SPace of Eulerian MeasureS (SPEMS): N-dimensional Treatment of Eulerian Analysis of the Chespeake Bay Mouth

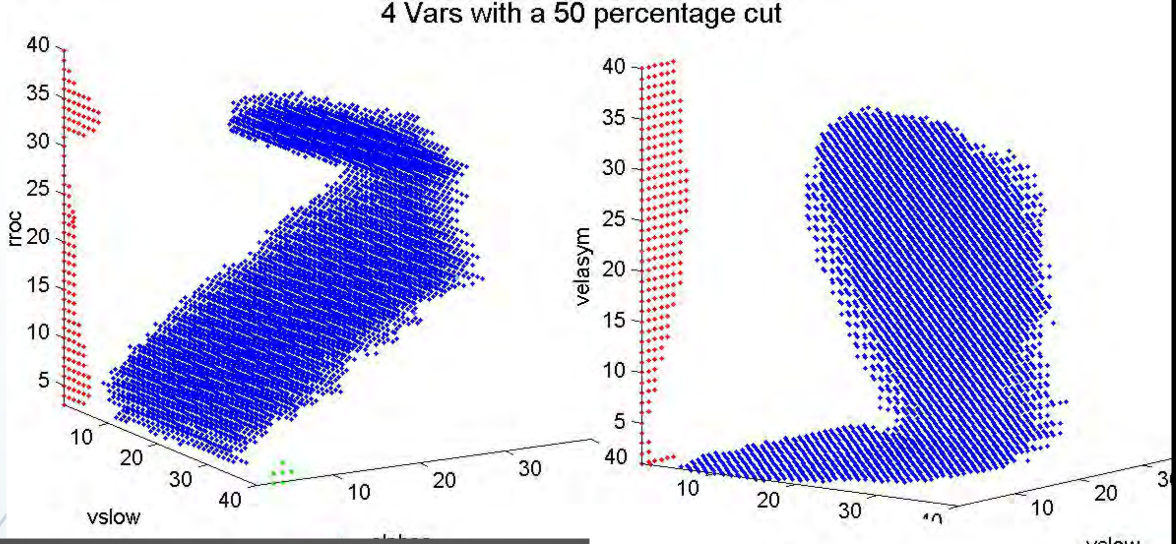
Kevin Mcilhany

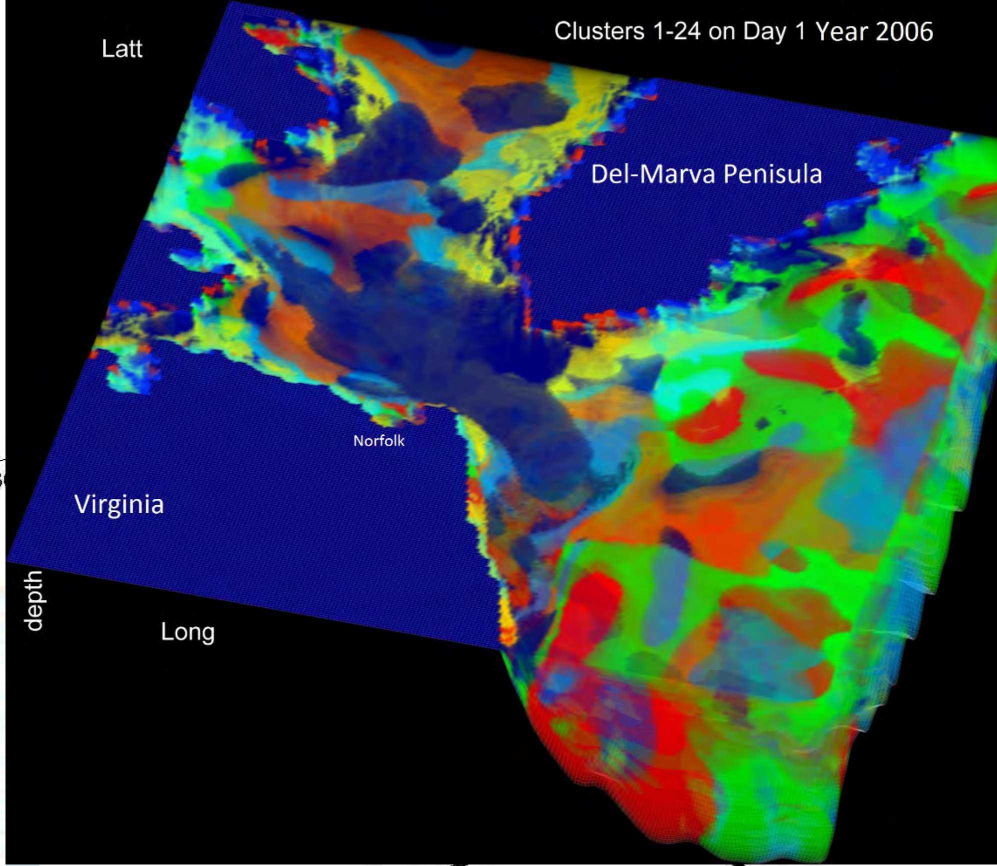
Physics Dept, US Naval Academy

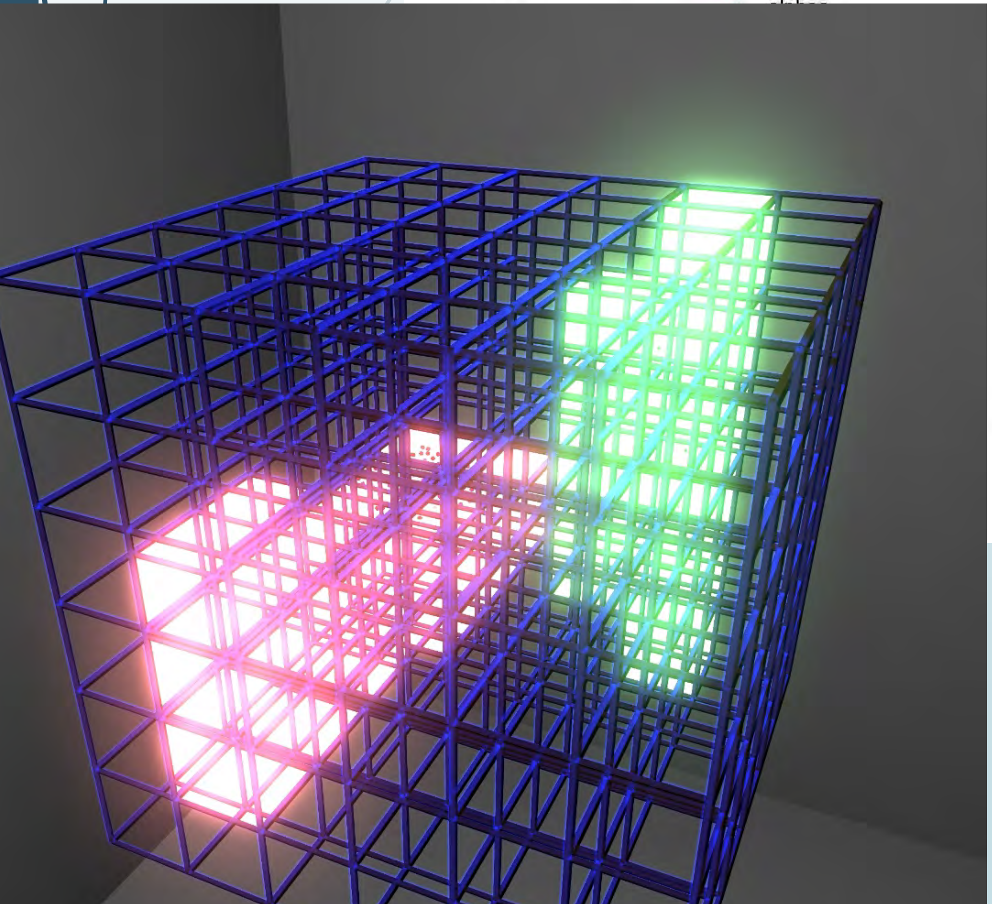
July 7, 2016

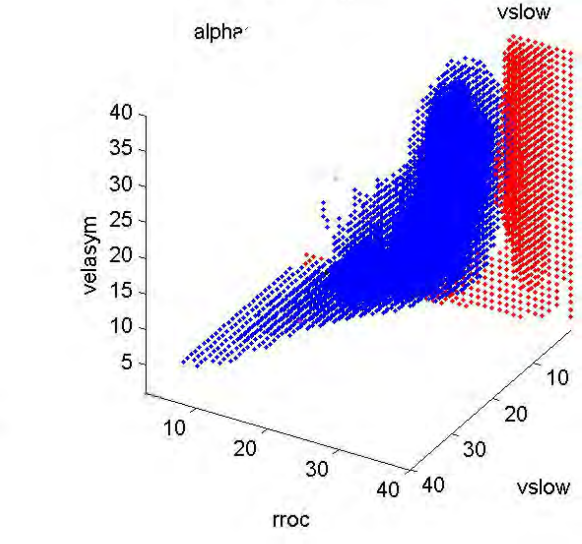
in collab: Wiggins, Malek-Madani

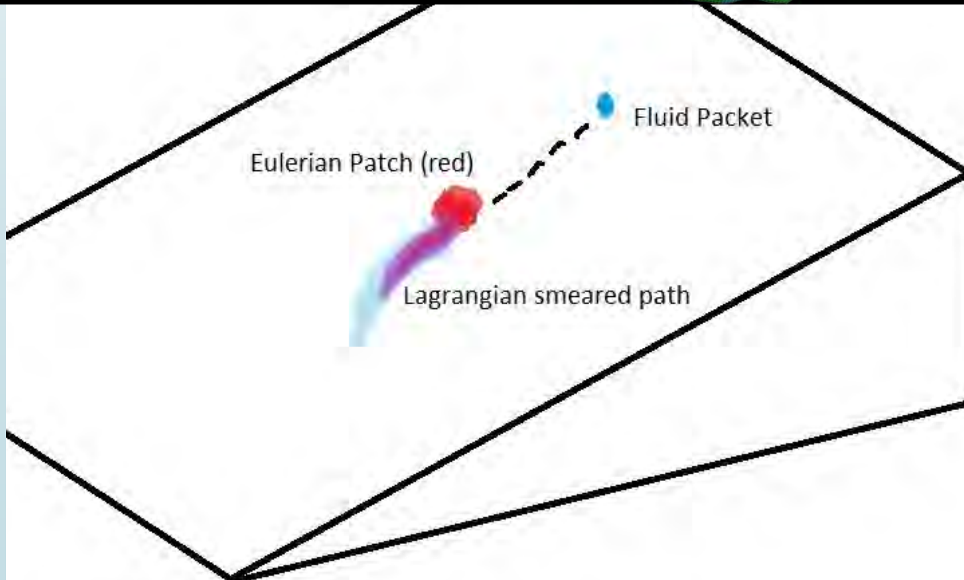
4 Slides:


4 Vars with a 50 percentage cut
rroc
vslow
velasym
alpha


Latt
Clusters 1-24 on Day 1 Year 2006
Del-Marva Penisula
Norfolk
Virginia
depth
Long




velasym
rroc
vslow
alpha


Fluid Packet
Eulerian Patch (red)
Lagrangian smeared path

Outline:

- Fluid analysis: Eulerian – Lagrangian
  - Eulerian Measures – KE, vorticity, OW, transversality, RROC, shear, mobility
- Data Analysis - techniques
  - Data Manifolds – N-dimensions
  - Clustering
  - Applying Clusters to Data
- Future Work
  - Seeking correlations between Eulerian and Lagrangian

ChesROMS – 2006 simulated year

- ChesROMS simulated by
  Kayo Ide, Bin Zhang (CSCAMM-UMD)
- Modified ChesROMS grid – 1km x 1km, 20 sigma layers, rectilinear
- Simulated every 10 minutes
- Collected every hour
- Mouth of Bay center +/- 60km x 80km
- 47,000 locations per layer per day

Eulerian Measures

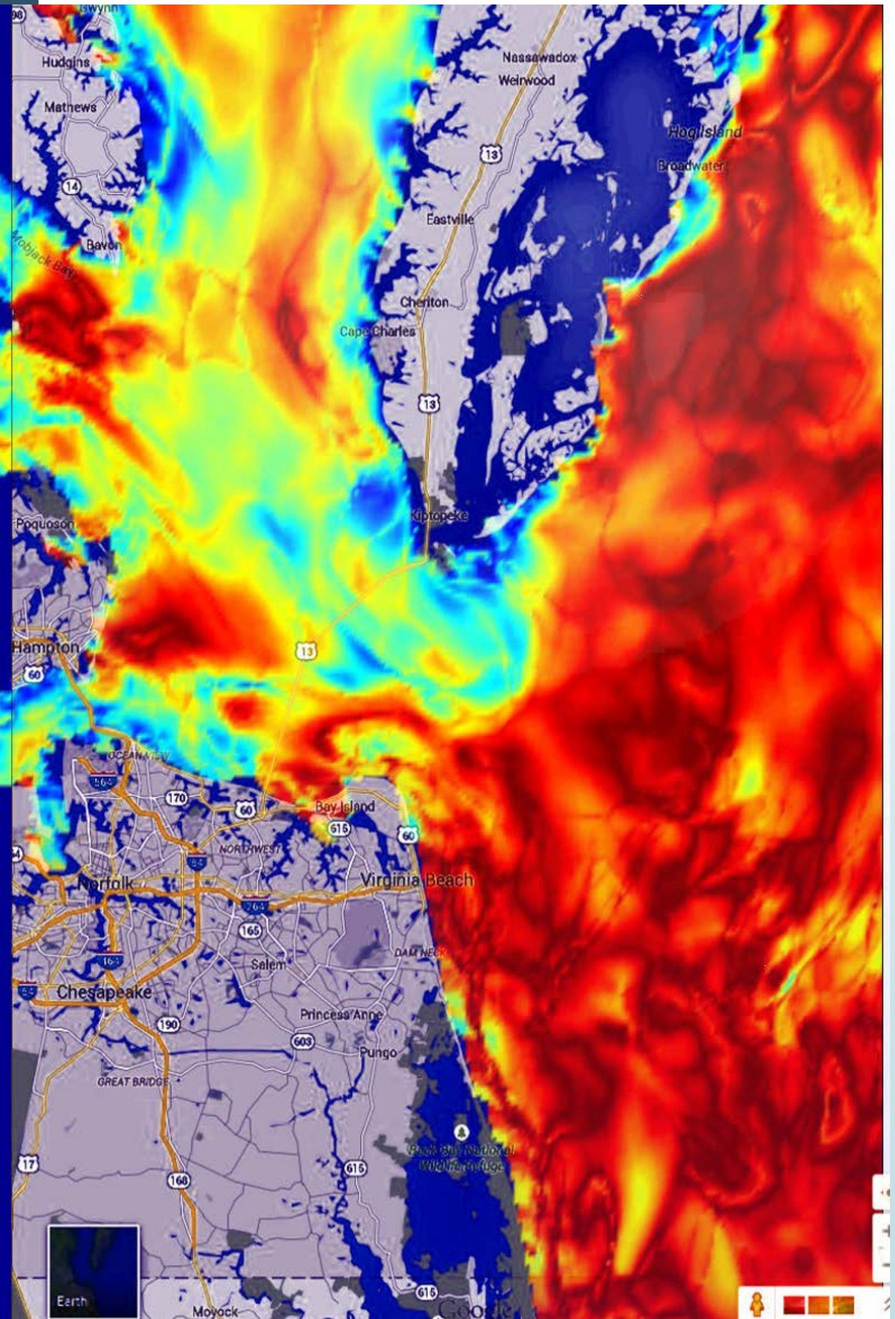
- Kinetic Energy $\approx |\vec{u}|^2$
- Vorticity $\vec{\omega} = \vec{\nabla} \times \vec{u}$
- Okubo-Weiss (Q-crit, shear) $= \sigma_n^2 + \sigma_s^2 - \omega^2$
- Transversality $\alpha = \mathcal{V}_\theta \equiv \frac{1}{T}\int_0^T (\theta(t) - \langle\theta\rangle)^2 dt$
- Relative Rate Of Change $\mathrm{RROC} = \frac{1}{T}\int_0^T \frac{\|\vec{u}(t+\Delta t) - \vec{u}(t)\|}{\|\vec{u}(t+\Delta t)\| + \|\vec{u}(t)\|} dt$
- V-Slow $= \left\langle |\vec{\mathbf{u}}(t)| \right\rangle = \frac{1}{T}\int_0^T |\vec{\mathbf{u}}(t)| dt$
- V-Fast $= \left| \langle \vec{\mathbf{u}}(t) \rangle \right| = \left| \frac{1}{T}\int_0^T \vec{\mathbf{u}}(t) dt \right|$
- Velocity Asymmetry $= \frac{V_{slow} - V_{fast}}{V_{slow} + V_{fast}}$
- Transverse Shear $\beta = |\vec{\boldsymbol{\rho}}|\ sin(\phi)$

Eulerian Measures

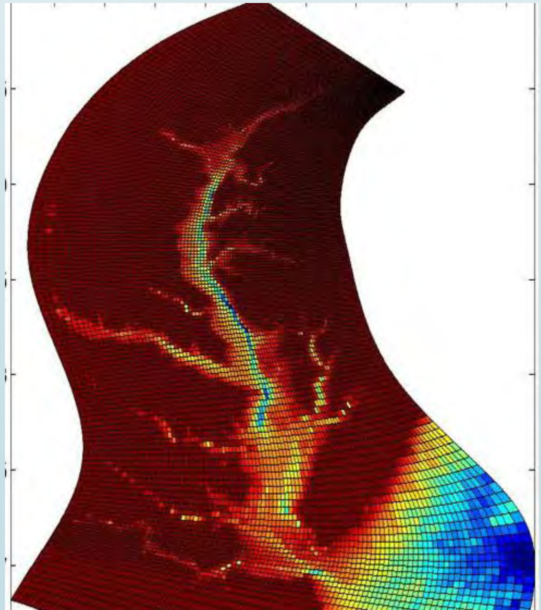
- 3 Types:
  - Instaneous (averaged over 24 hours)
  - Measure spatial derivatives via velocity gradient tensor (avg 24 hours)
  - Measure temporal derivatives, integrals, moments (24 hour window)
- References:

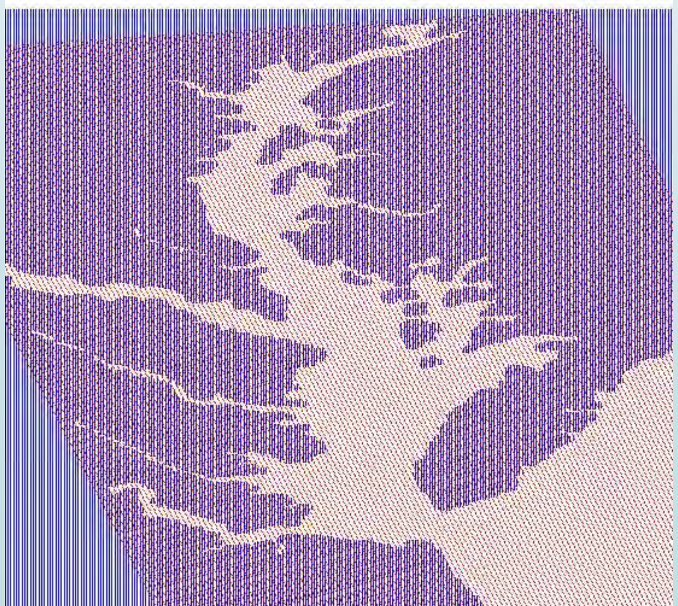
- McIlhany, K. L., Wiggins, S., "Optimizing Mixing in Channel Flows: Kinematic Aspects Associated with Secondary Flows in the Cross-Section", Microfluidics and Nanofluidics, 10, 2011
- McIlhany, K. L., Mott, D., Oran, E., Wiggins, S., "Optimizing mixing in lid-driven flow designs through predictions from Eulerian indicators", Phys. Fluids, 8-23, 2011
- McIlhany, K. L., Wiggins, S., "Eulerian indicators under continuously varying conditions", Phys.Fluids, 24-7, 2012
- Mcilhany, K. L., Guth, S., Wiggins, S., "Lagrangian and Eulerian Analysis of Transport and Mixing in the Three Dimensional, Time Dependent Hill's Spherical Vortex", Phys. Fluids, 27:6, 2015
- ELKI and Schubert, E. and Koos, A., Emrich, T., Zufle, A., Schmid, K.A., Zimek, A., "A Framework for Clustering Uncertain Data", http://www.vldb.org/pvldb/vol8/p1976-schubert.pdf,
- Haller, G. "Objective Definition of a Vortex", J. Fluid. Mech., 2005.

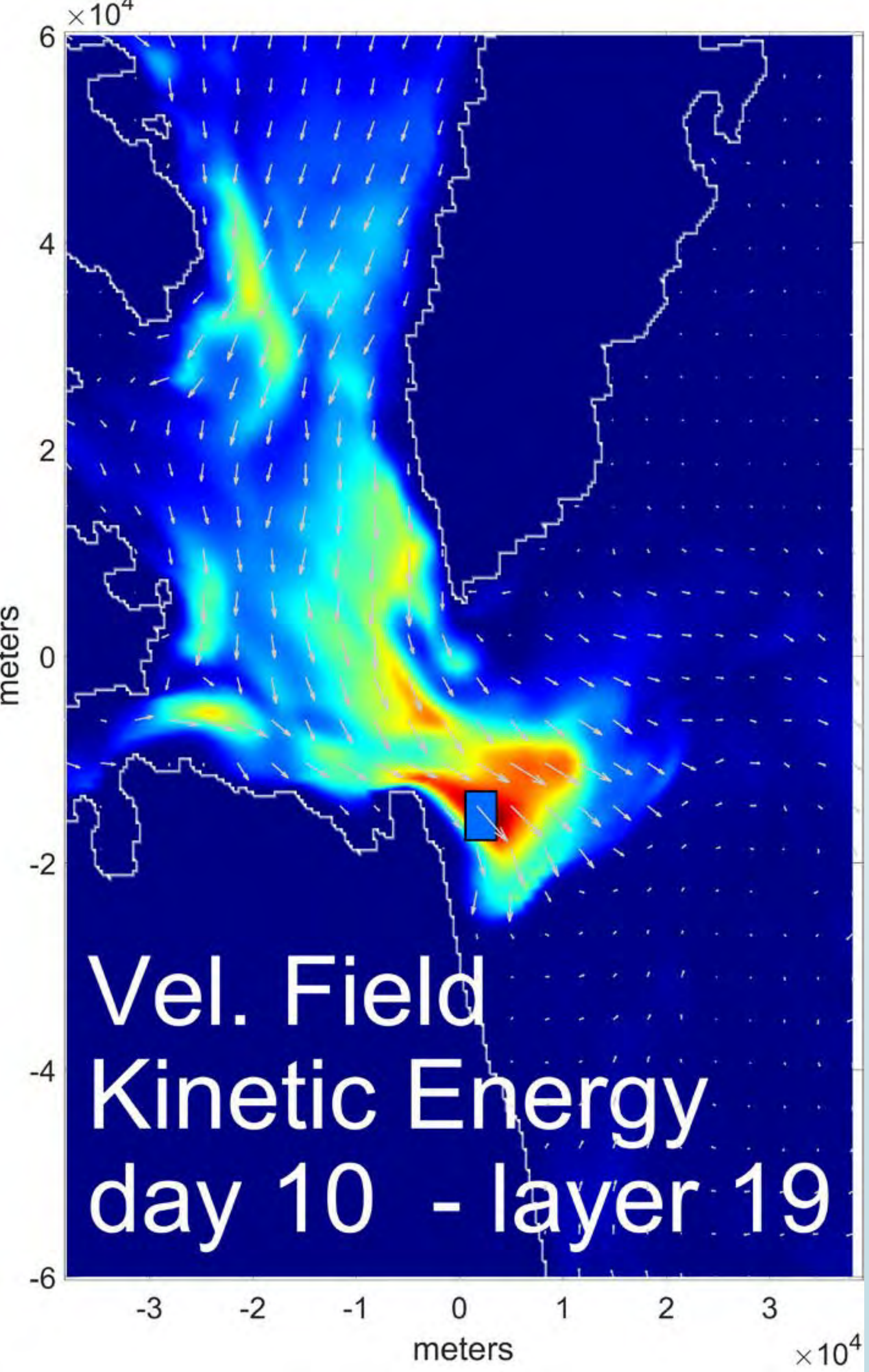
Geo-Referenced Chesapeake Bay Mouth



- Full Chesapeake Bay:
  - 180 miles North-South
  - ~50 miles East-West
  - Most narrow 5 miles across
  - 11,500 miles coastline (fractal-like)
  - Average depth, 8.4m
  - Maximum depth 24m along “spine”
- Chesapeake Bay Mouth:
  - Origin located ~half along mouth
  - +/- 80km North-South
  - +/- 60km East-West
  - Grid points every 1km on rectilinear grid

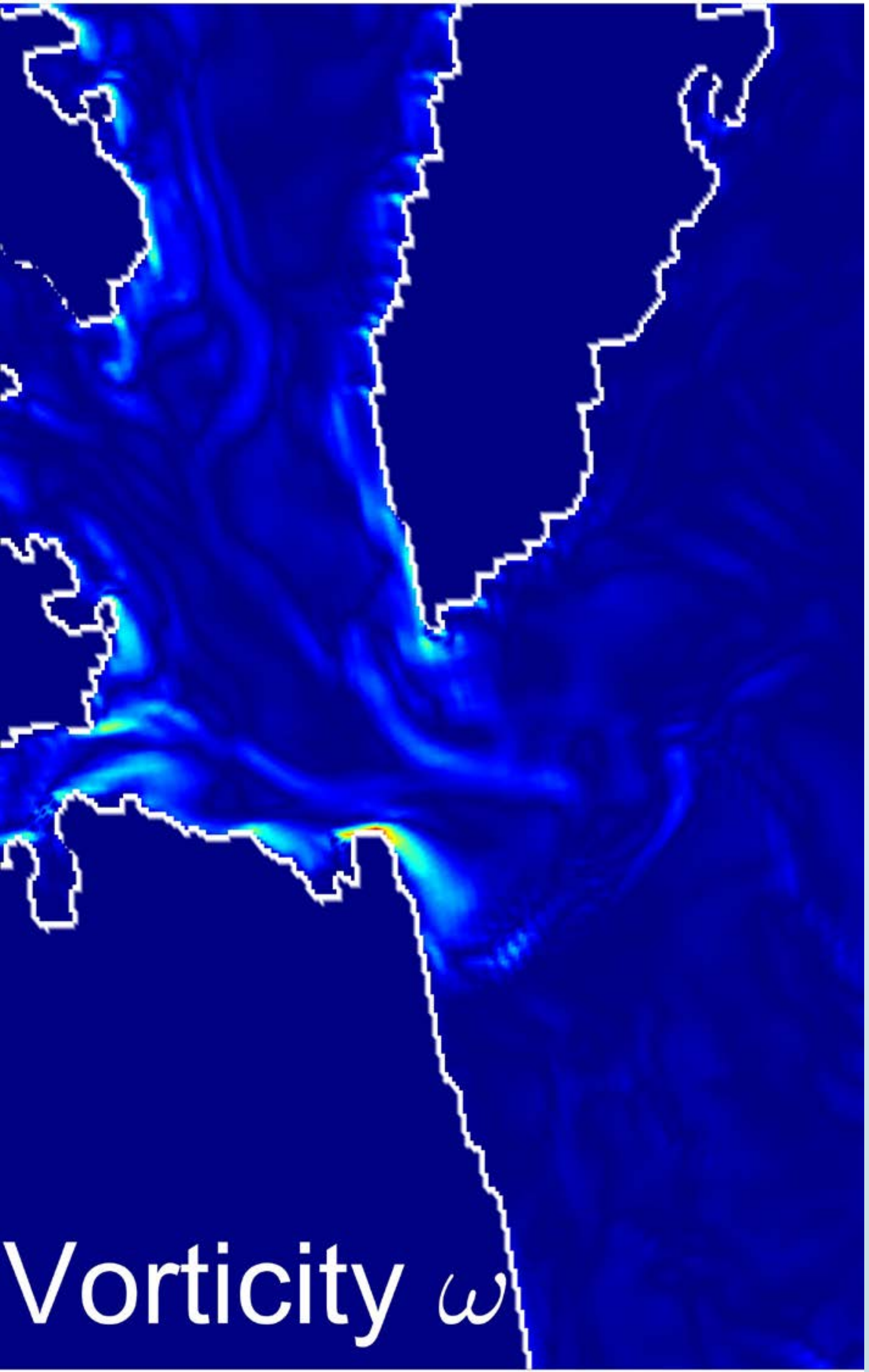




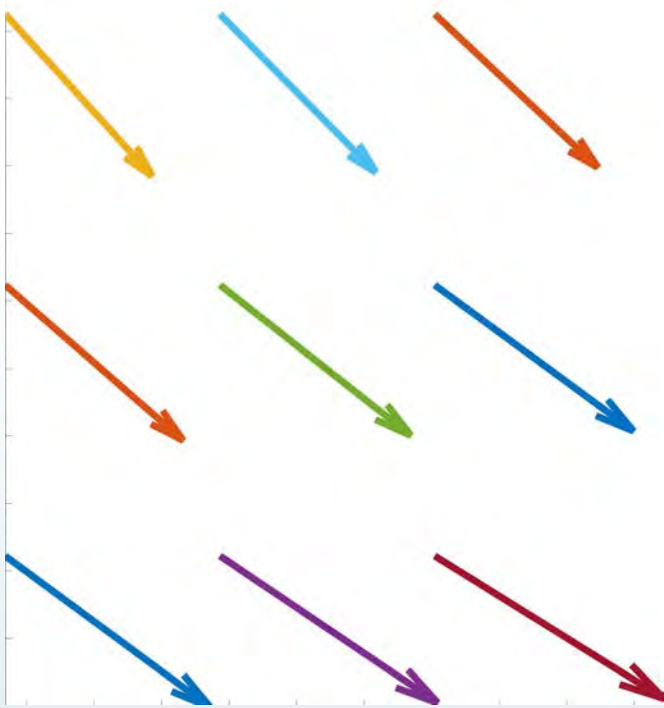


Eulerian Measure #1: Kinetic Energy

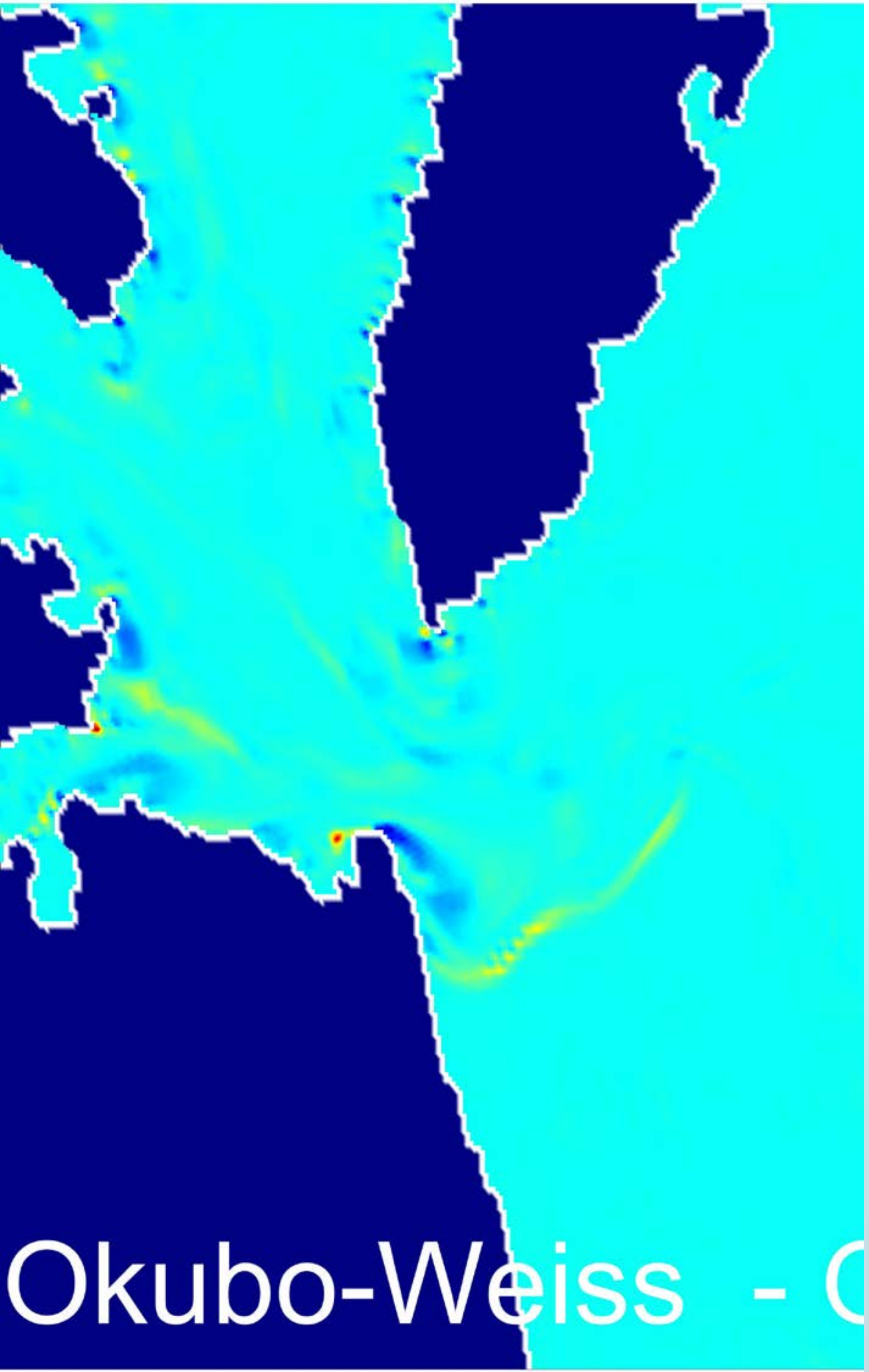
- L2-norm velocity $(magnitude^2)$
- Not the material derivative
- Overall magnitude measure



Eulerian Measure #2: Vorticity – $\vec{\omega}$



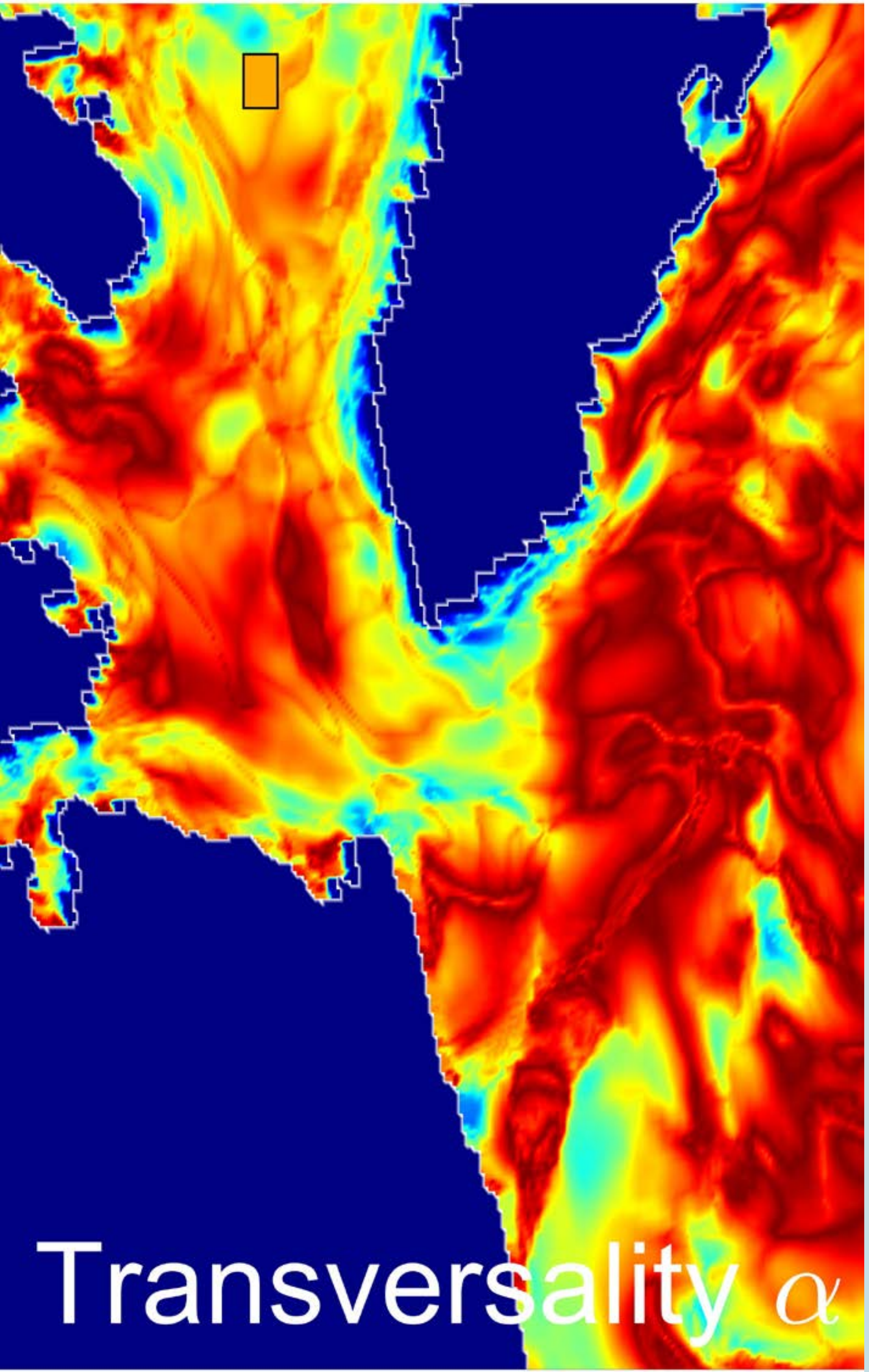
- Measure of field curvature
- Instantaneous
- Magnitude dependent
- $\vec{\omega} = \vec{\nabla} \times \vec{u}$



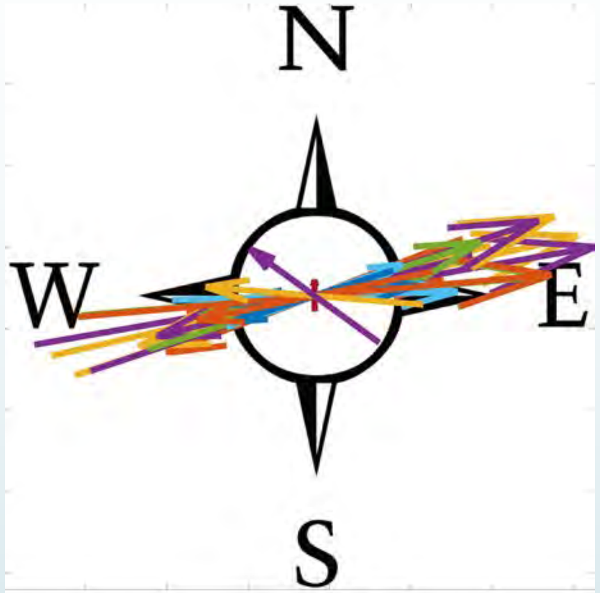
Eulerian Measure #3,4,5: Okubo-Weiss – OW

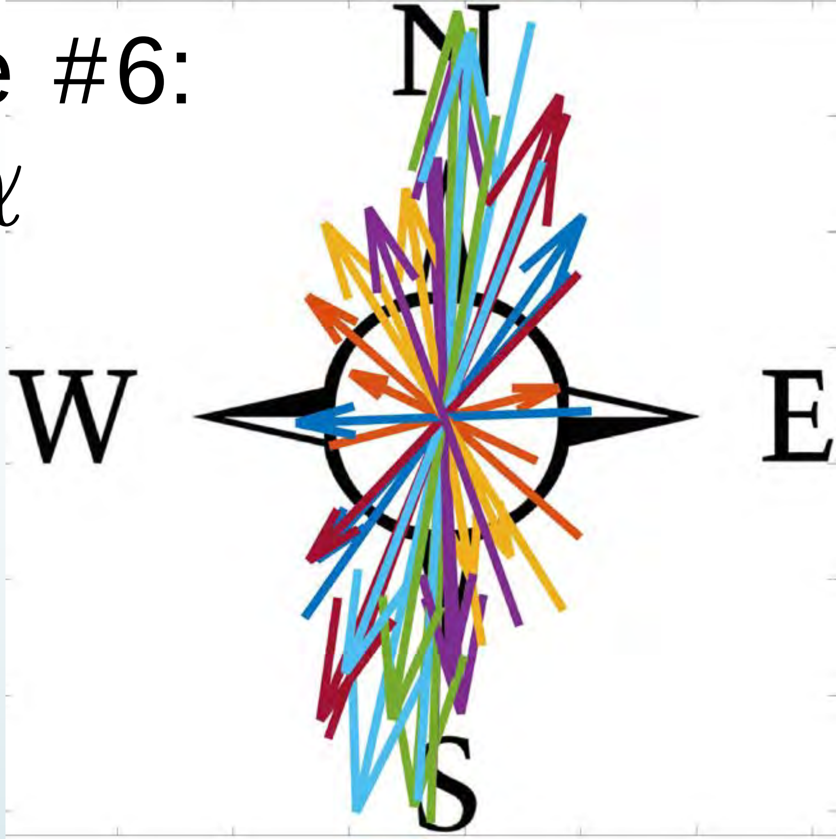
- Measure competition between strain and rotation
- Instantaneous
- Gradient dependent

Sigma^2 = shear+normal strain
(3D) Q = ½ (Omega^2-Sigma^2)

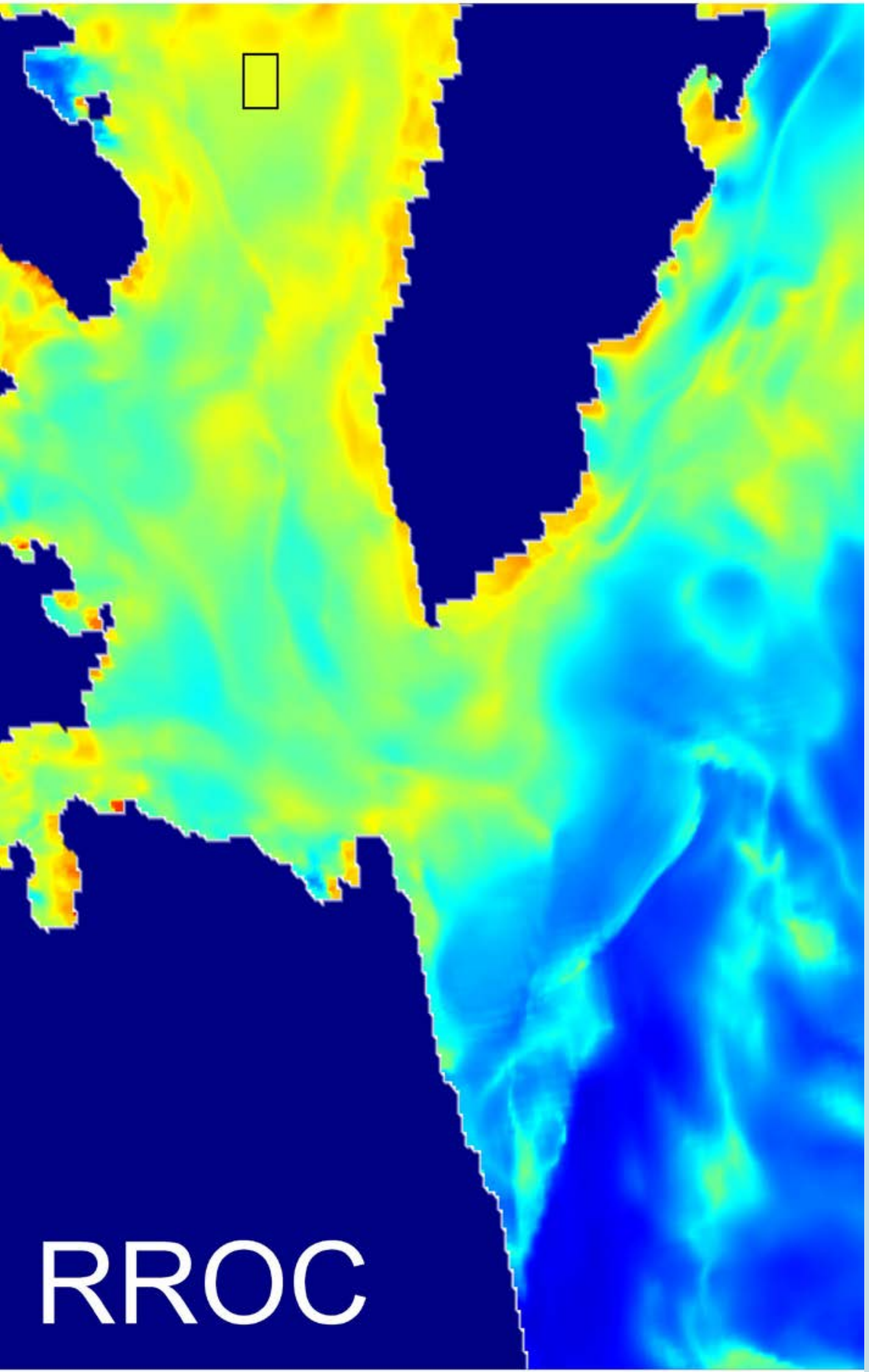


Eulerian Measure #6: Transversality – α





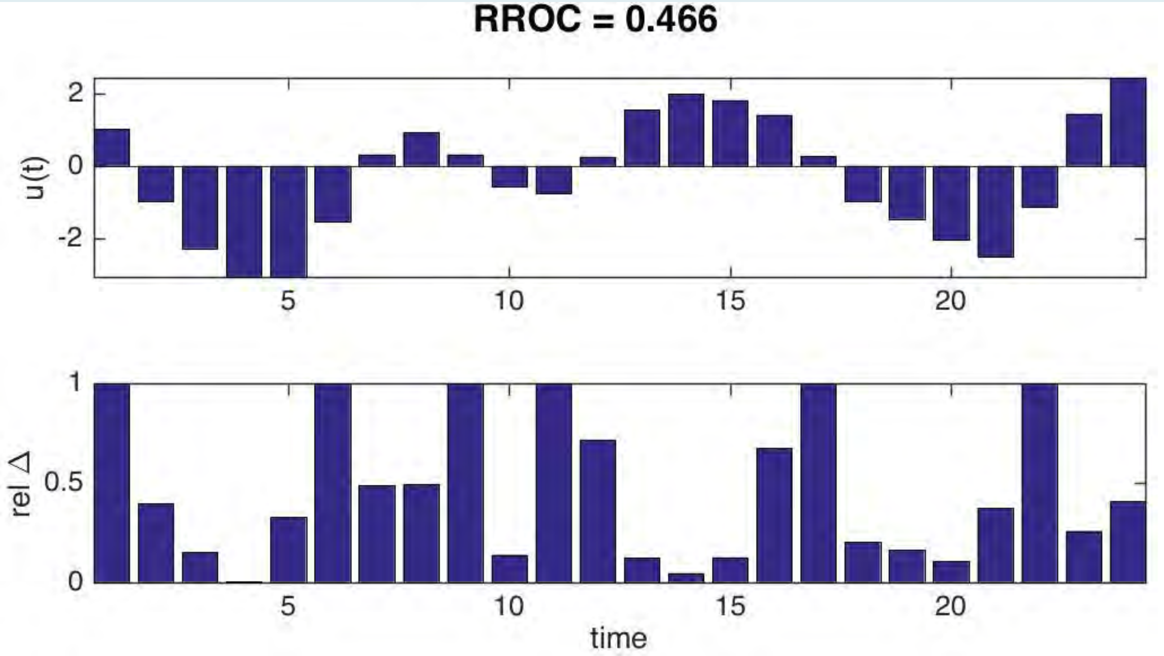
- Angular spread of velocity vs. average velocity direction
- Angles folded from 0-90
- Insensitive to magnitude
- $$\alpha = \mathcal{V}_\theta(\vec{r}) \equiv \frac{1}{T}\int_0^T (\theta(t) - \langle\theta\rangle)^2 dt$$

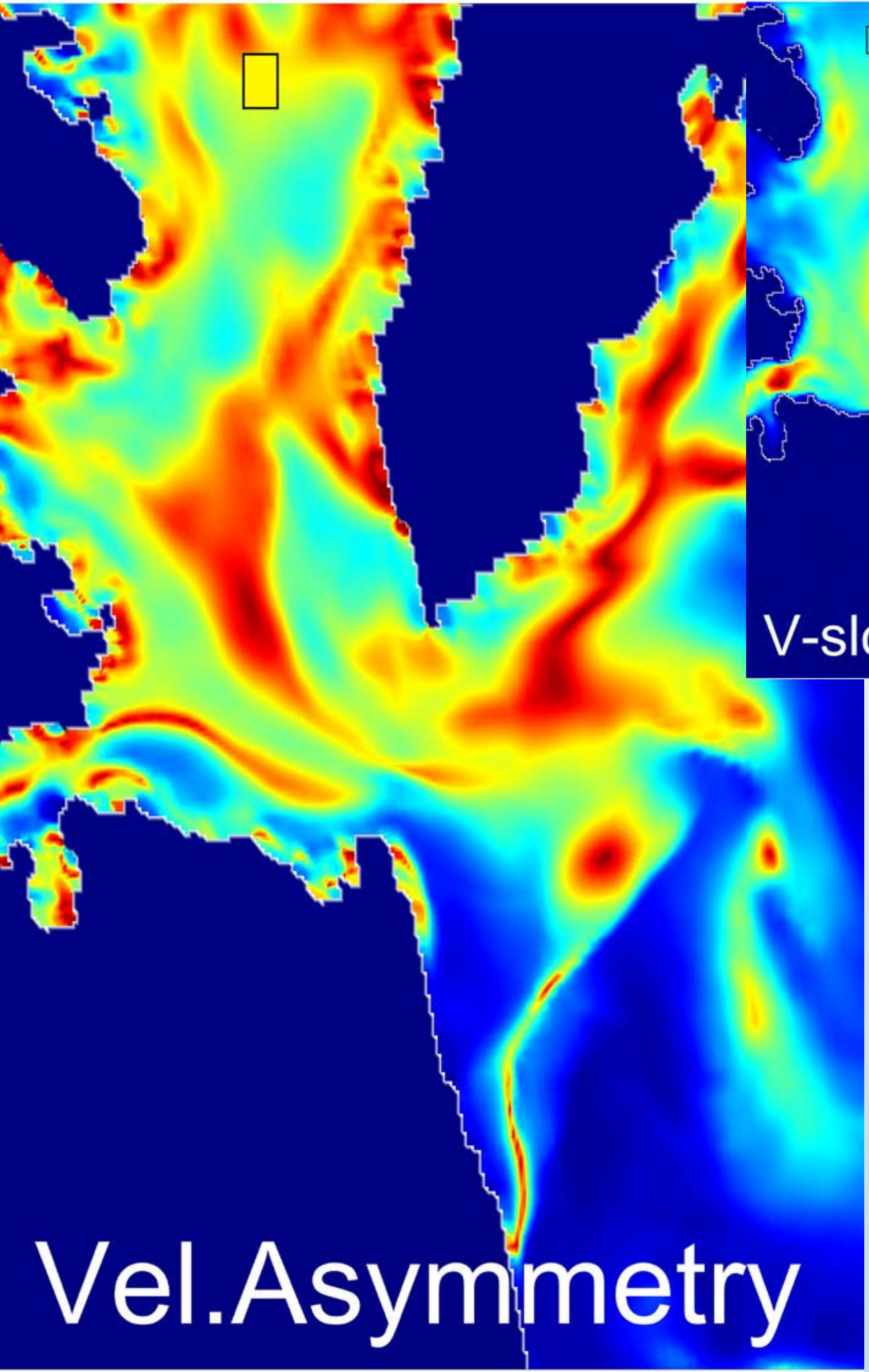


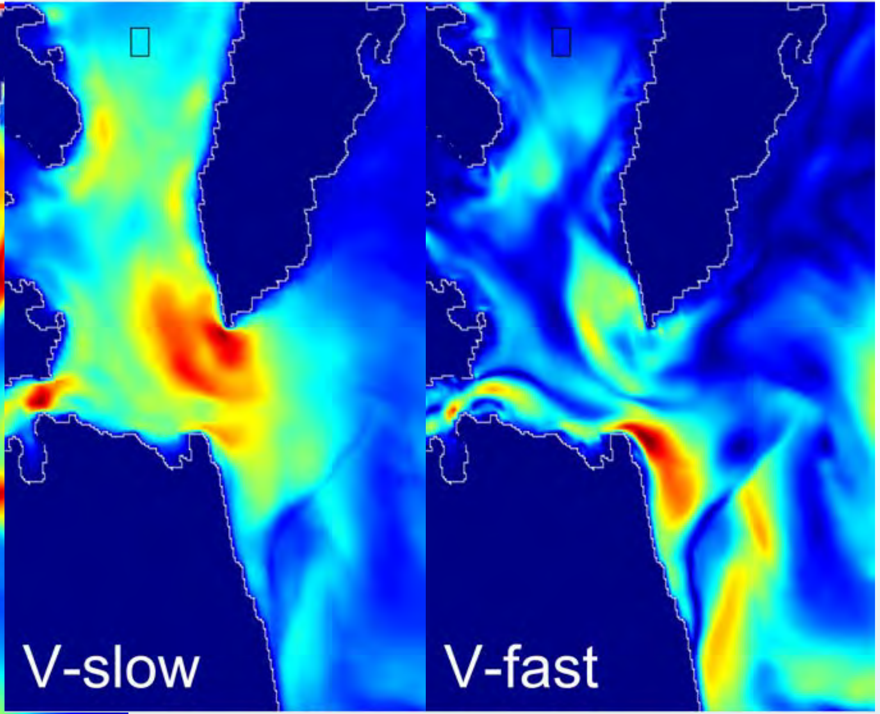
Eulerian Measure #7: Relative Rate of Change (RROC)

- Rate of jitter of a velocity vector
- Insensitive to magnitude
- Not the acceleration

$$\mathrm{RROC} = \frac{1}{T}\int_0^T \frac{\|\vec{u}(t+\Delta t) - \vec{u}(t)\|}{\|\vec{u}(t+\Delta t)\| + \|\vec{u}(t)\|}\,dt$$







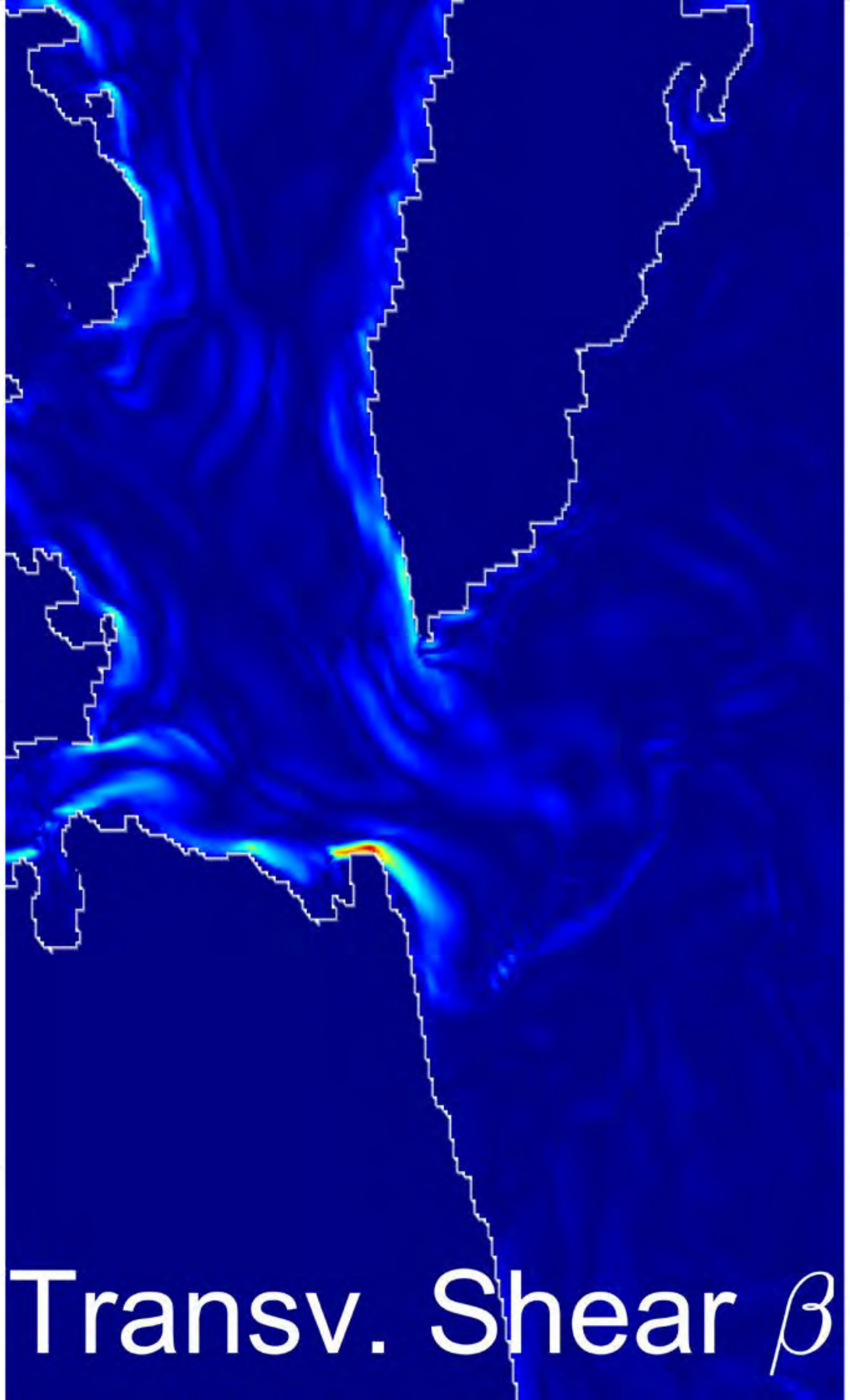
$$V_{\mathrm{slow}} = \left\langle |\vec{\mathbf{u}}(t)| \right\rangle = \frac{1}{T}\int_0^T |\vec{\mathbf{u}}(t)|dt$$

$$V_{\mathrm{fast}} = \left| \langle \vec{\mathbf{u}}(t) \rangle \right| = \left| \frac{1}{T}\int_0^T \vec{\mathbf{u}}(t)dt \right|$$

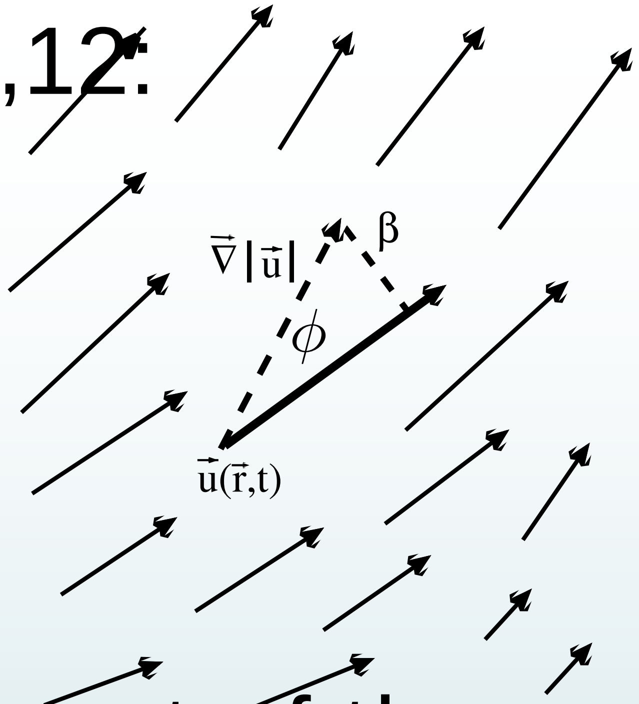
$$\text{vel asymmetry} = \frac{V_{slow} - V_{fast}}{V_{slow} + V_{fast}}$$

Eulerian Measure #8,9,10: Velocity – Slow, Fast, Asymmetry

- V-Slow – average of velocity magnitude
- V-Fast – magnitude of velocity average
- Vel-asymmetry – relative degree of frustrated transport
- Vel-asym – bounded from 0 to 1



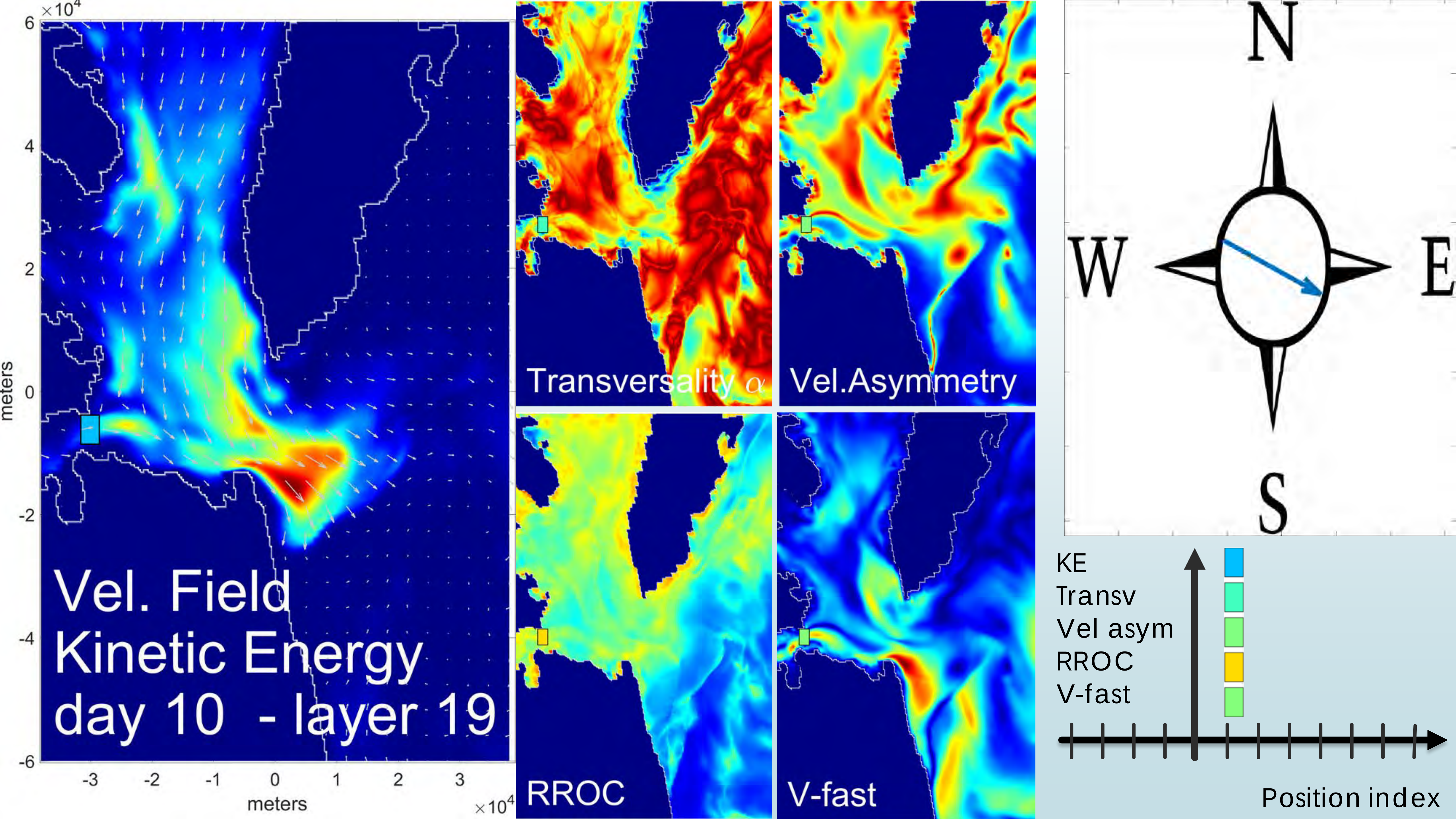
Eulerian Measure #11,12: Transverse Shear – β

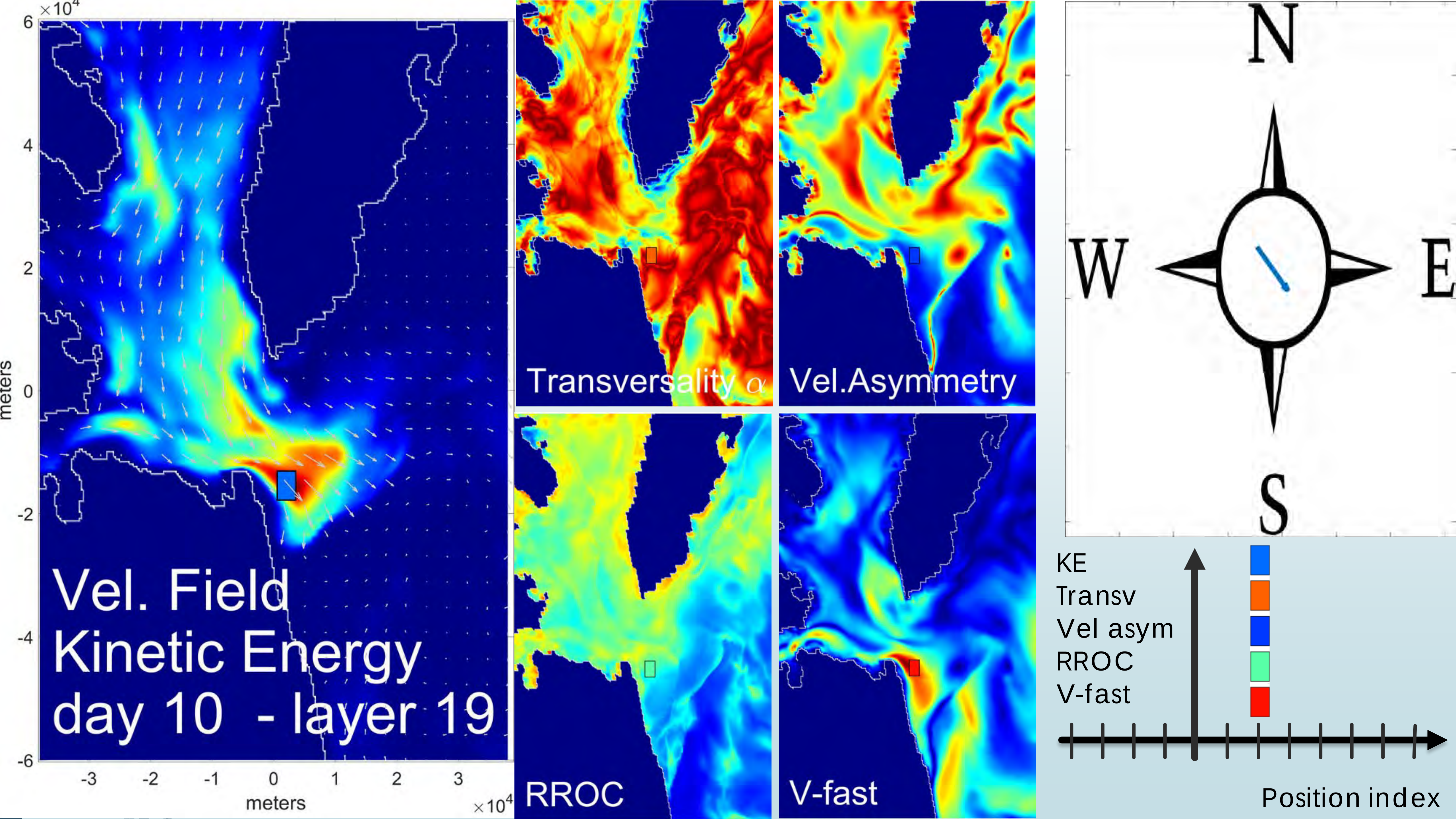


- Transverse component of the spatial gradient of the velocity magnitude
- $$\boldsymbol{\rho}(\mathbf{r},t) = \vec{\nabla}|\mathbf{u}(\mathbf{r},t)|$$

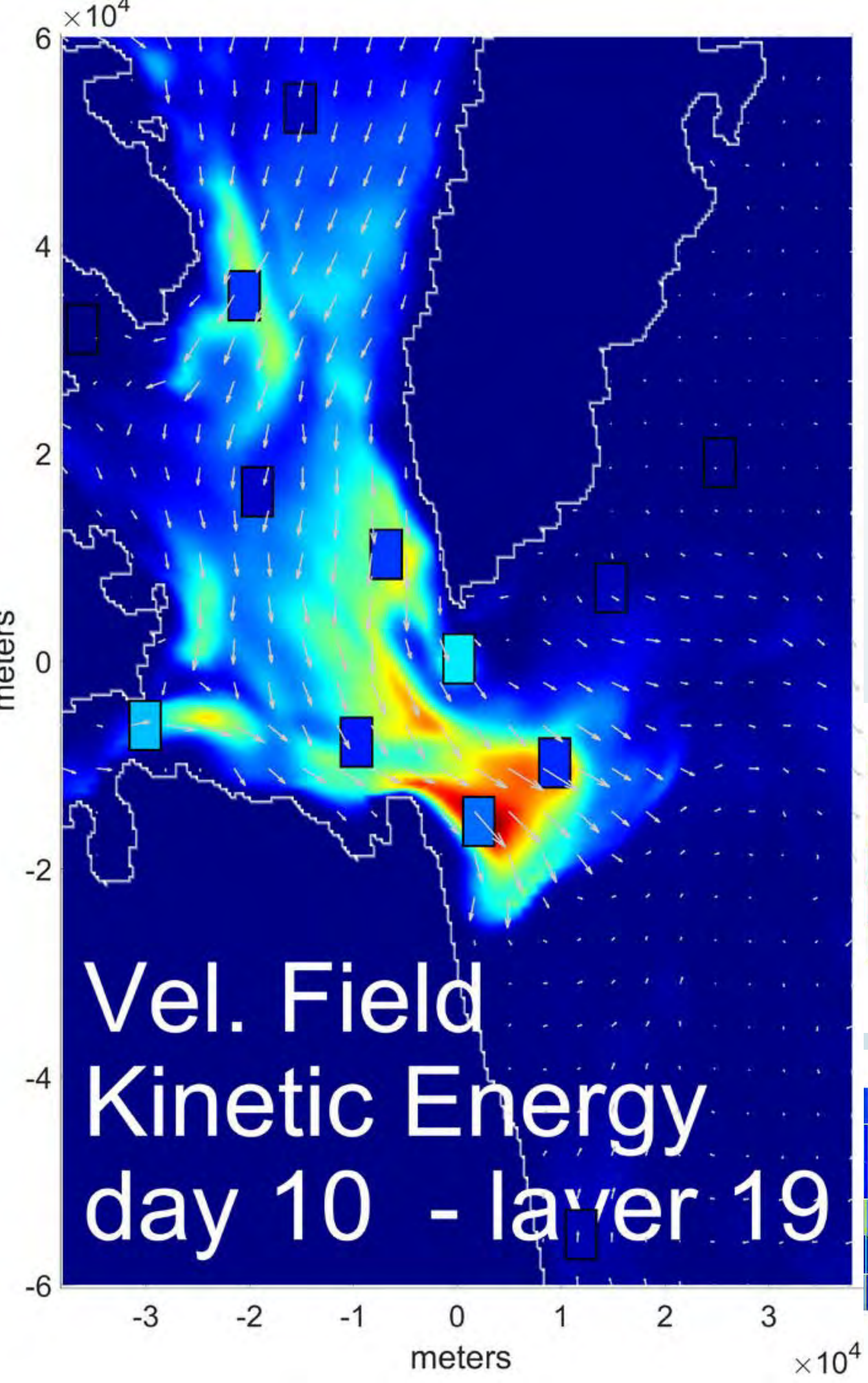
$$\boldsymbol{\rho}(\mathbf{r},t)\cdot\hat{\mathbf{v}} = |\boldsymbol{\rho}(\mathbf{r},t)|\; cos(\phi)$$

$$\beta(\mathbf{r},t) = |\boldsymbol{\rho}(\mathbf{r},t)|\; sin(\phi)$$


Vel. Field
Kinetic Energy
day 10 - layer 19
meters
meters
×10^4
×10^4
Transversality α
Vel.Asymmetry
RROC
V-fast
N
S
W
E
KE
Transv
Vel asym
RROC
V-fast
Position index

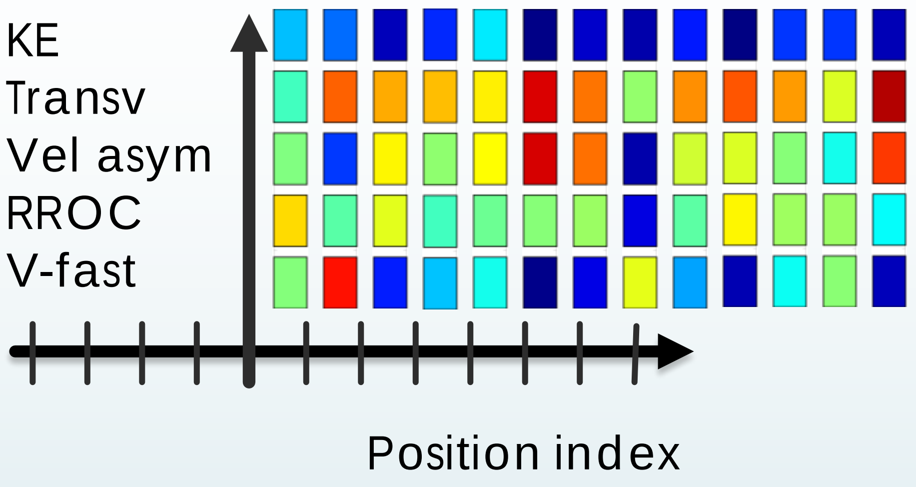


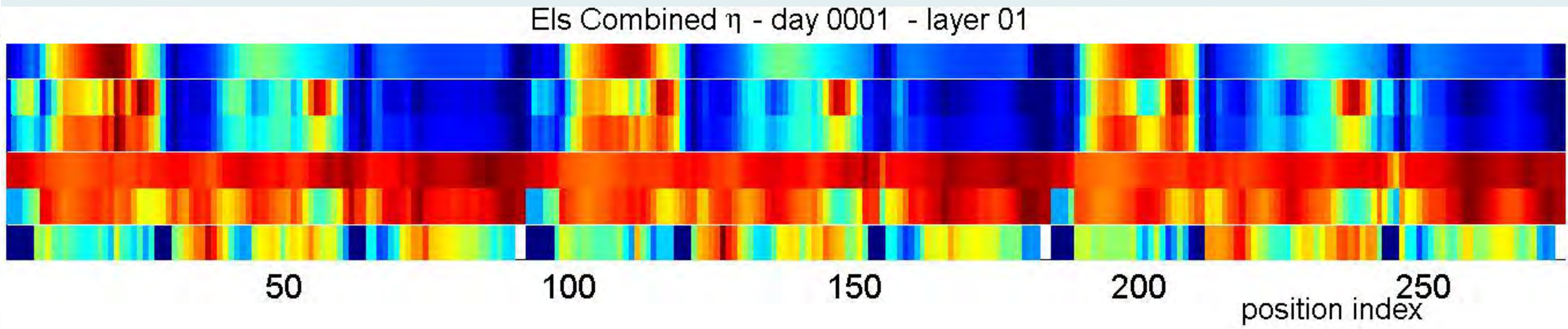
Vel. Field
Kinetic Energy
day 10 - layer 19
meters
meters
Transversality α
Vel.Asymmetry
RROC
V-fast
N
W
E
S
KE
Transv
Vel asym
RROC
V-fast
Position index

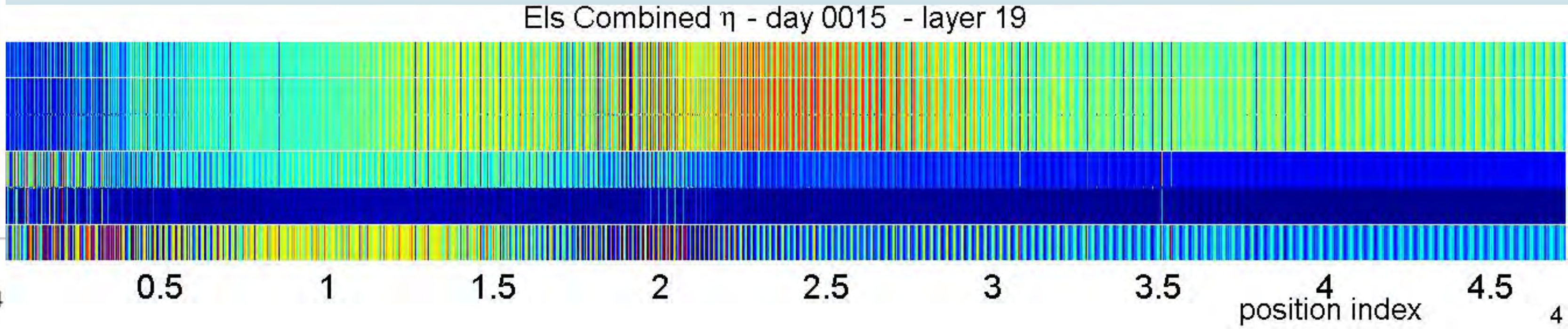


"DNA" Plots

- Shows N-dim data in one plot per location per time
- Exposes Data Clusters

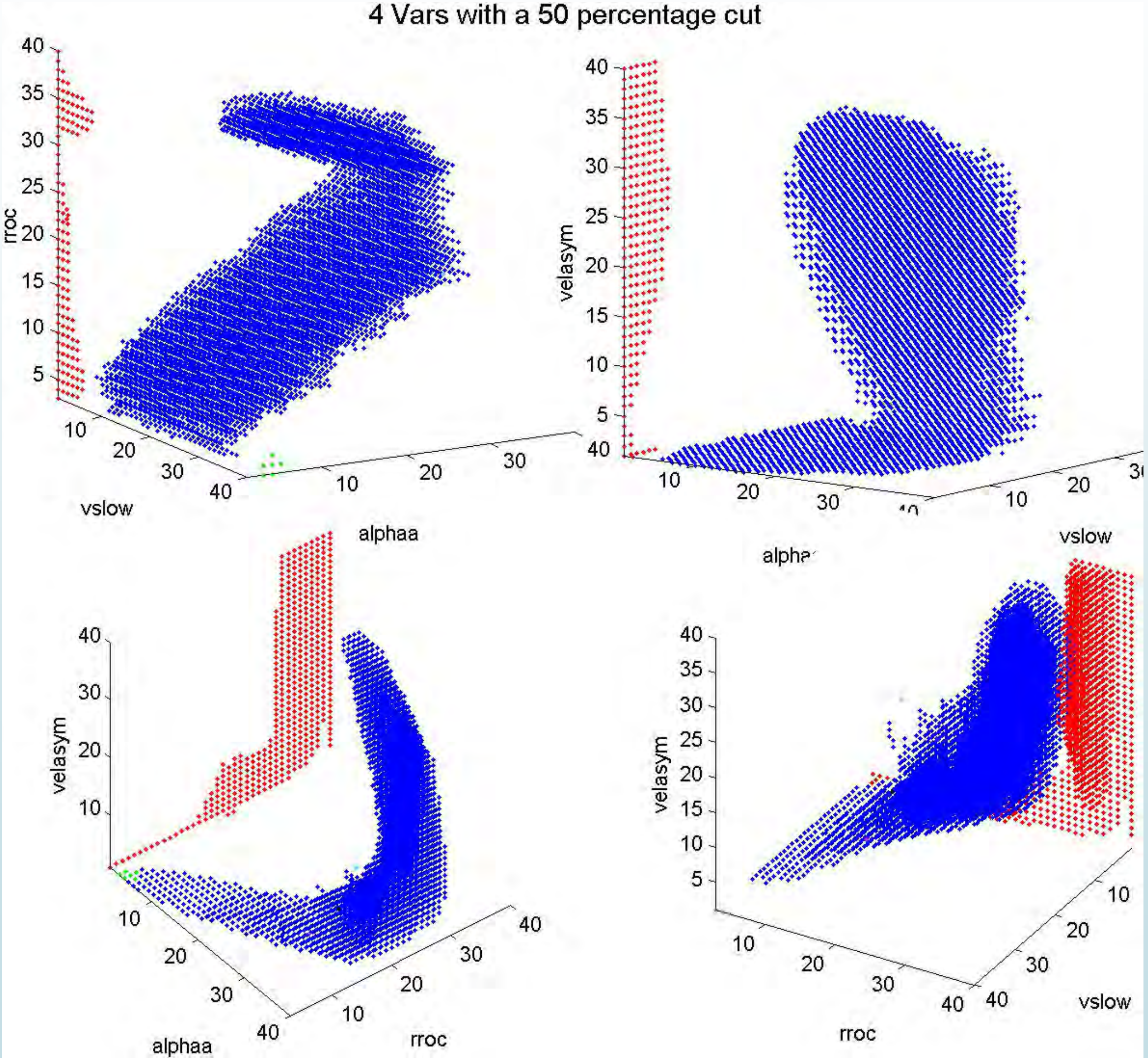






N-dimensional Data as a Point Cloud

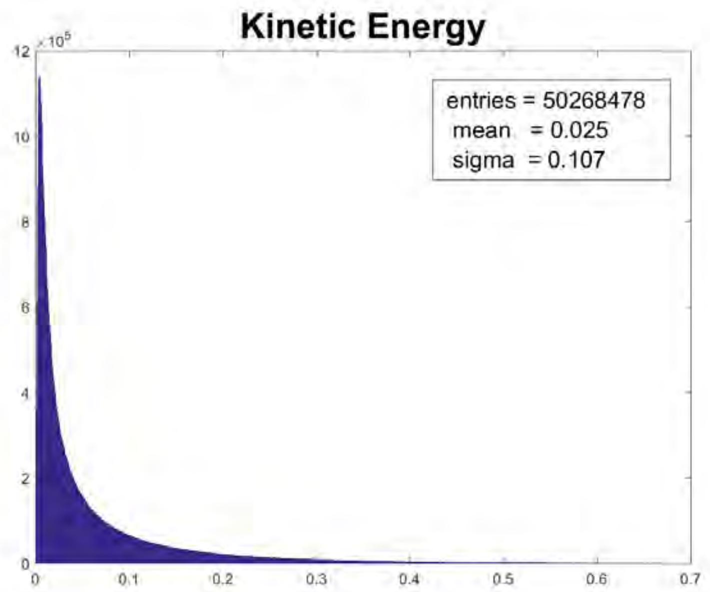
- Visualize: 1D, 2D, 3D, 4D … end of the road

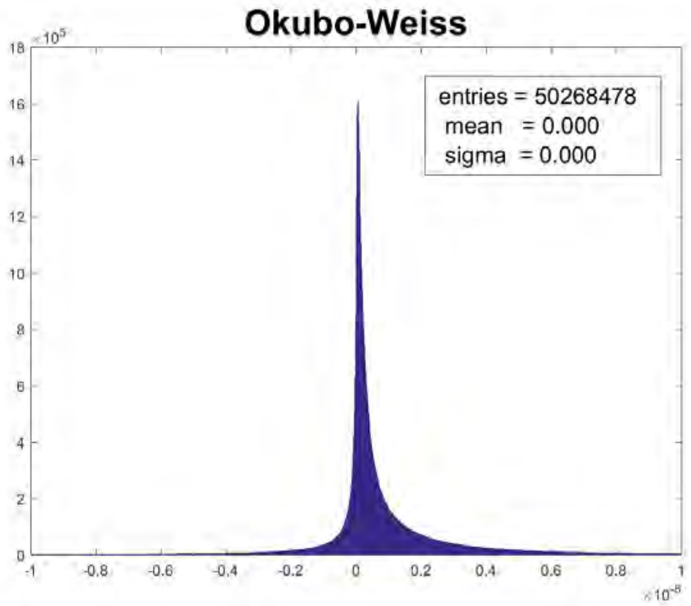


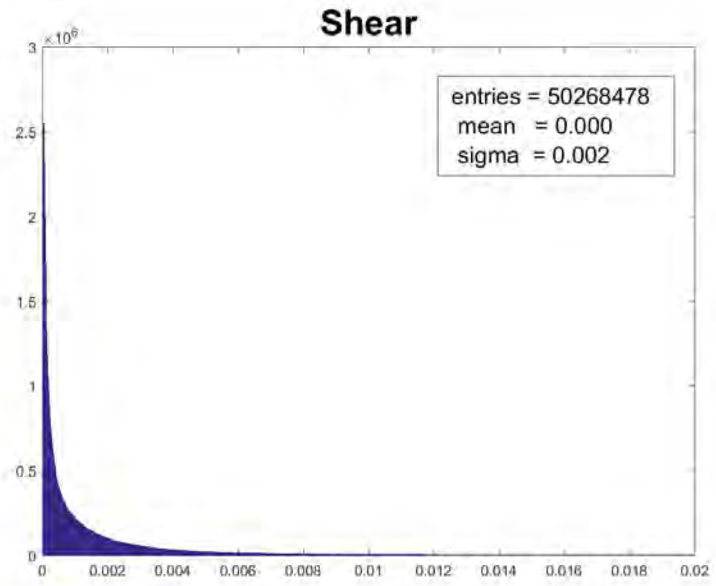
And now for something completely different

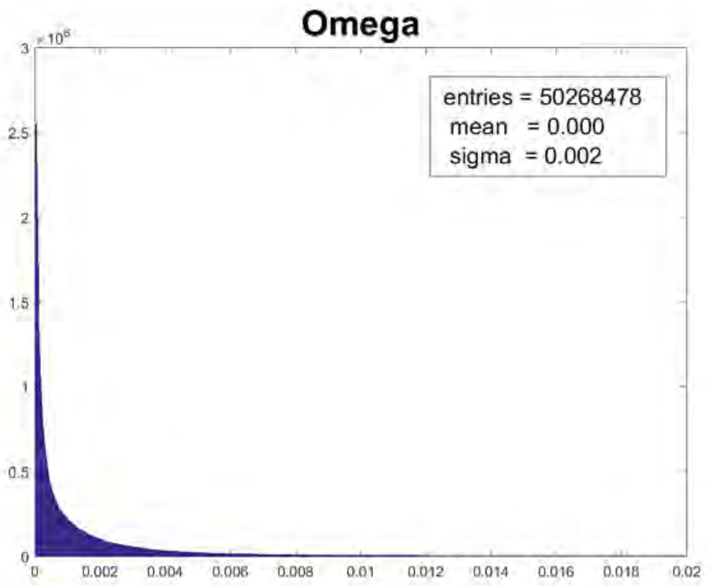
- Each location, each time consider as data
- 63million data in total for 2006 Chesapeake Bay Mouth
- Tend to think of data as geo-referenced
- Deck of cards
- Shuffle the cards
- Histogram the data
- Look for re-occuring patterns within the SPace of Eulerian MeasureS (SPEMS)
- ***Flow types*** categorized by these patterns

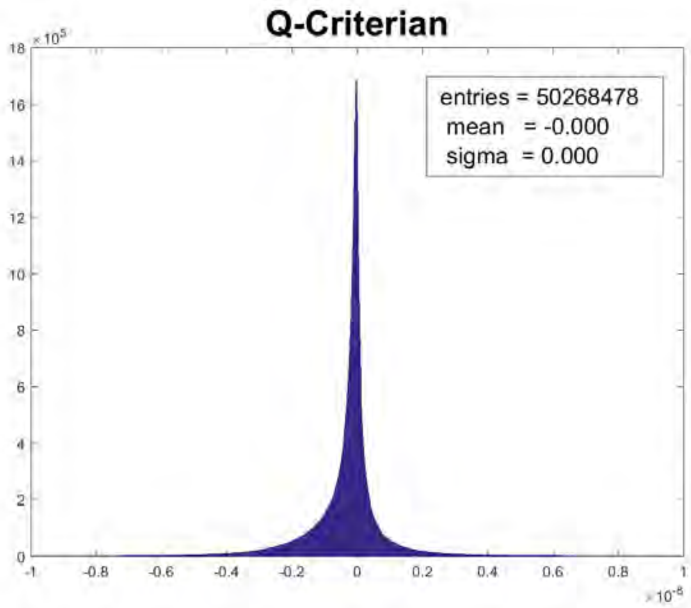
Eulerian Measures

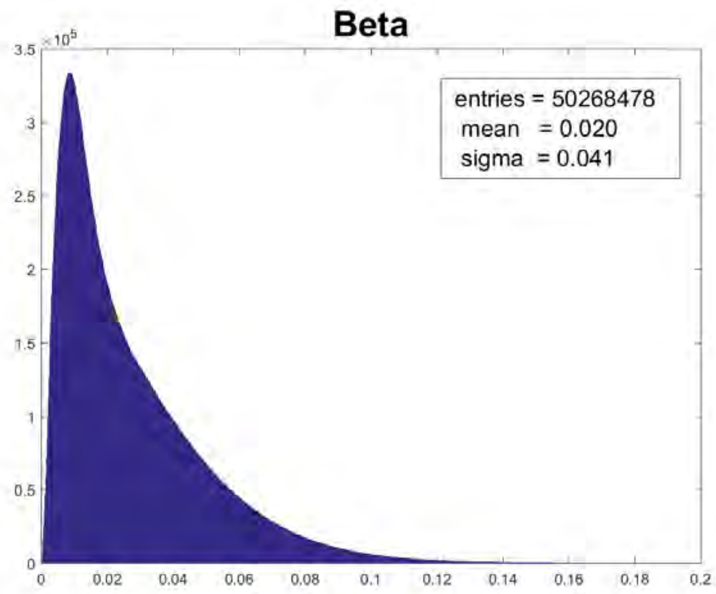




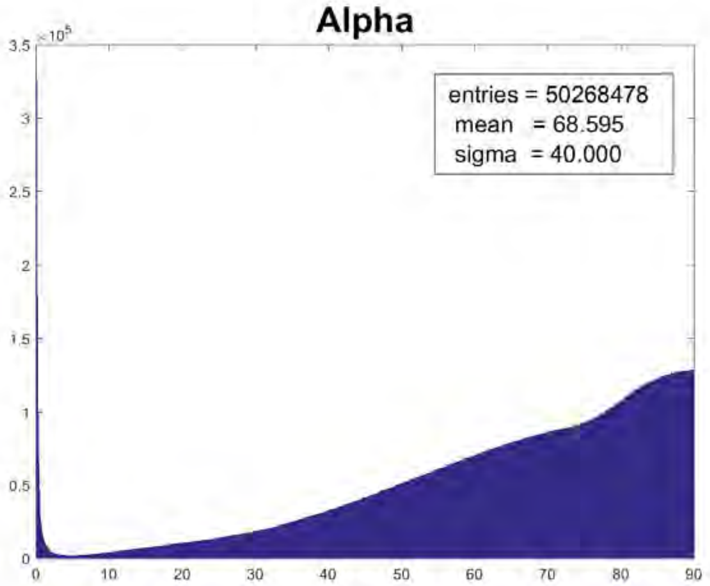


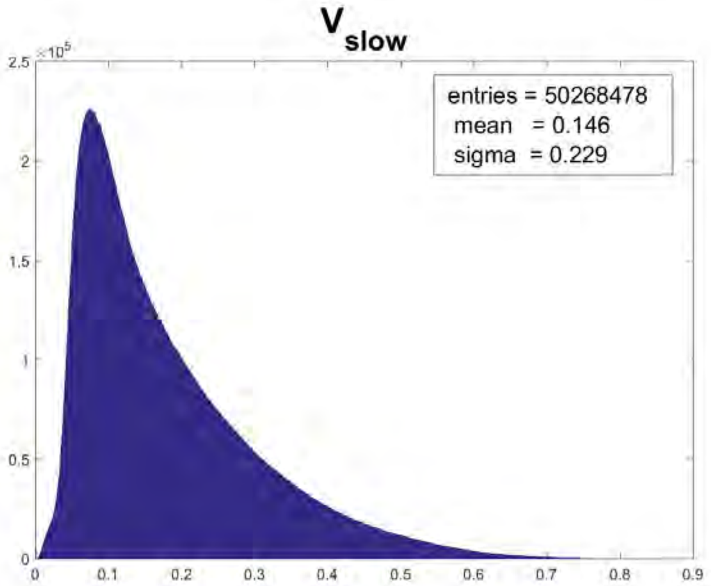


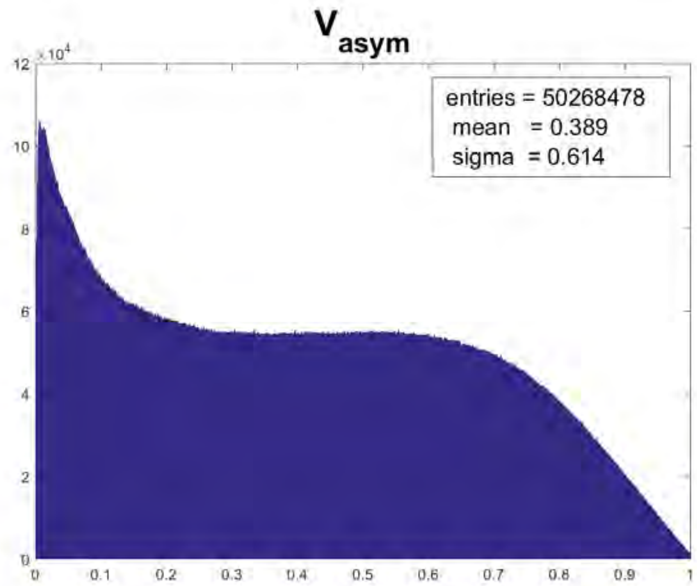


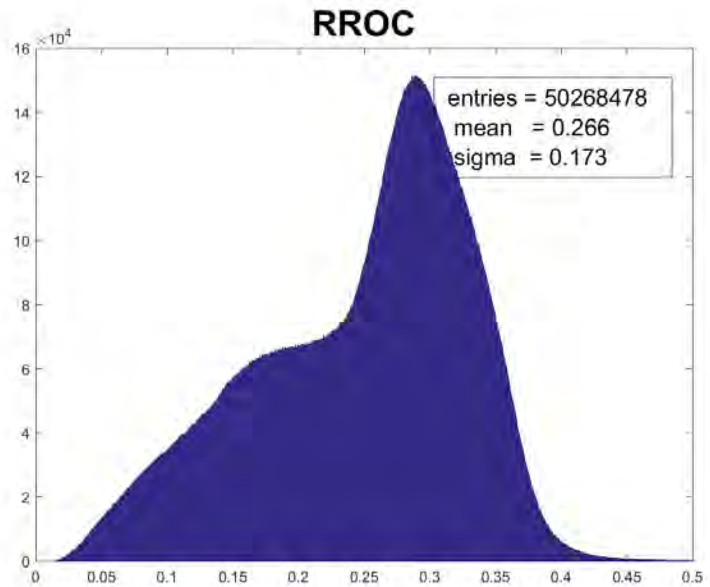


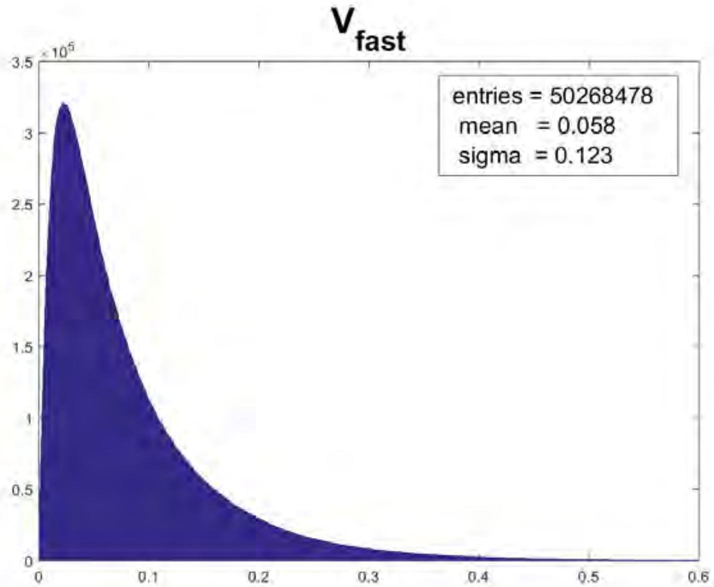
Eulerian Measures

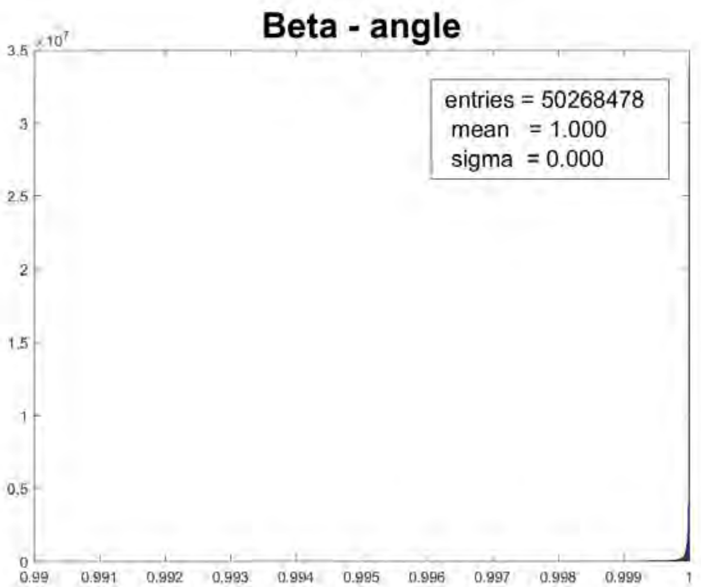








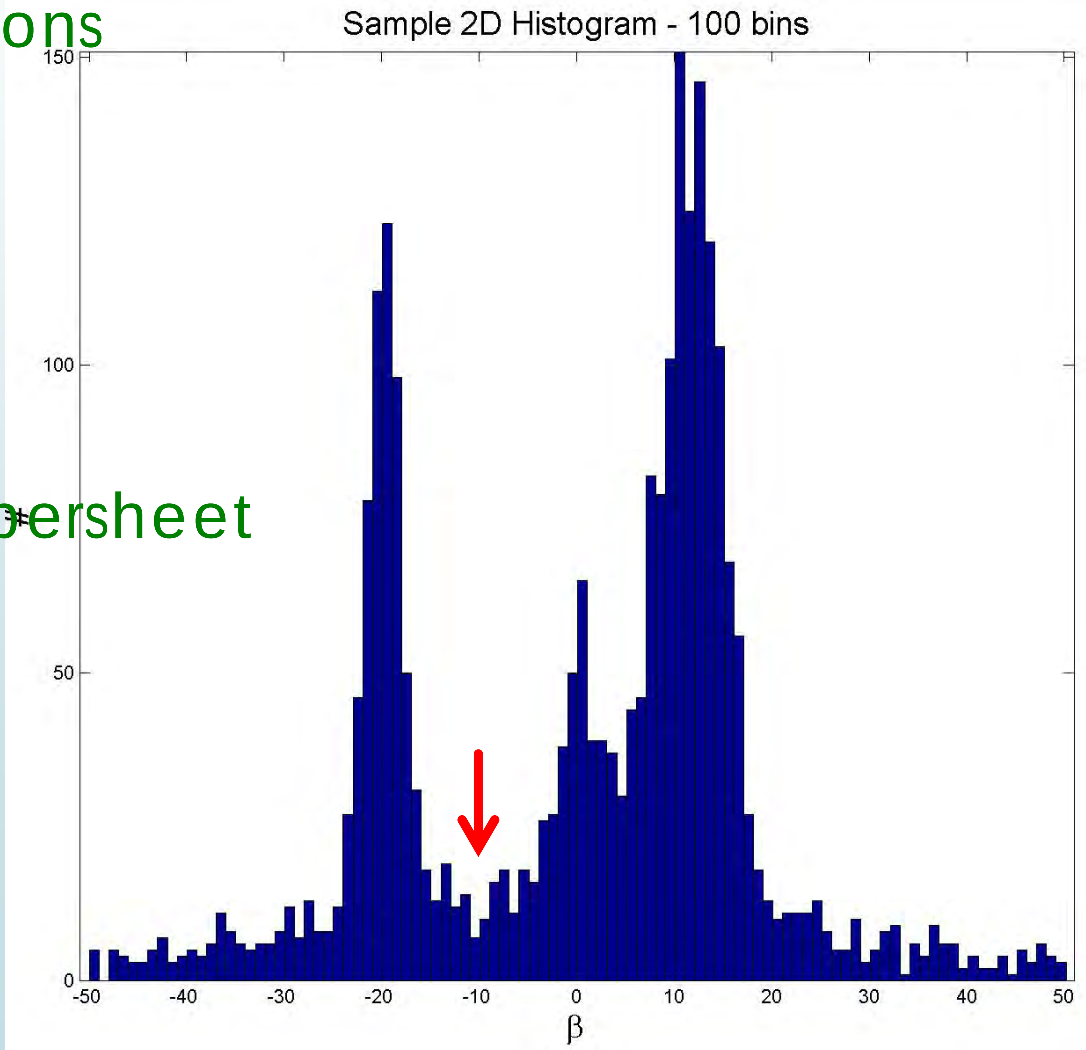




Data Analysis – Histograms and Cuts

Collect Data

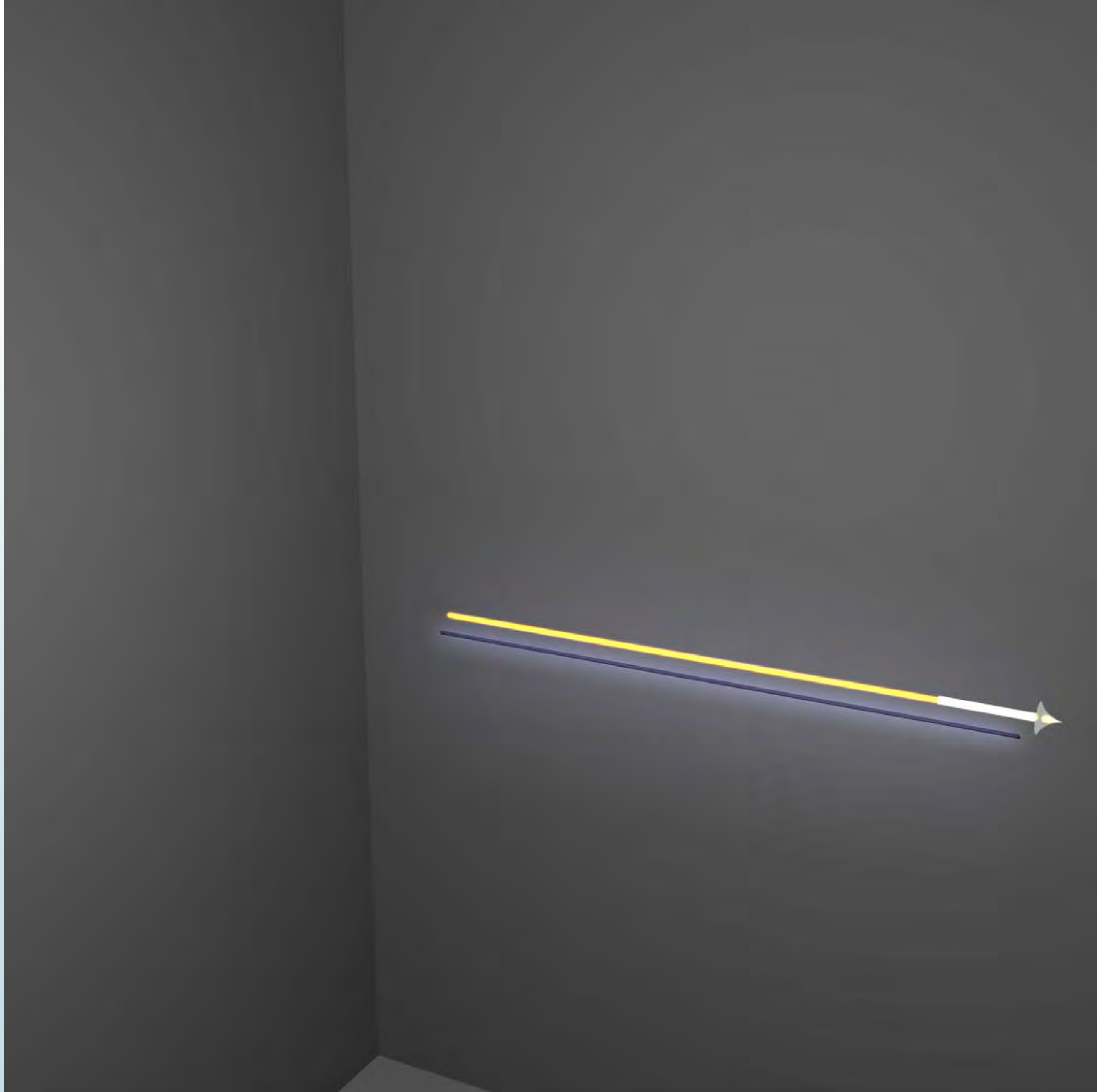
- Typically, we histogram versus some variable
- If LUCKY, we see separated distributions
- Apply a “cut” (threshold) to value to separate and classify data
- f(x,y) – 2 dimensional
- f(x,y) = C – 1-dim curve
- f(x,y,z,…) = C – (N-1) dimensional hypersheet



- Instead of partitioning data via cuts to isolate features
- Seek data clusters directly within the space

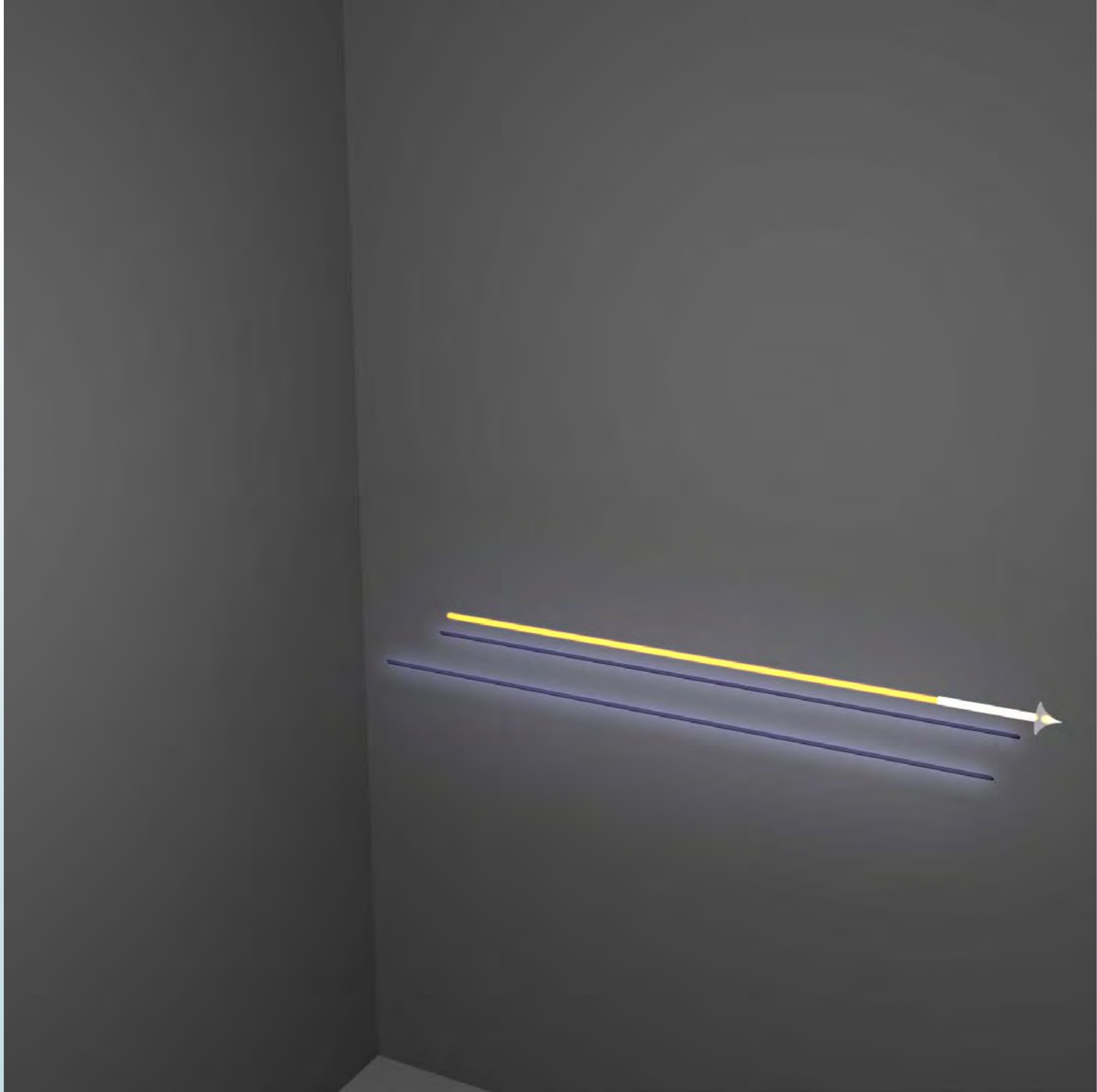
Building a space that contains data:

Start with one dimension



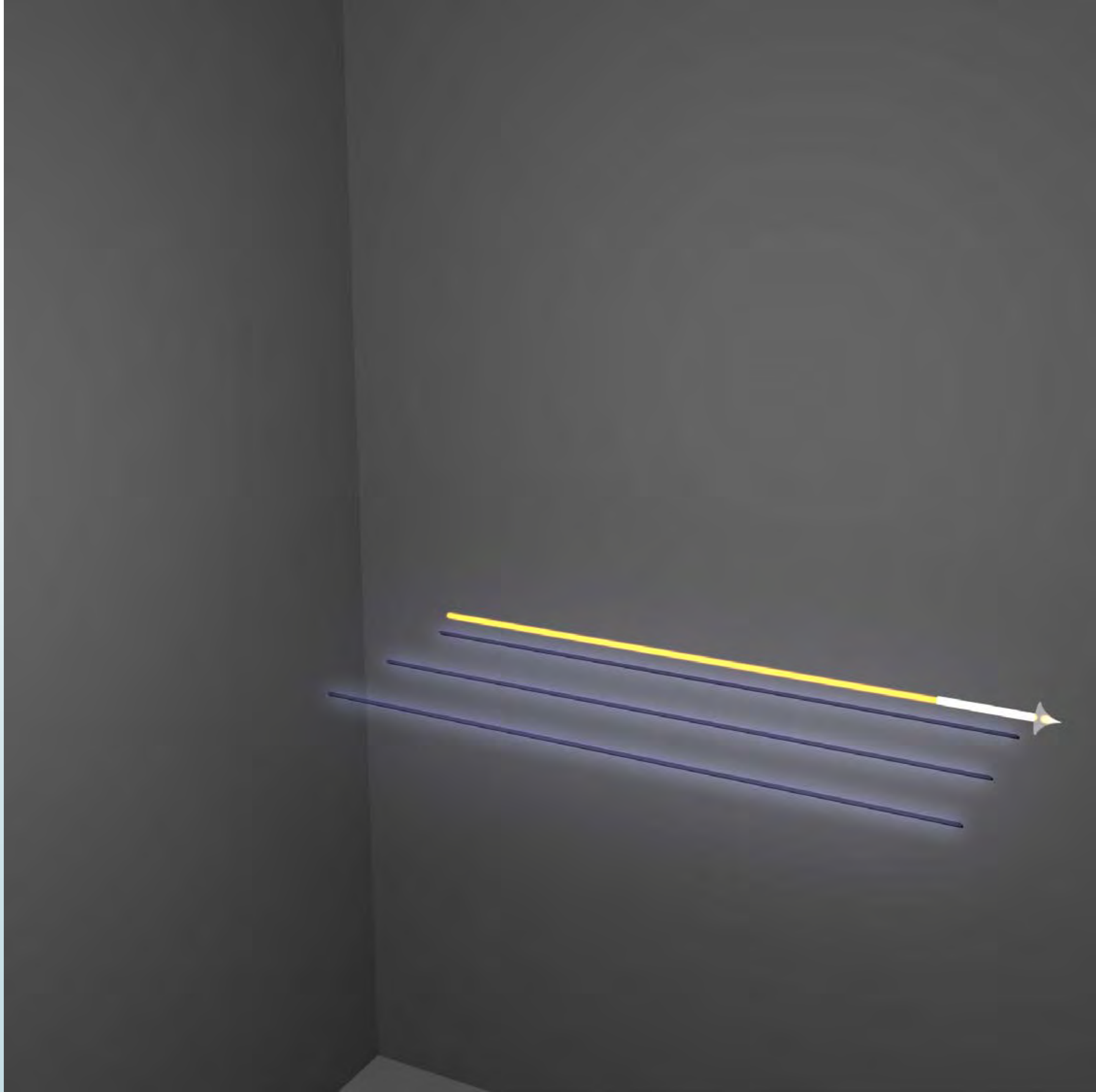
Building a space that contains data:

Start with one dimension



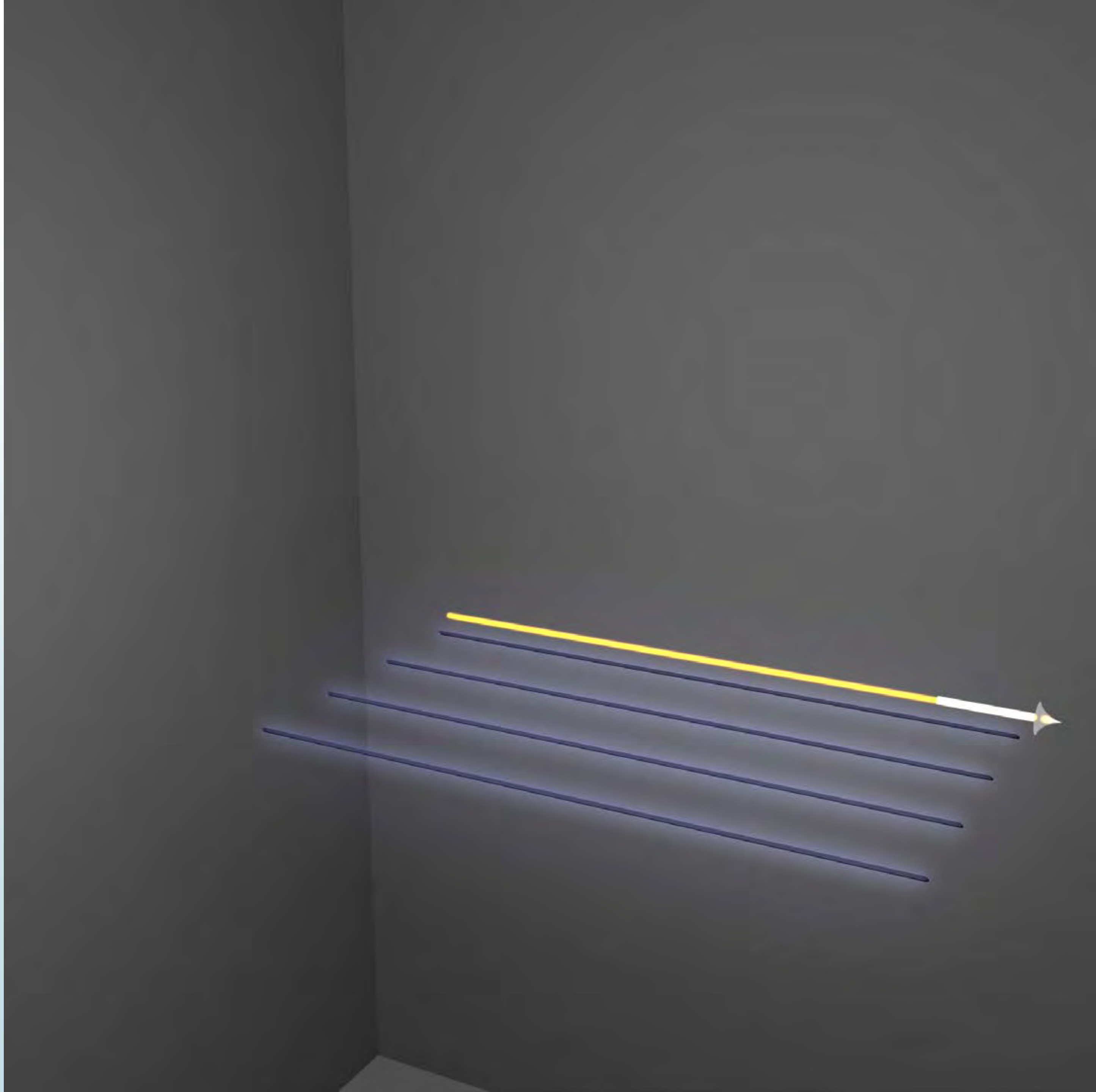
Building a space that contains data:

Start with one dimension



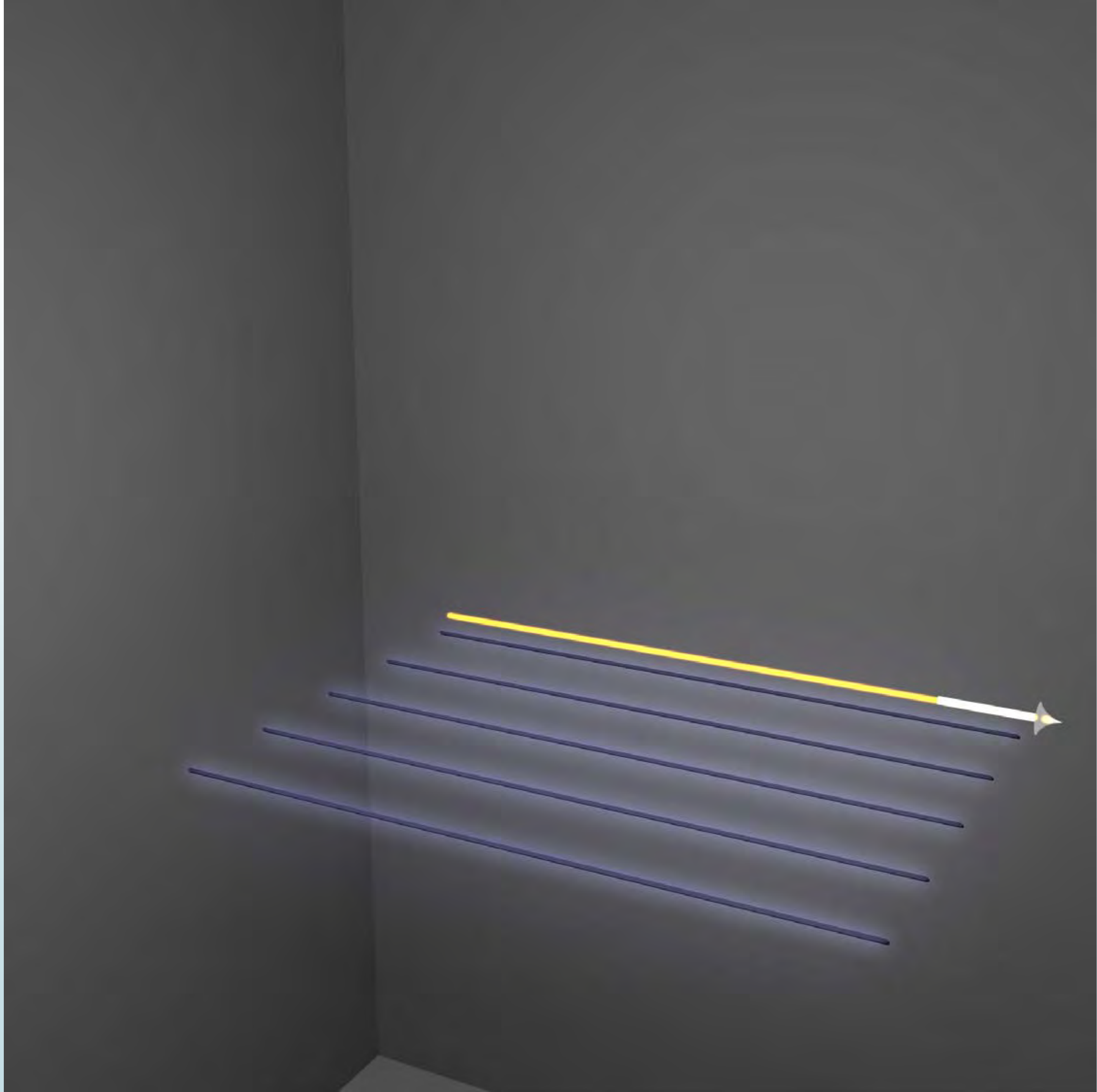
Building a space that contains data:

Start with one dimension



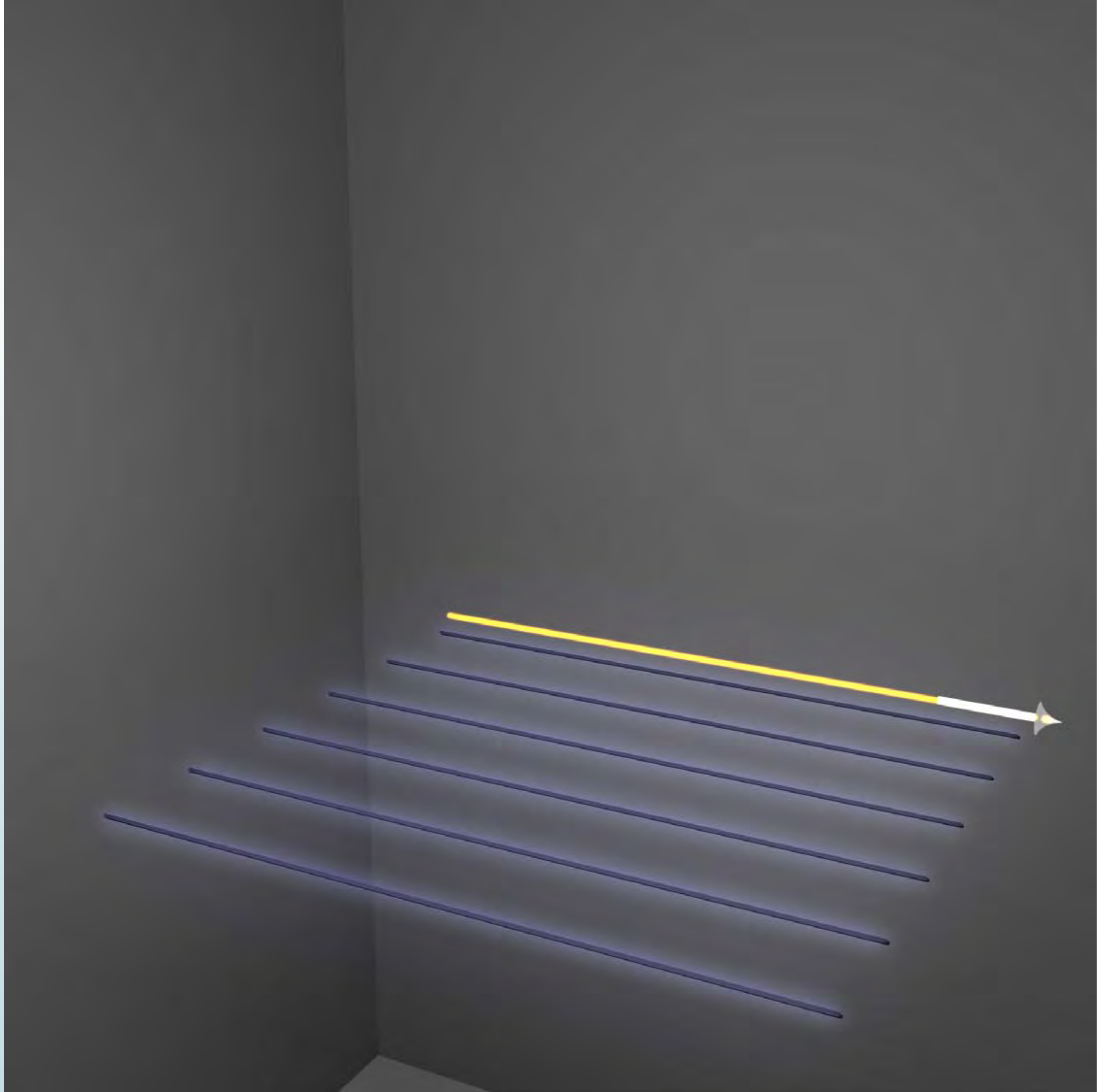
Building a space that contains data:

Start with one dimension



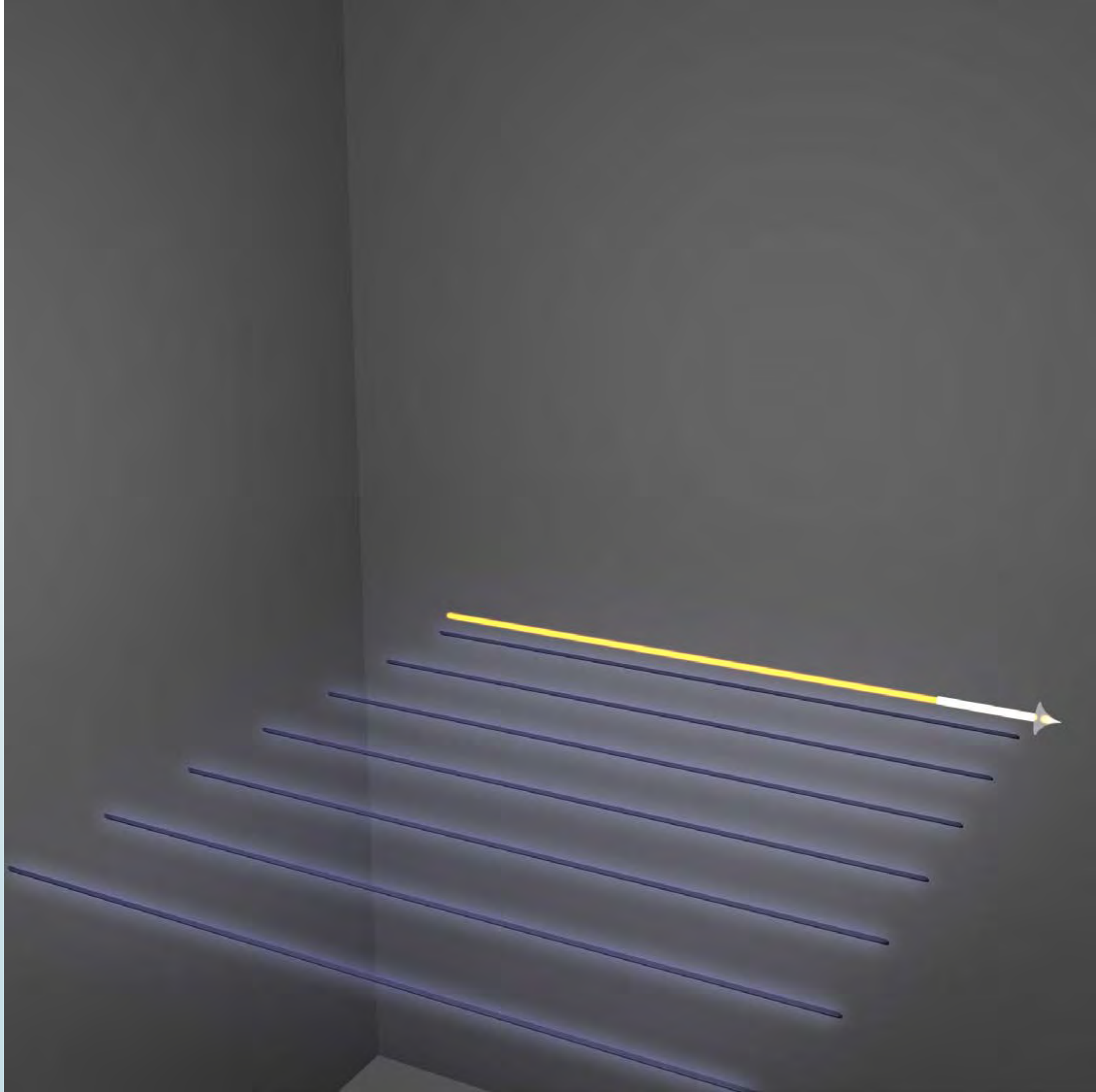
Building a space that contains data:

Start with one dimension



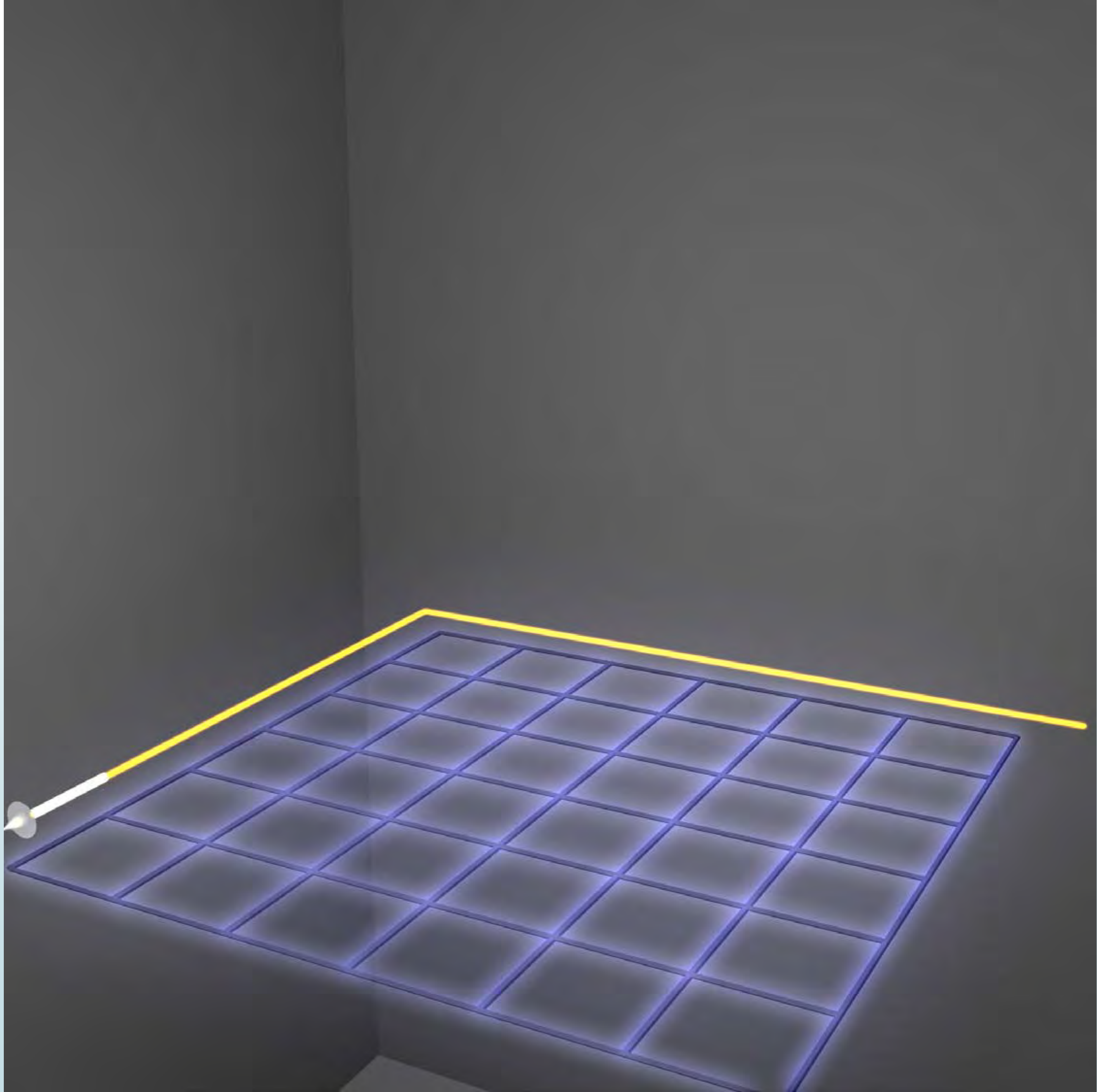
Building a space that contains data:

Start with one dimension



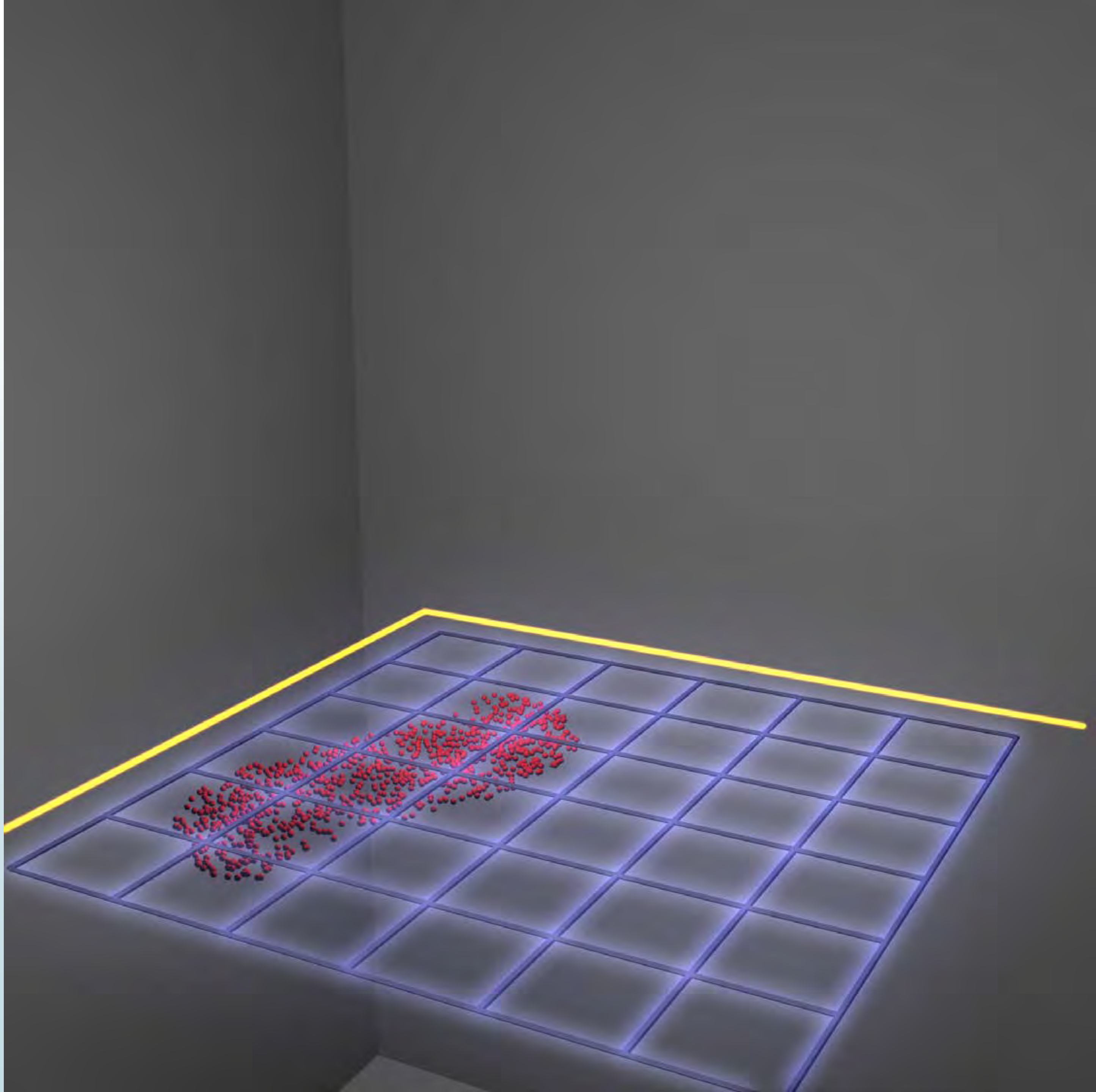
Building a space that contains data:

Now add another dimension (2D)



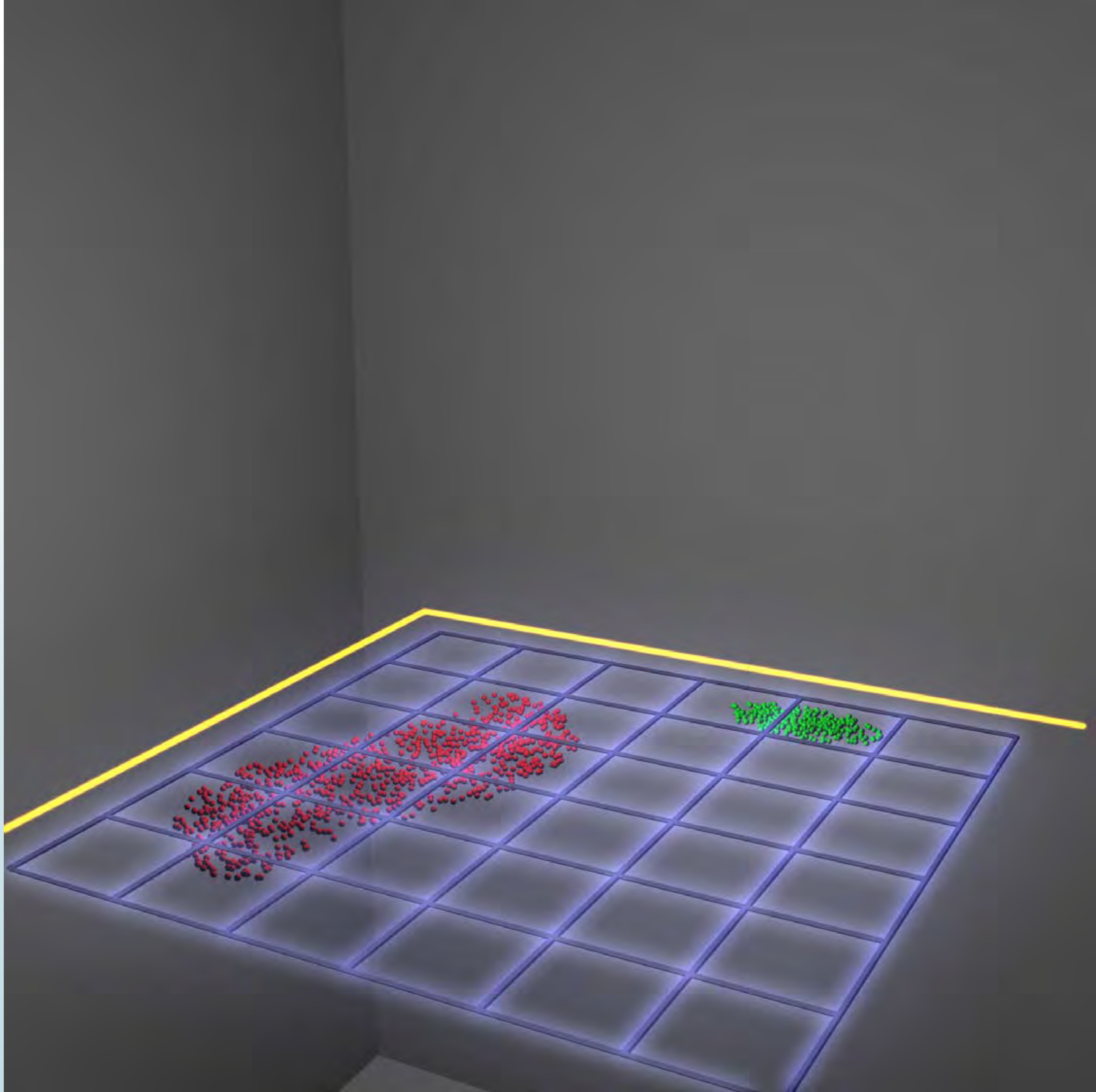
Building a space that contains data:

Add some data



Building a space that contains data:

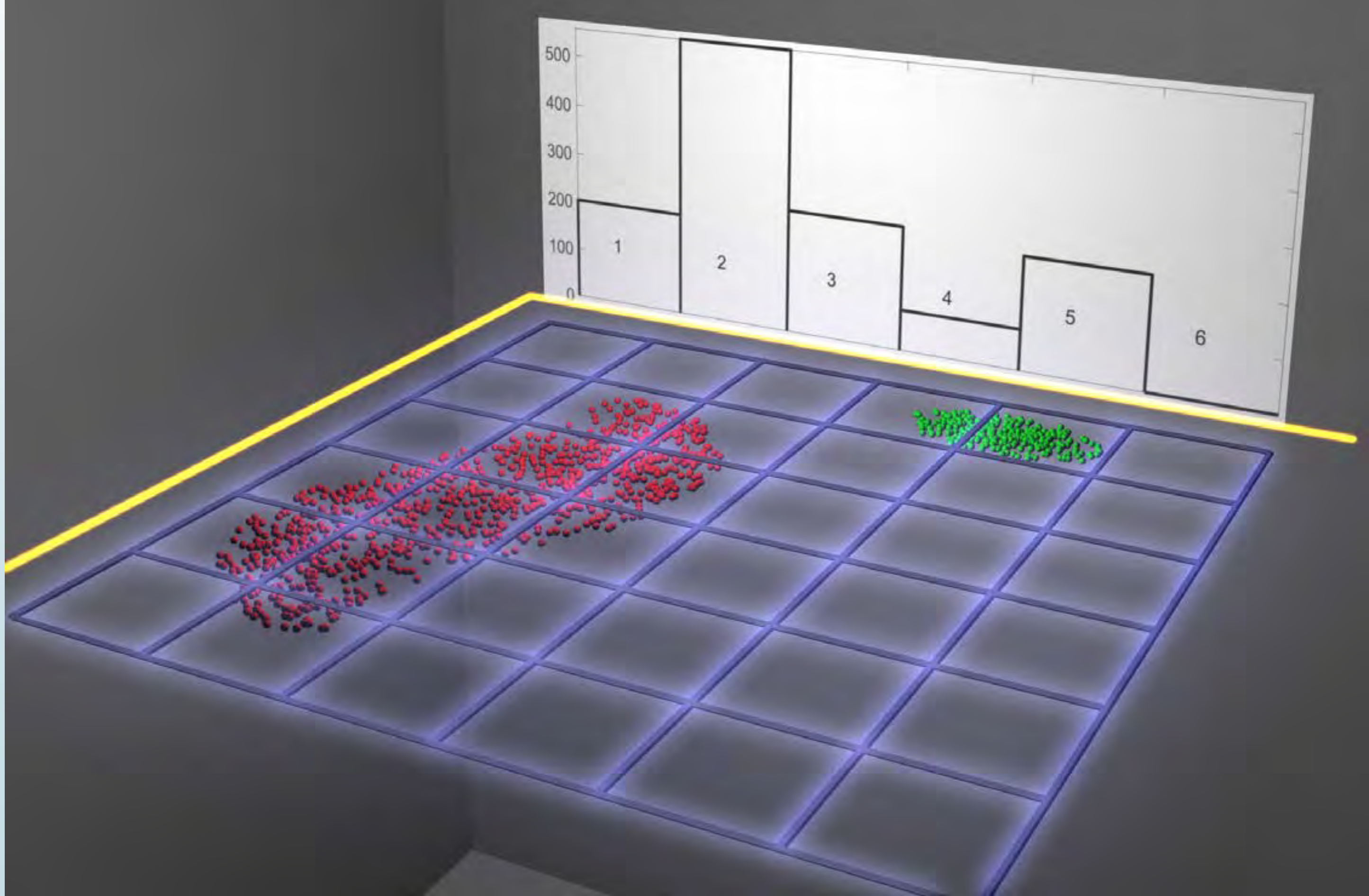
Add more data (different)



Building a space that contains data:

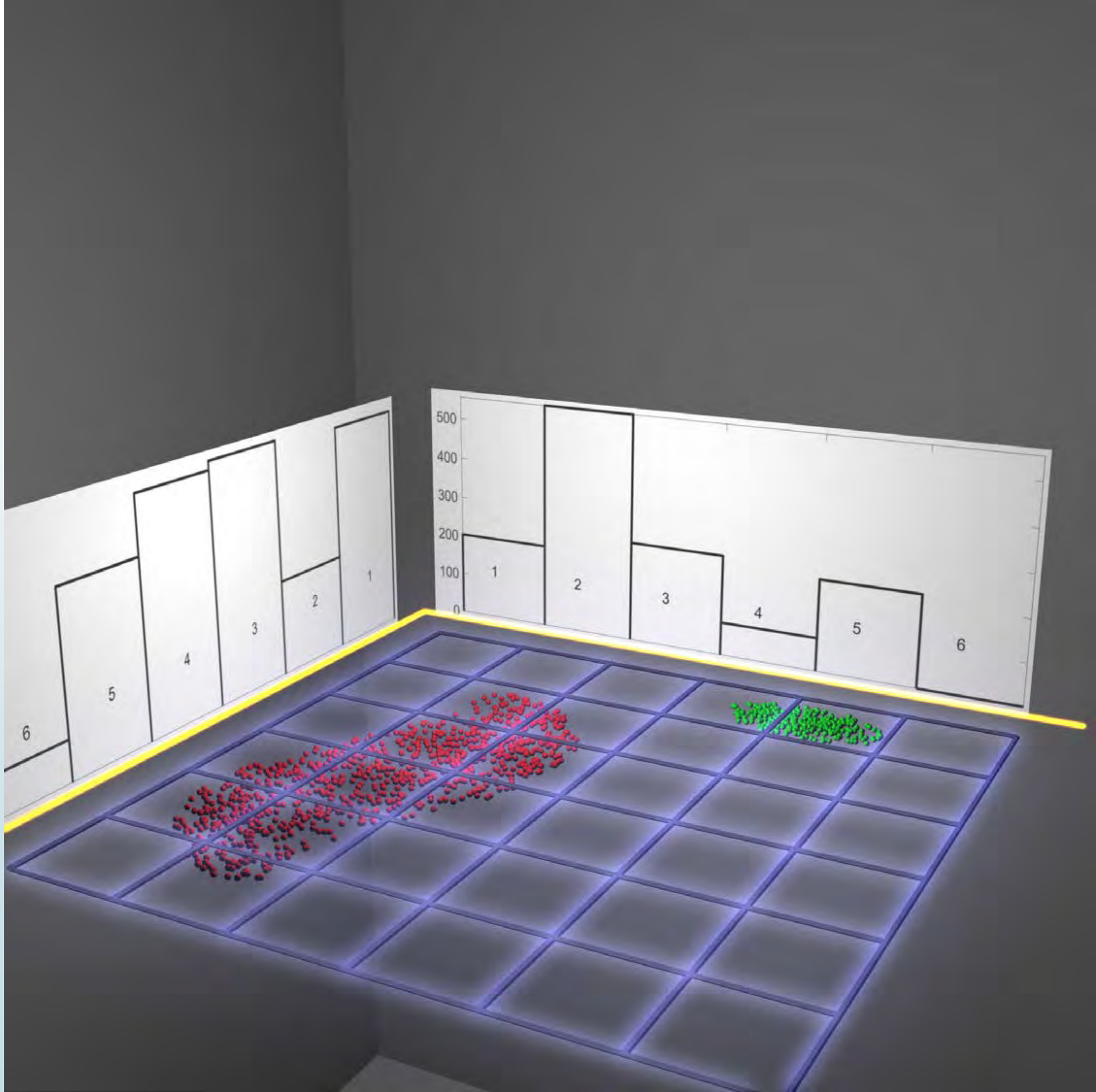
Histogram along one variable

Record bin address



Building a space that contains data:

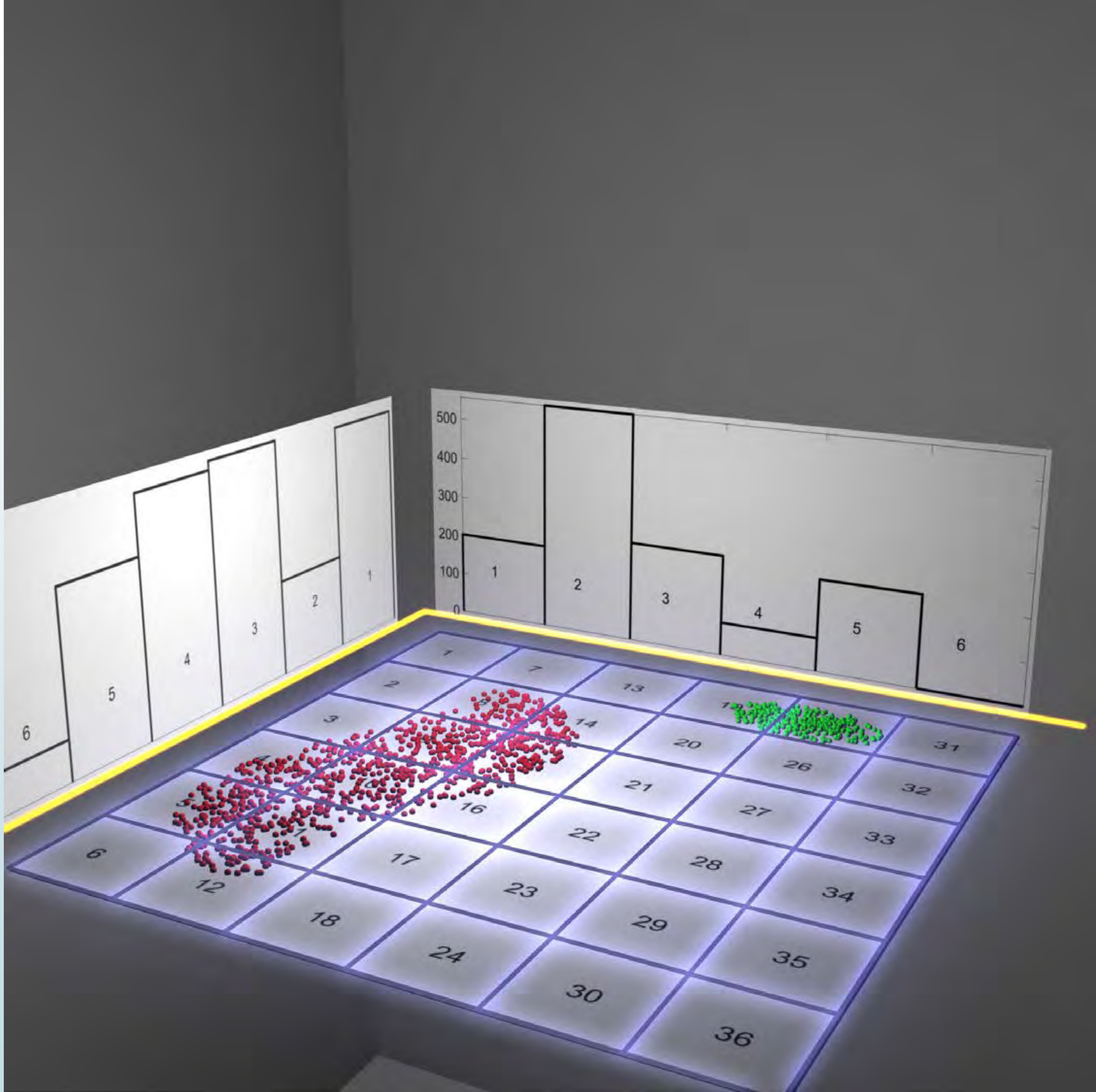
Repeat for each variable used



Building a space that contains data:

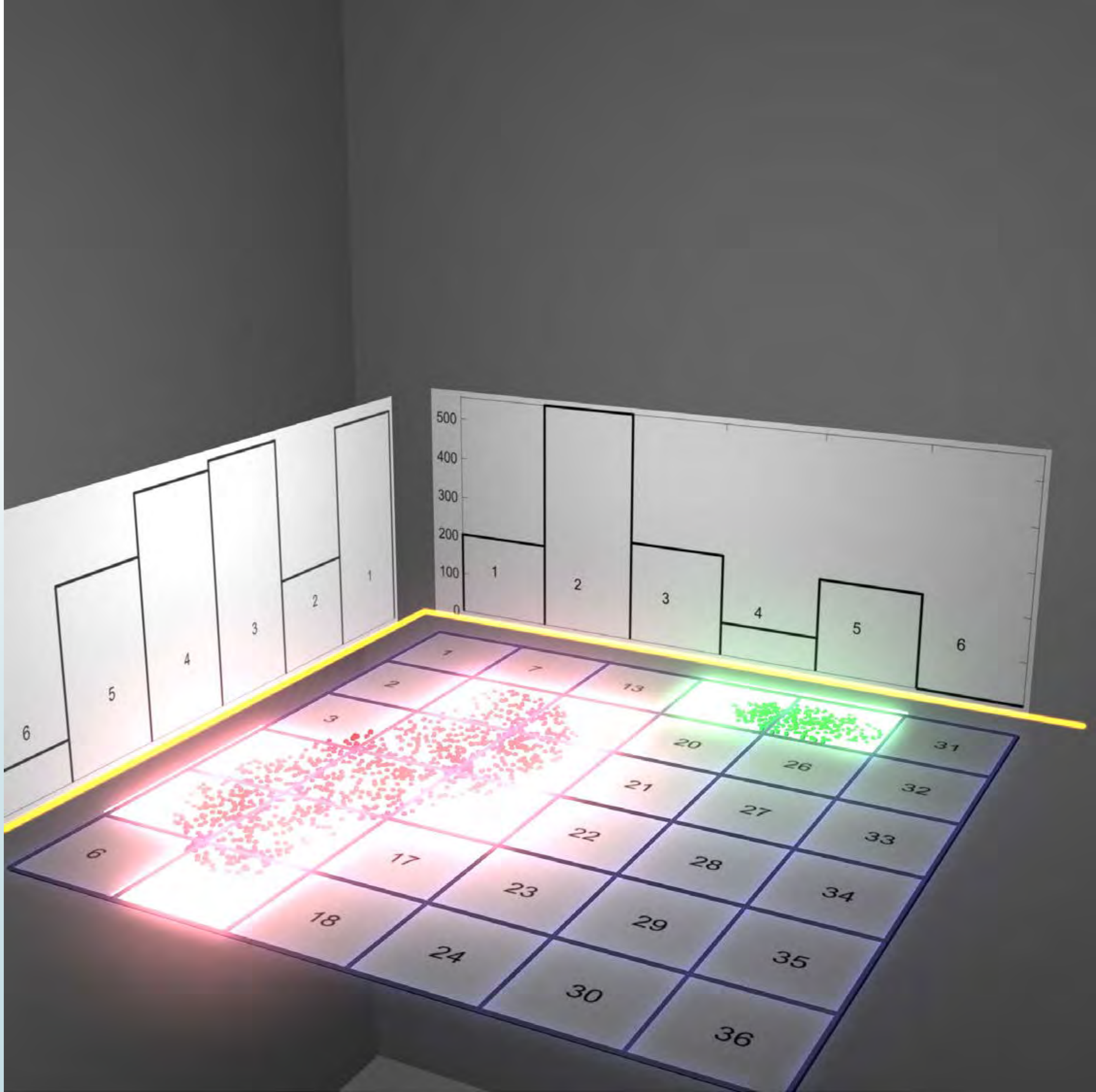
Form a single partition bin address from to set of individual bin addresses

Creates a single partition ID



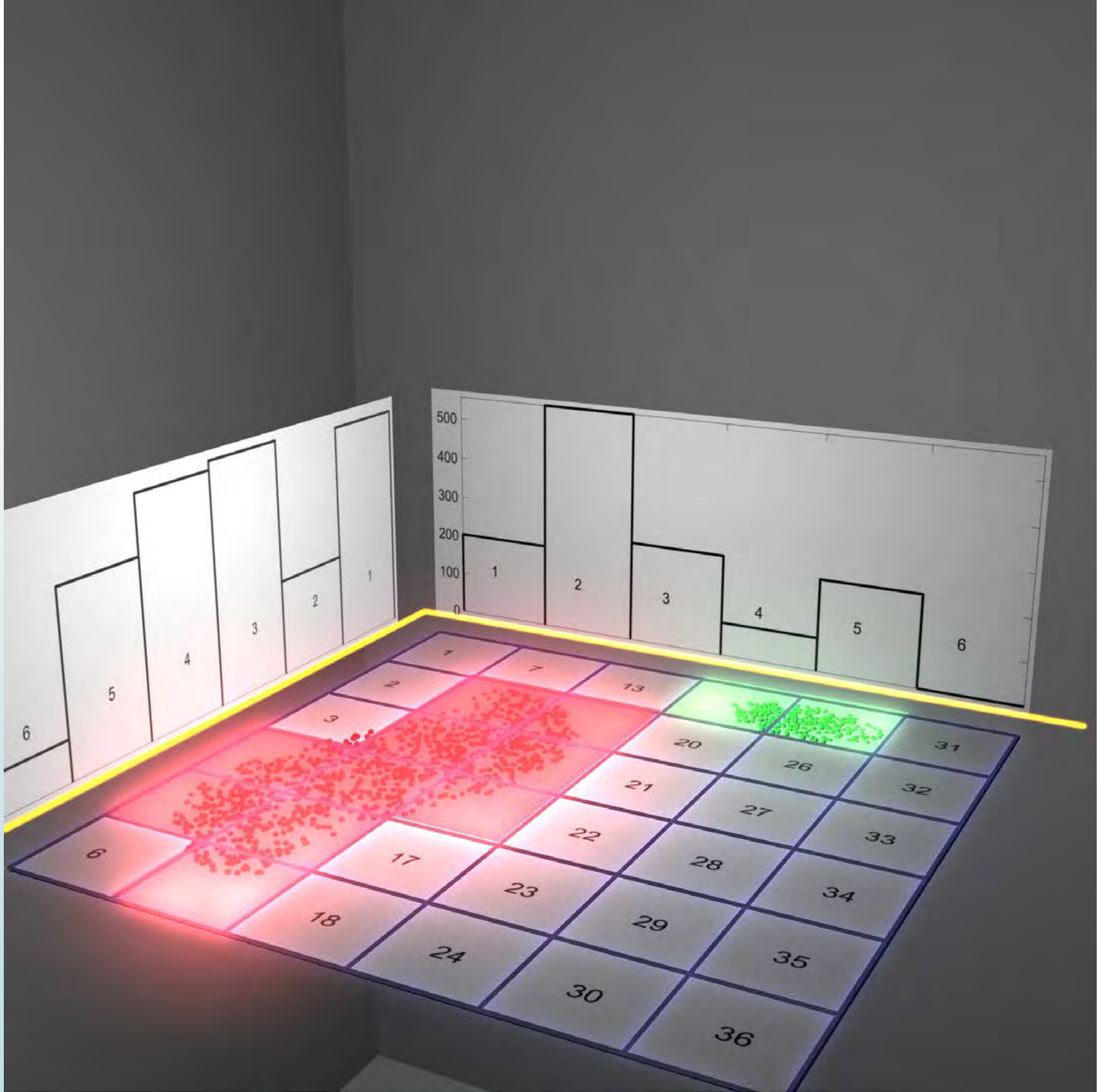
Building a space that contains data:

Only some partitions are populated



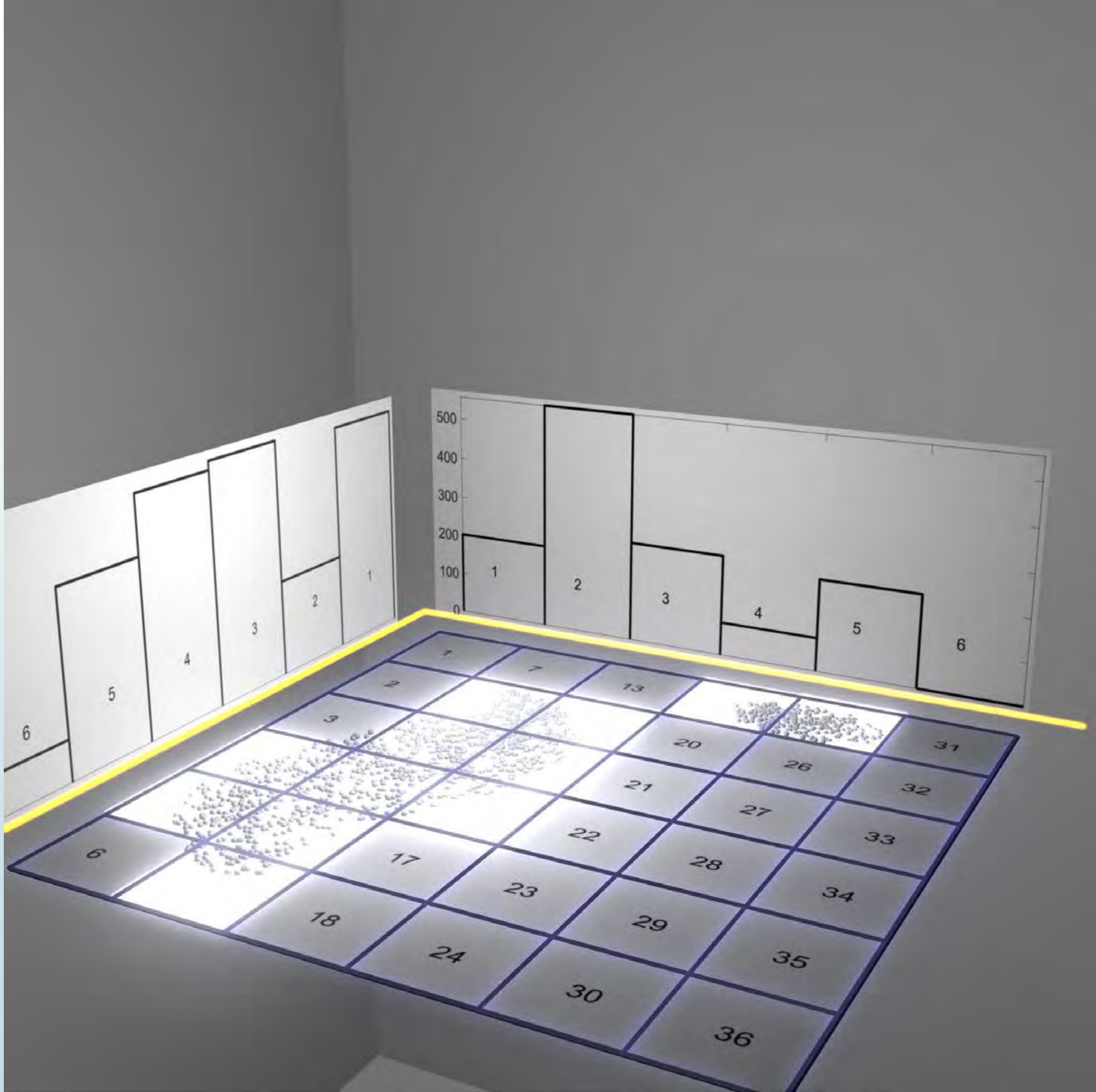
Building a space that contains data:

Only some partitions are populated



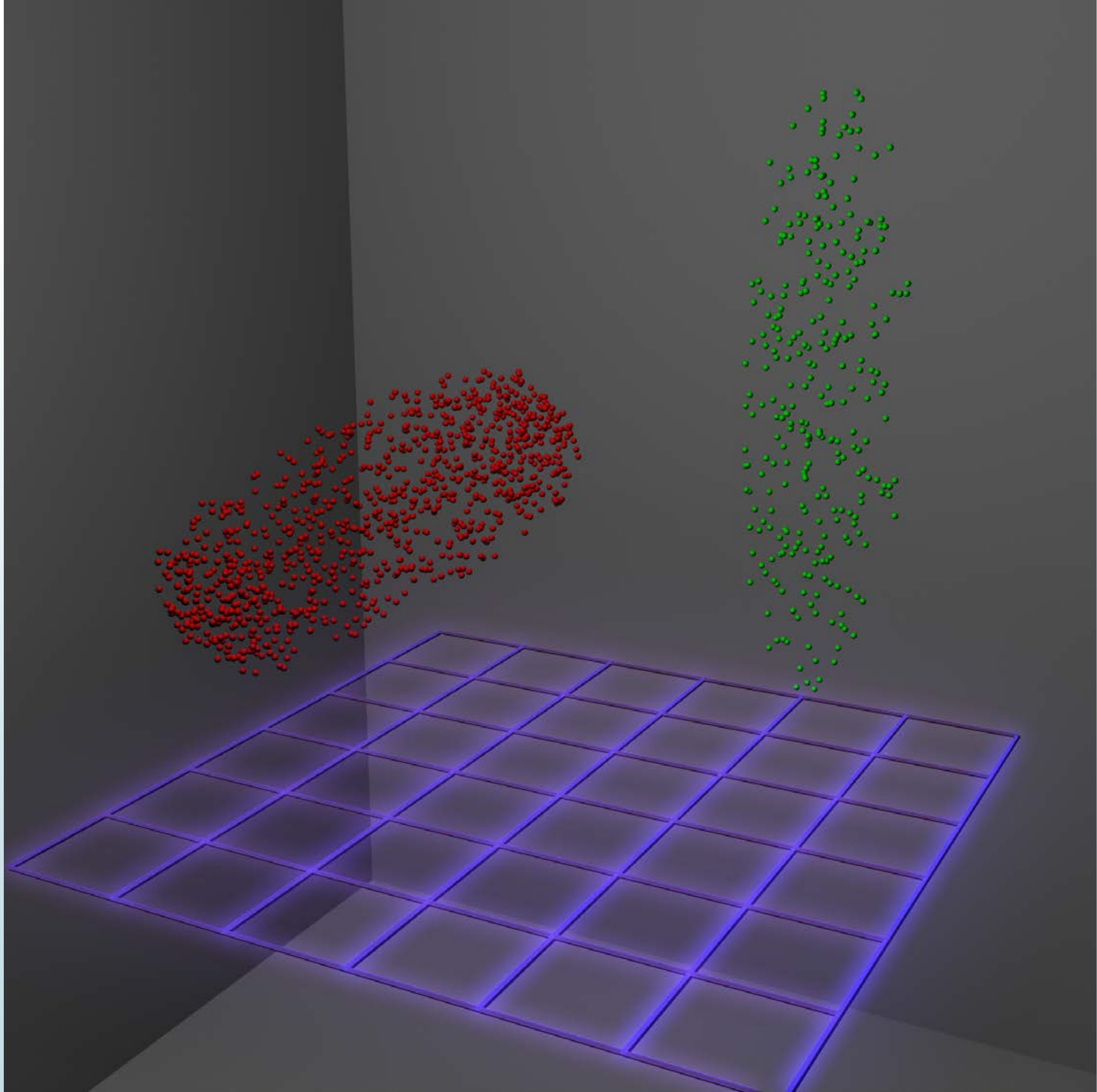
Building a space that contains data:

Before the analysis, all the data looks the same



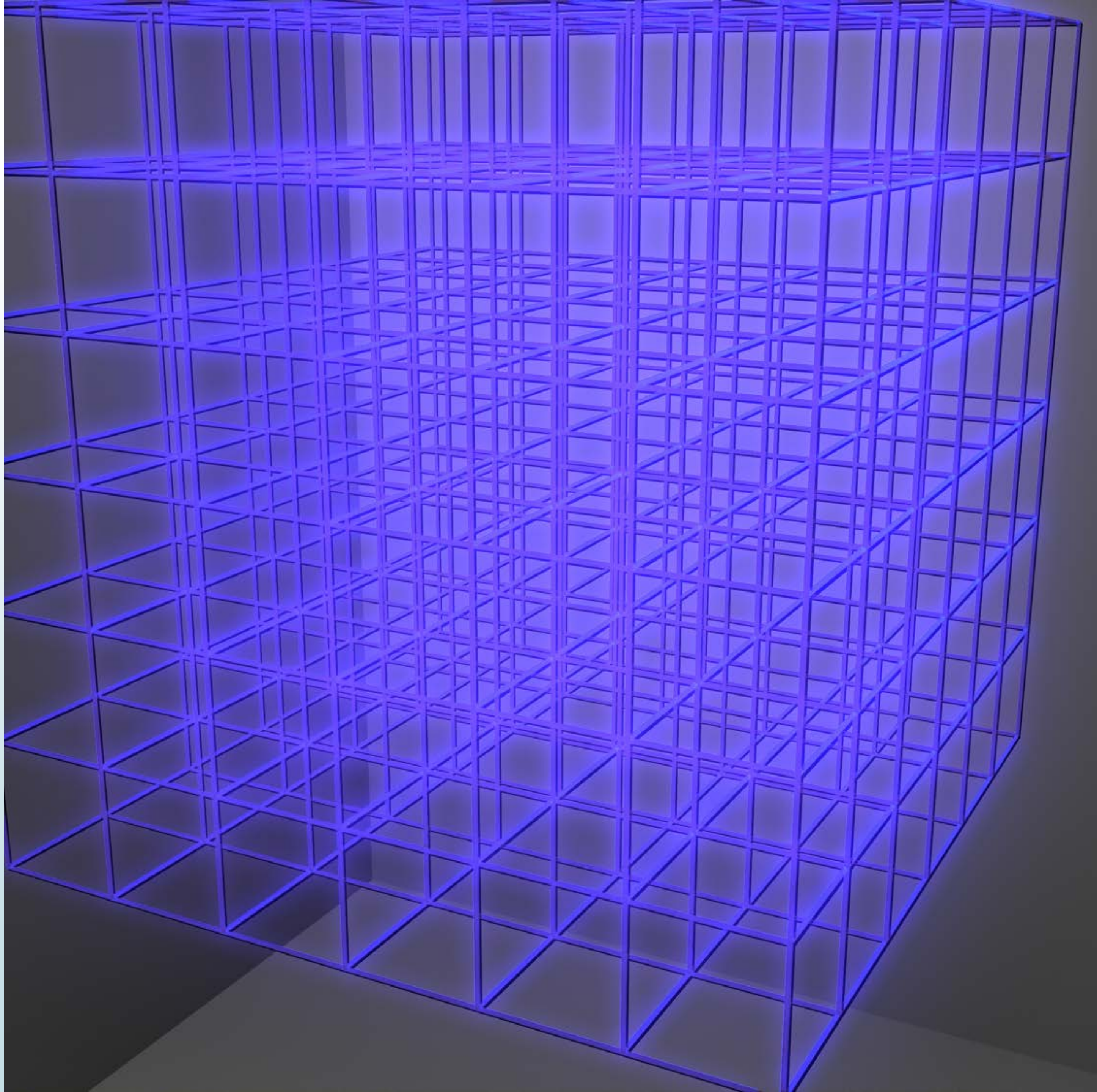
Building a space that contains data:

Going to 3-dim

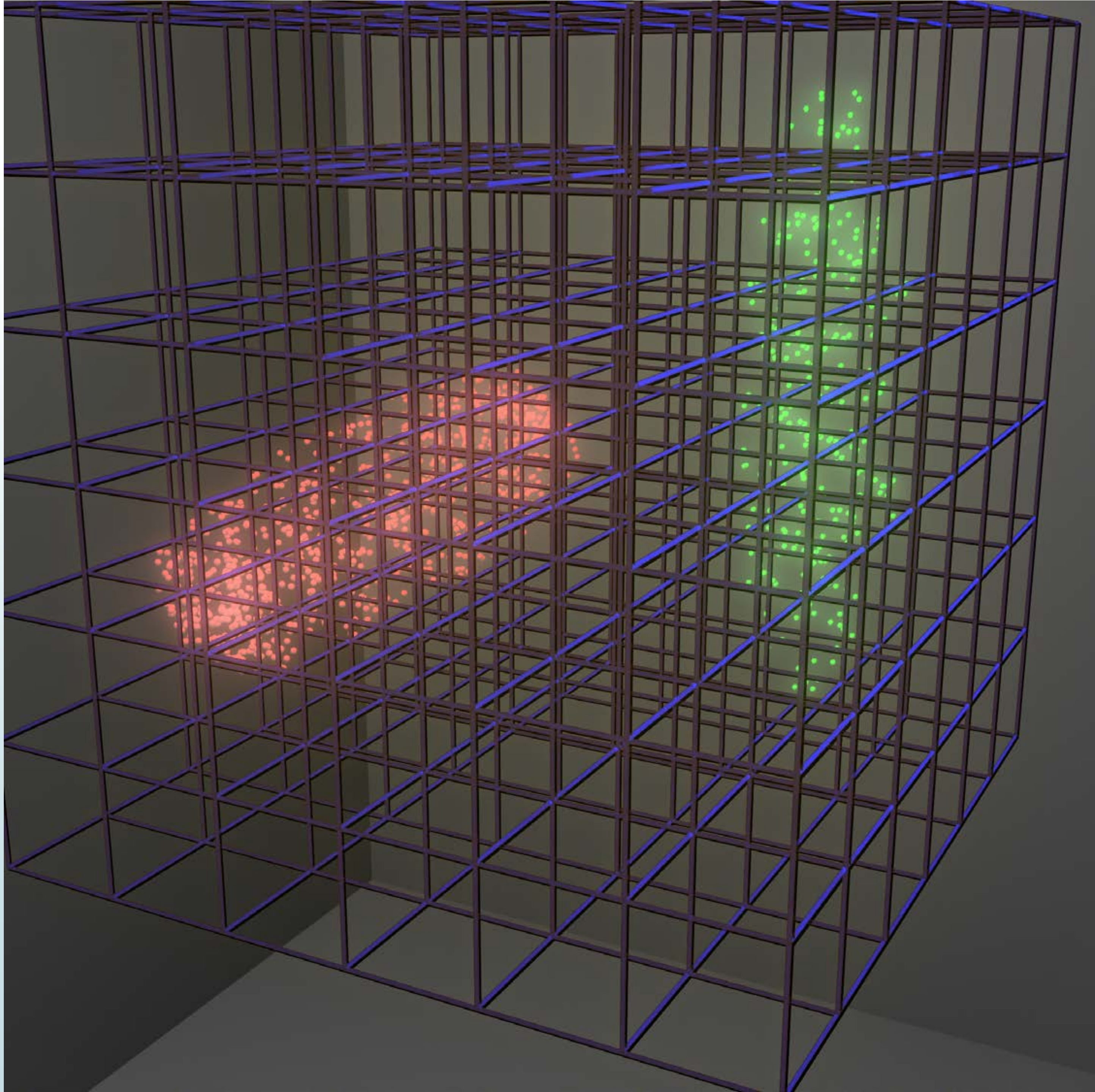


Building a space that contains data:

More partitions

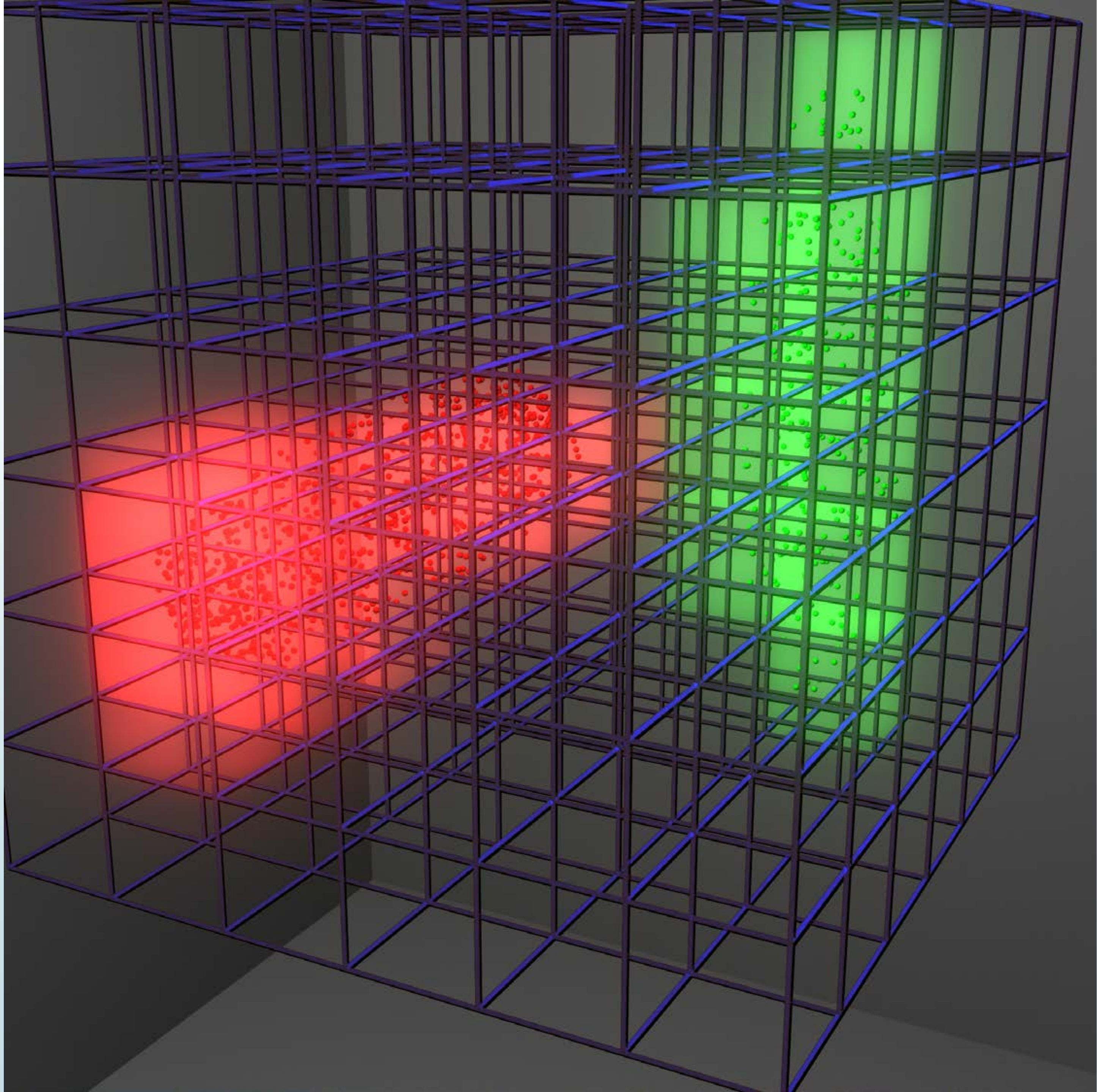


Building a space that contains data:



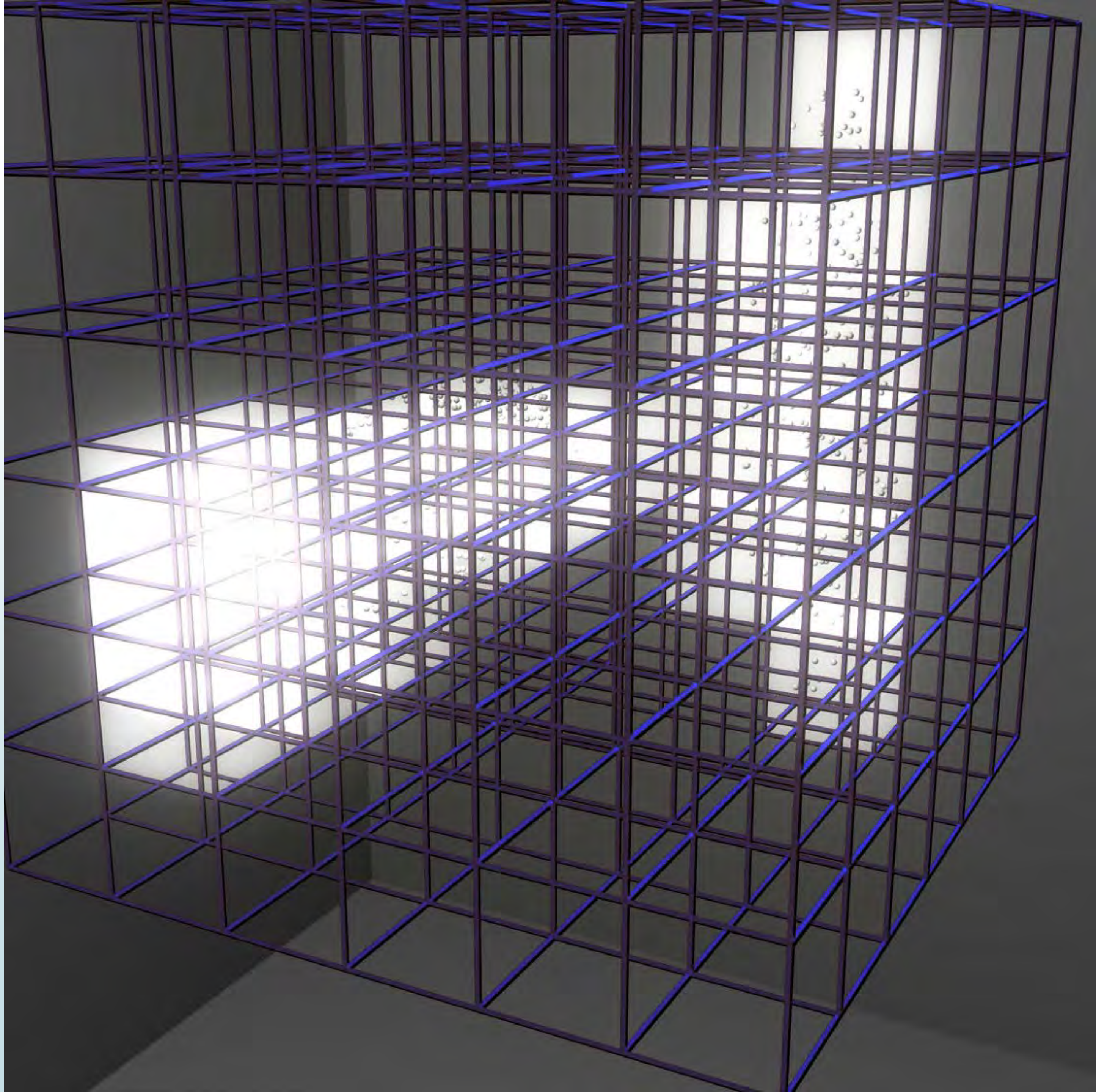
Building a space that contains data:

Find the populated partitions



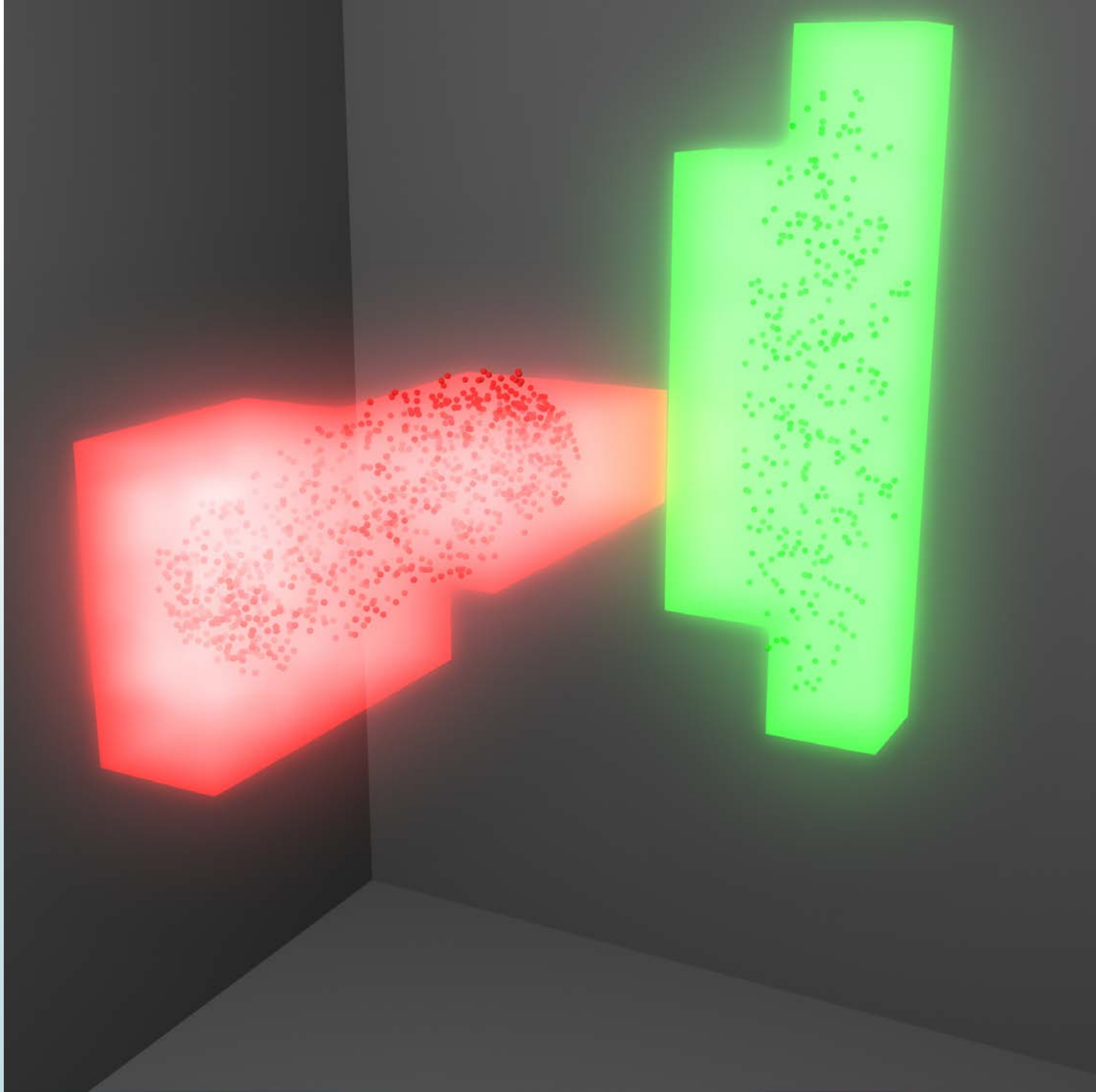
Building a space that contains data:

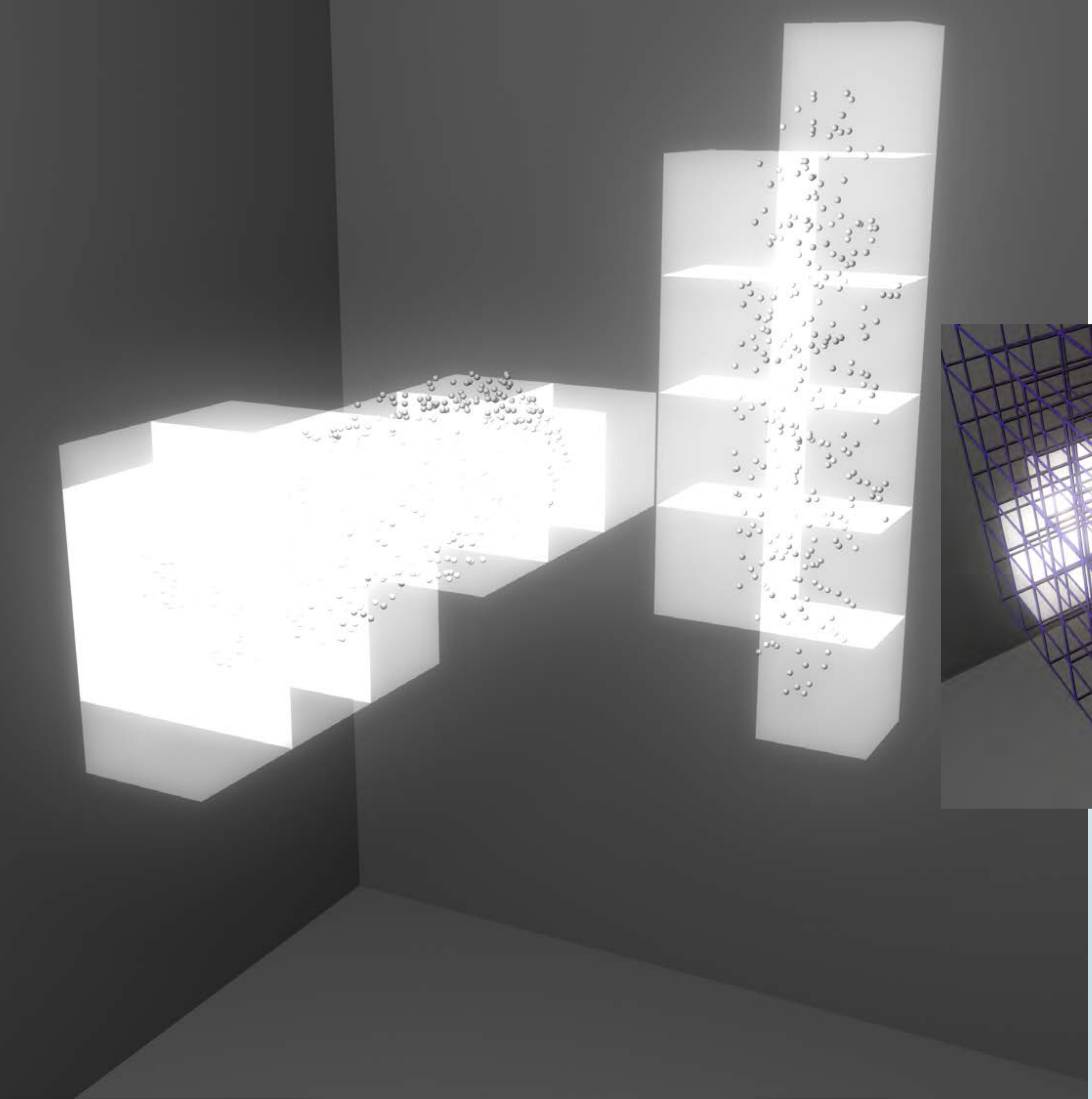
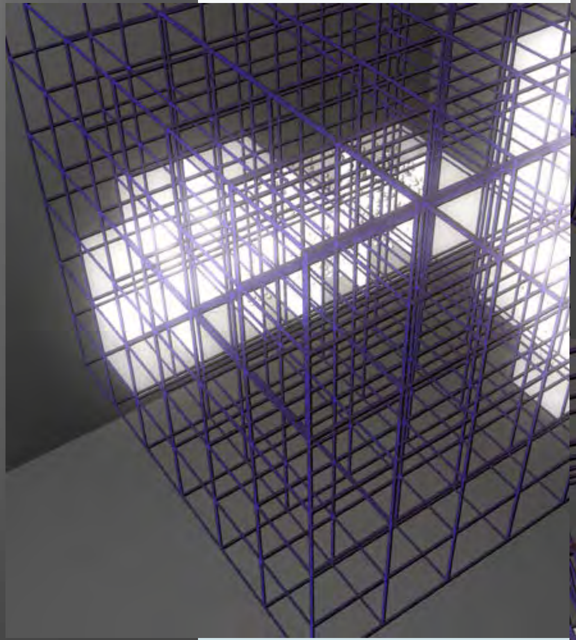
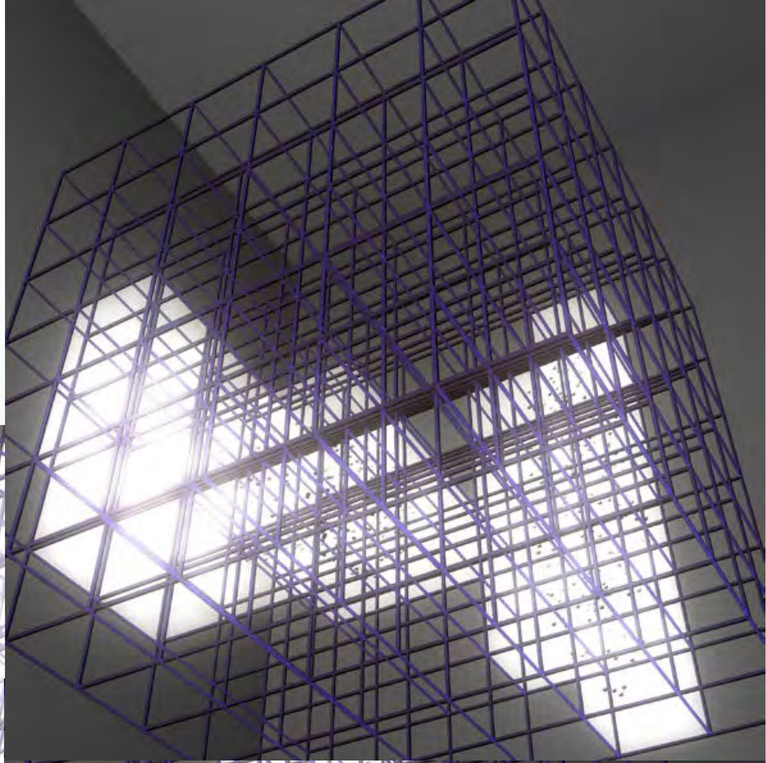
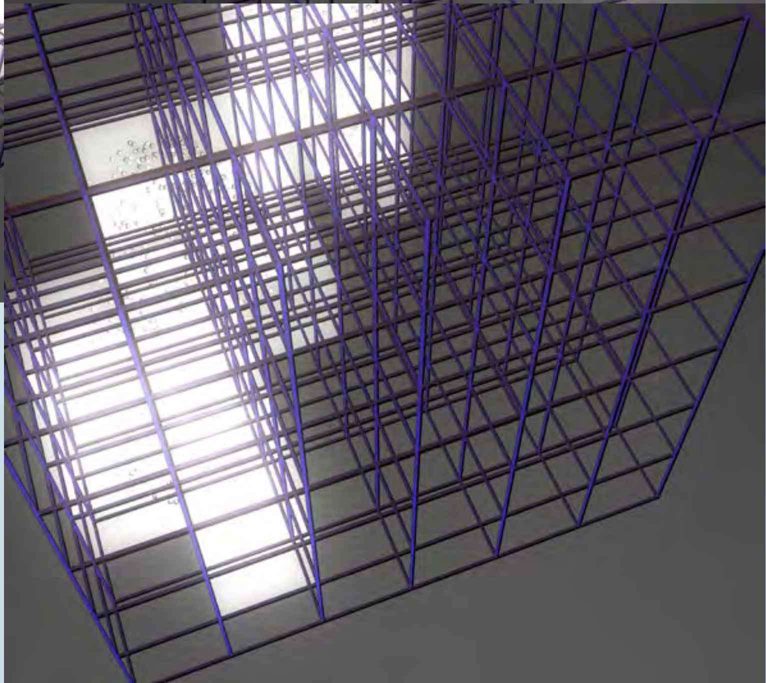
Data neutral initially



Building a space that contains data:

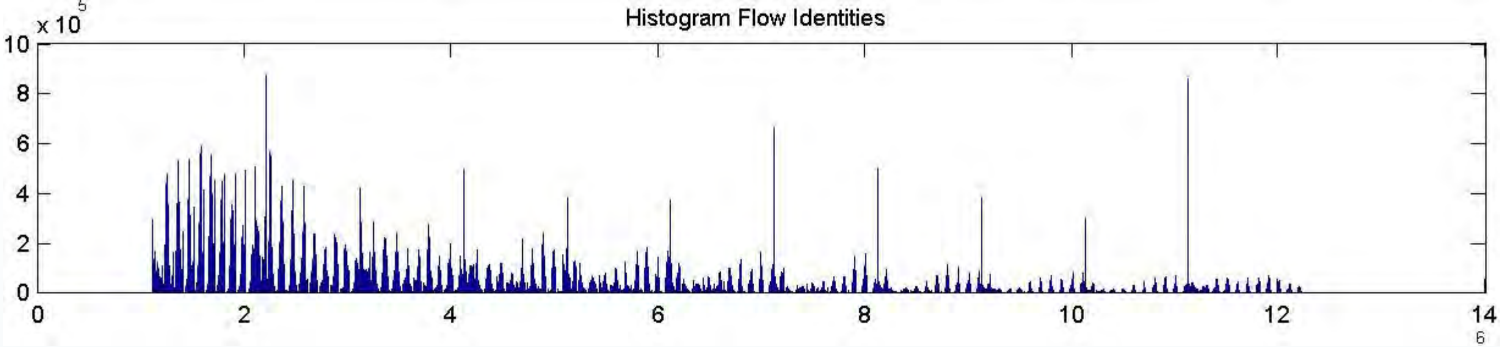
End goal: Separate the data types within the space (SPEMS)

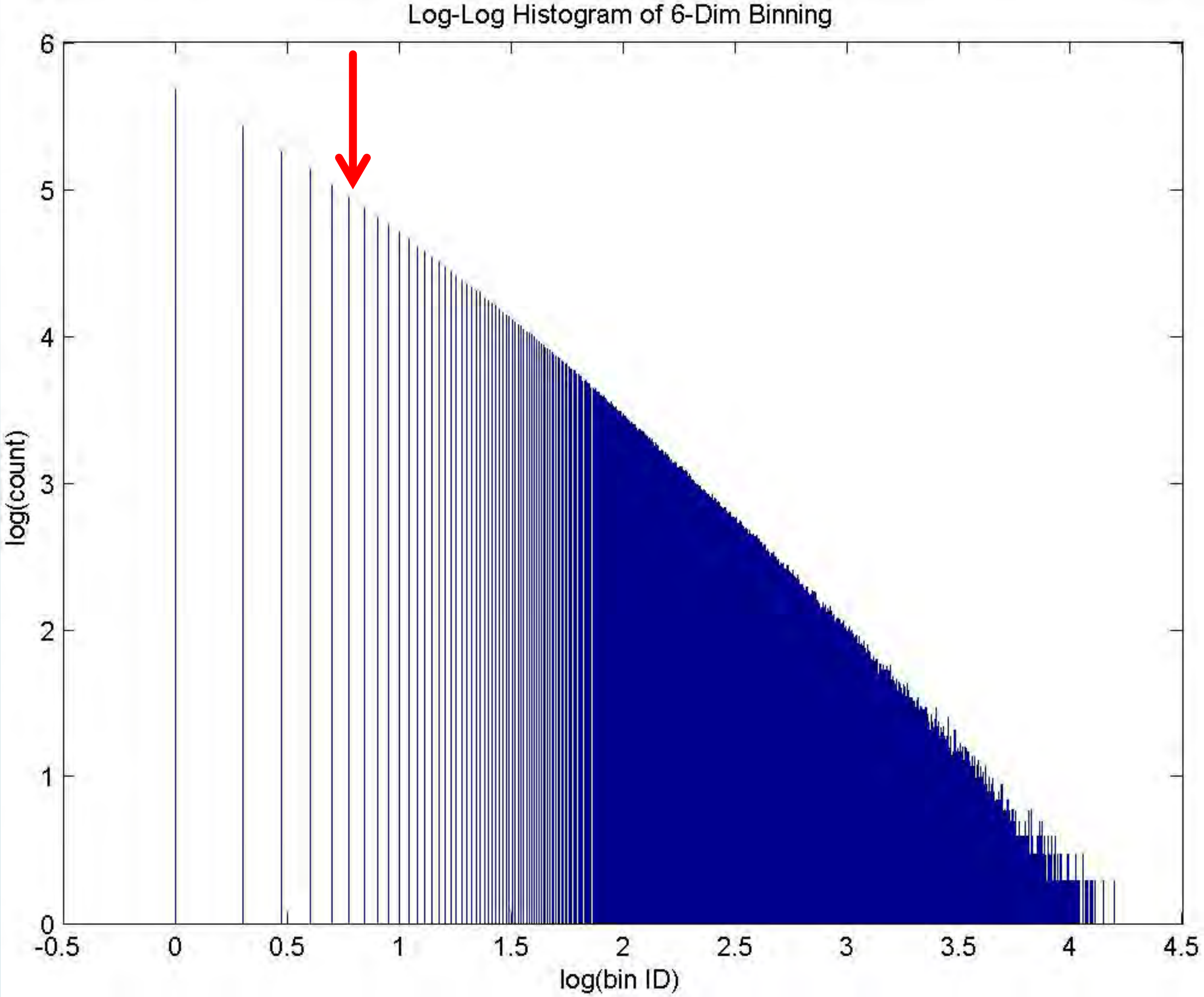


Reducing the Data from data points to partitions

- Starting from partition IDs
- Histogram partition IDs
- Set a threshold for the data to retain (arrow)
- Remove all partitions with lower populations (empties and noise)
- Map the remaining partition IDs to a serial index
- Assign the populations of the partitions to a weight value to each partition
- From here on, the “data” will be the partitions, N_P – Orders of Magnitude less



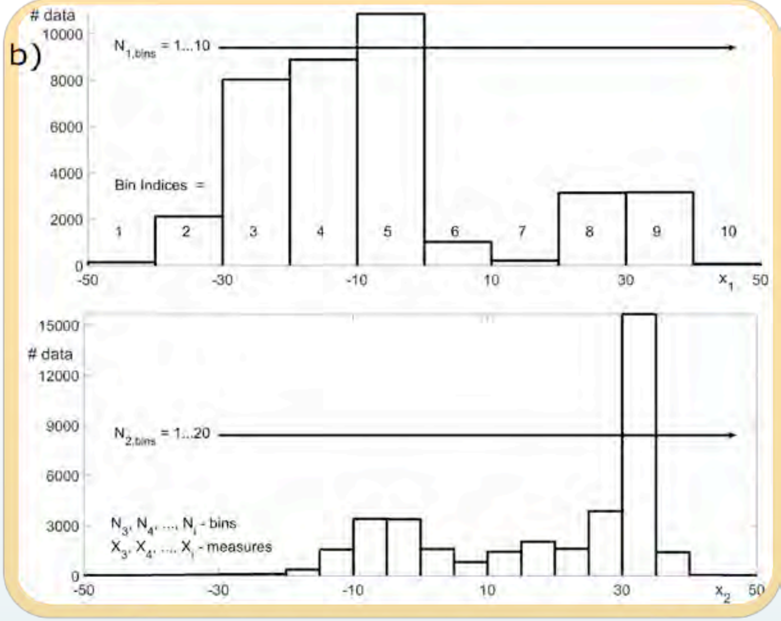


Process Outline

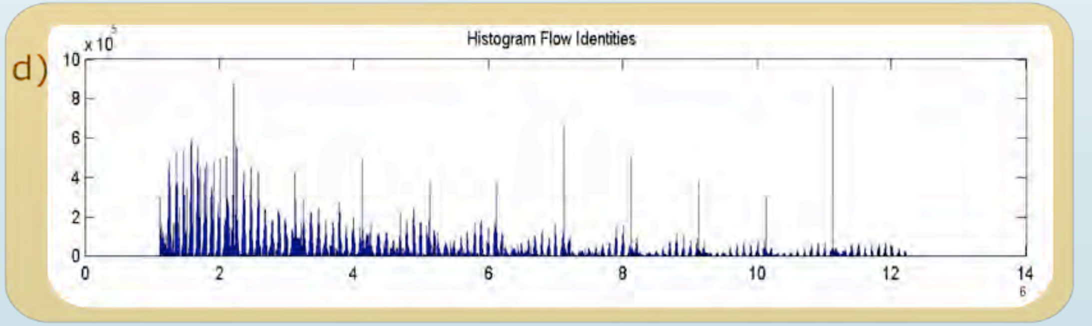
- Collect Data
- Choose variables
- Histogram each variable
- Set partition address (ID)
- Remove lower populations
- Remap partition IDs

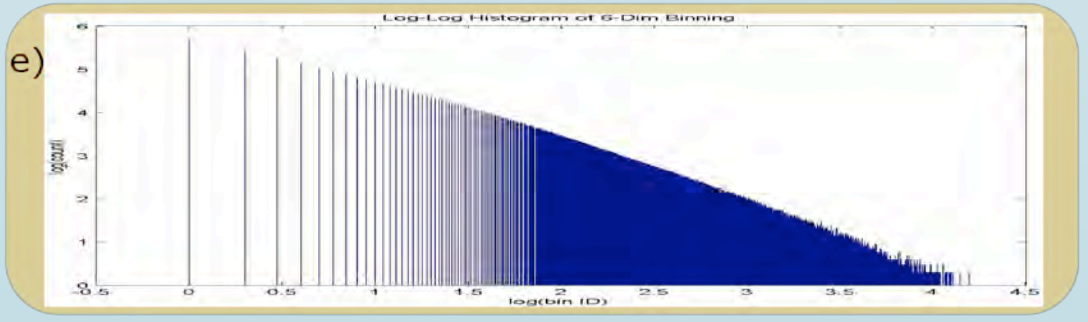
a)

j	X1	X2	X3	X4	X5	X6
1	9.5	-2.8	1.4	7.4	4.0	-0.8
2	-0.5	0.9	7.9	5.1	4.0	4.0
3	6.6	4.1	1.1	0.2	8.1	5.6
4	-1.0	-0.3	2.5	7.9	3.4	9.3
5	-0.3	3.0	-1.8	3.7	5.4	5.4
6	-1.8	6.1	-1.3	1.3	3.9	7.5
7	-2.9	1.2	-2.2	-1.6	0.3	0.3
8	7.8	1.2	0.9	4.8	4.8	2.7
9	5.8	2.5	-3.0	6.6	-0.9	7.7
10	3.1	5.9	2.2	-0.8	7.0	1.8
11	7.4	0.8	5.0	-1.6	4.3	0.3
12	-2.3	-0.5	0.0	4.7	9.5	0.3
13	9.2	3.8	9.7	7.0	9.3	-0.3
14	3.6	4.2	-2.5	9.3	7.1	4.6
15	4.0	6.5	9.3	4.7	2.0	5.8
16	2.8	8.9	3.6	1.7	4.3	2.9
17	-1.1	6.6	5.6	8.6	-2.4	5.1
18	8.1	-0.5	-2.0	2.6	2.9	7.0
19	-2.0	1.3	-1.0	9.5	7.9	8.2
20	0.0	6.2	0.9	8.7	9.7	6.5
21	7.9	-2.8	9.1	-1.7	6.3	7.2
22	9.2	2.3	7.6	0.1	5.7	5.2
23	1.3	3.9	-0.5	1.2	6.6	4.8
24	-1.5	-1.4	-2.9	7.8	7.8	4.3
25	8.8	0.6	-2.0	2.8	5.2	-2.6
...	...	...	...	...	...	...



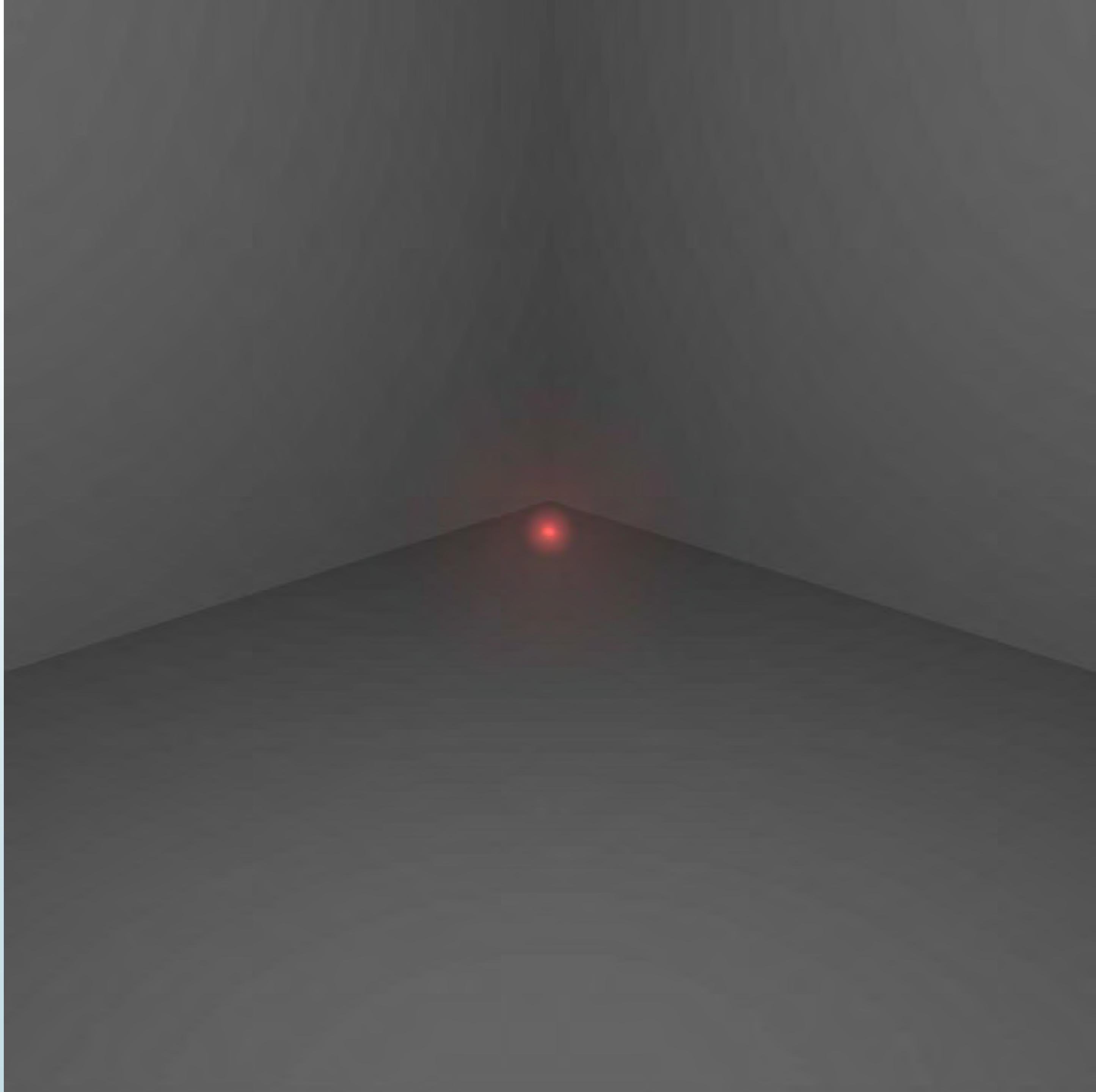
c) $$k_j = \sum_{i=1}^{N_D} x_{i,j} * \Pi_{q=1}^{i-1} N_{B,q}$$





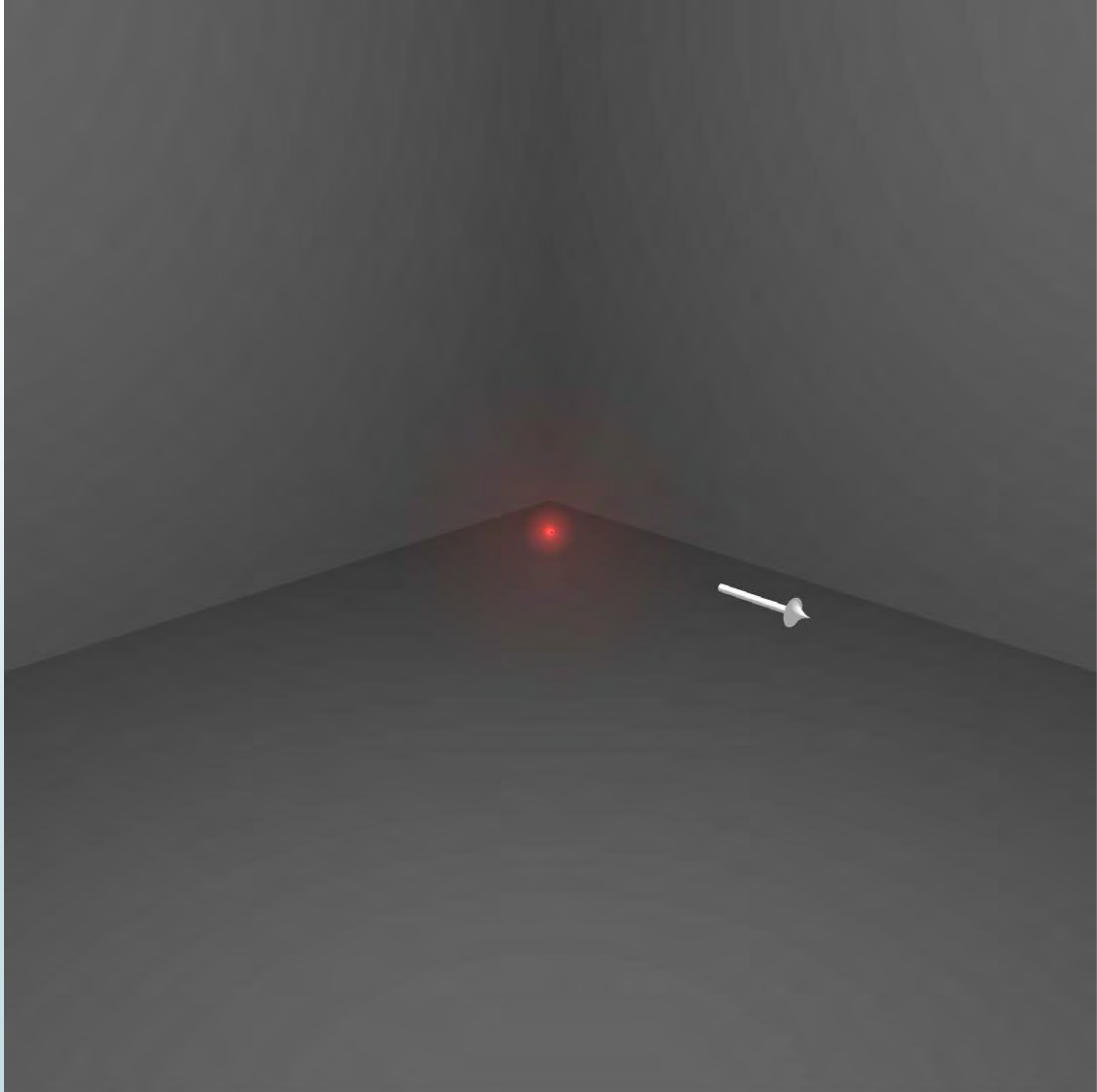
Building an N-dim first nearest neighborhood:

Start in 0-dims (point)



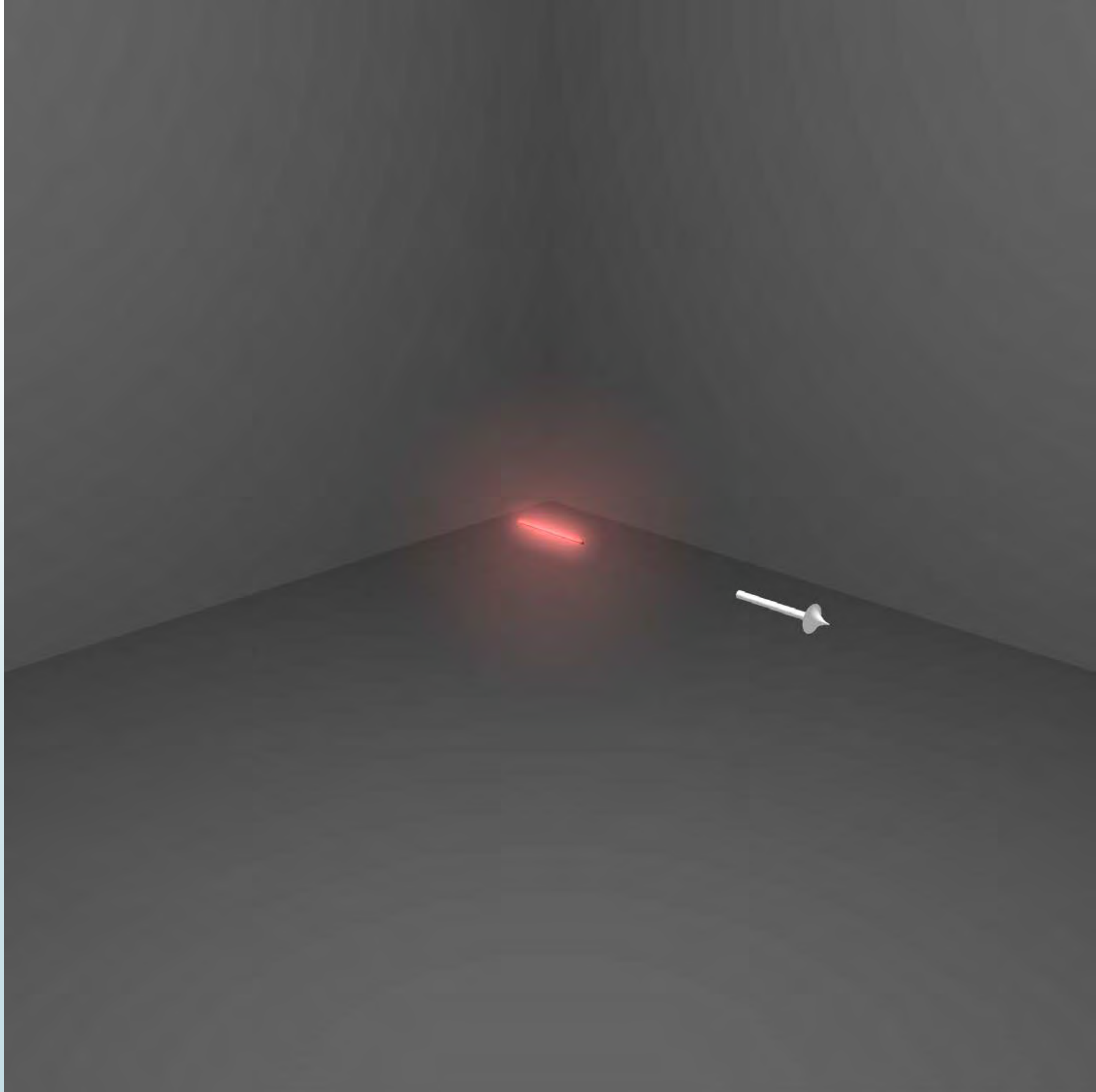
Building an N-dim first nearest neighborhood:

Start with one dimension



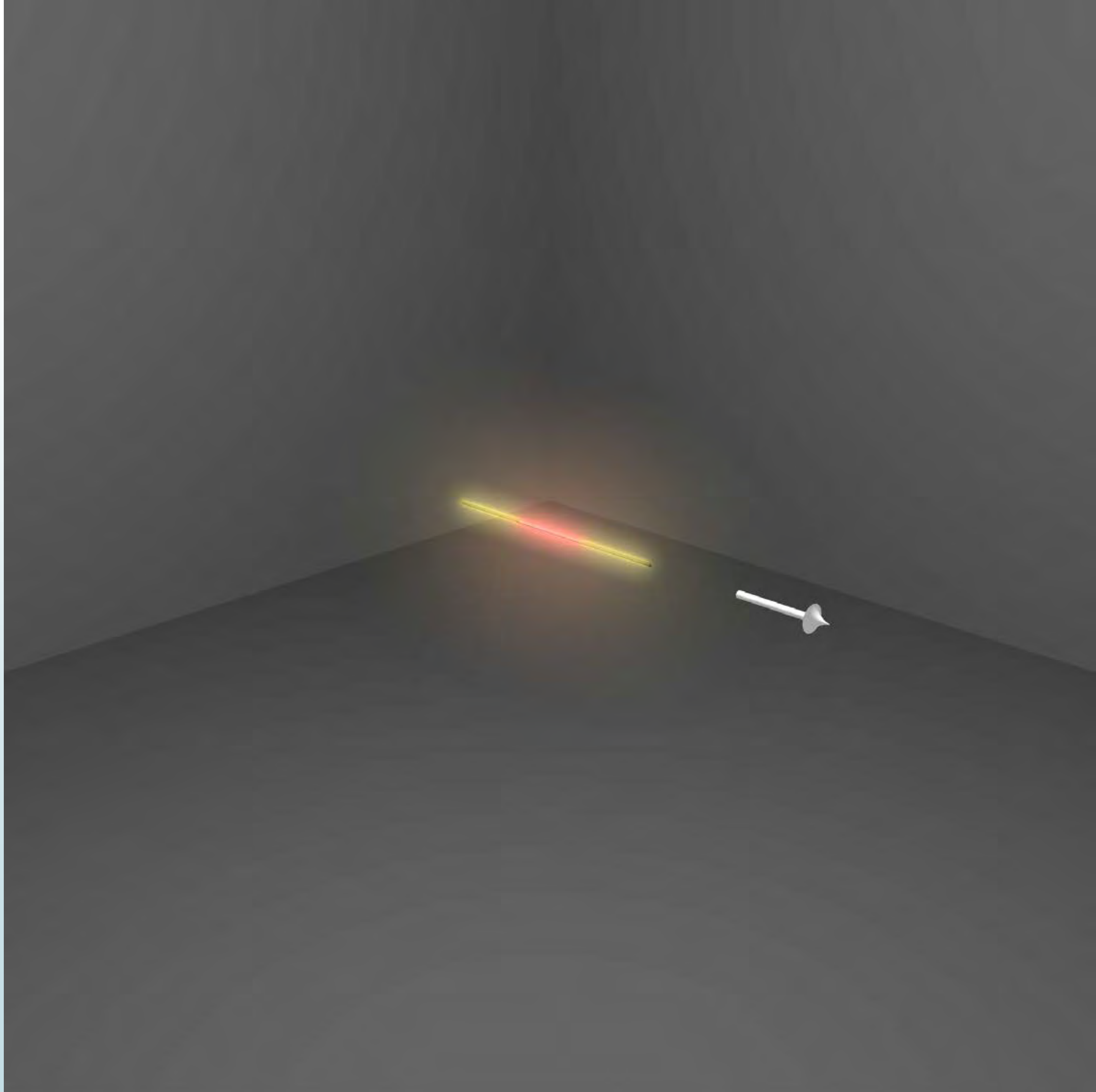
Building an N-dim first nearest neighborhood:

Extrude our point along the new direction by one unit length (line)



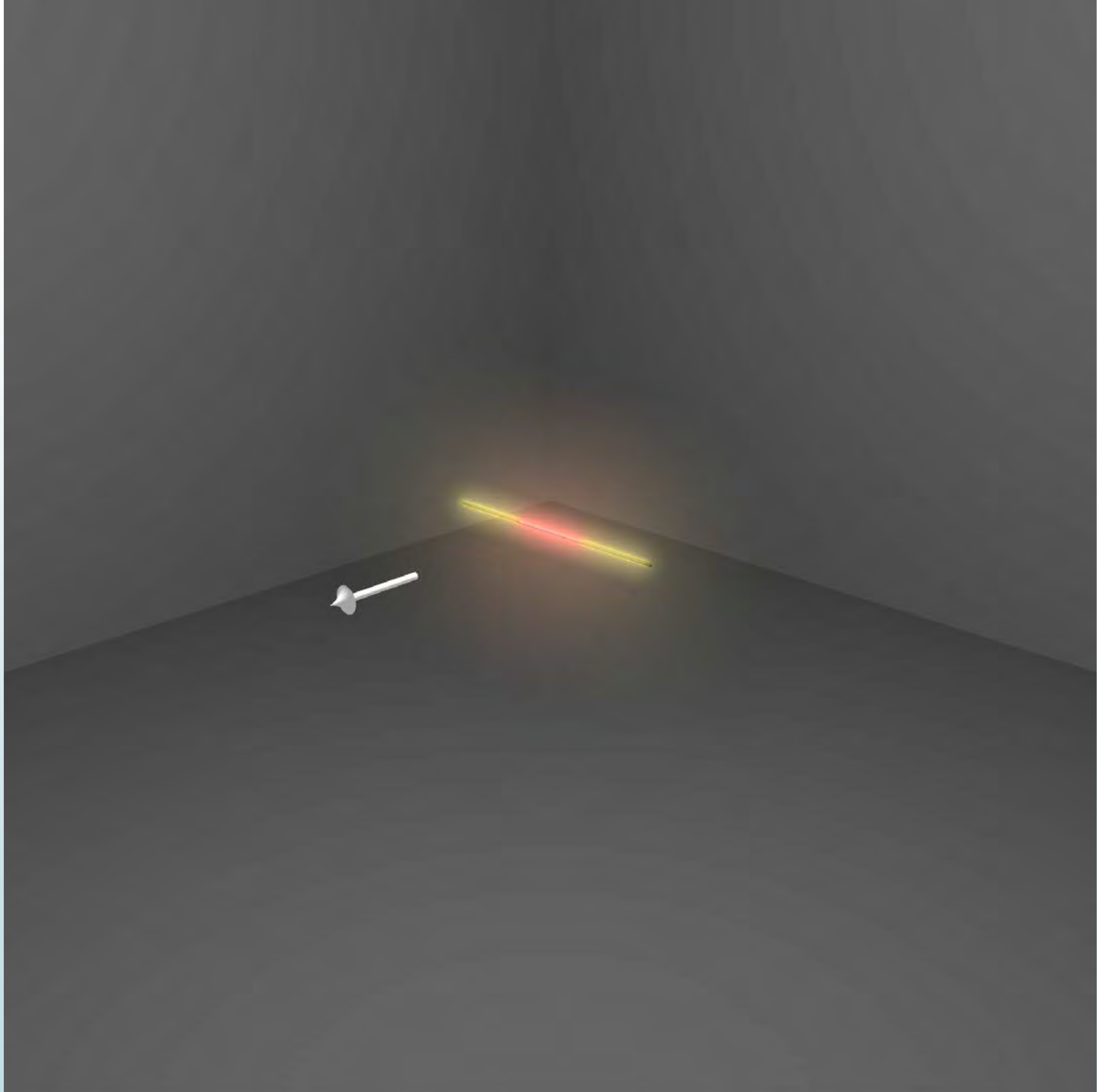
Building an N-dim first nearest neighborhood:

Create neighbors by copying the center one forward and one backward along new direction



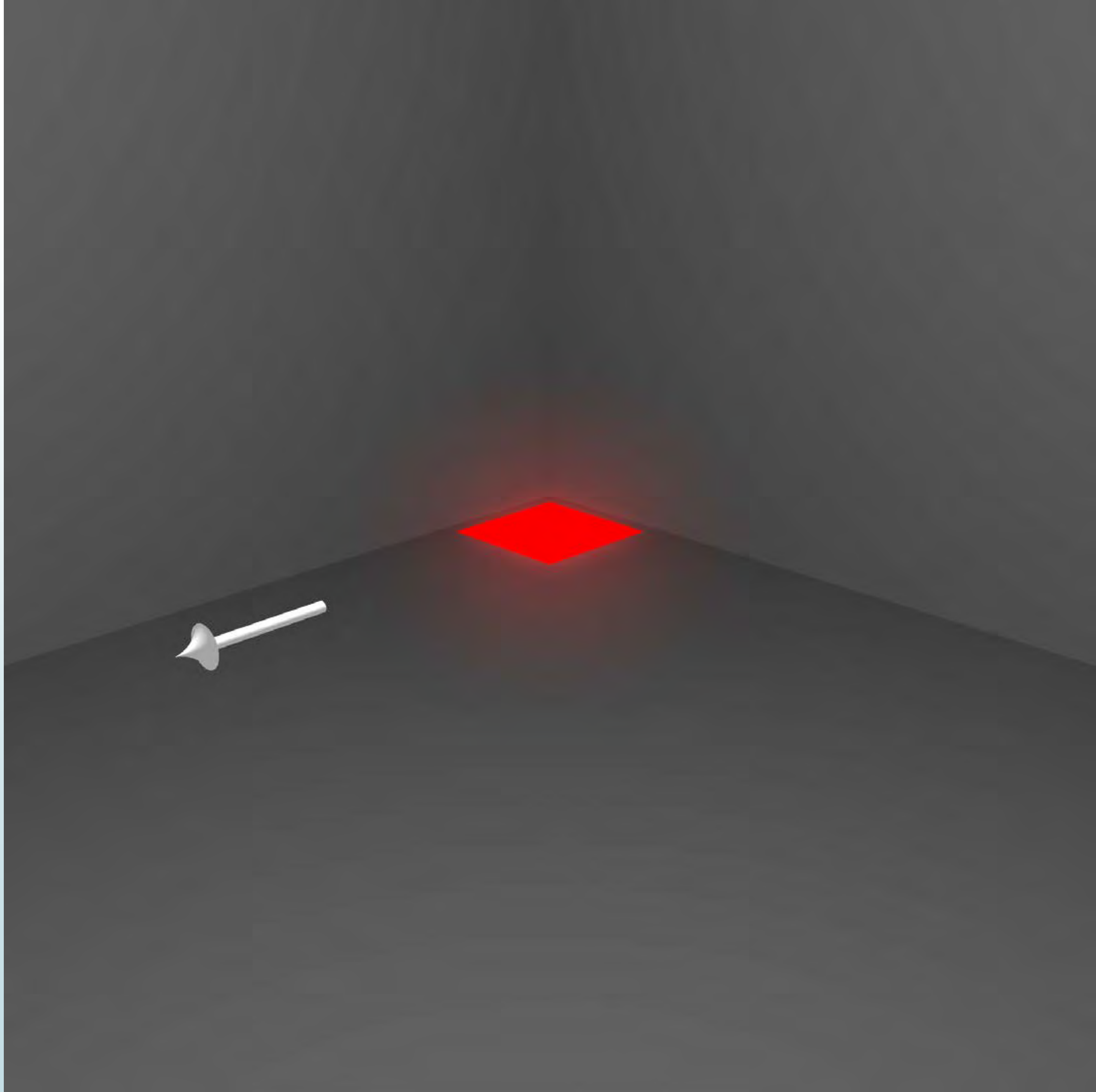
Building an N-dim first nearest neighborhood:

Choose a new direction (2D)



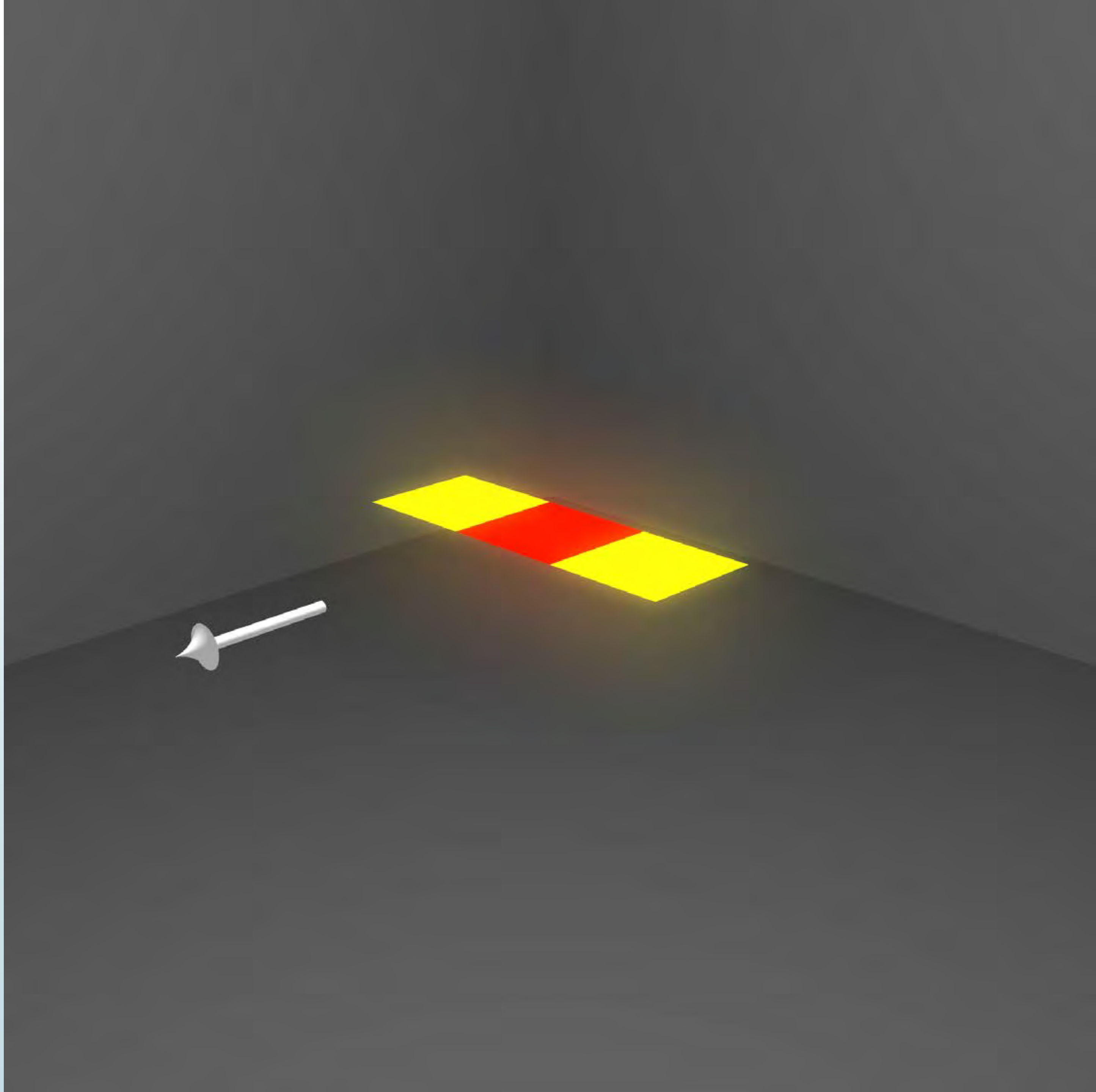
Building an N-dim first nearest neighborhood:

Extrude the center line into a unit square



Building an N-dim first nearest neighborhood:

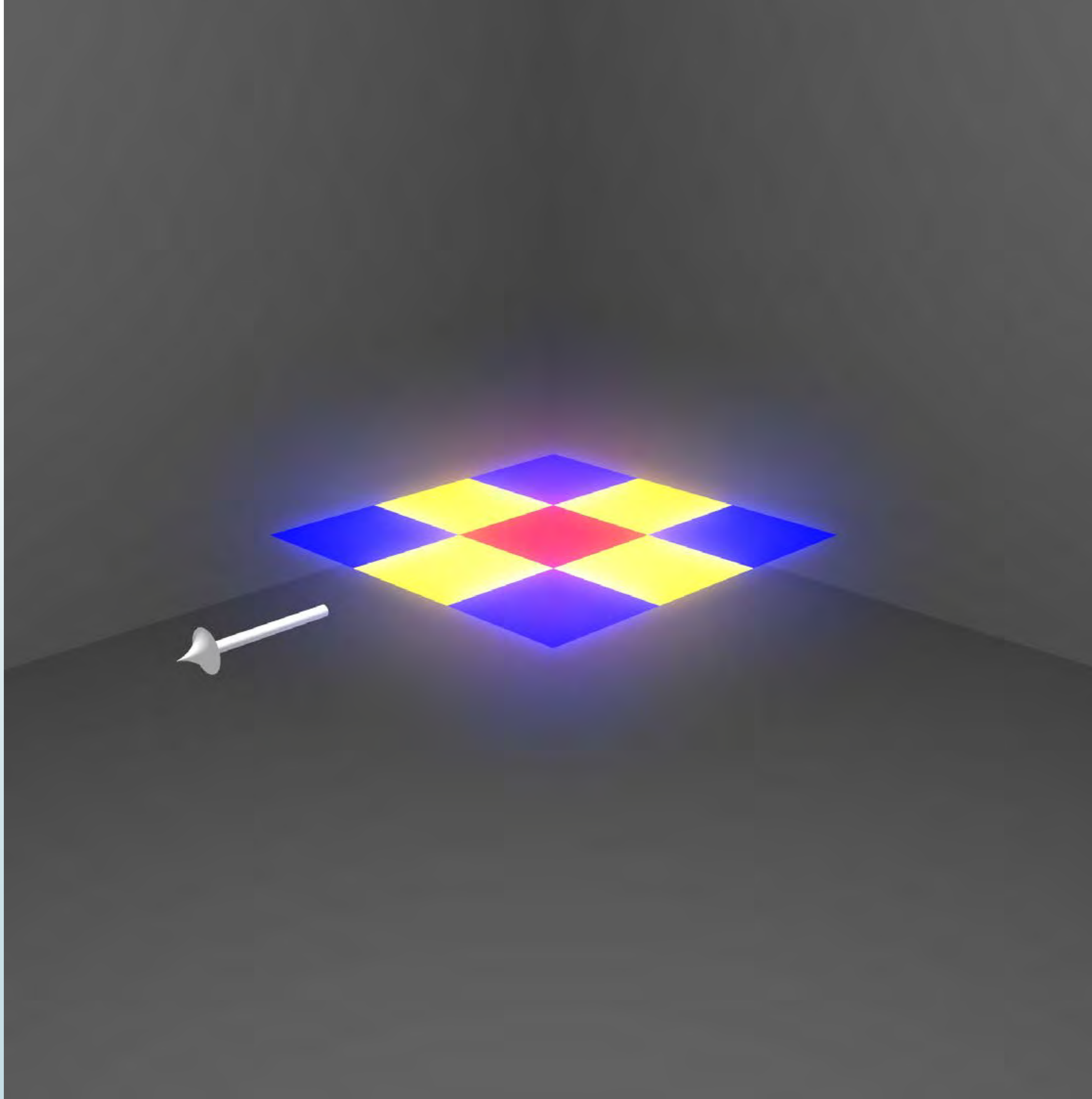
Extrude the neighboring lines as well into 2D



Building an N-dim first nearest neighborhood:

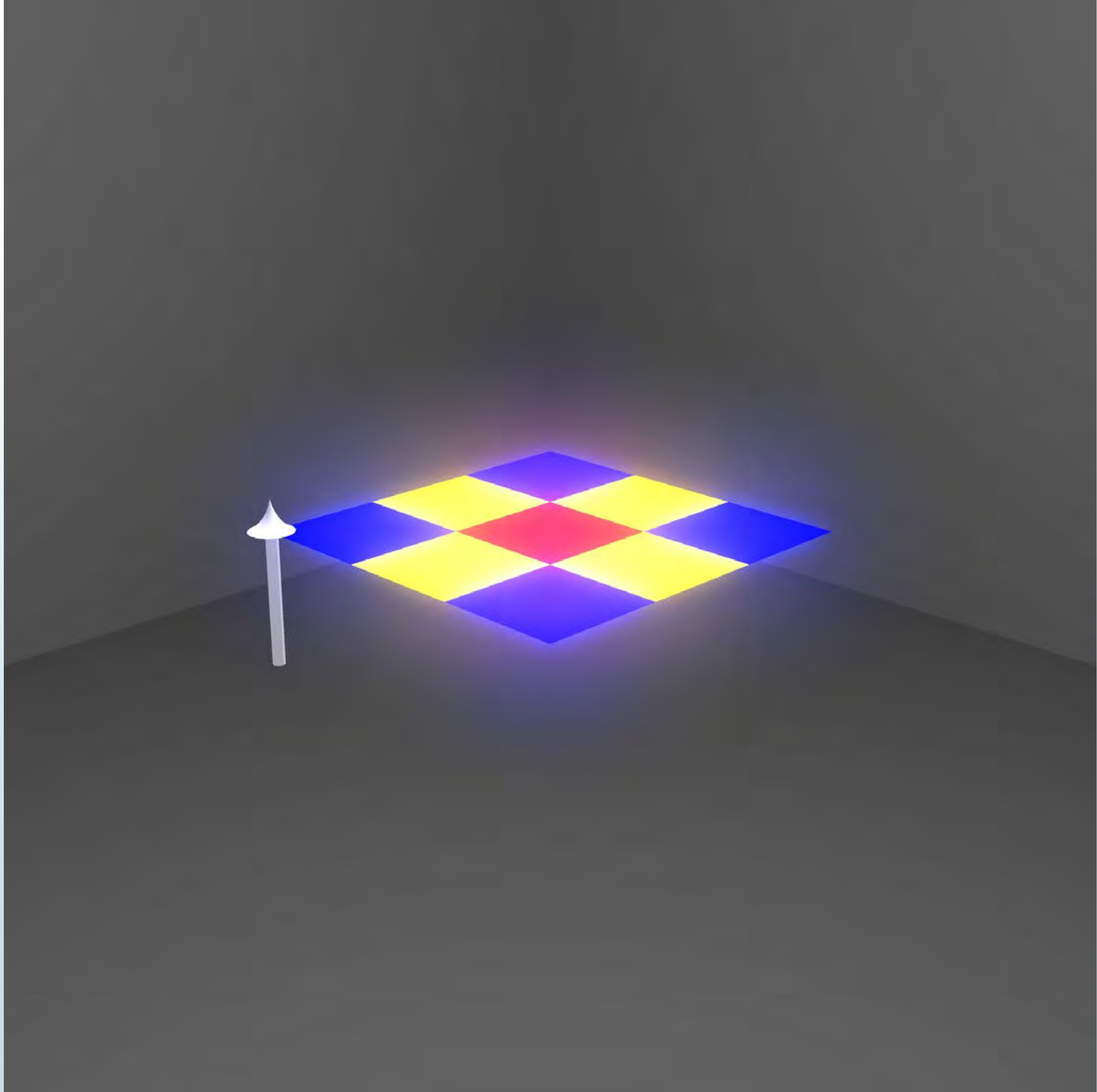
Copy the three unit squares one set forward and one backward along new direction

Two different types of neighbors (share a line-yellow and a point-blue)



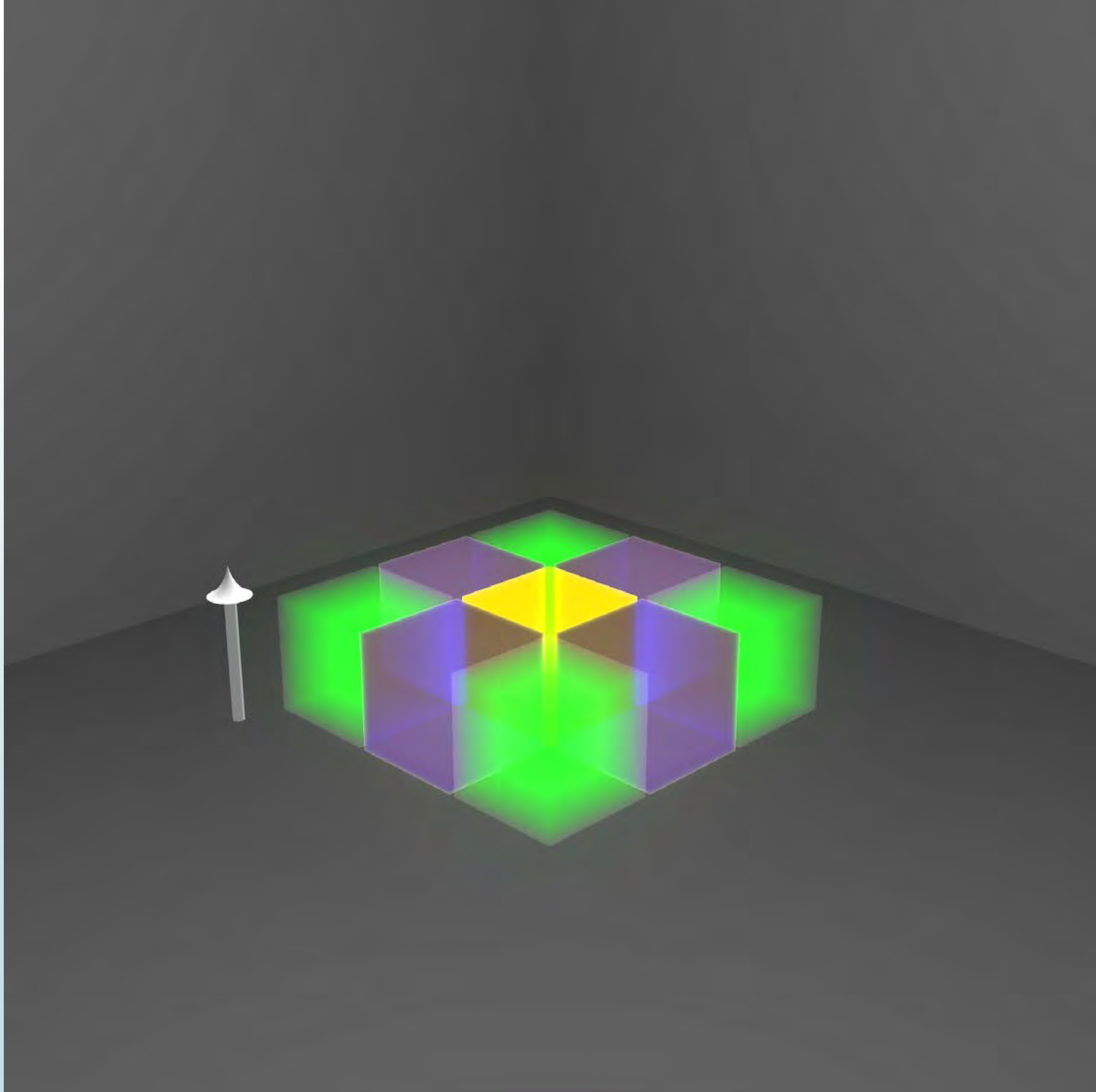
Building an N-dim first nearest neighborhood:

Going into 3D



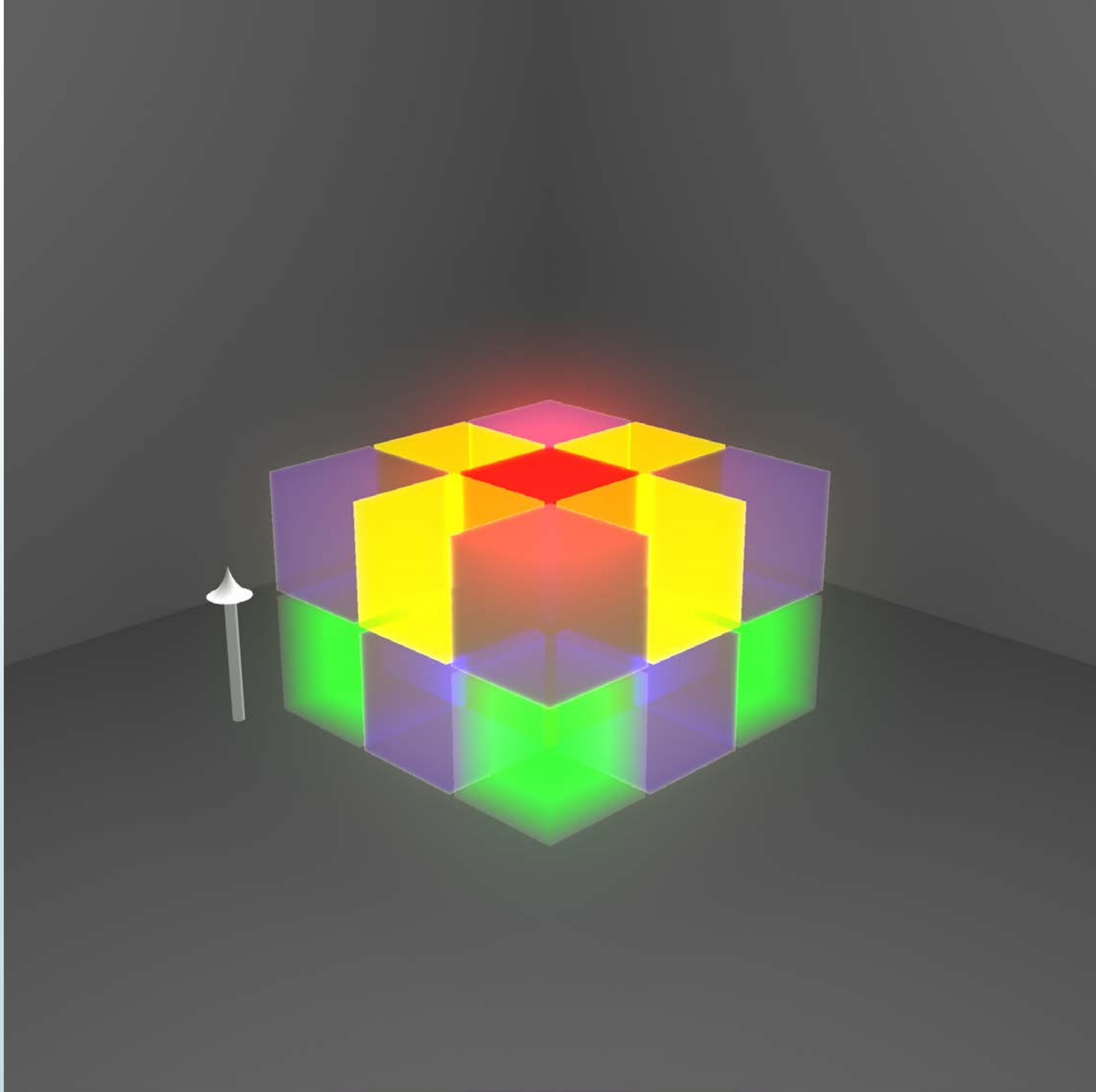
Building an N-dim first nearest neighborhood:

Extrude the 2D neighborhood along new direction



Building an N-dim first nearest neighborhood:

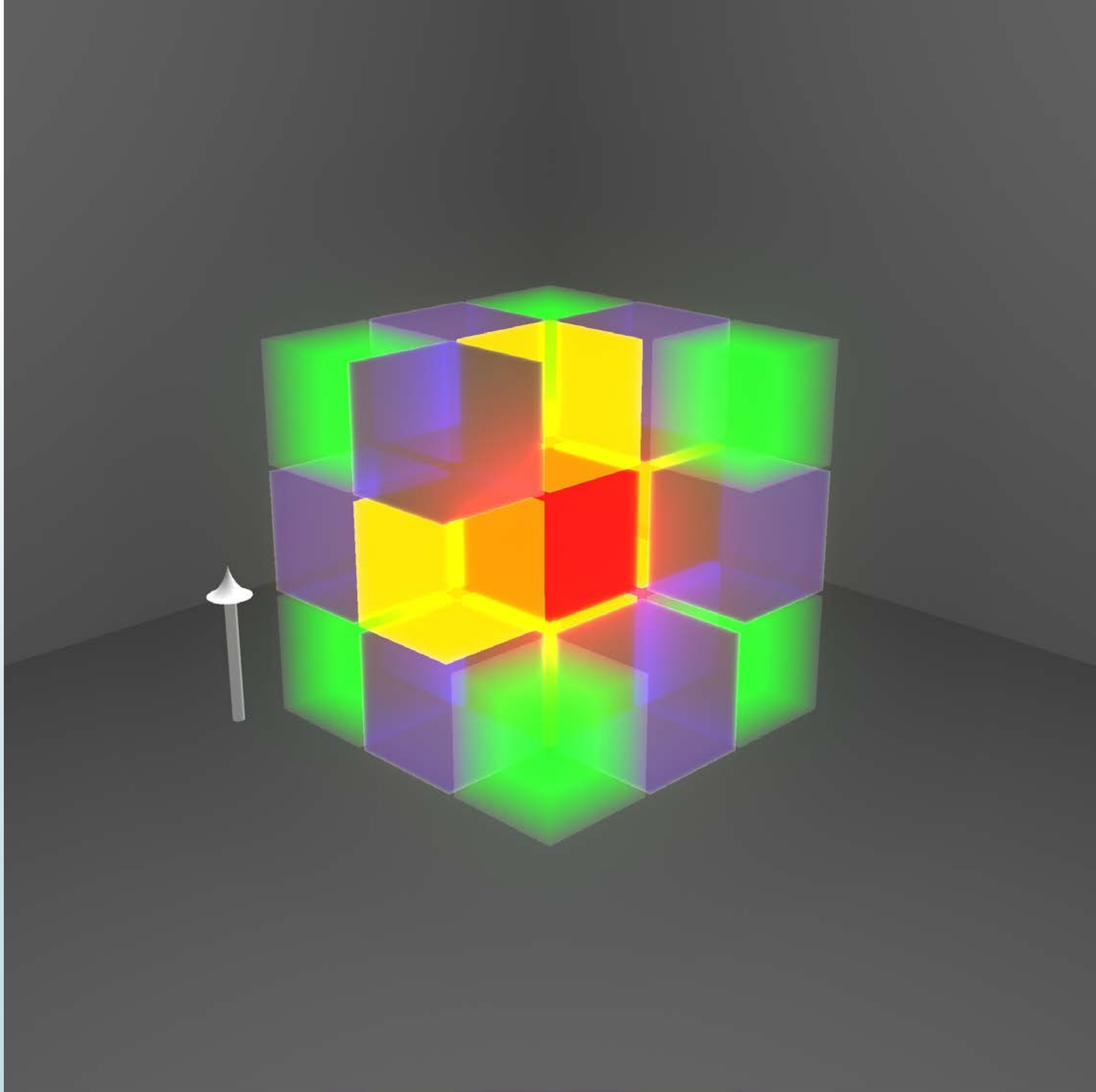
Center set is copied one forward and one backward



Building an N-dim first nearest neighborhood:

3D first nearest neighbors (cutaway)

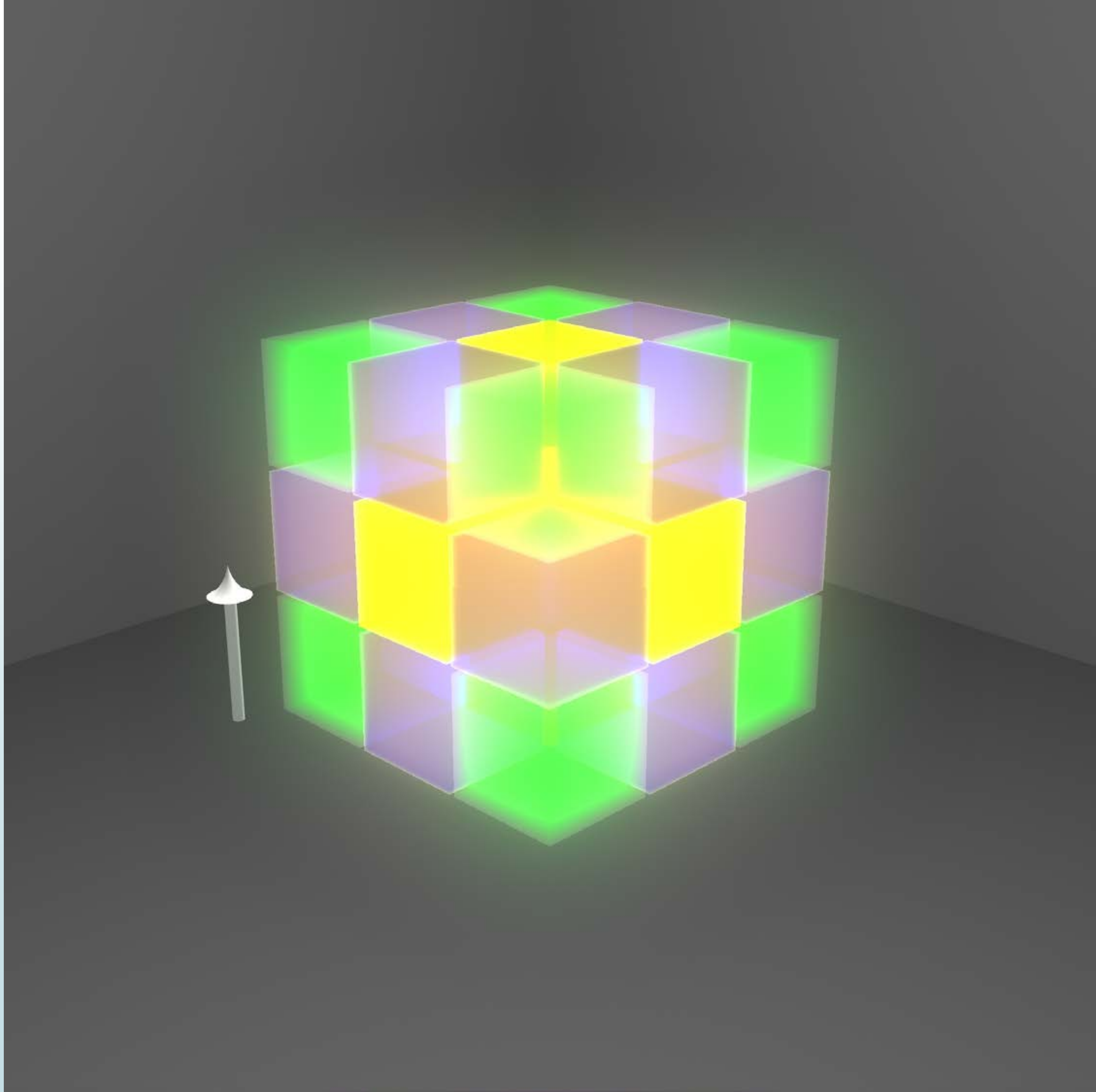
3 types: planes, lines, points



Building an N-dim first nearest neighborhood:

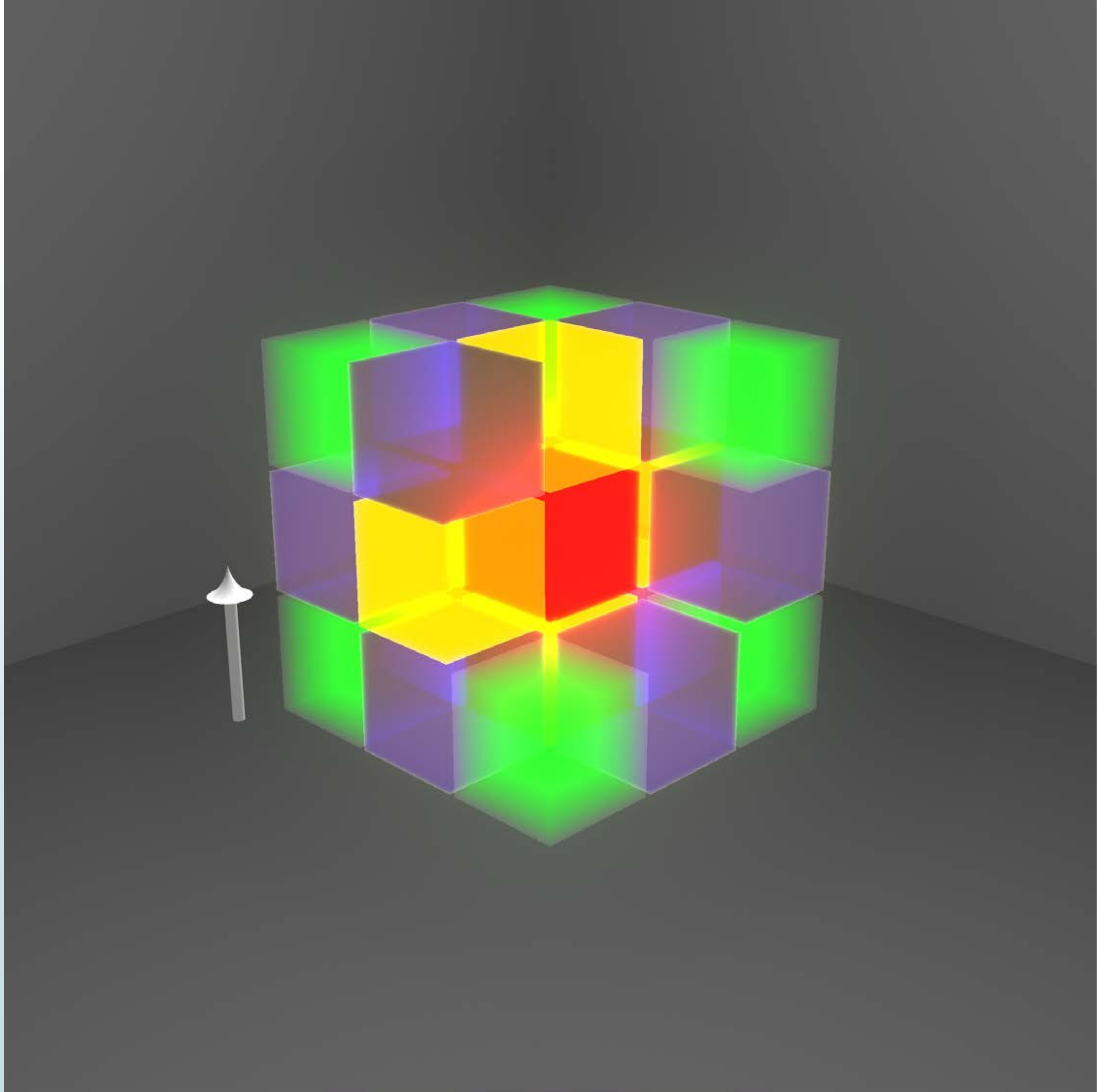
yellow = facial neighbors (2D)
blue = lines (1D)
green = points (0D)

All share a common geometry with the center



Building an N-dim first nearest neighborhood:

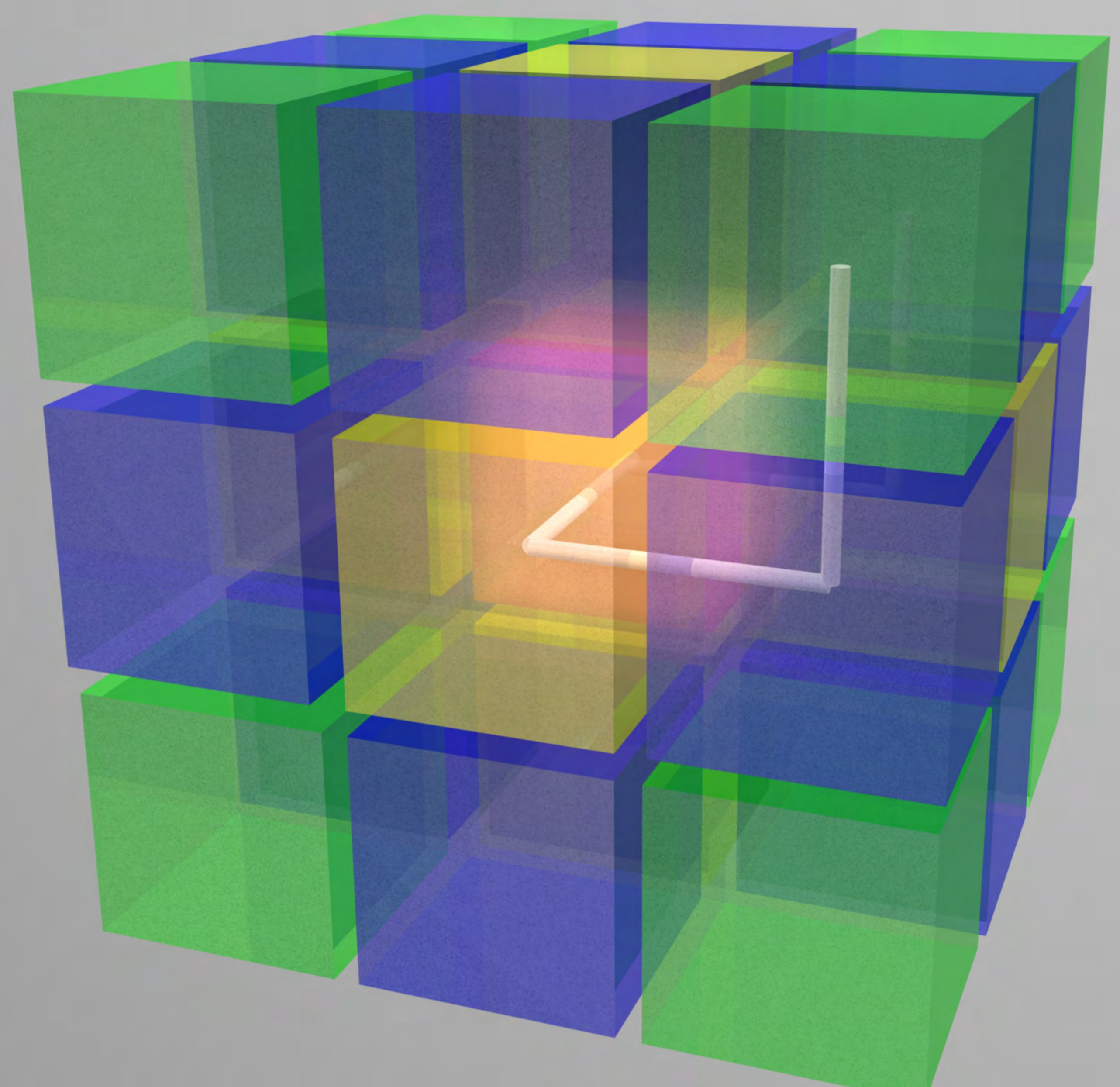
once more…



Building an N-dim first nearest neighborhood:

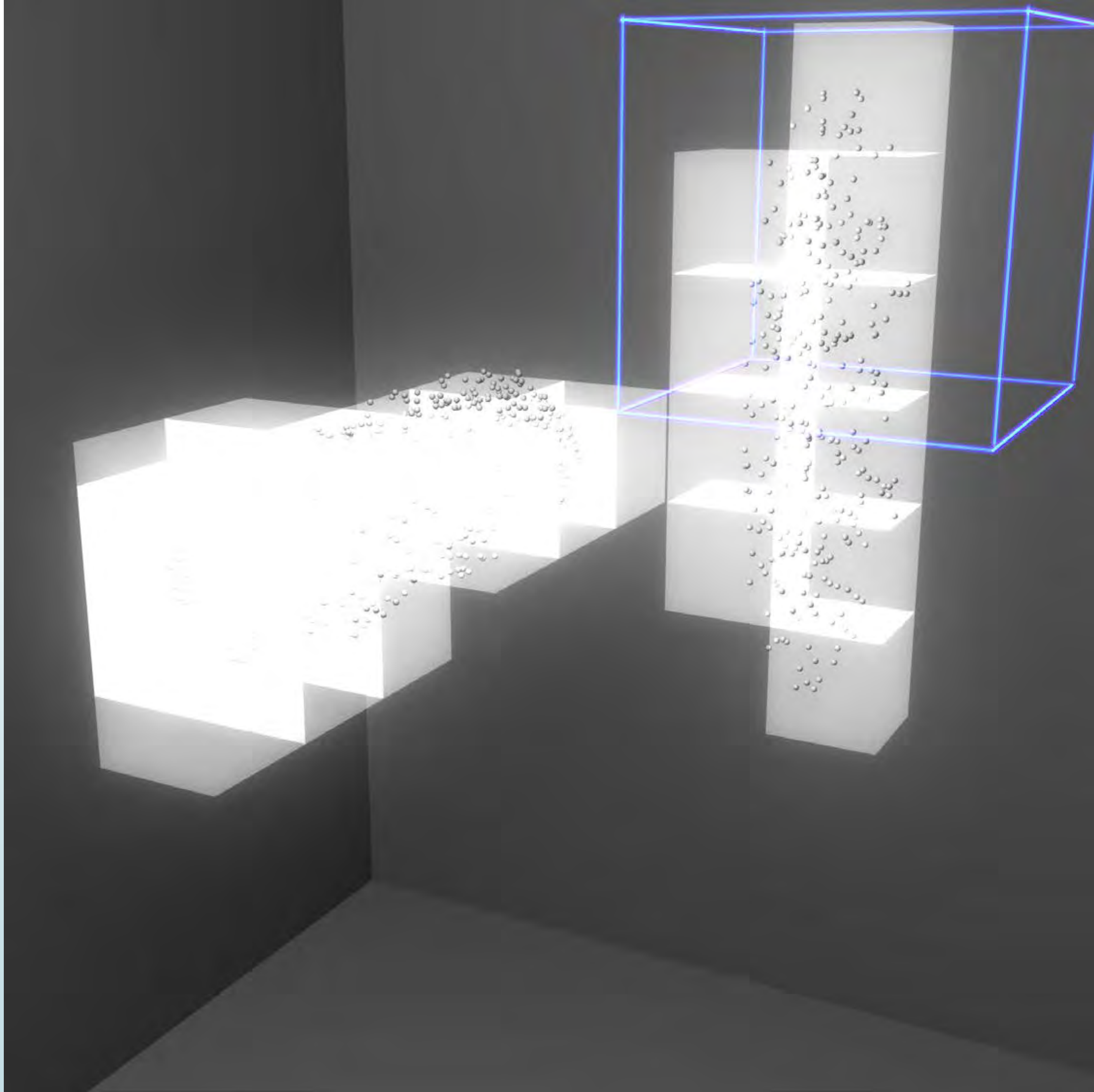
Distances:
How far away is the corner from the center?

In 4d? - problem



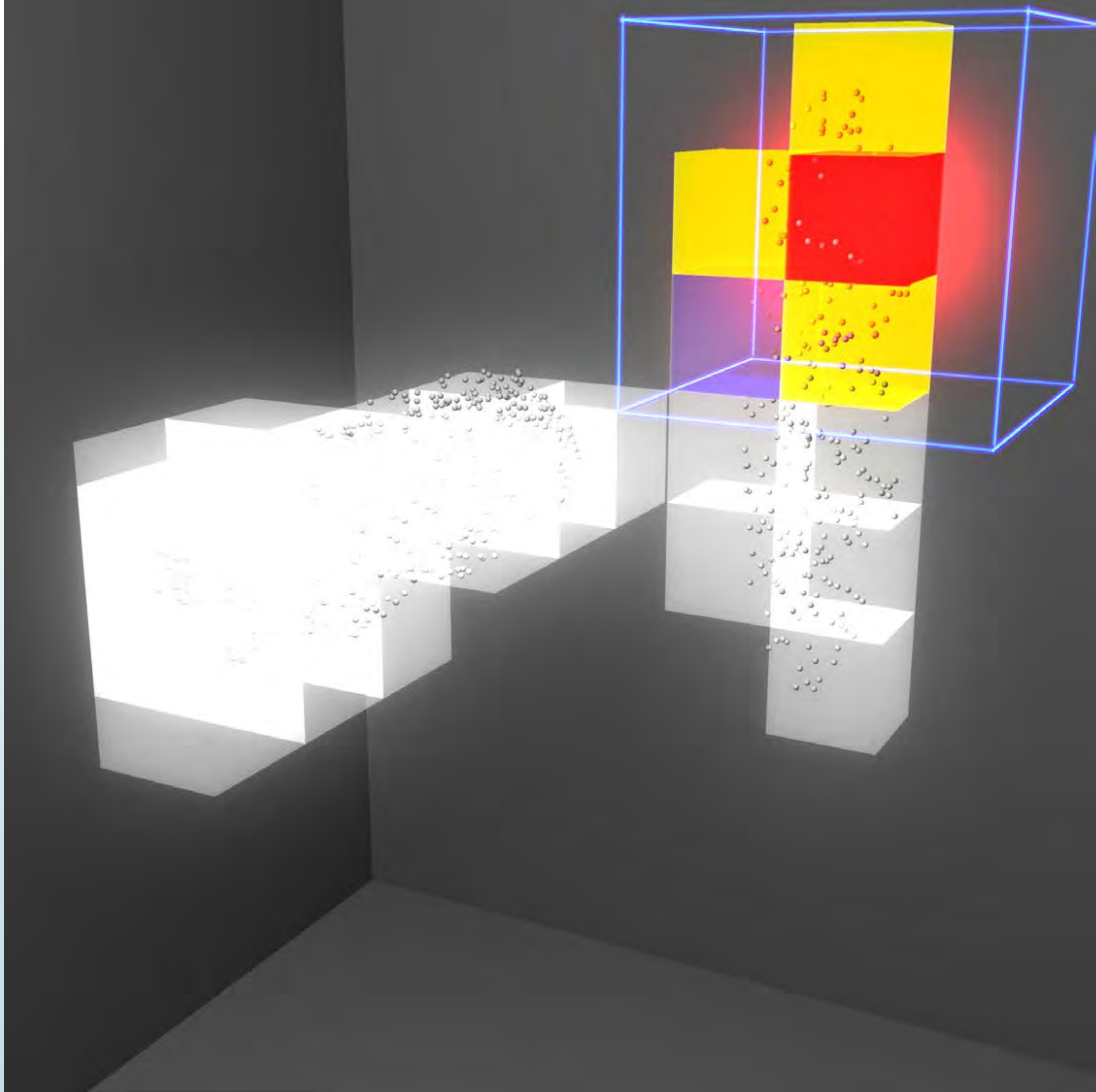
Building a space that contains data:

First Nearest Neighbors (1NN)



Building a space that contains data:

First Nearest Neighbors (1NN)



Matrices Calculated:

- Delta-r^2 – Euclidean distance between two partitions
- 1st Nearest Neighbor (1NN) – 0/1 for any partitions within neighborhood
- Delta-L^2 – Path length between two partitions (connected)
- Line-Of-Sight (LOS) – 0/1 for any partitions within Line-Of-Sight of each other

Matrices – Logical Array Hinge

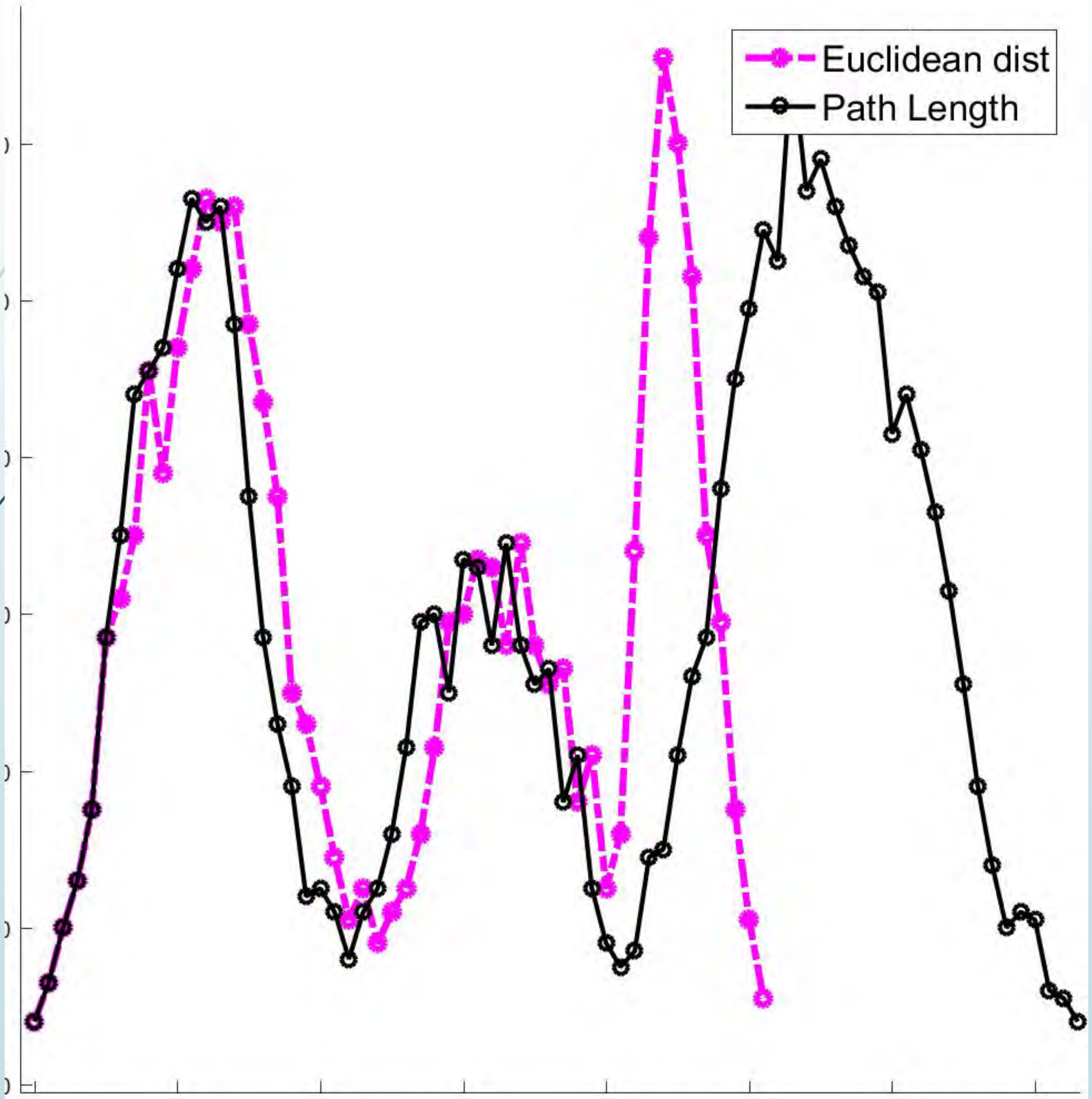
- Form the 1NN matrix:
  - Each row represents a single partition
  - Each column are all of the other partitions
  - for each partition, find all other partitions within +/-1 of individual variable bin addresses
  - Assign a “1” for the neighbors
- Form the Path Length matrix:
  - Starting from a partition
  - Find all 1NN of initial partition
  - “Swing” 1NNs to search for new rows
  - Find 1NN of the 1NN = 2NN
  - Repeat until done, store the path length
  - Logical Array Hinge
- Form the Connection Matrix
  - All partitions connected via a path
  - Replace path lengths with “1”

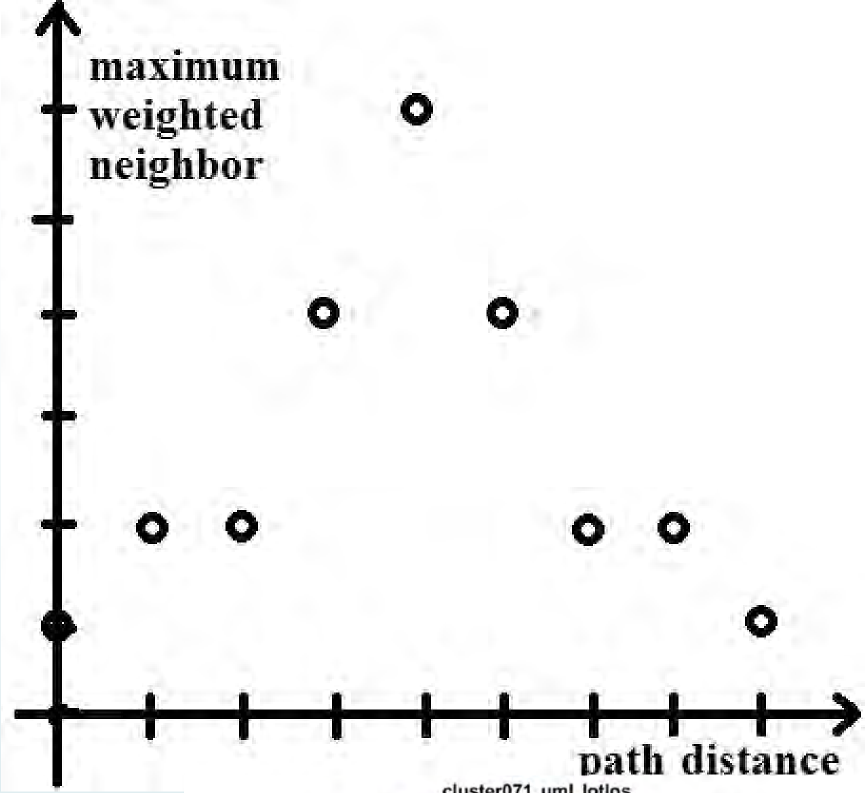
$$
\left(\begin{array}{ccccccccccccc}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1
\end{array}\right)
\begin{array}{l}
\Leftarrow \text{1NN of row 1} \\
\Leftarrow \text{2NN of row 1} \\
\Leftarrow \text{2NN of row 1} \\
\Leftarrow \text{2NN of row 1} \\
\Leftarrow \text{2NN of row 1} \\
\\
\\
\Leftarrow \text{2NN of row 1} \\
\\
\\
\\
\\
\\
\end{array}
$$

Clusters - Hierarchy

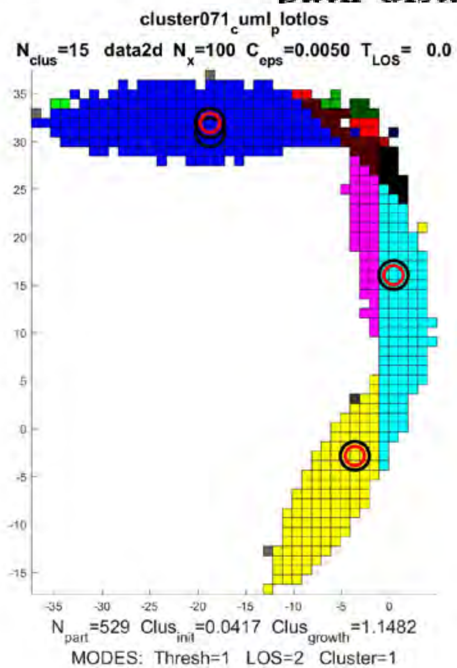
- Global - Δr_{kl}^2 based on weights, w_k
- Connected – all partitions connected via a path, $\Delta \ell_{kl}$
- Line of Sight – LOS – all partitions within view of each other
- Path Length – based on weights, w_k
- Simple nearest neighbors (1NN, 2NN, 3NN)
- Magnitude sorted (simplest)

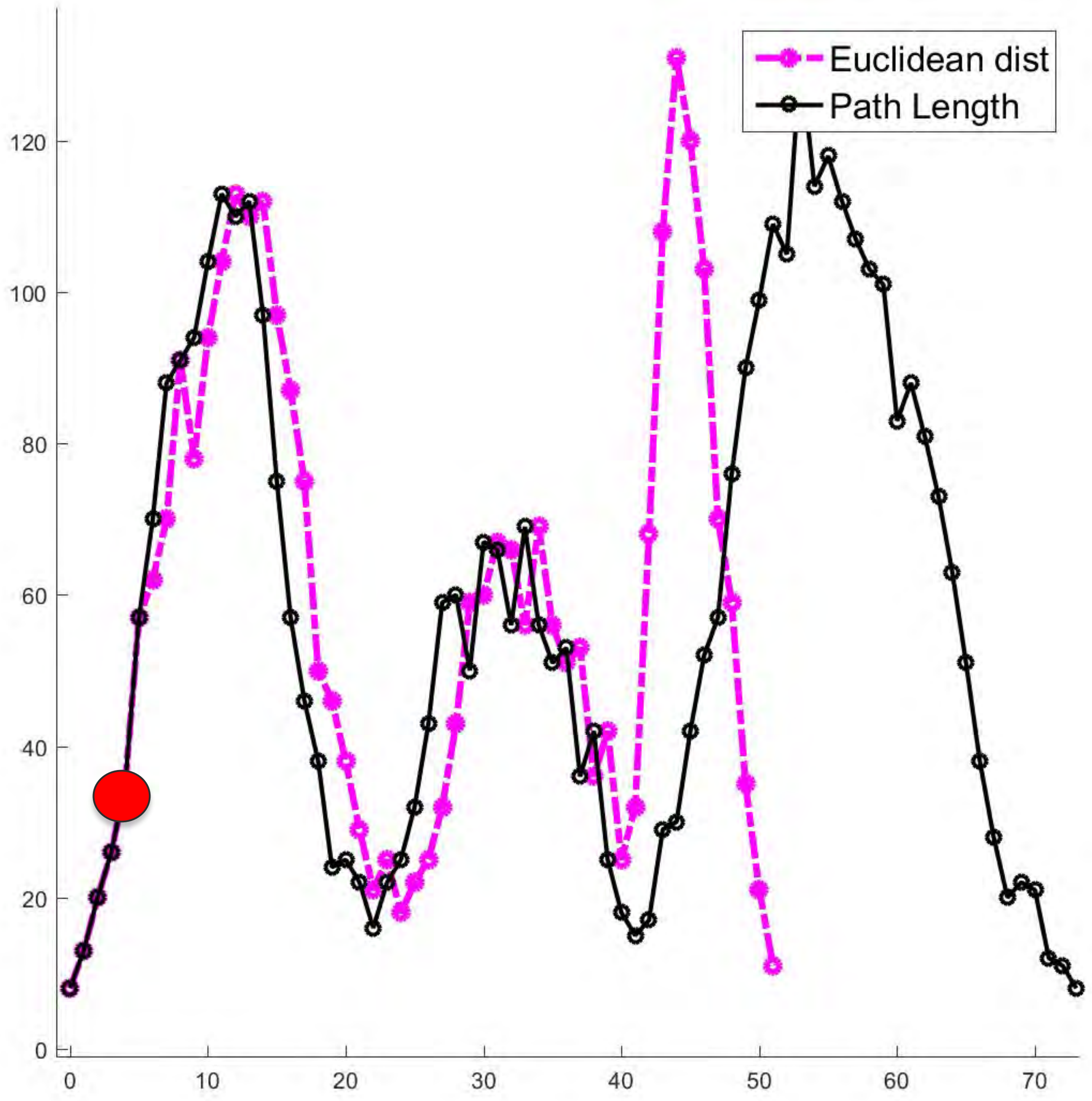
Global Delta-r^2 + Path Length



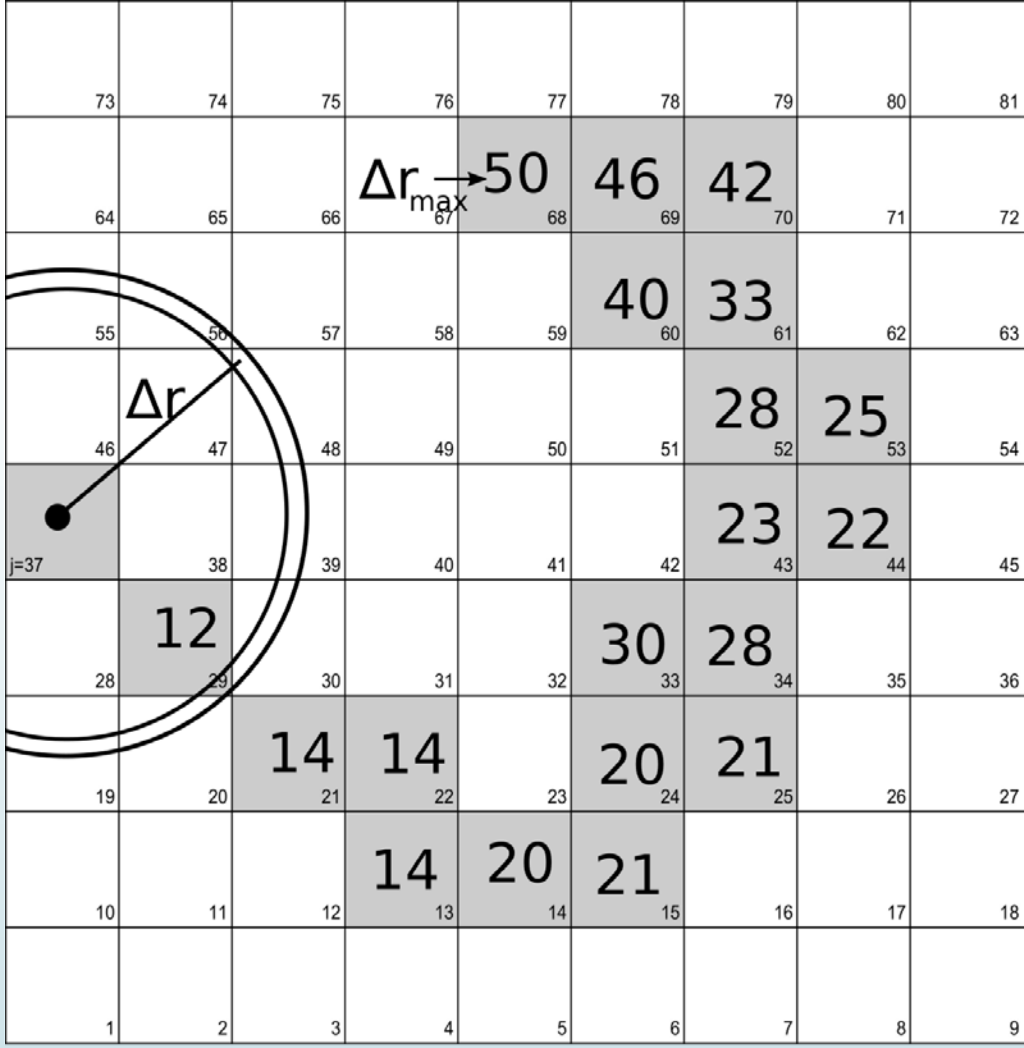


LOS

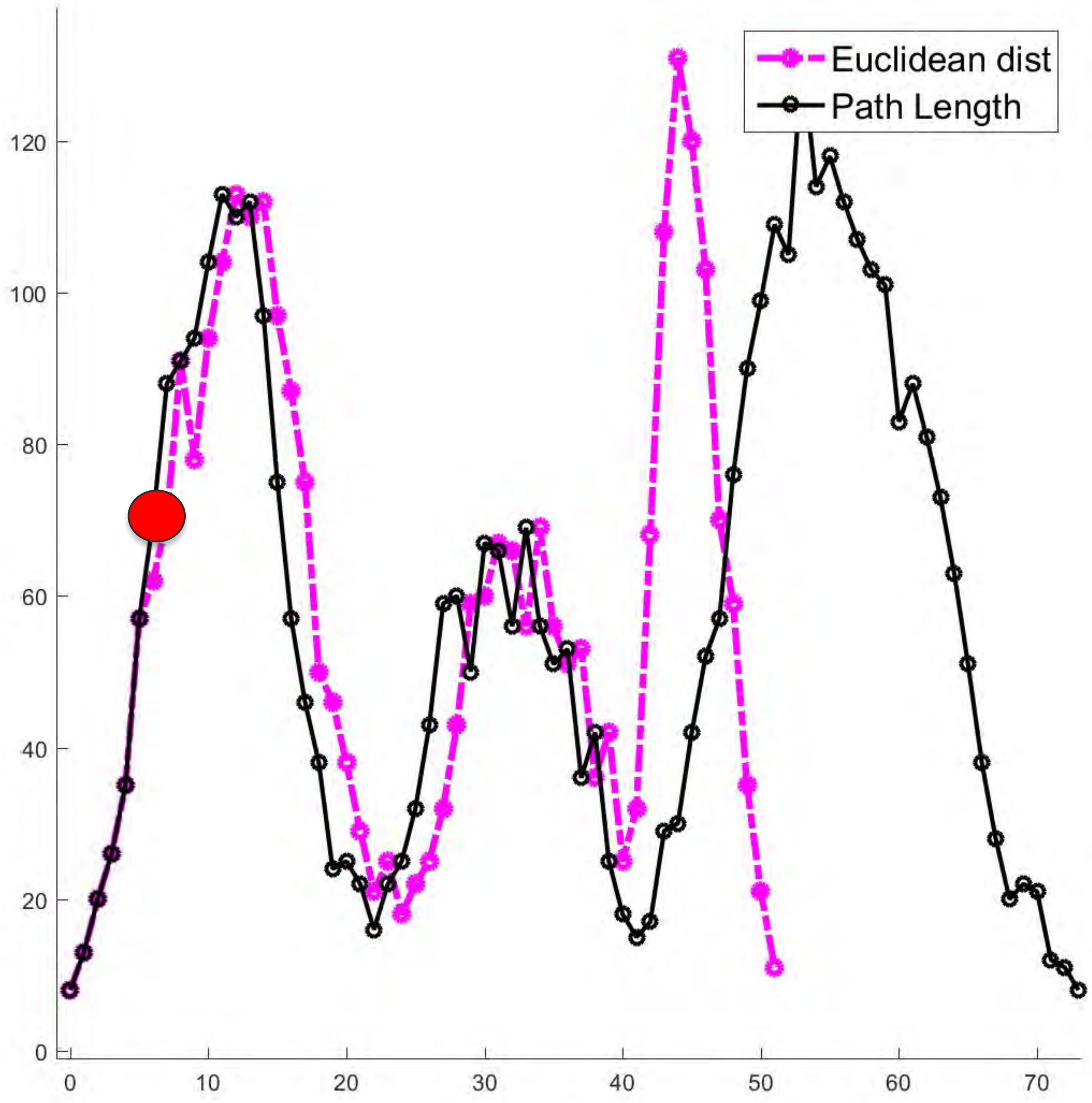




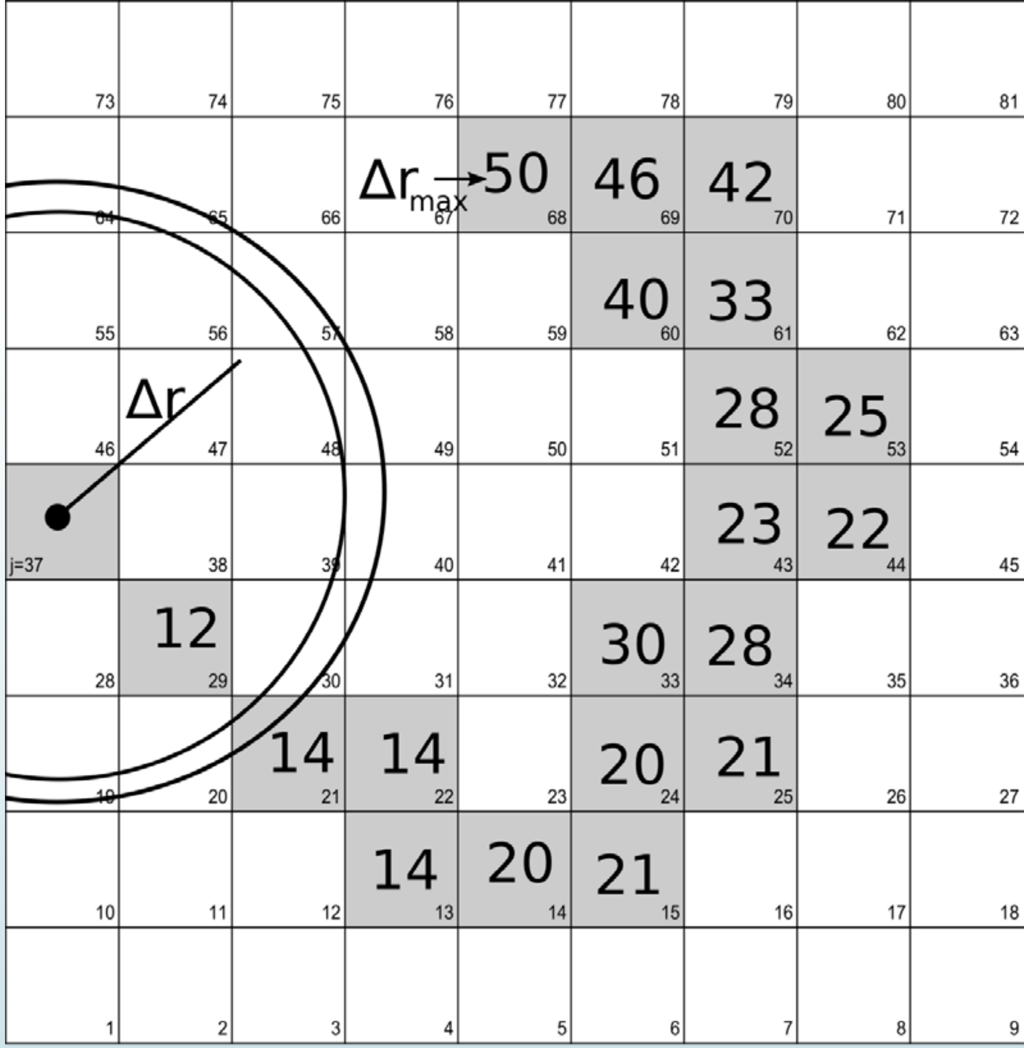
Euclidean dist
Path Length



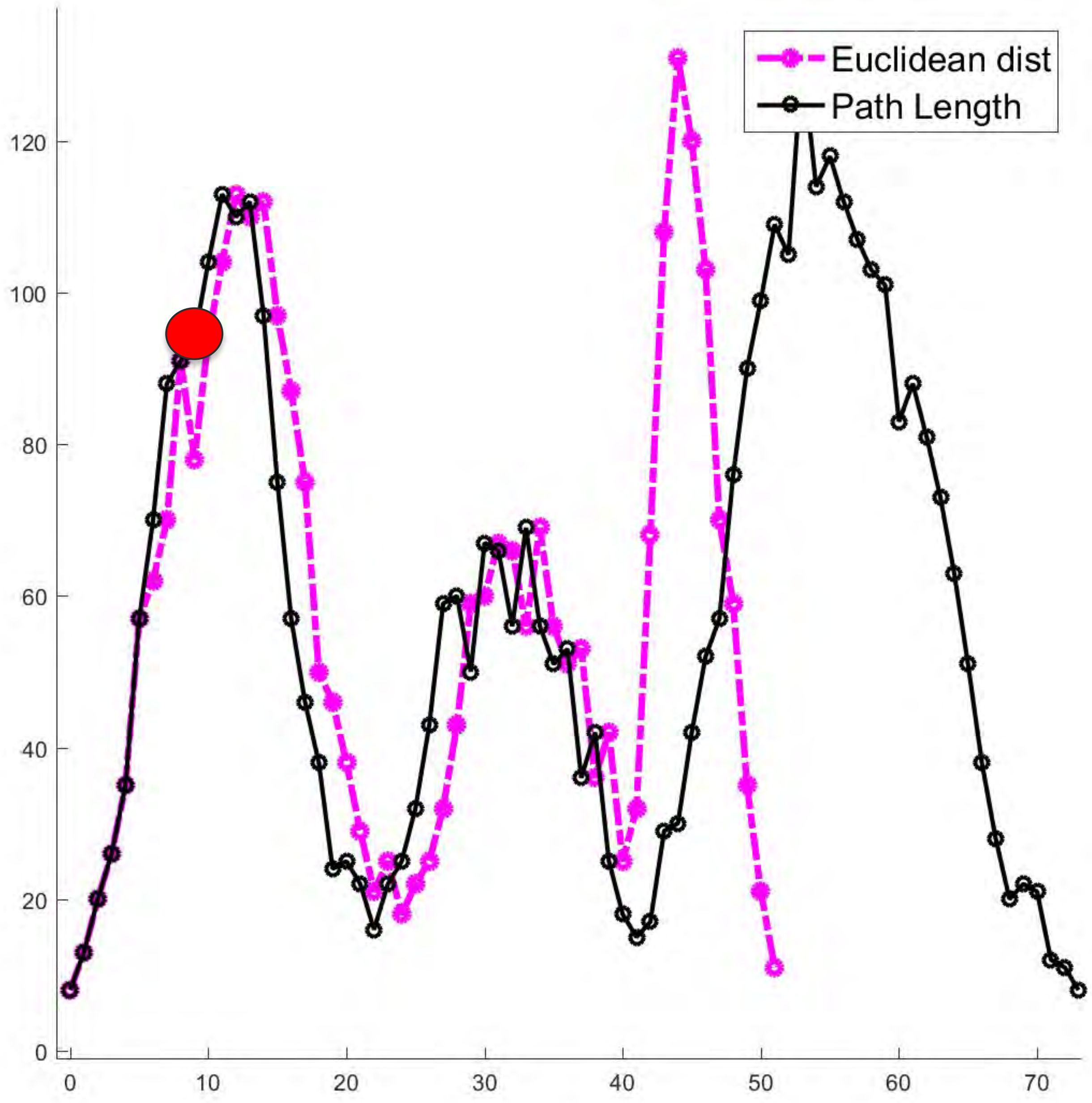
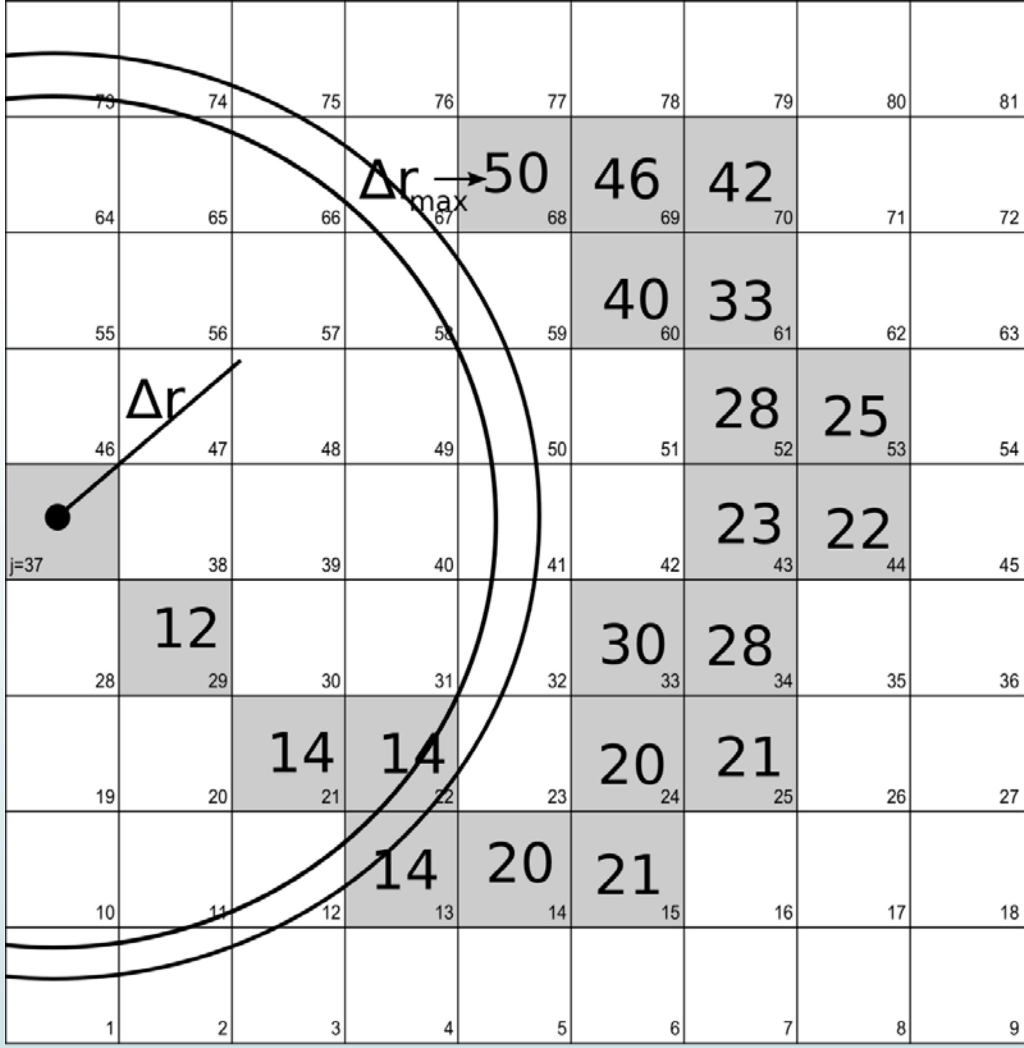
Δr_{max}
Δr
j=37

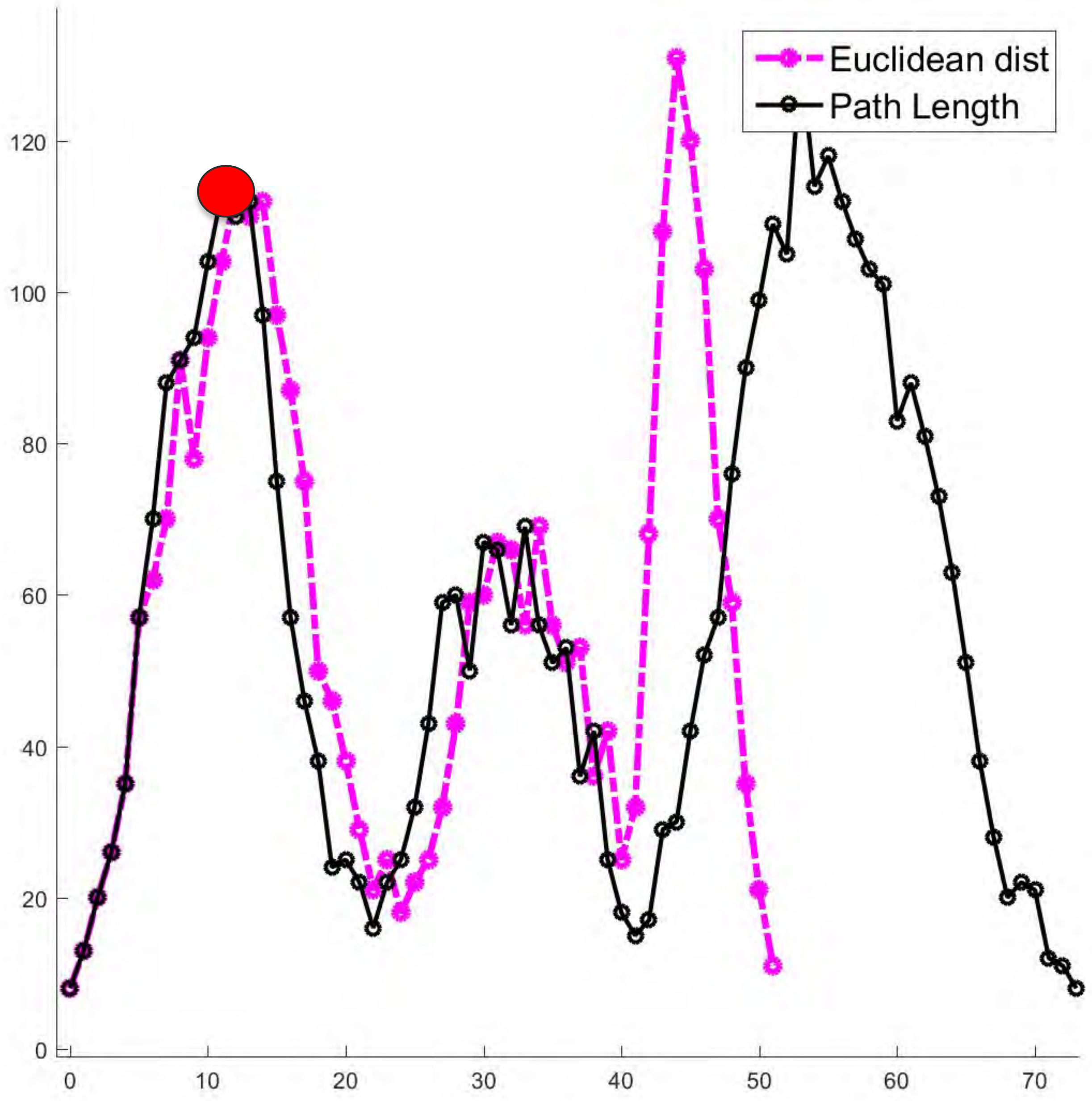


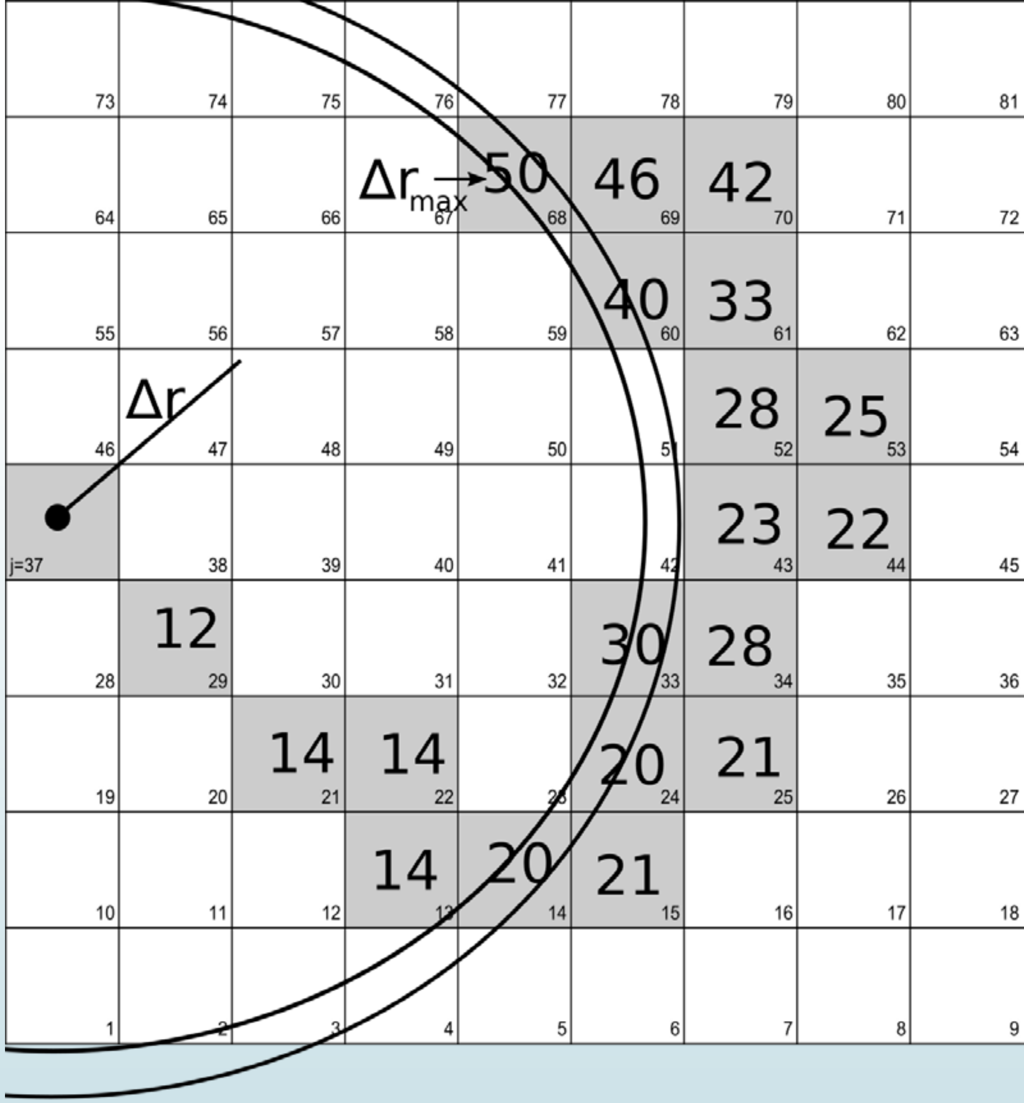
Euclidean dist
Path Length
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70

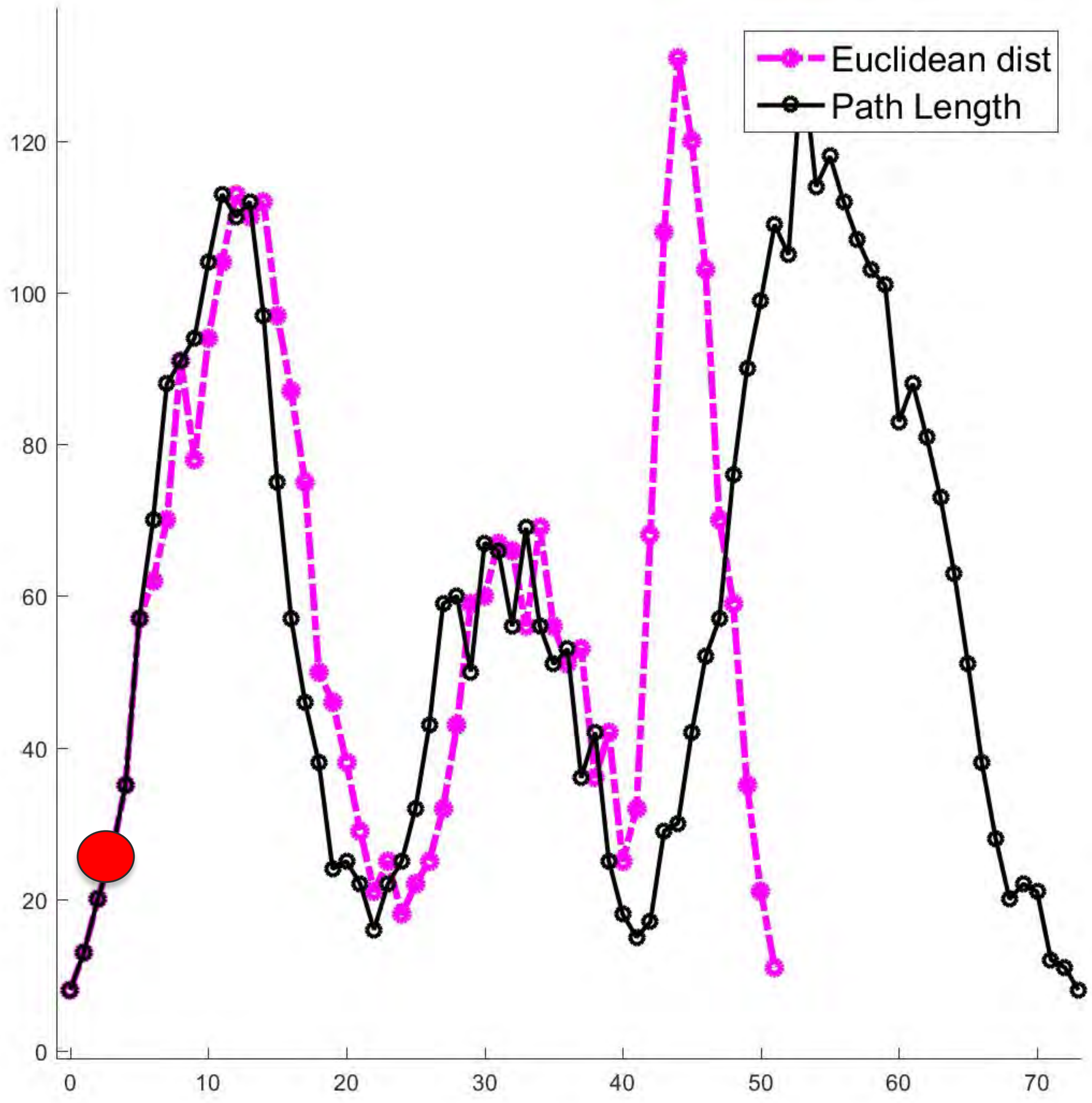


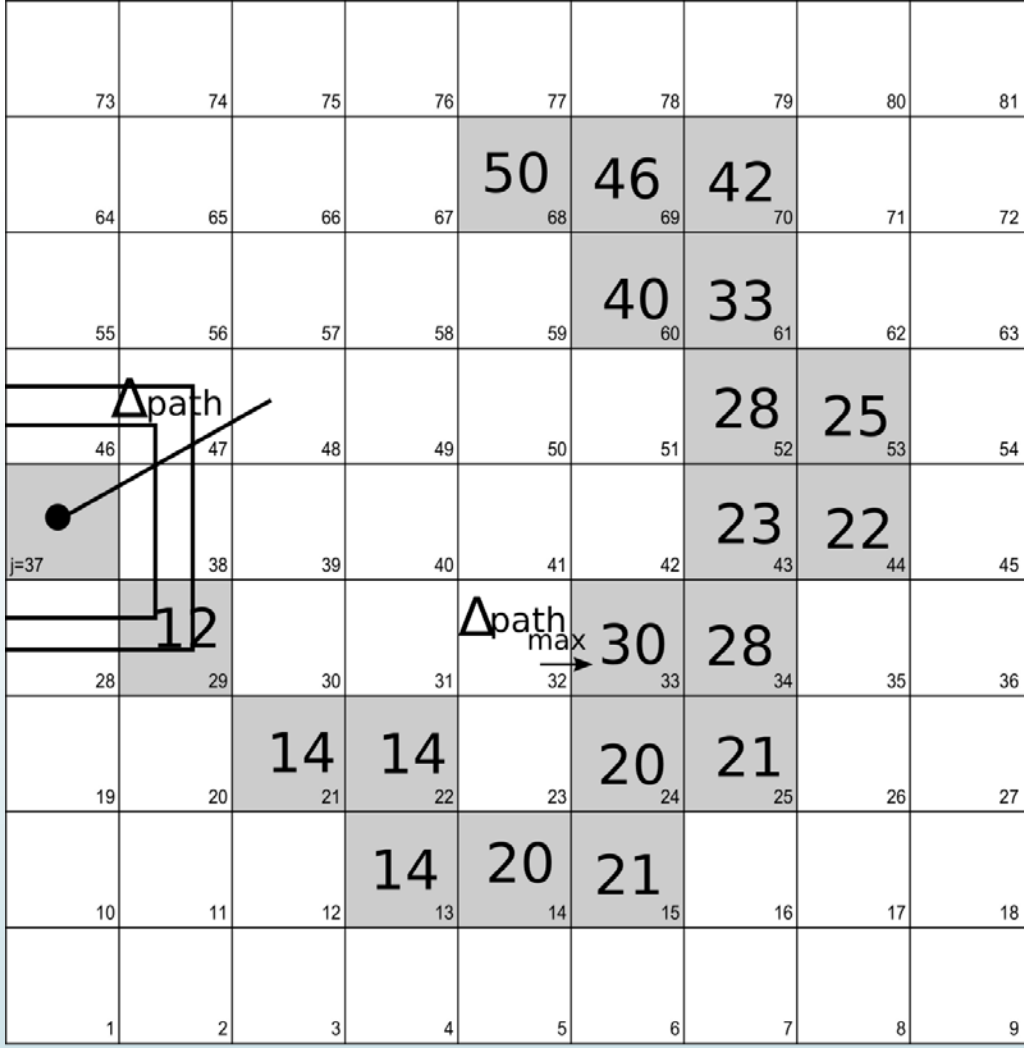
Δr
Δr_max
j=37
50
46
42
40
33
28
25
23
22
12
30
28
14
14
20
21
14
20
21

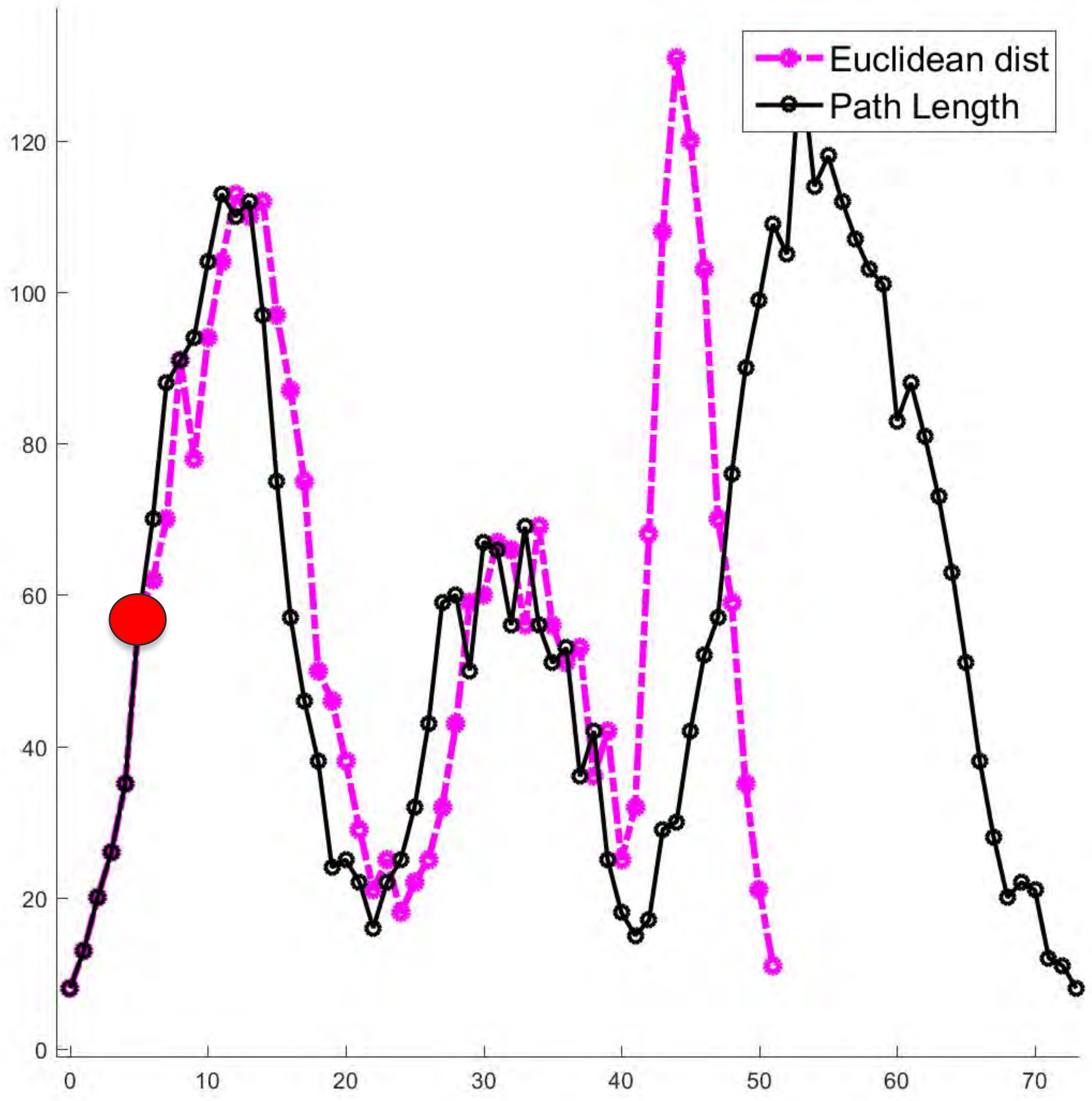

Euclidean dist
Path Length
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70

Δr
Δr_{max}
j=37
50
46
42
40
33
28
25
23
22
12
30
28
14
14
20
21
14
20
21

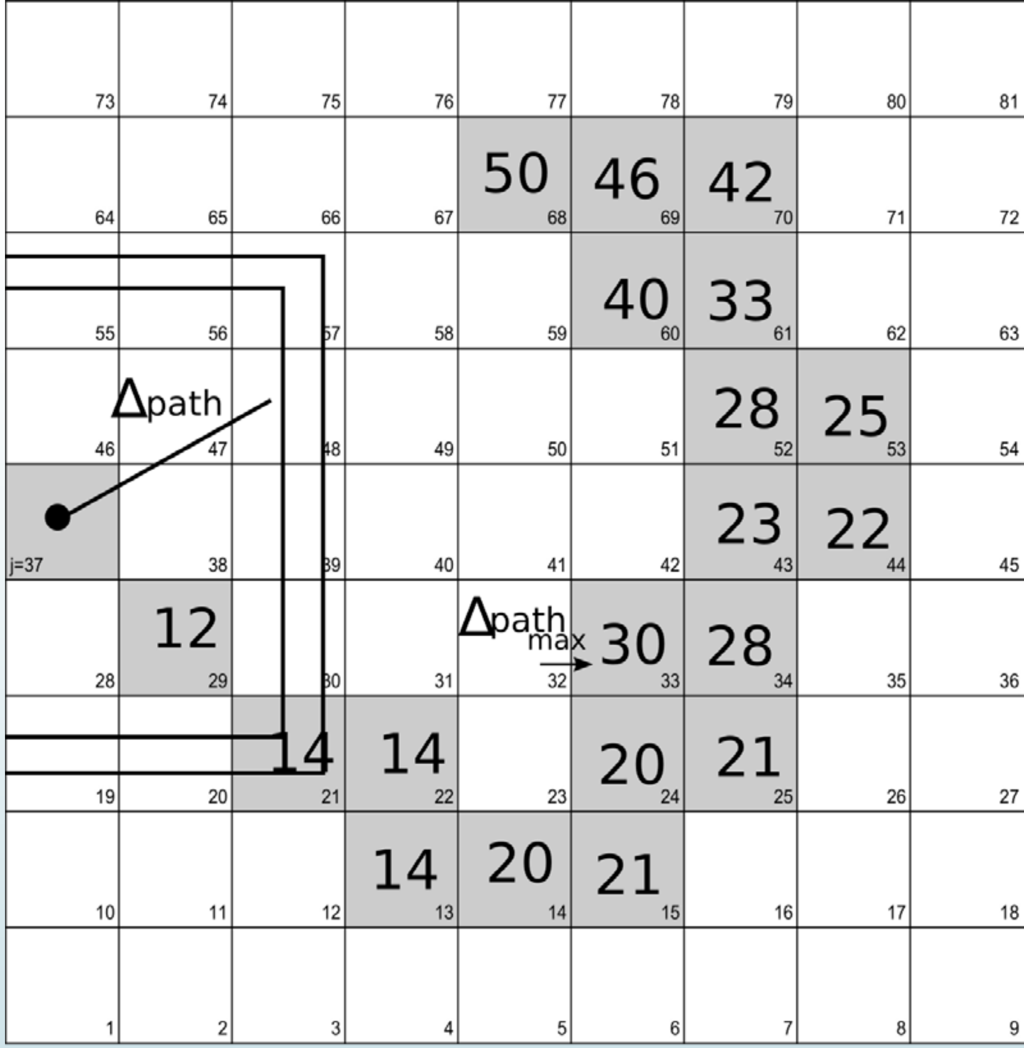

Euclidean dist
Path Length
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70

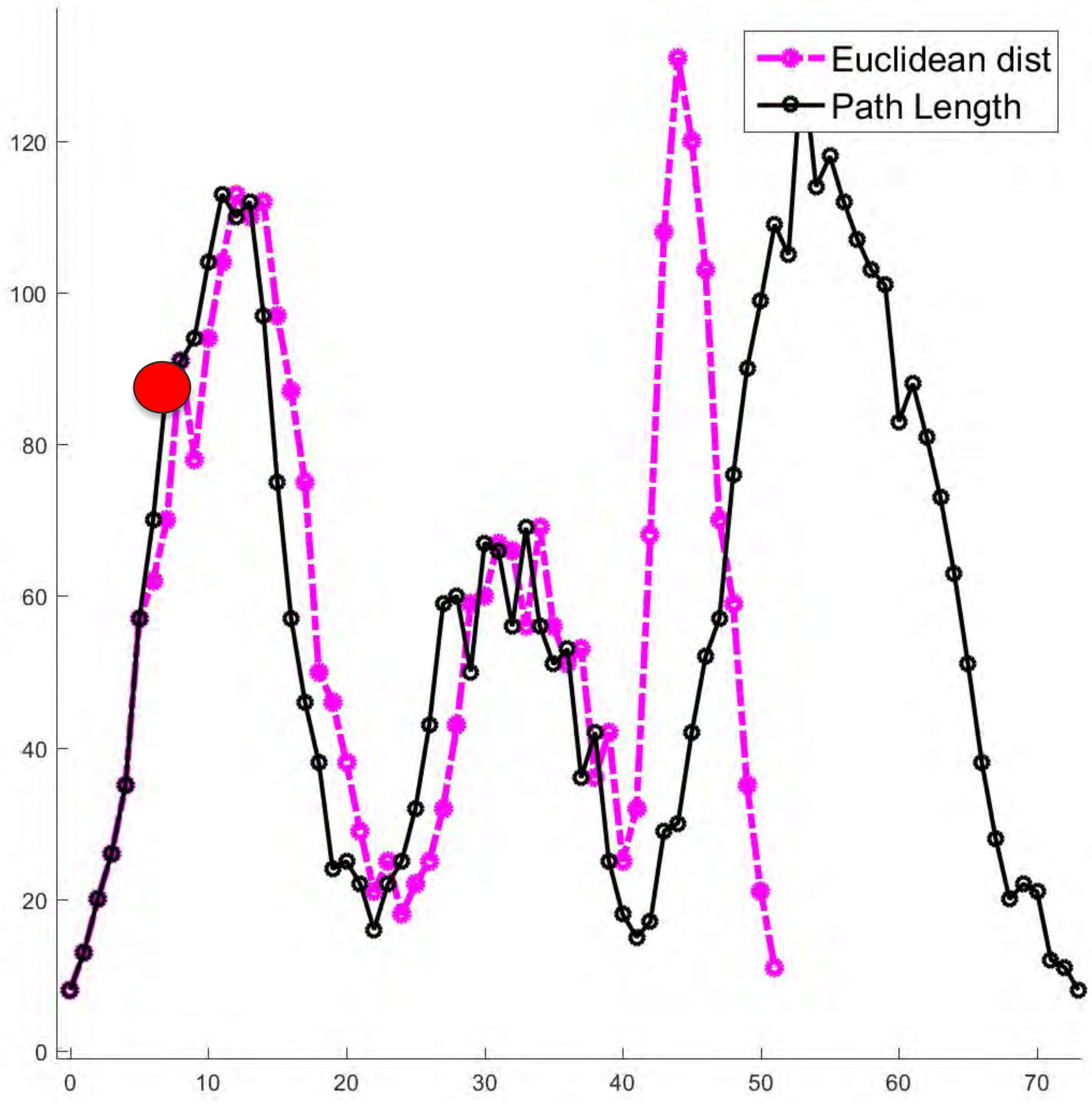

Δr
Δr_max
j=37
50
46
42
40
33
28
25
23
22
12
30
28
14
14
20
21
14
20
21


Euclidean dist
Path Length
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70

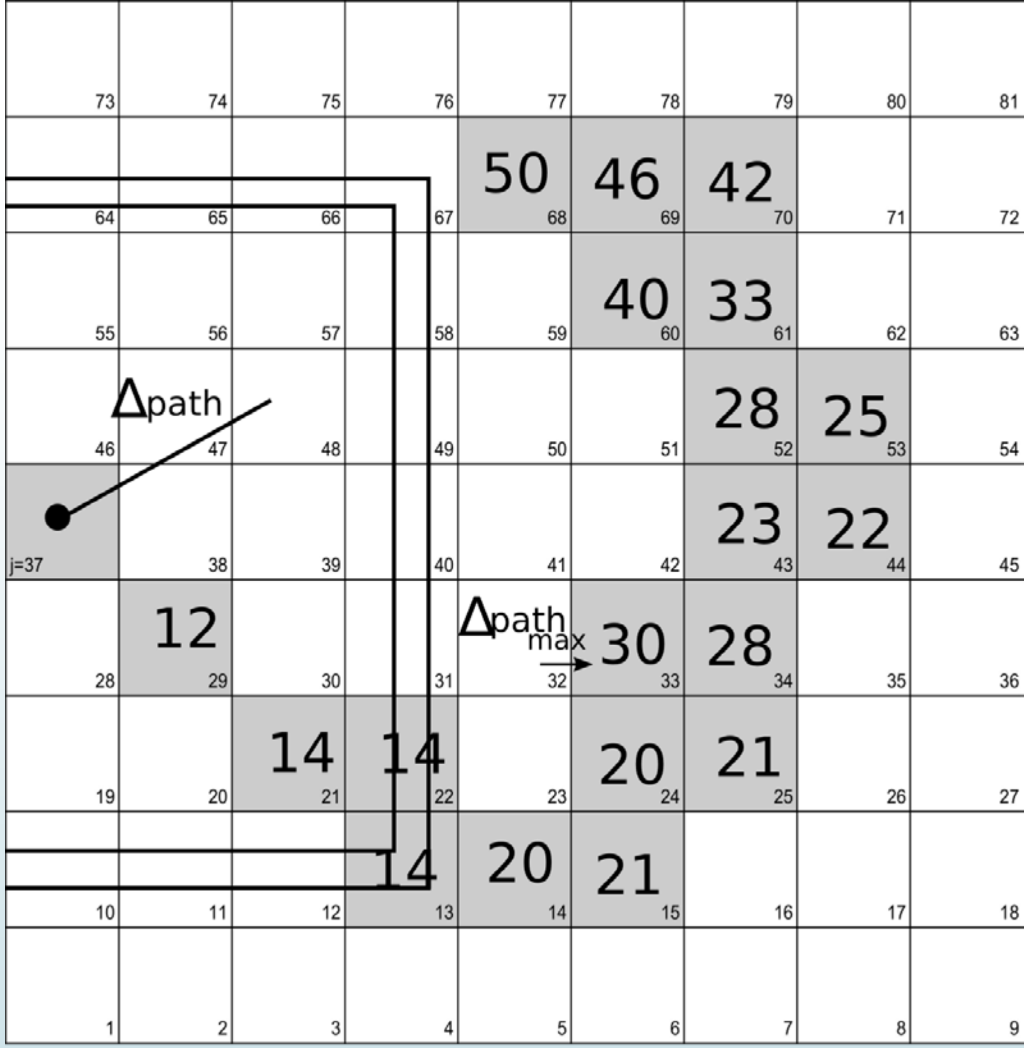

Δpath
j=37
12
Δpath max
50
46
42
40
33
28
25
23
22
30
28
14
14
20
21
14
20
21


Euclidean dist
Path Length
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70

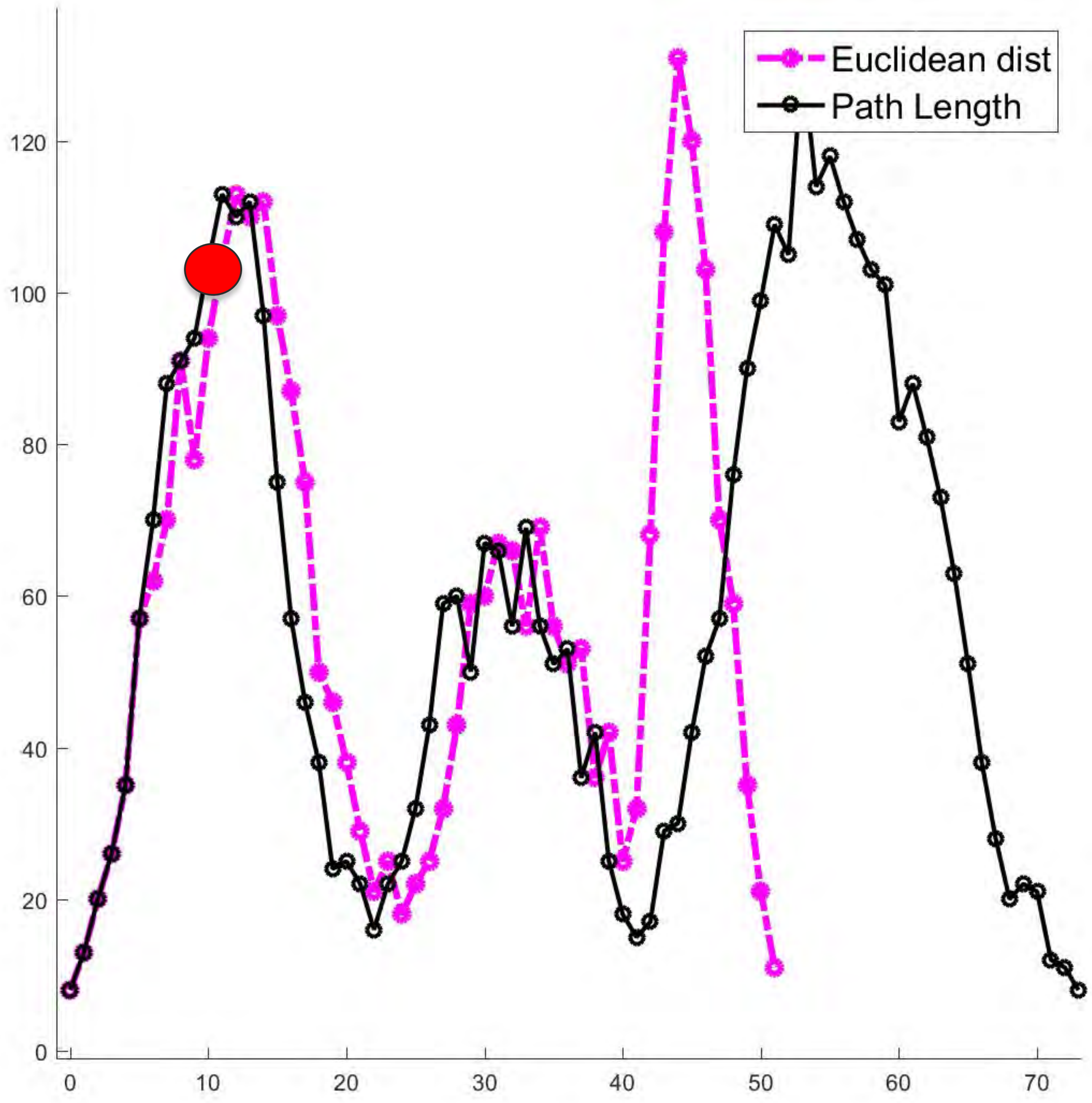

Δpath
Δpath max
j=37
50
46
42
40
33
28
25
23
22
12
30
28
14
14
20
21
14
20
21



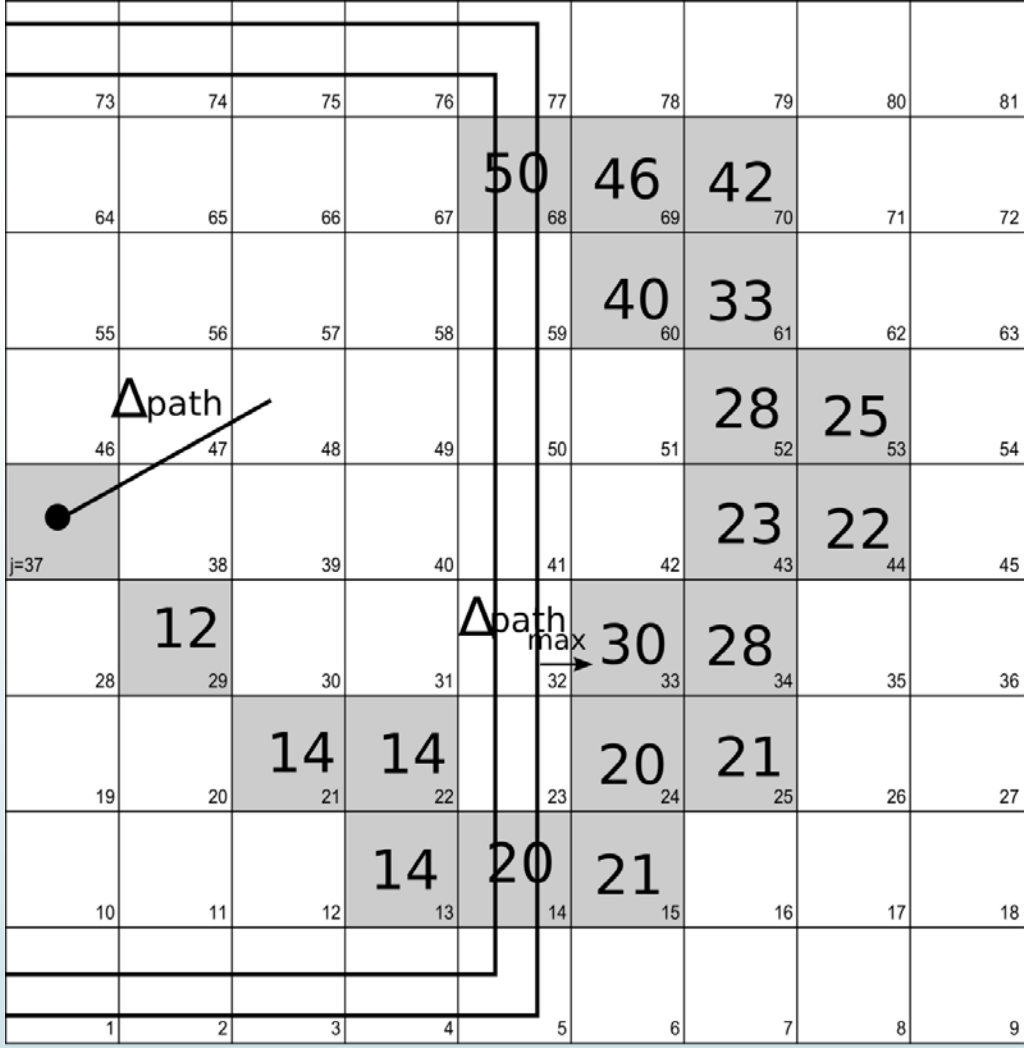
Euclidean dist
Path Length
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70



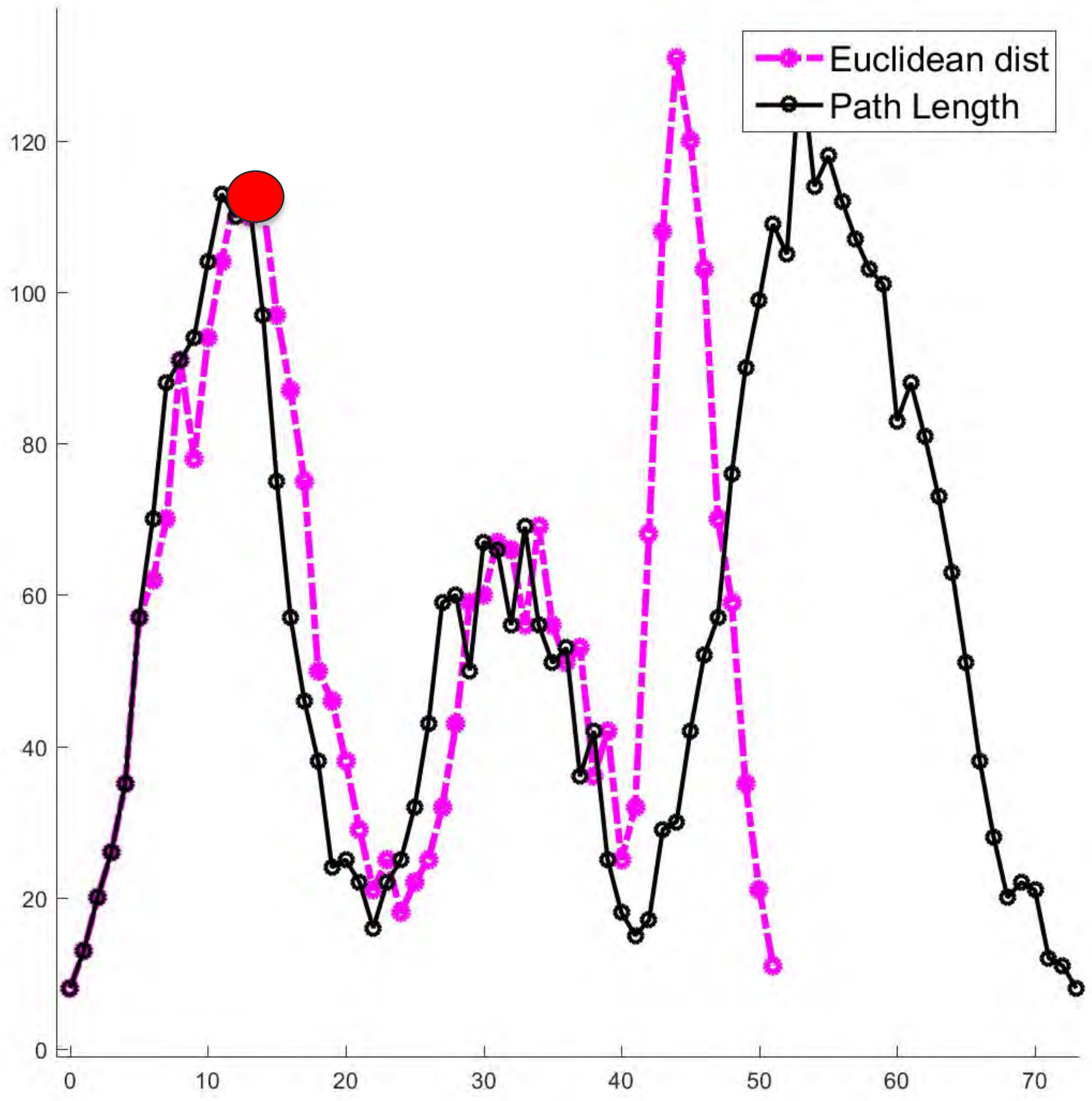
Δpath
Δpath max
j=37
50
46
42
40
33
28
25
23
22
12
30
28
14
14
20
21
14
20
21



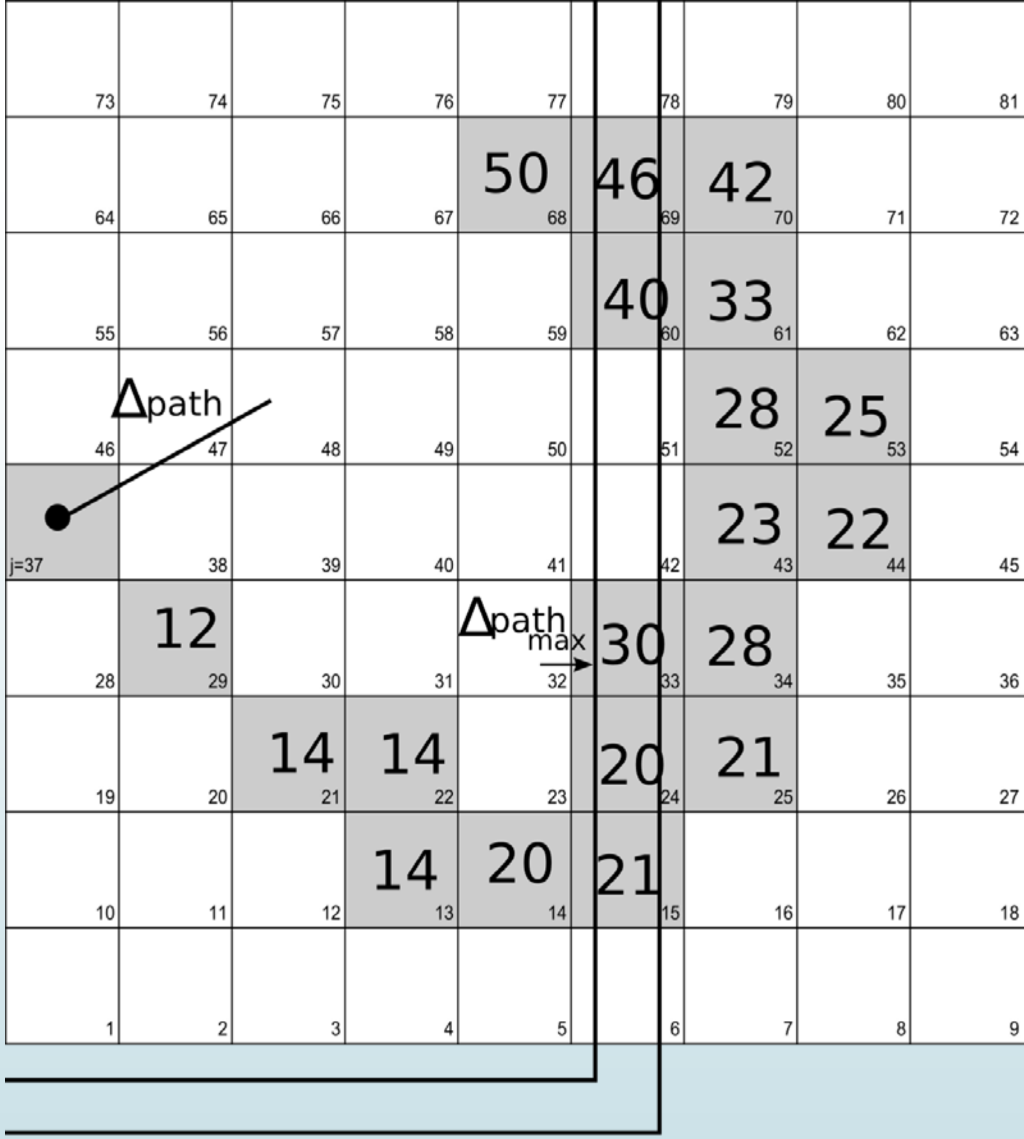
Euclidean dist
Path Length
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70



Δpath
Δpath max
j=37
50
46
42
40
33
28
25
23
22
12
30
28
14
14
20
21
14
20
21



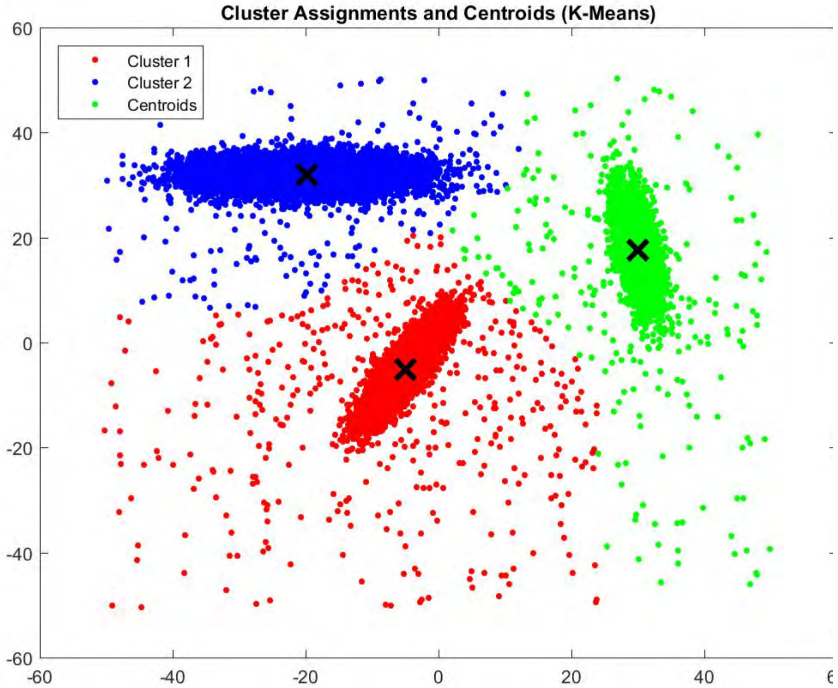
Euclidean dist
Path Length
0
20
40
60
80
100
120
0
10
20
30
40
50
60
70

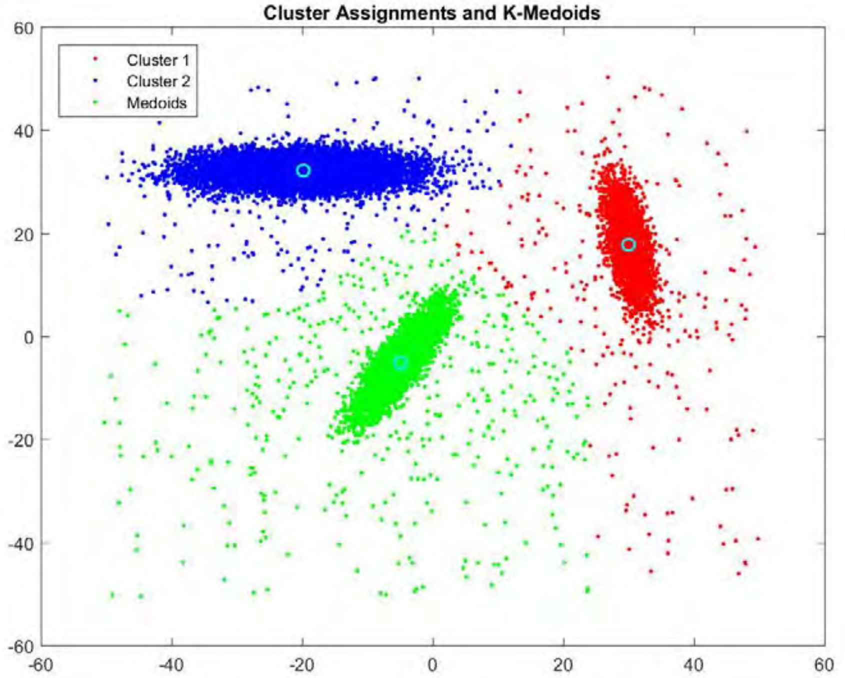


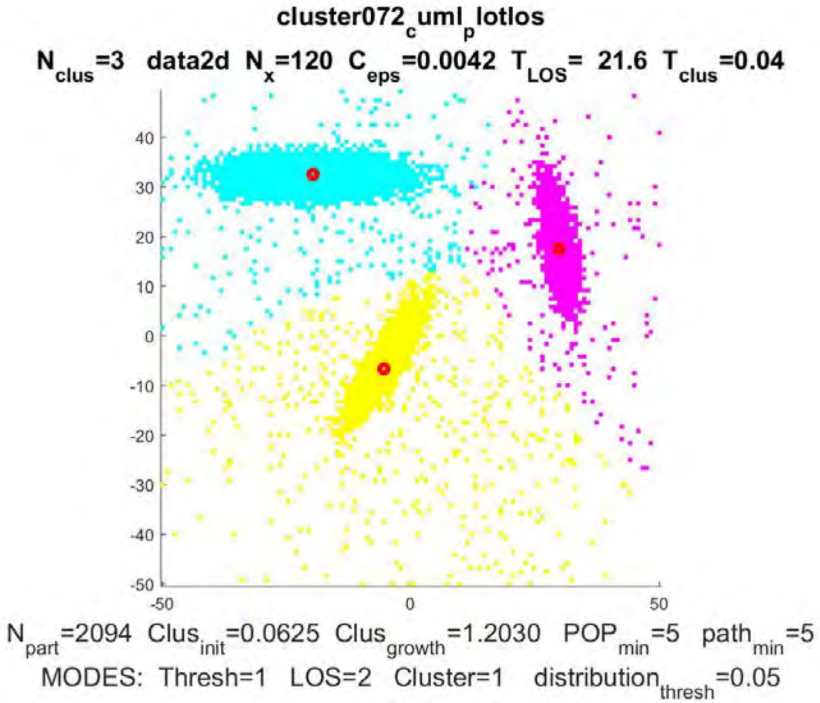
Δpath
Δpath$_{max}$
j=37
50
46
42
40
33
28
25
23
22
12
30
28
14
14
20
21
14
20
21

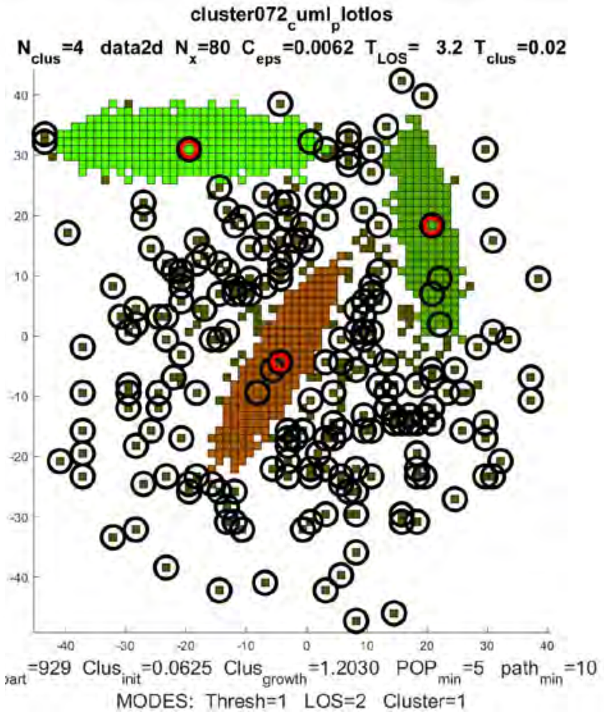
Other Techniques

- N-dim Cluster algoritms:
  - K-Means
  - K-Medoids
  - DBSCAN
  - EM
  - ELKI (package)

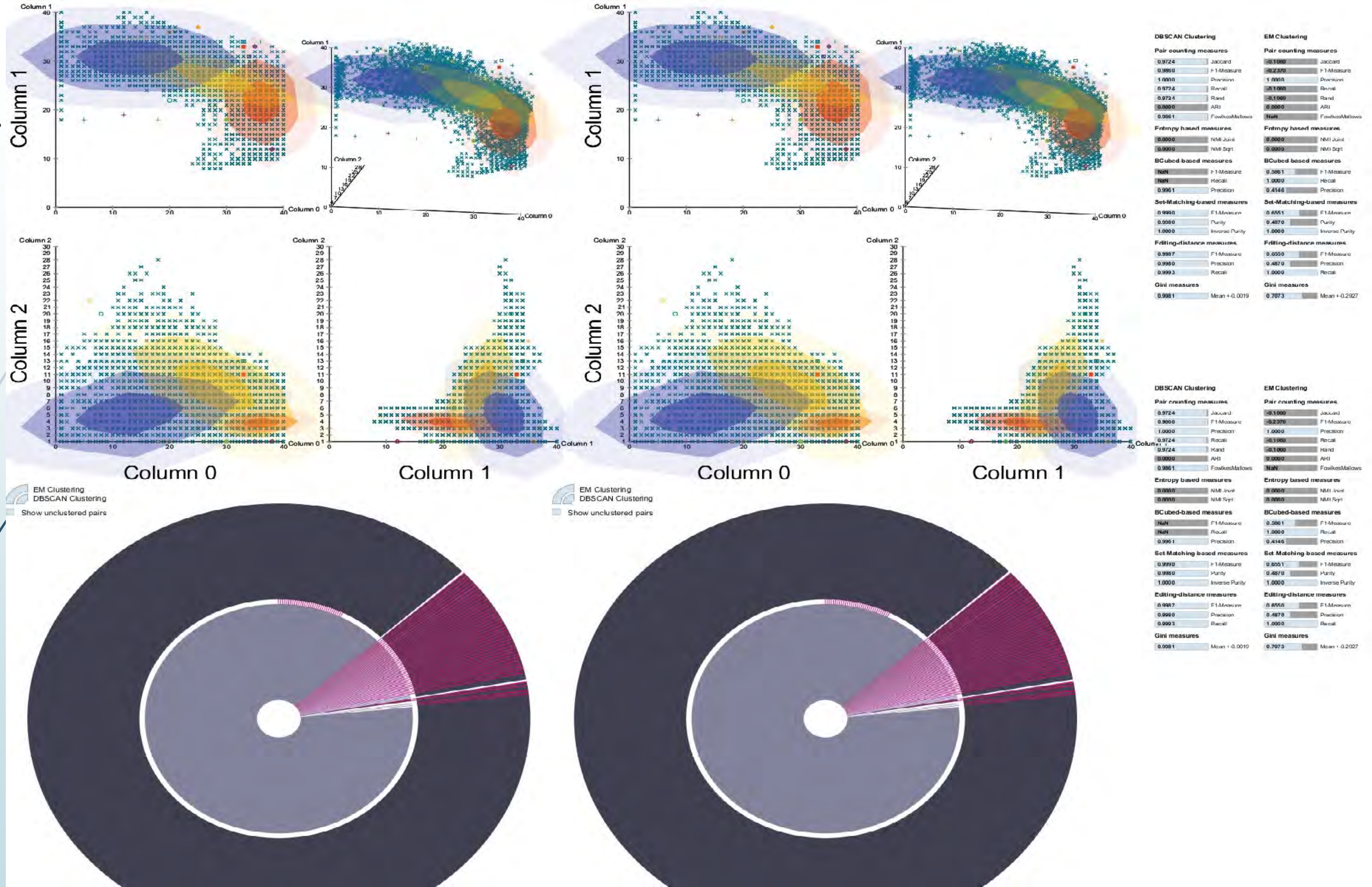








ELKI



Column 1
Column 0
Column 2
Column 1
Column 0
Column 2
Column 0
Column 1
EM Clustering
DBSCAN Clustering
Show unclustered pairs
EM Clustering
DBSCAN Clustering
Show unclustered pairs
DBSCAN Clustering
EM Clustering
Pair counting measures
Entropy based measures
BCubed based measures
Set-Matching-based measures
Editing-distance measures
Gini measures
DBSCAN Clustering
EM Clustering

Clustering

- Assemble data carefully
- Find unique partitions
- Calculate delta-r,L matrices
- Calculate 1NN, LOS, Connection
- Assign cluster #'s

Global wgtd ID →
Connected ID →
Connected wgtd →
LOS ID →
LOD wgtd ID →
Magnitude ID →

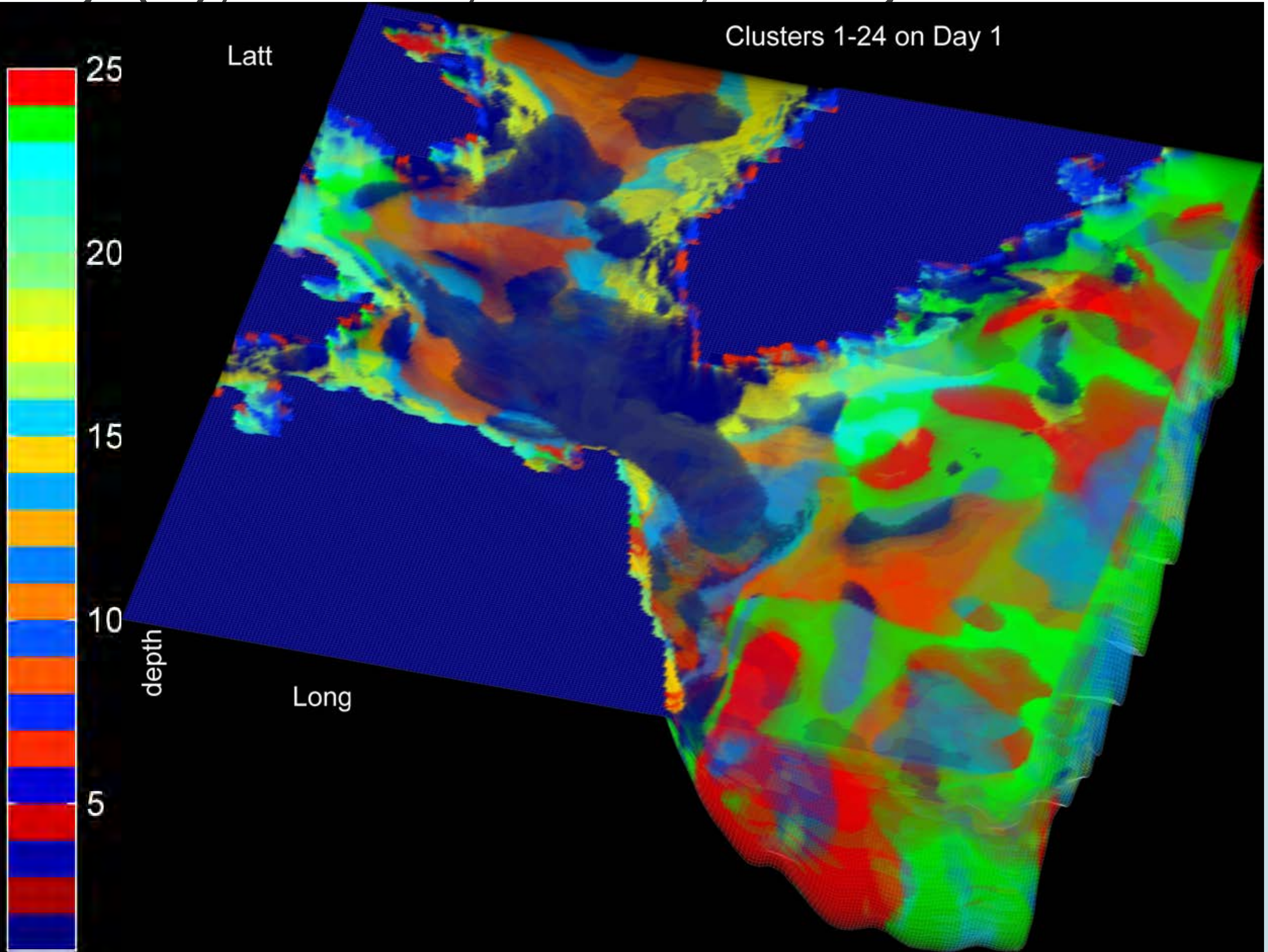
$$
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0
\end{pmatrix}
$$

Single Unique Cluster ID - # # # # # # # # # # # # #

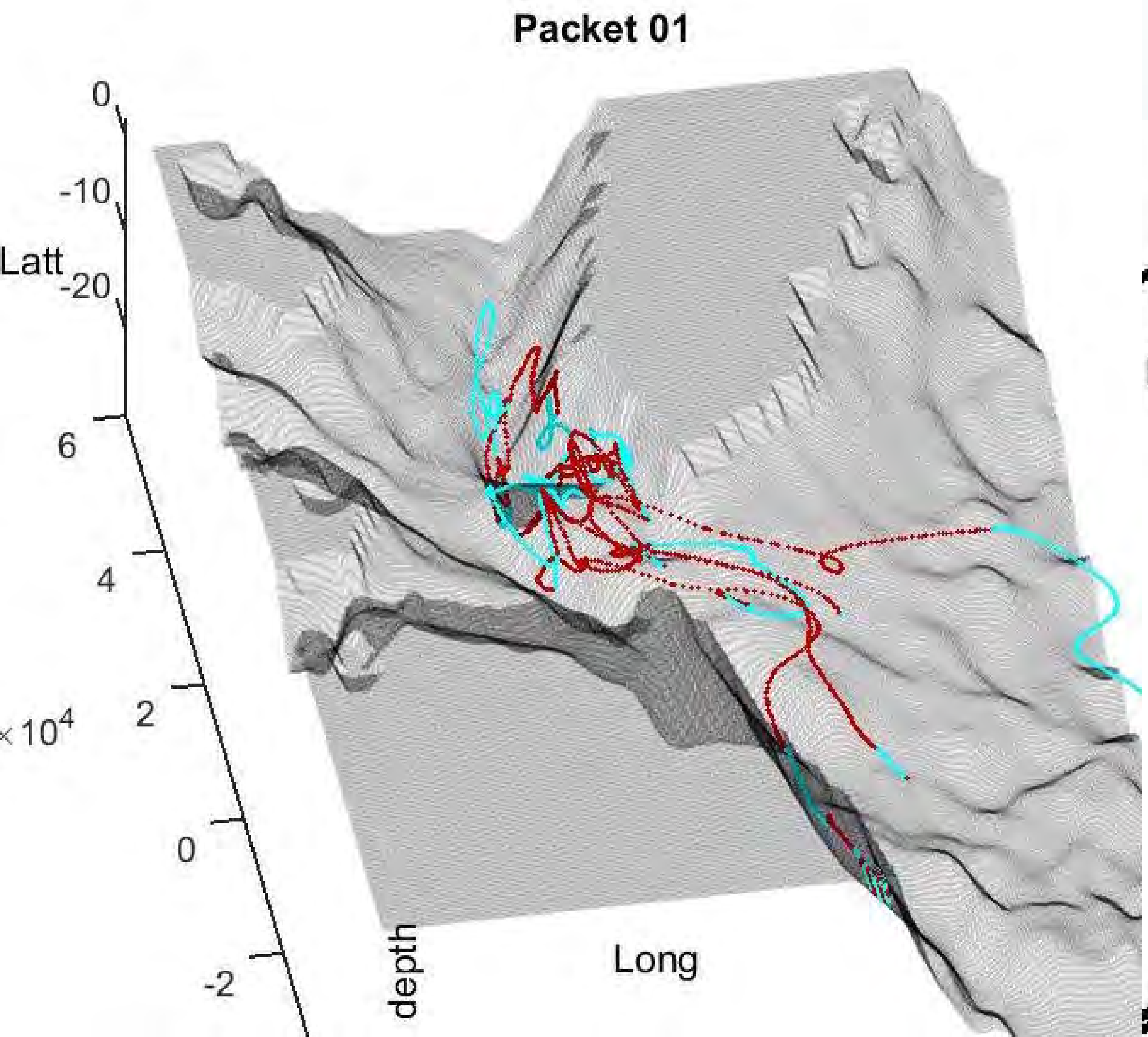
Chesapeake Bay – Magn. Simple Clustering

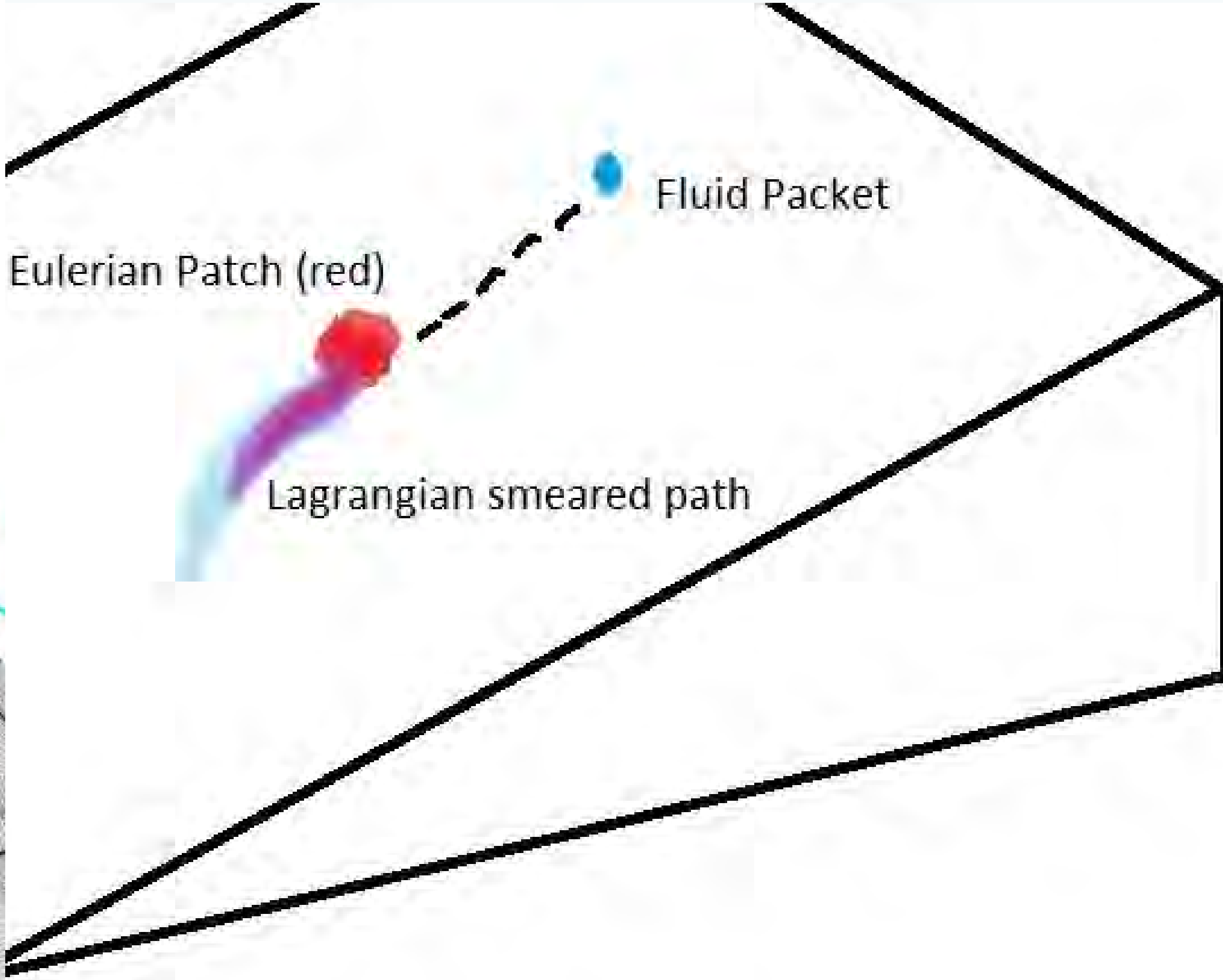
4D – transversailty (α), V-slow, RROC, V-asym

Year 2006 Population of bins	α	v_{slow}	RROC	v_{asym}
62313002	H	L	M	L
32128828	H	L	M	M
26004030	H	L	H	M
18396058	M	L	H	M
16810283	M	L	H	H
15433743	M	L	L	L
13283096	M	M	H	H
12508218	M	M	H	M
11161820	H	L	H	H
11065378	H	M	H	M
9556805	M	L	M	L
9329787	H	L	M	H
9272249	H	L	H	L
7966913	H	M	M	L
6969755	L	L	L	L
6787680	H	L	L	L
6655677	H	M	M	M
6577325	L	L	H	H
5558914	L	L	H	M
5108640	H	M	H	H
4172919	M	M	L	L
4080761	M	L	H	L
3816383	L	M	L	L
3447604	H	M	H	L



Lagrangian Trajectories with Eulerian Histories
- find Lagrangian features – seek correlations





Outline:

- Fluid analysis: Eulerian – Lagrangian
- Eulerian Measures – KE, vorticity, OW, transversality, RROC, shear, mobility
- Data Manifolds – N-dimensions
- Clustering
- Applying Clusters to Data

Future Work:

- Tracking Flow Clusters
- Tracking Particles
- Eulerian History Applied to Lagrangian Trajectories
- Eulerian – Lagrangian Correlation

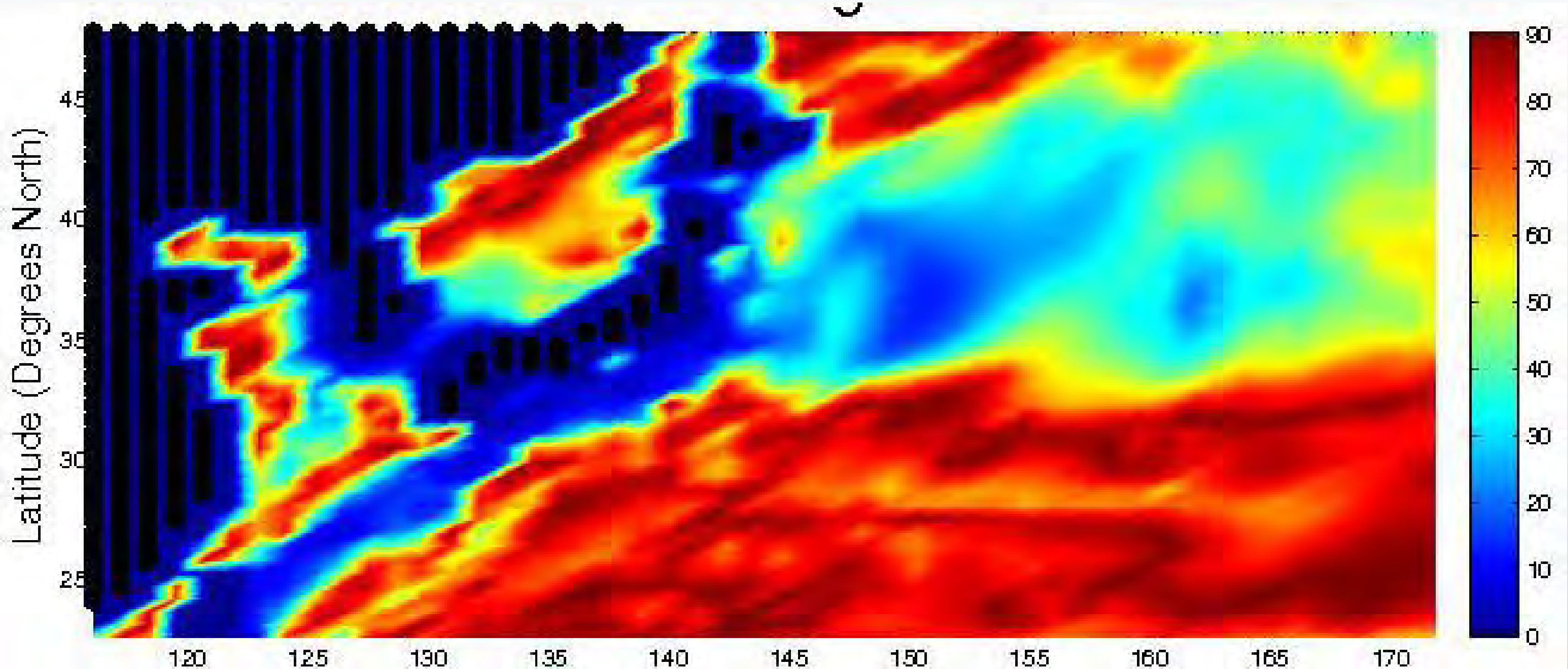
Questions?

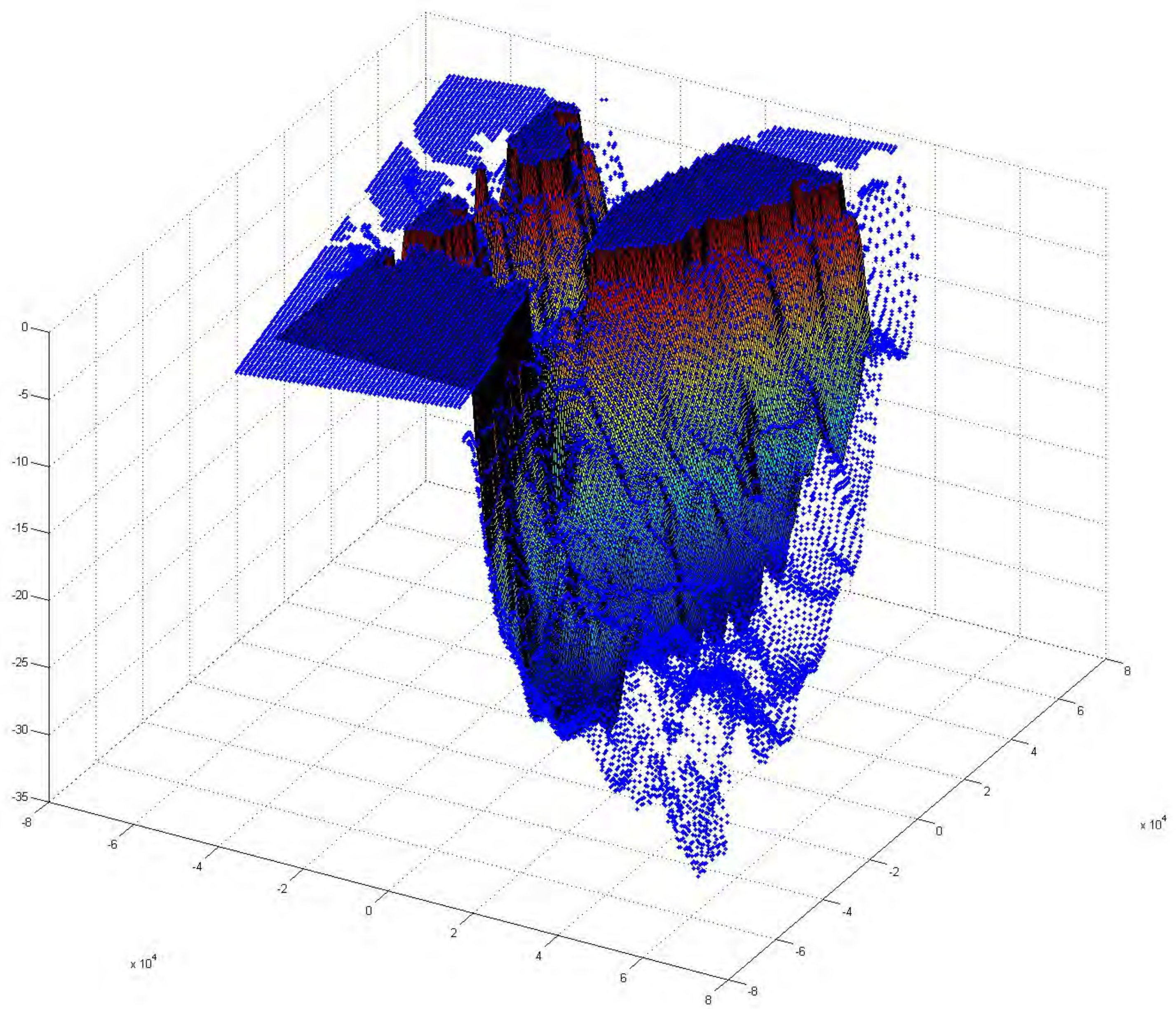
- Acknowledgements: ONR grants (multiple) – Reza Malek-Madani

SPEMS – Chesapeake Bay Mouth

- Seven Dimensions
  - KE
  - Vorticity
  - Okubo-Weiss
  - Transversality (alpha)
  - Transverse Shear (Beta)
  - Relative Rate Of Change (RROC)
  - Velocity-Asymmetry
- Took upper 70% of data to reduce computational load (for this talk only)

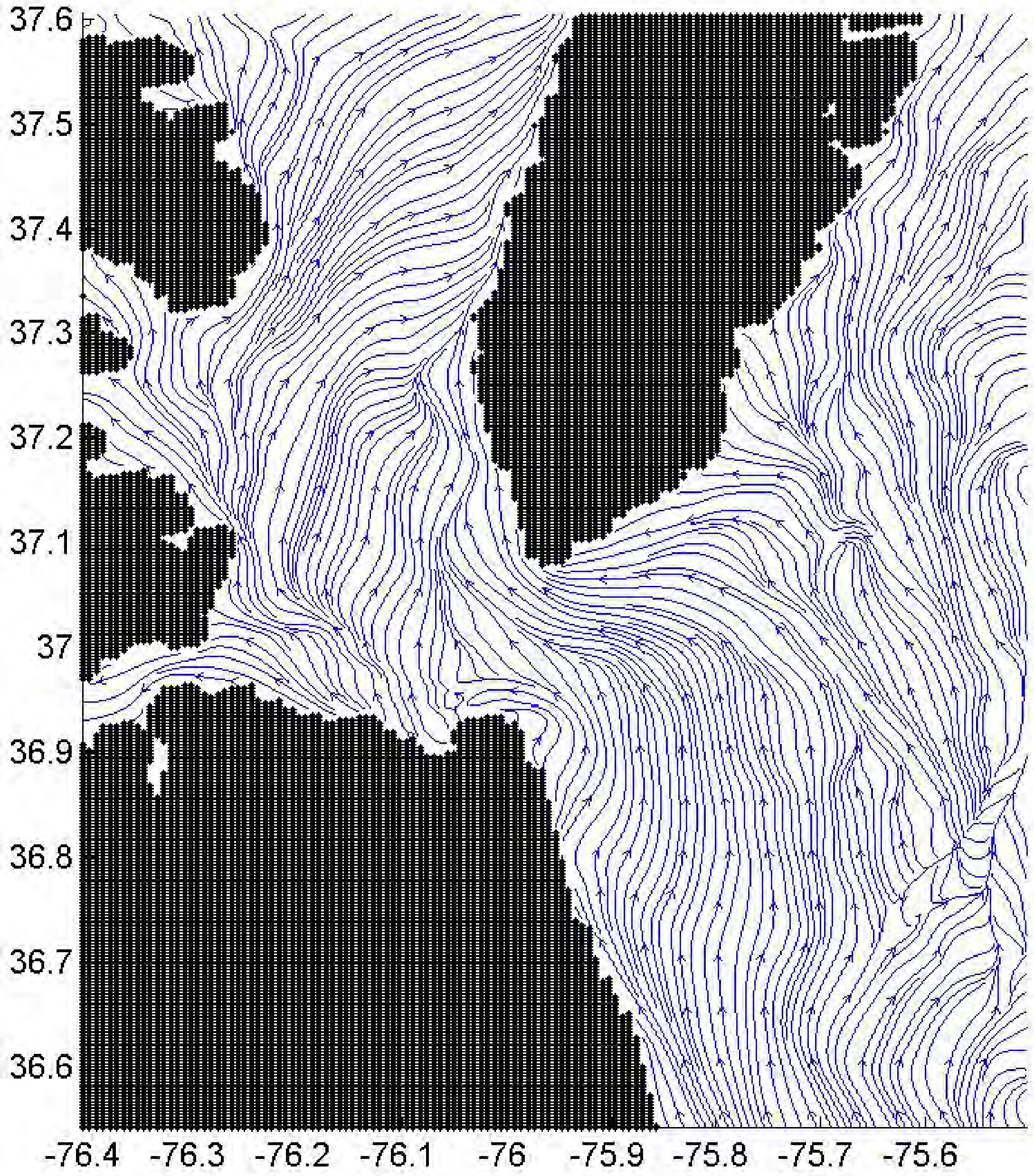
Kuroshio – Transversality





0
-5
-10
-15
-20
-25
-30
-35
-8
-6
-4
-2
0
2
4
6
8
x 10^4
8
6
4
2
0
-2
-4
-6
-8
x 10^4

Streamslice plot of day one, hour one - 10 slices



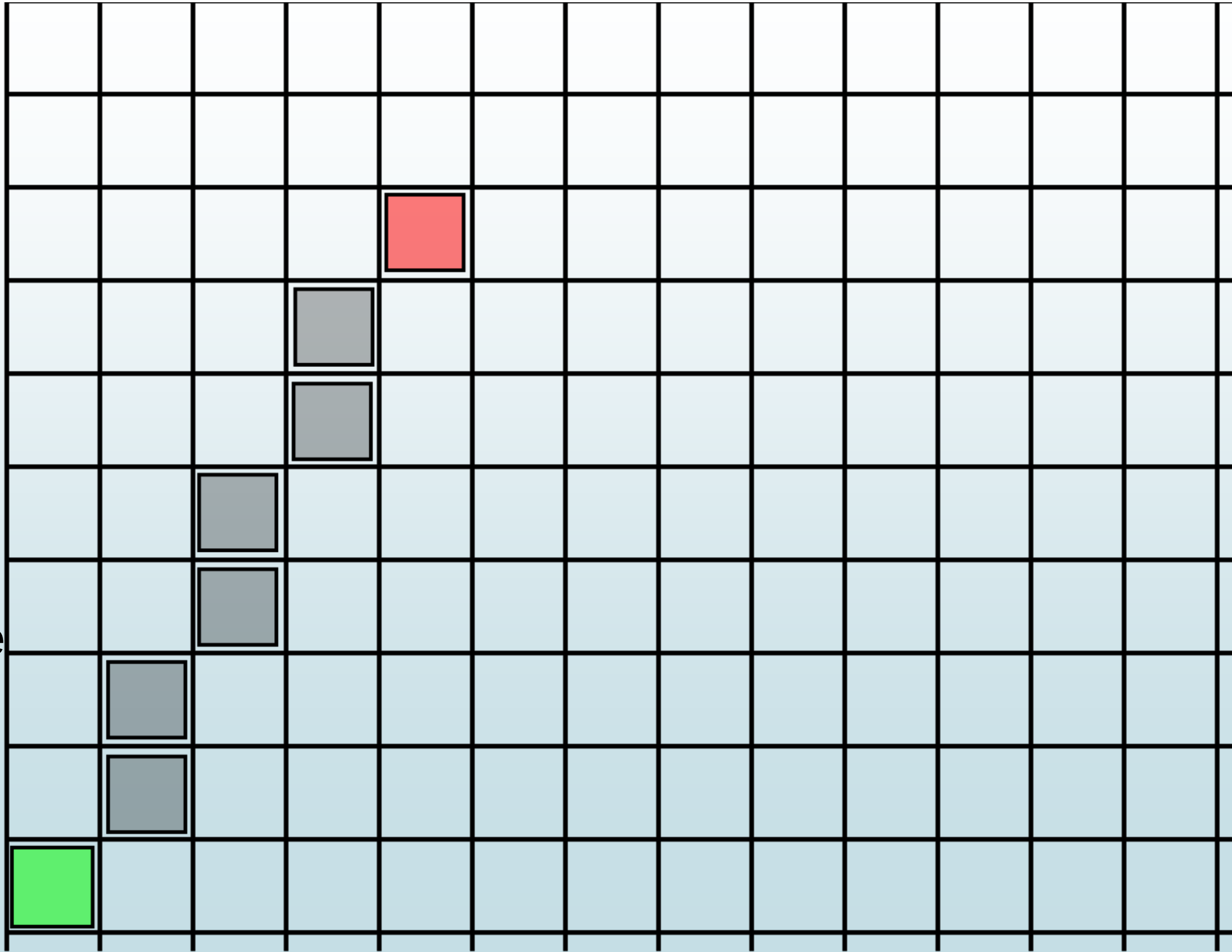
Line-Of-Sight search (LOS)

Problem with the
Line-Of-Sight approach:

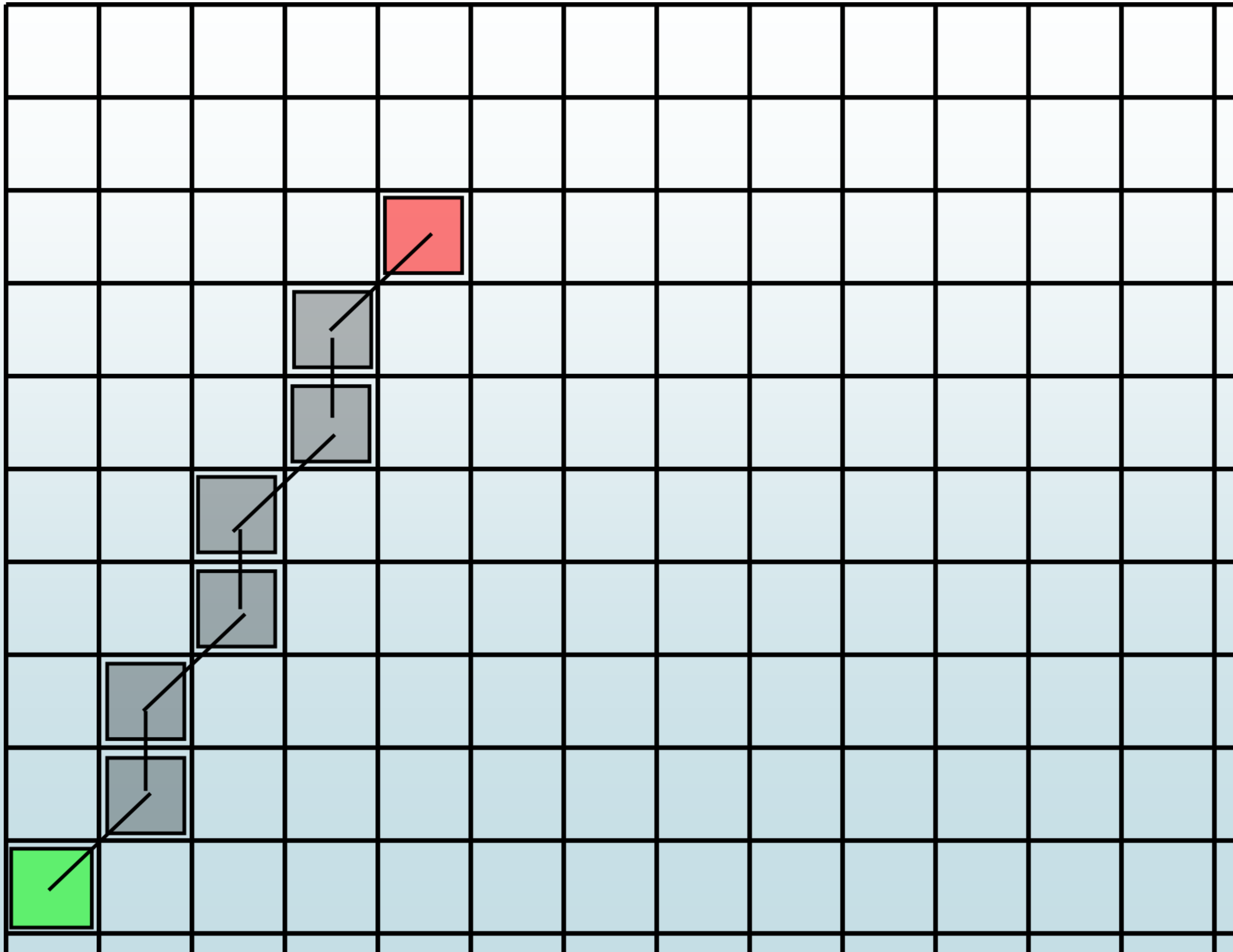
Currently, LOS is established
By demanding the path
Length be exactly the
Shortest distance from
Point A(green) to B (red).

Problem: There are multiple
Paths with the exact sqme
Length between A → B

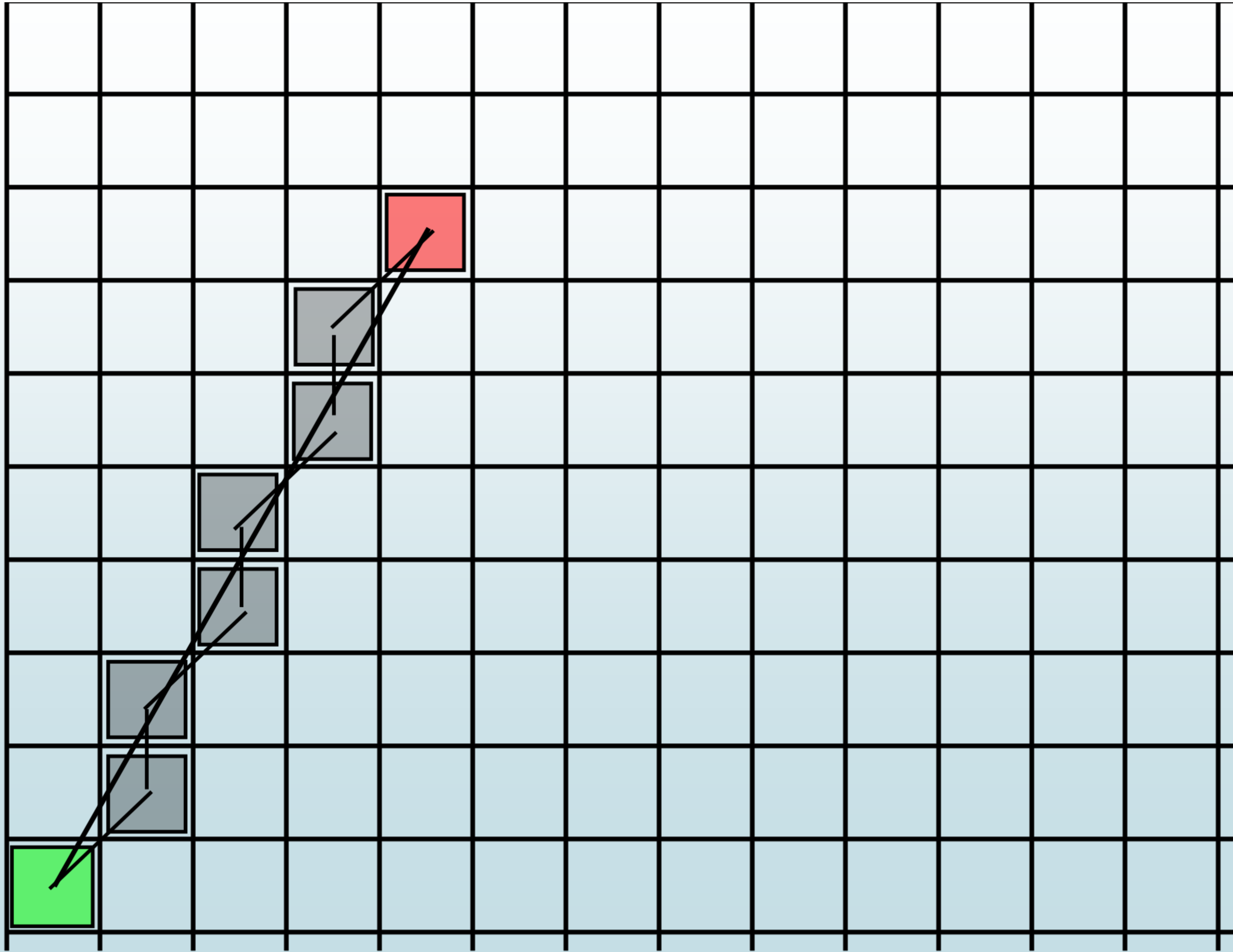
“Interior Hull” problem
(see following slides)



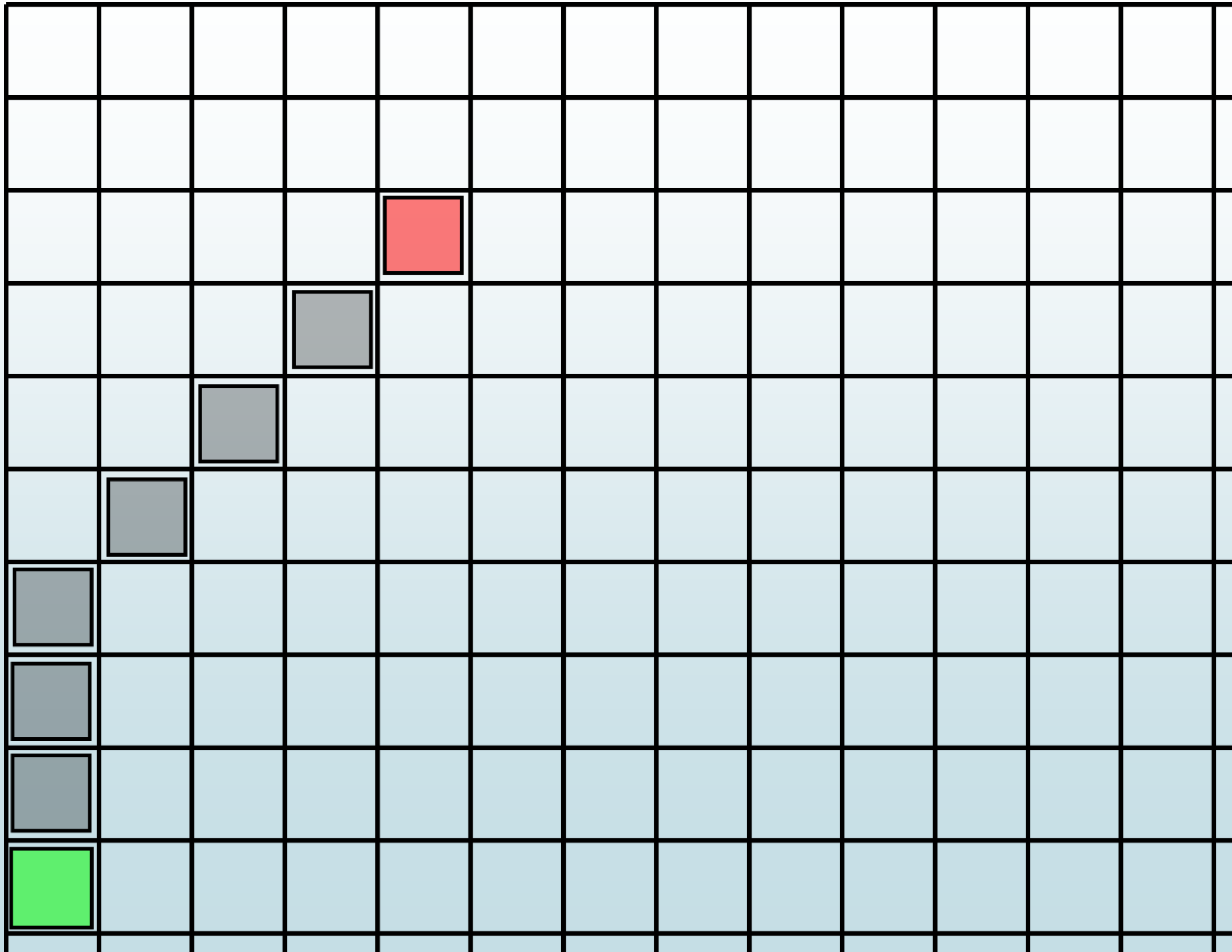
Line-Of-Sight search (LOS)



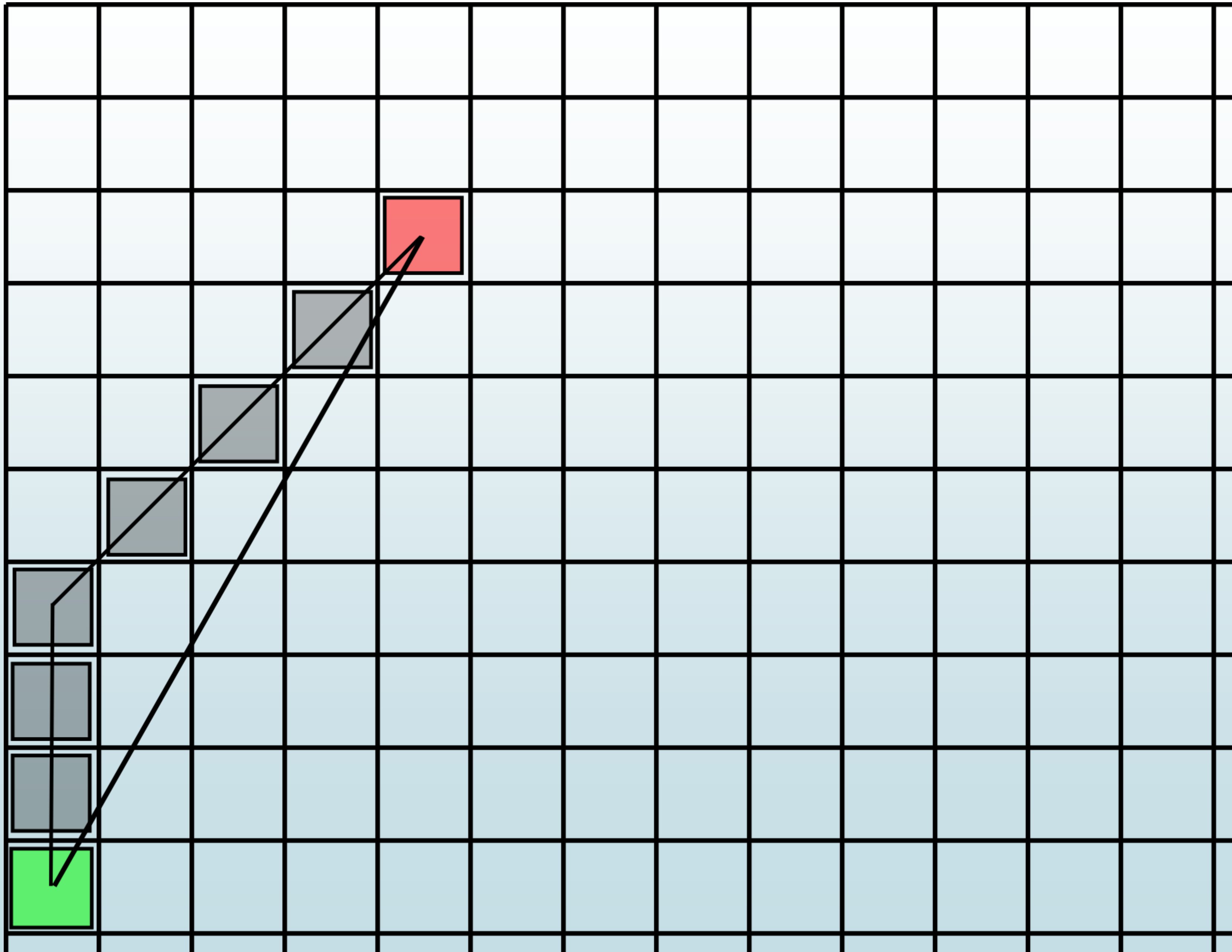
Line-Of-Sight search (LOS)



Line-Of-Sight search (LOS)

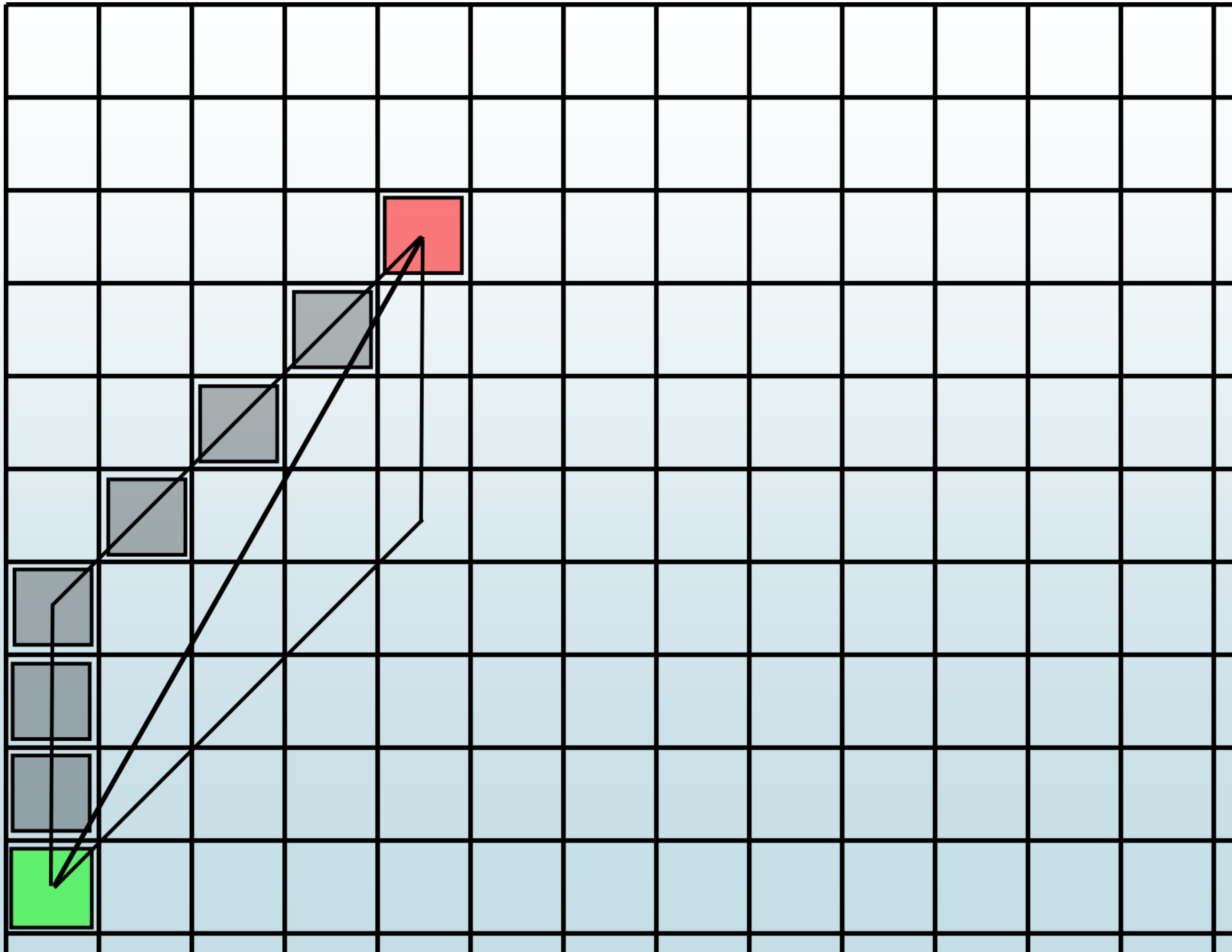


Line-Of-Sight search (LOS)



Line-Of-Sight search (LOS)

All pathes within the
Trapezoid have the same
Path length between
A → B

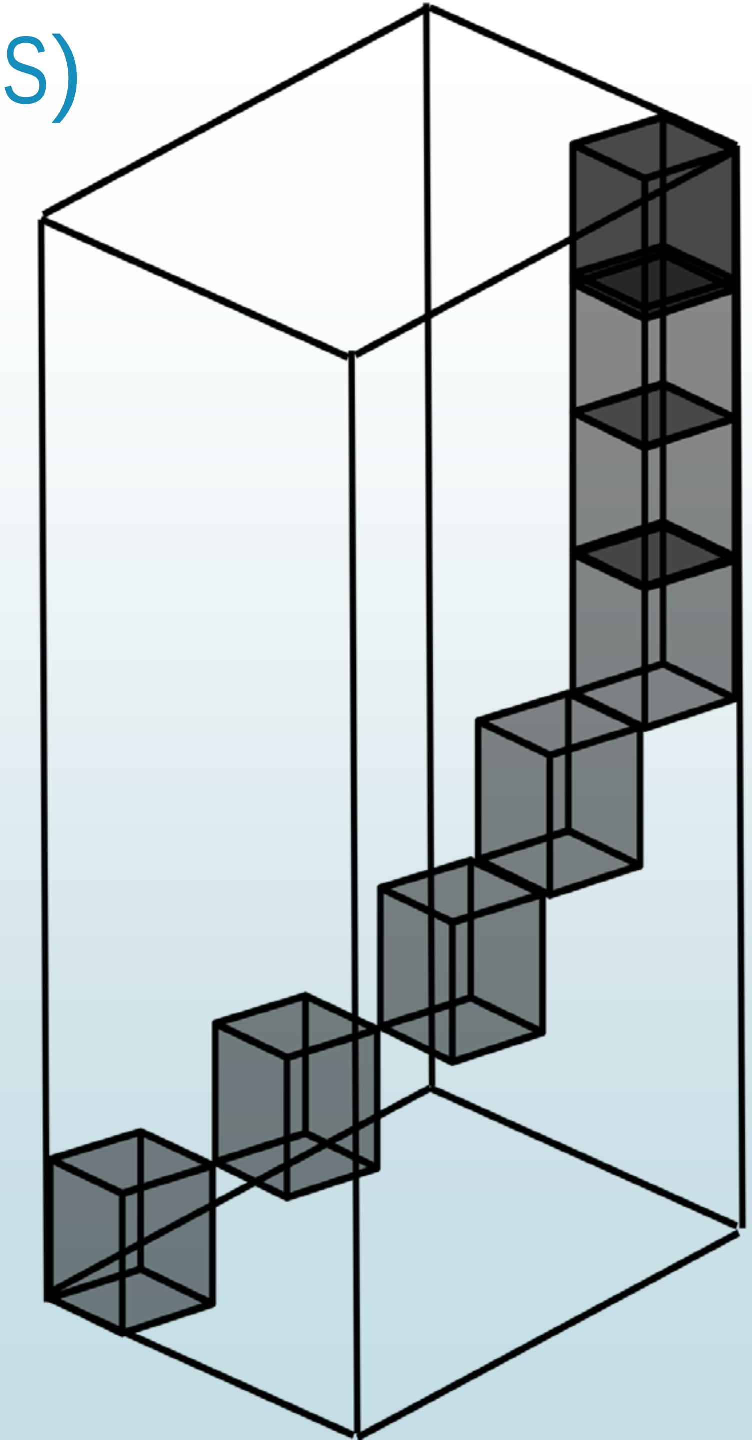


Problem: what if one of
The interior partitions is
Empty (no data present)
Which makes it
Equivalent to a “blocker”

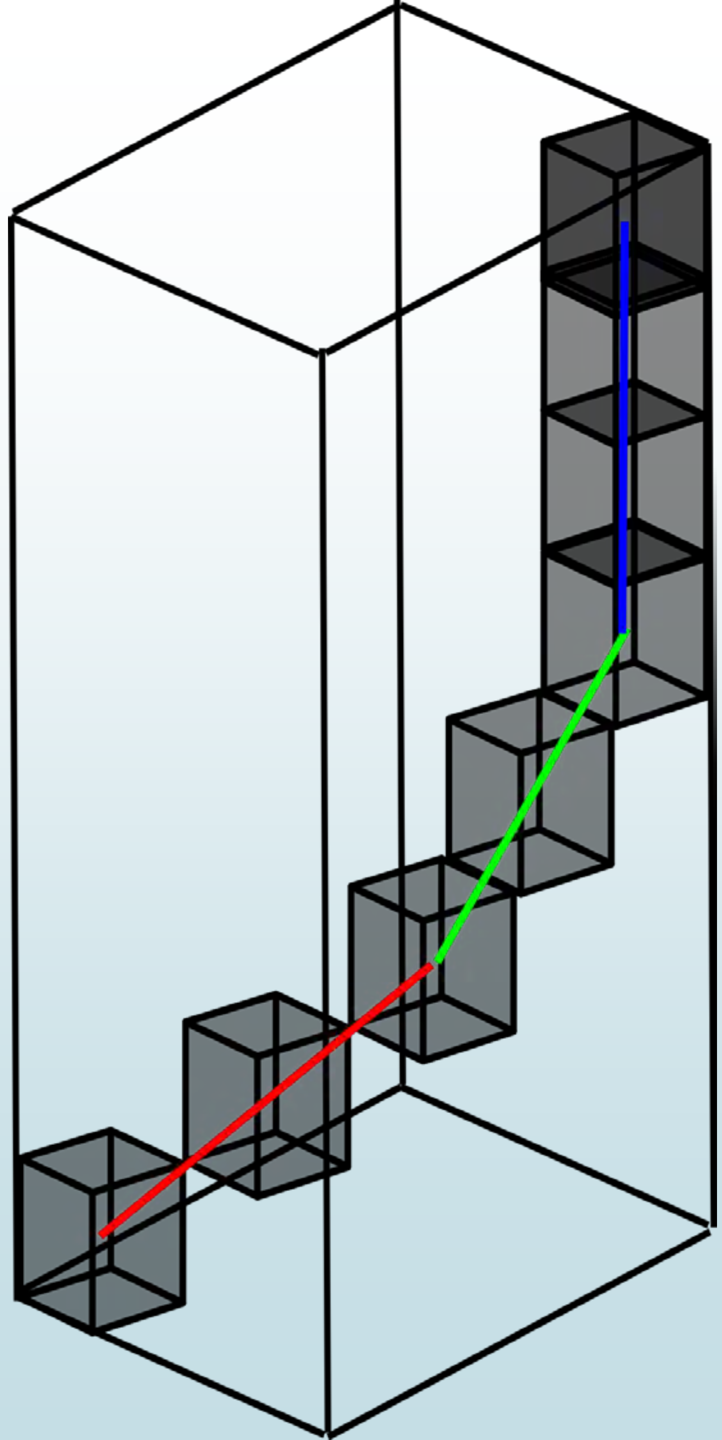
Line-Of-Sight search (LOS)

Same problem represented in 3D

The optimal path length is the same
As taking the path along the edge
Of the 3D trapezoid.
(next slides)



Line-Of-Sight search (LOS)



Line-Of-Sight search (LOS)