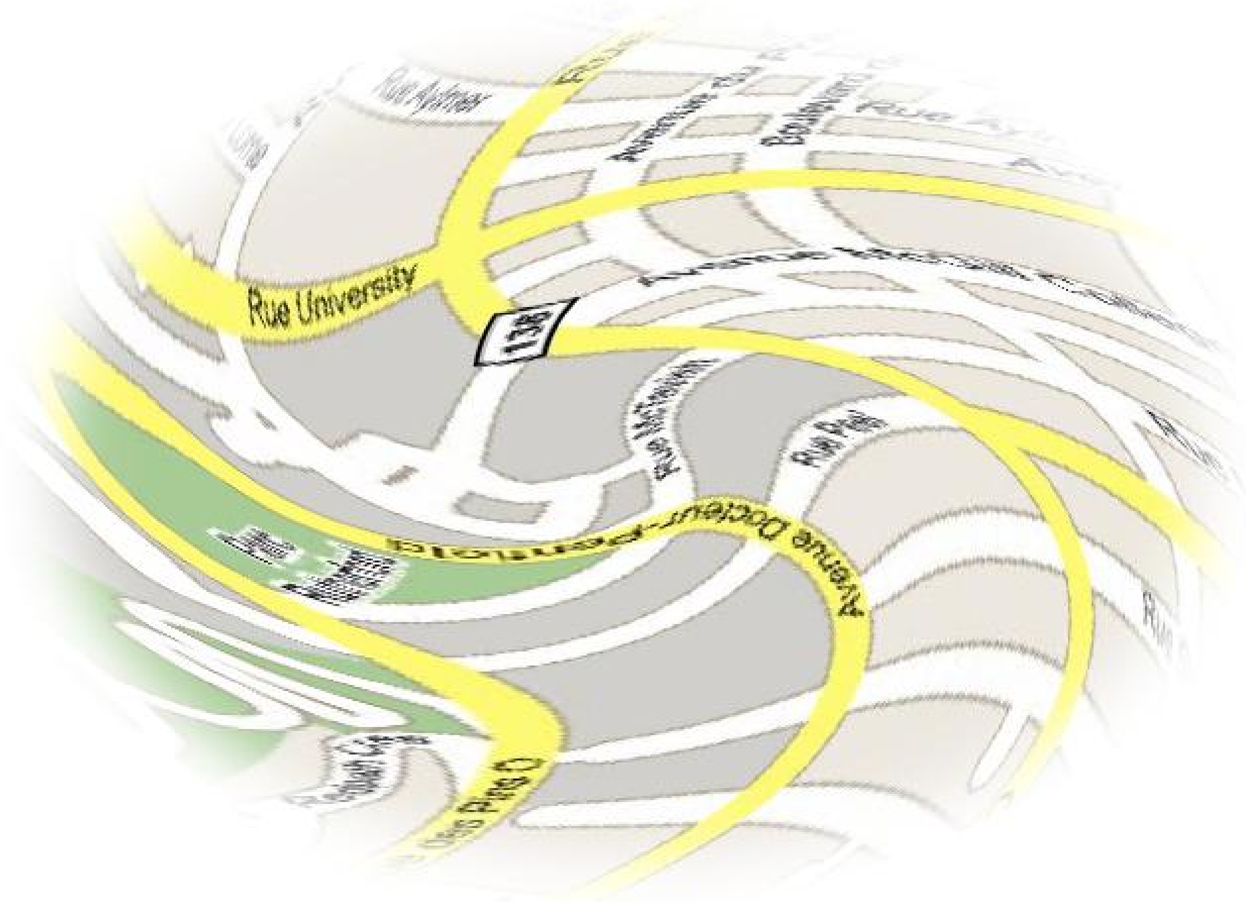
Subjective Mapping



Michael Bowling
University of Alberta

April 22, 2005

Acknowledgments

- Joint work with...

  Ali Ghodsi and Dana Wilkinson.

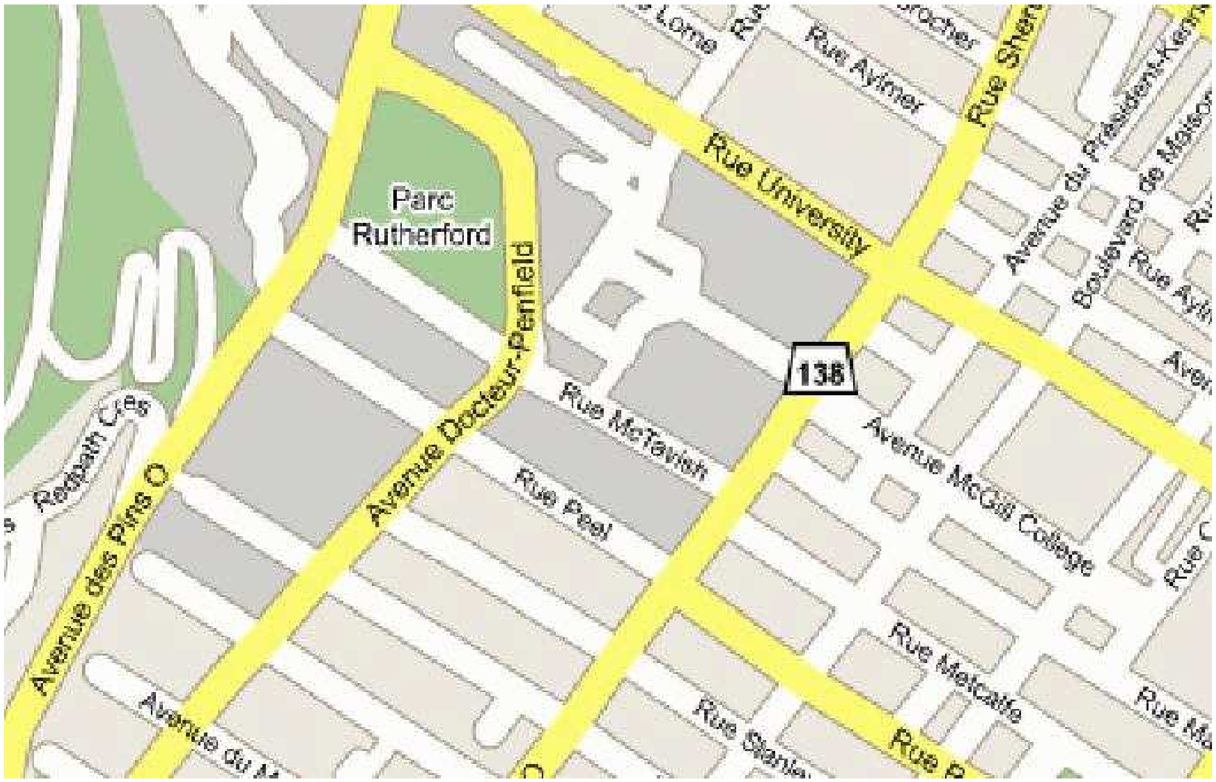
- Localization work with...

  Adam Milstein and Wesley Loh.

- Discussions and insight...

  Finnegan Southey, Dale Schuurmans, Tao Wang, Dan Lizotte, Michael Littman, and Pascal Poupart.

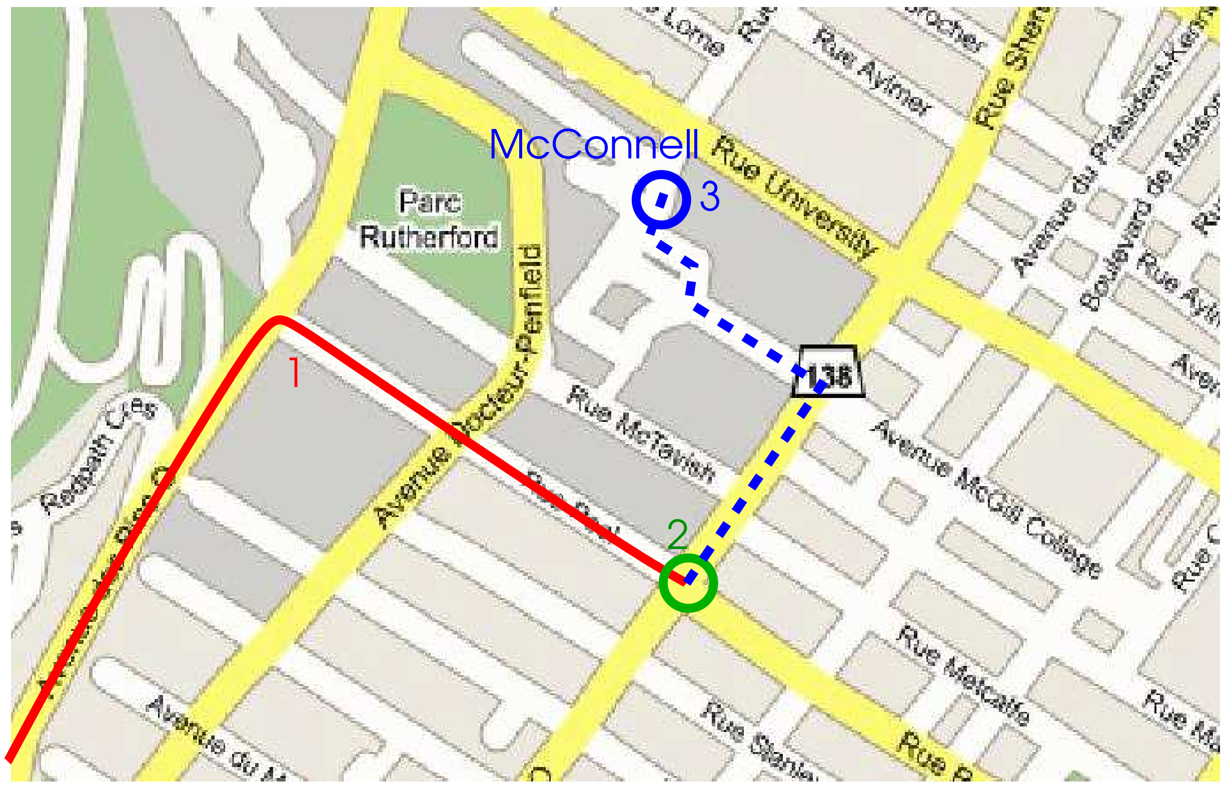
What is a Map?


Parc
Rutherford
Rue University
Rue Aylmer
Avenue Docteur-Penfield
Rue McTavish
Rue Peel
Avenue des Pins O
Redpath Cres
138
Avenue McGill College
Rue Metcalfe

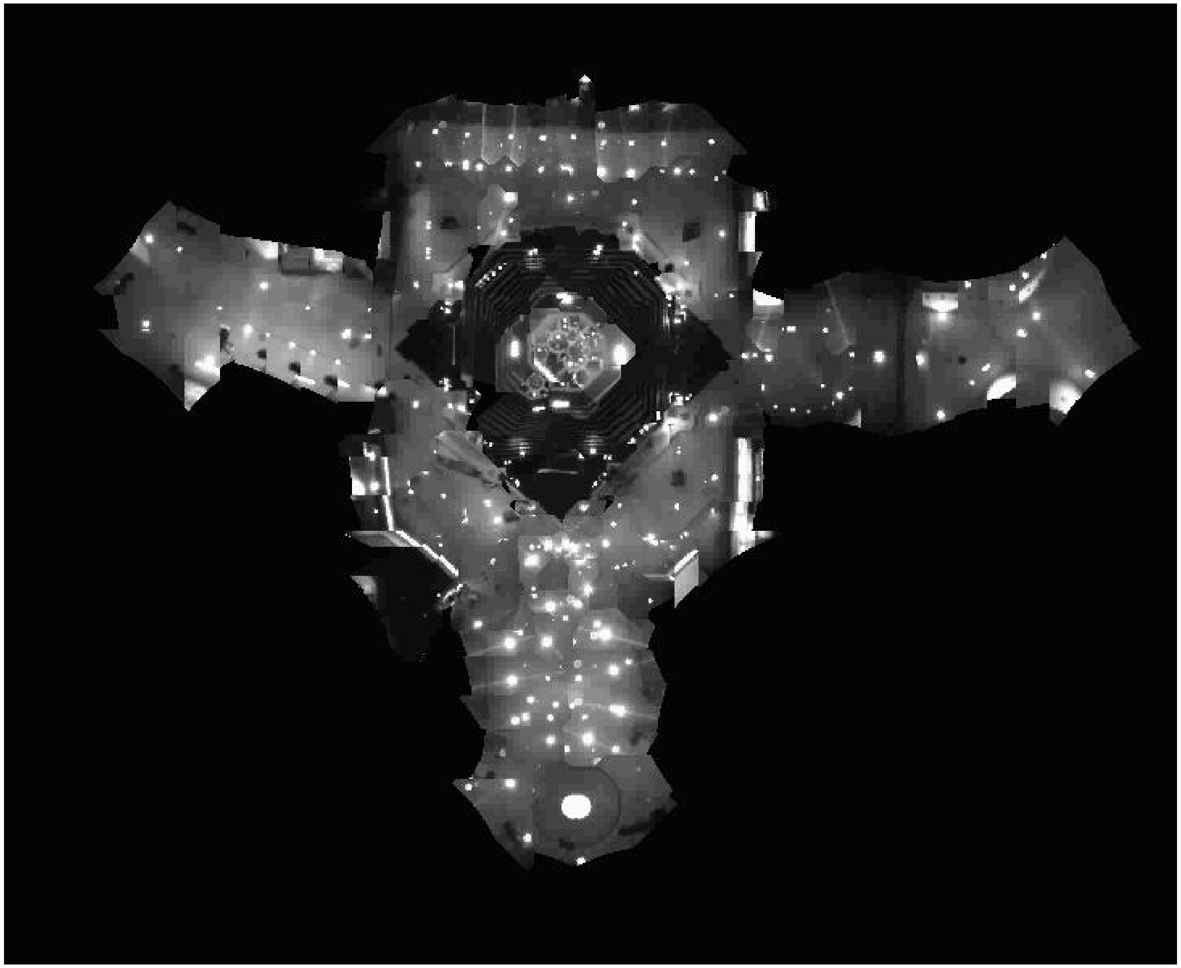
What is a Map?

Essentially maps answer three questions. . .

1. Where have I been?
2. Where am I now?
3. How do I get where I want to go?



Robot Maps



*From Sebastian Thrun's web page.

Robot Maps

- Maps are Models.
  - Motion: $P(x_{t+1}|x_t, u_t)$.
  - Sensor: $P(z_t|x_t)$.

x_t : pose
u_t : action or control
z_t : observation

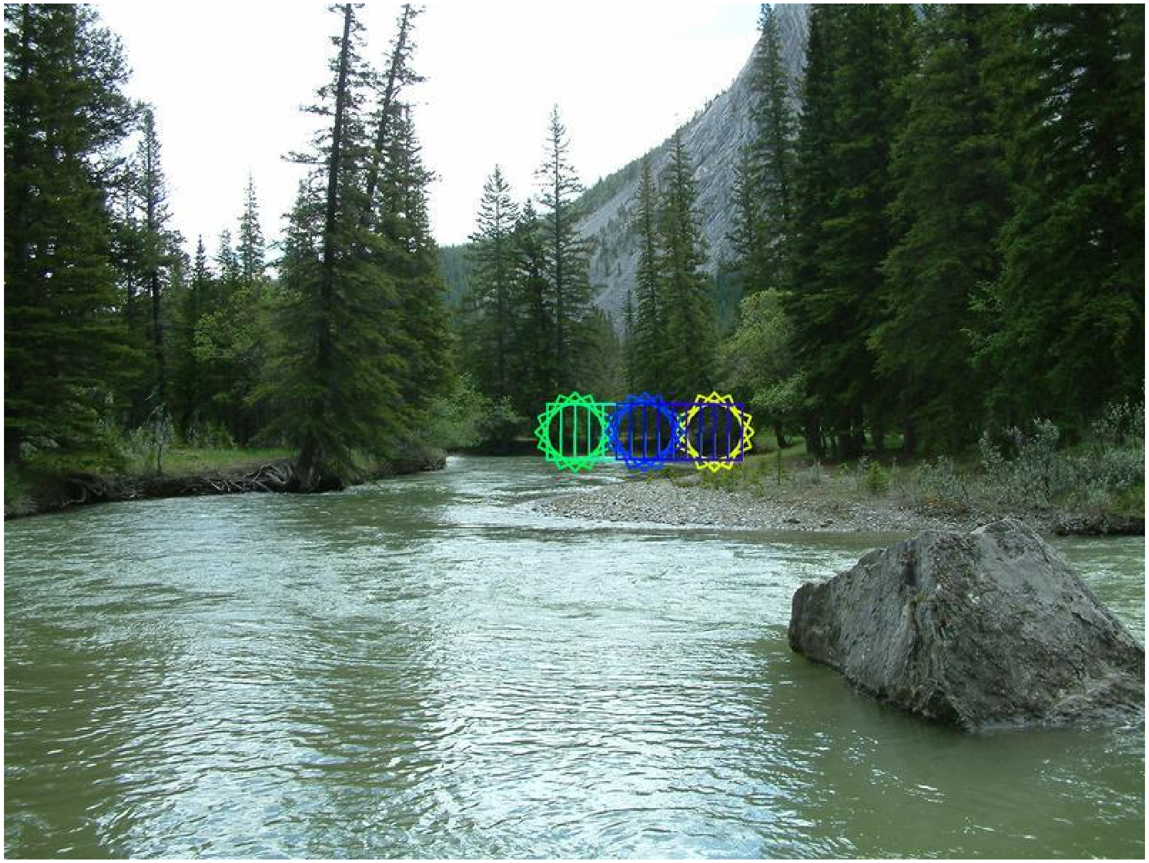
- Robot Pose: $x_t = (\mathsf{x}_t, \mathsf{y}_t, \theta_t)$.
  - Objective representation.
  - Models relate actions (u_t) and observations (z_t) to this frame of reference.

- Can a map be learned with only subjective experience?

$z_1, u_1, z_2, u_2, \ldots, u_{T-1}, z_T$ Not an objective map.

ImageBot

- A “Robot” moving around on a large image.

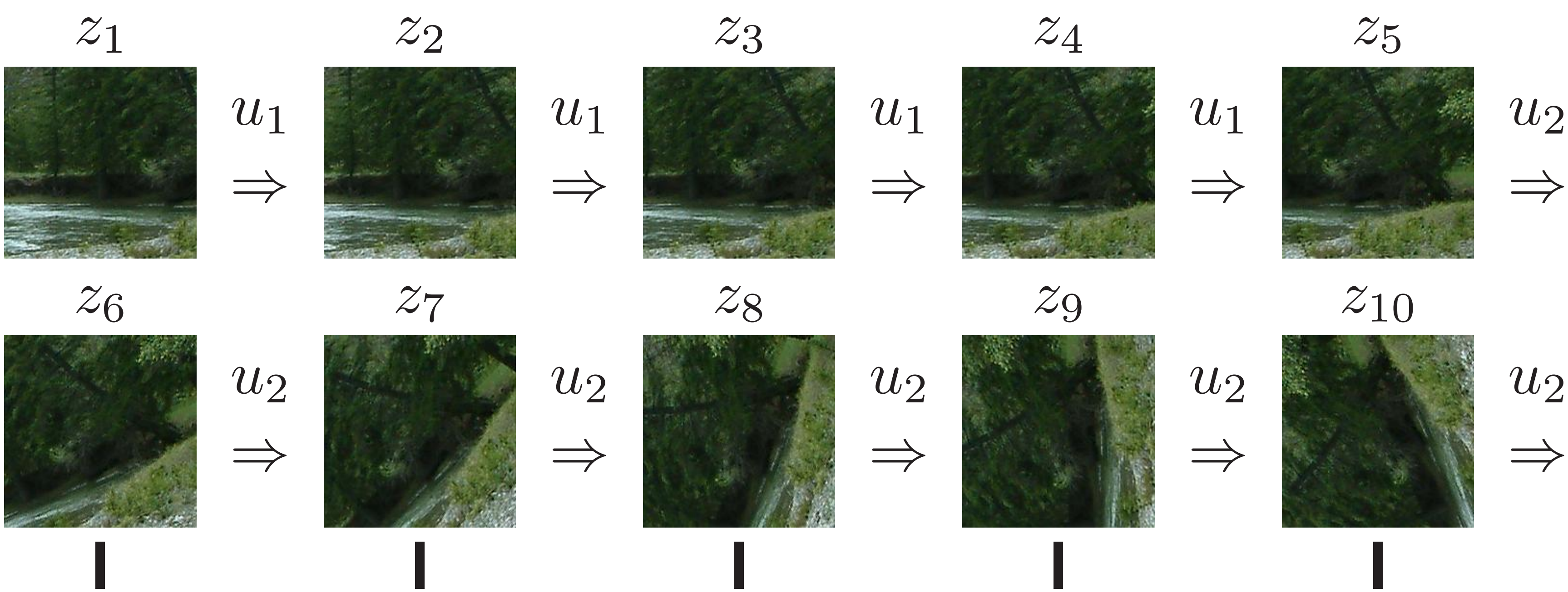


  - Forward
  - Backward
  - Left
  - Right
  - Turn-CW
  - Turn-CCW
  - Zoom-In
  - Zoom-Out

- Example: F $\times$ 5, CW $\times$ 8, F $\times$ 5, CW $\times$ 16, F $\times$ 5, CW $\times$ 8

ImageBot

- Construct a map from a stream of input.



- Actions are labels with no semantics.
- No image features, just high-dimensional vectors.
- Can a map be learned with only subjective experience?

Overview

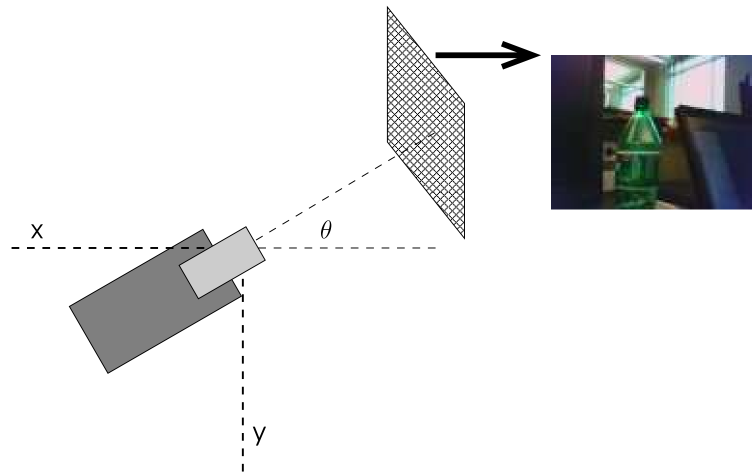
- What is a Map?
- Subjective Mapping
- Action Respecting Embedding (ARE)
- Results and Applications
- Future Work

Subjective Maps

- What is a subjective map?
  - Answers the map questions.
    1. Where have I been?
    2. Where am I now?
    3. How do I get where I want to go?
  - No models.
  - Representation (i.e., pose) can be anything.
- Subjective mapping becomes choosing a representation.
  - What is a good representation?
  - How do we extract it from experience?
  - How do we answer our map questions with it?

Subjective Representation

- $(\mathsf{x}, \mathsf{y}, \theta)$ is often a good representation. Why?
  - Sufficient representation for generating observations.



  - Low dimensional (despite high dimensional observations).
  - Actions are simple transformations.

$$x_{t+1} = (\mathsf{x}_t + F\cos\theta_t, \quad \mathsf{y}_t + F\sin\theta_t, \quad \theta_t + R)$$

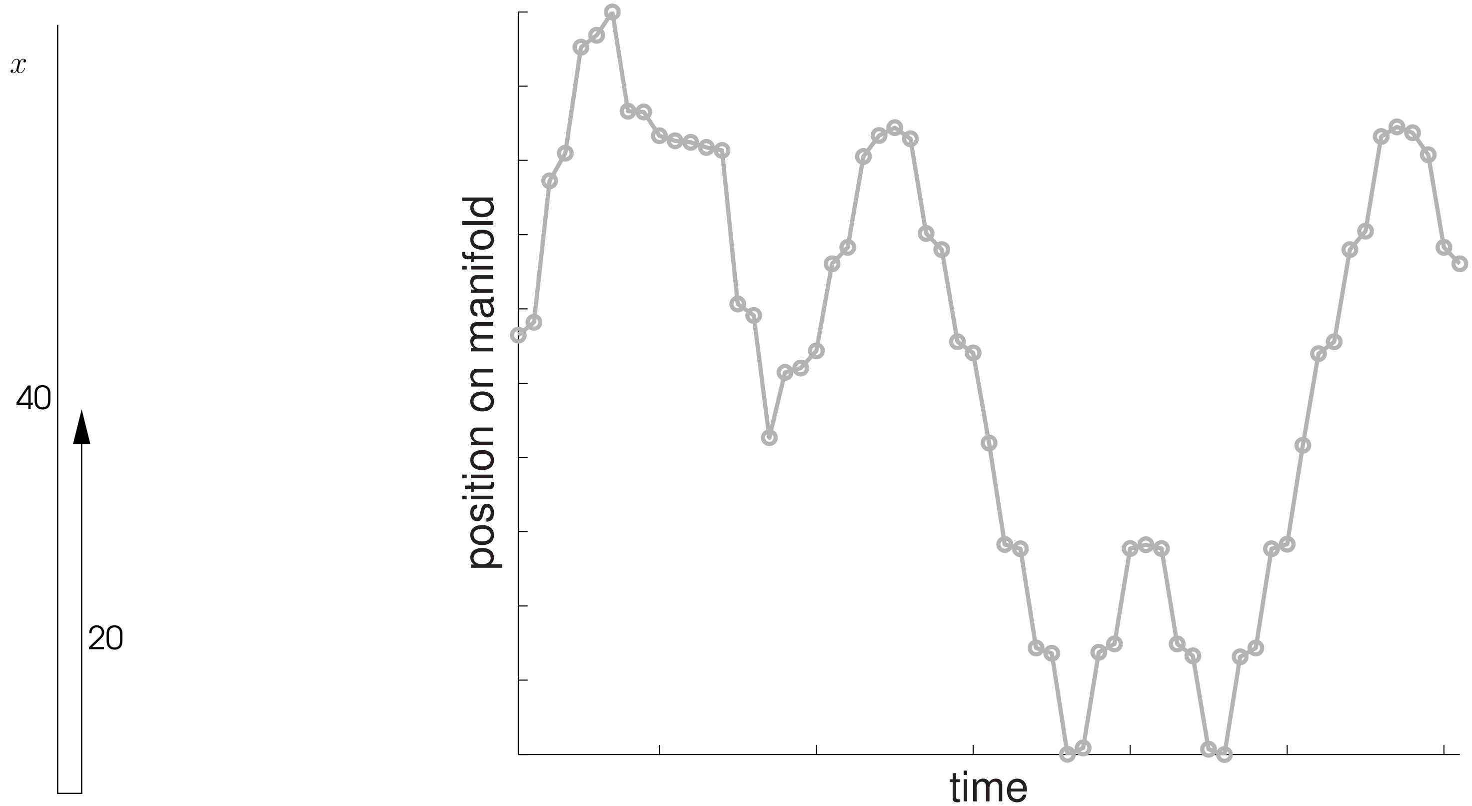
Subjective Representation

- How do we extract a representation like $(\mathrm{x}, \mathrm{y}, \theta)$?
  - Low dimensional description of observations?

Dimensionality Reduction

  - Respects actions as simple transformations?

Dimensionality Reduction Maps



Subjective Representation

- How do we extract a representation like $(\mathrm{x}, \mathrm{y}, \theta)$?
  - Low dimensional description of observations?

Dimensionality Reduction

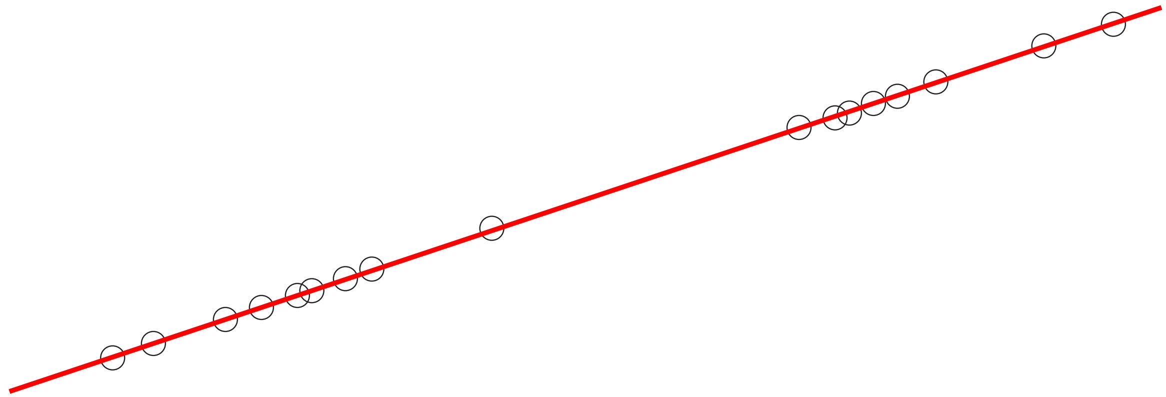
  - Respects actions as simple transformations?

Action Respecting Embedding

Overview

- What is a Map?
- Subjective Mapping
- Action Respecting Embedding (ARE)
  - Dimensionality Reduction 101
  - Non-Uniform Neighborhoods
  - Action Respecting Constraints
- Results and Applications
- Future Work

Dimensionality Reduction 101

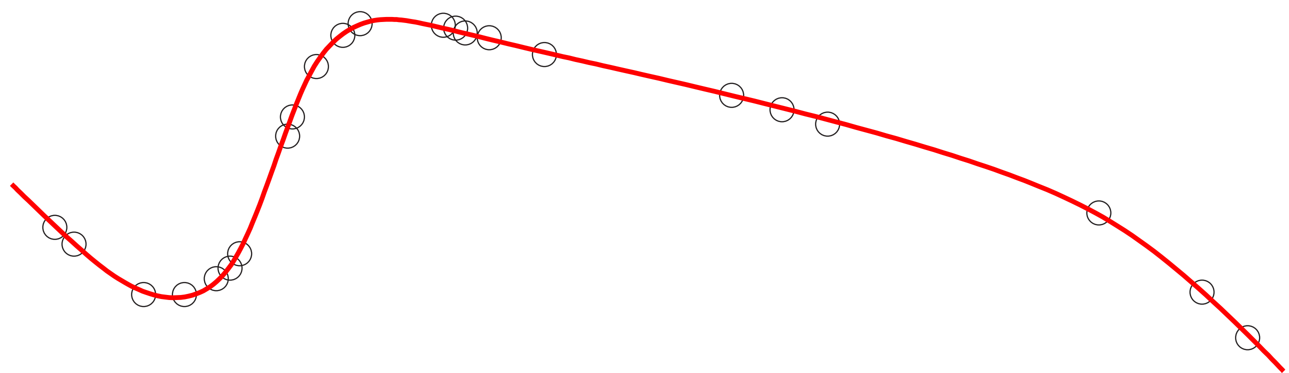


- Principal Components Analysis (PCA)

$$X = \begin{bmatrix} | & | & \cdots & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & \cdots & | \end{bmatrix}$$

  - Top d eigenvectors of $X^T X$ form subspace basis.
  - $X^T X$ is the inner product matrix of X (a.k.a. kernel).

Dimensionality Reduction 101



- Non-linear mapping followed by PCA.
  - Let Φ be a non-linear mapping, $\mathcal{X} \to \mathcal{Y}$.
  - Top d eigenvectors of $\Phi(X)^T \Phi(X)$ (a.k.a. the kernel).
  - Φ is expected to be very high dimensional.

Dimensionality Reduction 101

- Kernel PCA
  - Compute $K = \Phi(X)^T \Phi(X)$ directly.
    - K is $n \times n$ regardless of the dimensionality of $\mathcal{Y}$.
    - $K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle = k(x_i, x_j)$.
  - $Y = \text{diag}(\sigma) V^T$, where
    - σ is the square root of the top d eigenvalues of K.
    - V is the top d eigenvectors of K.
    - Y is $d \times n$: the low-dimensional embedding of X.

Semidefinite Embedding (SDE)

(Weinberger & Saul, 2004)

- Goal: Learn the kernel matrix, K, from the data.

- Optimization Problem:

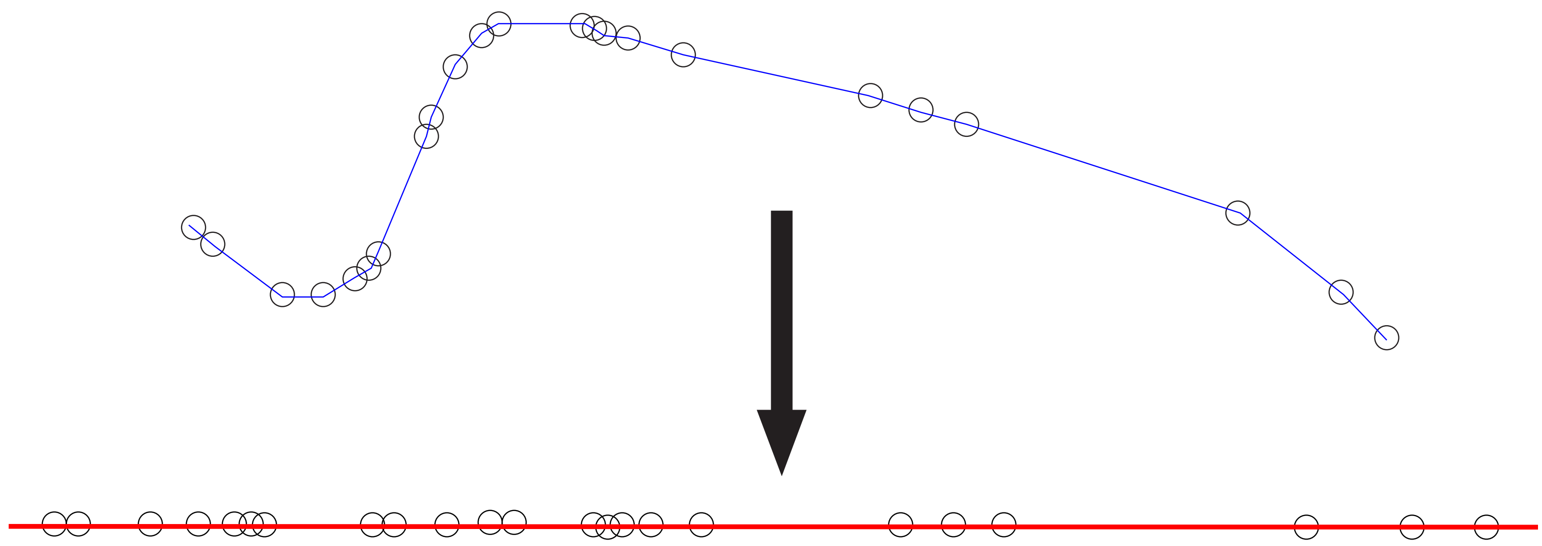
  Maximize: $\mathrm{Tr}(K)$

  Subject to:
  $$K \succeq 0$$
  $$\sum_{ij} K_{ij} = 0$$
  $$\forall i,j \quad \eta_{ij} > 0 \vee [\eta^T \eta]_{ij} > 0 \Rightarrow$$
  $$K_{ii} - 2K_{ij} + K_{jj} = ||x_i - x_j||^2$$

  where η comes from k-nearest neighbors.

- Use learned K with kernel PCA.

Semidefinite Embedding (SDE)



- It works.
- Reduces dimensionality reduction to a constrained optimization problem.

Subjective Mapping

- Given a stream of data,

$$z_1, u_1, z_2, u_2, \ldots, u_{T-1}, z_T,$$

  construct a good subjective representation $(x_1, \ldots, x_T)$?

  - Low dimensional description of observations.
  - Actions are simple transformations.

- SDE on z_i will find a low dimensional representation.

  - Does not make use of action labels.
  - Resulting representations (already seen) are poor.

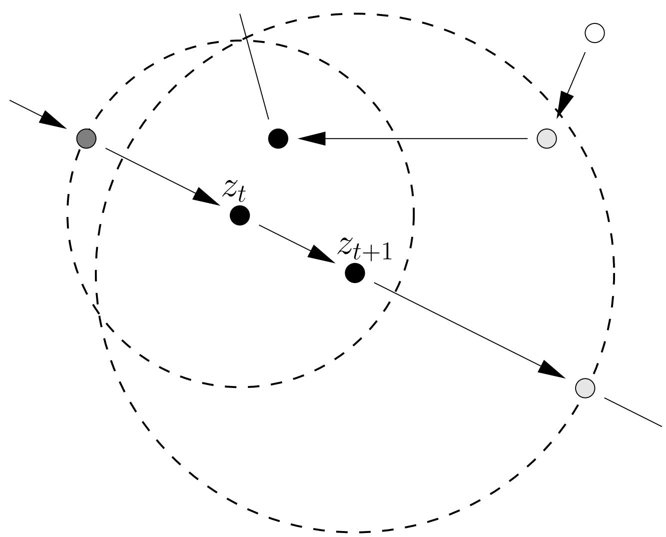
- Action Respecting Embedding to the rescue.

Action Respecting Embedding (ARE)

- Like SDE, learns a kernel matrix K through optimization.
- Uses the kernel matrix K with kernel PCA.
- Exploits the action data (u_t) in two ways.
  - Non-uniform neighborhood graph.
  - Action respecting constraints.

Non-Uniform Neighborhoods

- Actions only have a small effect on the robot's pose.
- If $z_t \xrightarrow[u_t]{} z_{t+1}$, then $\Phi(z_t)$ and $\Phi(z_{t+1})$ should be close.
- Set z_t's neighborhood size to include z_{t-1} and z_{t+1}.



- Neighbor graph uses this non-uniform neighborhood size.

Action Respecting Constraints

- Constrain the representation so that actions must be simple transformations.

- What's a simple transformation?
  - Linear transformations are simple.

$$f(x) = Ax + b$$

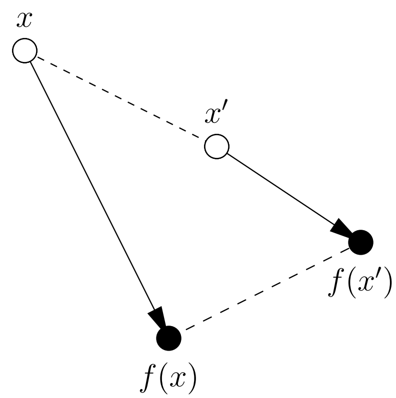
  - Distance preserving ones are just slightly simpler.

$$f(x) = Ax + b \text{ where } A^T A = I.$$

    i.e., rotation and translation, but no scaling.

Action Respecting Constraints

- Distance preserving $\iff ||f(x) - f(x')|| = ||x - x'||$.



- For our representation, if $u_i = u_j = u$,

$$\begin{aligned} ||f_u(\Phi(z_i)) - f_u(\Phi(z_j))|| &= ||\Phi(z_i) - \Phi(z_j)|| \\ ||\Phi(z_{i+1}) - \Phi(z_{j+1})|| &= ||\Phi(z_i) - \Phi(z_j)|| \\ K_{(i+1)(i+1)} - 2K_{(i+1)(j+1)} + K_{(j+1)(j+1)} &= K_{ii} - 2K_{ij} + K_{jj} \end{aligned}$$

Action Respecting Embedding

- Optimization Problem:

$$\begin{array}{ll}
\text{Maximize:} & \mathrm{Tr}(K) \\
\text{Subject to:} & K \succeq 0 \\
& \sum_{ij} K_{ij} = 0 \\
& \forall i,j \quad \eta_{ij} > 0 \vee [\eta^T\eta]_{ij} > 0 \Rightarrow \\
& \qquad K_{ii} - 2K_{ij} + K_{jj} = ||z_i - z_j||^2 \\
& \forall i,j \quad u_i = u_j \Rightarrow \\
& \qquad K_{(i+1)(i+1)} - 2K_{(i+1)(j+1)} + K_{(j+1)(j+1)} = \\
& \qquad K_{ii} - 2K_{ij} + K_{jj}
\end{array}$$

where η is the non-uniform neighborhood graph.

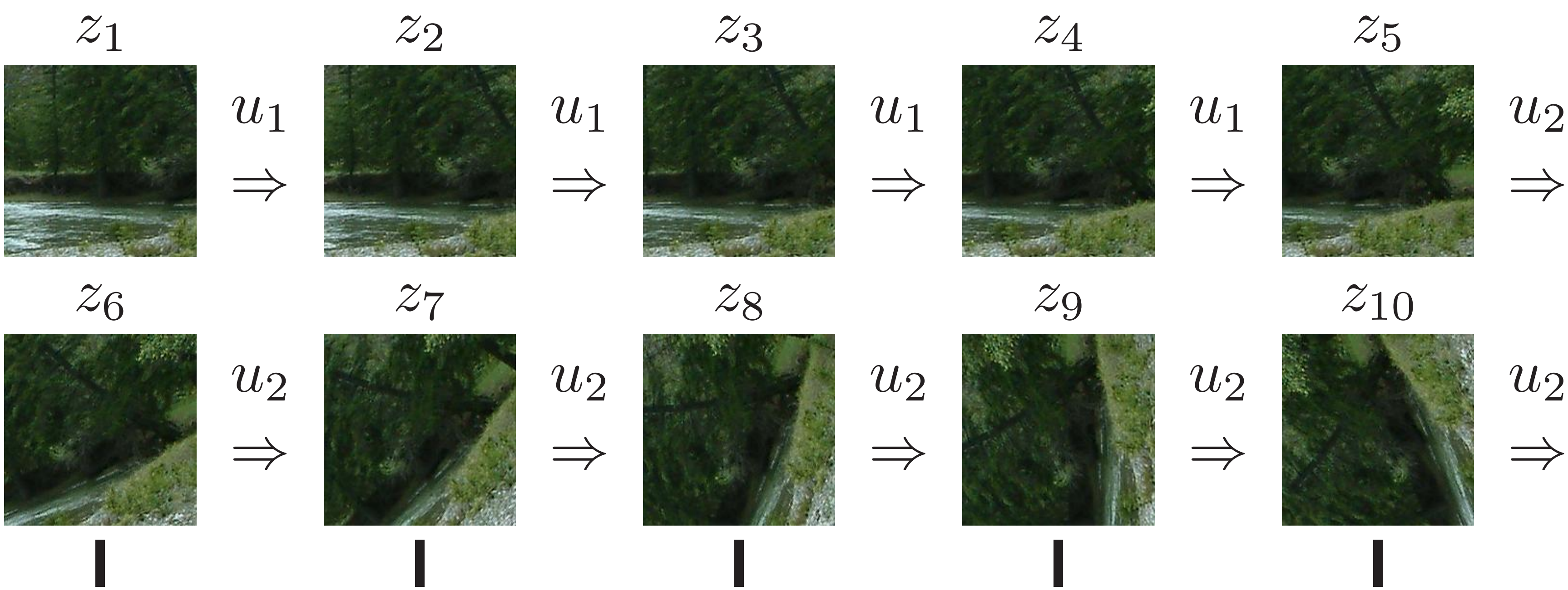
- Use learned K with kernel PCA to extract $x_1, \ldots, x_T$.

Overview

- What is a Map?
- Subjective Mapping
- Action Respecting Embedding (ARE)
- Results
  - Representations
  - Planning
  - Localization
- Future Work

ImageBot

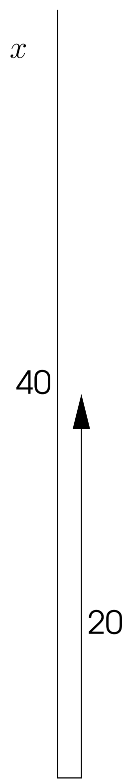
- Construct a map from a stream of input.

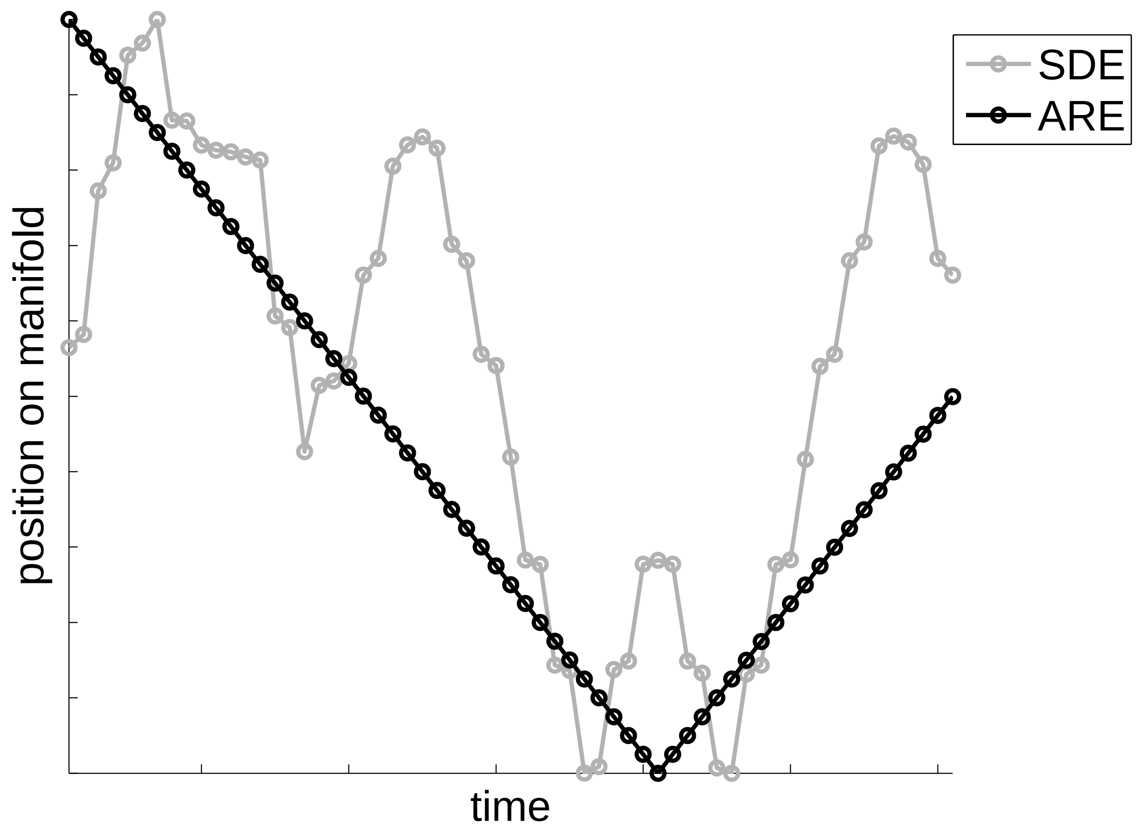


- Actions are labels with no semantics.

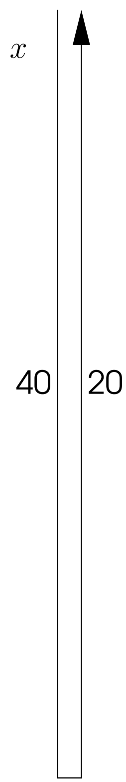
- No image features, just high-dimensional vectors.

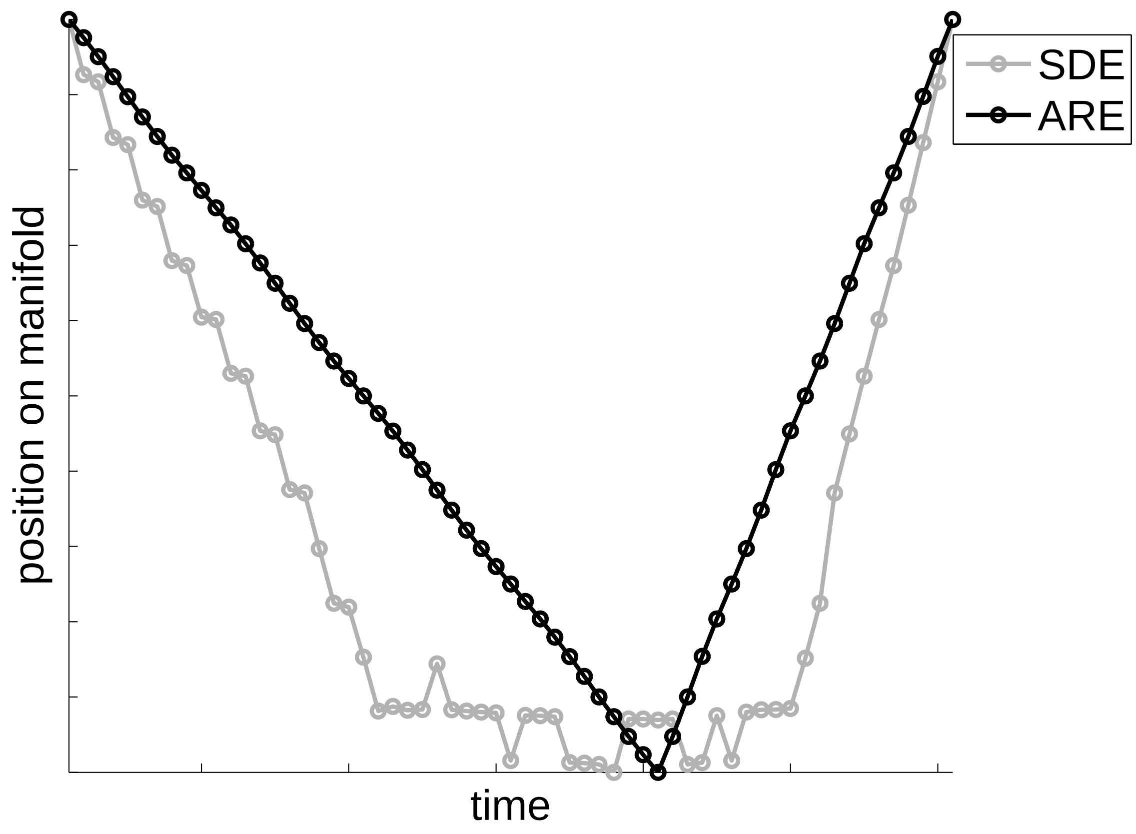
Learned Representations



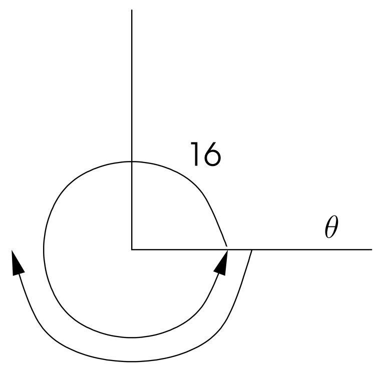


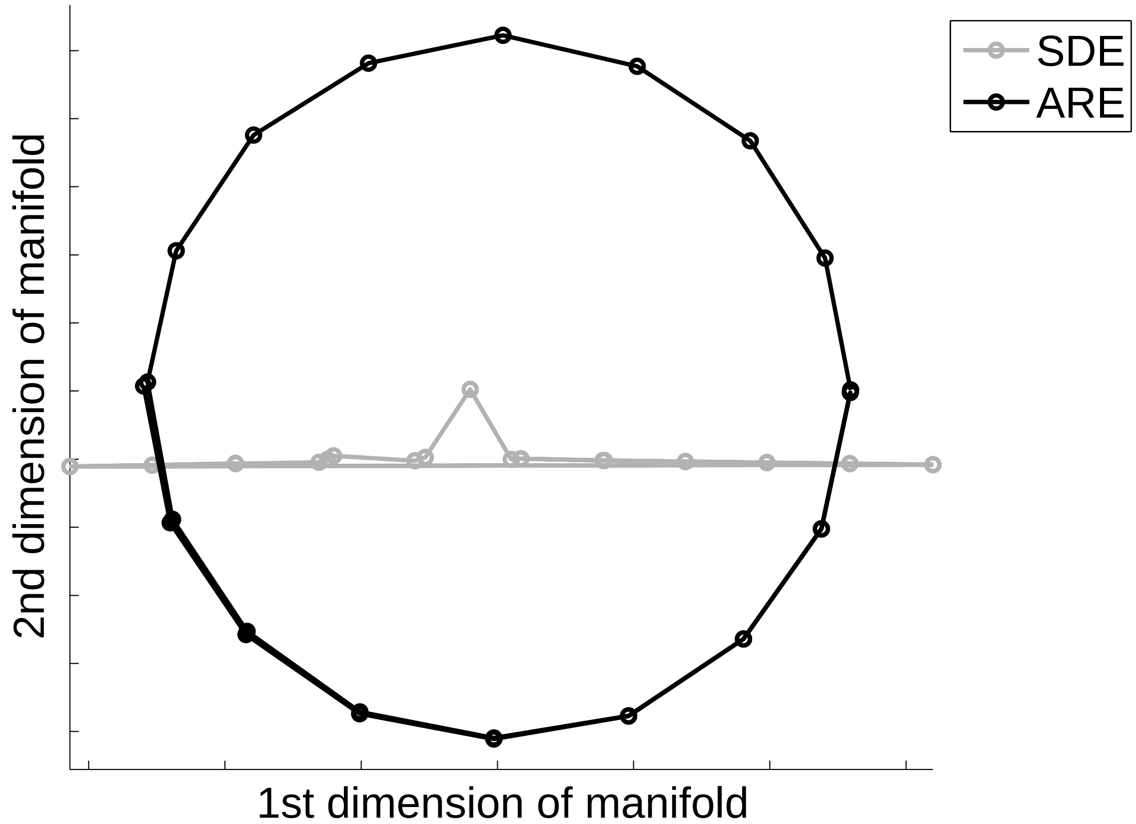
Learned Representations



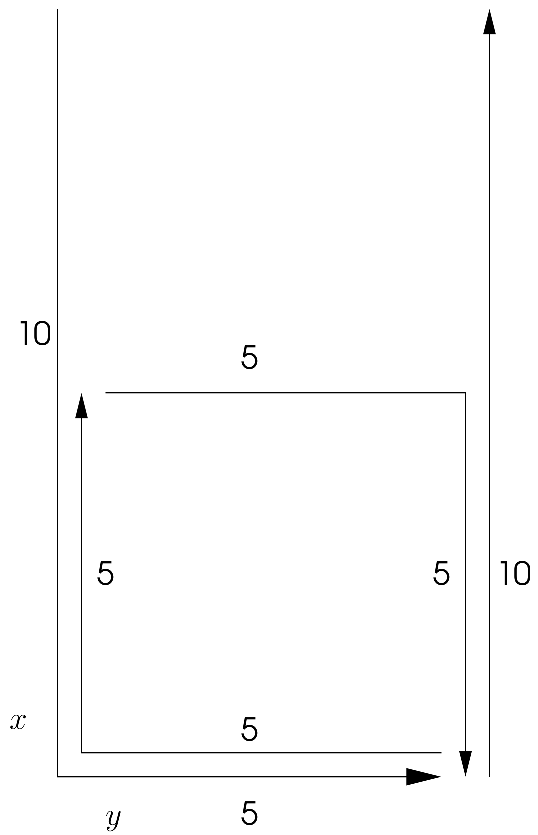


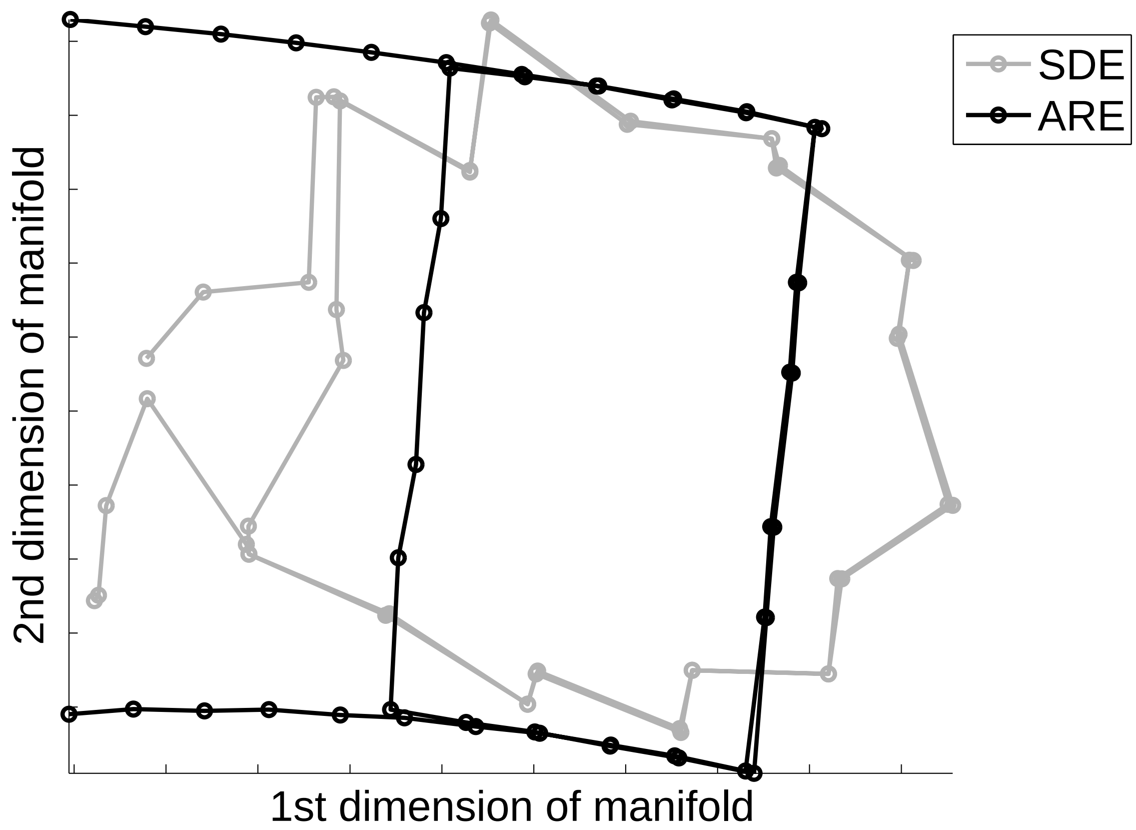
Learned Representations



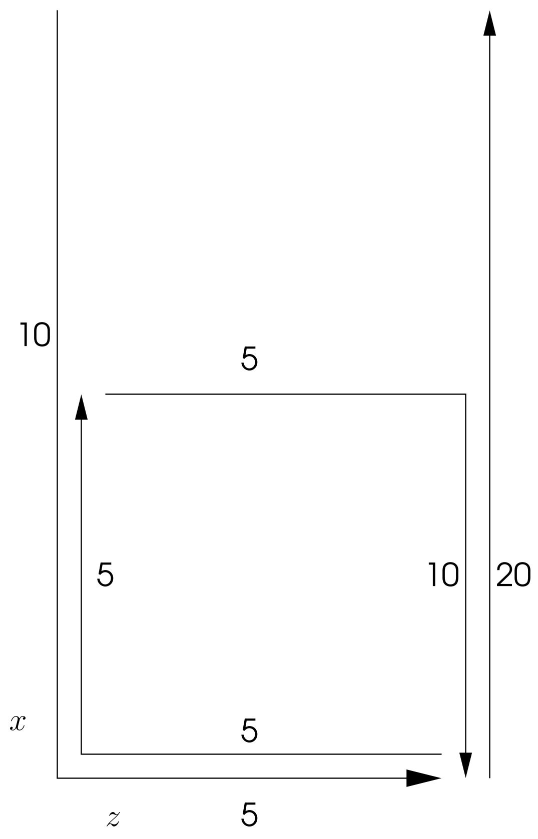


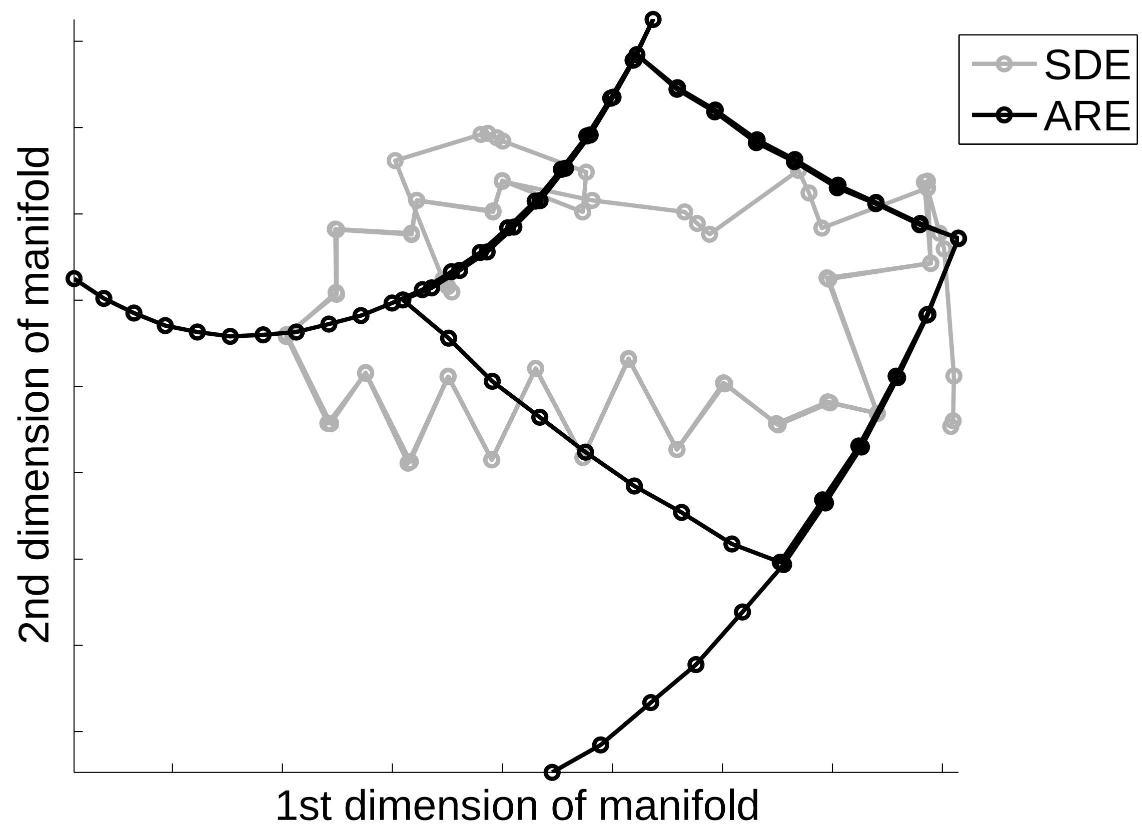
Learned Representations



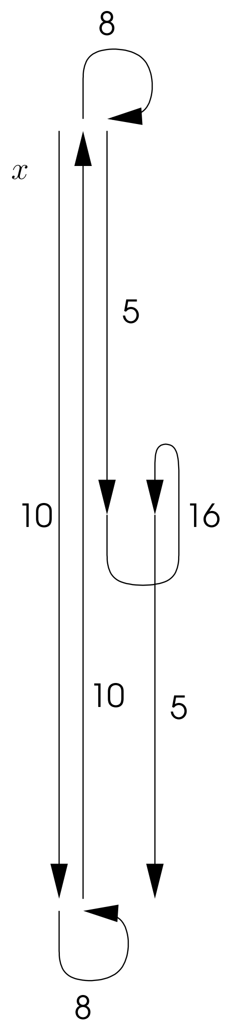


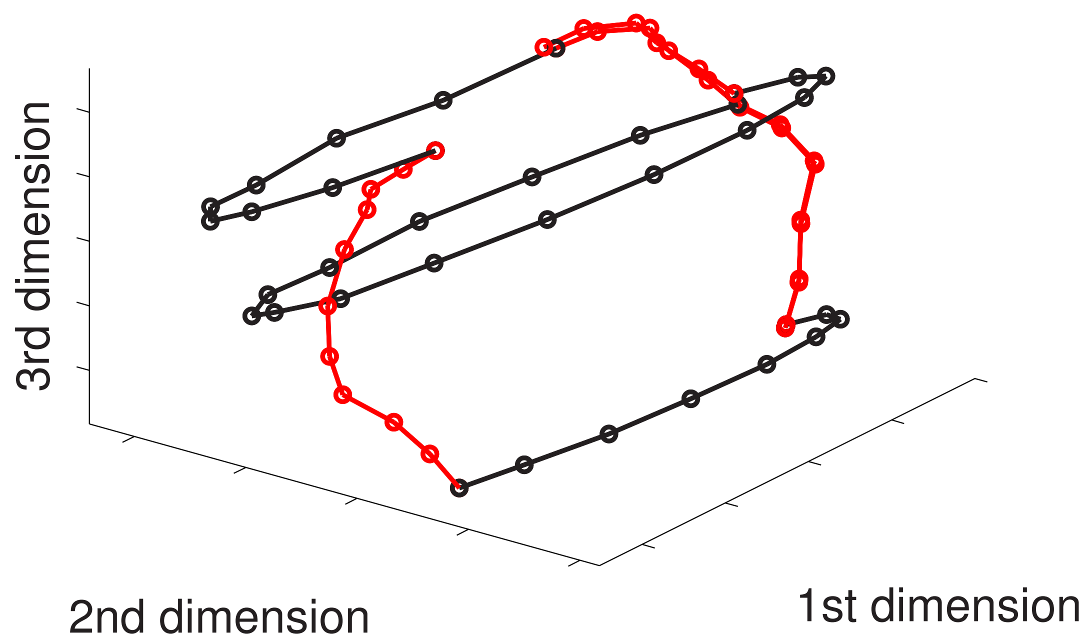
Learned Representations





Learned Representations





Map Questions

1. Where have I been?

2. Where am I now?

3. How do I get where I want to go? (a.k.a planning)

Planning

- Problem: from x_T find a sequence of actions that will reach x_* where $x_* \in \{x_1, \ldots, x_T\}$.
- Need a Model: If we take action u from x_t, what is x_{t+1}?
  - ARE guarantees the existence of a distance preserving function, f_u,

$$\forall t \quad u_t = u \Rightarrow f_u(x_t) = x_{t+1}.$$

  - We just need to extract f_u from the representation.

Planning with Procrustes

- Variation on the extended orthonormal Procrustes problem.

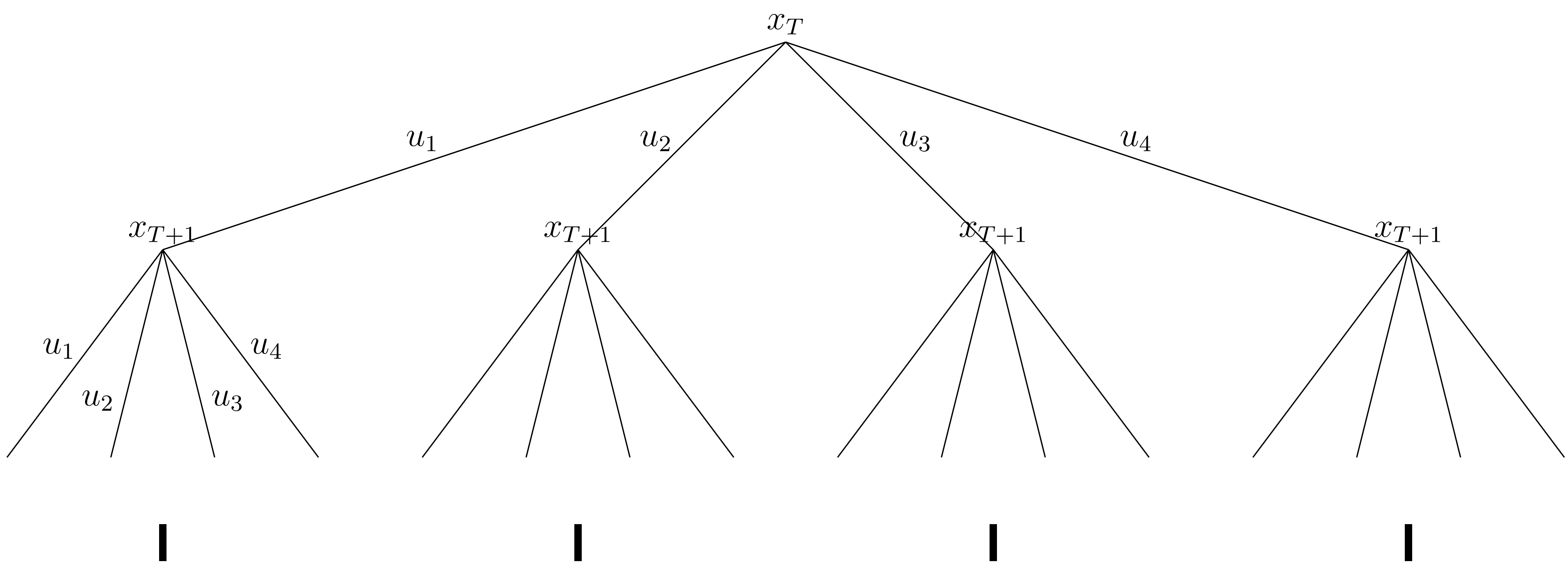
  Minimize: $\sum_t ||Ax_t + b - x_{t+1}||^2 \cdot \mathbf{1}_{u_t=u}$
  A, b

  Subject to: $A^T A = I$

- Although not convex, it has an analytic solution!

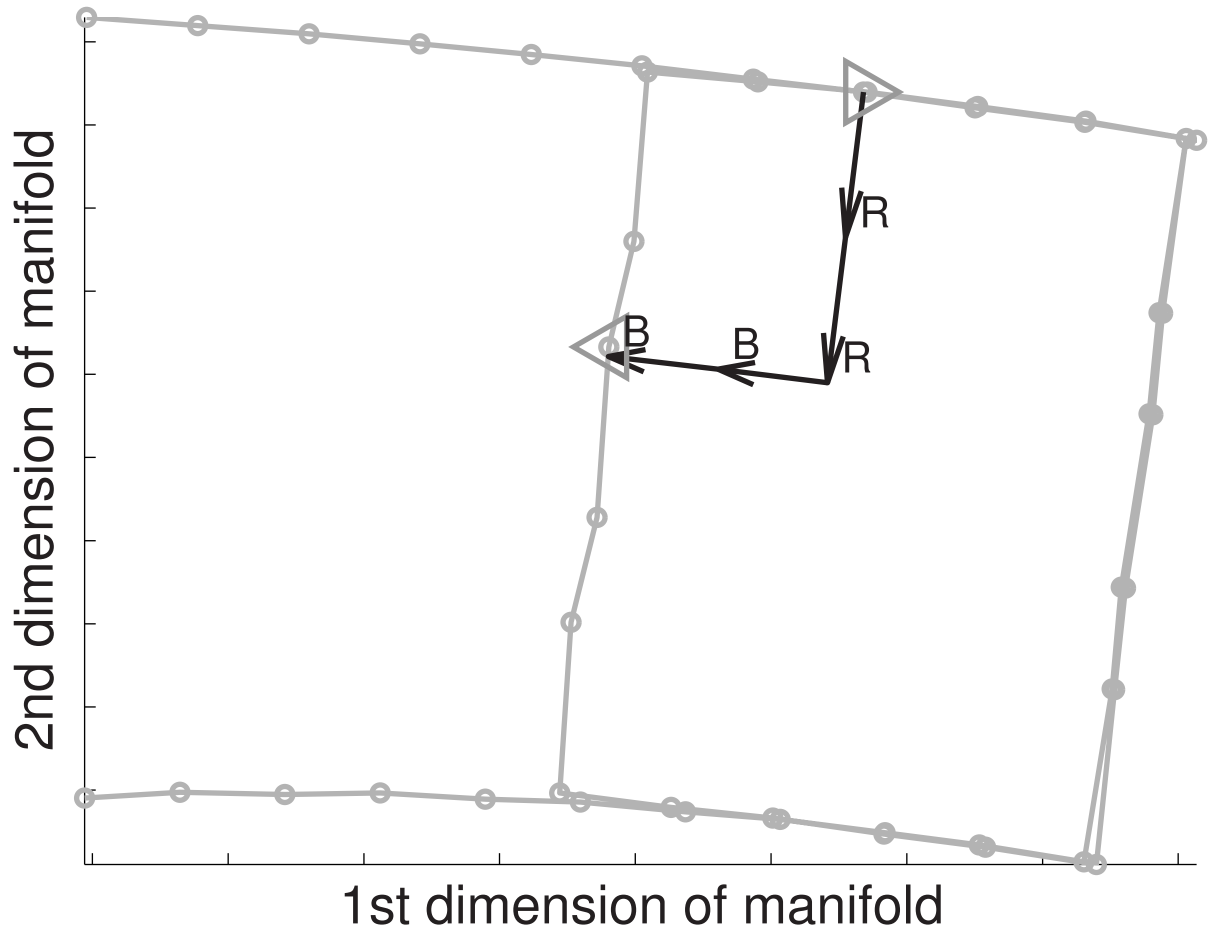
Planning with Search

- Using our action models, we can search for a plan.

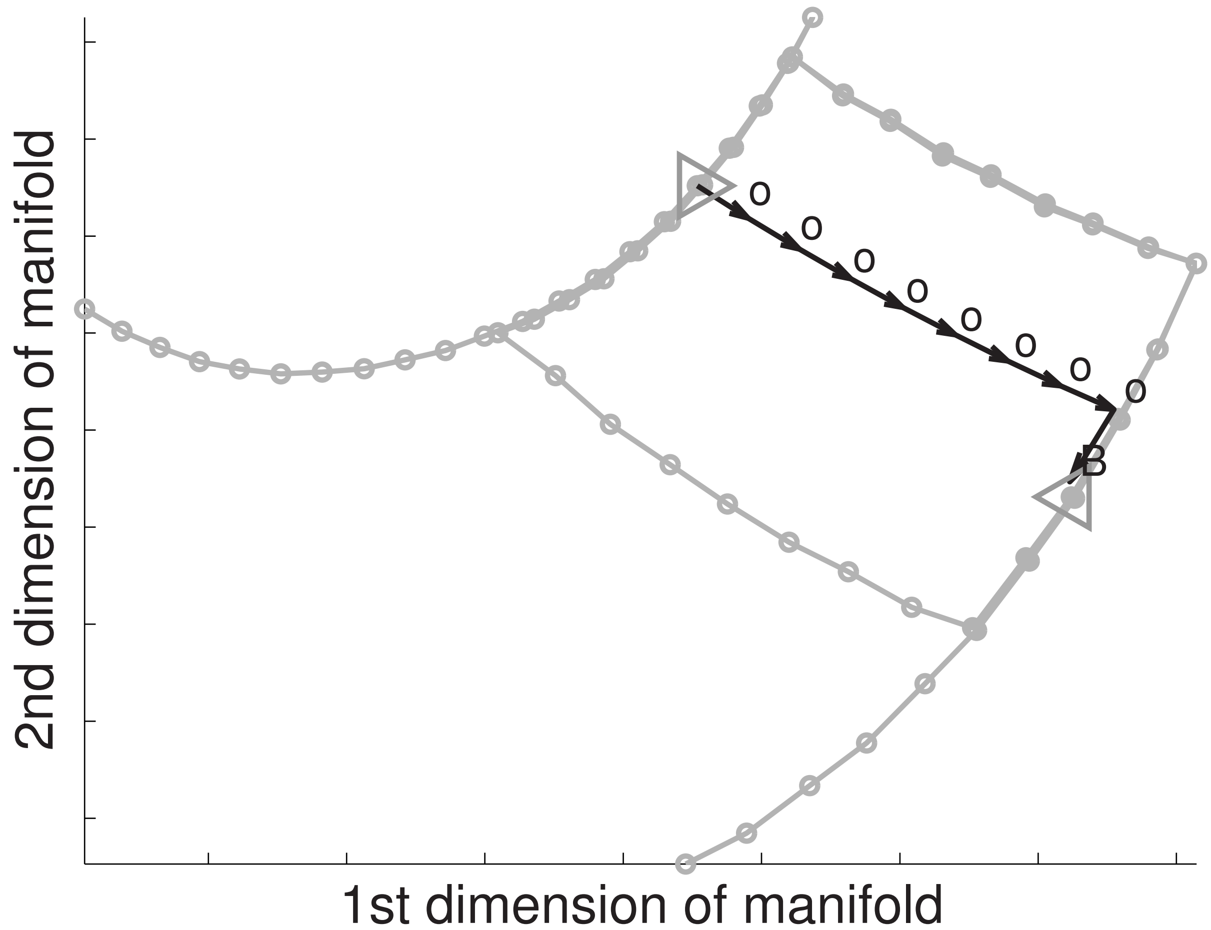


- Simple iterative deepening depth-first search.

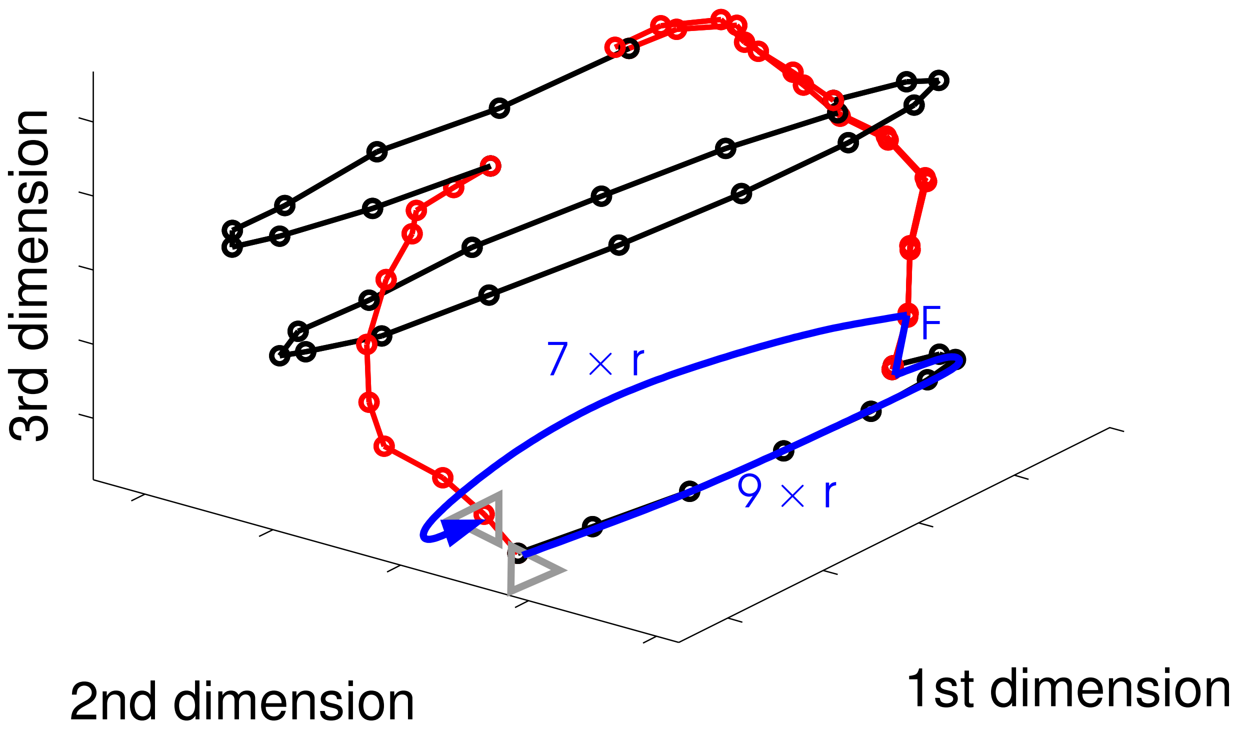
Planning Results



Planning Results



Planning Results



Map Questions

1. Where have I been?

2. Where am I now? (a.k.a localization)

3. How do I get where I want to go?

Localization

- Maps are Models.
  - Motion: $P(x_{t+1}|x_t, u_t)$.
  - Sensor: $P(z_t|x_t)$.
- Extract noisy action models from experience.
  - Noise is encoded in the later principal components.
  - Procrustes extracts the mean.
  - Regression errors define the variance.
- Extract sensor model based on correlation between image distances and distance in the representation.
- Monte Carlo localization (MCL).

Localization Results

- Noisy actions in ImageBot.
- No ground truth with subjective representations.
- Choose the "closest" point in the training trajectory.

Trajectory	Mean Error (Pixels)		
	Oracle	ARE	Random
Straight line	4.82	10.39	86.83
"A" with translation	3.62	14.81	104.56
"A" with zoom	1.71	19.58	84.67

Summary

- ARE automatically extracts a representation from only subjective experience of observations and actions.
- The representation is low-dimensional, action-respecting.
- And can answer all three map questions.
  1. Where have I been?
  2. Where am I now?
  3. How do I get where I want to go?

Overview

- What is a Map?
- Subjective Mapping
- Action Respecting Embedding (ARE)
- Results and Applications
- Future Work

Future Work

- Faster ARE. Customized solver?
- Better ARE.
  - Learning representations with walls.
  - State aliasing.
  - Continuous actions.
- Using ARE.
  - Heuristics for planning with ARE.
  - Behavior learning with ARE representations.
- Subjective maps for real robots.

Questions?