Distributed Coverage and Exploration in Unknown Non-Convex Environments

University of Pennsylvania

Subhrajit Bhattacharya | Nathan Michael | Vijay Kumar

GRASP Laboratory

Problem Definition

- N agents **explore** an unknown (or partially known) environment.
- Share exploration tasks in a fair manner.
- Achieve good **coverage** of the environment during and at the end of exploration.
- Do all these in a distributed fashion.

Related Work

1. Lloyd's Algorithm and its Continuous Time Version for Centroidal Voronoi Tessellation

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

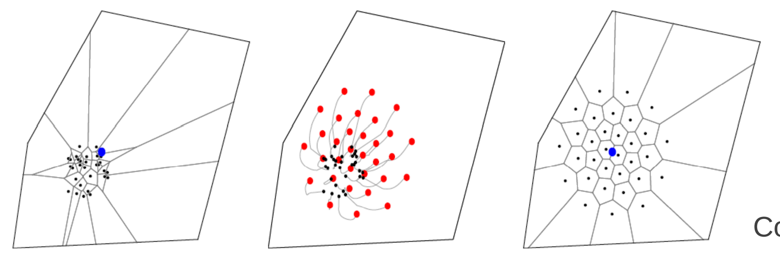
(Gradient descent approach for moving towards generalized centroid)

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

3. Other coverage algorithms

A. Breitenmoser and M. Schwager and J. C. Metzger and R. Siegwart and D. Rus, "Voronoi Coverage of Non-Convex Environments with a Group of Networked Robots", Proc. of the International Conference on Robotics and Automation (ICRA 10).

M. Pavone and A. Arsie and E. Frazzoli and F. Bullo, "Equitable Partitioning Policies for Robotic Networks", Proceedings of the International Conference on Robotics and Automation (ICRA 09).

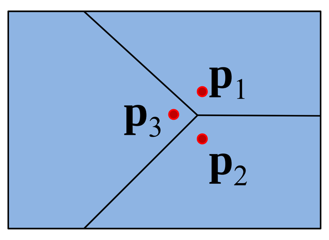


Cortes *et al.*

The Algorithm

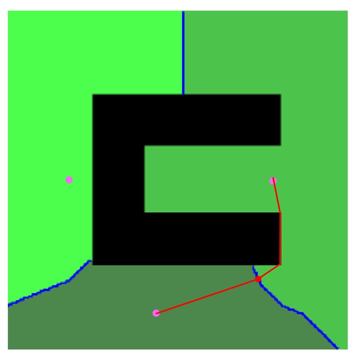
Assignment for Exploration and Coverage in *known environment*

Voronoi Tessellation:



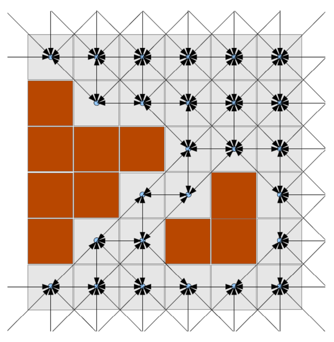
$$V_i(P) = \{\mathbf{q} \in \Omega \,|\, d(\mathbf{q}, \mathbf{p}_i) \le d(\mathbf{q}, \mathbf{p}_j), \forall j \neq i\}$$

Ω



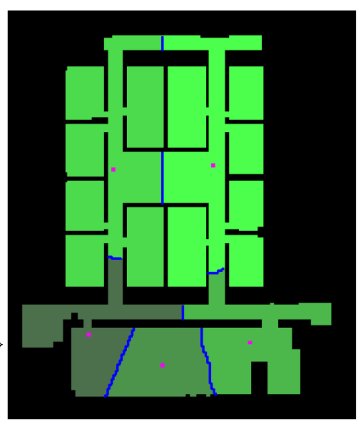
In non-convex environments:
d is the *Geodesic* Distance

Search-based algorithm for Computing *Geodesic Voronoi Tessellation*



1. Discretize the environment, and form a graph, $\mathcal{G}$
2. 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 $\mathbf{p}_i$



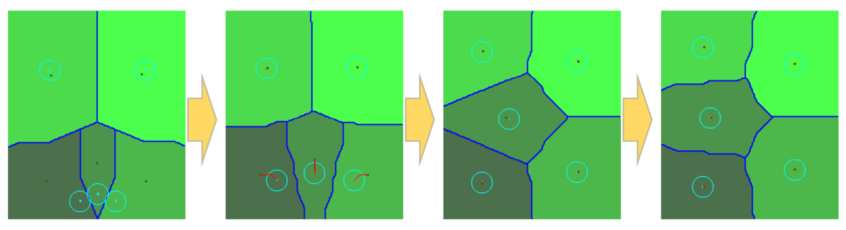
Geodesic Voronoi Tessellation:

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

Achieving good Coverage *(known environment)*

Continuous time Lloyd's Algorithm

Follow the weighted centroid of tessellation: $\mathbf{u}_i = k(\mathbf{C}_{V_i} - \mathbf{p}_i)$



where,

$$\mathbf{C}_{V_i} = \frac{\int_{V_i} \mathbf{q}\phi(\mathbf{q})d\mathbf{q}}{\int_{V_i} \phi(\mathbf{q})d\mathbf{q}},$$

Minimizes $\mathcal{H}(P, V(P)) = \sum_{i=1}^{n} \int_{V_i} f(d(\mathbf{q}, \mathbf{p}_i))\phi(\mathbf{q})d\mathbf{q}$

Converges to **Centroidal Voronoi Tessellation!**

Lloyd's Algorithm in non-convex environment:

- Follow gradient of $\mathcal{H}$, *OR*
- Follow Geodesic path to Generalized Centroid,

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

Direct computation is infeasible

Practical implementations of Lloyd's Algorithm in non-convex environment

Approach – I: Follow gradient of $\mathcal{H}$

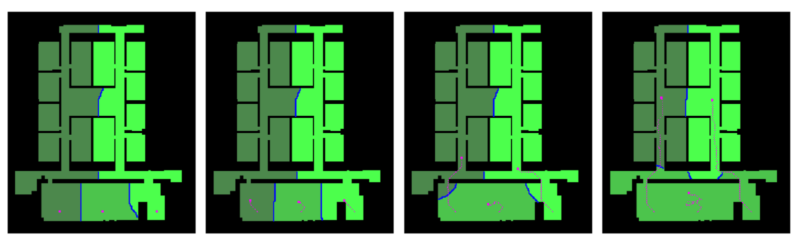
$$\frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} = \int_{V_i(P)} \frac{\partial}{\partial \mathbf{p}_i} f(d(\mathbf{q}, \mathbf{p}_i))\phi(\mathbf{q})d\mathbf{q}.$$

In a discretized setup

$$\mathbf{p}_i^{t+1} = \underset{\mathbf{p} \in \mathcal{N}(\mathbf{p}_i^t)}{\operatorname{argmin}} \int_{V_i(P^t)} f(d(\mathbf{q}, \mathbf{p}))\phi(\mathbf{q})d\mathbf{q}$$

$$\approxeq \underset{\mathbf{p} \in \mathcal{N}(\mathbf{p}_i^t)}{\operatorname{argmin}} \sum_{\mathbf{q} \in V_i(P^t)} f(d(\mathbf{q}, \mathbf{p}))\phi(\mathbf{q})\Delta\mathbf{q},$$

Gradient search in discretized environment:
- **Expensive!**
- Fast convergence – **not suited for exploration!**



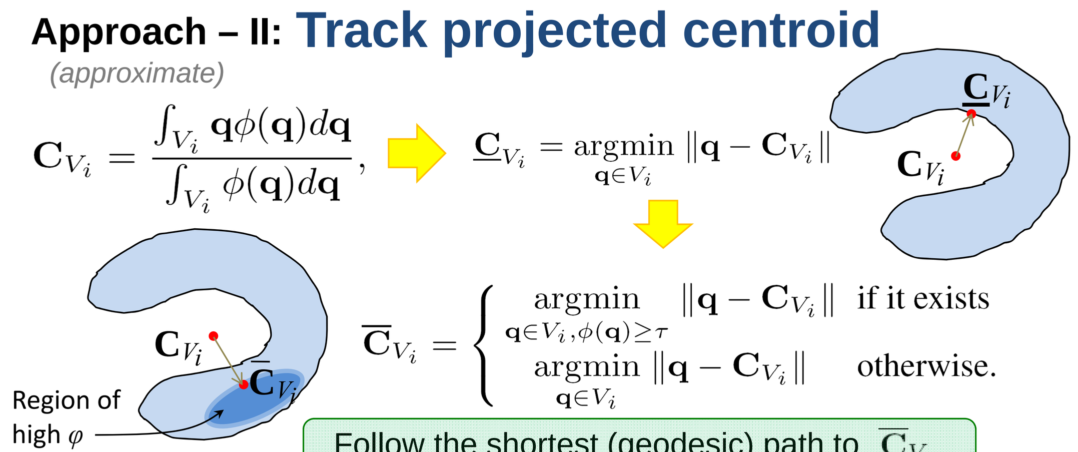
(a) $t = 0$ (b) $t = 10$ (c) $t = 50$ (d) $t = 150$ (converged)

Approach – II: Track projected centroid

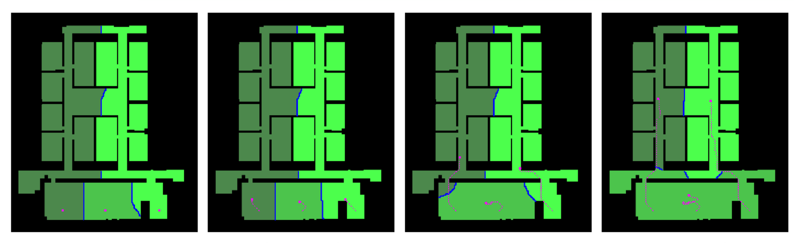
(approximate)

$$\mathbf{C}_{V_i} = \frac{\int_{V_i} \mathbf{q}\phi(\mathbf{q})d\mathbf{q}}{\int_{V_i} \phi(\mathbf{q})d\mathbf{q}} \Rightarrow \underline{\mathbf{C}}_{V_i} = \underset{\mathbf{q} \in V_i}{\operatorname{argmin}} \|\mathbf{q} - \mathbf{C}_{V_i}\|$$

$$\overline{\mathbf{C}}_{V_i} = \begin{cases} \underset{\mathbf{q} \in V_i, \phi(\mathbf{q}) \ge \tau}{\operatorname{argmin}} \|\mathbf{q} - \mathbf{C}_{V_i}\| & \text{if it exists} \\ \underset{\mathbf{q} \in V_i}{\operatorname{argmin}} \|\mathbf{q} - \mathbf{C}_{V_i}\| & \text{otherwise.} \end{cases}$$



Follow the shortest (geodesic) path to $\overline{\mathbf{C}}_{V_i}$
- **Easy to compute**
- **More suited for exploration tasks**



(e) $t = 0$ (f) $t = 10$ (g) $t = 50$ (h) $t = 150$ (converged)

Conjecture based on Experimental analysis:
Projection of centroid control method always converges and in cluttered environments differs little from the results obtained by the gradient search method.

Sensing and Exploration in unknown and partially known environments

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

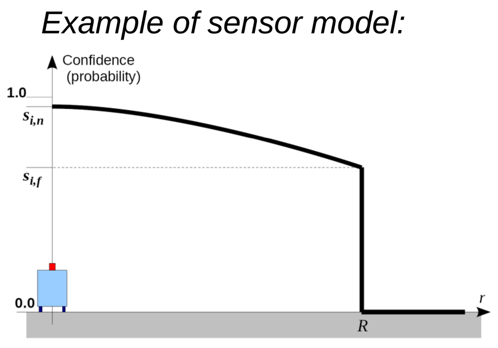
$$e(\mathbf{q}) = p(\mathbf{q})\ln(p(\mathbf{q})) + (1 - p(\mathbf{q}))\ln(1 - p(\mathbf{q}))$$

Entropy updated with time based on sensor model and sensor reading:

Sensor model (confidence as a function of distance):

$$s_i(r) = \begin{cases} s_{i,n} + \frac{r^2}{R_i^2}(s_{i,f} - s_{i,n}) & \text{if } r \le R_i \\ 0 & \text{otherwise,} \end{cases}$$

Example of sensor model:



Probability of occupancy:

$$u_i^t(\mathbf{q}) = z_i^t(\mathbf{q})\, s_i(\|\mathbf{q} - \mathbf{p}_i^t\|) + (1 - z_i^t(\mathbf{q}))(1 - s_i(\|\mathbf{q} - \mathbf{p}_i^t\|))$$

Sensor fusion model:

$$p^t(\mathbf{q}) = g^{-1}\left(\frac{\sum_{i,t'} g(u_i^{t'}(\mathbf{q}))}{\sum_{i,t'} 1}\right)$$

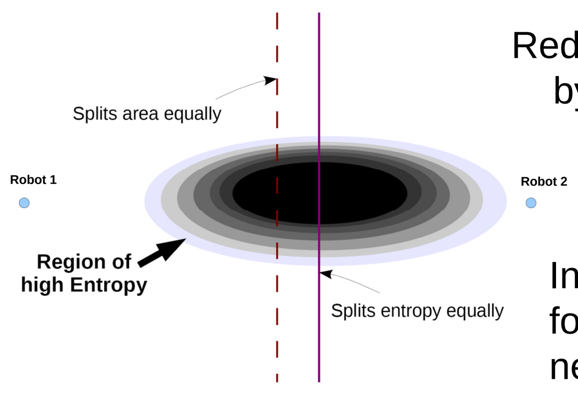
g is strictly increasing

Uses of Shannon Entropy for Exploration and Coverage

We exploit the flexibility in choosing d and φ in the Lloyd's Algorithm

Entropy-based metric for tessellation

We want equal splitting of entropy rather than area:



Redefine d – by weighting it by entropy:

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

In our search-based algorithm for Voronoi tesselation, all we need to do is a **weighted Dijkstra's search.**

Cost of an edge, ε, in the graph:

$$c(\varepsilon) = \frac{e(\mathbf{v}_s(\varepsilon)) + e(\mathbf{v}_t(\varepsilon))}{2} + \eta\|\mathbf{v}_s(\varepsilon) - \mathbf{v}_t(\varepsilon)\|_2$$

$\mathbf{v}_s(\varepsilon)$ ε $\mathbf{v}_t(\varepsilon)$

Entropy as weight/density function

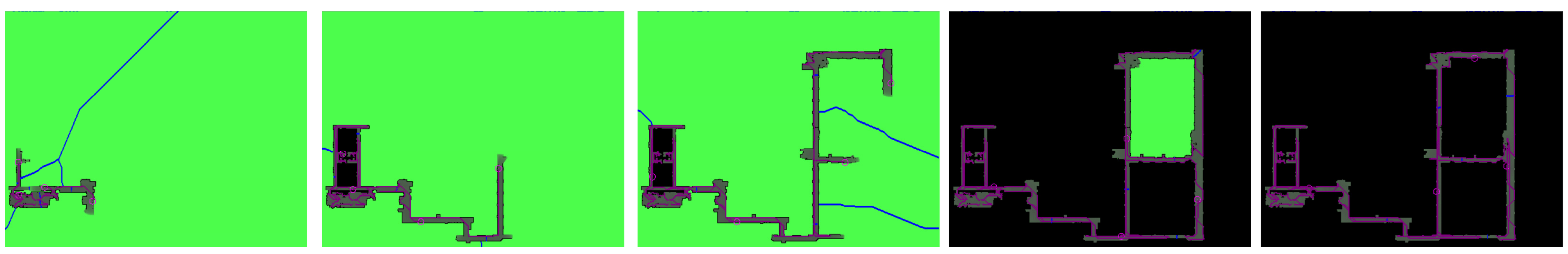
We identify the weight/density function, φ, with the Shannon Entropy, e.

With the *projection of centroid* control algorithm, this guarantees,

1. **Coverage** – entropy of every point in the environment will be reduced below τ.
2. **Convergence** *(conjecture)* – Since the projection of centroid algorithm is conjectured to converge, this too will converge.

Theoretical guarantee of convergence in star-shaped environments (*w.r.t.* projected centroid).

Simulation Result – Large Environment

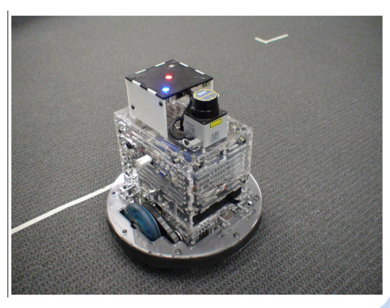


t = 200 t = 800 t = 1600 t = 2200 t = 2800 (convergence achieved)

Details:
- 4 robots exploring unknown environment
- 1000x783 discretized environment
- Each iteration (running on single processor) takes ~1.7s
- C++ implementation

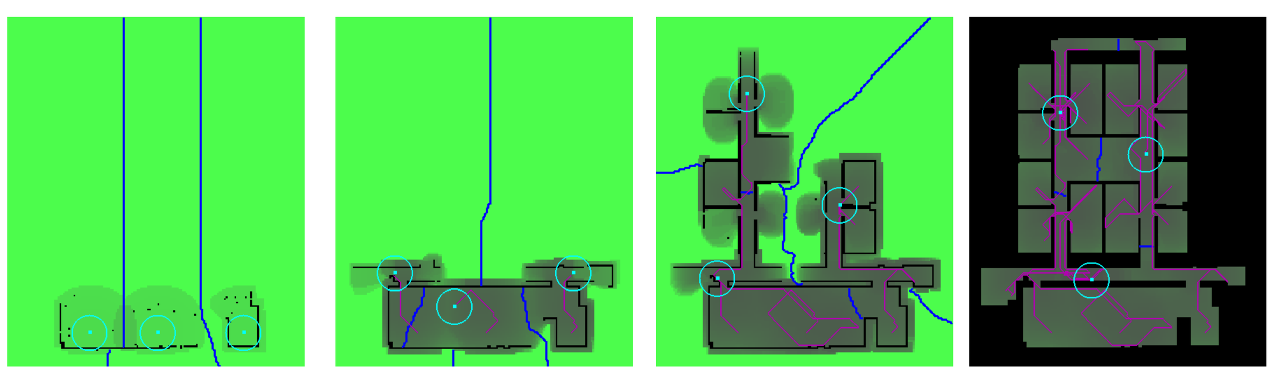
Experimental analysis

- ROS integration under progress.
- Being tested on *Scarab* mobile robot platforms.



Scarab mobile robot with on-board processors and Laser sensors

Simulation Result – Office Environment



$t = 0$ $t = 34$ $t = 170$ $t = 747$ *(convergence)*

Details:
- 3 robots exploring unknown environment
- 170x200 discretized environment
- Each iteration (running on single processor) takes ~0.3s
- C++ implementation

Conclusions:

- Developed an efficient way of computing Voronoi tessellations in non-convex environments using search-based algorithm.
- For unknown environments used sensor models and sensor data fusion to maintain and update entropy maps.
- Used Shannon entropy as a metric in computing Voronoi tessellations.
- Identified Shannon entropy with the density/weight function of Lloyd's Algorithm.

Acknowledgements

We gratefully acknowledge support from ONR grant no. N00014-09-1-1052, NSF grant no. IIS-0427313, ARO grant no. W911NF-05-1-0219, ONR grants no. N00014-07-1-0829 and N00014-08-1-0696, and ARL grant no. W911NF-08-2-0004.