

Comparison of Constrained Geometric Approximation Strategies for Planar Information States

Yang Song, Jason M. O’Kane

Email: song24@email.sc.edu

jokane@cse.sc.edu



UNIVERSITY OF
SOUTH CAROLINA®

Outlines

- 1 Problem Statement
- 2 Range Space
 - Range Space Definition
 - Disk Range Space
 - Rectangle Range Space
- 3 Double-Rectangle Range Space
- 4 Experiments
- 5 Conclusions

Problem Statement

- **Goal:**

For an extremely simple robot with:

- computation limitations
- moving and sensing uncertainties

represent and reason about uncertainty in its own states efficiently.

- **Basic Idea:**

Explicitly represent what the robot knows as an information state (*I-state*).

- **Intuition:**

Accelerate time-consuming operations by maintaining only an **overapproximation** of the true *I-state*, and constraining this approximation to have a simple geometric form.

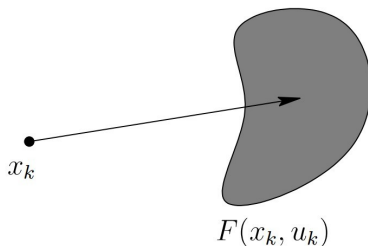


SRV-1 Surveyor Robot

States

For a robot whose current state is $x_k \in X$ at stage k , we can update its state using a transition function $F(x_k, u_k)$ by applying its current action u_k .

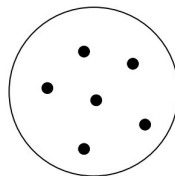
Consider the moving uncertainty, we can describe the set of possible states at stage $k + 1$



Robot's state transition with uncertainty [*Planning Algorithms*, S. LaValle, 2006]

Information States

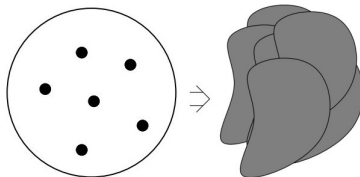
Assume that current real state of the robot could not be observed directly. The robot could maintain a set of possible states, which is *I-state* η_k , to make its decisions.



η_k : contains all possible states at stage k [*Planning Algorithms*, S. LaValle, 2006]

Set Transition Function

Assume that current real state of the robot can not be observed directly. The robot could maintain a set of possible states, which is *I-state* η_k , to make its decisions.



For each $x_k \in \eta_k$, applying set-valued transition function F yields $\bigcup_{x_k \in \eta_k} F(x_k, u_k)$. [Planning Algorithms, S. LaValle, 2006]

Set Observation Function

- Assume the observation preimage $H(y_k) \in X$ derived from the observation function $h : X \rightarrow Y$, is a planar disk, where $y_k \in Y$ is the observation.
- The *I-state* could be updated by intersecting transited set of possible states with the observation preimage:

$$\eta_{k+1} = \left[\bigcup_{x_k \in \eta_k} F(x_k, u_k) \right] \cap H(y_k). \quad (1)$$

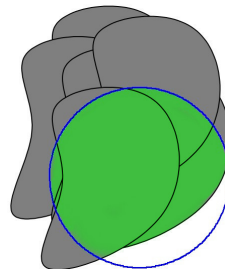


Figure: The green region is updated *I-state* η_{k+1}

Prior Work

Prior research done by (B. Tovar and S. M. LaValle) and (J. van den Berg, P. Abbeel, and K. Goldberg) used probabilistic representations for planning

Prior work by the authors (J.O'Kane) has used preliminary versions of the constrained geometric approximation method using specific, fixed range spaces.

New contributions

- 1 A careful formulation of the operations in the range space \mathcal{R} .
- 2 Algorithms for double-rectangle range space \mathcal{R}_{direct} .
- 3 A series of experiments for effectiveness comparison of different range spaces.

Outlines

- 1 Problem Statement
- 2 Range Space
 - Range Space Definition
 - Disk Range Space
 - Rectangle Range Space
- 3 Double-Rectangle Range Space
- 4 Experiments
- 5 Conclusions

Range Space

Definition

A **range space** $\mathcal{R} \subseteq \mathcal{I}$ is a set of I-states, contains approximation of *I-states*, $A(\eta_k) \in \mathcal{R}$, equipped with two operations:

- 1 An *approximate observation update function* $O : \mathcal{R} \times Y \rightarrow \mathcal{R}$, such that if $\eta_k \subseteq A(\eta_k)$, then

$$\eta_k \cap H(y_k) \subseteq O(A(\eta_k), u_k)$$
- 2 An *approximate action update function* $T : \mathcal{R} \times U \rightarrow \mathcal{R}$, such that if $\eta_k \subseteq A(\eta_k)$, then

$$\bigcup_{x_k \in \eta_k} F(x_k, u_k) \subseteq T(A(\eta_k), u_k)$$

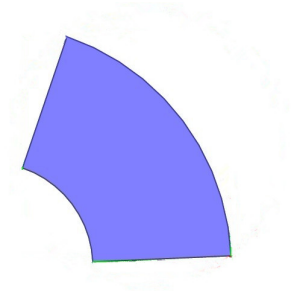


Figure: The blue region denotes the *I-state* η_k

Range Space

Definition

A **range space** $\mathcal{R} \subseteq \mathcal{I}$ is a set of I-states, contains approximation of *I-states*, $A(\eta_k) \in \mathcal{R}$, equipped with two operations:

- 1 An *approximate observation update function* $O : \mathcal{R} \times Y \rightarrow \mathcal{R}$, such that if $\eta_k \subseteq A(\eta_k)$, then

$$\eta_k \cap H(y_k) \subseteq O(A(\eta_k), u_k)$$

- 2 An *approximate action update function* $T : \mathcal{R} \times U \rightarrow \mathcal{R}$, such that if $\eta_k \subseteq A(\eta_k)$, then

$$\bigcup_{x_k \in \eta_k} F(x_k, u_k) \subseteq T(A(\eta_k), u_k)$$

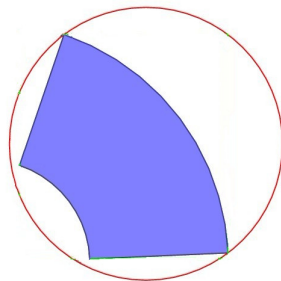


Figure: Disk overapproximation $A(\eta_k) \in \mathcal{R}_{disk}$ in red.

Range Space

Definition

A **range space** $\mathcal{R} \subseteq \mathcal{I}$ is a set of I-states, contains approximation of *I-states* $A(\eta_k) \in \mathcal{R}$, equipped with two operations:

- 1 An *approximate observation update function* $O : \mathcal{R} \times Y \rightarrow \mathcal{R}$, such that if $\eta_k \subseteq A(\eta_k)$, then

$$\eta_k \cap H(y_k) \subseteq O(A(\eta_k), u_k)$$

- 2 An *approximate action update function* $T : \mathcal{R} \times U \rightarrow \mathcal{R}$, such that if $\eta_k \subseteq A(\eta_k)$, then

$$\bigcup_{x_k \in \eta_k} F(x_k, u_k) \subseteq T(A(\eta_k), u_k)$$

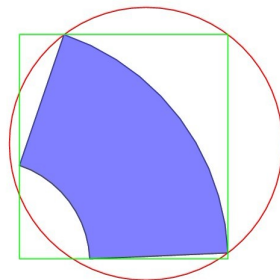
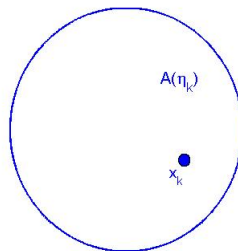


Figure: Rectangle overapproximation $A(\eta_k) \in \mathcal{R}_{rect}$.

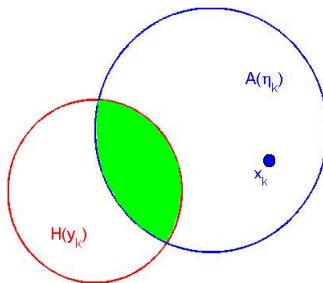
Observation Update in Disk Range Space \mathcal{R}_{disk}

Approximated I-state $A(\eta_k) \in \mathcal{R}_{disk}$, where x_k denotes the real state but unknown to robot.



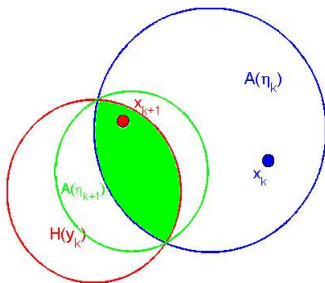
Observation Update in Disk Range Space \mathcal{R}_{disk}

Approximated I-state $A(\eta_k) \in \mathcal{R}_{disk}$ intersects with observation preimage $H(y_k)$.



Observation Update in Disk Range Space \mathcal{R}_{disk}

The green region of intersection is the updated I-state η_{k+1} , and the green disk is the approximation $A(\eta_{k+1})$ of η_{k+1} .



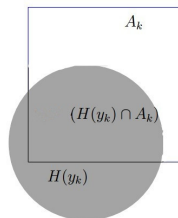
Observation Update in \mathcal{R}_{rect}

Definition

$AABB(S)$: For any compact set $S \subset \mathbb{R}^2$, let $AABB(S)$ denote the its smallest “axis-aligned bounding box.”

In \mathcal{R}_{rect} , computing approximate observation update function O_{rect} takes $O(1)$ time:

$$O_{rect}(A_k, y_k) = AABB(H(y_k) \cap A(\eta_k)), A_k = A(\eta_k) \in \mathcal{R}_{rect}$$



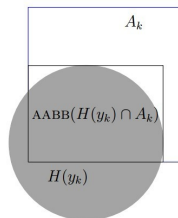
Observation Update in \mathcal{R}_{rect}

Definition

$AABB(S)$: For any compact set $S \subset \mathbb{R}^2$, let $AABB(S)$ denote the its smallest “axis-aligned bounding box.”

In \mathcal{R}_{rect} , computing approximate observation update function O_{rect} takes $O(1)$ time:

$$O_{rect}(A_k, y_k) = AABB(H(y_k) \cap A(\eta_k)), A_k = A(\eta_k) \in \mathcal{R}_{rect}$$



Action Update in \mathcal{R}_{rect}

In \mathcal{R}_{rect} , computing approximate action update function T_{rect} :

$$T_{rect}(A(\eta_k), u_k) = AAB B(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus AAB B(\Theta(u_k))])$$

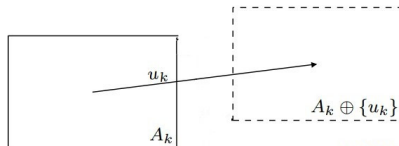


Figure: Transition of $A(\eta_k)$ given action u_k , \oplus is Minkowski addition of two sets.

Action Update in \mathcal{R}_{rect}

In \mathcal{R}_{rect} , computing approximate action update function T_{rect} :

$$T_{rect}(A(\eta_k), u_k) = AAB B(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus AAB B(\Theta(u_k))])$$

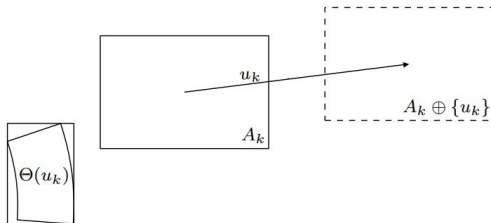


Figure: Consider approximation of the bounded motion uncertainty $\Theta(u_k)$.

Action Update in \mathcal{R}_{rect}

In \mathcal{R}_{rect} , computing approximate action update function T_{rect} :

$$T_{rect}(A(\eta_k), u_k) = AAB B(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus AAB B(\Theta(u_k))])$$

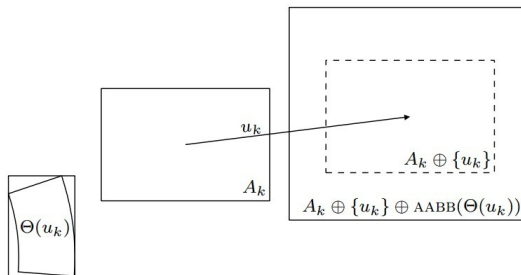


Figure: Find Minkowski sum of bounded noise and approximation transition

Action Update in \mathcal{R}_{rect}

In \mathcal{R}_{rect} , computing approximate action update function T_{rect} :

$$T_{rect}(A(\eta_k), u_k) = AAB B(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus AAB B(\Theta(u_k))])$$

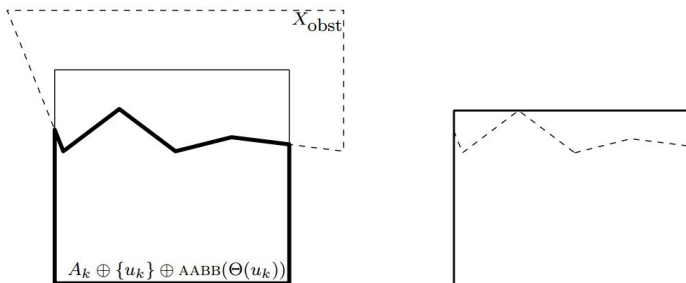


Figure: If there is obstacles, intersect with X_{free} first and then find the bounding box of the intersection.

Outlines

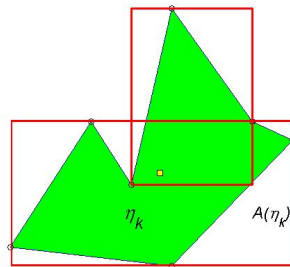
- 1 Problem Statement
- 2 Range Space
 - Range Space Definition
 - Disk Range Space
 - Rectangle Range Space
- 3 Double-Rectangle Range Space
- 4 Experiments
- 5 Conclusions

Double-Rectangle approximated I-state

- For better overapproximation quality for non-convex *I-states*, we proposed a more expressive range space of *double rectangles*:

$$\mathcal{R}_{direct} = \{R_1 \cup R_2 \mid R_1, R_2 \in \mathcal{R}_{rect}\} \quad (2)$$

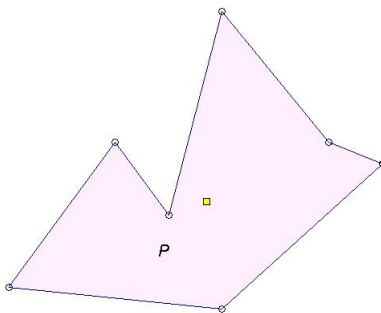
- Aims to improve the approximation quality.



DRAP Algorithm

“Double Rectangle Around Polygon” (**DRAP**) algorithm:

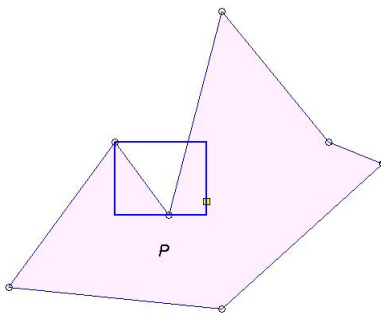
- input is a n -edge polygonal region of the plane
- output is a small double rectangle containing that polygon
- run time: $O(n^3)$



DRAP Algorithm

“Double Rectangle Around Polygon” (**DRAP**) algorithm:

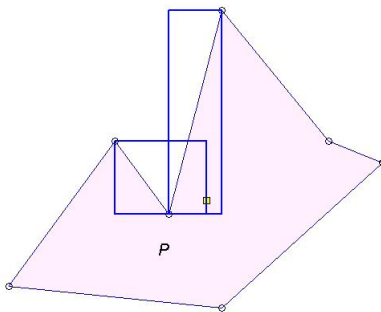
- input is a n -edge polygonal region of the plane
- output is a small double rectangle containing that polygon
- run time: $O(n^3)$



DRAP Algorithm

“Double Rectangle Around Polygon” (**DRAP**) algorithm:

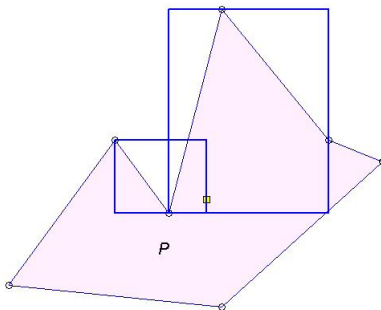
- input is a n -edge polygonal region of the plane
- output is a small double rectangle containing that polygon
- run time: $O(n^3)$



DRAP Algorithm

“Double Rectangle Around Polygon” (**DRAP**) algorithm:

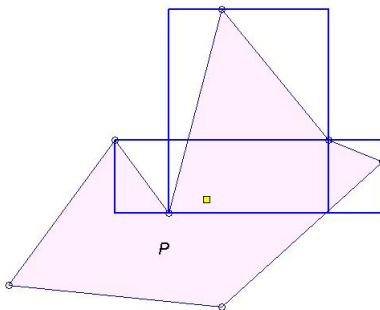
- input is a n -edge polygonal region of the plane
- output is a small double rectangle containing that polygon
- run time: $O(n^3)$



DRAP Algorithm

“Double Rectangle Around Polygon” (**DRAP**) algorithm:

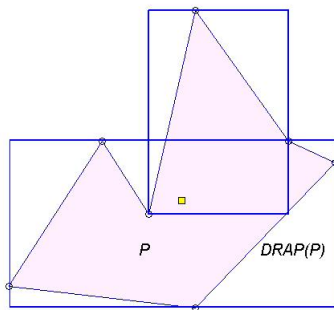
- input is a n -edge polygonal region of the plane
- output is a small double rectangle containing that polygon
- run time: $O(n^3)$



DRAP Algorithm

“Double Rectangle Around Polygon” (**DRAP**) algorithm:

- input is a n -edge polygonal region of the plane
- output is a small double rectangle containing that polygon
- run time: $O(n^3)$

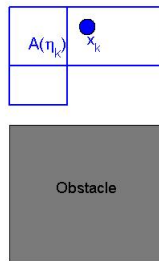


Operations in \mathcal{R}_{direct}

Based on **DRAP** Algorithm, we can define the range space operations on \mathcal{R}_{direct} in a manner similar to those for \mathcal{R}_{rect} . For a double rectangle approximated *I-state* $A(\eta_k) = R_1 \cup R_2$, we have

$$T_{direct}(A(\eta_k), u_k) = DRAP(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus DRAP(\Theta(u_k))]), \quad (3)$$

$$O_{direct}(A(\eta_k), y_k) = AAB B(H(y_k) \cap R_1) \cup AAB B(H(y_k) \cap R_2). \quad (4)$$

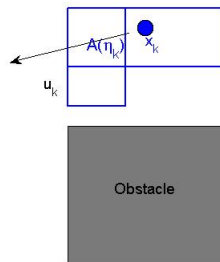


Operations in \mathcal{R}_{direct}

Based on **DRAP** Algorithm, we can define the range space operations on \mathcal{R}_{direct} in a manner similar to those for \mathcal{R}_{rect} . For a double rectangle approximated *I-state* $A(\eta_k) = R_1 \cup R_2$, we have

$$T_{direct}(A(\eta_k), u_k) = DRAP(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus DRAP(\Theta(u_k))]), \quad (5)$$

$$O_{direct}(A(\eta_k), y_k) = AAB B(H(y_k) \cap R_1) \cup AAB B(H(y_k) \cap R_2). \quad (6)$$

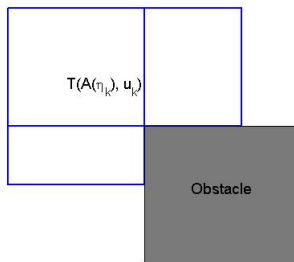


Operations in \mathcal{R}_{direct}

Based on **DRAP** Algorithm, we can define the range space operations on \mathcal{R}_{direct} in a manner similar to those for \mathcal{R}_{rect} . For a double rectangle approximated *I-state* $A(\eta_k) = R_1 \cup R_2$, we have

$$T_{direct}(A(\eta_k), u_k) = DRAP(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus DRAP(\Theta(u_k))]), \quad (7)$$

$$O_{direct}(A(\eta_k), y_k) = AAB B(H(y_k) \cap R_1) \cup AAB B(H(y_k) \cap R_2). \quad (8)$$

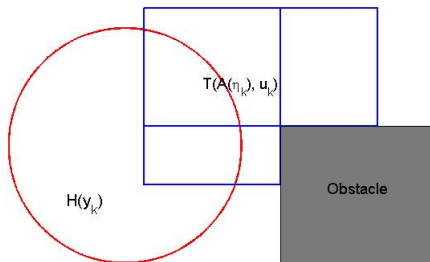


Operations in \mathcal{R}_{direct}

Based on **DRAP** Algorithm, we can define the range space operations on \mathcal{R}_{direct} in a manner similar to those for \mathcal{R}_{rect} . For a double rectangle approximated *I-state* $A(\eta_k) = R_1 \cup R_2$, we have

$$T_{direct}(A(\eta_k), u_k) = DRAP(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus DRAP(\Theta(u_k))]), \quad (9)$$

$$O_{direct}(A(\eta_k), y_k) = AABB(H(y_k) \cap R_1) \cup AABB(H(y_k) \cap R_2). \quad (10)$$

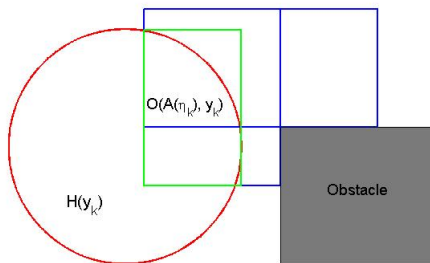


Operations in \mathcal{R}_{direct}

Based on **DRAP** Algorithm, we can define the range space operations on \mathcal{R}_{direct} in a manner similar to those for \mathcal{R}_{rect} . For a double rectangle approximated *I-state* $A(\eta_k) = R_1 \cup R_2$, we have

$$T_{direct}(A(\eta_k), u_k) = DRAP(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus DRAP(\Theta(u_k))]), \quad (11)$$

$$O_{direct}(A(\eta_k), y_k) = AAB B(H(y_k) \cap R_1) \cup AAB B(H(y_k) \cap R_2). \quad (12)$$

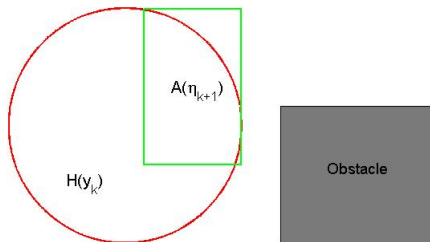


Operations in \mathcal{R}_{direct}

Based on **DRAP** Algorithm, we can define the range space operations on \mathcal{R}_{direct} in a manner similar to those for \mathcal{R}_{rect} . For a double rectangle approximated *I-state* $A(\eta_k) = R_1 \cup R_2$, we have

$$T_{direct}(A(\eta_k), u_k) = DRAP(X_{free} \cap [A(\eta_k) \oplus \{u_k\} \oplus DRAP(\Theta(u_k))]), \quad (13)$$

$$O_{direct}(A(\eta_k), y_k) = AAB B(H(y_k) \cap R_1) \cup AAB B(H(y_k) \cap R_2). \quad (14)$$



Outlines

- 1 Problem Statement
- 2 Range Space
 - Range Space Definition
 - Disk Range Space
 - Rectangle Range Space
- 3 Double-Rectangle Range Space
- 4 Experiments
- 5 Conclusions

Experiments: Task in Various Environments

To verify the effectiveness and efficiency of CGA for a navigation task, in comparison with using the true *information state*, we conducted experiments using 3 environments, and 3 range spaces \mathcal{R}_{disk} , \mathcal{R}_{rect} , and \mathcal{R}_{direct} .

ASSUMPTIONS:

- A point robot follows predefined waypoints guided by centroid point of the approximated *I-state*
- Robot can detect presence but not distance to the waypoints
- Landmarks are pseudo-randomly generated (Number Matters)
- Initial I-state η_0 is given.

► Start movie

Experiments Results

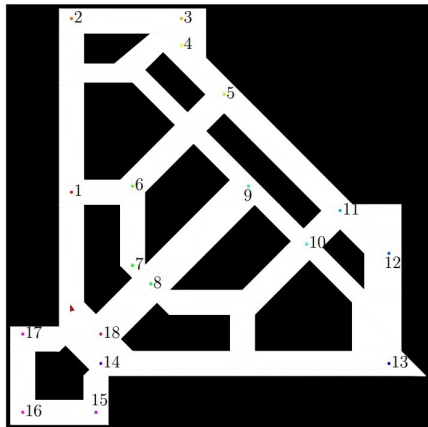
For each environment, we measure:

- the relationship between task completion and number of landmarks,
- the time required to compute approximated *I-state* compared to the high-quality polygonal representation of the exact *I-state*,
- the approximation ratio Q_k

$$Q_k = \frac{1}{k} \sum_{i=1}^k \frac{\mathbb{A}(\eta_i)}{\mathbb{A}(A(\eta_i))} \quad (15)$$

where $\mathbb{A}(\diamond)$ denotes the area of set $\diamond \subset \mathbb{R}^2$

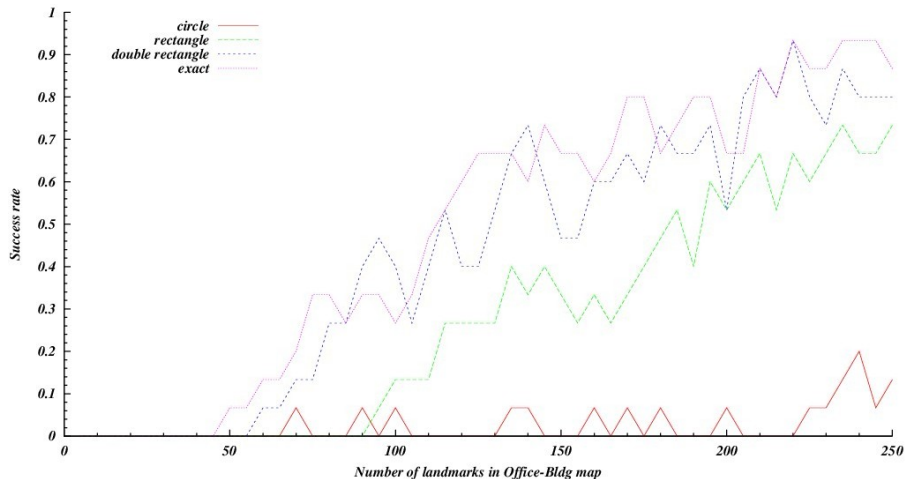
Comparison of Efficiency and Accuracy



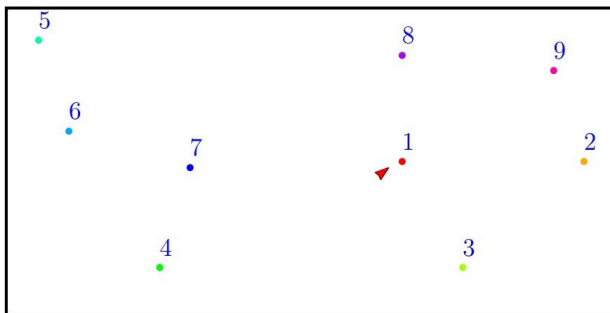
Comparison of the computation time and approximation ratio for approximated I -states to the corresponding computation for the exact I -states shows the trade-off between time and accuracy.

Range Space	Time (s)	Approximation Ratio
\mathcal{R}_{disk}	0.292	0.220
\mathcal{R}_{rect}	0.415	0.661
$\mathcal{R}_{dblrect}$	1.491	0.720
\mathcal{I}	26.895	1.000

Comparison of Success Rate in Office-like Environment

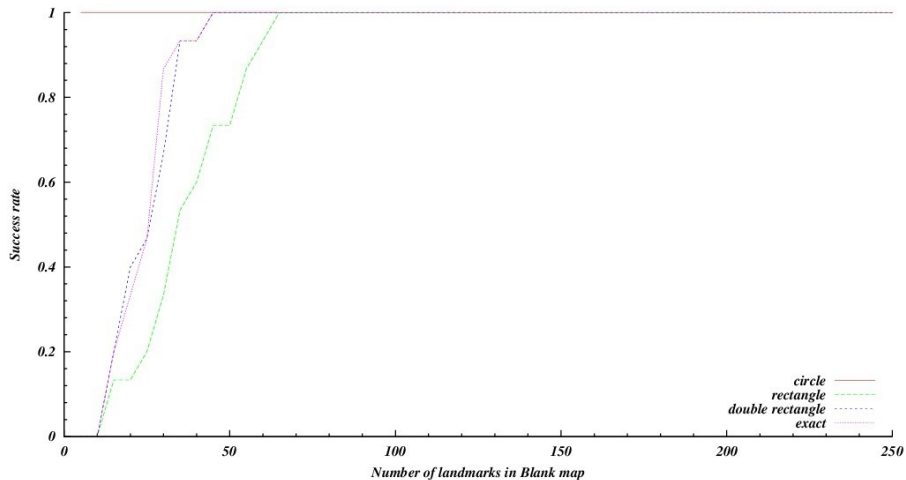


Computation and Approximation ratio in Obstacle-free Environment

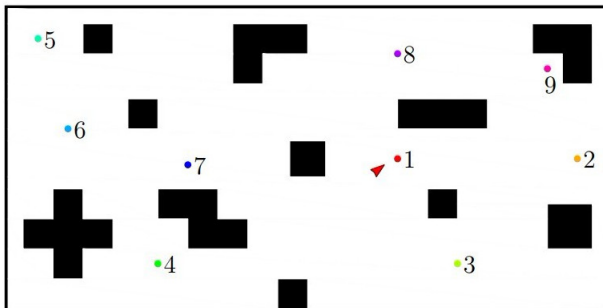


Range Space	Time (s)	Approximation Ratio
\mathcal{R}_{disk}	0.163	0.155
\mathcal{R}_{rect}	0.396	0.642
$\mathcal{R}_{dblrect}$	1.021	0.684
\mathcal{I}	10.074	1.000

Comparison of Success Rate in Obstacle-free Environment

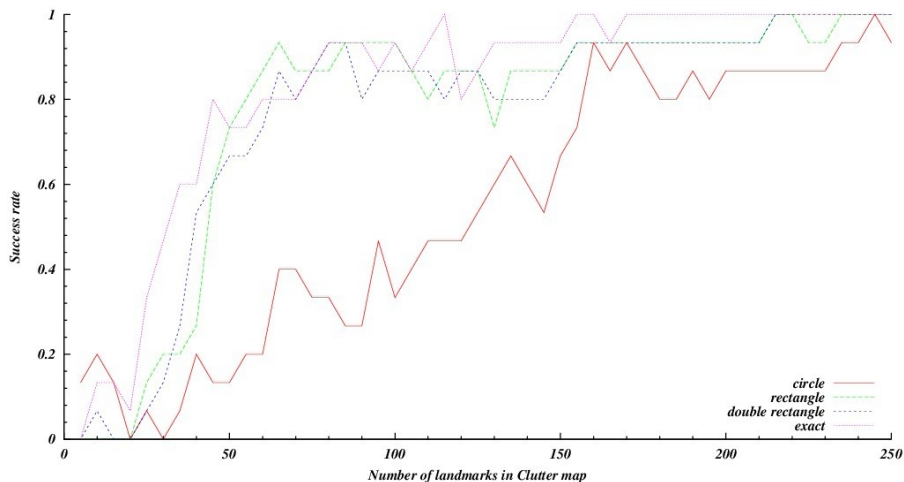


Computation and Approximation ratio in Obstacle-clutter Environment



Range Space	Time (s)	Approximation Ratio
\mathcal{R}_{disk}	0.162	0.155
\mathcal{R}_{rect}	0.441	0.632
$\mathcal{R}_{dblrect}$	1.122	0.691
\mathcal{I}	10.218	1.000

Comparison of Success Rate in Obstacle-clutter Environment



Outlines

- 1 Problem Statement
- 2 Range Space
 - Range Space Definition
 - Disk Range Space
 - Rectangle Range Space
- 3 Double-Rectangle Range Space
- 4 Experiments
- 5 Conclusions

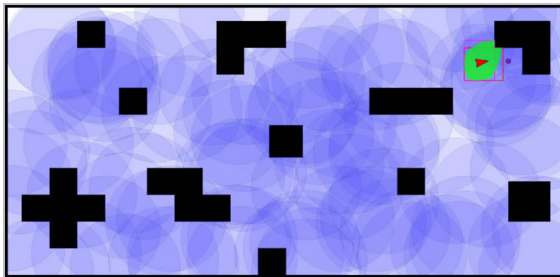
Conclusions and Future Work

• Conclusions:

- 1 Constrained geometric approximation is effective for representing a robot's uncertain information about the current state.
- 2 The form of double-rectangle is more accurate in approximating the non-convex *I-state*.
- 3 The robot can complete the navigation task using approximated I-state with low approximation accuracy.

• Future Work:

- 1 Additional range spaces, such as k -fold union of rectangles, will be considered to be used for high-accuracy approximation.
- 2 There may also be some advantage to algorithms that also generate provable **under-approximates** of the I-state.



song24@email.sc.edu