Scattered Data Interpolation in Complex Domains

Oliver Kreylos

kreylos@cs.ucdavis.edu

Center for Image Processing and Integrated Computing (CIPIC)

Department of Computer Science

University of California, Davis

Outline

- Sensor networks as scattered data problem
- Overview of existing scattered data interpolation methods
- Representation of complex domains
- A distance metric for complex domains
- Visualizing sensor network data

Sensor Networks as Scattered Data Problem

Scattered Data Problem

- Given: A vector $S = ((p_1, v_1), \ldots, (p_N, v_N))$ of N samples
  - p_i: Sample position (*site*) in some domain D
  - v_i: Sample value in some range R
- Samples are assumed to be measurements of some (unknown) function $F \colon D \longrightarrow R$, i. e., $\forall i = 1, \ldots, N : F(p_i) = v_i$
- Scattered data problem: Reconstruction of F based only on sample vector S

Interpreting Sensor Network Data

- p_i are sensor positions, v_i are sensor measurements
- For time-varying sensor data: S is time-varying, but the p_i stay fixed
- For mobile sensors: The p_i vary as well
- Analysis of sensor data requires reconstructing the measured field first

Interpreting Sensor Network Data

- Leads to three levels of scattered data problems
  - Static sites and static values
  - Static sites and dynamic values
  - Dynamic sites and dynamic values
- Additional fourth level: Optimizing sensor placement based on reconstruction result

Existing Scattered Data Methods

Scattered Data Interpolation

- Defined by distance metric $d \colon D \times D \to \mathbf{R}^+$
- Several classifications possible
  - Direct (gridless) methods: Rely only on d
  - Indirect (grid-based) methods: Construct grid on D (based on d), and use grid to reconstruct F
  - Global methods: Use all samples to evaluate $F(p)$
  - Local methods: Use only samples “close” to p to evaluate $F(p)$

Example Data Set

- Interpret color image as trivariate scattered data by sampling at 1600 “random” positions
- Source image: Digital photo (400×300 pixel)

Shepard's Global Method

- Shepard's formula:

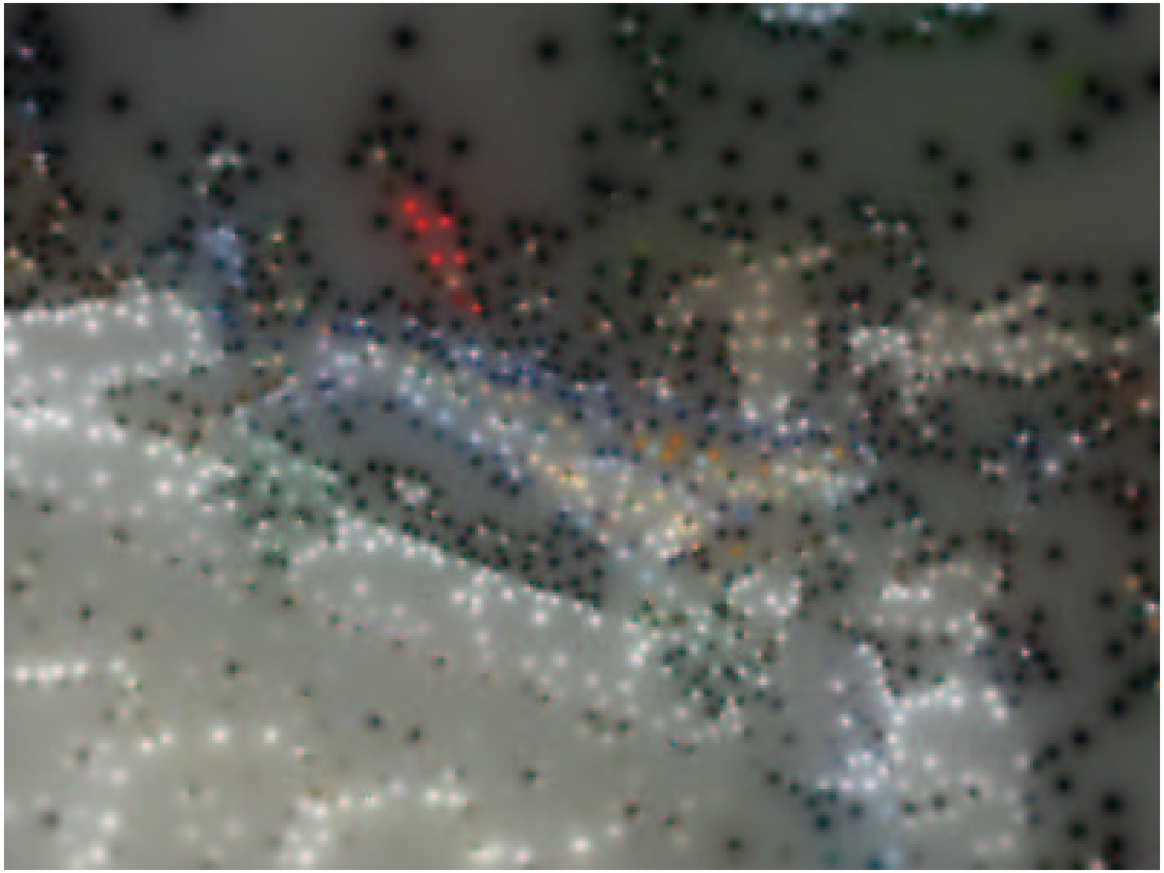
$$F(p) := \frac{\displaystyle\sum_{i=1}^{N} \frac{1}{d^2(p, p_i)} v_i}{\displaystyle\sum_{i=1}^{N} \frac{1}{d^2(p, p_i)}}$$

Shepard's Global Method

- Shepard's formula:

$$F(p) := \begin{cases} v_i & \text{if } \exists i : d(p_i, p) = 0 \\ \dfrac{\displaystyle\sum_{i=1}^{N} \frac{1}{d^2(p, p_i)} v_i}{\displaystyle\sum_{i=1}^{N} \frac{1}{d^2(p, p_i)}} & \text{otherwise} \end{cases}$$

Shepard’s Global Method

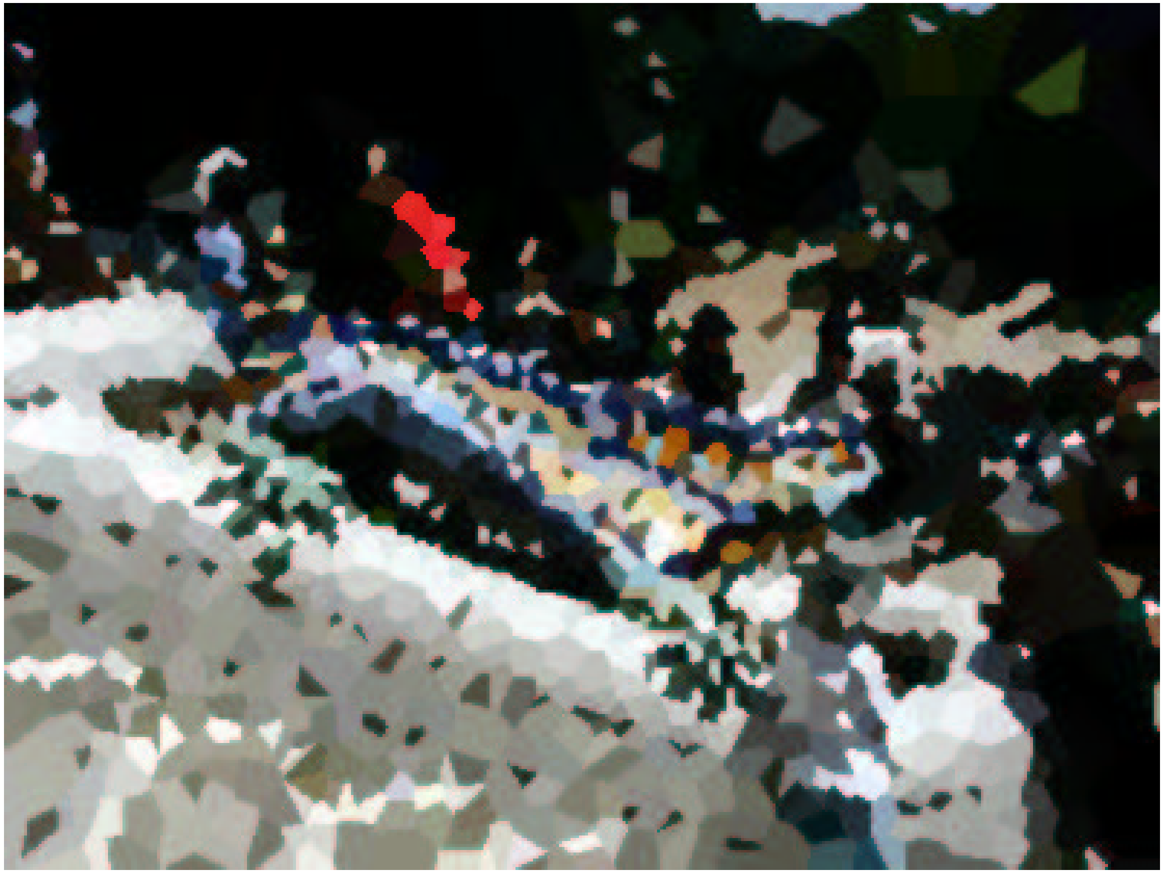


Shepard's Local Method

- Find k nearest neighbours of point p in S
- Evaluate Shepard's formula for those k points

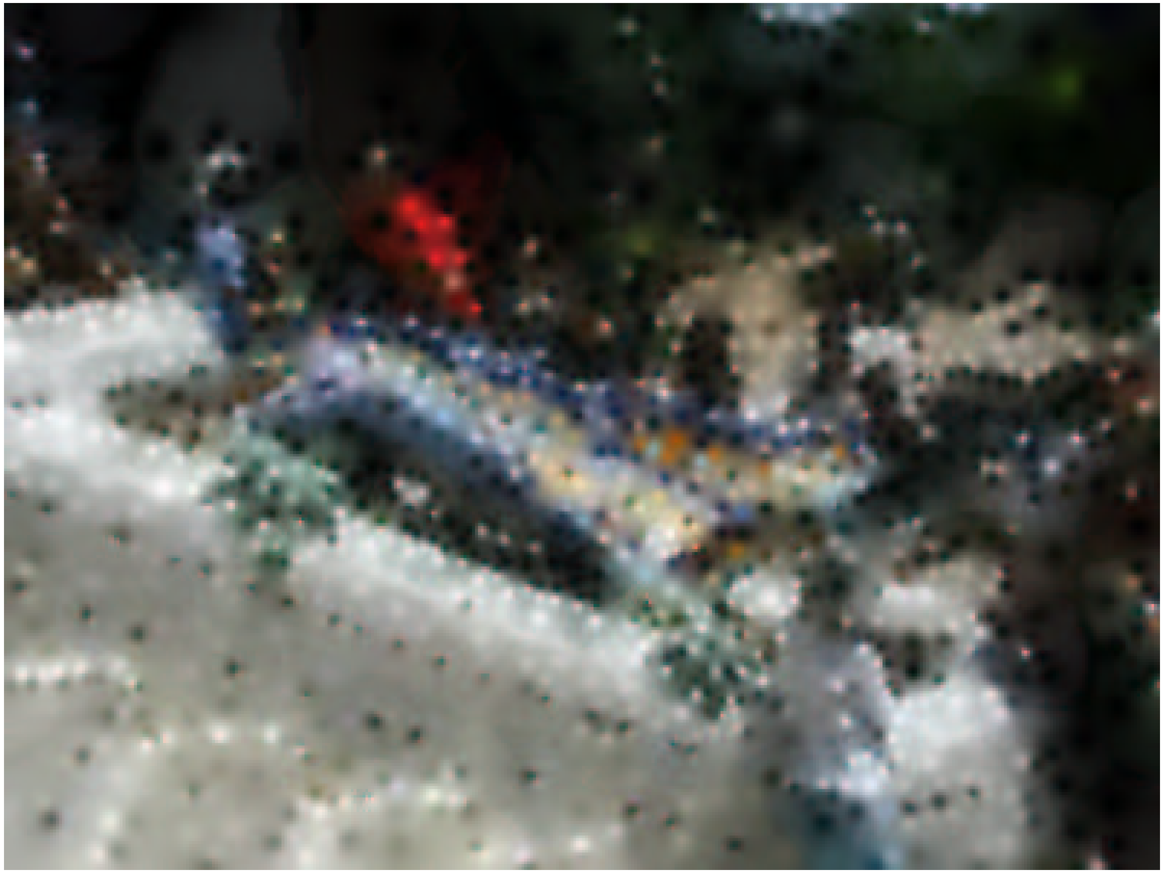
Shepard's Local Method

$k = 1$:



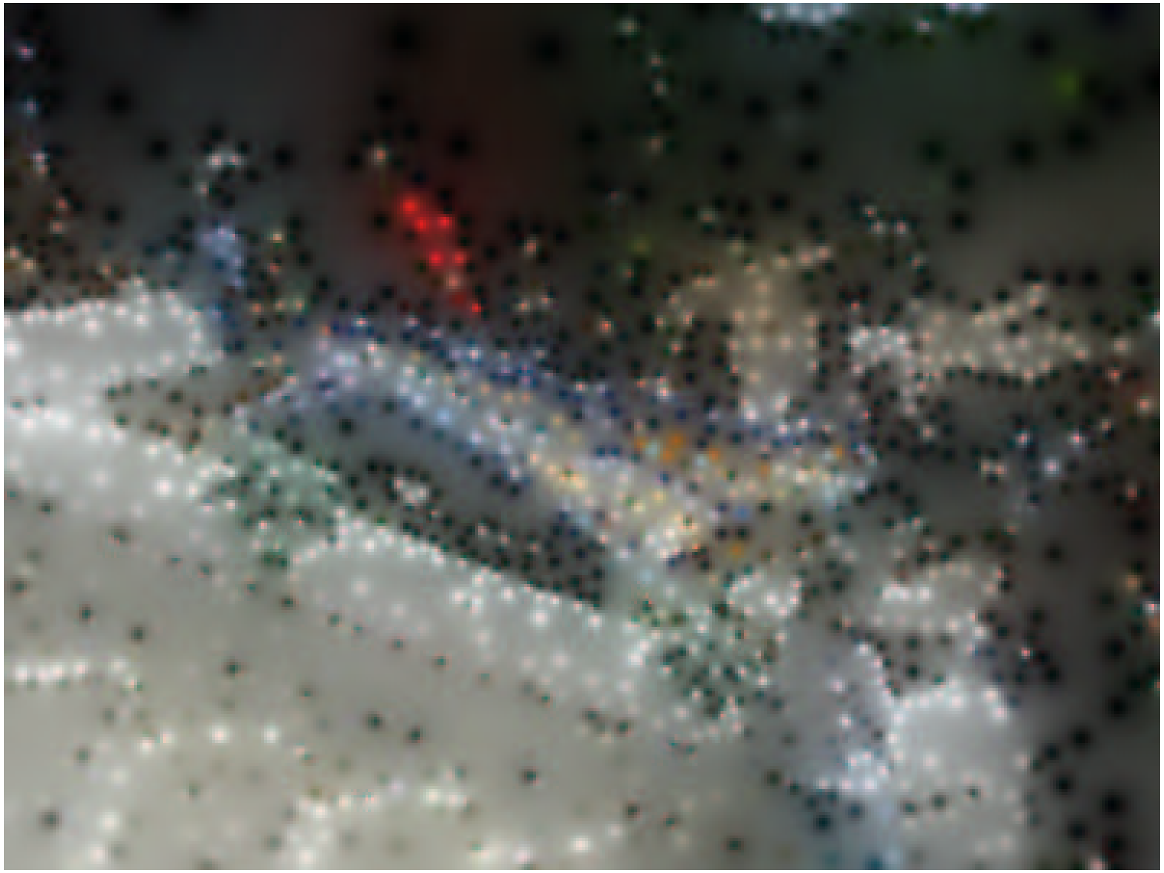
Shepard's Local Method

$k = 10$:



Shepard's Local Method

$k = 100$:



Hardy's Global Method

- Construct $N \times N$-matrix $H := ((h_{ij}))$,
  $h_{ij} := (d^2(p_i, p_j) + R^2)^\alpha$ for parameters R^2, α
- Solve linear system
  $(v_1, \ldots, v_N) = H \cdot (c_1, \ldots, c_N)$
- Hardy's formula:

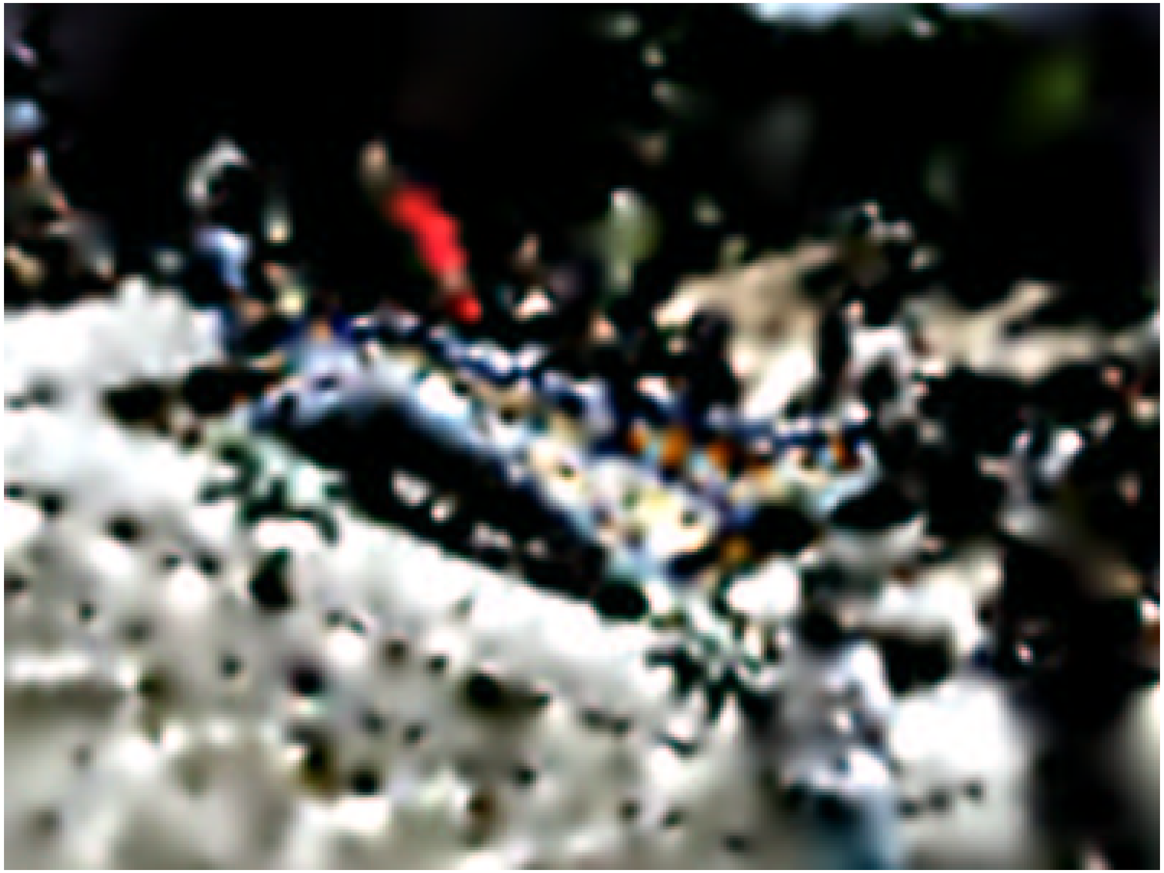
$$F(p) := \sum_{i=1}^{N} (d^2(p, p_i) + R^2)^\alpha c_i$$

Hardy's Global Method

$R^2 = 1.0, \alpha = 0.5$:

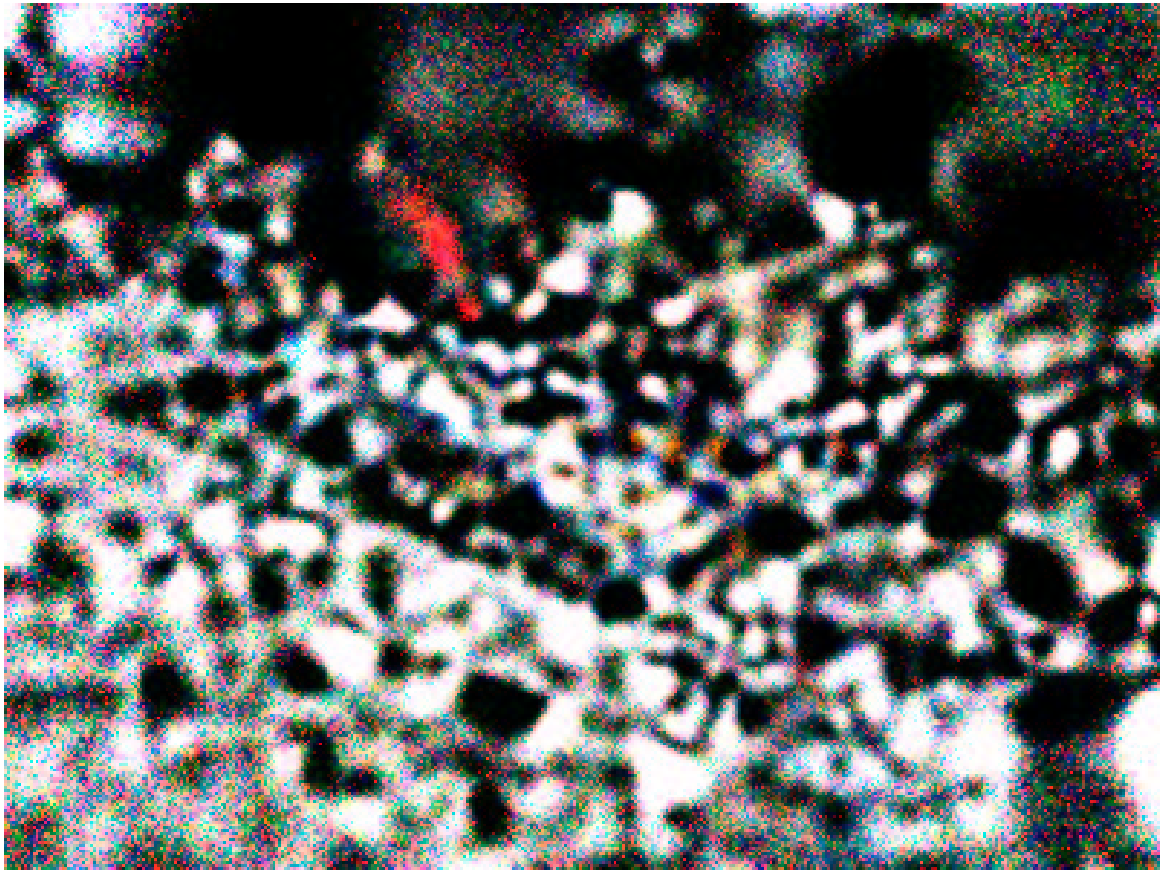
Hardy's Global Method

$R^2 = 10.0, \alpha = 0.5$:



Hardy's Global Method

$R^2 = 100.0, \alpha = 0.5$:

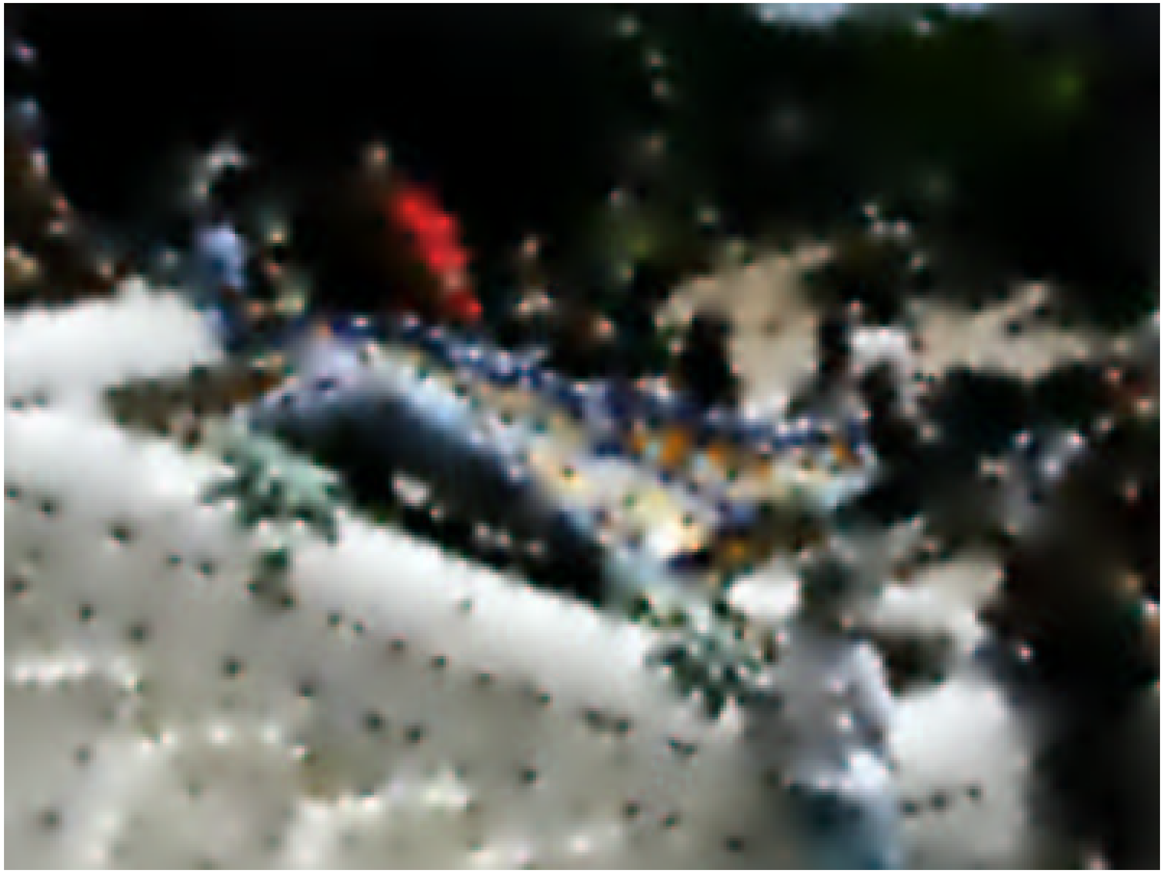


Hardy’s Global Method

$R^2 = 1.0, \alpha = 0.5$:

Hardy's Global Method

$R^2 = 1.0, \alpha = 0.1$:



Hardy's Local Method

- Find k nearest neighbours of point p in S
- Construct matrix H for those k points
- Solve linear system for $(c_1, \ldots, c_k)$
- Evaluate Hardy's formula for those k coefficients

Hardy's Local Method

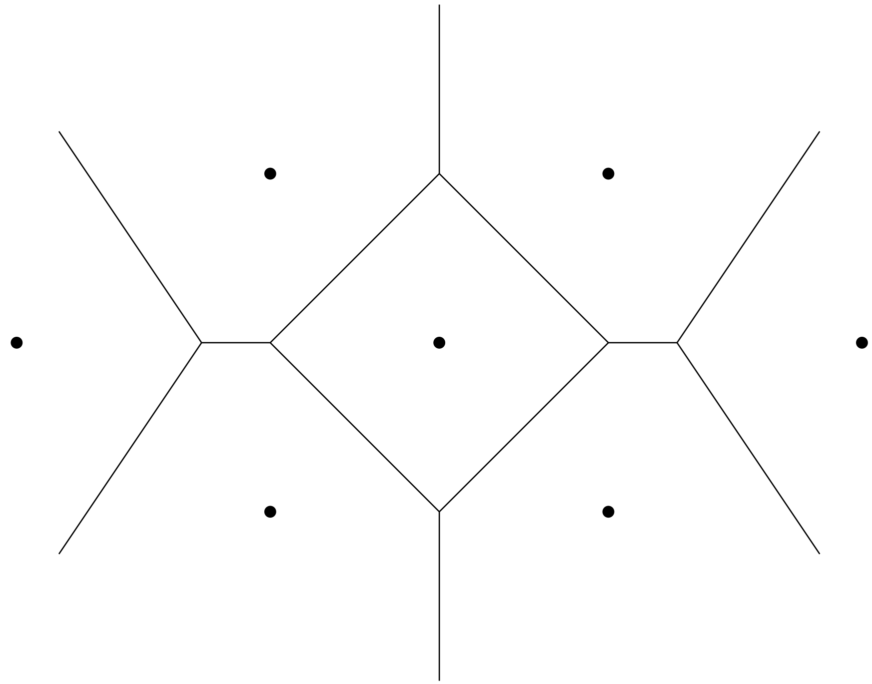
$R^2 = 1.0, \alpha = 0.5, k = 10$:

Hardy's Local Method

$R^2 = 1.0, \alpha = 0.5, k = 100$:

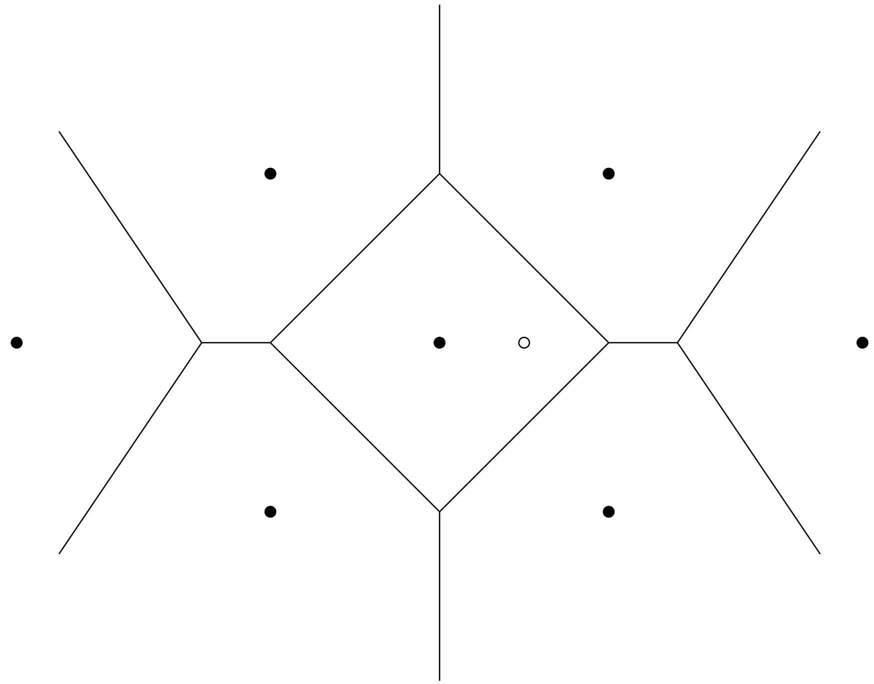
Sibson's Method

- Create Voronoï diagram V of points p_i



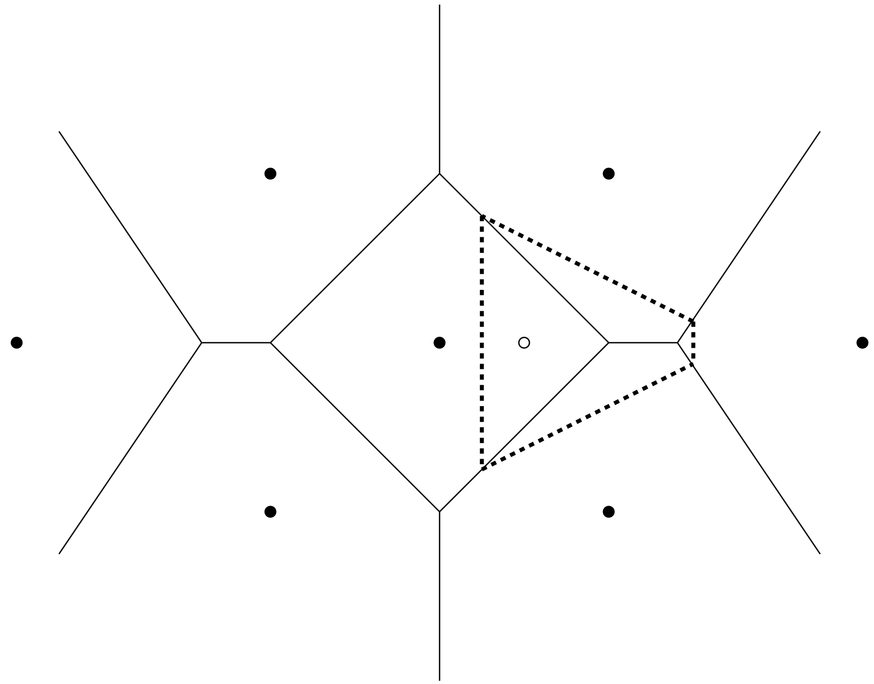
Sibson's Method

- "Tentatively" insert point p into V



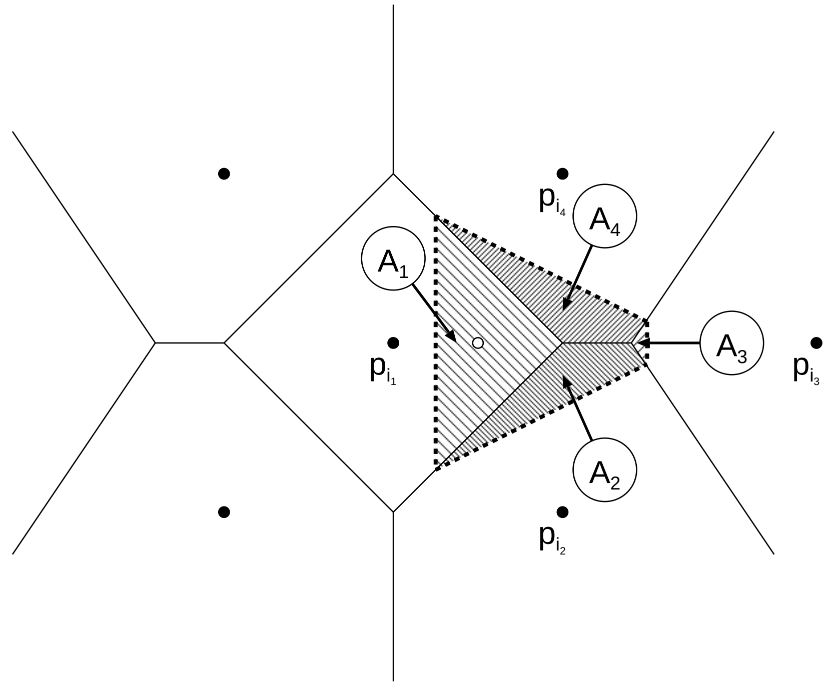
Sibson's Method

- "Tentatively" insert point p into V



Sibson's Method

- Calculate areas A_j of intersecting p's Voronoï tile with tiles of k surrounding points p_{i_j}



Sibson's Method

- Sibson's formula:

$$F(p) = \frac{\displaystyle\sum_{j=1}^{k} A_j v_{i_j}}{\displaystyle\sum_{j=1}^{k} A_j}$$

Sibson's Method

Triangulation Method

- Create (Delaunay) triangulation T of points p_i
- Find triangle t_i containing point p
- Linearly interpolate values at vertices of t_i at position p

Triangulation Method

Optimal Triangulation Method

- Reconstruct F using triangulation of sites
- Optimize site positions based on estimated reconstruction error
- Can yield superior approximation quality using same number of samples

Optimal Triangulation Method

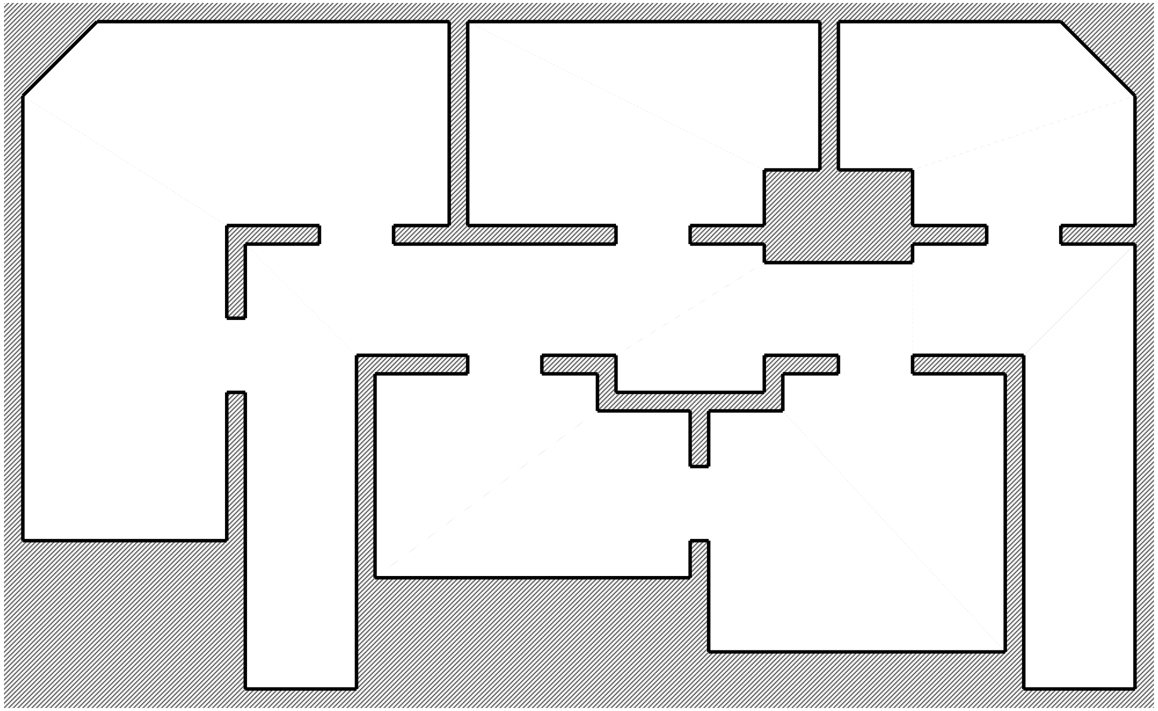
Limitations of Previous Methods

- Described methods are based on Euclidean metric $d(p,q) := (p-q)^2$
- Methods assume compact convex domain D, i. e., no obstacles
- Wrong assumption for building interiors!
- Solution: Define distance metric for buildings
- First: Need to represent complex domains

Representation of Complex Domains

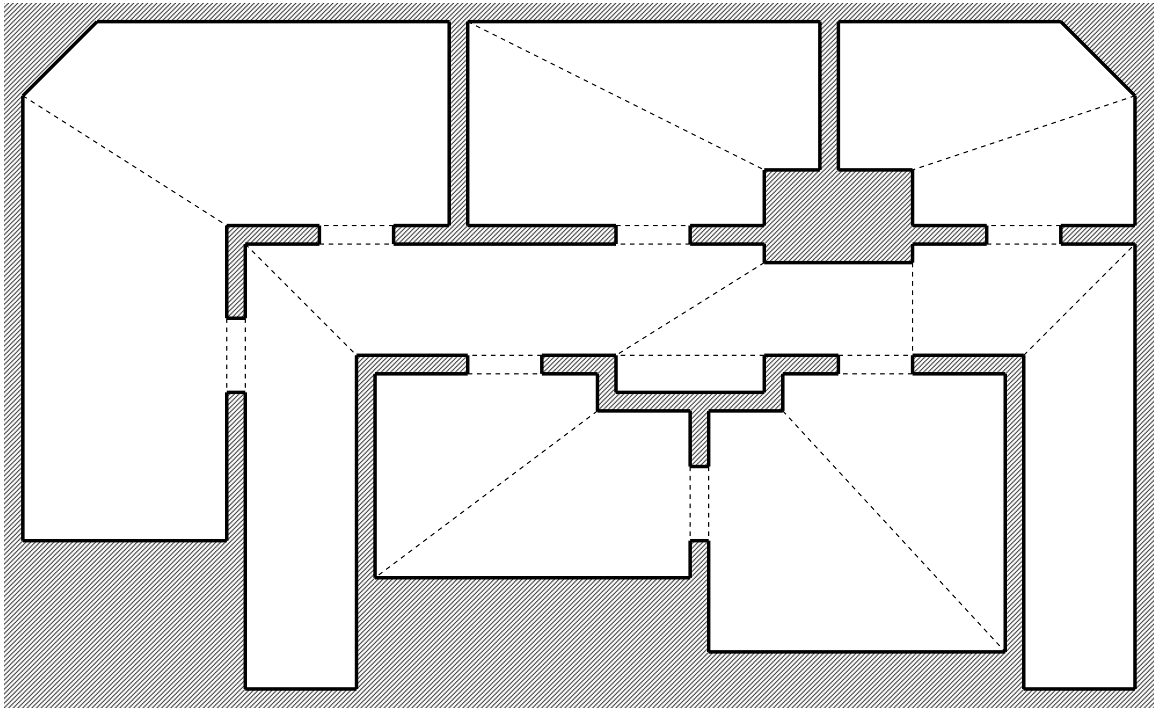
Representing Complex Domains

- Example domain: Hallway with offices



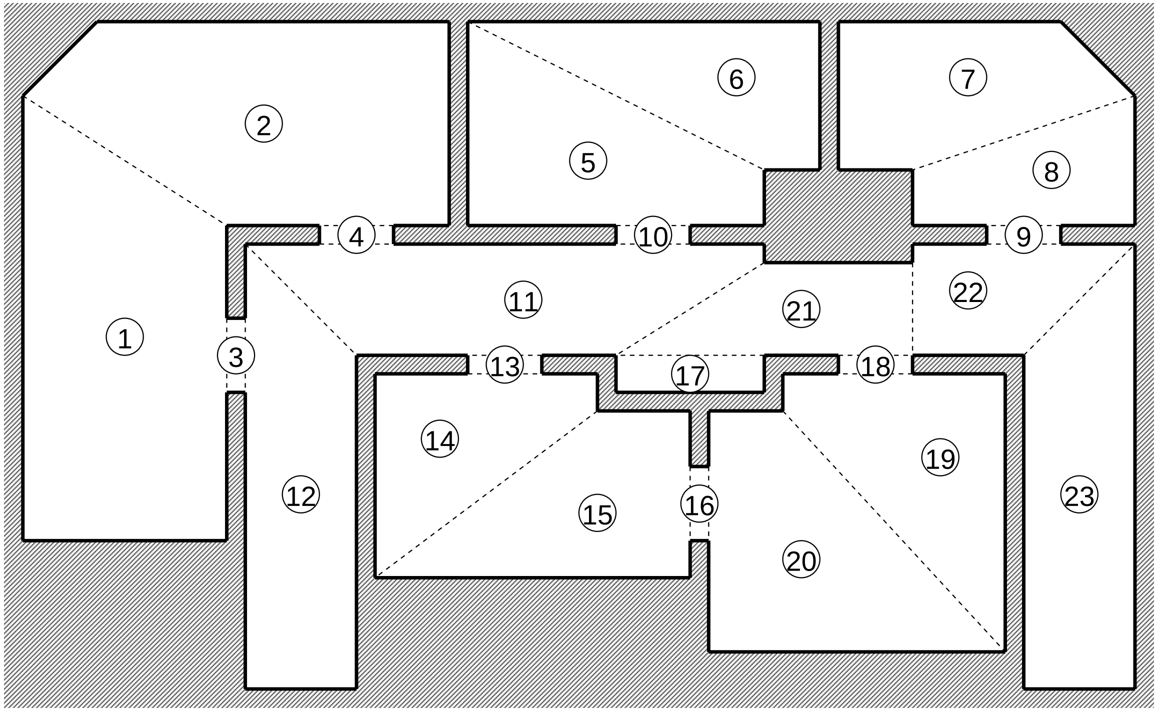
Representing Complex Domains

- Fundamental idea: Represent domain as union of convex polyhedral *sectors* explicitly connected by *portal* polygons



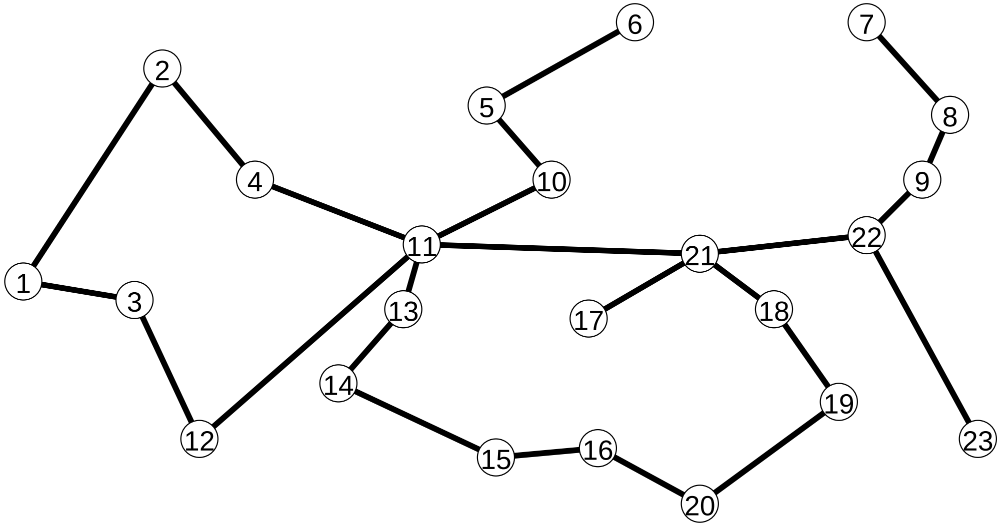
Representing Complex Domains

- Fundamental idea: Represent domain as union of convex polyhedral *sectors* explicitly connected by *portal* polygons



Representing Complex Domains

- Leads to representation of domain as undirected graph

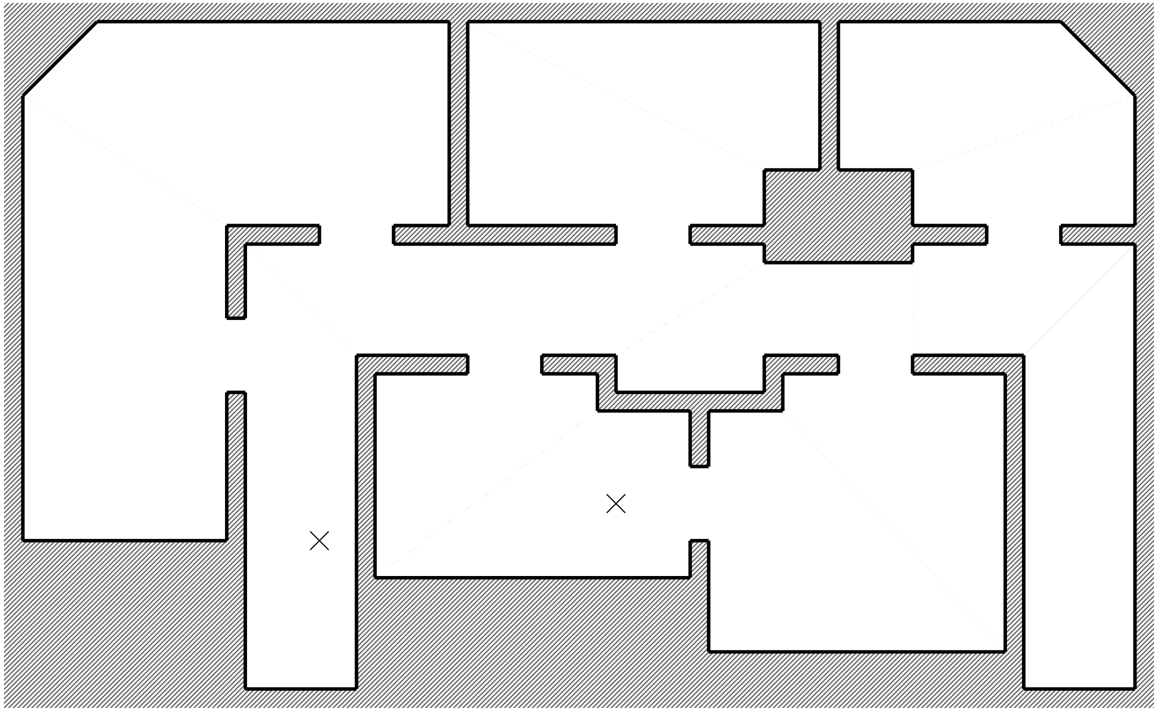


Current State and Outlook

- Domain representation data structure in place
- Need tool to generate domain representation from
  - Architectural blueprints
  - CAD models
  - . . .

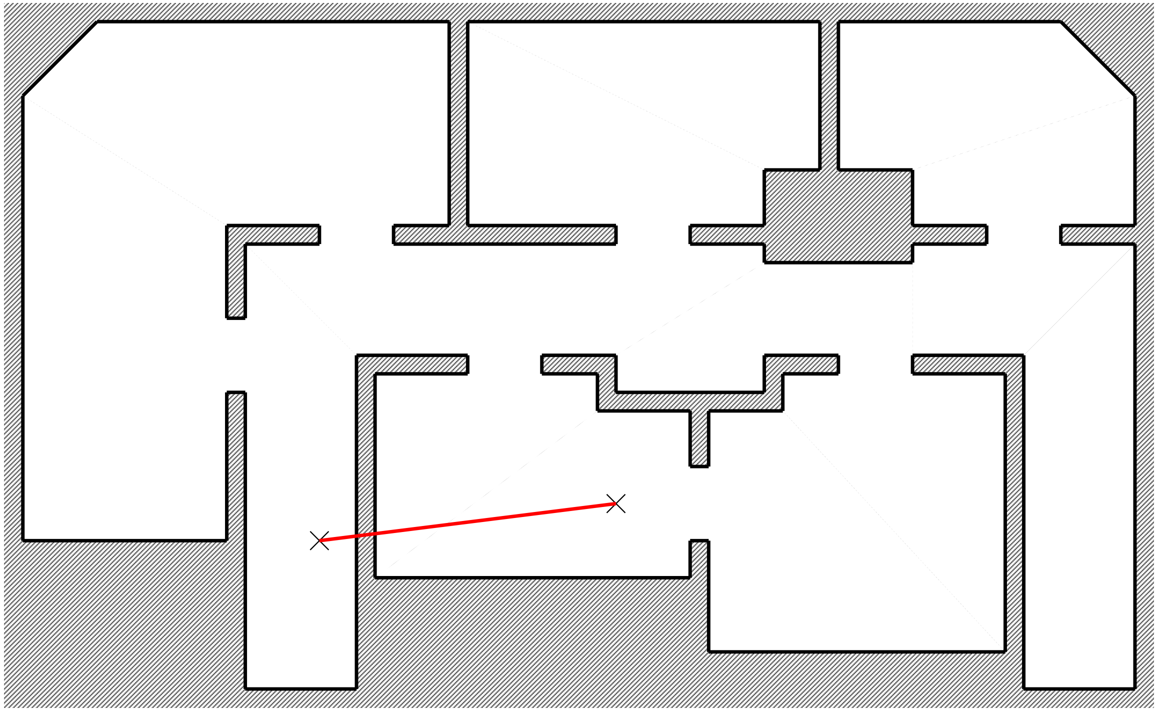
Distance Metric for Complex Domains

- Calculating distance between two query points



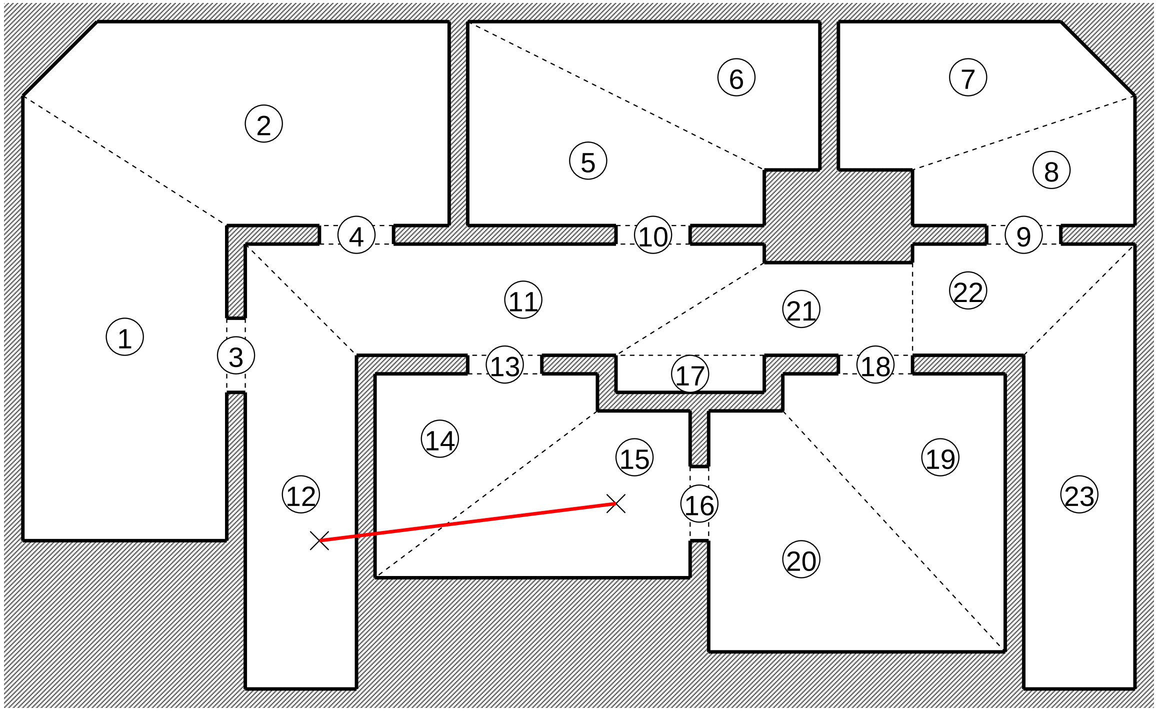
Distance Metric for Complex Domains

- Euclidean metric would ignore boundaries



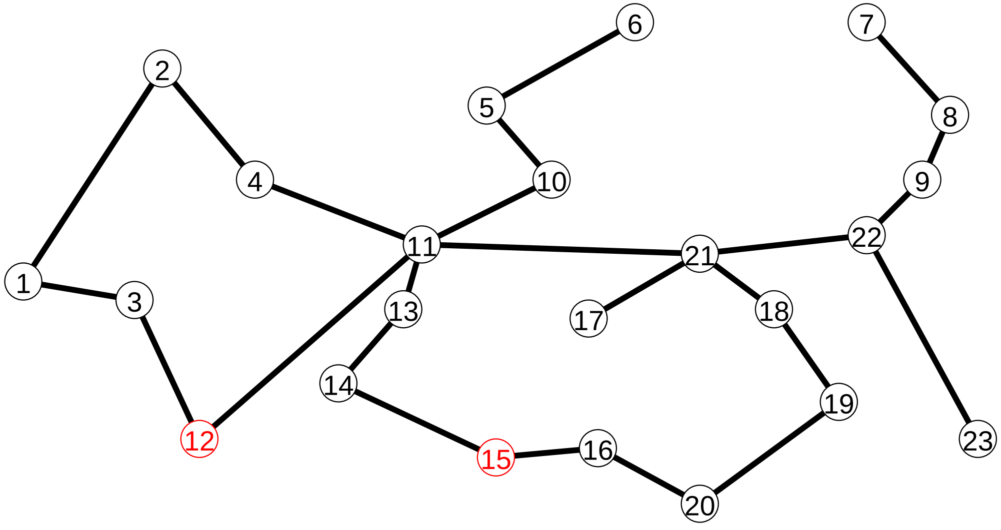
Distance Metric for Complex Domains

- Determine sector(s) containing query points



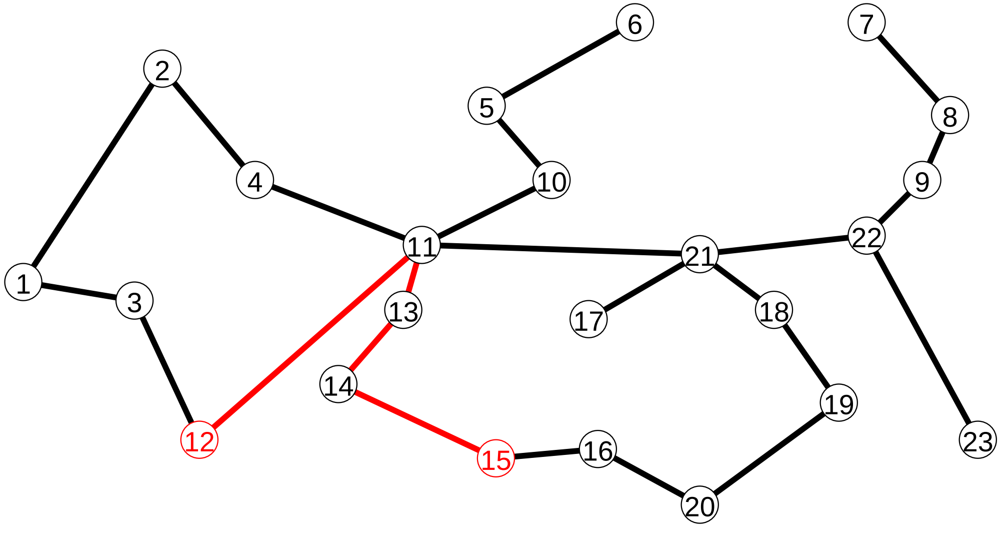
Distance Metric for Complex Domains

- Consider graph to find path between sectors



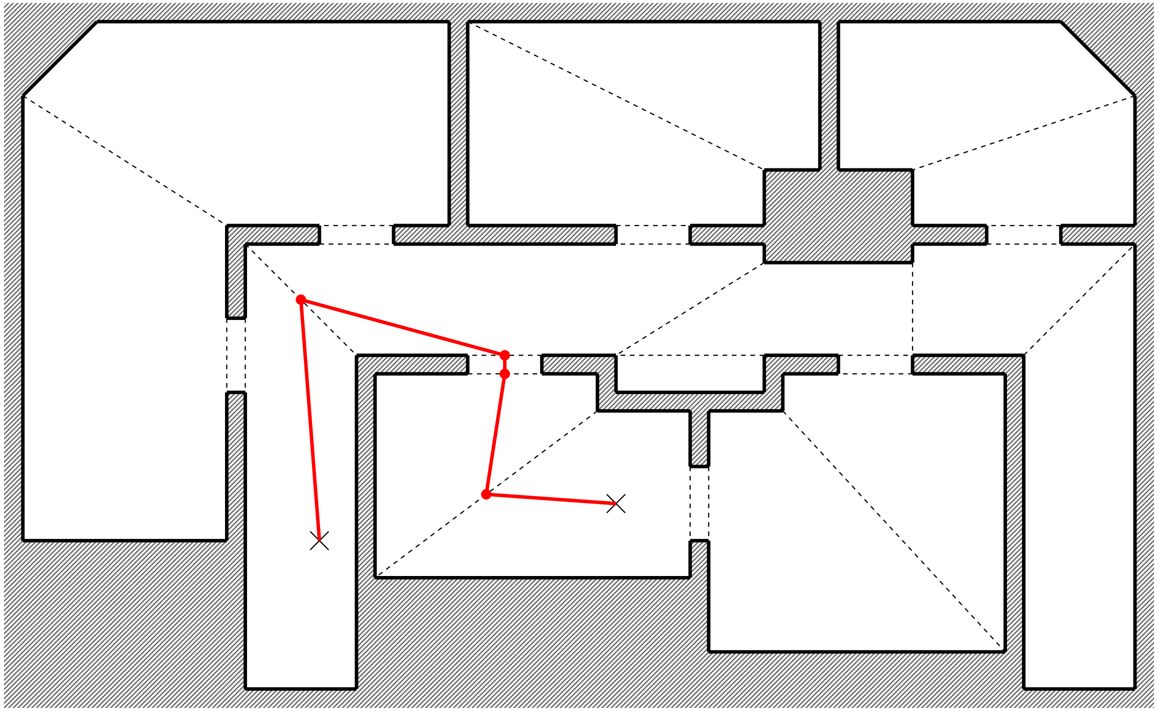
Distance Metric for Complex Domains

- Graph traversal yields sequence of portals to cross



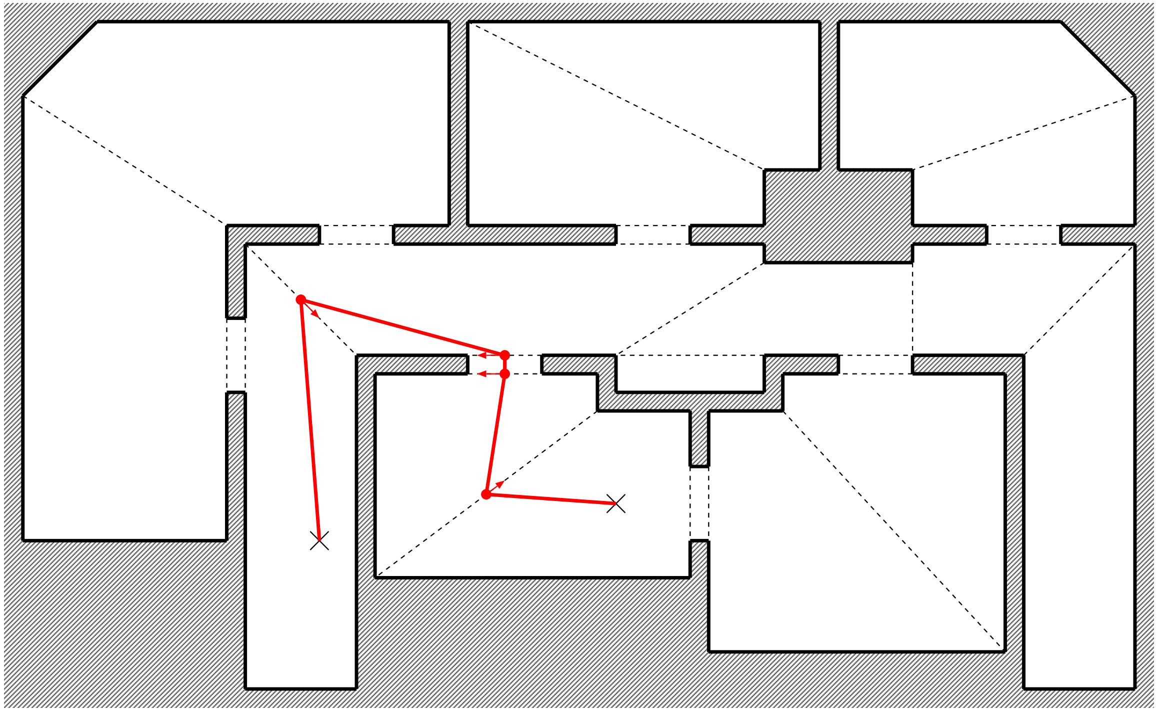
Distance Metric for Complex Domains

- Cross portals through their centroids initially



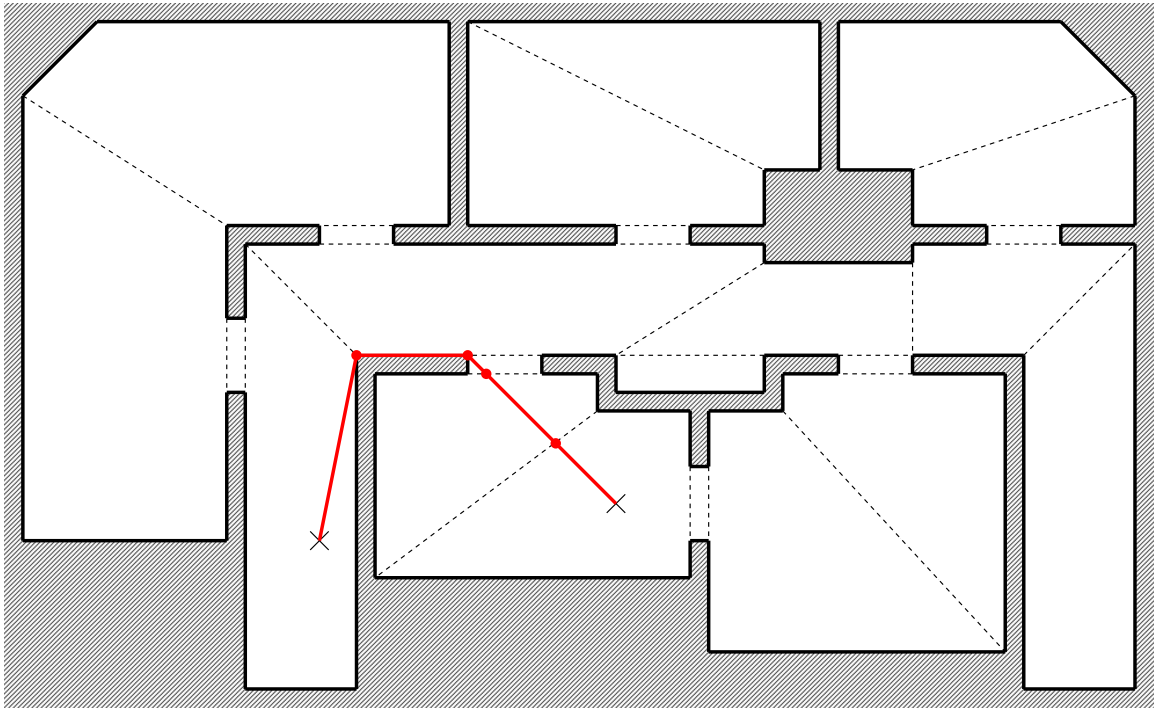
Distance Metric for Complex Domains

- Optimize placement of intermediate points



Distance Metric for Complex Domains

- Optimize placement of intermediate points



Metric Properties

- If there exists a direct line-of-sight between two query points, the described metric agrees with the Euclidean metric

  $\Rightarrow$ Artificial (and arbitrary) boundaries introduced by domain decomposition do not influence distances

- Metric can be computed efficiently

Metric Properties

- Metric can be used as drop-in replacement for Euclidean metric in Shepard's and Hardy's method
- Metric defines Voronoï diagram (and Delaunay triangulation) for Sibson's and triangulation method
  - Shape of Voronoï tiles?
  - Efficient computation?

Current State and Outlook

- One graduate student working on optimization algorithm
- One undergraduate student will soon start working on scattered data methods compatible with metric

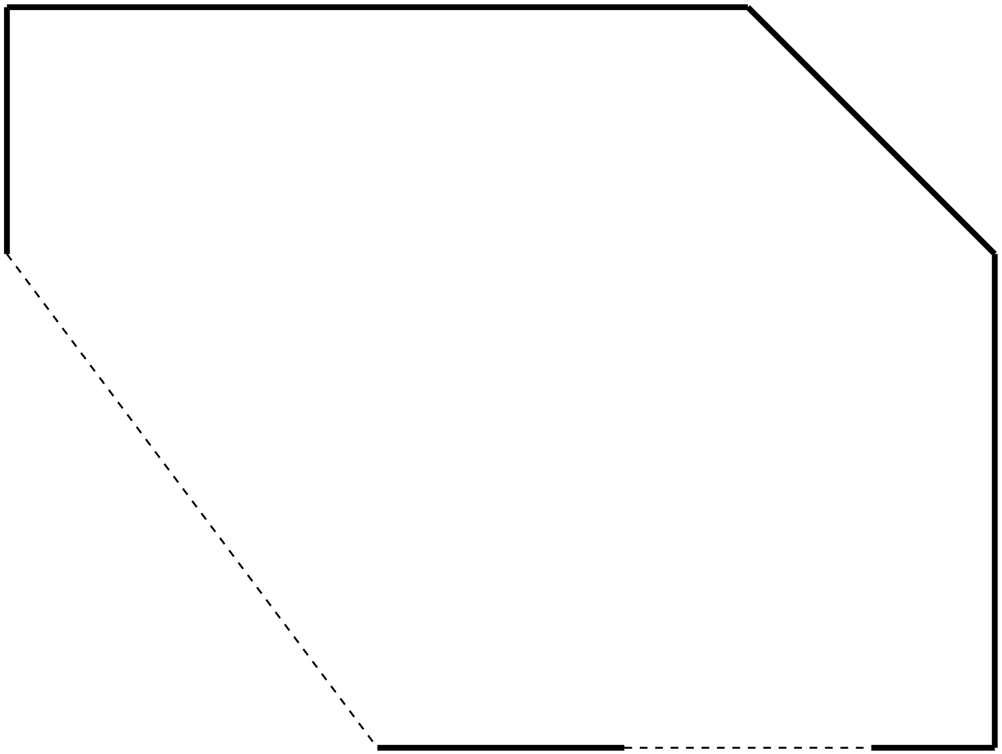
Visualizing Sensor Network Data

Visualization Strategy

- Basic idea: To visualize sensor data in context, combine rendering of domain boundaries with data visualization
- Domain representation is also efficient for interactive walk-throughs
- Sensor data can be reconstructed and visualized separately in each sector
- Since each sector is a convex polyhedron, standard visualization methods can be used

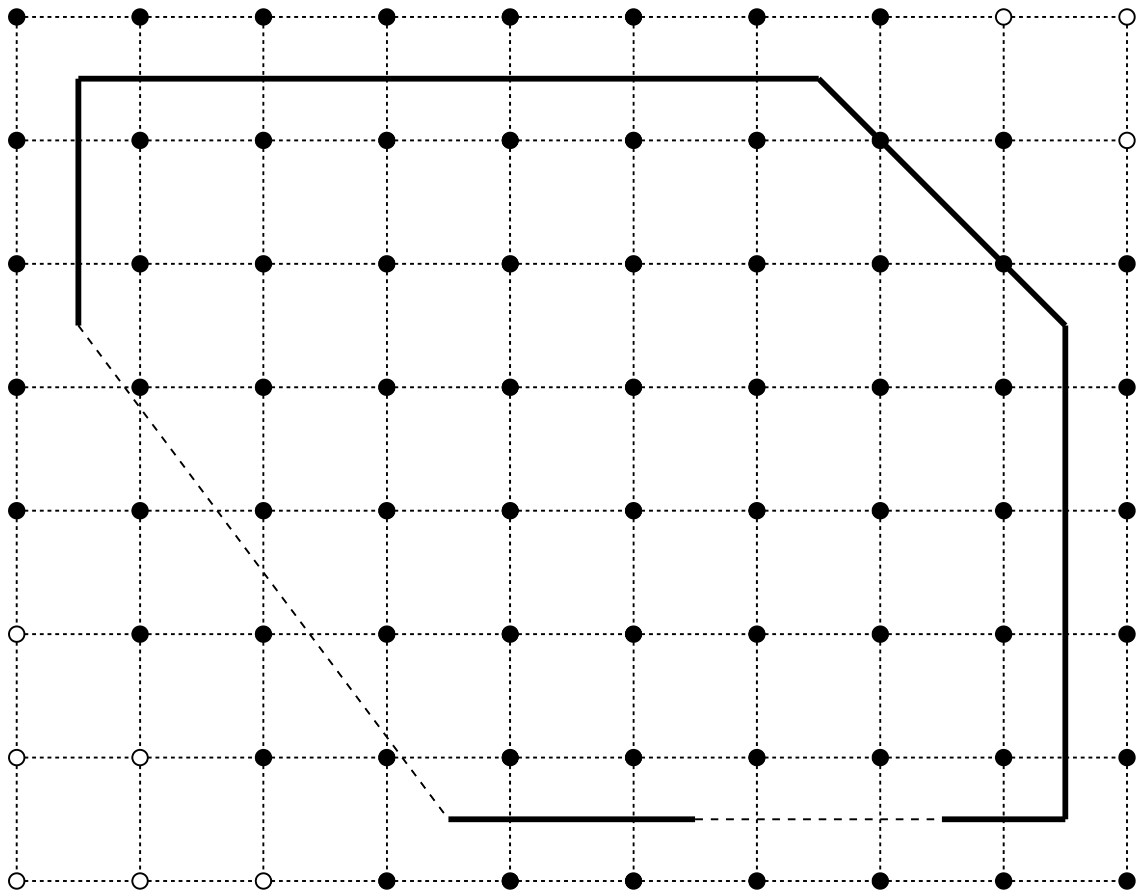
Per-Sector Visualization

- Single sector is convex polyhedron



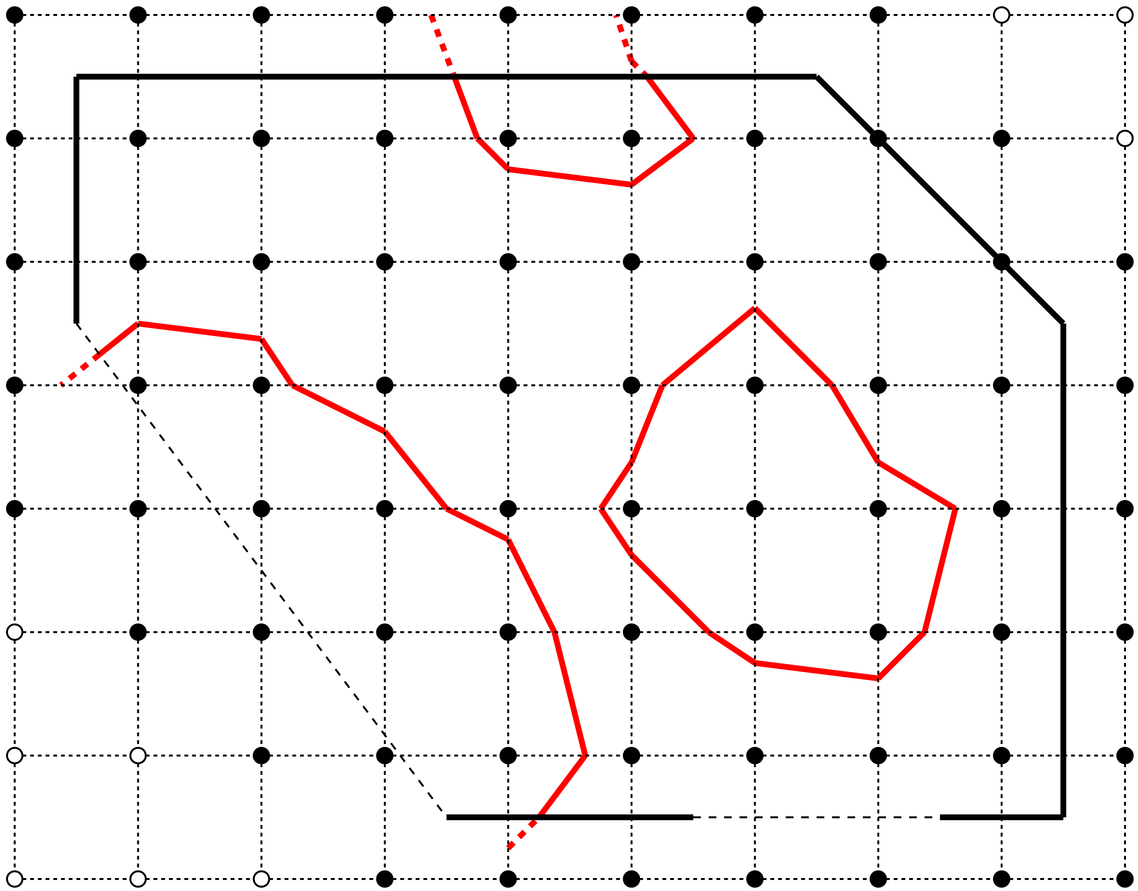
Per-Sector Visualization

- Overlay sector with regular grid



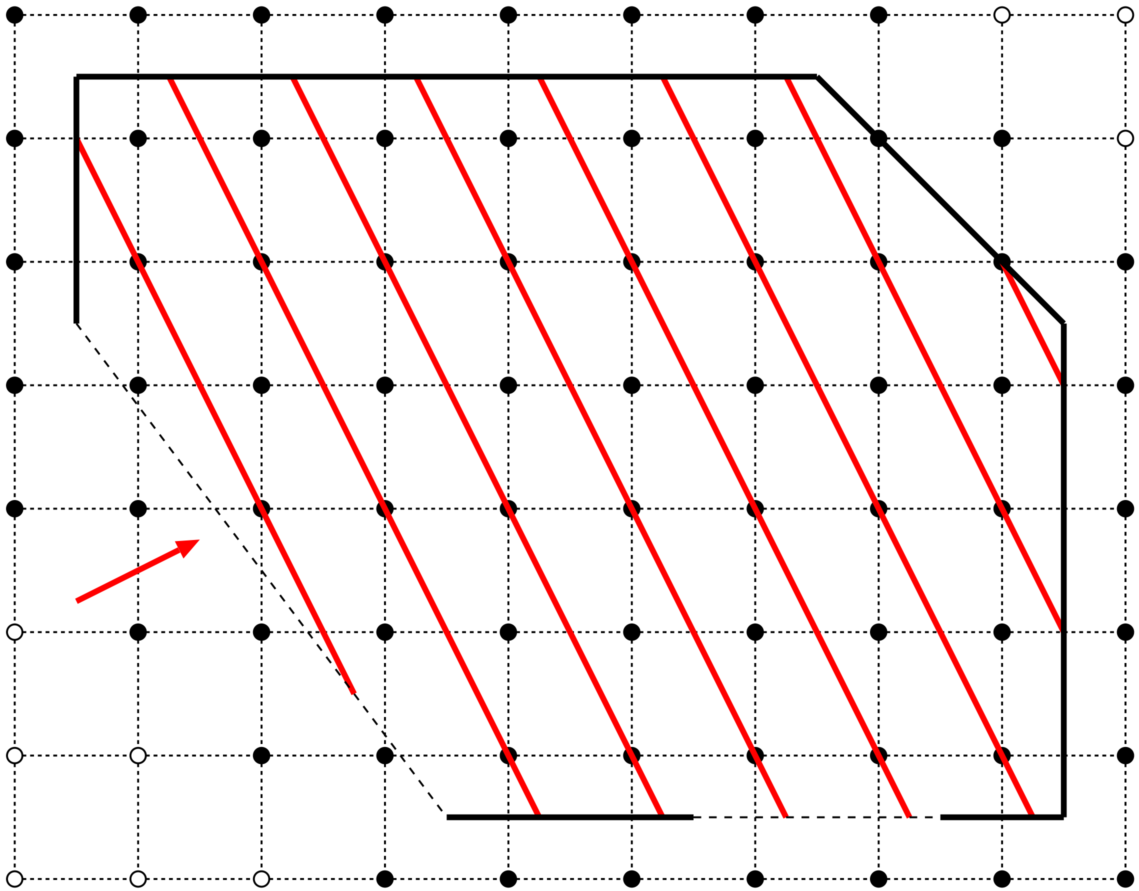
Per-Sector Visualization

- Generate isosurface using Marching Cubes

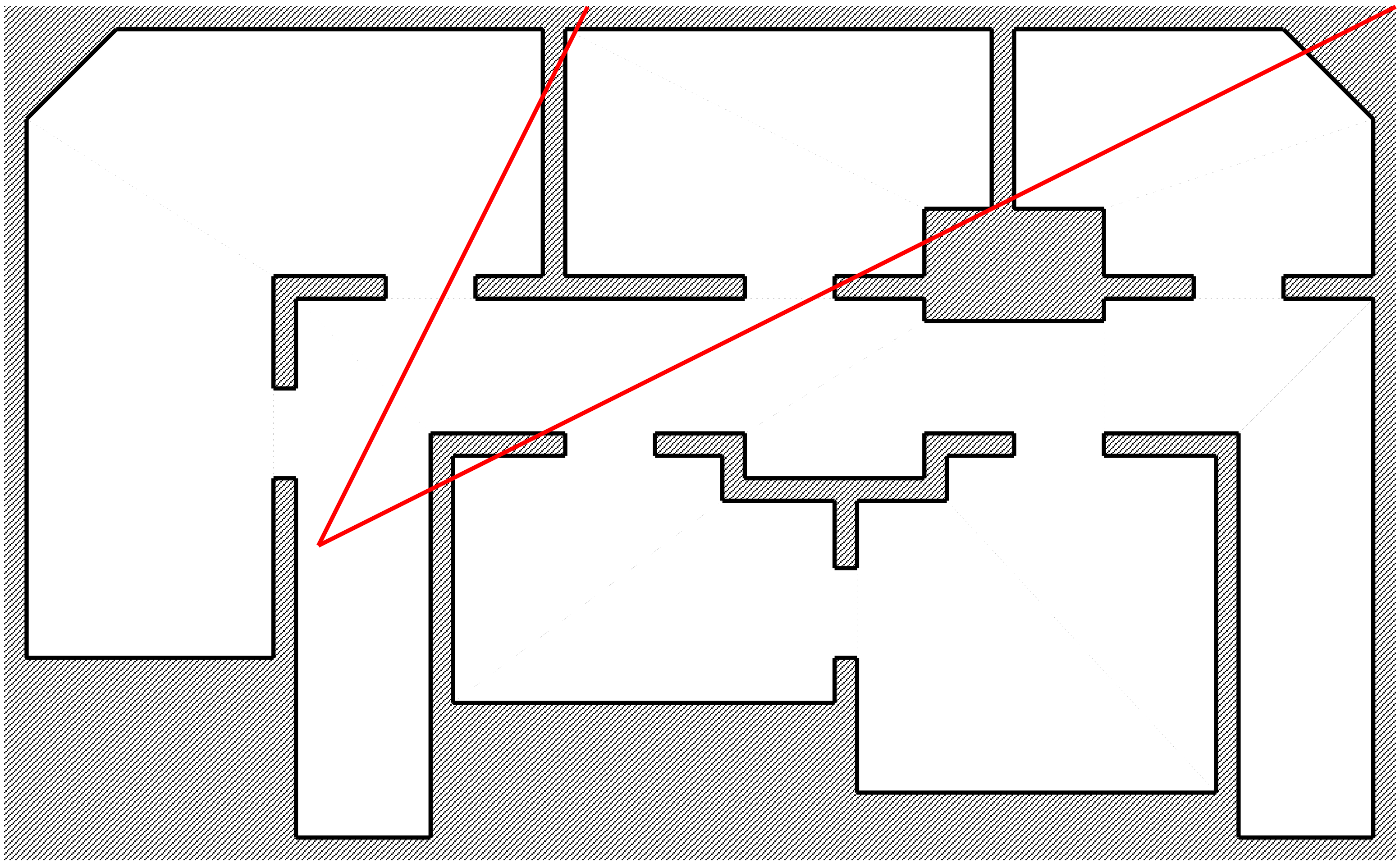


Per-Sector Visualization

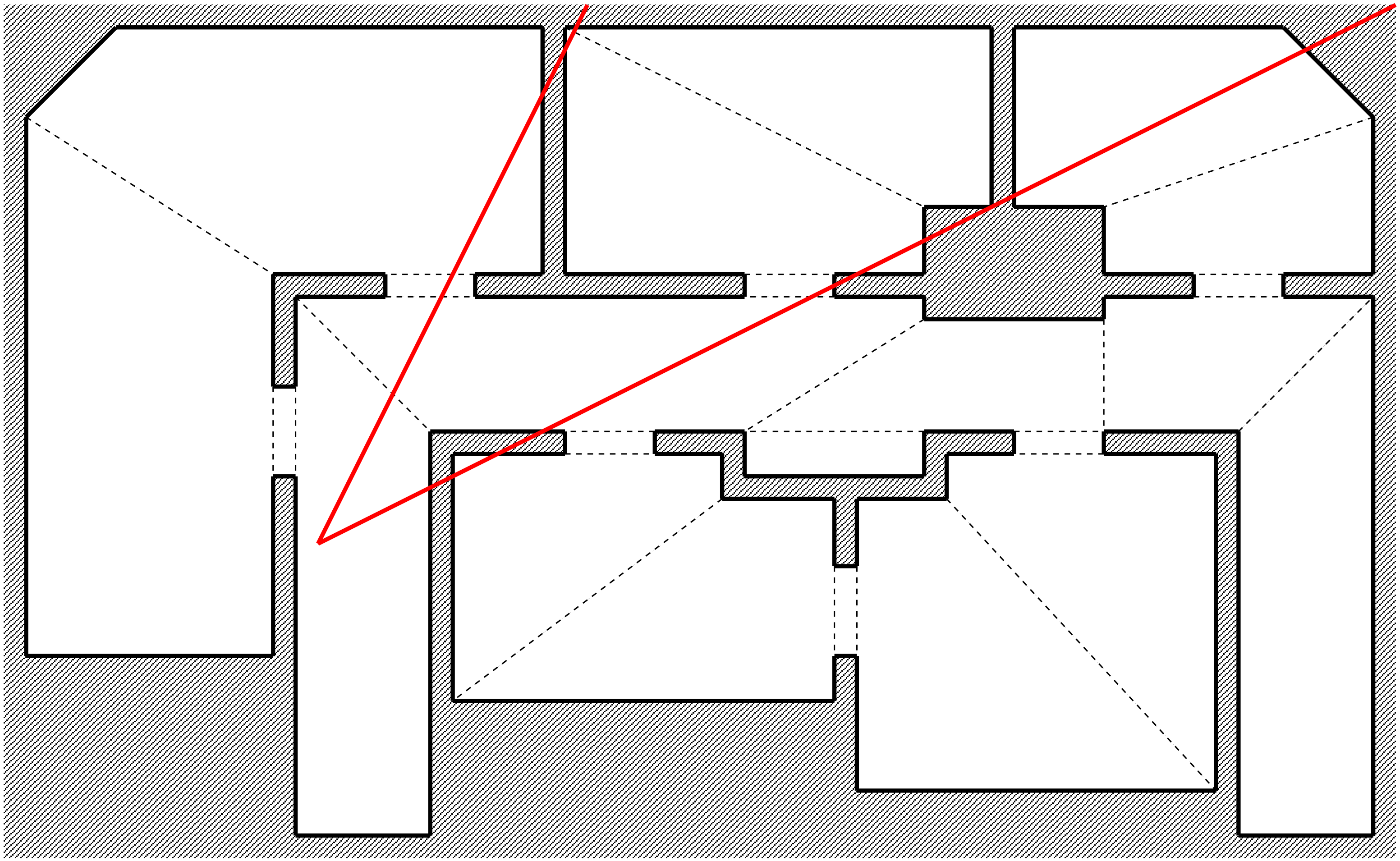
- Generate slices for volume rendering



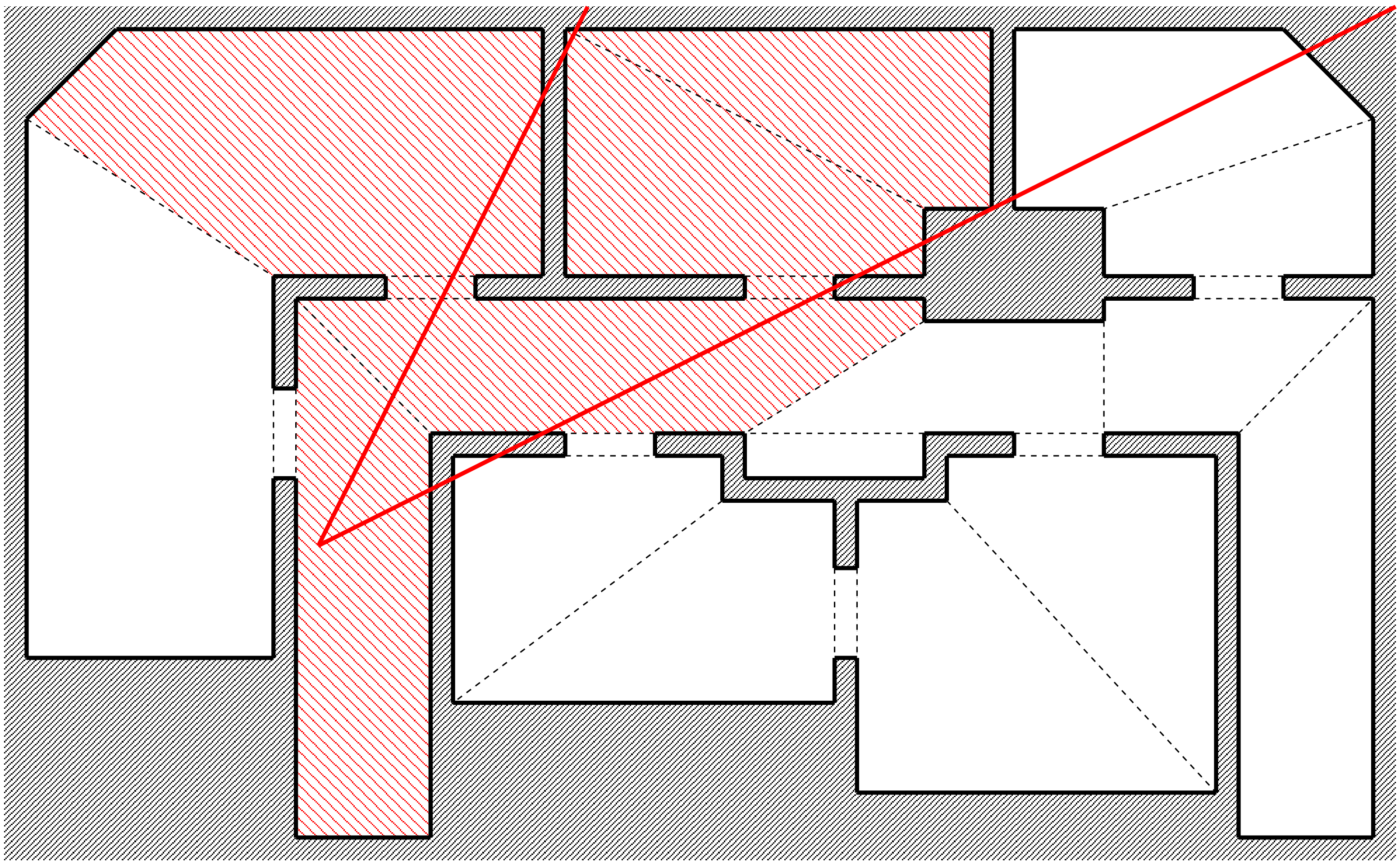
Visualizing Entire Domain



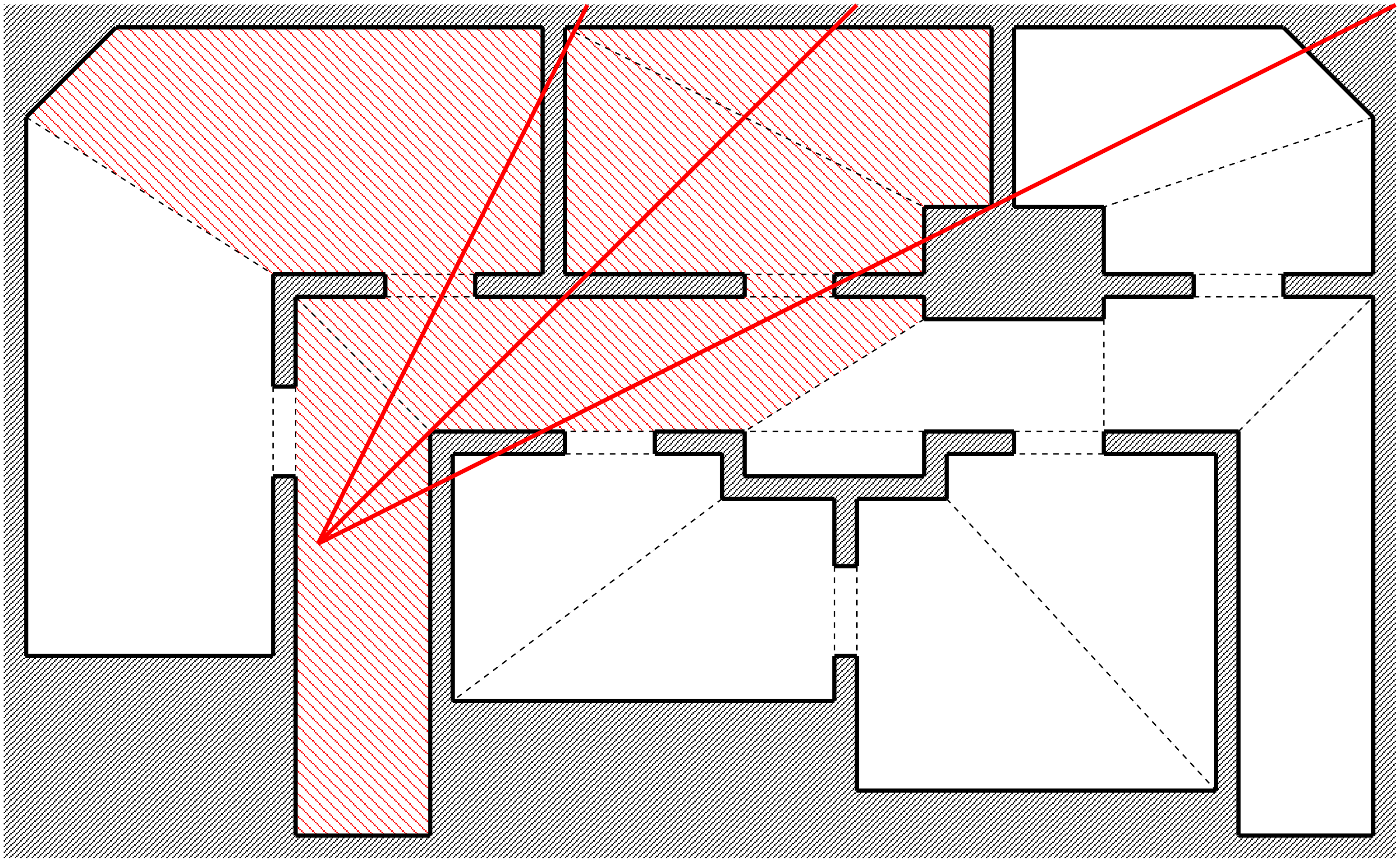
Visualizing Entire Domain



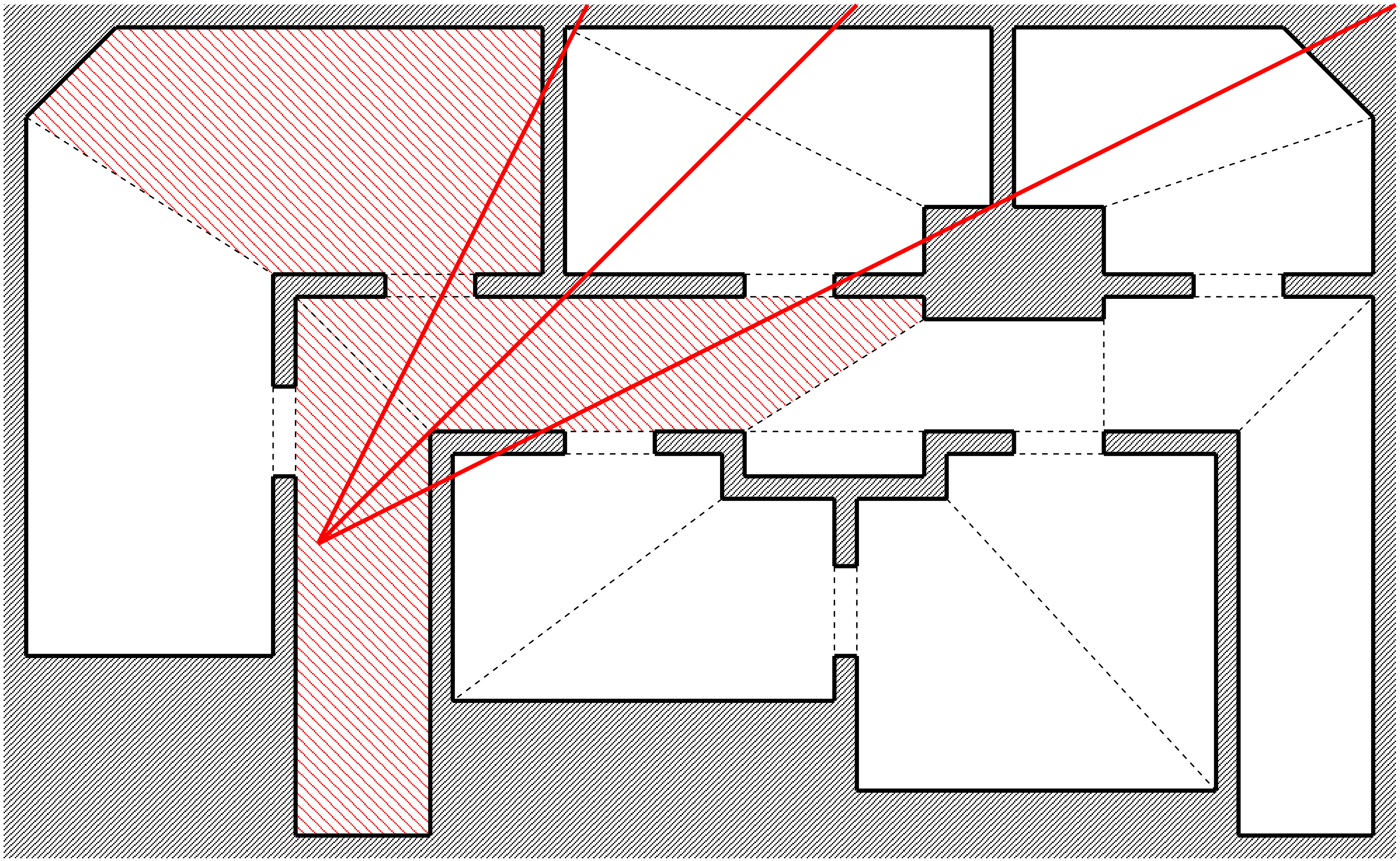
Visualizing Entire Domain



Visualizing Entire Domain



Visualizing Entire Domain



Current State and Outlook

- Walk-throughs of complex domains will be done Real Soon Now™
- Walk-through will double as tool to “polish” sensor data input
- Sensor data visualization will be integrated when metric and scattered data method are in place
- Visualization will run in desktop and Virtual Reality environment

Conclusions

Short-Term Goal

- Combine
  - domain representation
  - distance metric calculation
  - (direct) scattered data interpolation method

to visualize static sensor network data inside buildings

Long-Term Goals

- Analyze sensor network data based on reconstruction
- Move to indirect (grid-based) scattered data methods
- Move to time-varying and dynamic sensor data
- Optimize sensor placement based on reconstruction

The End