Visualization of Sensor Network Coverage with Location Uncertainty

Tim Sodergren and Jessica Lohse

May 3, 2017

1 Motivation

Wireless Sensor Networks (WSN's) are becoming increasingly prevalent in the modern computing environment. We define a sensor network as consisting of a number of physical devices (nodes), spatially distributed within a geographic region of interest or domain, with some sensing capability out to a fixed distance. A key component to many types of sensor networks is coverage. In other words, are my sensors arrayed in such a way that every location within my domain is within the sensing range of at least one node? Coverage is important in many areas. For example, in mobile phone networks, one may want to identify the existence of spectrum violators or intruder detection in security applications.

If the physical location of each sensor is known, determination of coverage is a fairly straightforward geometric problem. If, on the other hand, there is uncertainty in the location of each sensor, determining coverage becomes more challenging. The solution becomes probabilistic and increases in size and complexity. This is a more realistic situation if, for example, we have lower quality equipment in our network. In this case we may only know the approximate location of each sensor within some probability distribution. It is the goal of this research to solve the coverage problem given these conditions. We will consider a simple case of location uncertainty to be described in more detail below.

2 Problem setup

We will consider a simple model of location uncertainty referred to as "indecisive data" For a network consisting of n sensors, each sensor will have a finite set of k possible locations, of equal likelihood, $Prob = 1/k$. As an additional constraint, each set of possible locations for each sensor are constrained within a disk of radius ϵ, assumed to be less than the coverage radius, r_c. An example of a simple network with $n = 4$ and $k = 8$ is shown in figure 1.

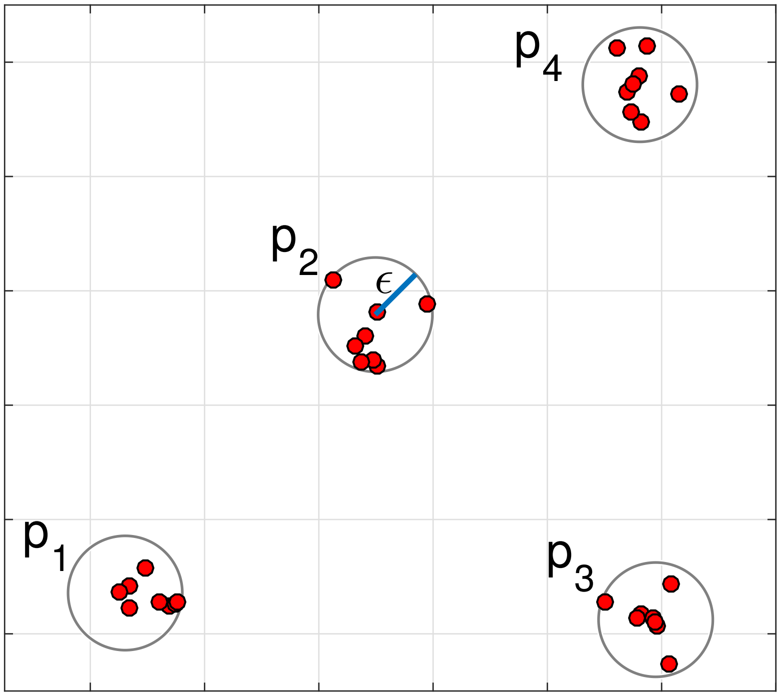

Figure 1: An example of a sensor network with uncertain locations. Potential sensor locations are shown in red with each having an equal probability, $1/k$

Sensors are arrayed in a domain, $D \subset \mathbb{R}^2$, in euclidean space. So for a set of sensors, P:

$$P = \{p_1, p_2, ..., p_n\}$$

Each uncertain point, P_i has exactly k possible locations:

$$p_i \in \{p_{i1}, p_{i2}, ..., p_{ik}\}$$

We define an "instance" as one possible realization of the network where the location of each sensor is fixed at one of it's possible values (figure). Then there are K^n possible instances.

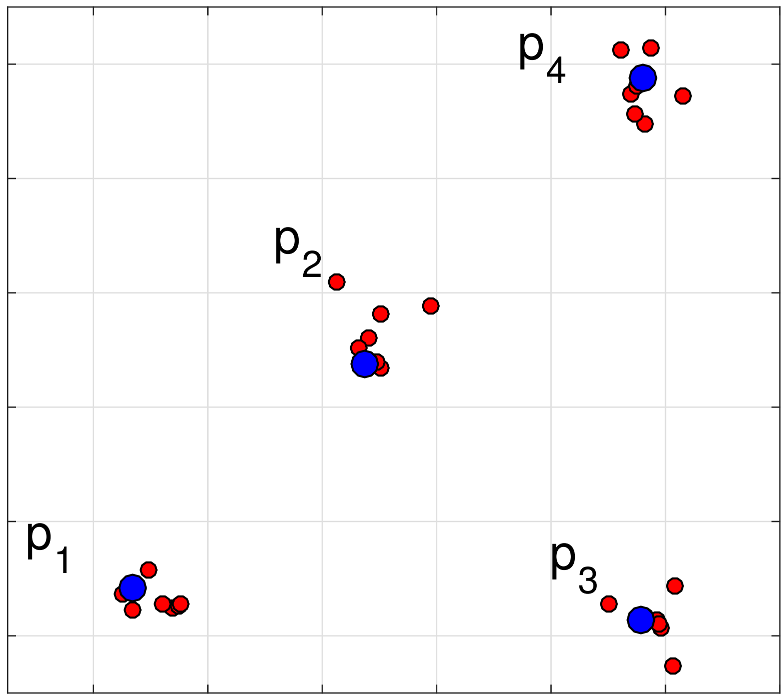

Figure 2: One possible instance from figure 1 with sensor locations highlighted in blue.

2.1 Objectives

Our objective is to use an abstract representation of sensor network topology via simplicial complexes, namely the Čech and Vietoris-Rips complexes. We expand the definition of these objects to include probabilities for each simplex. We then propose a graphical scheme for these probabilistic complexes, using sound design principles, that allows for effective visualization and analysis.

3 Related Work

The primary work utilizing simplicial complexes to compute sensor network coverage was done by de Silva and Ghrist [1]. They describe a method for guaranteeing coverage in a wireless sensor network by computing the persistent homology of the Vietoris-Rips complex. Shulz, et al present a method for visualizing uncertain networks as graphs [2] by employing a Monte Carlo process to determine location probabilities and then display as point clouds. They use a process of edge-bundling and splatting for displaying the graphs. Loffler et. all discuss various methods of dealing with uncertain data [3]. In particular, we employ their techniques for calculating the minimum enclosing disk on so-called "indecisive" data sets. Finally, the vizualization tool was adapted from previous work by the author [4]. This tool considered only a static network with fixed, known node locations.

4 Technical Details

4.1 Generating Simplicial Complexes with Probability

We determine coverage by evaluating the connectivity of each network node to other nodes via simplicial complex. For this computation it is necessary to evaluate the pairwise distances between all sensors in the network. An edge, E, exists between two sensors if their respective coverage areas overlap:

$$if \quad d(p_i, P_j) \leq r_c \quad \exists E_{i,j}$$

For any pair of vertices, the edge probability can be determined recursively by examining each possible instance, summing the number of times the above criteria is met, and dividing by the number of possible instances. For each node pair there are k^2 possible combinations and there are $\binom{n}{2} = \frac{n!}{2(n-2)!}$ possible node pairs. This computation time is improved by only considering node pairs for which the radius of coverage falls within the annulus of uncertainty, defined as the space between the smallest and largest minimum enclosing disk described by the radius of uncertainty, ϵ (figure 3). For all other possible node pairs we do not need to iterate over every possible instance. If the coverage radius exceeds that of the outer circle, $Prob(edge) = 1$. If the coverage radius is less than that of the inner circle, $Prob(edge) = 0$. We also save computation by comparing the squared euclidean distance to the squared coverage diameter. Each instance that produces an edge is stored for later computation.

Once all of the edges have been determined we compute the face probability for the Vietoris-Rips Complex. For every possible combination of three nodes, we lookup the probability of each of the three edges computed in the previous step. If the probability of all edges is 1, the probability of that face is also 1. If the probability of any of the edges is 0, then the probability of the face is zero. If the probability of 2 of the edges is 1 then the probability of a face is equal to the probability of the third edge. For all other cases we have to iterate over all possible edges for the three nodes. For each instance we add a face if each edge exists as determined in the previous step. Again, we store all possible triangles for use in the next step. The probability of each face is the number of triangles that meet this criteria divided by the number of possible triangles per node triple, k^3.

For the Čech complex, a triangle, or face, F, exists between 3 or more sensors that

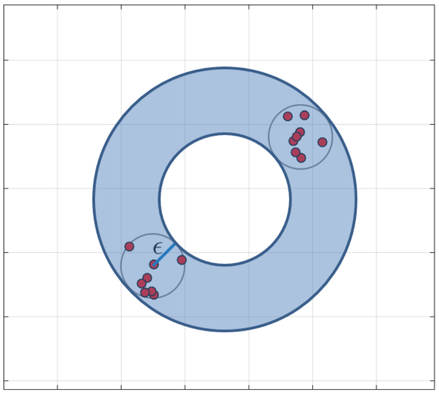

Figure 3: Annulus of uncertainty for two nodes. One need only consider coverage radii between the inner and outer rings for computation of probability.

share a common intersection. Looking at the instance from figure 2, we see five edges, $E = \{e_{1,2}, e_{2,3}, e_{1,2}, e_{2,4}, e_{3,4}\}$ and one face, $F = \{f_{1,2,3}\}$.

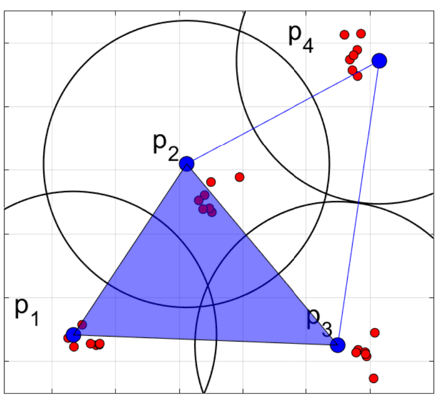

Figure 4: For the instance from figure 2, there is just one face between points 1,2, and 3. Edges exist between points 2,3 and 4 but they coverage disks do not have a common point of intersection.

To compute the Čech complex we iterate through the possible triangles for each node triple stored from the Vietoris-Rips computation. We calculate the circumcircle (i.e. minimum enclosing disk) radius There is a deterministic solution for identifying the probability of existence of an edge or face. An edge occurs if the euclidean distance between 2 sensors is $\leq 2r$. Another way to think of this is to consider the minimum enclosing disk of any 2 sensors as a the minimum circle enclosing the points. For an edge or line, the radius of the minimum enclosing ball is simply half of the euclidean distance between each sensor. For faces, we can determine the minimum enclosing disk of any three points. If the radius of the disk is less than or equal to the coverage radius, then we add a face.

4.2 Visualization Design

Since we are using D3.js as our visualization tool, we had the ability to very easily control what our visualization would look like and how to designate different meanings of objects. Using opacity, width, and color gradients we were able to show the uncertainties of edges and faces and clearly show the coverage of each node.

When our page is loaded, a user will see 20 nodes and their corresponding radii. A simplified version of this is shown below. Each node is represented by the opaque purple circle. This circle represents the radius of uncertainty, and thus all possible locations, represented by black circles, are found within this radius. The darker but slightly transparent circle surrounding these anchors represents the coverage radius. This is the guaranteed coverage of the node based on the central location of the radius of uncertainty. The last circle surrounding a node adds the radius of uncertainty to the coverage radius. This designates that there is a probability of coverage in this area, but that it is not guaranteed. Changing the radius of either the coverage or the radius of uncertainty updates those circles.

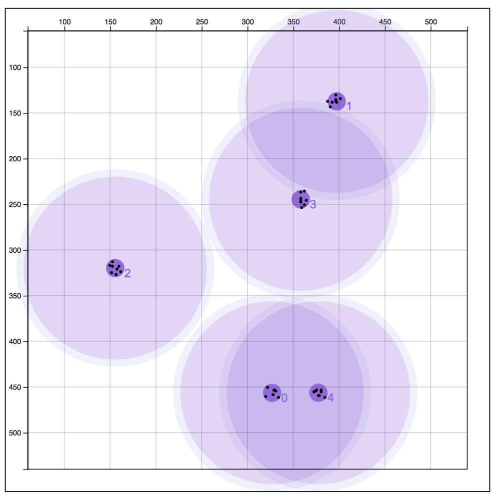

Figure 5: Nodes and Their Corresponding Radii.

By using opacity on the coverage radius and the radius of uncertainty plus the coverage radius, it becomes very clear when areas overlap. By looking at the example above, the areas where circles overlap (between nodes 1 and 3 and between nodes 0 and 4) are darker than the areas that do not overlap. As more circles overlap, that area becomes darker and darker, indicating the increased probability that an edge or face could occur in that area.

The goal of this project was to show the uncertainty of complexes based on these uncertain node locations. The important part of this visualization is to show those probabilities for faces and edges of a complex. To do this, we use double encoding of opacity and stroke width for edges. The more probable that an edge exists, the more opaque and thicker it is. A color gradient scale for faces, so that lighter green designates that a face is less likely to occur than a darker green triangle. Examples of these can be seen below, along with the legend that is included on our site.

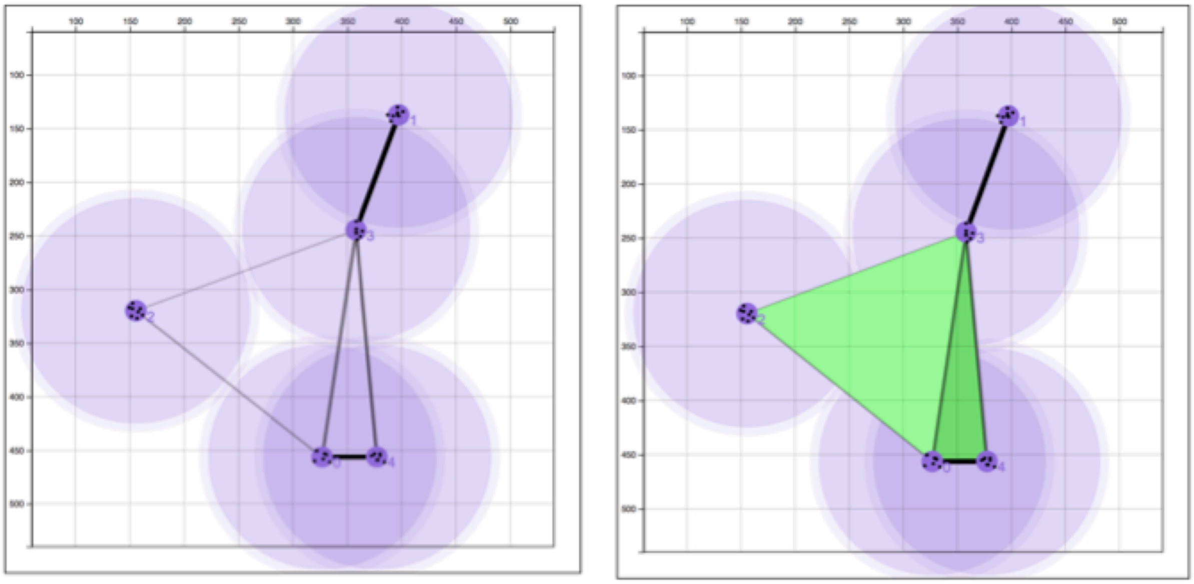

Figure 6: Edges with Double Encoding and Faces with Color Gradient Scale.

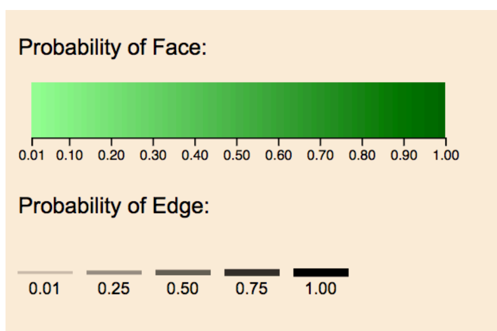

Figure 7: Scaling Gradient Legend for Edges and Faces.

We also give users the ability to see all possible instances of edges that could occur. An example of this is seen below. We do not use the same double encoding on these edges as we did for the representative edges. All instances have the same width and opacity. By doing this, it is easy to see each instance without others overwhelming them. Also, because the opacity is the same for each and the edges may overlap, the additive property of these edges makes darker edges between nodes based on the number of instances. In a way, this ends up creating the double encoding used for representative edges.

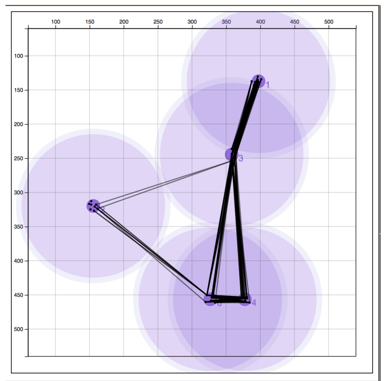

Figure 8: All Possible Edges of the Network.

Below is an example of the data that is generated on page load, instead of the simplified version above. We sort the edges and faces based on their probability. This ensures that as radii are increased and more faces are added, the newer, lighter faces are not showing up over faces that have a lower probability of occurring.

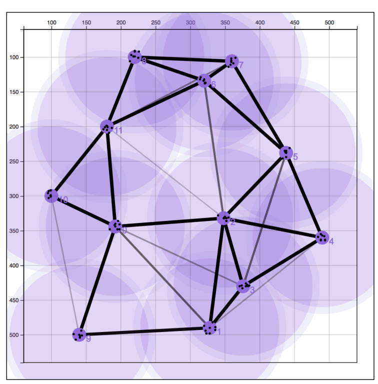

Figure 9: Edges in More Complicated Arrangement.

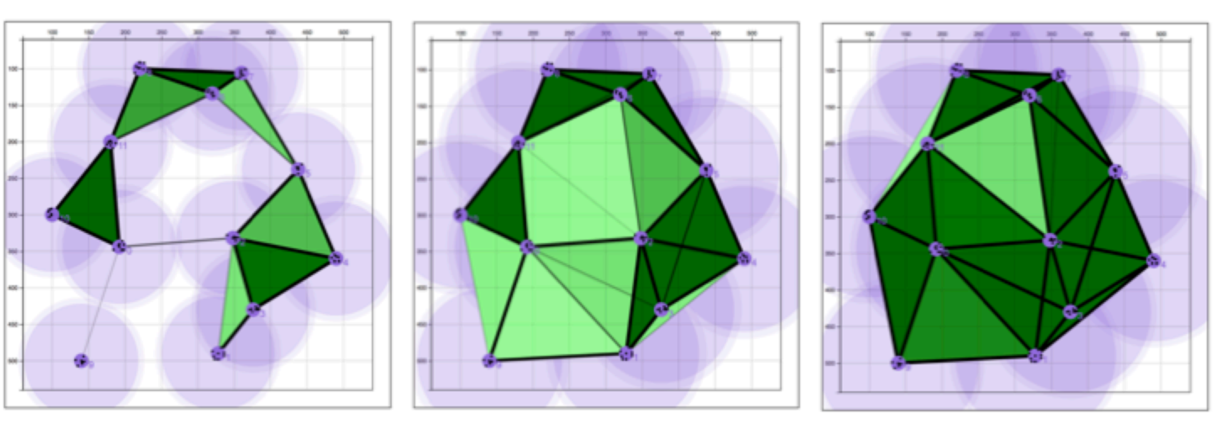

Figure 10: As Radii Increase (From Left to Right), Sorting Ensures Higher Probability Faces Stay Above Lower Probability Faces.

5 Results

The code for our algorithm and tool can be found at $https://github.com/jalohse/sensor\_network\_topolog$ and can be found online at $http://www.sci.utah.edu/\ tsodergren/prob_net_vis/$. Users have the ability to import data of their own or manipulate the data we give to achieve the desired network. The figure below shows the user interface.

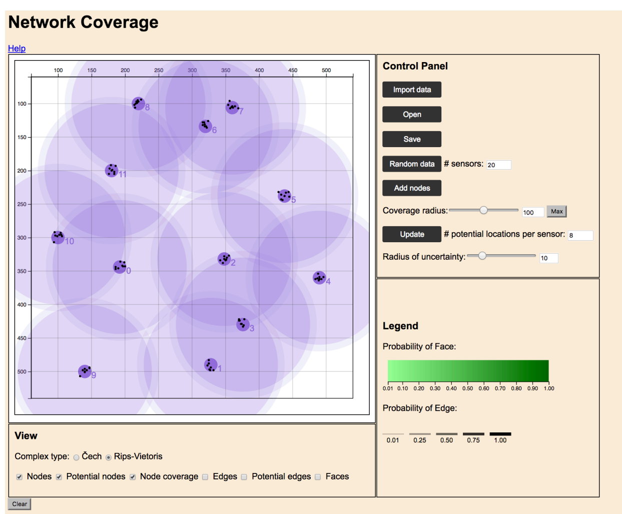

Figure 11: The User Interface for Our Visualization Tool

To the right of the network graph, users have the ability to manipulate the parameters of the network. They can adjust the number of sensors and randomly generate their position. It is also possible for users to add sensors of their own in locations they desire. The coverage radius of each sensor can be changed by the slider or by inputting a certain number. The maximum coverage radius possible is 200 by default, but it can be changed by the user.

The number of potential locations per sensor can also be changed. Since we are focusing on uncertain locations, we do not give users the ability to add locations themselves, but only to generate those locations randomly. The radius of uncertainty can also be changed, with a maximum value of 50. Increasing the radius of uncertainty does not generate new sensor locations, since all existing locations still lie in that radius. Lowering the radius will generate new potential locations.

Below the network graph is where users can change what the graph shows the complex type. Changing the complex type could change the complexes being drawn. Toggling the different types of complexes will not change the graph themselves.

By using the scroll wheel on a mouse and holding down the "z" key, users can zoom in and out on the graph. This allows them to create larger networks or to see small details. Users can also move nodes by clicking on a node and then dragging. Once a node has been selected, it is also possible to delete that node by pressing the "delete" key. Any change in the nodes will cause the complex to be recomputed and rendered again.

6 Conclusions

As our technologies continue to advance, Wireless Sensor Networks will become important to our daily lives. Whether it is attempting to find a cell phone in a certain area or sending data between drones, we will have to start thinking about connectivity between sensors more seriously. This is especially true if the location of the sensor is unknown. Without having definite locations it could be impossible for data to be retrieved or sent.

To solve this problem, we used topological data analysis to compute probabilistic simplical complexes. Once these complexes have been computed, we show users the network, along with the probabilities of 1- and 2- dimensional complexes. This data can be input into coverage models and used with persistent homology to determine which coverage radii and radii of uncertainty are needed based on a certain standard.

By producing a visualization tool, we also have given users the ability to tweak radii and the number of sample sensors as they see fit. Not only does this make the tool very helpful in an educational environment, but it could also be used to prototype networks. This tool takes a theoretical concept and changes it into something that users can see and understand.

7 Contributions

Given a Wireless Sensor Network, we implemented an algorithm that allows for uncertain locations of sensors. This allows us to determine "probabilistic simplicial complexes," or complexes that are tied to the probability of that complex existing. These complexes can then be used as inputs into other systems to determine persistent homology.

Along with this algorithm, we also developed a tool that not only allows the creation of an uncertain Wireless Sensor Network but also the ability to manipulate parameters of the network. The tool serves as an educational tool for users, whether they are learning about simplicial complexes and filtration or they want to know how the placement of sensors effects their network.

7.1 Individual Contributions

For this project, we were able to split the project into algorithm development and visualization design. Tim took the algorithm development which also included computation of simplicial complexes and determining probabilities of those complexes. Jessica focused on the visualization and the ways to most effectively render the complexes that were being generated with their

8 Future Work

Given the recursive nature of the algorithm we employ, there is a limited degree of scalability of our visualization tool. There are several algorithmic improvements that can be made:

- Computation of the probability of each potential node pair can be reduced by employing a "greedy permutation" approach which may reduce computation to $O(k \log k)$ and to $O(k^2 \log k)$ for each triple.
- Computation time can be further reduced by segmenting our domain into rectangles with dimensions equal to the coverage radius. Then we need only compare nodes in adjacent segments.

In addition to these improvements, the long term objective is to take these probabilistic complexes and use them for more advanced topological data analysis. In particular, we are interested in looking at the persistent homology of these data sets and determining the probability of existence of cycles or holes in coverage. Scalability may also be enhanced by moving to a distributed computing environment.

References

[1] V. De Silva and R. Ghrist. Homological sensor networks. *Notices Amer. Math. Soc*, pages 10–17, 2007.

[2] C. Schulz, A. Nocaj, J. Goertler, O. Deussen, U. Brandes, and D. Weiskopf. Probabilistic graph layout for uncertain network visualization. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):531–540, Jan 2017.

[3] Allan Jørgensen, Maarten Löffler, and Jeff M. Phillips. Geometric computations on indecisive points. In *Proceedings of the 12th International Conference on Algorithms and Data Structures*, WADS'11, pages 536–547, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] Tim Sodergren. *Network Coverage*, 2016. `http://www.sci.utah.edu/~tsodergren/network_vis/`.