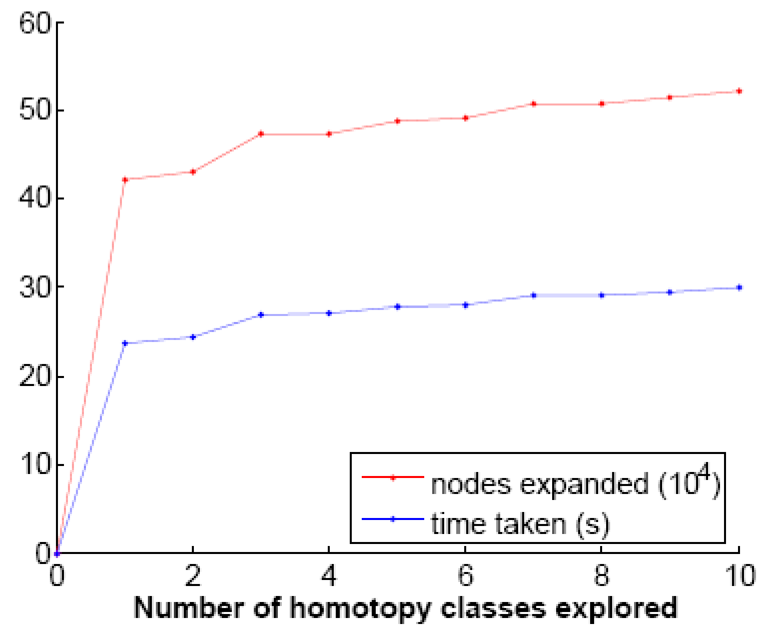


Exploration of 4 homotopy classes in presence of 4 SHIOs

Results

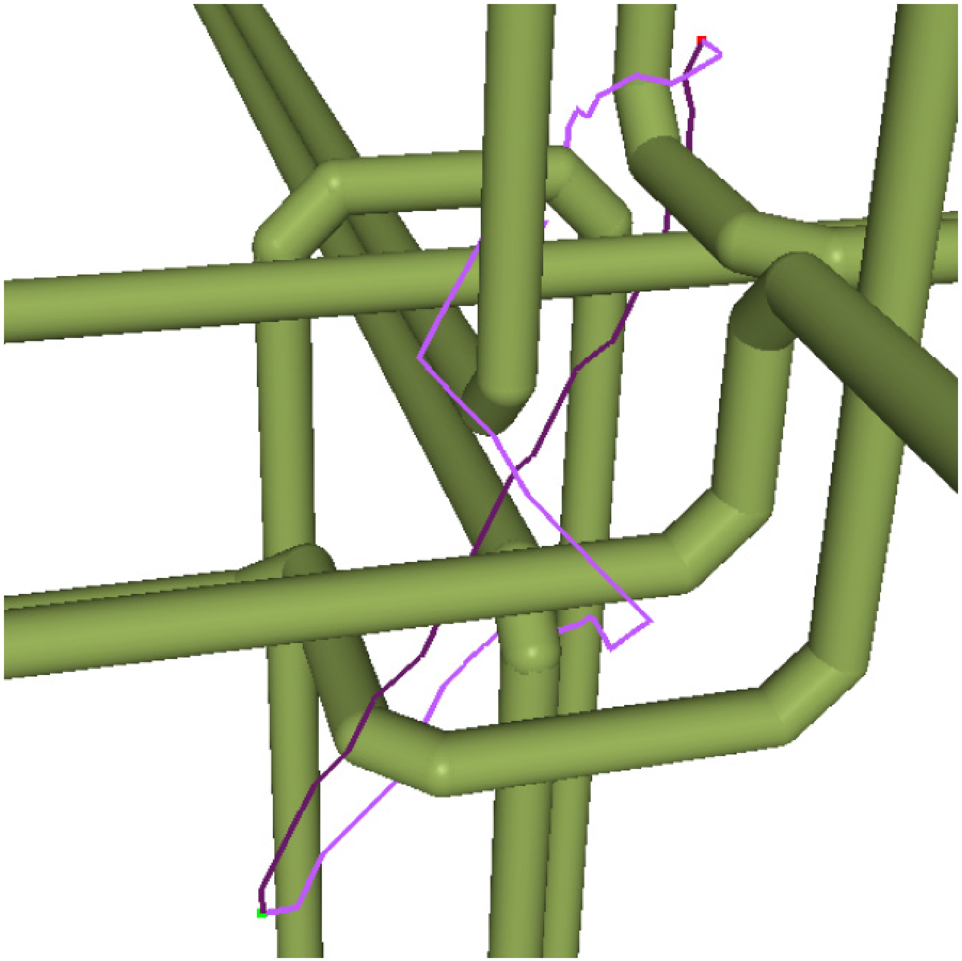


Exploration of 10 homotopy classes in presence of 7 SHIOs



44 x 44 x 44 discretized workspace – We precomputed the *h*-signatures for all edges in the graph (takes ~20 mins)

Planning for paths in Complementary homotopy classes

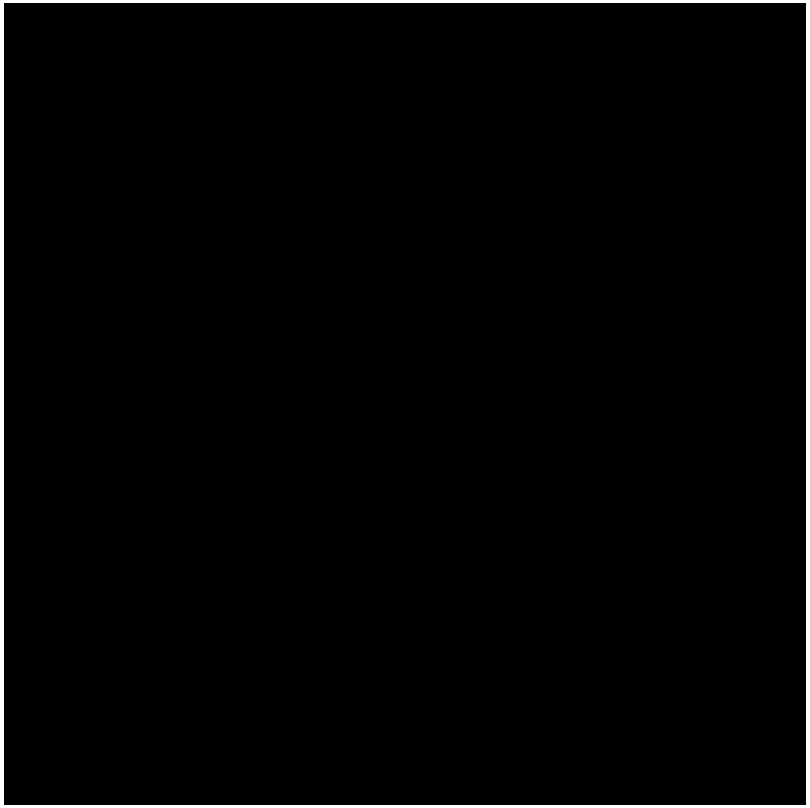


The two trajectories pass on the *opposite sides* of each and every pipe

The *h*-signature of trajectory complimentary to τ:

$$\mathcal{H}(\tau) - \mathrm{sign}(\mathcal{H}(\tau)).$$

Planning in X-Y-Time configuration space – Moving obstacles in 2 dimensions



Movie

Homotopy vs. Homology

Does the method discussed for ***3-dimensional case*** really let us impose homotopy constraints?

***Note:** This issue does not arise in the 2-D case*

No! Strictly speaking, it is ***homology*** constraints and homology classes that we are looking into.

However, since trajectories belong to the **same** ***homeomorphism class*** **(homeomorphic to [0,1])**, the identification of homotopy and homology is justified in most **practical cases.** e.g.,

- They ***exactly agree in X-Y-Time*** configuration spaces. But some pathological cases may arise in X-Y-Z spaces with linked obstacles.
- Since "***NOT homologous" implies "NOT homotopic"***, the problems of exploring different homotopy classes is not affected!

Can we do similar things in 4 and higher dimensional configuration space?

Yes, we can!

Bhattacharya, Likhachev, Kumar

"A Homotopy-like Class Invariant for Sub-manifolds of Punctured Euclidean Spaces".

Electronic pre-print. arXiv:1103.2488 *[work in progress]*

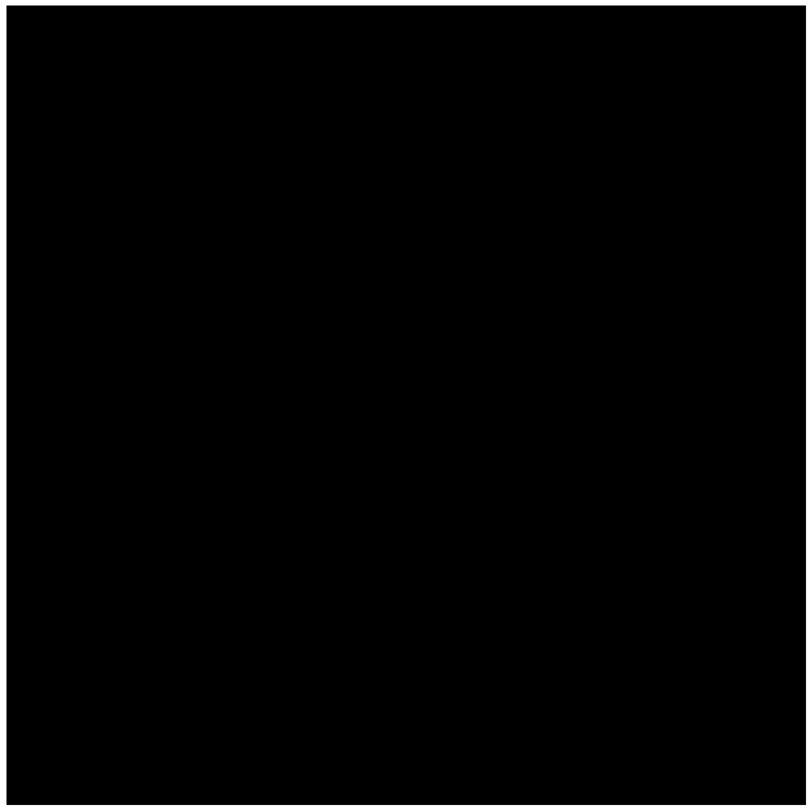
Recent collaboration with Dr. Robert Ghrist and Dr. David Lipsky

Similar treatment for homology of N-1 dimensional boundaryless manifolds embedded in D dimensional Euclidean ambient manifolds with D-N dimensional discontinuities.

- **Unification** of theorems/laws from Complex analysis, Electromagnetism and Electrostatics.
- **Generalization** to higher dimensional spaces.

Skip details!

Exploration of homotopy classes in a 4 dimensional configuration space – **X-Y-Z-Time – Moving obstacles in 3D**



Movie

Future directions

- Reformulate some of the theoretical analysis in terms of ***homology and cohomology*** instead of cobordism & surgery theory. ***(primary – ongoing)***
- Extend to ***topologically non-Euclidean*** configuration spaces (e.g. joint space of robotic arms).

Overview of My Work

- Planning with Topological constraints – Homotopy & Homology class constraints
  (AAAI 2010, RSS 2011)
- Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric
  (DARS 2010)
- Transformation for Efficient Optimal Planning in Environments with Non-uniform Metric
- Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow
  (RSS 2010, ICRA 2010)

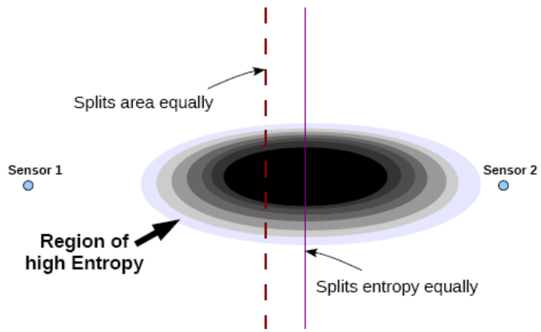
Entropy-based metric for exploration and coverage

Bhattacharya, Michael, Kumar

10th International Symposium on Distributed Autonomous Robotics Systems **(DARS 2010)**

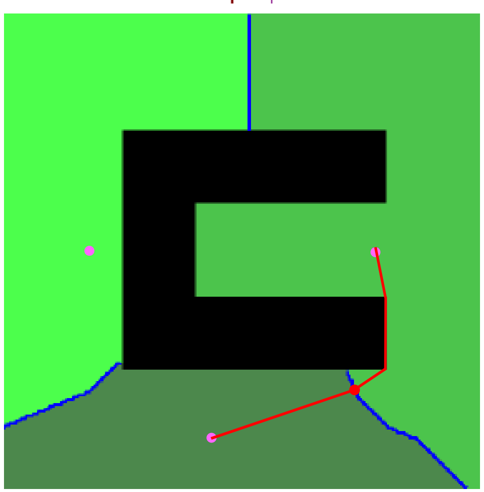
Recall: Voronoi tessellation $V_i(P) = \{\mathbf{q} \in \Omega \,|\, d(\mathbf{q}, \mathbf{p}_i) \leq d(\mathbf{q}, \mathbf{p}_j), \forall j \neq i\}$

Uncertainty (lack of knowledge of occupancy of a discretized cell) in an environment is modeled by ***entropy:*** $e(\mathbf{q}) = p(\mathbf{q}) \ln(p(\mathbf{q})) + (1 - p(\mathbf{q})) \ln(1 - p(\mathbf{q}))$

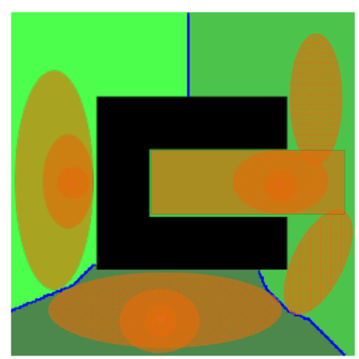


Non-Euclidean *d* – weighted by entropy:

$$d(\mathbf{p}, \mathbf{q}) = \min_{\gamma \in \Gamma(\mathbf{p}, \mathbf{q})} \int_{\gamma} e(\mathbf{r}) dl,$$



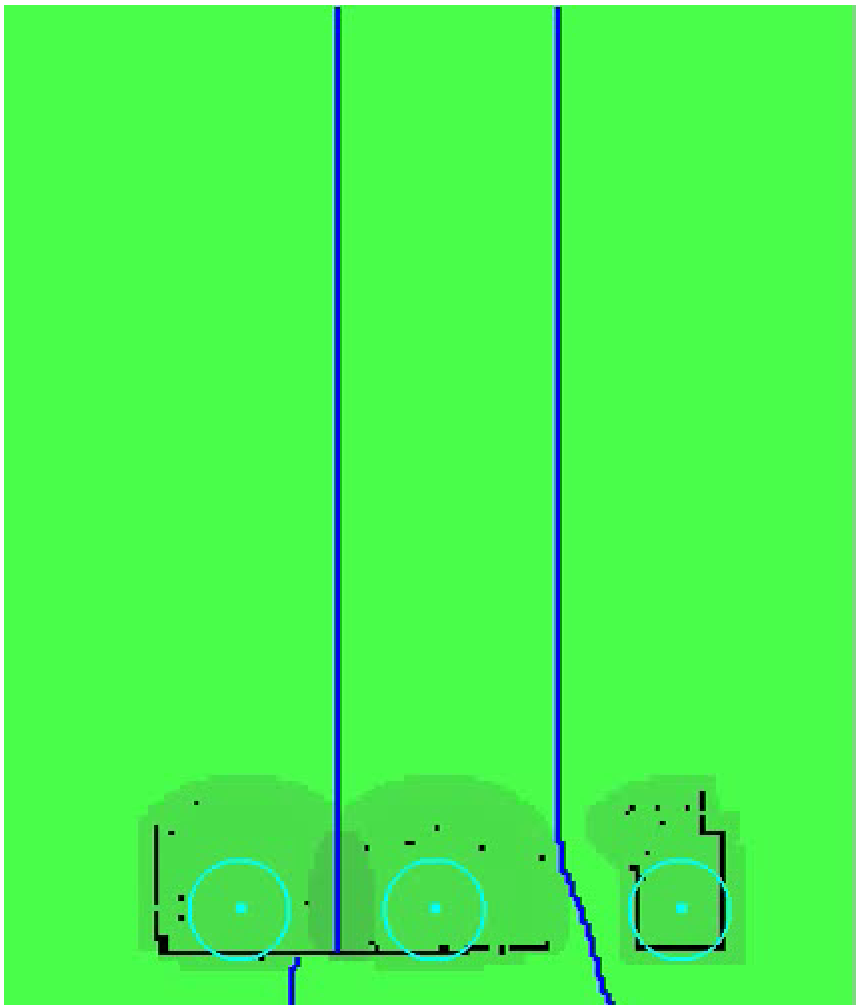
In non-convex environment, *d* is the "geodesic distance".



Solution:

Graph-search based wave-front expansion for determining Voronoi tessellations.

- We use a modified version of the Lloyed's algorithm for exploration.

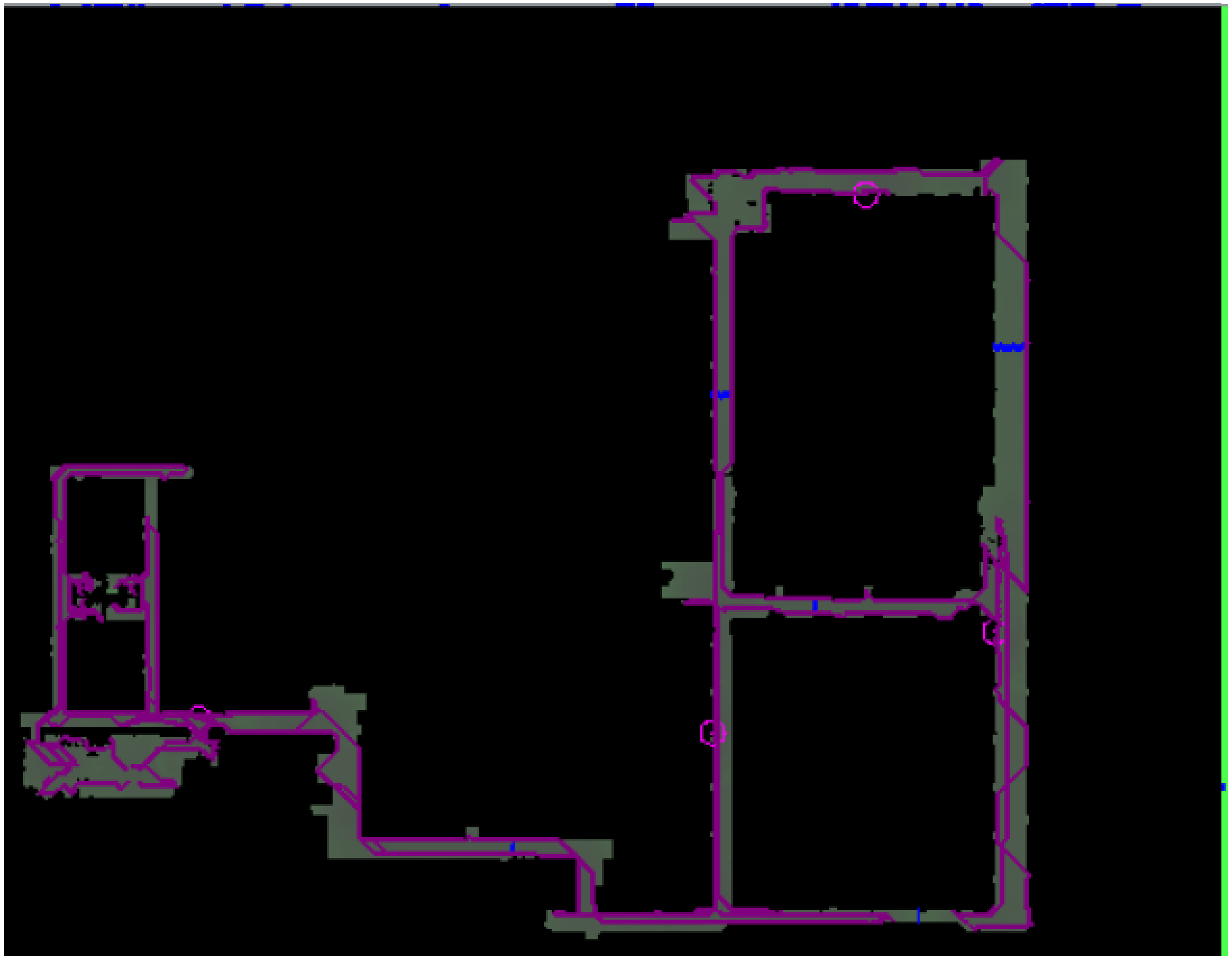


Note:

Lloyd's algorithm guarantees convergence to a <u>local</u> minima of

$$\sum_{i=1}^{n} \int_{W_i} f(d(\mathbf{q}, \mathbf{p}_i))\phi(\mathbf{q})d\mathbf{q},$$

Result



t = 2800 (convergence achieved)

4 robots in a 1000 x 783 uniformly discretized environment.

Each iteration (time-step) takes about 1.7 s.

(C++ implementation running on a single processor)

Future directions

- For the modified Lloyd's algorithm for non-convex regions we use an analog of generalized centroid (projected centroid). We would like to investigate the possibility of actual computation of ***generalized centroid***.

$$\mathbf{C}_{V_i}^{gen} = \underset{\mathbf{p}_i \in V_i}{\operatorname{argmin}} \int_{V_i} f(d(\mathbf{q}, \mathbf{p}_i))\phi(\mathbf{q})d\mathbf{q}$$

(proposed
-- feasibility to be investigated)

Overview of My Work

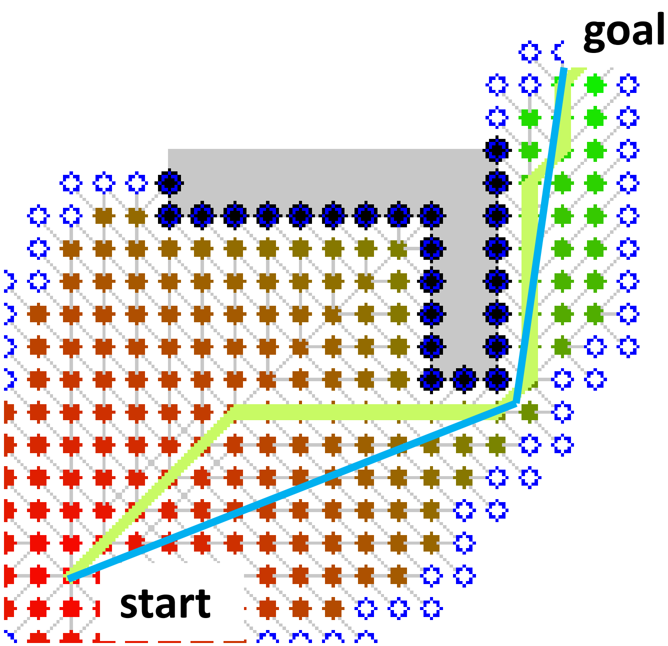
- Planning with Topological constraints – Homotopy & Homology class constraints
  (AAAI 2010, RSS 2011)
- Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric
  (DARS 2010)
- Transformation for Efficient Optimal Planning in Environments with Non-uniform Metric
- Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow
  (RSS 2010, ICRA 2010)

Graph-search techniques are well-suited for:

- Non-convex environments
- Non-uniform (non-Euclidean) metrics

BUT

Although the solution is ***least-cost in the graph***, it may not be so in the original continuous configuration space!



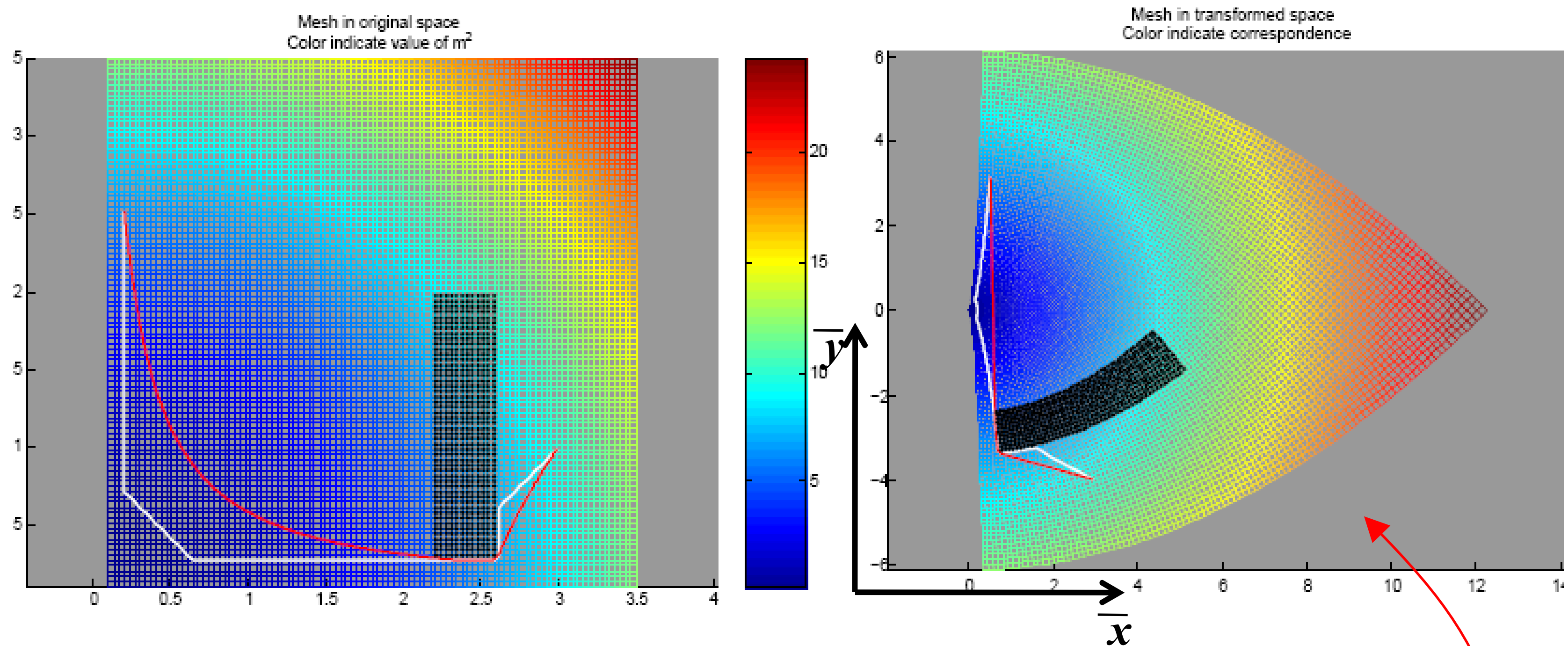
However, ***if the metric is flat*** (i.e. Euclidean), we can use ***visibility-based approaches***:

- Do post-processing
- Employ visibility graph
- Use theta-star algorithm *[Nash, et al.]*
- etc.

Question:

Given an arbitrary metric space, can we find a transformation to a flat metric space?

Study under progress – preliminary results!



$$g = (x^2 + y^2)\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \xrightarrow{\quad T \quad} \overline{g} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Isotropic, but non-uniform metric

$$\overline{x}(x, y) = Im(z^2/2) = xy$$

$$\overline{y}(x, y) = -Re(z^2/2) = (y^2 - x^2)/2.$$

Isometric embedding of *barred coordinates* in Euclidean plane

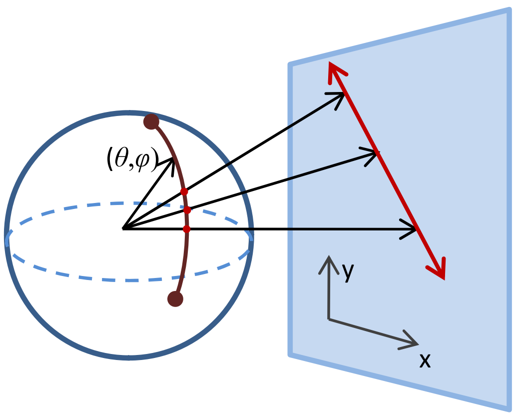
This can be written as $|z|^2$. This is hence a conformal map. This is the ***<u>same metric spaces with zero scalar curvature</u>*, *being described by 2 different coordinate charts.***

Relaxed question: Given a metric space, can we find a coordinate chart, whose natural embedding in Euclidean plane maps geodesics to straight lines (possibly ***non-isometrically***)?

A major future direction:

Other metric spaces to be investigated:

Real projective Plane (admits constant **positive curvature** metric)



$$(\theta,\varphi) \longrightarrow (x, y)$$

$$\begin{bmatrix} 1 & 0 \\ 0 & \sin^2(\theta) \end{bmatrix}$$

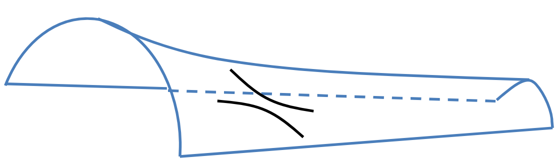
(spherical metric)

Geodesics map to "straight lines" on the plane (upon embedding using usual Euclidean metric), but ***non-isometrically***!

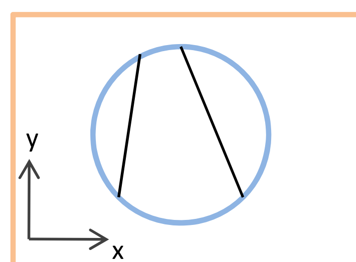
Hyperbolic Space (admits constant **negative curvature** metric)

Beltrami–Klein model:

- The whole hyperbolic plane is mapped to the interior of a circle
- Geodesics on hyperbolic plane maps to straight lines.



$\longrightarrow$



Future directions

- Characterization of metric spaces that has at least one representation (i.e. coordinate chart) whose ***natural embedding in Euclidean space maps geodesics to straight lines*** (may not be isometrically).

(major future direction)

Overview of My Work

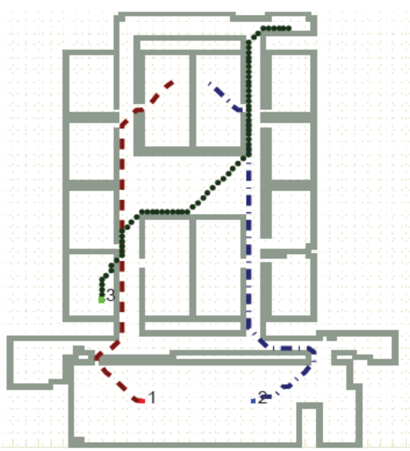
- Planning with Topological constraints – Homotopy & Homology class constraints
  (AAAI 2010, RSS 2011)
- Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric
  (DARS 2010)
- Transformation for Efficient Optimal Planning in Environments with Non-uniform Metric
- Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow
  (RSS 2010, ICRA 2010)

Distributed Optimization with Pairwise Constraints and its Application to Multi-robot Path Planning

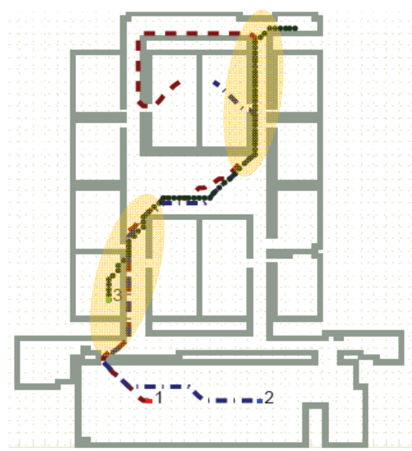
Bhattacharya, Kumar, Likhachev

Robotics: Science and Systems **(RSS 2010)**

A motivational example:



Unconstrained solution



Optimal plan satisfying communication constraints

Cluttered, non-convex environment

– discretization and graph-search based techniques desired for fast, optimal planning.

Robot configuration spaces coupled by constraints

– size of joint statespace increases exponentially with N (number of robots).

How to efficiently and optimally solve this huge problem?

Dimensional decomposition!

Problem definition: $\{\pi_1^*, \ldots, \pi_N^*\} = \mathrm{argmin}_{\pi_1 \ldots \pi_N} \sum_{j=1 \ldots N} c(\pi_j)$

(Goal directed navigation of N **heterogeneous robots**)

s.t. $\Omega_{ij}(\pi_i, \pi_j) = 0$ (e.g., time-parametrized distance constraint)

Subproblem: $\pi_r^{k+1} = \mathrm{argmin}_{\pi_r} \left[c_r(\pi_r) + \sum_{i=1 \ldots N,\, i \neq r} W_{ir}^{k+1} \Omega_{ir}(\pi_i^k, \pi_r) \right]$

Solved using discrete graph search for i^{th} robot.

How to increase the ***weight vector*** so as to guarantee i. convergence, ii. optimality?

$$W^{k+1} = W^k + \epsilon^k \, ComputeStepDirection(W^k, \{\pi\}^k, r) \qquad \pi_j^{k+1} = \pi_j^k \ \forall \ j \neq r$$

Separable optimal flow:

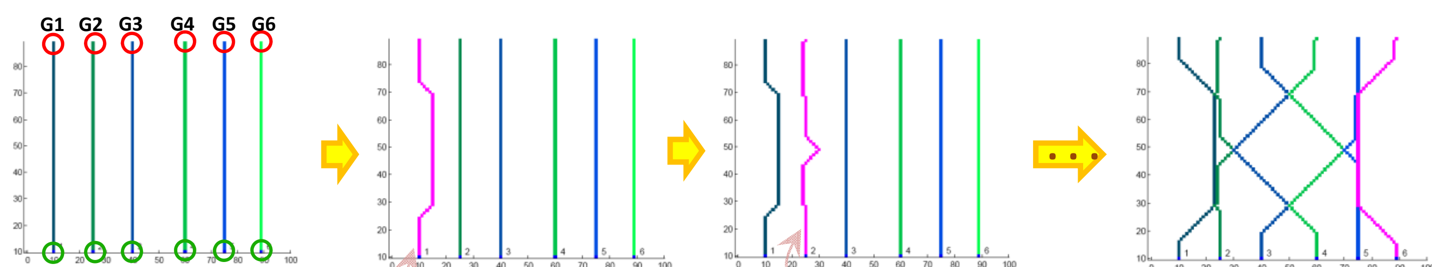
$$\Psi_r(W + \epsilon V, W) - \Psi_r(W, W) \le \Psi_r(W + \epsilon V, W + \epsilon V) - \Psi_r(W, W + \epsilon V)$$

and, $V_{ij} = 0, \ \forall \{i,j\}$ such that $r \notin \{i,j\}$

Ascent direction:

$$\sum_{\{ij\} \in \mathcal{P}^N} V_{ij} \Omega_{ij}(\overline{\Pi}_i(W), \overline{\Pi}_j(W)) > 0$$

Six robots planning iteratively to satisfy **rendezvous constraints** in an **empty environment**:



Iteration (k) = *0*

Iteration (k) = *1*
Planning robot (r) = *1*

Iteration (k) = *2*
Planning robot (r) = *2*

Iteration (k) = *12*
Final converged solution satisfying constraints

More details if time permits

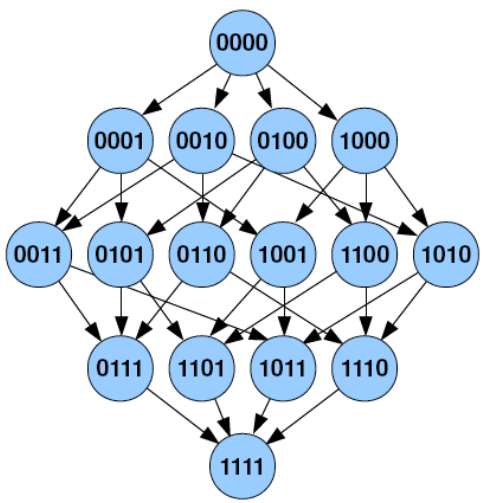
jump to slide

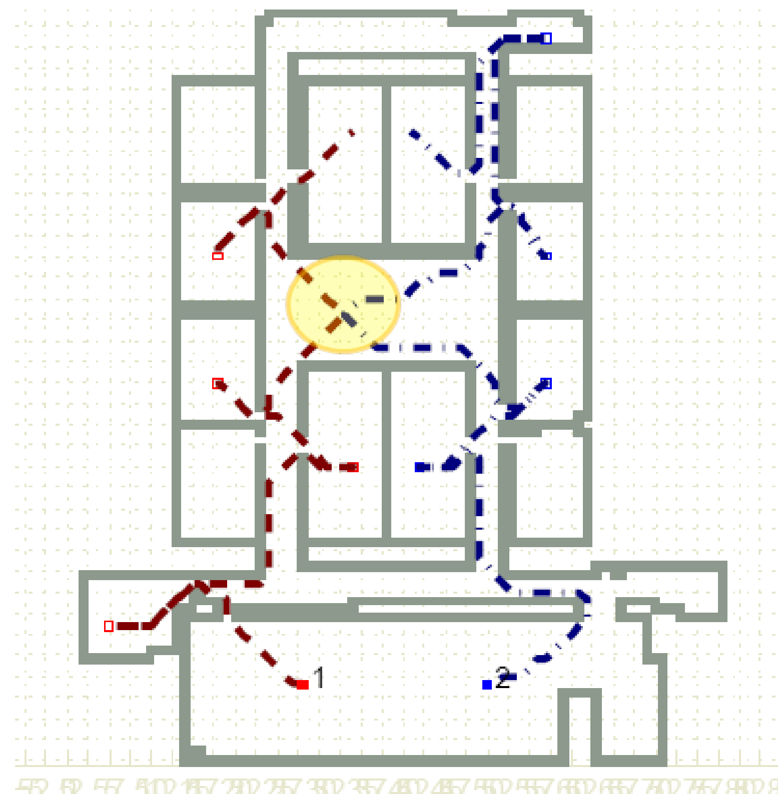
Additional complexity - ***Tasks***

Bhattacharya, Likhachev, Kumar

International Conference on Robotics and Automation **(ICRA 2010)**

Introduce the notion of *task graph*:





Modified graph to search in:

Product of configuration space ***discretization graph*** & the ***task graph***

Movies

Future directions

- To investigate the conditions under which "separable optimal flow" and "ascent direction" are guaranteed to exist.

(proposed future direction)

Proposed timeline

- **Until September/October'11:**
  1. Re-writing of the chi-homotopy in terms of homology and co-homology.
  2. How to find coordinate chart on metric spaces such that it's natural embedding in Euclidean plane maps geodesics to straight lines.

- **October'11 – January'12:**
  1. Investigation of conditions for existence of "separable optimal flow" and "ascent direction".
  2. Finding generalized centroid of non-convex regions

Acknowledgements

I would like to thank my advisors, my committee members and colleagues in the GRASP lab.

Also, I would like to acknowledge support of ONR, NSF, ARO, ARL

Codes available at

http://fling.seas.upenn.edu/~subhrabh/

Thank you!

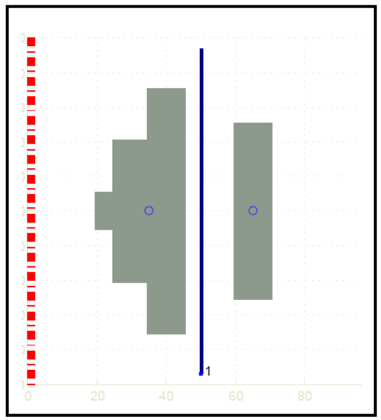
Questions?

Additional slides

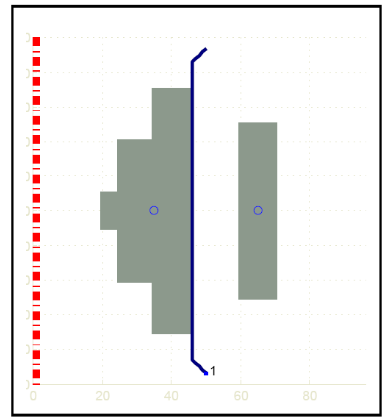
Results

(Non-Euclidean Cost function)

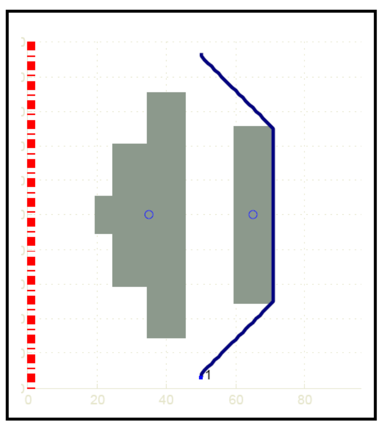
$$c = \int_{\tau} ds + w \int_{\tau} x(s) ds$$



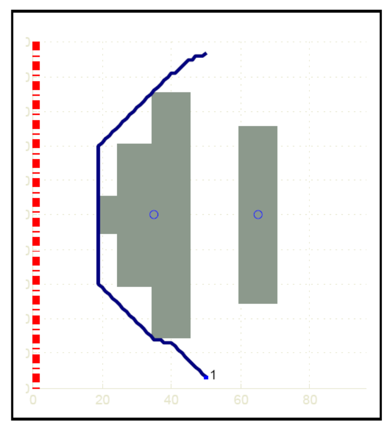
(a) $w = 0.0, \mathcal{B} = \{\}$



(b) $w = 0.01, \mathcal{B} = \{\}$



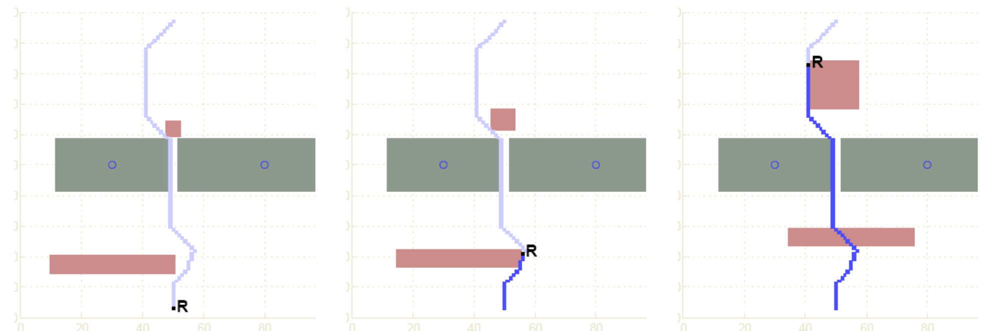
(c) $w = 0.0, \mathcal{B} = \{-8.41 + 8.41i\}$



(d) $w = 0.01, \mathcal{B} = \{-8.41 + 8.41i\}$

Results
(Planning with additional coordinates)
Planning in *X-Y-Time*

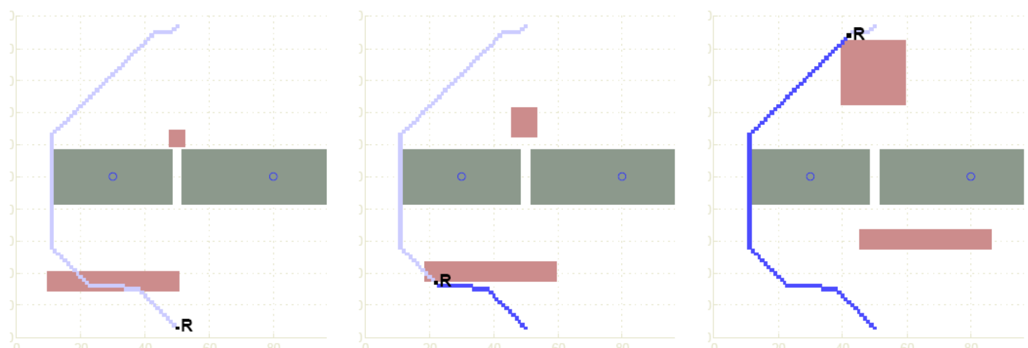
Planning in dynamic environment ***without*** homotopy class constraint



(f) $t = 1$ (g) $t = 19$ (h) $t = 81$

Planning in dynamic environment ***with*** a homotopy class blocked

Homotopy classes defined by taking projection on *X-Y* plane



(c) $t = 1$ (d) $t = 30$ (e) $t = 113$

Definitions

$\mathcal{N}^N = \{1, 2, \cdots, N\}$
$\mathcal{P}^N = \{\{1,2\}, \{1,3\}, \cdots, \{1,N\}, \{2,3\}, \{2,4\}, \cdots, \{N-1, N\}\}$
$\mathcal{P}_r^N = \{\{1,r\}, \cdots, \{r-1,r\}, \{r+1,r\}, \cdots, \{N,r\}\}$
V and W are vectors with $N(N-1)/2$ elements

$$\{\overline{\Pi}\}(W) := \arg\min_{\{\pi\}} \Big[\sum_{k\in\mathcal{N}^N} c_k(\pi_k) + \sum_{\{kl\}\in\mathcal{P}^N} W_{kl}\Omega_{kl}(\pi_k, \pi_l) \Big]$$

$$\Psi_r(W_1, W_2) \quad := \quad \min_{\pi_r} \Big[c_r(\pi_r) + \sum_{\{kr\}\in\mathcal{P}_r^N} W_{1,kr}\Omega_{kr}(\overline{\Pi}_k(W_2), \pi_r) \Big]$$

For a small ϵ, V is a ***Separable Optimal Flow Direction*** for Ψ_r at W iff:

$$\Psi_r(W+\epsilon V, W) - \Psi_r(W, W \leq \Psi_r(W+\epsilon V, W+\epsilon V) - \Psi_r(W, W+\epsilon V)$$
$$\Rightarrow (\epsilon V)^T \left[\Psi_r^{(1,1)}(W, W)\right] (\epsilon V) \geq 0$$
and, $V_{ij} = 0, \quad \forall\{i,j\}$ such that $r \notin \{i,j\}$

V is an ***Ascent Direction*** at W iff:

$$\sum_{\{ij\}\in\mathcal{P}^N} V_{ij}\Omega_{ij}(\overline{\Pi}_i(W), \overline{\Pi}_j(W)) > 0$$

Theorem 1: If the *Step Direction* returned by procedure *ComputeStepDirection* at the k^{th} iteration of the Algorithm, along with a small step size ϵ^k, define a *Separable Optimal Flow* at W^k for Ψ_{r_κ}, $\forall$ k, then $\forall$ k
$\{\pi_1^k, \ldots, \pi_N^k\} = \arg\min_{\{\pi\}} \left[\sum_{i\in\mathcal{N}^N} c(\pi_i) + \sum_{\{ij\}\in\mathcal{P}^N} W_{ij}^k \cdot \Omega_{ij}(\pi_i, \pi_j)\right]$. *i.e.* $\pi_i^k = \overline{\Pi}_i(W^k), \quad \forall i, k$

Theorem 2: If the condition in Theorem 1 holds, and the *Step Direction* returned by procedure *ComputeStepDirection* at the k^{th} iteration of the Algorithm is also an *Ascent Direction* at W^k, for all k, then the Algorithm converges to an optimal solution, if one exists.

Theorem 3: If the functions c_r and Ω_{ij} are differentiable up to second order, and $\Omega_{ij}(\pi_i, \pi_j)$ is of the form $G_{ij}(\pi_i - \pi_j)$, where G_{ij} is continuous, smooth and even, then we can compute a *Step Direction*, if one exists, that satisfy Theorems 1 & 2, at a given W^k.