Topological and Geometric Techniques in Graph Search-based Robot Planning

Subhrajit Bhattacharya

Under the guidance of

Prof. Vijay Kumar

Prof. Maxim Likhachev

University of Pennsylvania

GRASP LABORATORY

Collaborators

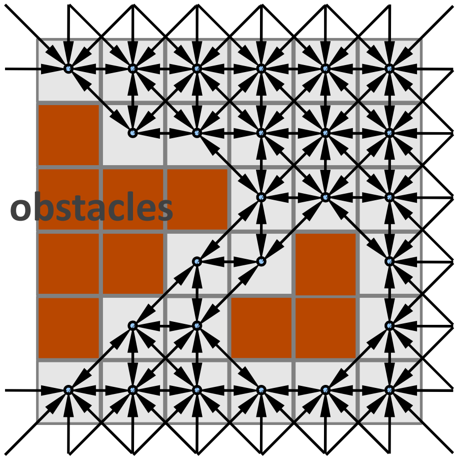
- Dr. D. Lipsky and Dr. R. Ghrist for collaboration on work related to algebraic topology.

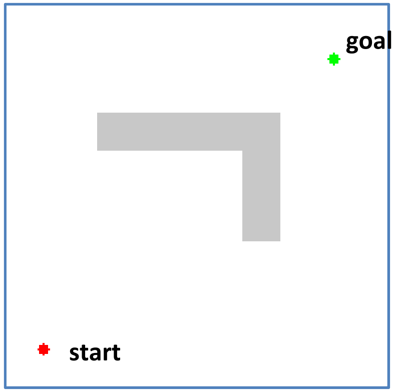
- Dr. N. Michael and Dr. L. Pimenta for collaboration on work related to voronoi decomposition and exploration.

Approaches in Path planning problem in robotics

Discretize approach

Graph Search-based

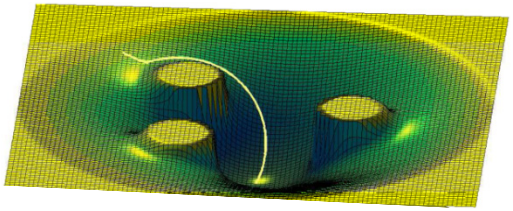




Use ***graph search algorithms***
To find optimal path in graph

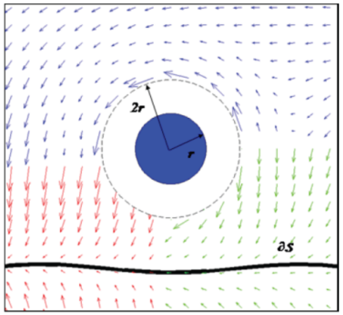
i. Fast, efficient, ***robust***.
ii. Complete (will find solution if exits)
iii. Indifferent to ***non-convexity, holes*** in the environment
iv. Works well for ***non-Euclidean metric***
v. Globally ***optimal*** (in the graph)

Continuous approaches



Navigation Function

(Koditschek, Rimon, ‘90)



Global Vector Field

(A. Hsieh, ’07)

$$\frac{d^2x^\gamma}{d\tau^2} + \Gamma^\gamma{}_{\beta\mu}\frac{dx^\beta}{d\tau}\frac{dx^\mu}{d\tau} = 0$$

Solve Geodesic Equation

(for optimality)

i. ***Difficult constructions*** for general environments.
ii. Susceptible to ***non-convergence, slow convergence*** or getting stuck at ***local minima*** (due to obstacles/holes).
iii. Difficult to guarantee ***optimality*** (geodesic between two points, and with holes in environment, is difficult to find)

Graph search based approach is the preferred, robust solution, **but**….

- Discretization + graph construction discards all ***topological information*** about the environment.
- By restricting to the graph we also lose a lot of information about the original ***metric*** in the underlying space.
- Size of graph, and complexity of search algorithms increase ***exponentially with dimension*** of configuration space.

The overall big question:

> Can we use some of the ideas from continuous approaches to make up for the drawbacks of search-based approaches?

Overview

1. Planning with Topological constraints – Homotopy & Homology class constraints

2. Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric

3. Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow

4. Transformation for Efficient Optimal Planning in Environments with Non-Euclidean Metric

Overview

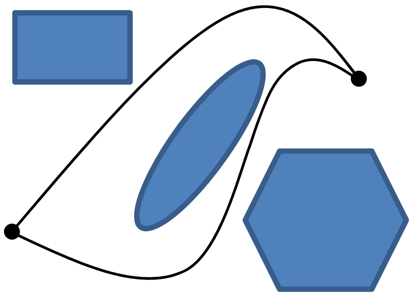
1. Planning with Topological constraints – Homotopy & Homology class constraints

2. Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric

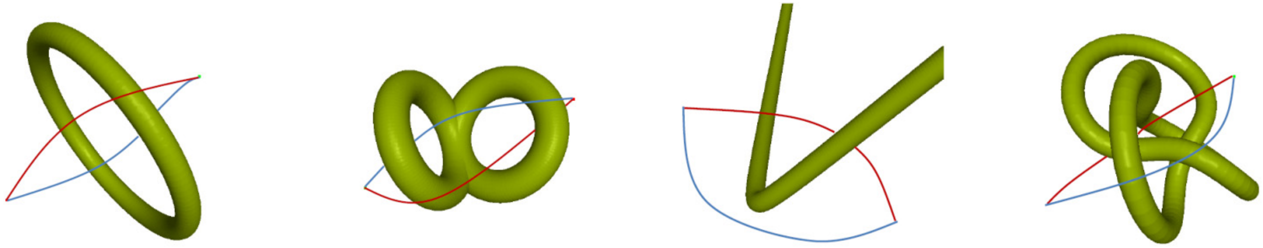
3. Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow

4. Transformation for Efficient Optimal Planning in Environments with Non-Euclidean Metric

Motivation: Homotopy Classes of Trajectories

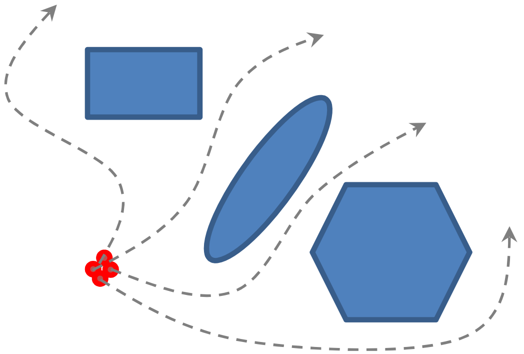


Different homotopy class of trajectory in 2D

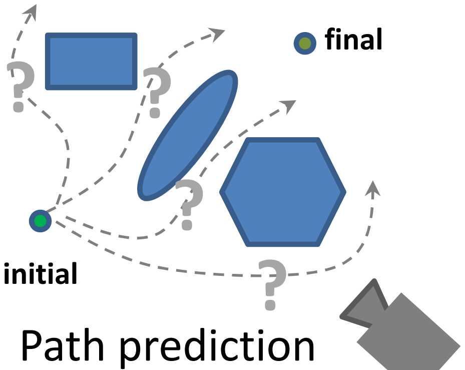


Trajectories in different homotopy classes in 3D

Applications in robotics



Deploying multiple agents



Path prediction

e.g. in tracking dynamic agents through multiple occlusions

J. Shi, et al.

Planning trajectories in specific homotopy class (e.g. trajectory through a window)

D. Mellinger, et al.

Homotopy Vs. Homology – Similar equivalence relations

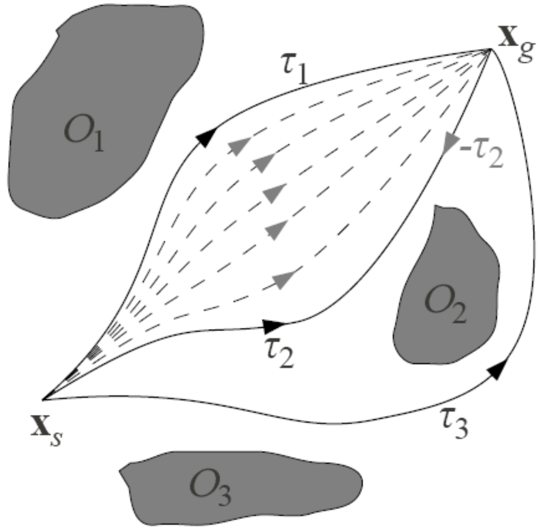


Illustration of equivalence of homotopy

$$\tau_1 \sim \tau_2 \nsim \tau_3$$

Since there is a continuous sequence of trajectories between τ_1 and τ_2.

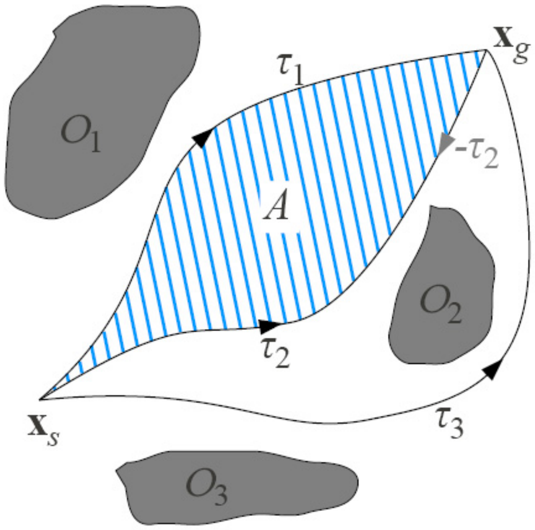
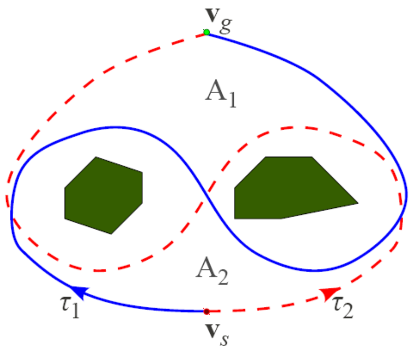


Illustration of equivalence of homology

$$\tau_1 \sim \tau_2 \nsim \tau_3$$

Since there exists an area A, the boundary of which is τ_1 U -τ_2.

- Similar & related concepts, but subtly different.
- Homotopy is intuitive, but computationally difficult. Homology is more abstract, but computationally favorable.
- ***Homotopic implies homologous***, but converse may not always be true!



Example where trajectories are homologous, but not homotopic.

Part-A

(Our Contribution)

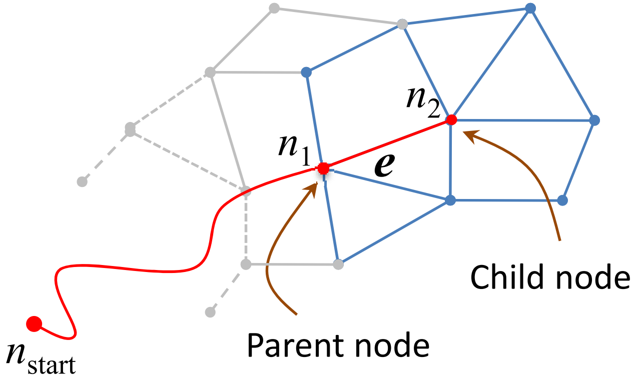
- Design an **efficient representation** of homology classes of trajectories in 2D and 3D.
- Use the representation for :
  - Finding least cost paths with homology class constraints (solve the **constrained optimization problem**).
  - Explore different homotopy classes in a 2/3D configuration space.
- Integrate the representation with **graph-search based planning** algorithms.
  - Plan for **optimal cost paths**, for **arbitrary cost function** (not necessarily Euclidean distances), arbitrary **discretization schemes** (Uniform, unstructured, triangulation, visibility graph, etc.) using any **graph search algorithms** (Dijkstra's, A*, D*, ARA*, etc.).

Incorporating topological constraints in planning problems

<u>Key idea</u>: Given a D-dimensional configuration space, $\mathcal{C}$, how to find a **differential 1-form,** $\omega \in \Omega^1_{dR}(\mathcal{C})$, such that for **any given trajectory,** τ, the value of the integral $H(\boldsymbol{\tau}) = \int_\tau \omega$ can be used to identify the **<u>homology class</u> of $\boldsymbol{\tau}$**.

Additive in $\boldsymbol{\tau}$

Can be used efficiently in graph search-based planning:



n_1 in "closed" list (expanded)

– next node to expand is n_2 ....

Easy to compute $H\left(\widetilde{n_{start}n_2}\right)$ from $H\left(\widetilde{n_{start}n_1}\right)$:

$$\underbrace{H\left(\widetilde{n_{start}n_2}\right) = H\left(\widetilde{n_{start}n_1}\right) + \int_{\boldsymbol{e}} \omega}$$

Analogous to cost computation:

$$c(n_{start}n_2) = c(n_{start}n_1) + \text{cost}(\boldsymbol{e})$$

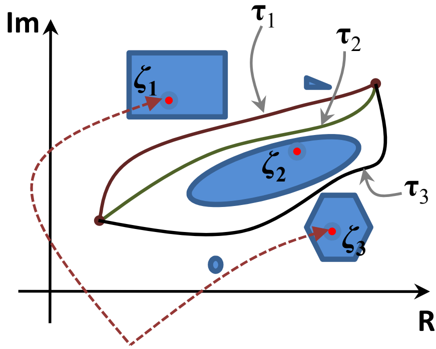
Solving the problem on Two Dimensional Plane

Key ideas:

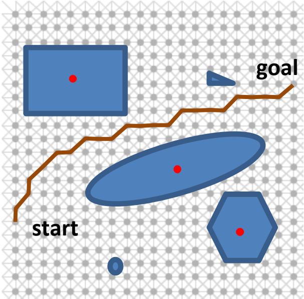
1. Represent the configuration space by a complex plane
2. Construct an analytic function with singularties at “representative points”.
3. Leverage Cauchy Integral and Residue Theorems to design an <u>additive</u> *complete homology class invariant*.

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

$$H(\tau) = \int_\tau \underbrace{\mathcal{F}(z)dz}_{\omega \text{ (diff. 1-form)}}$$

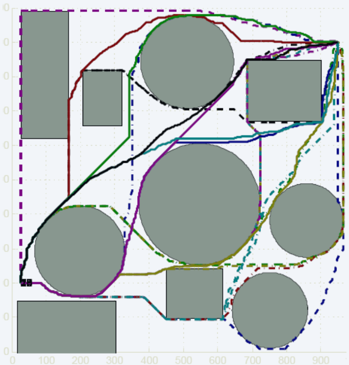


“Representative points” inside obstacles.

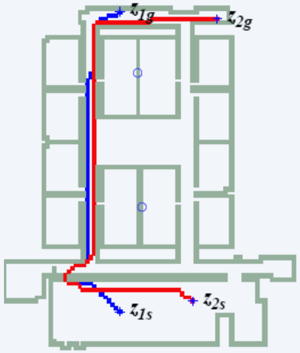


Can be used for **graph-search based planning** with **H-signature constraints**.

Applications & examples:

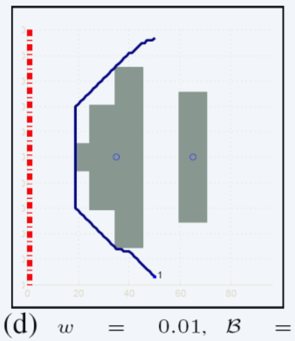


Homotopy class exploration in a large environment (1000x1000 discretized)

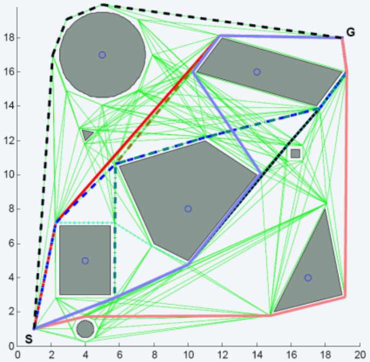


Planning with H-signature constraints *(visibility constraint)*

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

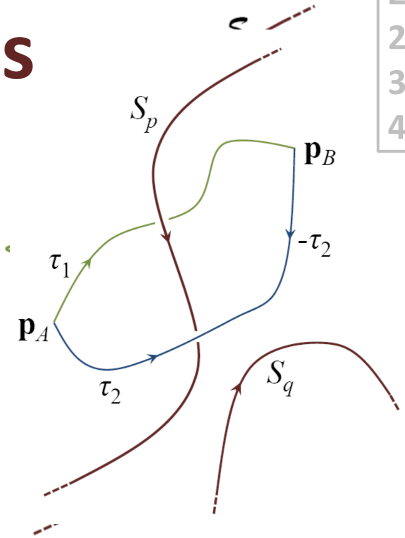


Planning with non-Euclidean metric and homology class constraints



Planning on a *Visibility Graph*

Solving the problem in Three Dimensional Euclidean Space with Obstacles



Key ideas:

1. Exploit theorems from Electromagnetism:

Biot-Savart's Law

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

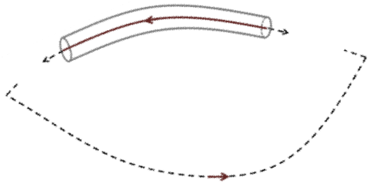
Ampere's Law

$$\Xi(\mathcal{C}) := \int_{\mathcal{C}} \underbrace{\mathbf{B}(\mathbf{l})\cdot d\mathbf{l}}_{\omega\ \text{(diff. 1-form)}} = \mu_0 I_{encl}$$

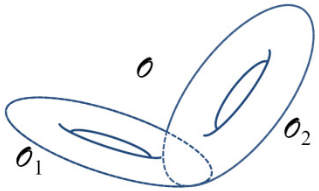
where, $\mathbf{B}$: Magnetic field vector
μ_0: Magnetic constant (can be chosen as 1 with proper choice of units)

2. Model the skeleton of each genus-1 obstacle as a current carrying conductor:

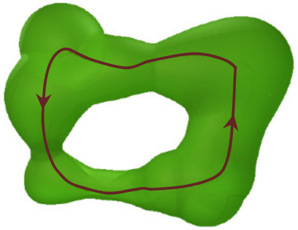
Arbitrary obstacles



Close unbounded obstacles



Virtually decompose genus > 1 obstacles



Construct skeleton of genus-1 obstacle and model that as a current carrying conductor

3. Leverage the integral of Ampere's law to design an additive *complete homology class invariant:*

Virtual Magnetic Field due to *i*th skeleton

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

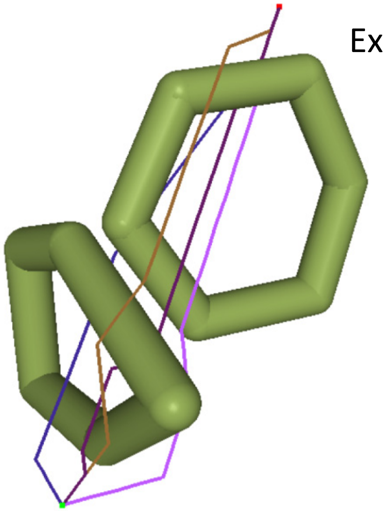
h*-signature of trajectory *τ

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

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

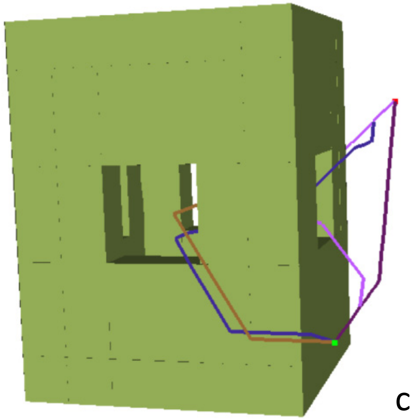
Results for the 3D case

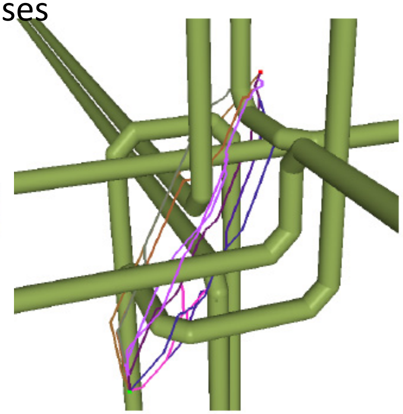
Exploration of multiple homotopy classes:



Exploration of 4 homotopy classes in presence of 2 SHIOs

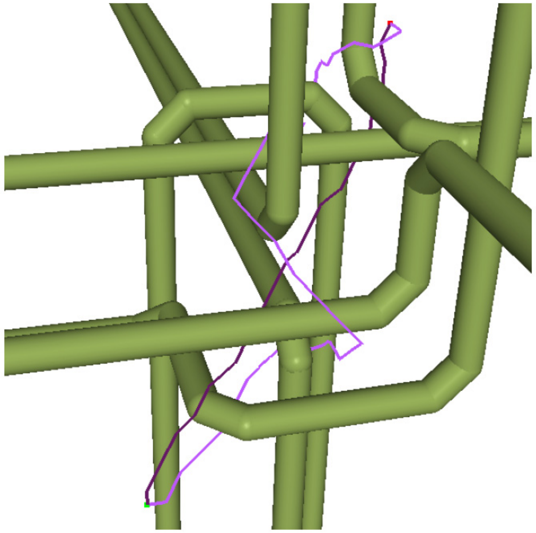
Exploration of 4 homotopy classes in presence of 4 SHIOs:



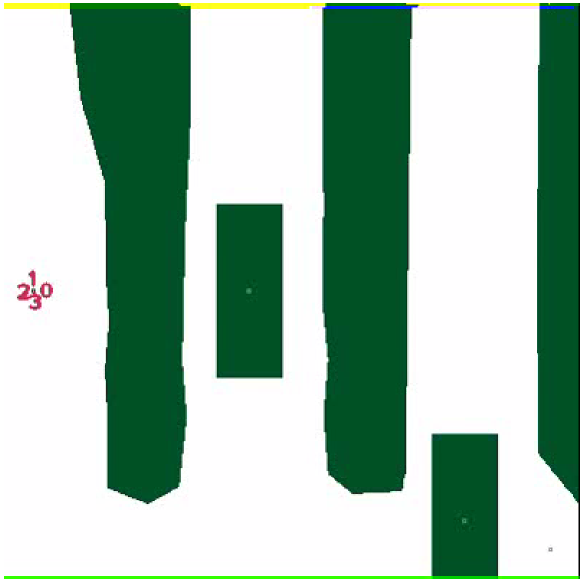


Exploration of 10 homotopy classes in presence of 7 SHIOs

Planning with H-signature constraint:



Planning in X-Y-Time configuration space:



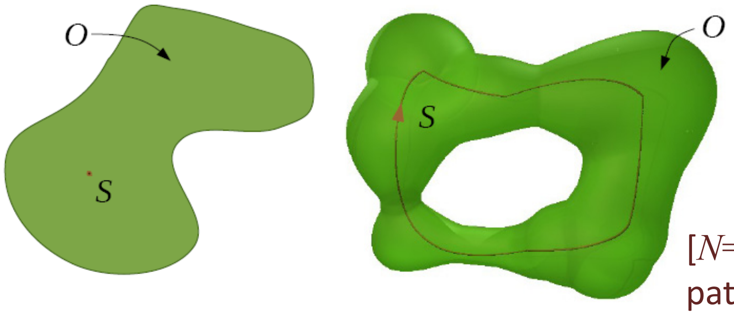
Part - B

Using tools from Algebraic Topology:

- Establish theoretical justification behind the previously described constructions:
  - Justify the replacement of obstacles by “representative points” or “skeletons”.
  - Establish a connection between the proposed differential 1-forms, linking number and homology classes.
  - To show that using the formulae mentioned, we in fact computed *complete invariants* for *homology classes* of trajectories.
- To generalize the method for >3 dimensional Euclidean space with obstacles.

We will discuss the intuitive concepts. More details of the proofs are in thesis.

Replacements for Obstacles



[N=2 for all robot path planning problems.]

A basic result from algebraic topology:

If S is a subspace of O such that inclusion map, $i{:}S \to O$, induces isomorphisms $H_{D-N}(S) \cong H_{D-N}(O)$

(*e.g.* deformation retracts)

Question: What can we say about the inclusion $\bar{i} : (\mathbb{R}^D - O) \hookrightarrow (\mathbb{R}^D - S)$?

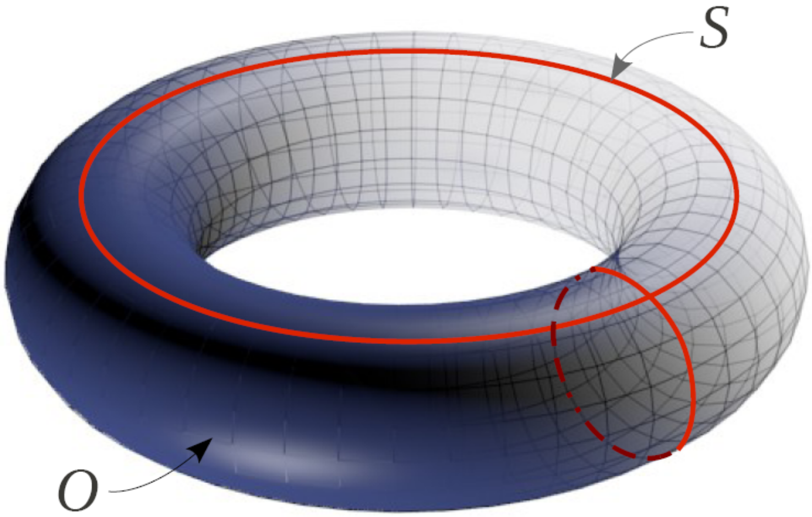
In particular, is the induced map, $\bar{i}_{*:N-1} : H_{N-1}(\mathbb{R}^D - O) \to H_{N-1}(\mathbb{R}^D - S)$, an isomorphism?

– it is, for “nice” spaces like orientable manifolds.

[Exact conditions: compact, locally contractible, orientable]

What can we do when there is no *deformation retract* to the required dimensional S ?

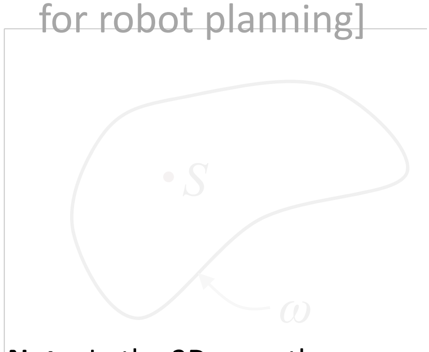
Example: A hollow (or thickened) torus in D=3.



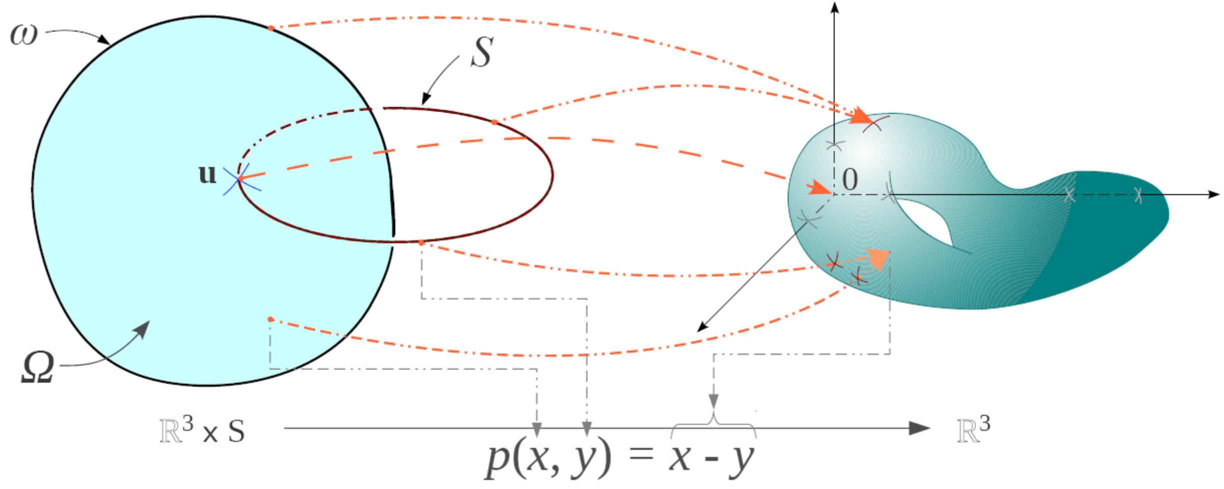
It is sufficient to choose generating (D-N)-cycles of homology group of O as the S.

Linking Number

[S is (D-2)-dimensional for robot planning]



Note: In the 2D case, the map p is simply a translation to the origin



What we had previously done by taking the integrals is compute ***linking number*** between closed loops ω and the *representative points* (in $\mathbb{R}^2$) or *skeletons* (in $\mathbb{R}^3$), represented by S.

- Definition of linking number is derived from definition of ***intersection number***:

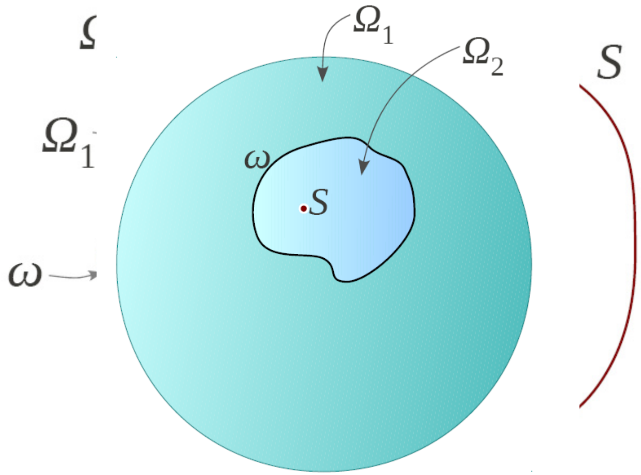
$$L(\omega, S) := I(\Omega, S), \text{ s.t., } \omega = \partial\, \Omega$$

[Seifert, et al., ‘80]

- A formal algebraic definition requires that we define a map, p: $\mathbb{R}^3 \times S \rightarrow \mathbb{R}^3$.

[A.Dold, ‘95]

Side-note: Uniqueness of linking number:



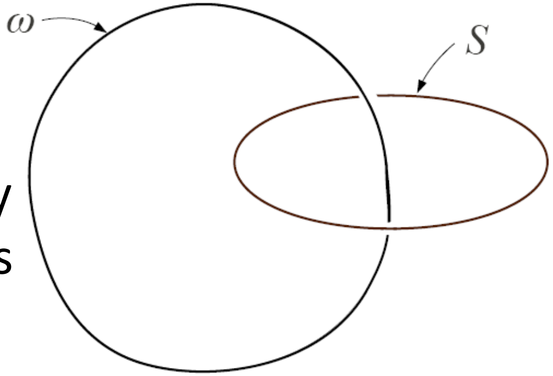
Computation of linking number:

Take advantage of the map p – the co-domain of p is much simpler and have well-known closed but non-exact differential (D-1)-forms:

If $\eta_0 \in \Omega^{D-1}_{dR}(\mathbb{R} - \{0\})$ s.t., $[\eta_0]$ is a generator of $H^{D-1}_{dR}(\mathbb{R} - \{0\})$, the linking number is given by,

$$L(\omega, S) = \int_{\omega \times S} p^*(\eta_0) \quad \left(= \int_{p(\omega \times S)} \eta_0 \right)$$

Complete Invariant for Homology class



Theorem (simplified statement):

If S is a path-connected manifold (of the appropriate dimension), then the linking number $L(\omega, S)$ precisely tells us about the *homology class* of ω in ($\mathbb{R}^D$- S) – it's a bijective map.

Thus, a *complete invariant* for homology class of ω is,

$$\int_{\omega \times S} p^*(\eta_0)$$

The required differential 1-form that can be integrated over ω :

$$\psi_S = \int_S p^*(\eta_0)$$

Explicit formula (in D=2, N=2 case

A particular (symmetric) choice of η_0:

$$\begin{aligned} \psi_{\mathbf{S}} &= U_1^1(\mathbf{x})\ \mathrm{d}x_2 + U_1^2(\mathbf{x})\ \mathrm{d}x_1 \\ &= \frac{1}{2\pi}\ \frac{(x_1 - s_1)\ \mathrm{d}x_2 - (x_2 - s_2)\ \mathrm{d}x_1}{|\mathbf{x} - \mathbf{S}|^2} \\ &= \frac{1}{2\pi}\ Im\left(\frac{1}{z - \mathbf{S}_c}\ \mathrm{d}z\right) \end{aligned}$$

$$U_1^1(\mathbf{x}) = \frac{1}{2\pi}\ (-1)^{2-2+1+1}(1)\frac{x_1 - S_1}{|\mathbf{x} - \mathbf{S}|^2} = \frac{1}{2\pi}\ \frac{x_1 - s_1}{|\mathbf{x} - \mathbf{S}|^2}$$

$$U_1^2(\mathbf{x}) = \frac{1}{2\pi}\ (-1)^{2-2+2+1}(1)\frac{x_2 - S_2}{|\mathbf{x} - \mathbf{S}|^2} = -\frac{1}{2\pi}\ \frac{x_2 - s_2}{|\mathbf{x} - \mathbf{S}|^2}$$

S,

$$\boxed{\psi_S = \sum_{k=1}^{D} \sum_{\substack{j=1 \\ j \neq k}}^{D} U_j^k(\mathbf{x}; S\ )\ \mathrm{d}x_j}$$

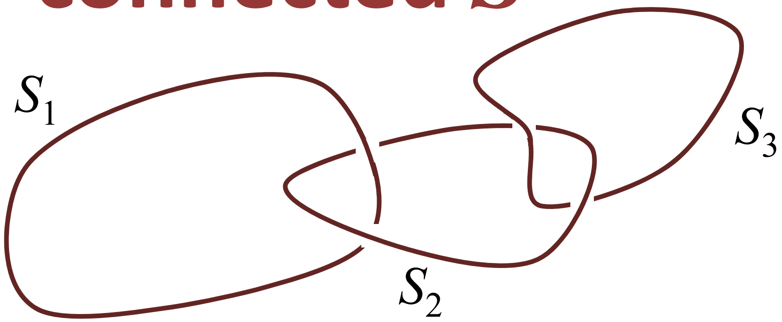
Reduces to known formulae upon plugging D=2,3

where,

$$U_j^k(\mathbf{x}; S) = (-1)^{k-j-1-\mathbf{is}(j<k)} \int_S \mathcal{G}_k(\mathbf{x} - \mathbf{x}')\ \mathrm{d}x'_1 \wedge \mathrm{d}x'_2 \cdots \wedge \widehat{x'_j, x'_k} \wedge \cdots \wedge \mathrm{d}x'_D$$

Multiple Path-connected S

$$\widetilde{S} = \bigcup_{i=1}^{m} S_i$$



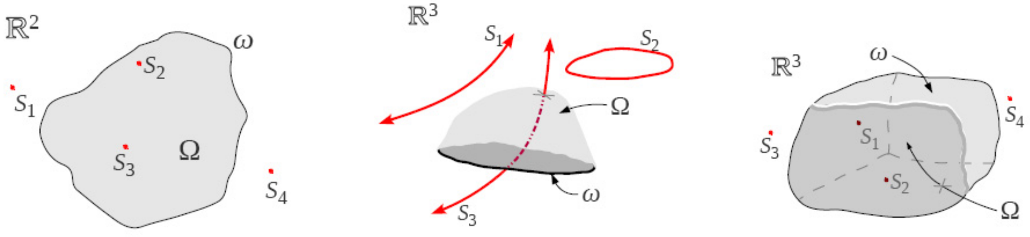
Theorem:

$$H_{N-1}(\mathbb{R}^D - \widetilde{S}) \;\approx\; \bigoplus_{k=1}^{m} H_{N-1}(\mathbb{R}^D - S_k)$$

is induced by direct sum of inclusion maps $\tilde{i}_k : (\mathbb{E}^D - \widetilde{S}) \hookrightarrow (\mathbb{E}^D - S_k)$

This implies that the *H*-signature (complete invariant for homology class) is simply,

$$\mathcal{H}(\tau) = \int_\tau \begin{bmatrix} \psi_{S_1} \\ \psi_{S_2} \\ \vdots \\ \psi_{S_m} \end{bmatrix}$$



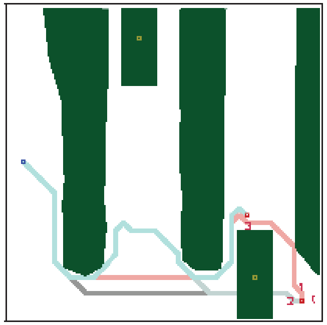
(a) $D = 2, N = 2$ (b) $D = 3, N = 2$ (c) $D = 3, N = 3$

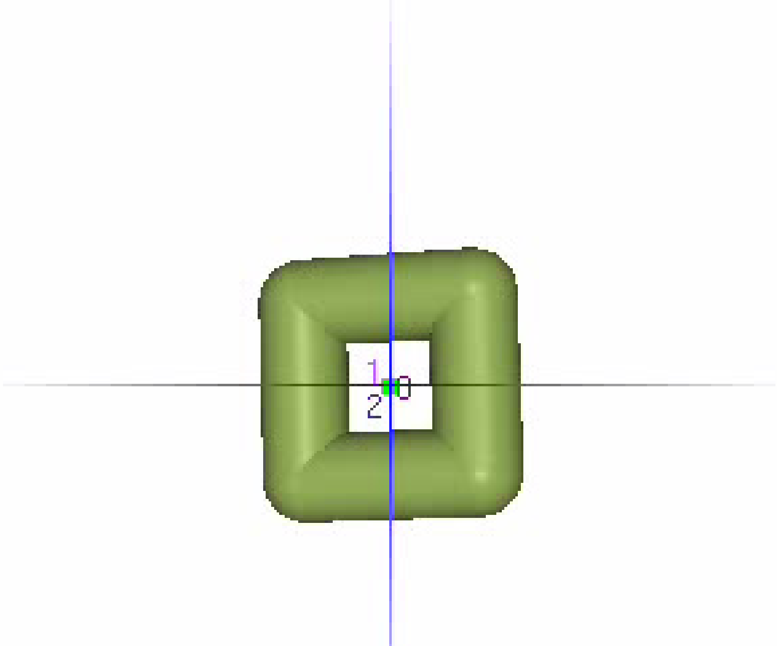
Achievement: Found a way to compute homology class of (N-1)-dimensional closed manifolds embedded in D-dimensional Euclidean ambient space with punctures.

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

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

Recall the X-Y-Time example:





Future direction:
Similar ways of determining homology classes of trajectories in punctured non-Euclidean spaces (e.g. configuration space of a robotic arm), and use search-based approaches for that.

Overview

1. Planning with Topological constraints – Homotopy & Homology class constraints

2. Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric

3. Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow

4. Transformation for Efficient Optimal Planning in Environments with Non-Euclidean Metric

Metric in Search-based Robot Planning Problems

- Optimal goal-directed path planning

  (A*, Dijkstra’s, other search algorithms)

  – well-understood, already discussed.

- Coverage problems
- Exploration problems

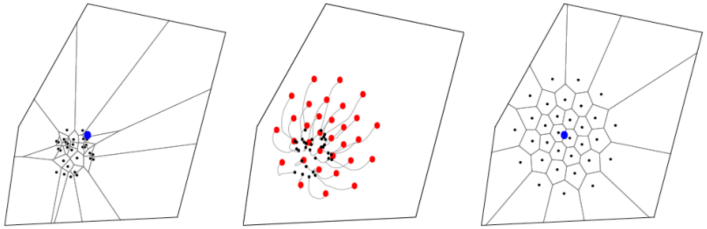
Related Work (coverage)

Coverage problem in robotics:

1. Lloyd’s Algorithm and its Continuous Time Version

S. P. Lloyd, “Least squares quantization in PCM,” IEEE Trans. Inf. Theory, vol. 28, pp. 129–137, 1982.

J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” IEEE Trans. Robot. Autom., vol. 20, no. 2, pp. 243–255, Apr. 2004.



2. Lloyd’s Algorithm in non-convex environments

(With polygonal obstacles:

Gradient descent approach for moving towards generalized centroid)

L. C. A. Pimenta, V. Kumar, R. C. Mesquita, and G. A. S. Pereira, “Sensing and coverage for a network of heterogeneous robots,” in Proc. of the IEEE Conf. on Decision and Control, Cancun, Mexico, Dec. 2008, pp. 3947–3952

(Geometric, difficult to compute in non-polygonal environment, not suited for exploration.)

Part A

Coverage in Non-convex Environments with non-Euclidean Metric

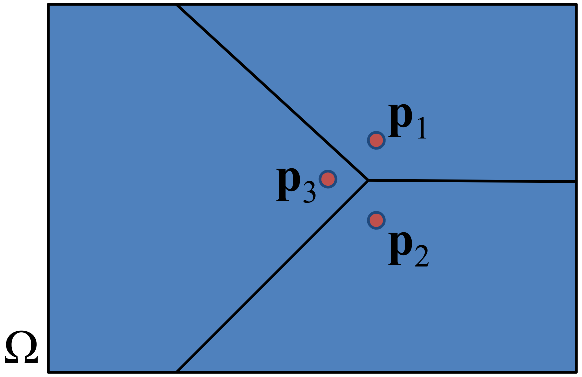
- Voronoi Tessellation, and
- Lloyd’s algorithm,

in

non-convex environments with possibly ***non-Euclidean metric***

Assignment for Exploration and Coverage (*known environment*)

Voronoi Tesellation:



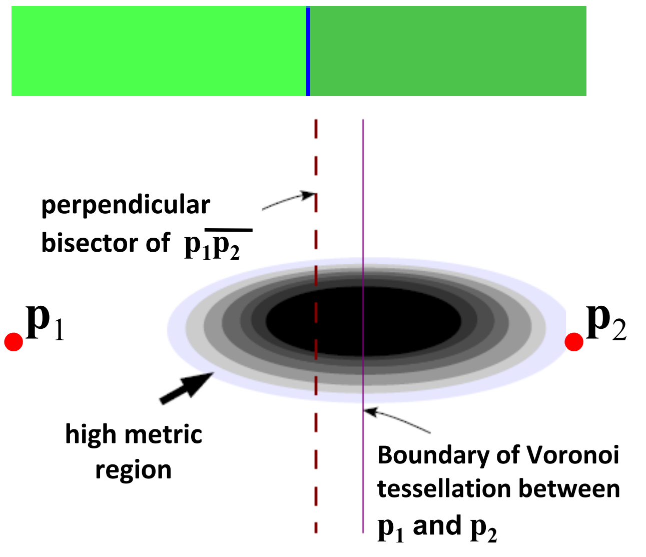
$$V_i(P) = \{\mathbf{q} \in \Omega \,|\, d(\mathbf{q}, \mathbf{p}_i) \leq d(\mathbf{q}, \mathbf{p}_j), \forall j \neq i\}$$

metric

In convex environments and Euclidean metric: Tessellation boundaries are perpendicular bisectors of lines joining the robots

In non-convex environments:

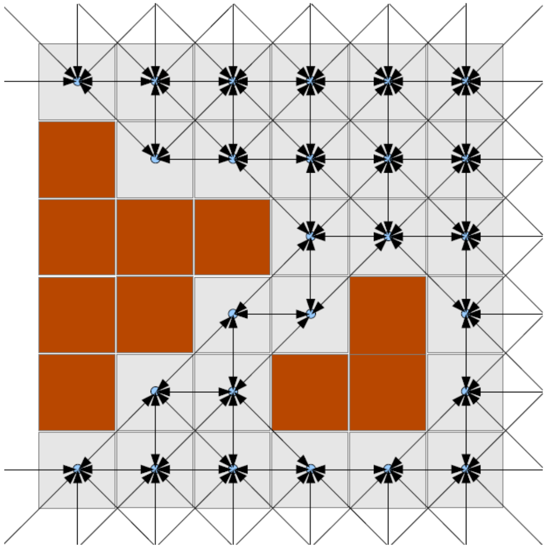
d is the *Geodesic* Distance


perpendicular bisector of $\overline{\mathbf{p}_1\mathbf{p}_2}$
$\mathbf{p}_1$
$\mathbf{p}_2$
high metric region
Boundary of Voronoi tessellation between $\mathbf{p}_1$ and $\mathbf{p}_2$

Due to non-Euclidean metric

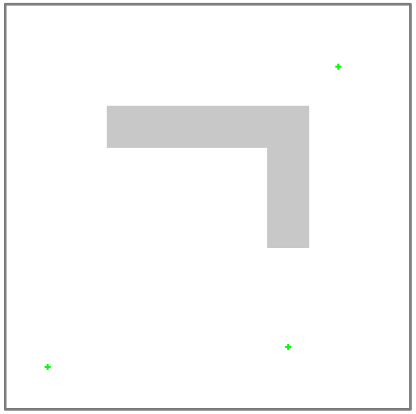
Due to non-convexity

Search-based algorithm for Computing Geodesic Voronoi Tesellation



Discretize the environment,
and form a graph, $\mathcal{G}$

For each agent, i, expand nodes of $\mathcal{G}$,
starting from $\mathbf{p}_i$
using Dijkstra's Algorithm

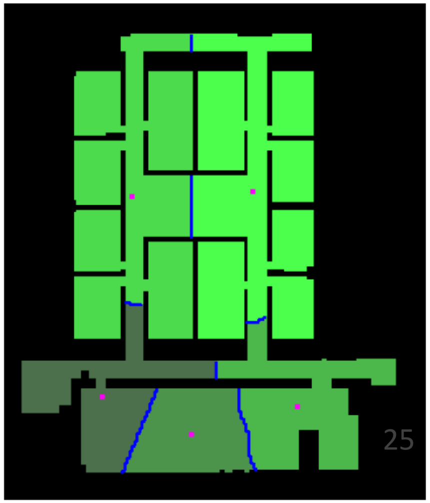


Result: We obtain $g_i(\mathbf{q})$ –
the Geodesic distance of each
node $\mathbf{q}$ in $\mathcal{G}$ from $\mathbf{p}_i$

Geodesic Voronoi Tesellation:

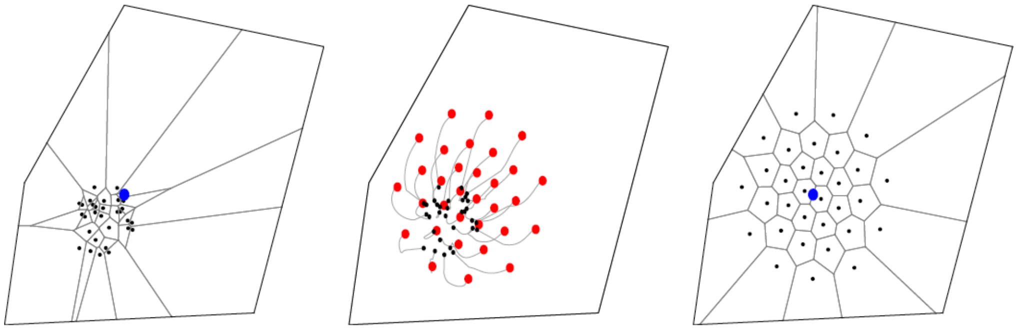
$$V_i = \{\mathbf{q} \in \mathcal{V}(\mathcal{G}_\Omega) \,|\, g_i(\mathbf{q}) \leq g_j(\mathbf{q}), \forall j \neq i\}$$

Easy to incorporate non-Euclidean metric d –
weigh the edges of the graph with the metric.



Mathematical basis for Lloyd's Algorithm

Lloyd's Algorithm for Optimal Coverage:



(Cortes, et al., MED '02)

Minimize

$$\mathcal{H}(P, W) = \sum_{i=1}^{n} \mathcal{H}(\mathbf{p}_i, W_i)$$

Measure of how bad the coverage is

metric

weight/density function

$$= \sum_{i=1}^{n} \int_{W_i} f(d(\mathbf{q}, \mathbf{p}_i))\phi(\mathbf{q})d\mathbf{q} \qquad [\text{where, } f(x) = x^2\,]$$

General solution *(in any metric, any type of space):*

$$W_i = V_i, \quad \mathbf{p}_i^* = \operatorname*{argmin}_{\mathbf{p}_i \in (V_i \cup \partial V_i)} \int_{V_i} f(d(\mathbf{q}, \mathbf{p}_i))\phi(\mathbf{q})d\mathbf{q}$$

~~In convex tessellation with Euclidean metric,~~

~~***centroid*** → $\mathbf{p}_i^* = \frac{\int_{V_i} \mathbf{q}\phi(\mathbf{q})d\mathbf{q}}{\int_{V_i} \phi(\mathbf{q})d\mathbf{q}}$.~~

~~Control law (follow gradient of $\mathcal{H}$):~~

~~$\mathbf{u}_i = -k(\mathbf{p}_i - \mathbf{p}_i^*)$~~

generalized tessellation

need to solve this directly

In non-convex tessellation, non-Euclidean metric (d).

Solution is called "generalized centroid" - extremely difficult to solve

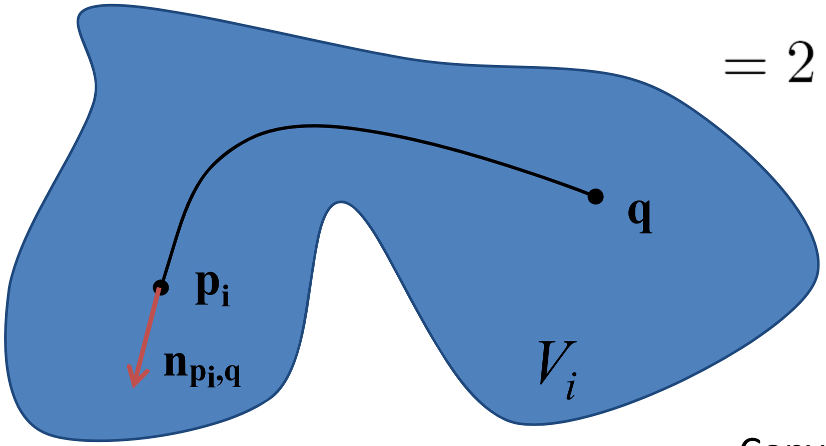
Lloyd's Algorithm via Direct computation of Gradient

Instead of finding the minima of $\mathcal{H}$ and hence construct a control law, we try to **find the gradient of $\mathcal{H}$ and follow the gradient**.

Gradient of $\mathcal{H}$:

$$\frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \int_{V_i} \underbrace{\frac{\partial}{\partial \mathbf{p}_i} f(d(\mathbf{q}, \mathbf{p}_i))}_{}\phi(\mathbf{q}) d\mathbf{q}$$

$$= 2 \int_{V_i} d(\mathbf{p}_i, \mathbf{q}) \underbrace{\mathbf{n}_{\mathbf{p}_i \mathbf{q}}}_{} \phi(\mathbf{q}) d\mathbf{q}$$



Tangent vector at $\mathbf{p_i}$ to the geodesic connecting $\mathbf{p_i}$ and $\mathbf{q}$ (multiplied by a constant that does not depend on $\mathbf{q}$).

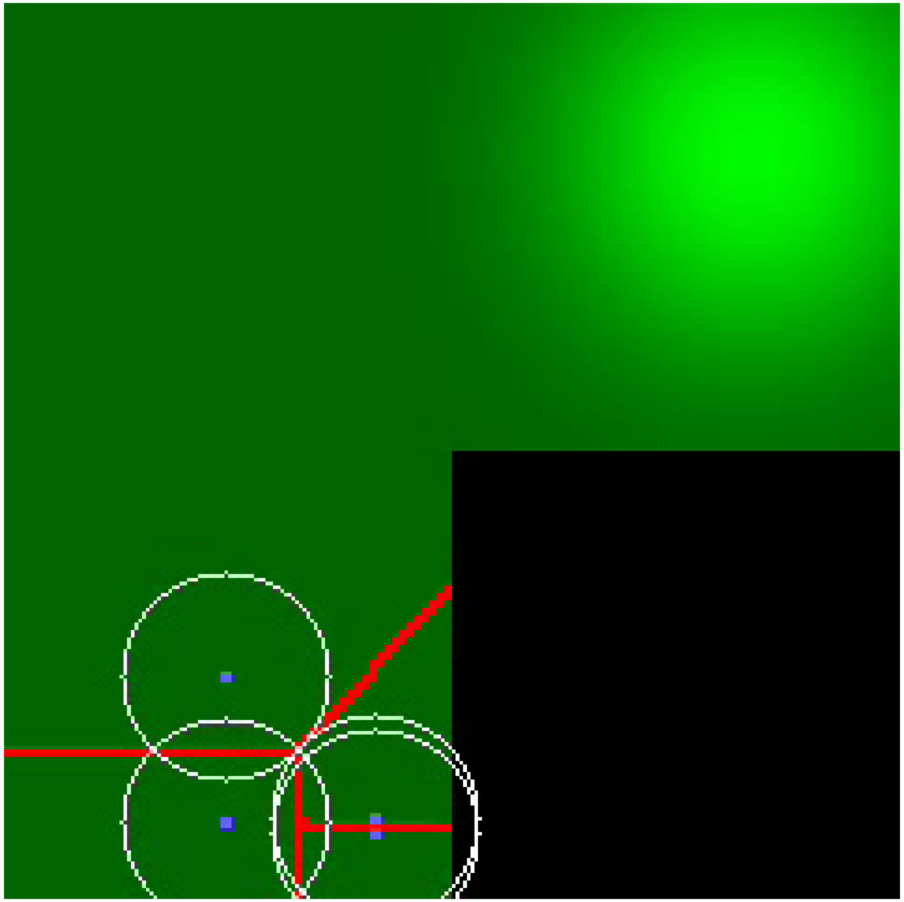
Control law:

$$\mathbf{u}_i = -k \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = 2k \int_{V_i} d(\mathbf{p}_i, \mathbf{q}) \phi(\mathbf{q}) \mathbf{n}_{\mathbf{p}_i, \mathbf{q}} d\mathbf{q}$$

Convenient for the graph-search based discrete approach:

- Geodesics are compued by the Dijkstra's algorithm – gives us $\mathbf{n_{p_i,q}}$.
- The integration can be computed during the tessellation algorithm – **very efficient**!

Example: Lloyd’s algorithm in non-convex environment



Notes:

- The metric (d) is Euclidean.
- Intensity of green is the weight/density function (φ)
- We used “Power Voronoi Tessellation” in order to incorporate finite robot radii.

Part-B

- Use the tools developed for ***distributed coverage and exploration*** of ***unknown or partially known non-convex environments***.

Unknown/Partially known environment

Shannon Entropy assigned to each node, $\mathbf{q}$, in graph:

$$e(\mathbf{q}) = p(\mathbf{q})\ln(p(\mathbf{q})) + (1 - p(\mathbf{q}))\ln(1 - p(\mathbf{q}))$$

→ **a measure of uncertainty/lack of knowledge.**

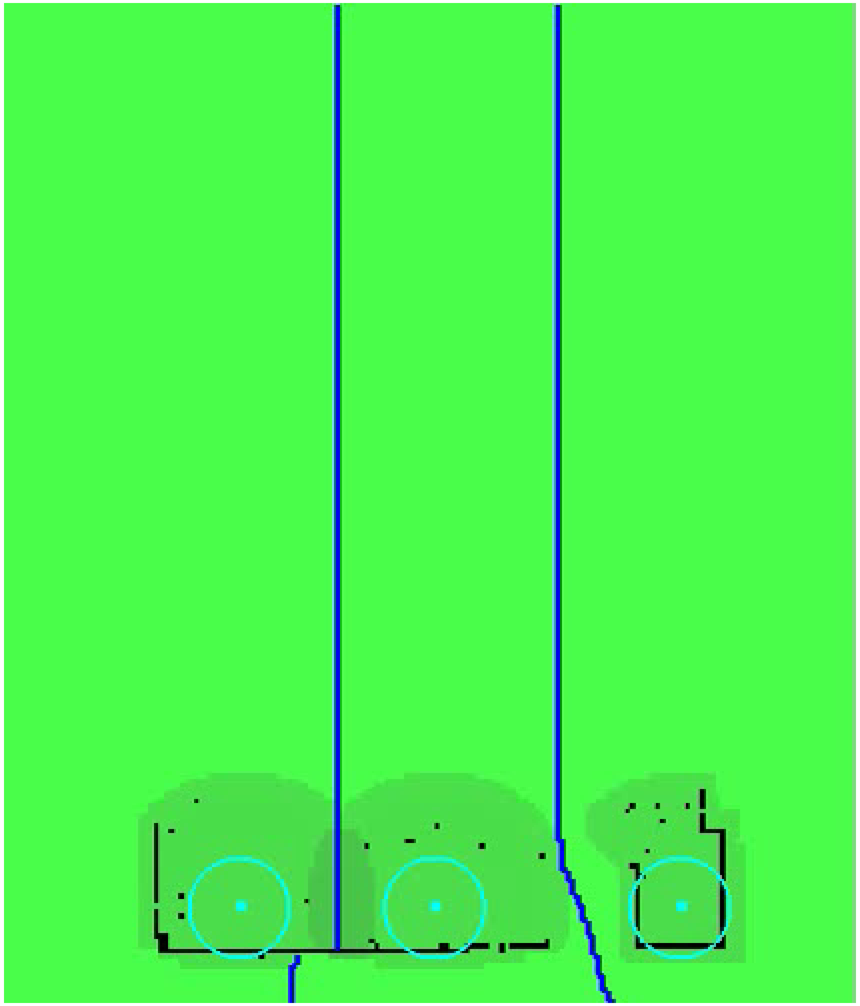
Objective:

- We want to explore an unknown/partially known environment (i.e. reduce entropy)
- Maintain good coverage of the environment during and after exploration.

Overall idea:

- Maintain and update entropy map through sensor data and inter-robot communication.
- ***Control law:*** Use the Voronoi tessellation & Lloyd’s algorithm for non-convex environment and non-Euclidean metric
  - Entropy-weighted metric (non-Euclidean)
  - Entropy as weight function.

Result



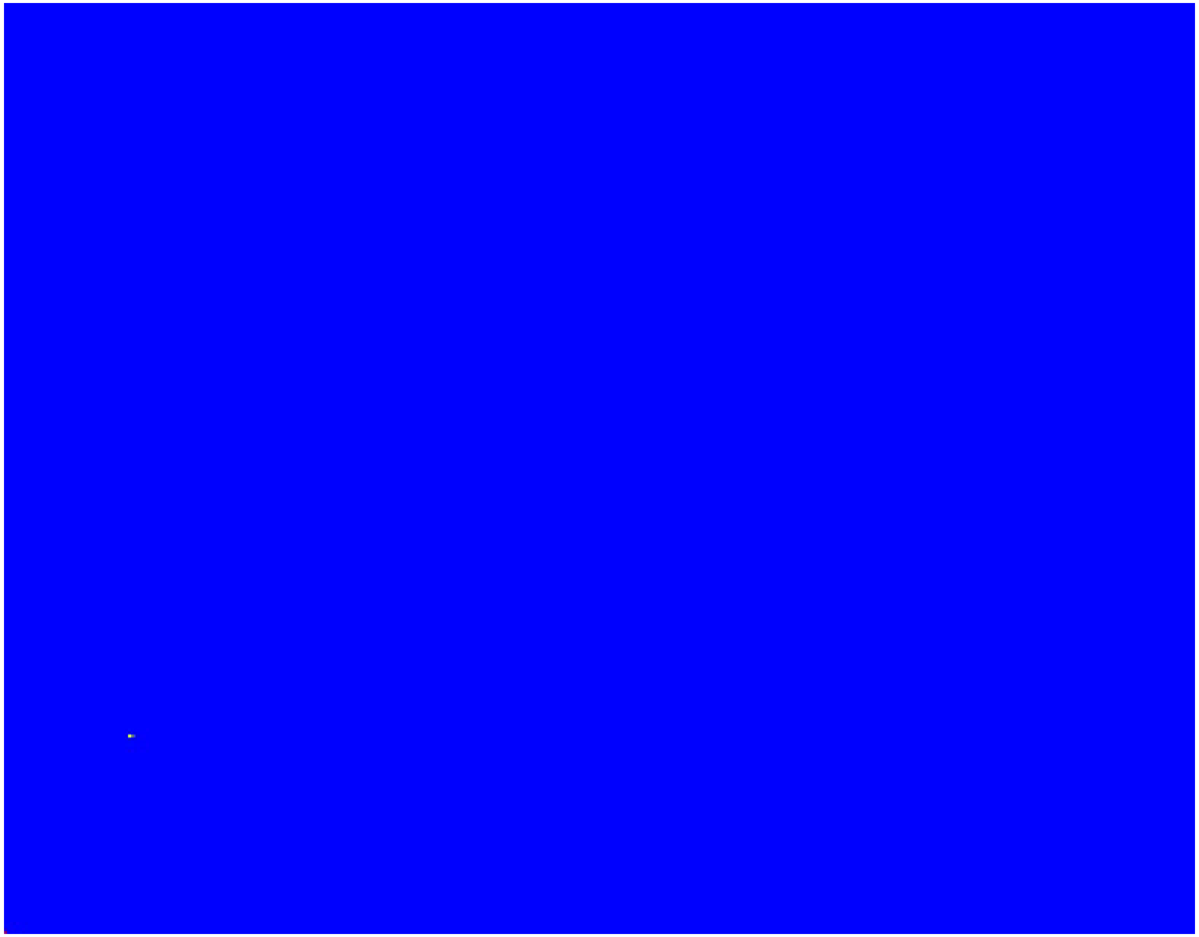
3 robots in a 170 x 200 discretized environment.

Each iteration (time-step) takes about 0.3 s.

(C++ implementation running on a single processor)

Results use a slightly different version of the control law than what described earlier.

ROS implementation – truly distributed (multi-thread) implementation



Realistic simulation of sensors, robot kinematics, communication.

4 robots in a 1000 x 783 uniformly discretized environment.

Parallel (multi-thread) computation wherever possible.

Main thread runs at ~1Hz for each robot (note: all threads run on a single processor).

Overview

1. Planning with Topological constraints – Homotopy & Homology class constraints

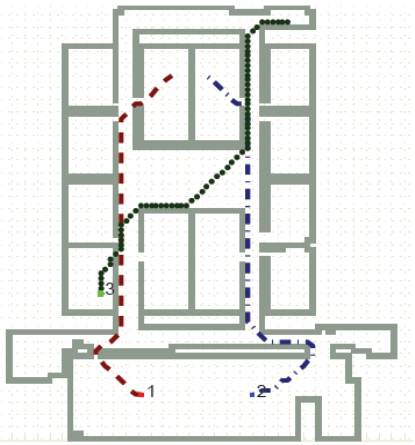
2. Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric

3. Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow

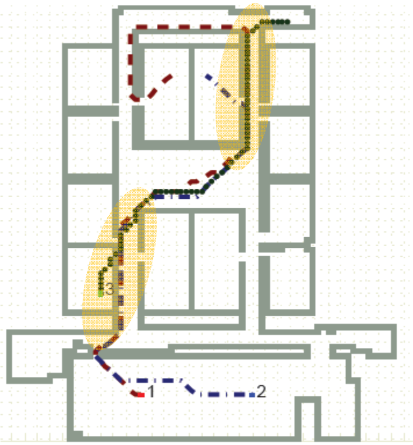
4. Transformation for Efficient Optimal Planning in Environments with Non-Euclidean Metric

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

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

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

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

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

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:

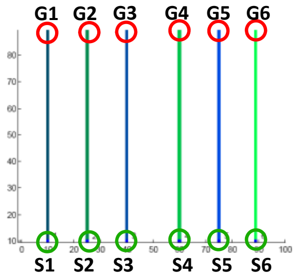
$$\begin{aligned} &\Psi_r(W + \epsilon V, W) - \Psi_r(W, W) \\ &\qquad \leq \Psi_r(W + \epsilon V, W + \epsilon V) - \Psi_r(W, W + \epsilon V) \end{aligned}$$

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

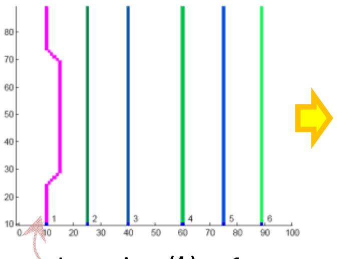
Ascent direction:

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

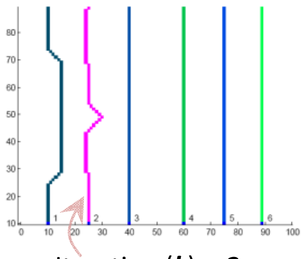
Six robots planning iteratively to satisfy **rendezvous constraints** in an **empty environment**:



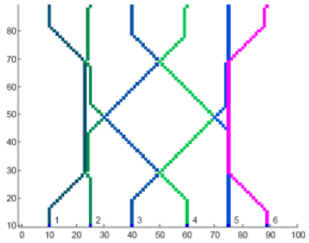
Iteration (*k*) = ***0***



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



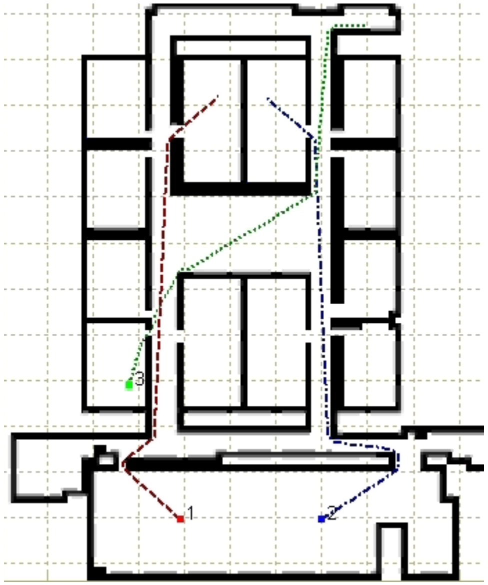
Iteration (*k*) = ***2***
Planning robot (*r*) ***= 2***



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

Definitions

Theorems



1A
1B
2A
2B
3
4

$\mathcal{N}^N = \{1, 2, \cdots, N\}$

$\mathcal{P}^N = \{\{1,2\}, \{1,3\}, \cdots, \{1,N\}, \{2,3\}, \{2,4\}, \cdots, \{N-1, N\}\}$

$\mathcal{P}_r^N = \{\{1,r\}, \cdots, \{r-1, r\}, \{r+1, r\}, \cdots, \{N, r\}\}$

V and W are vectors with $N(N\text{-}1)/2$ elements

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 $r \notin \{i,j\}$

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

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

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

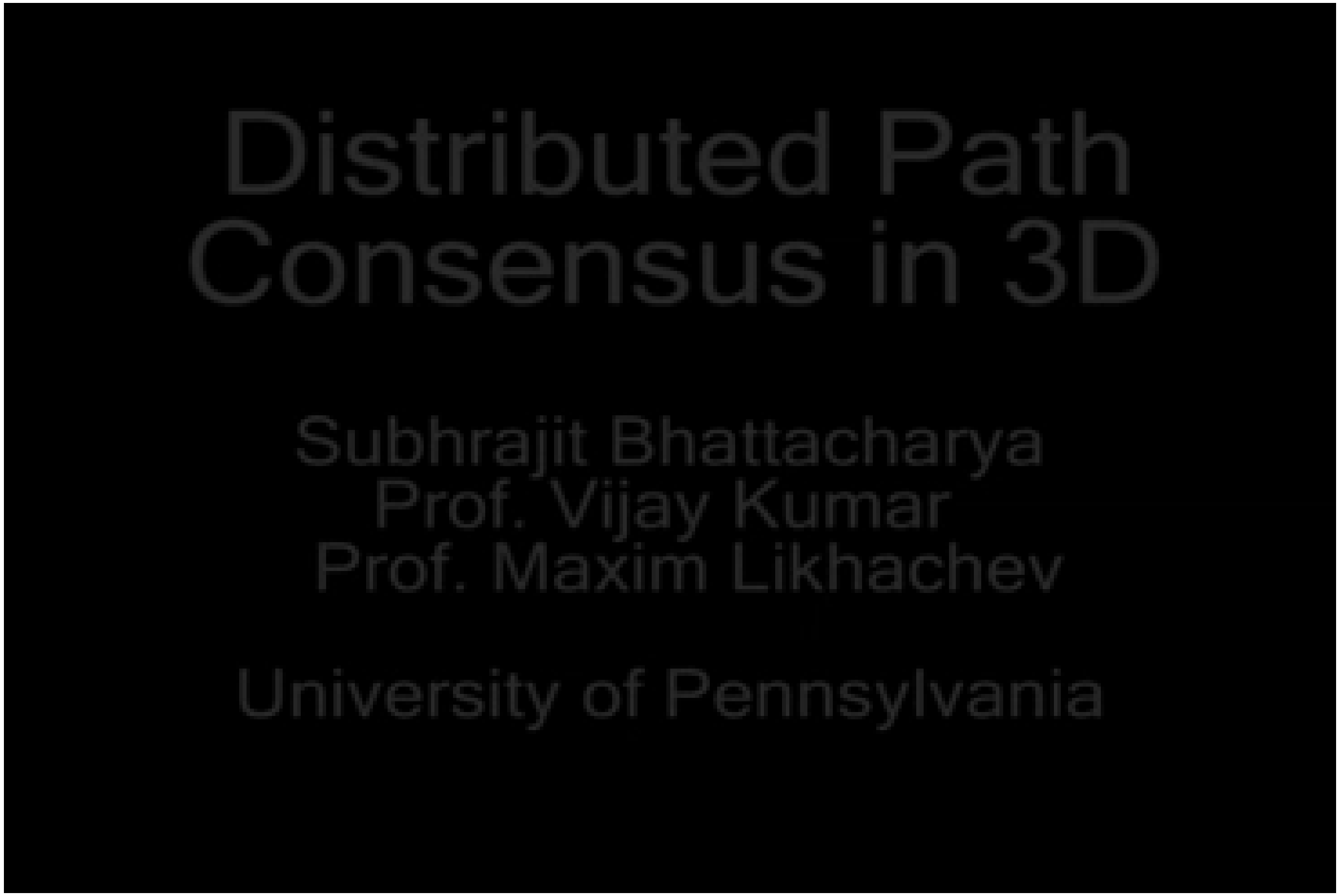
$$\{\pi_1^k, \ldots, \pi_N^k\} = \arg\min_{\{\pi\}} \left[\sum_{i \in \mathcal{N}^N} c(\pi_i) + \sum_{\{ij\} \in \mathcal{P}^N} W_{ij}^k \cdot \Omega_{ij}(\pi_i, \pi_j)\right] .\ i.e.\ \pi_i^k = \overline{\Pi}_i(W^k), \quad \forall i, k$$

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.

Result

(X-Y-Z-Time configuration space)

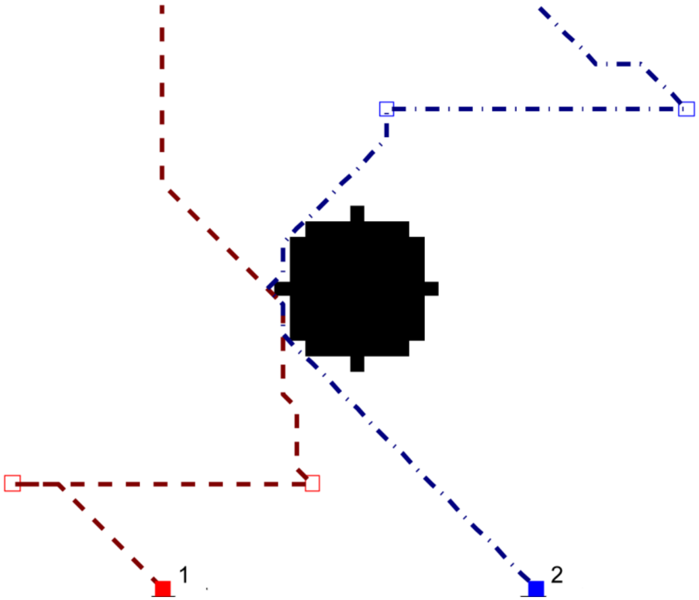


Distributed Path Consensus in 3D

Subhrajit Bhattacharya
Prof. Vijay Kumar
Prof. Maxim Likhachev

University of Pennsylvania

Additional complexity – Introducing ***Tasks***

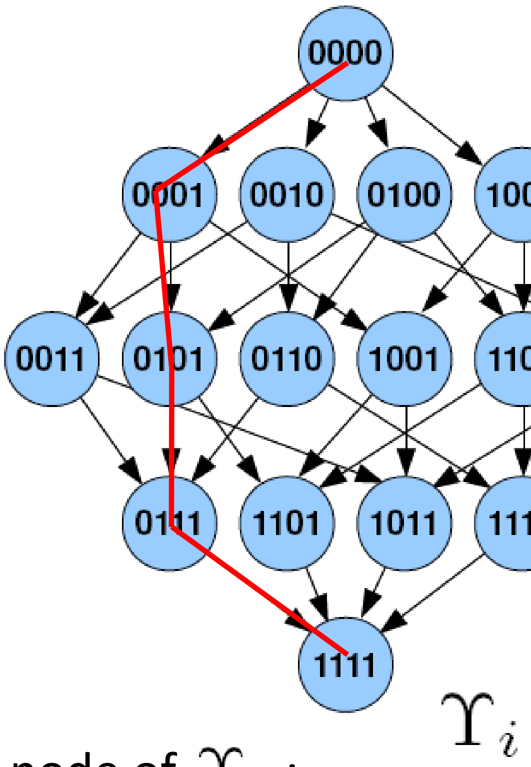


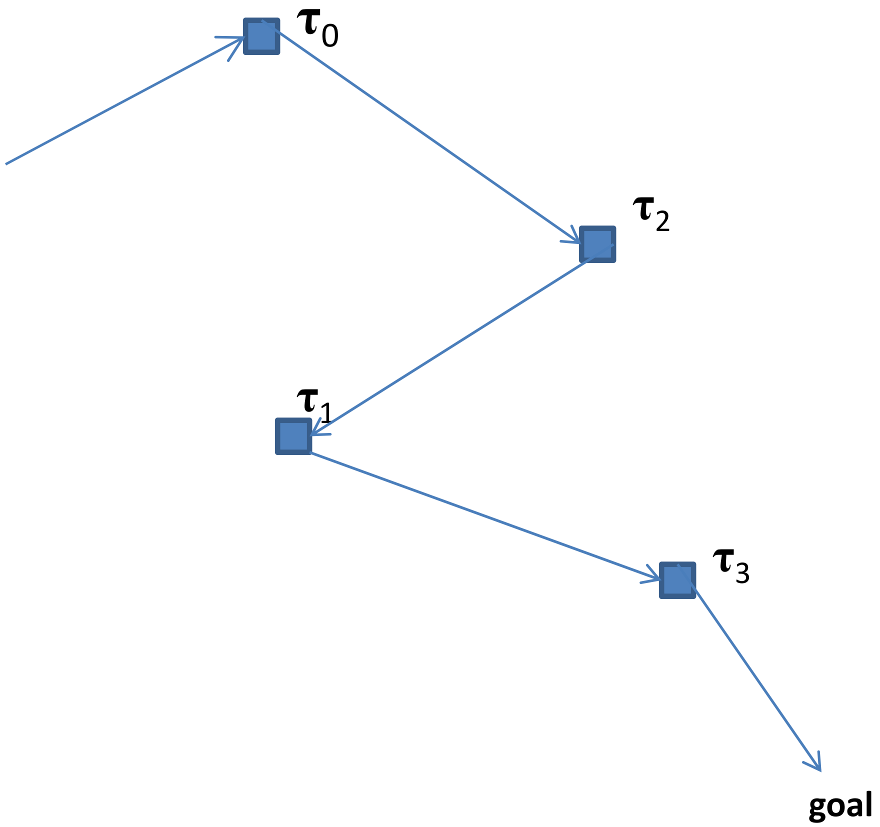
Goal-directed navigation with tasks – unconstrained

Need to exchange information midway and execute tasks

- Each robot is given an unordered list of tasks (coordinates in space).
- Determining the optimal order of execution of the tasks is often nontrivial . . .
- . . . especially in presence of obstacles and constraints

The Task Graph



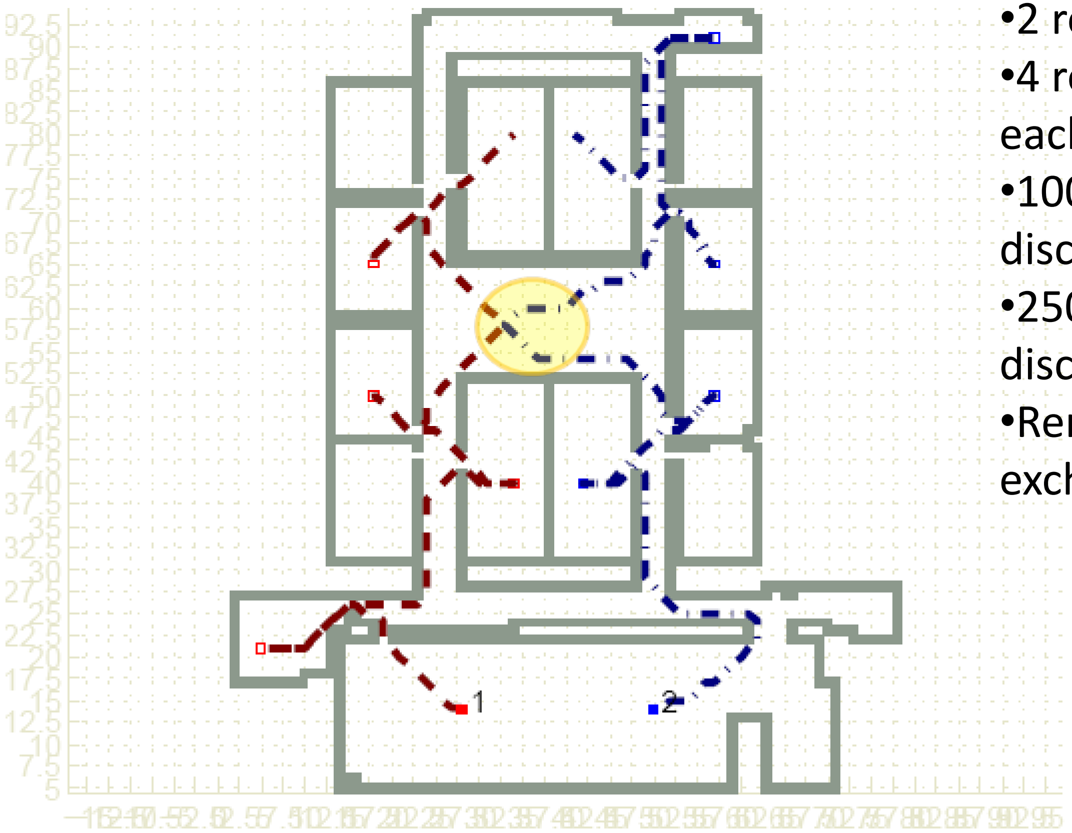


A node of Υ_i :

$\beta = B_M(\{j_1, j_2, \cdots, j_k\})$ is a binary sequence of *M* bits, with 1's at the positions $j_1, j_2, \cdots, j_k$

Take ***product of this graph*** with the graph G formed by discretization of environment to obtain ***final state graph***.

Results



- 2 robots
- 4 rooms to explore for each
- 100x100 spacial discretization
- 250 temporal discretization
- Rendezvous to exchange information

Joint state space: $250 \times (100 \times 100 \times 2^4)^2 = 6.4 \times 10^{12} = 6.4 \text{ trillion states}$

Individual state space: $250 \times 100 \times 100 \times 2^4 = 4 \times 10^7$ = 40 million states

Planning time: $1311 \text{ seconds in } 17 \text{ iterations}$ (single processor implementation)

Overview

1. Planning with Topological constraints – Homotopy & Homology class constraints
2. Incorporating Metric Information using search-based techniques – Voronoi Tessellation in Non-convex Environment with Non-uniform metric
3. Dimensional Decomposition – Distributed Optimization using Separable Optimal Flow
4. Transformation for Efficient Optimal Planning in Environments with Non-Euclidean Metric

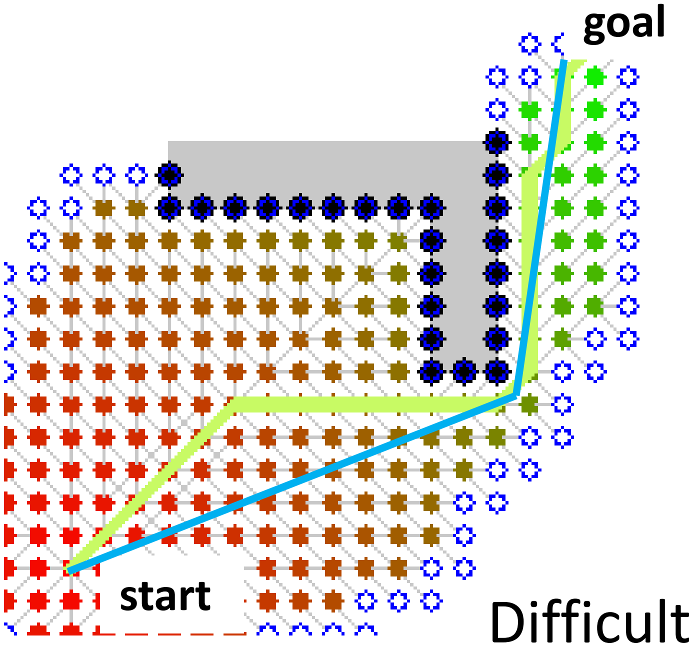
Motivation

Graph-search techniques are well-suited for:

- Non-convex environments
- Non-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 Euclidean***, we can use ***visibility-based approaches***:

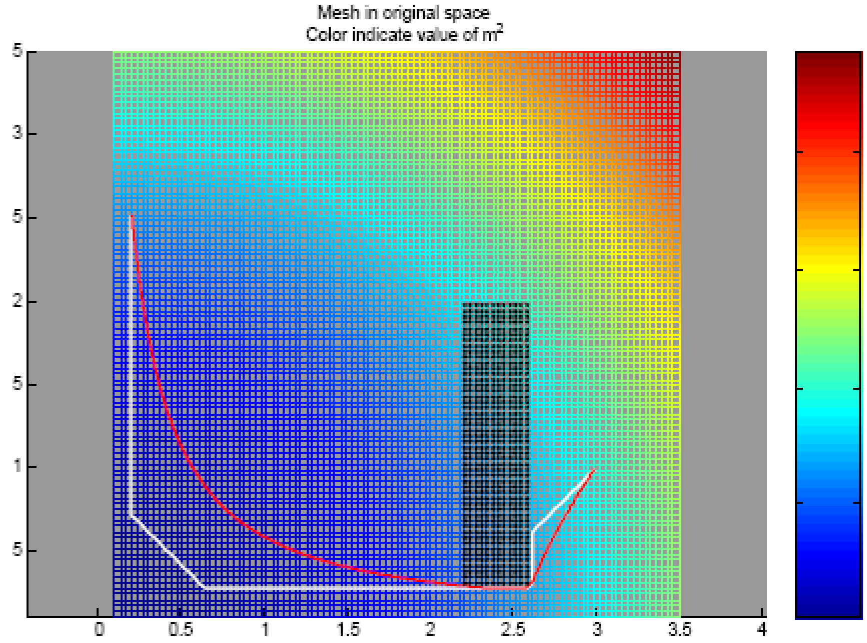
- Do post-processing
- Employ visibility graph
- Use theta-star algorithm *[Nash, et al.]*
- Etc.

Difficult for non-Euclidean metric – not efficient to compute geodesic connecting 2 arbitrary points.

Question:

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

Motivating example: A flat space (zero curvature)



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

Isotropic, but non-uniform metric

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

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

Isometric embedding of the metric space in Euclidean plane

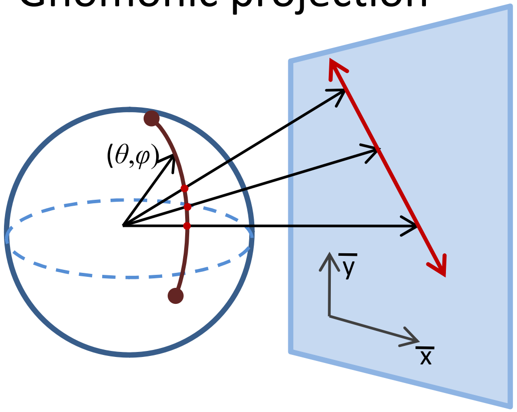
This can be written as $|z|^2$. This is hence a conformal map. This is the ***same metric spaces with zero scalar curvature**, being described by 2 different coordinate charts.*

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***)?

Non-isometric embedding into Euclidean space with geodesics mapping to “straight lines”

The sphere (admits constant **positive curvature** metric)

Gnomonic projection



$$(\theta,\varphi) \longrightarrow (\bar{x},\bar{y})$$

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

(spherical metric)

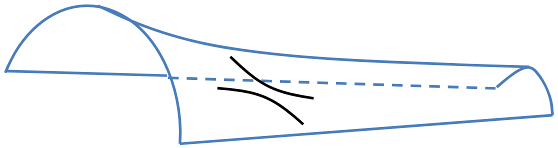
A coordinate chart maps geodesics in the metric space to “straight lines” on the plane (i.e. geodesics using the usual Euclidean metric on the plane), but ***non-isometrically***!

-- **Orthogeodesic embedding**

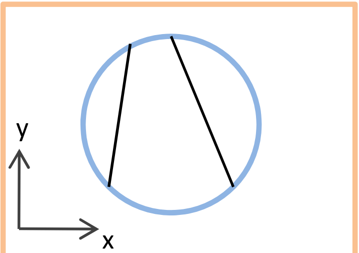
Hyperbolic Space (admits constant **negative curvature** metric)

Beltrami–Klein model:

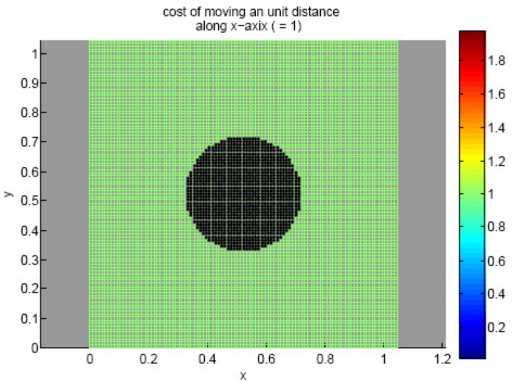
- The whole hyperbolic plane is mapped to the interior of a circle
- Geodesics on hyperbolic plane maps to straight lines.



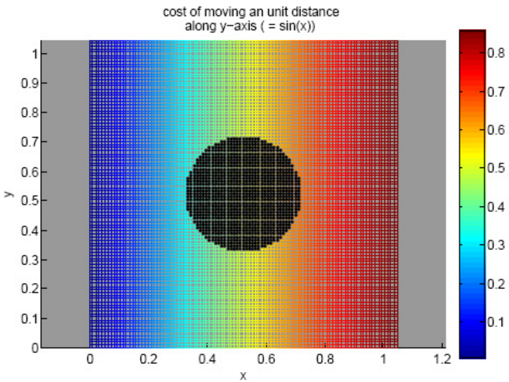
$\longrightarrow$



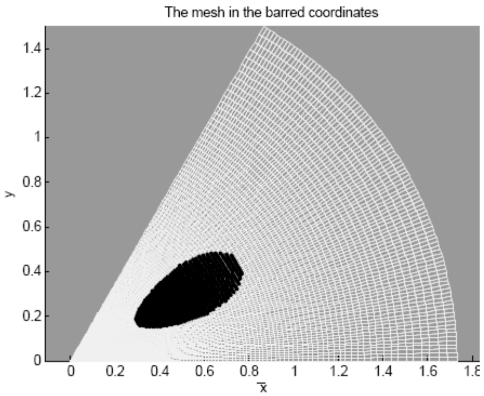
Result with Gnomonic projection



Cost of moving an unit distance along x-axis (=1).

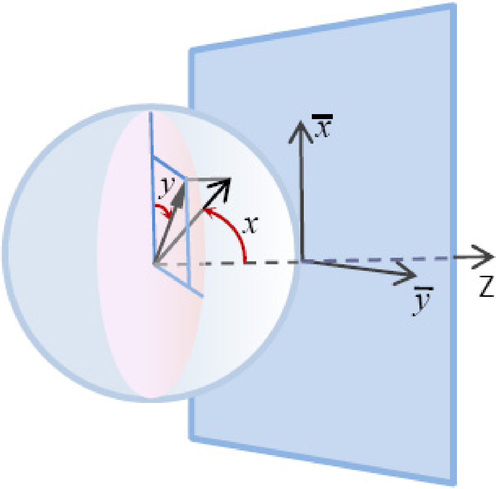


Cost of moving an unit distance along y-axis (=sin(x)).

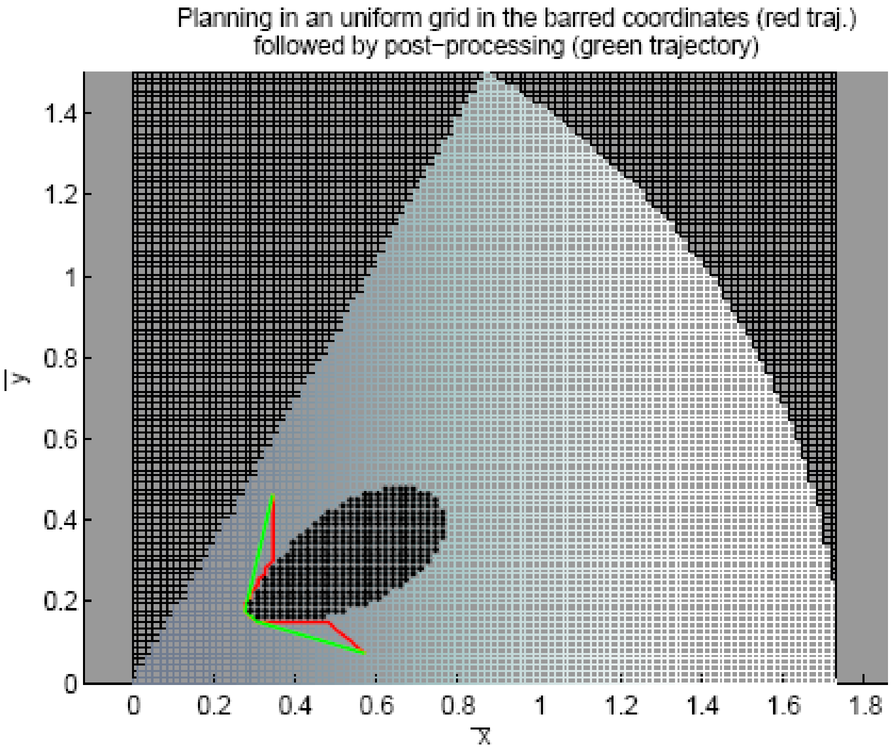


$$g = \begin{bmatrix} 1 & 0 \\ 0 & \sin^2(x) \end{bmatrix} \xrightarrow{T} \bar{g} = \begin{bmatrix} \dfrac{1+\bar{y}^2}{1+\bar{x}^2+\bar{y}^2} & \dfrac{-\bar{x}\ \bar{y}}{1+\bar{x}^2+\bar{y}^2} \\ \dfrac{-\bar{x}\ \bar{y}}{1+\bar{x}^2+\bar{y}^2} & \dfrac{1+\bar{x}^2}{1+\bar{x}^2+\bar{y}^2} \end{bmatrix}$$

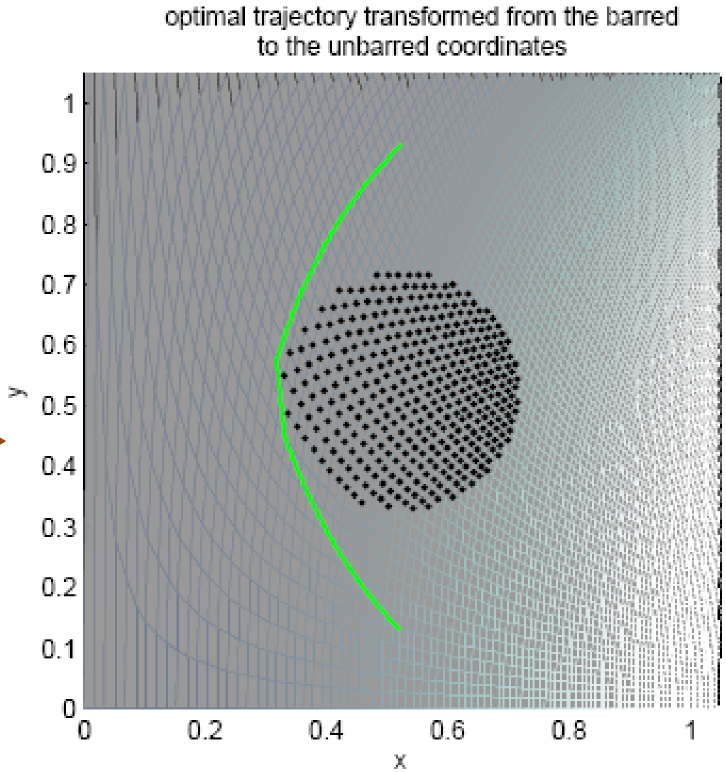
$$\begin{aligned} \bar{x} &= \tan(x)\cos(y) \\ \bar{y} &= \tan(x)\sin(y) \end{aligned}$$



Result with Gnomonic projection

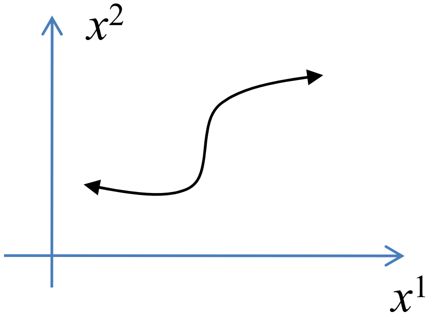

Planning in an uniform grid in the barred coordinates (red traj.) followed by post−processing (green trajectory)

T^{-1}


optimal trajectory transformed from the barred to the unbarred coordinates

Condition for Orthogeodesic embedding

Suppose we are given a coordinate chart consisting of coordinate variables $\mathbf{x} = (x^1, x^2, \cdots, x^N)$ and a matrix representation of the metric, **g**



Geodesic equation: $$\frac{\mathrm{d}^2 x^i}{\mathrm{d}t^2} + \Gamma^i_{jk} \frac{\mathrm{d}x^j}{\mathrm{d}t} \frac{\mathrm{d}x^k}{\mathrm{d}t} = 0$$

Condition for geodesics being “straight lines”: $$\left[\frac{\mathrm{d}^2 x^1}{\mathrm{d}t^2}, \frac{\mathrm{d}^2 x^2}{\mathrm{d}t^2}, \cdots \right] = \Theta(\mathbf{x}, \dot{\mathbf{x}}) \left[\frac{\mathrm{d}x^1}{\mathrm{d}t}, \frac{\mathrm{d}x^2}{\mathrm{d}t}, \cdots \right]$$

(acceleration parallel to velocity)

Reduces to,

$$\Gamma^i_{jk}(\mathbf{x}) \;\; e^j \; e^k \;\; = \;\; \theta(\mathbf{x}, \mathbf{e}) \; e^i \quad \text{for every} \quad \mathbf{x} \in \mathbb{R}^N, \;\; [e^1, e^2, \cdots] \in \mathbb{R}^N$$

Final condition:

$$g_{pj,k} + g_{pk,j} - g_{jk,p} \;\; = \;\; \theta_j g_{kp} + \theta_k g_{jp}, \qquad \forall p, j, k$$

Eliminate the N unknown θ’s from the $N^2(N+1)/2$ equations to obtain the final conditions.

Future direction: Existence? How to find a orthogeodesic coordinate chart?

Major Future Directions

- Extend the ideas of planning with topological constraints to non-Euclidean spaces with punctures (e.g. robotic arms).
- To combine the problems of planning with topological constraints and the problem of coverage/exploration in non-convex environments with non-Euclidean metric.
- Study the conditions under which an orthogeodesic embedding may exist, and how to find one, given a metric.

Acknowledgements

- My advisors, Dr. Vijay Kumar and Dr. Maxim Likhachev.
- Other members of my thesis committee, Dr. George J. Pappas, Dr. Daniel E. Koditschek and Dr. Robert Ghrist.
- Dr. D. Lipsky, Dr. R. Ghrist (Collaborators on work involving algebraic topology)
- Dr. N. Michael and Dr. L. Pimenta (Collaborators on work involving coverage and exploration)
- Colleagues, faculty and staff of MEAM and GRASP.
- Support from ONR, NSF, ARO, ARL

Most of the code available at

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

Thank you!

Questions?