Research Statement

Lawrence H. Erickson

What makes certain robotic problems hard? Sometimes, it is an inherent complexity present in the problem. Other times, an inadequate model of the robotic system might introduce complexity that could be avoided. A robot with one set of sensors may find performing a certain task very easy, while a different set of sensors could make the task intractable. How can one rigorously describe the relationship between the power of robotic sensors and the complexity of a task?

My goal is to understand the mathematics underlying robotics problems. This includes determining the relationship between the complexity of the robot's surroundings and the complexity of a desired task, determining algorithms that have provable performance guarantees, and categorizing robotic tasks by the sensor suites required to accomplish them. I am particularly interested in problems that have a combinatorial or geometric aspect to them. Some topics that I have worked on recently and intend to continue are listed below.

Chromatic Art Gallery Problem

Suppose that a robot is navigating in an environment populated by a set of landmarks that the robot detects visually. These landmarks are partially distinguishable to the robot's sensors and the robot's motion primitives are based on these classes. For example, if the classes represent colors, then the robot may have primitives like "drive toward the red landmark", "drive away from the blue landmark", "drive in a clockwise circle around the yellow landmark", etc. Two conditions are necessary (though depending on the motion primitives, perhaps not sufficient) for such a robot to navigate in the entirety of the environment.

- Each point in the environment must be visible from a landmark, as otherwise there may be places where the robot has no bearings on which to base its motion primitves.
- No point in the environment is visible from two landmarks that share a class.

The first condition is related to the *art gallery problem*, in which the goal is to place a set of point guards with omnidirectional vision into an input polygon such that each point of the input polygon is seen by a guard while using as few guards as possible. This problem was largely solved by Chvatal [3], who showed that $\lfloor n/3 \rfloor$ guards are always sufficient and sometimes necessary for an n-vertex polygon. The art gallery problem and its variants have significant applications in surveillance and navigation problems.

I have introduced a version of the art gallery problem that studies both conditions simultaneously. In the *chromatic art gallery problem*, like the original problem, we wish to place a set of point guards such that each point of an input polygon is seen by a guard. Two guards *conflict* if they see a common point. Conflicting guards cannot be assigned the same color. The goal is to place and color the point guards so that each point of the input polygon is seen by a guard, and to do this using as few colors as possible. Examples are shown in Figure 1. The chromatic art gallery problem relates the complexity of the environment (measured in the number of vertices, reflex vertices, or other parameter) to the discriminatory power of the sensors required to navigate in that environment (measured by the number of colors).

Results on bounds for the number of required colors in terms of the number of polygon vertices are found in [4], with results for the connected visibility graph variant in [5]. This problem is the subject of my dissertation, and additional results will appear there.

Intersection Graphs of Visibility Regions

An *intersection graph* is a graph in which each vertex is a set, with two vertices joined by an edge if the corresponding sets have a non-empty intersection. The *visibility region* of a point q in a polygon P is the

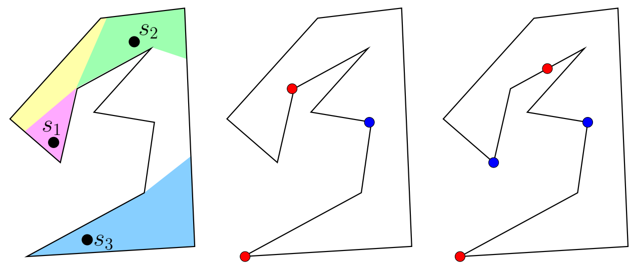

Figure 1: An instance of the minimum constraint removal problem. A path from the start (q_s) to the goal (q_g) that intersects the minimum number of obstacles (O_2 and O_5) is shown.

set of points $\{p \in P \mid \overline{pq} \in P\}$. For brevity purposes, I will abbreviate "intersection graph of visibility regions" as *IGVR*. Figure 2 shows some examples. What properties do IGVRs have? Additionally, how do restrictions on P translate into restrictions on the IGVRs possible in P?

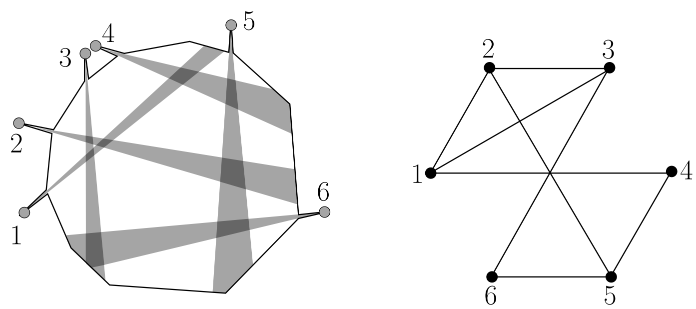

Figure 2: [left] The visibility regions of a set of points in a polygon. [right] The intersection graph of the visibility regions.

Partially, this is motivated by the chromatic art gallery work in the previous section. However, there are other reasons to examine IGVRs. A common task in surveillance problems is to determine the *vision graph* of a set of cameras [2], [8], [1]. In this vision graph, each vertex corresponds to a camera, and two vertices are joined by an edge if the corresponding cameras can see a common point. A similar structure, the *CN-complex* is a simplicial complex that also tracks higher order intersections [7]. Generally, the purpose of creating a vision graph is to simplify the task of tracking a target with the cameras. However, most of the existing work on vision graphs is algorithmic (given a set of camera viewpoints, how does one find landmarks that indicate common points of visibility?). There has been little study on the structural properties of these graphs. Knowledge of such properties would be useful as sanity checks for existing vision graph and CN-complex algorithms.

I (with John Kim) have found two families of graphs that cannot appear as induced subgraphs of IGVRs. We believe that these induced subgraph restrictions, along with a trivial inclusion of IGVRs as a subset of the circle-polygon graphs, fully characterize the IGVRs. Furthermore, we have shown that if P is monotone, then any IGVRs in P are the union of two interval graphs. Our results generalize intersection graphs of *k-link visibility regions*. The k-link visibility region of a point q in a polygon P is the set of points $\{p \in P \mid p$ and q can be joined by a path of k line segments that is internal to $P\}$. Future work includes proving that the restrictions we have found fully characterizes the IGVRs, determining whether or not the k parameter places any restriction on the possible IGVRs, and developing methods to test IGVR membership.

Due to its relevance to the chromatic art gallery problem, the results on this topic are due to appear in my dissertation.

Tracking in Graphs

Suppose that we wish to determine the distribution of people in a building through the use of sensors that detect when someone leaves or enters a room (for example, turnstiles). If the initial distribution of the people is known, then this is an easy task. If the turnstile that leads from room A to room B is activated, then the count of people in room A is decremented and the count of people in room B is incremented. If no such initial distribution is known, then this is a significantly more difficult task. However, certain sequences of sensor readings allow a distribution to be determined "from scratch" without requiring knowledge of the initial distribution.

The building can be represented by a directed graph. When a person (moving body) moves between rooms (graph vertices), the directed edge that was travelled through is returned as sensor data. If we have a probabilistic movement model for the moving bodies, then the number of sensor readings required to determine the number of moving bodies in each room is a random variable and we can determine properties like the expectation and variance, so long as certain minor structural conditions are met (for example, if the graph contains an isolated vertex, then any moving bodies initially located at that vertex are entirely undetectable, as they have no opportunity to move and return sensor data).

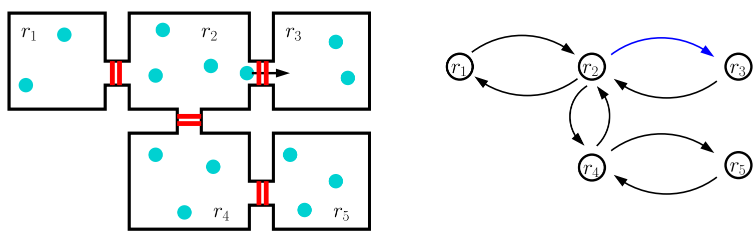

Figure 3: [left] A building with rooms separated by sensors. An agent is moving from room r_2 to room r_3. [right] The graph representation of this building, with the highlighted edge being returned as sensor data.

For the case where each moving body is equally likely to be the next one to change rooms, my co-authors and I determined bounds on the expected number of sensor readings required before a count of the number of people in each room is determined [6]. These bounds depend only on the number of moving bodies present and not on the structure of the underlying directed graph. Future work includes determining the influence of the structure of the graph on the expected number of required sensor readings (directed girth is expected to be an important parameter) and on developing bounds for other movement models.

Sensors as Oracles

An *oracle* is an abstract automaton that can compute a specific function in constant time, typically used in studies of computational complexity. A sensor in a robotic system can be viewed as an oracle that provides some sort of data about the system's state in constant time, and different sensor suites can dramatically alter the complexity of the robot's tasks. For example, a robot with a sufficiently detailed map of its environment and some way of visually sensing landmarks around it could determine its location in the environment, but the amount of computation required to do so would generally scale with the complexity of the environment. By contrast, a GPS system would tell the robot its location in constant time. Alternately, if a robot is provided with a map of the environment and its location, then it could determine its visibility region in linear time at best. If the robot is instead provided with a 360-degree laser rangefinder, then that sensor provides (a close approximation to) the robot's visibility region in constant time. The GPS sensor is an oracle for the robot's location, and the rangefinder is an oracle for the robot's visibility region.

What tasks become feasible when certain subtasks are solvable in constant time? How can the presence or absence of certain oracles (sensors) affect the amount of resources (such as fuel) that the robot needs to expend to accomplish a desired task? Which functions could plausibly be "oracled away" by a sensor (not necessarily a sensor that currently exists)? Answers to these questions can inform both sensor design and algorithm design.

References

[1] M. Antone and S. Teller, "Scalable extrinsic calibration of omni-directional image networks," *International Journal of Computer Vision*, vol. 49, no. 2-3, pp. 143–174, 2002.

[2] Z. Cheng, D. Devarajan, and R. J. Radke, "Determining vision graphs for distributed camera networks using feature digests," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007.

[3] V. Chvatal, "A combinatorial theorem in plane geometry," *Journal of Combinatorial Theory Series B*, vol. 18, no. 1, pp. 39–41, 1975.

[4] L. H. Erickson and S. M. LaValle, "An art gallery approach to ensuring that landmarks are distinguishable," in *Robotics: Science and Systems*, 2011.

[5] ——, "Navigation among visually connected sets of partially distinguishable landmarks," in *Proc. IEEE International Conference on Robotics and Automation*, 2012, pp. 4829–4835.

[6] L. Erickson, J. Yu, Y. Huang, and S. LaValle, "Counting moving bodies using sparse sensor beams," *Automation Science and Engineering, IEEE Transactions on*, vol. 10, no. 4, pp. 853–861, 2013.

[7] E. Lobaton, R. Vasudevan, R. Bajcsy, and S. Sastry, "A distributed topological camera network representation for tracking applications," *Image Processing, IEEE Transactions on*, vol. 19, no. 10, pp. 2516–2529, 2010.

[8] X. Zou, B. Bhanu, B. Song, and A. K. Roy-Chowdhury, "Determining topology in a distributed camera network," in *IEEE International Conference on Image Processing*, vol. 5, 2007, pp. 133–136.