CSC 544
Data Visualization

Joshua Levine
josh@arizona.edu

Lecture 16
Interpolation

March 15, 2023

Today's Agenda

- Reminders:
  - A04 questions (due Mar. 27)? P02 (due Mar. 29)?
- Goals for today:
  - Discuss the effects of interpolation in visualization

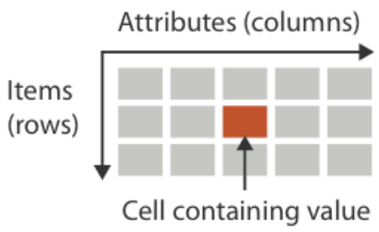
Field Data

Data and Dataset Types

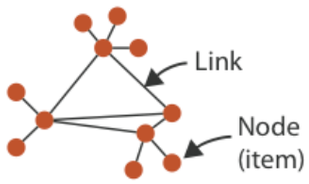
Tables	Networks & Trees	Fields	Geometry	Clusters, Sets, Lists
Items	Items (nodes)	Grids	Items	Items
Attributes	Links	Positions	Positions	
	Attributes	Attributes		

Dataset Types

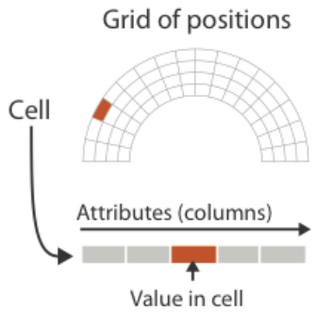
Tables



Networks

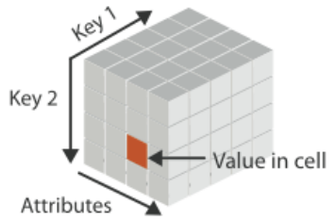


Fields (Continuous)



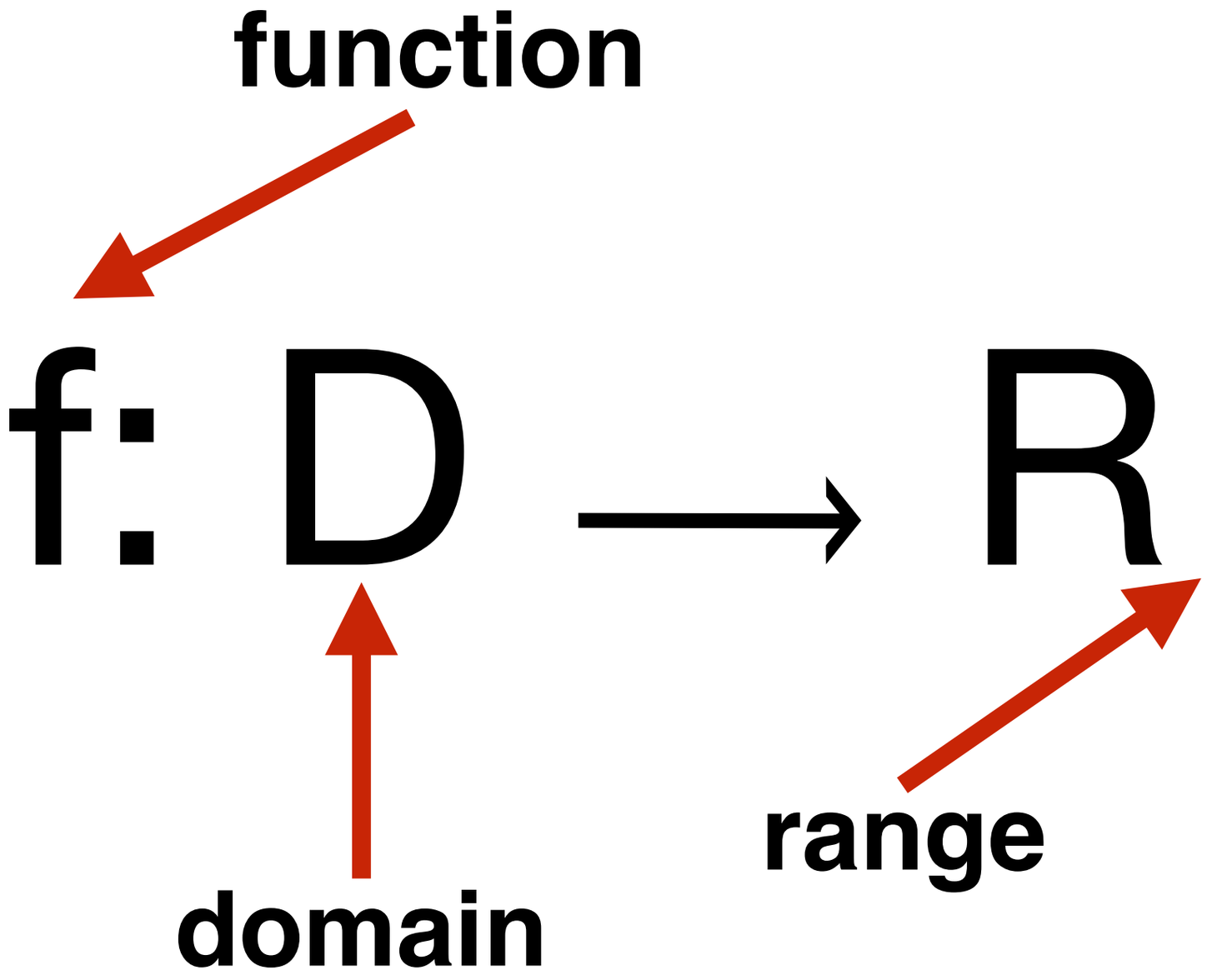
Geometry (Spatial)

Multidimensional Table



Trees

Field Data Models
Functions



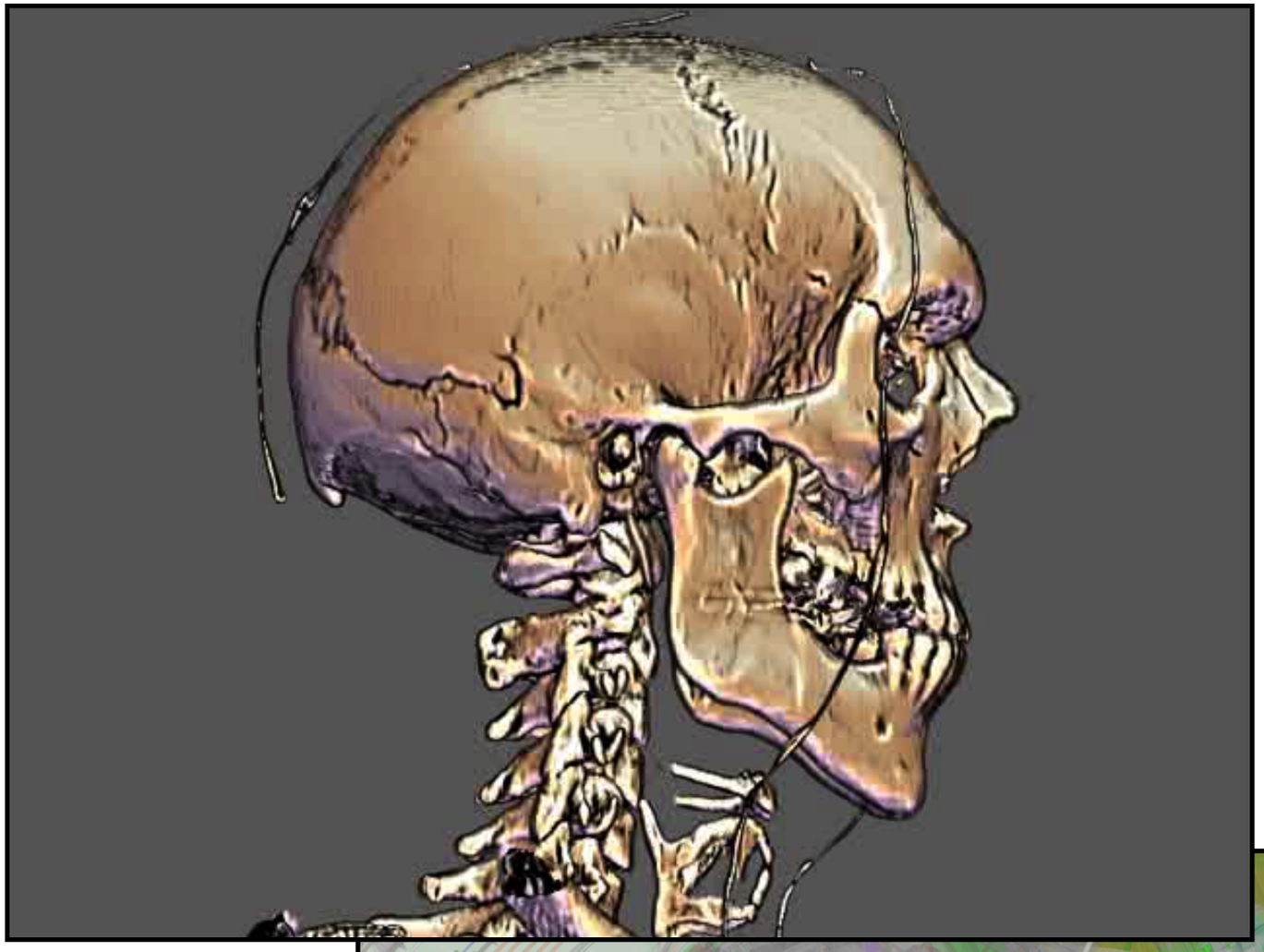
Field Data Models
Functions

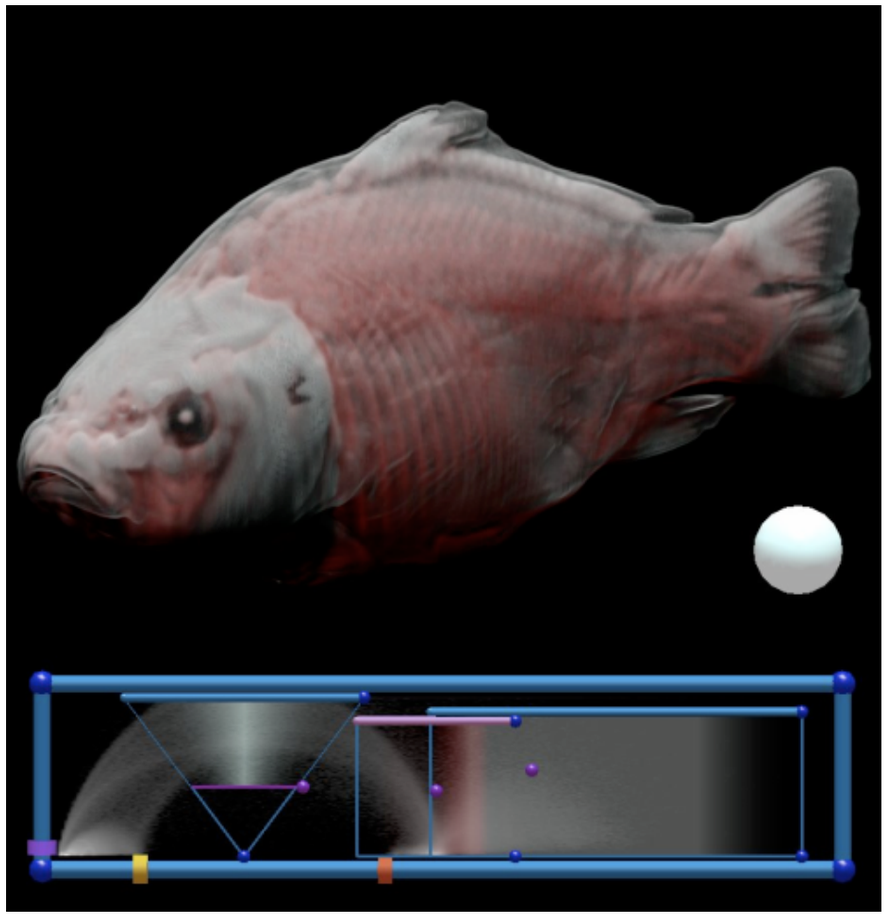
- A **function** f maps elements of some **domain** D to values in some **range** R

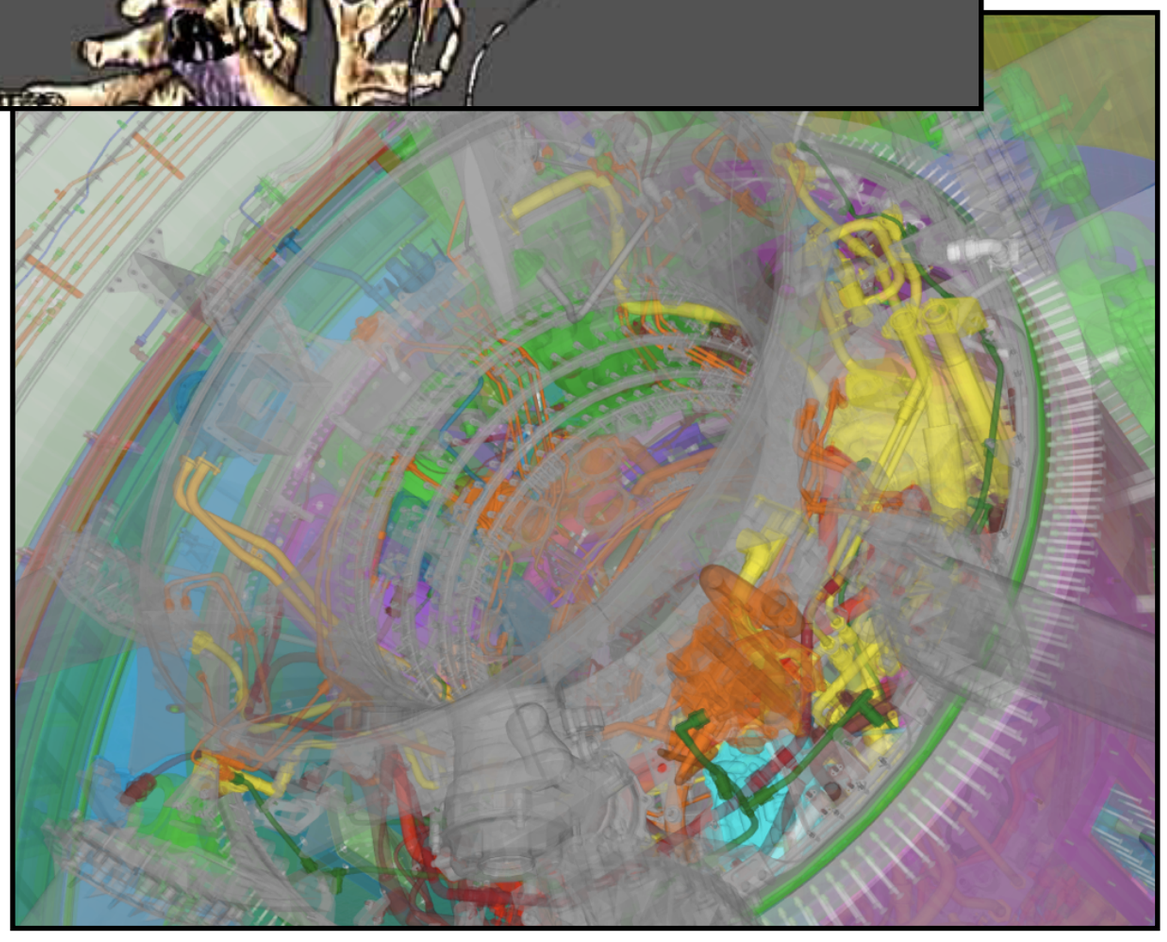
- In our setting: domains typically model some spatial context.

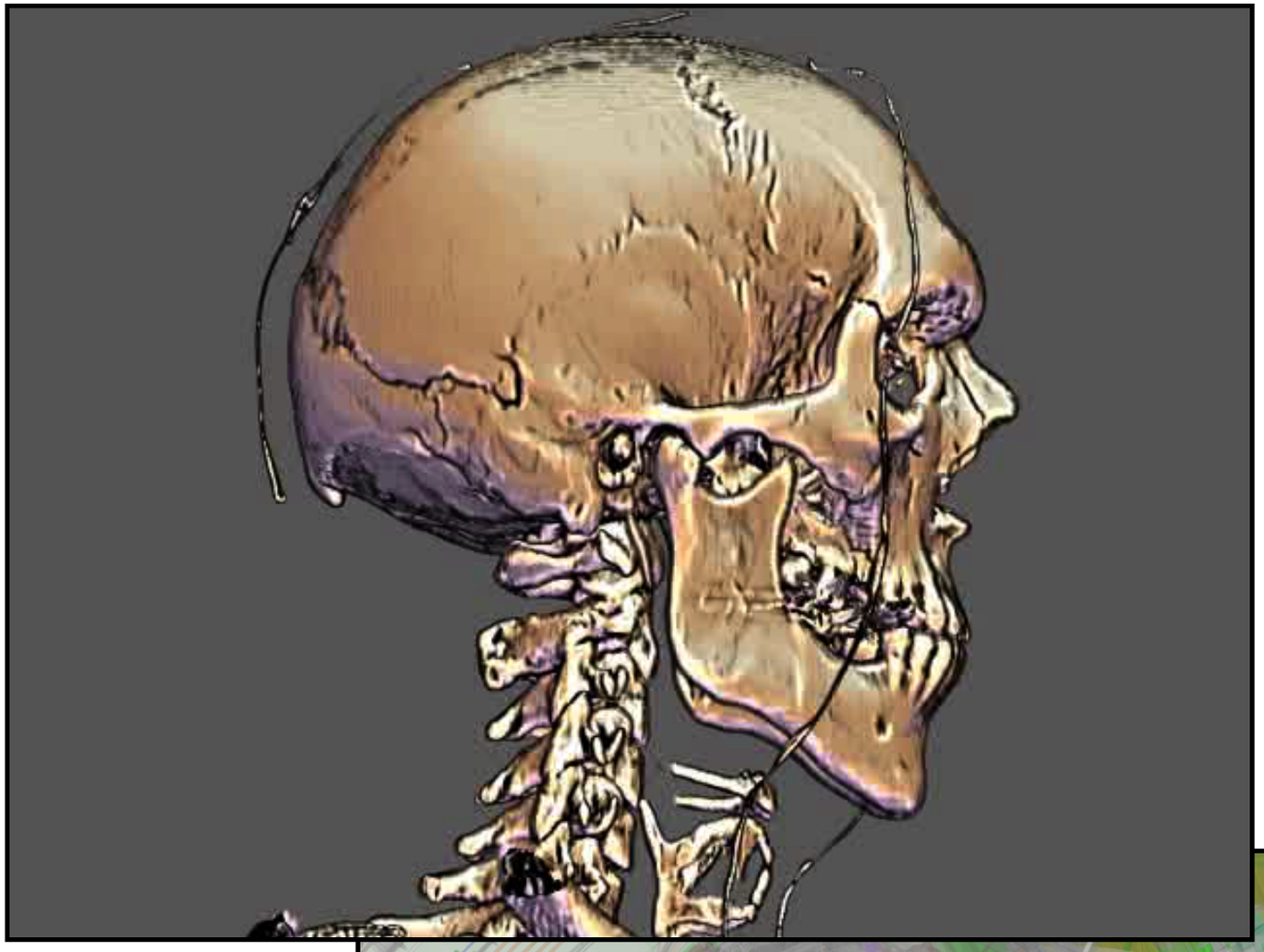
- Ranges typically model data attributes.

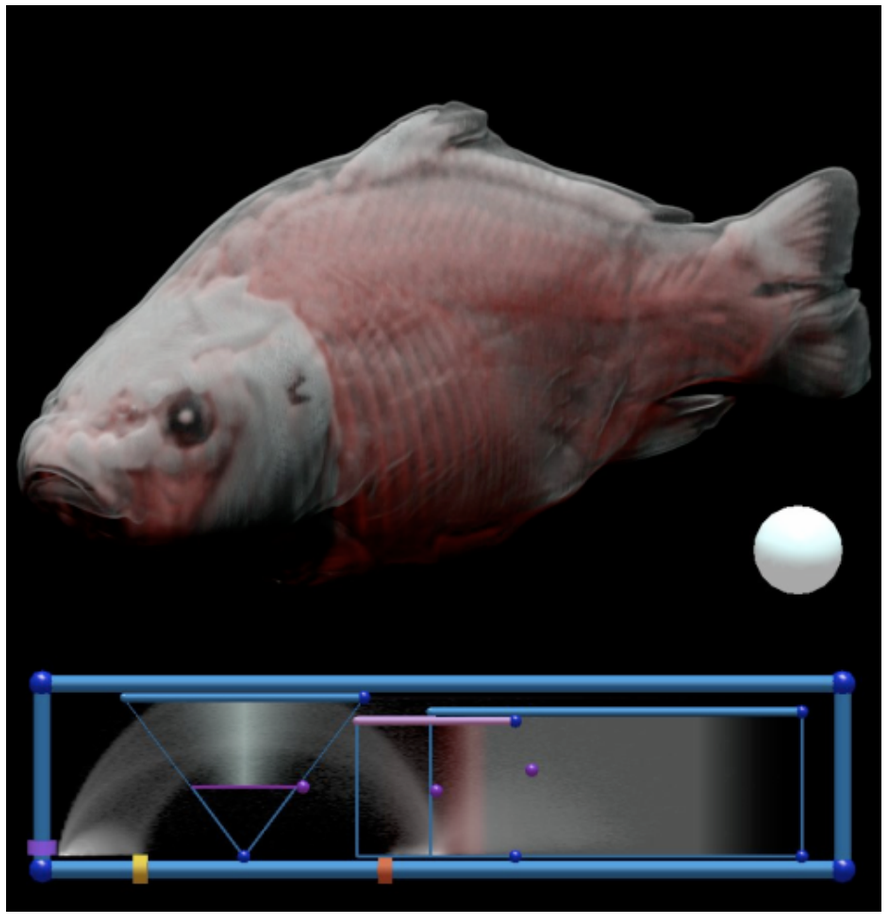
- Many types of data fit this model!

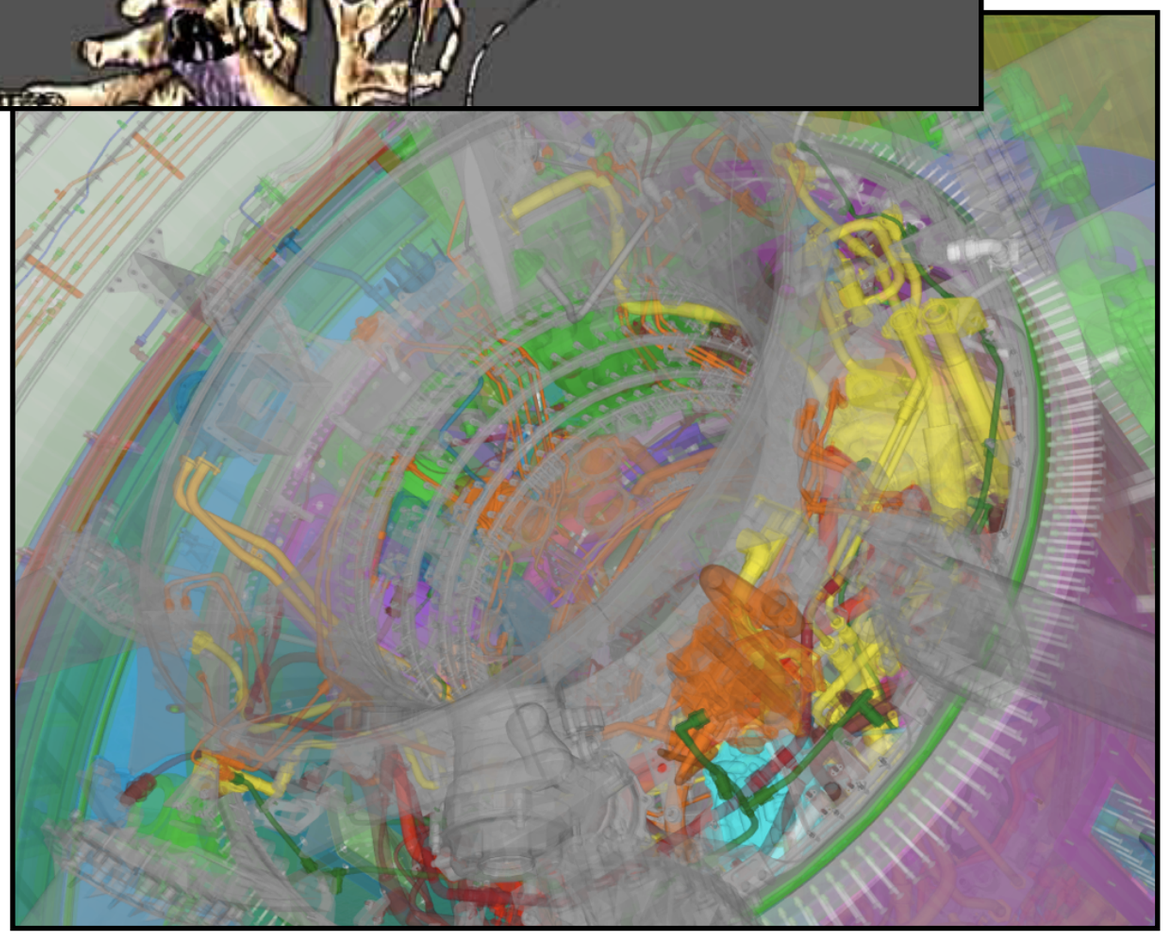










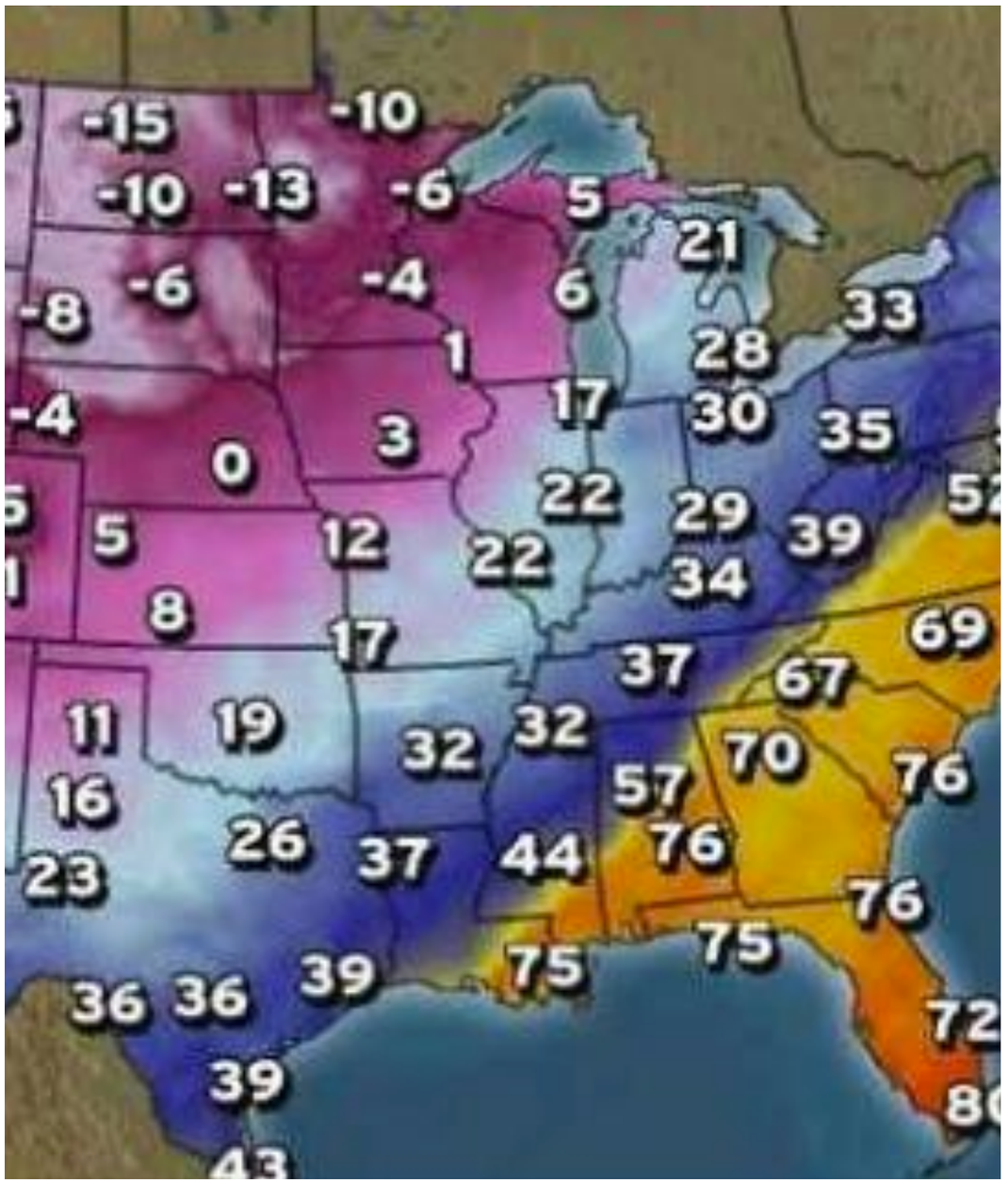


How Do We Represent Spatial Data?

- In the real world, there's infinitely many data points in a weather map.

- In a computer, we only have finite memory and finite time.

- How do we solve this problem?



Working with Domains

Working with Domains

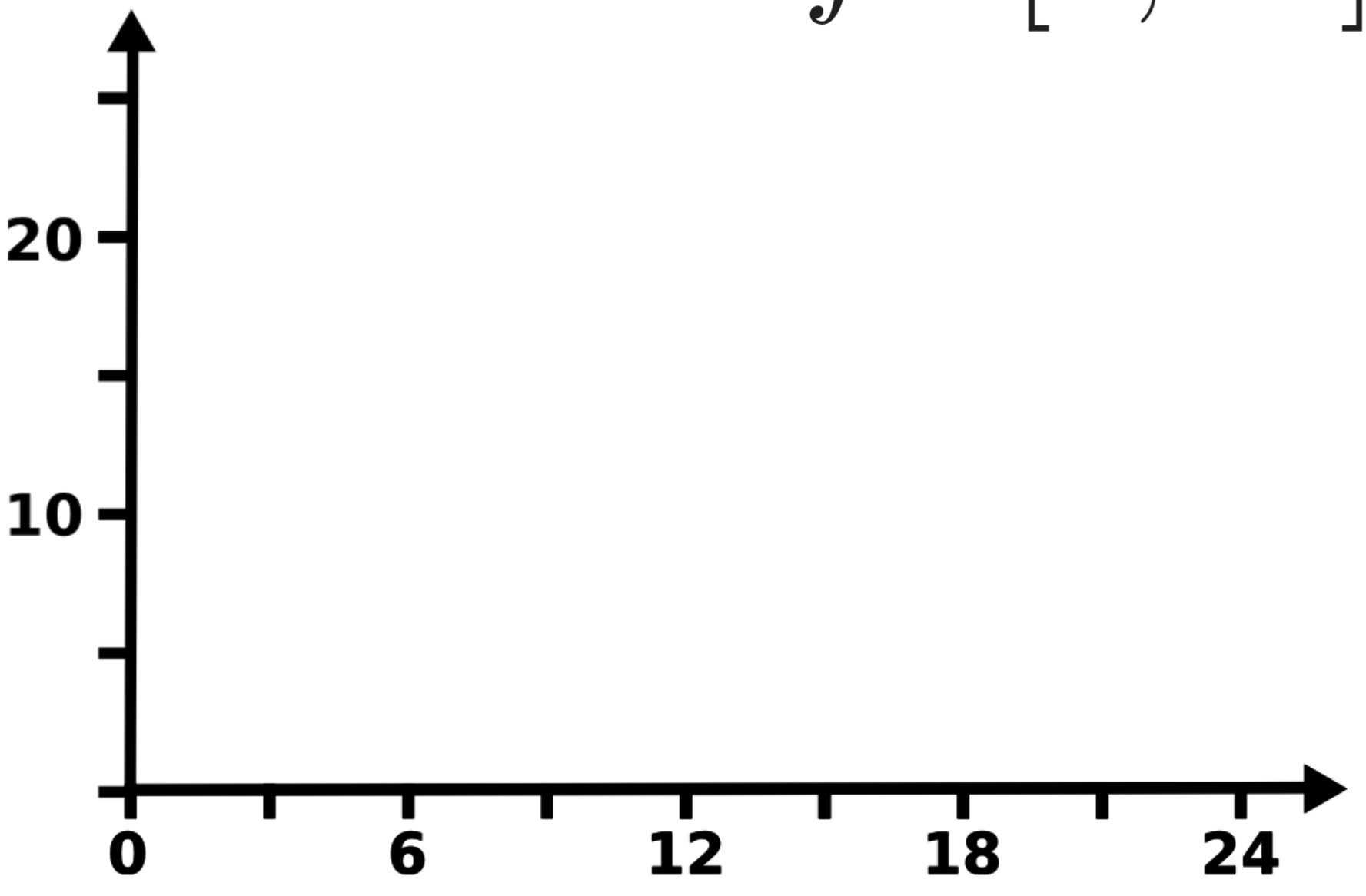
$$f : [0, 24] \to \mathbb{R}$$

Working with Domains

$$f : [0, 24] \to \mathbb{R}$$

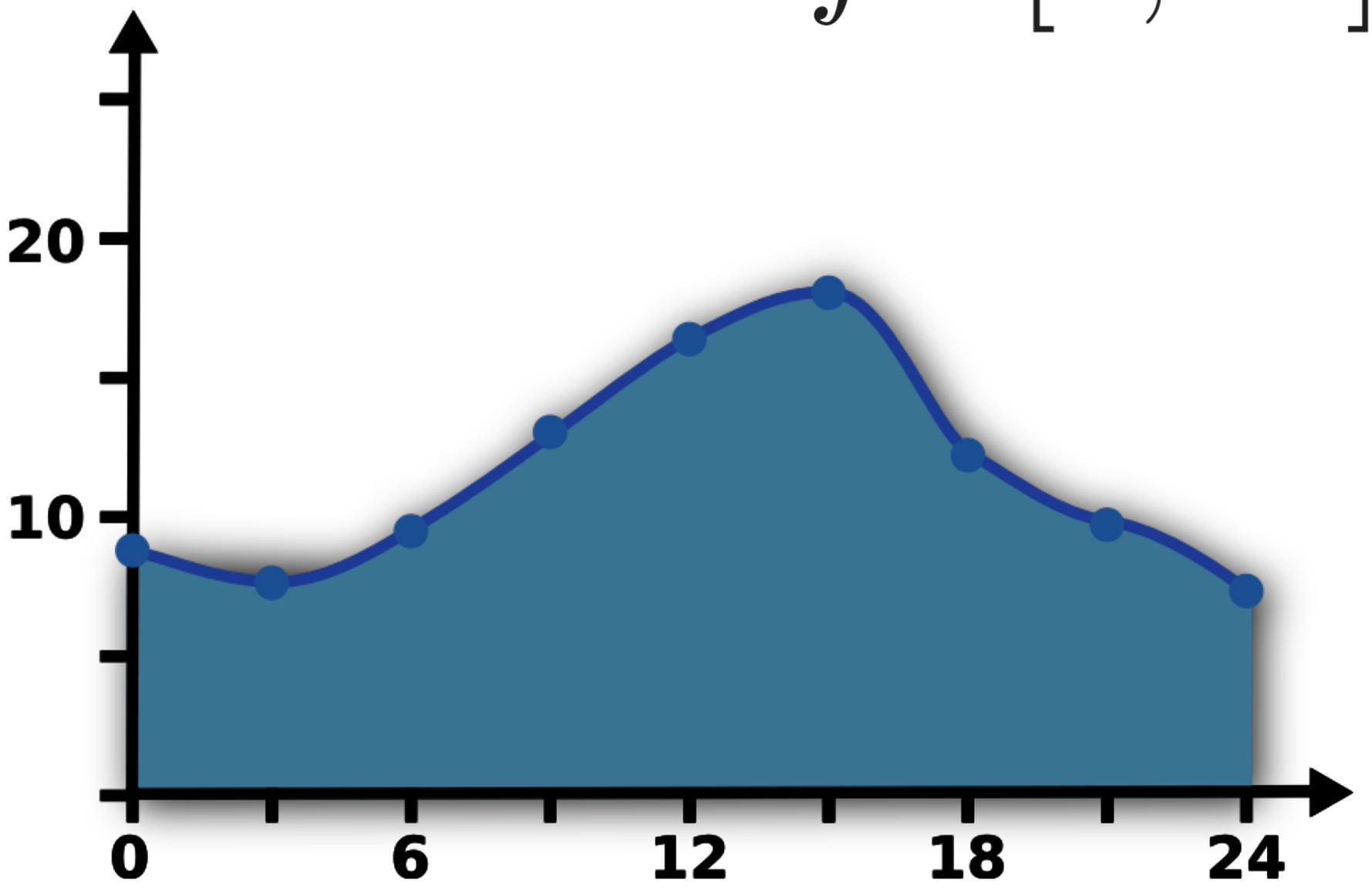
Working with Domains

$$f : [0, 24] \to \mathbb{R}$$



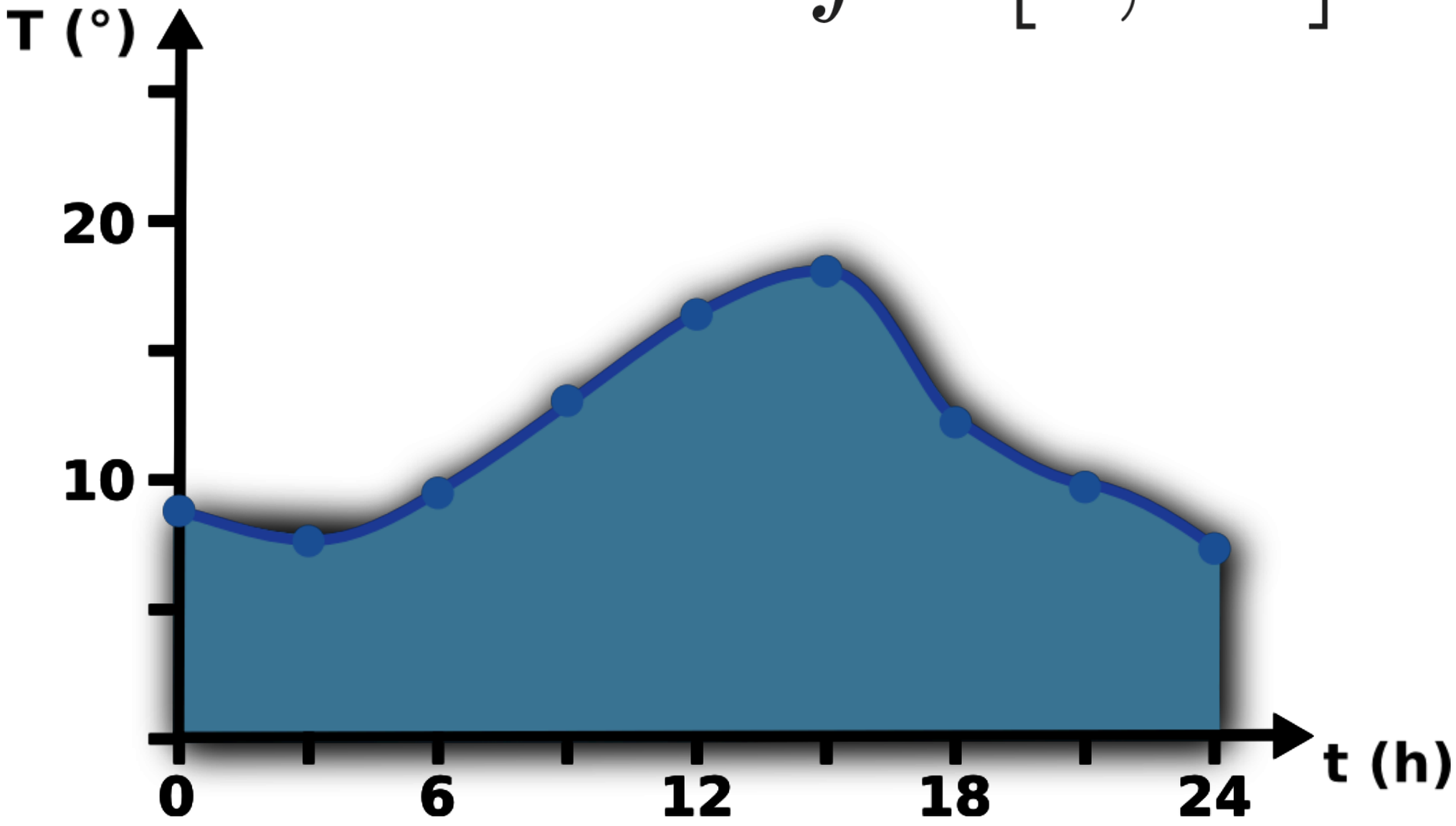
Working with Domains

$$f : [0, 24] \to \mathbb{R}$$



Working with Domains

$$f : [0, 24] \to \mathbb{R}$$

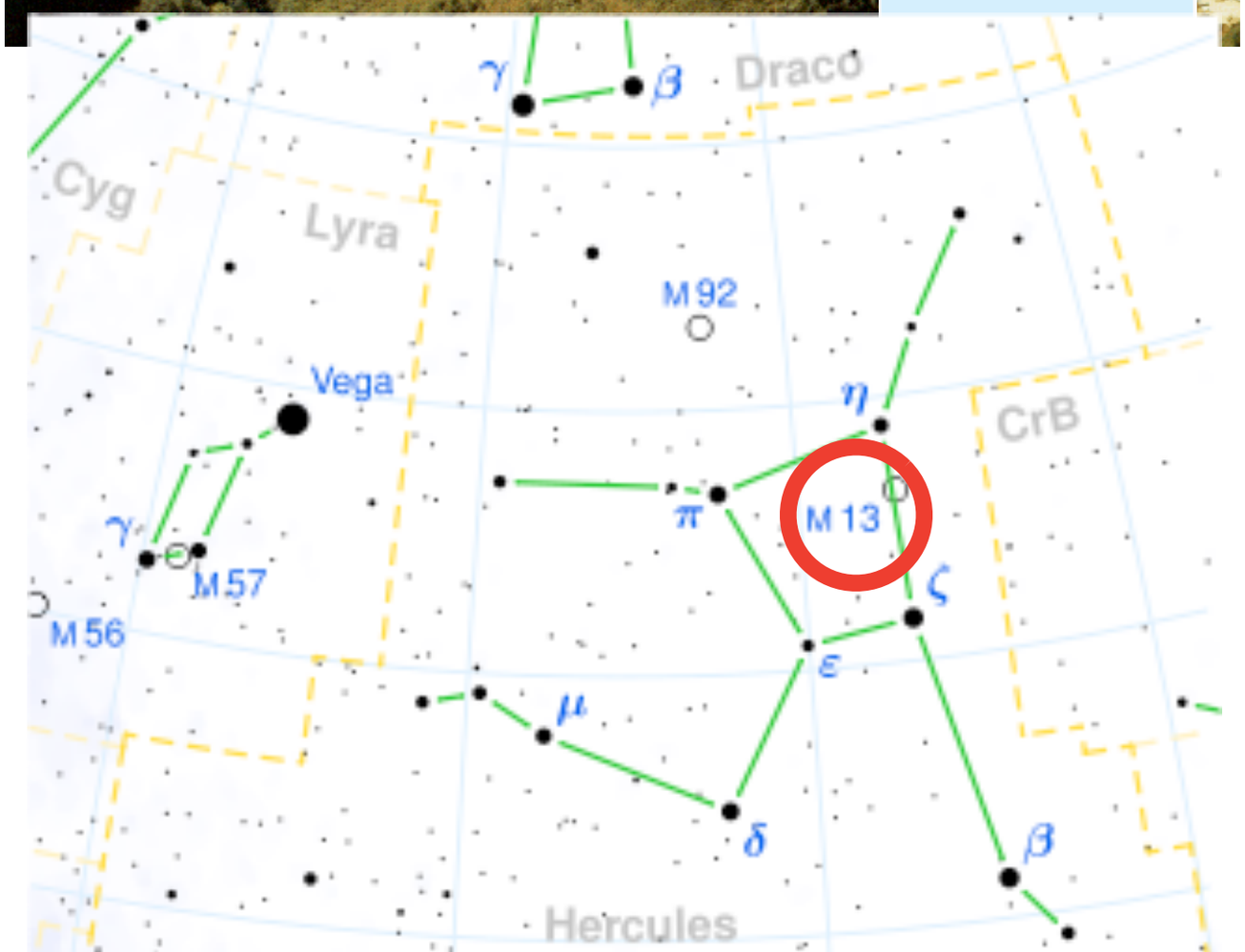


Spatial Domains as a Design Choice

Arecibo Message

http://en.wikipedia.org/wiki/Arecibo_message

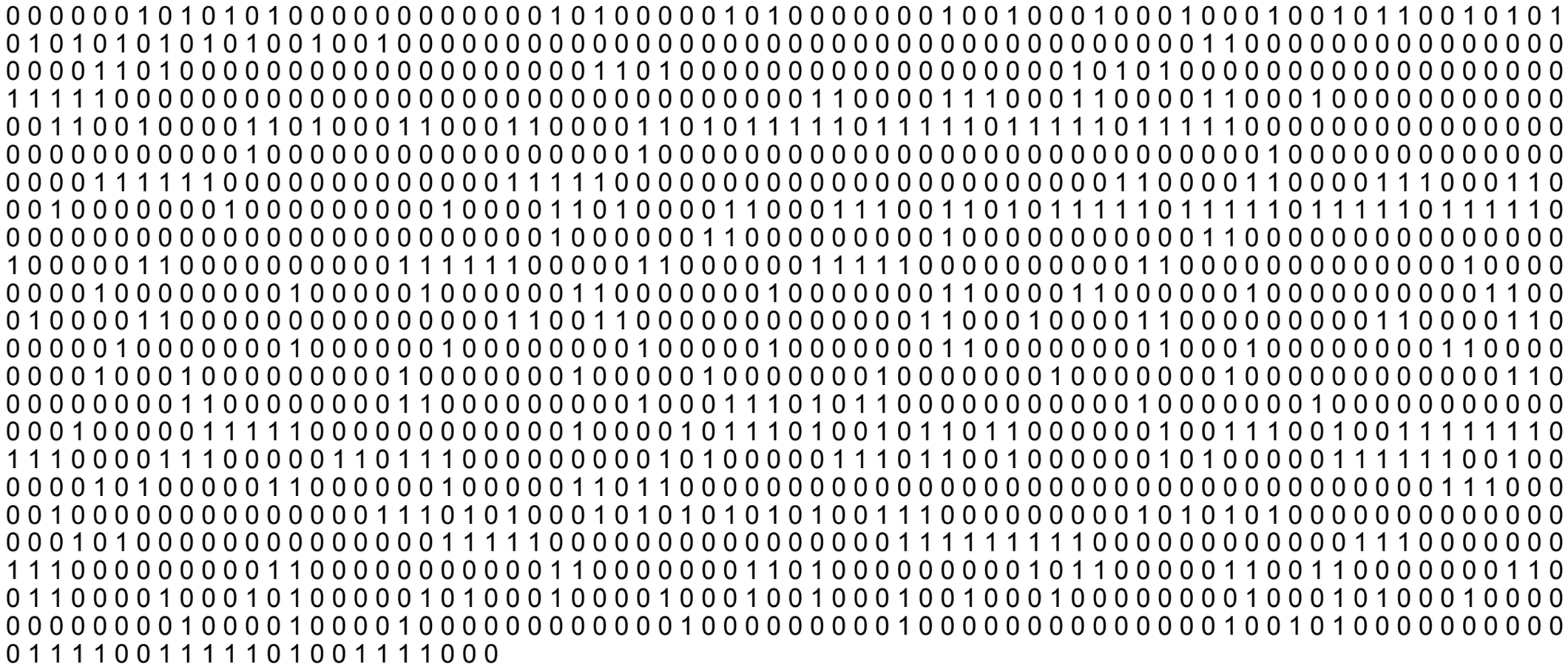
- Way of understanding mechanics of raster image representation
- Radio telecope in Puerto Rico
- built in 1964, renovated in 1974, **decommissioned in 2020**
- To celebrate: Frank Drake and Carl Sagan (Cornell University) sent message to M13 in Hercules (25,000 light years away)
- 1679 bits, frequency modulate 2380 MHz



The Message

http://www.physics.utah.edu/~cassiday/p1080/lec06.html

1679 bits were encoded as 2380MHz plus and minus some frequency



This is a **1-D** sequence of bits in time

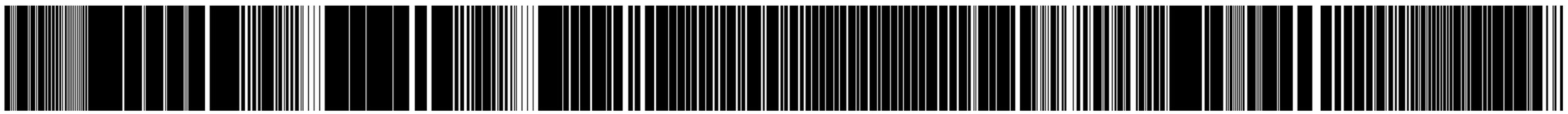
How will an alien understand this list of bits?

(will have different symbols than “0” “1”)

No meta-information!

Understanding the message

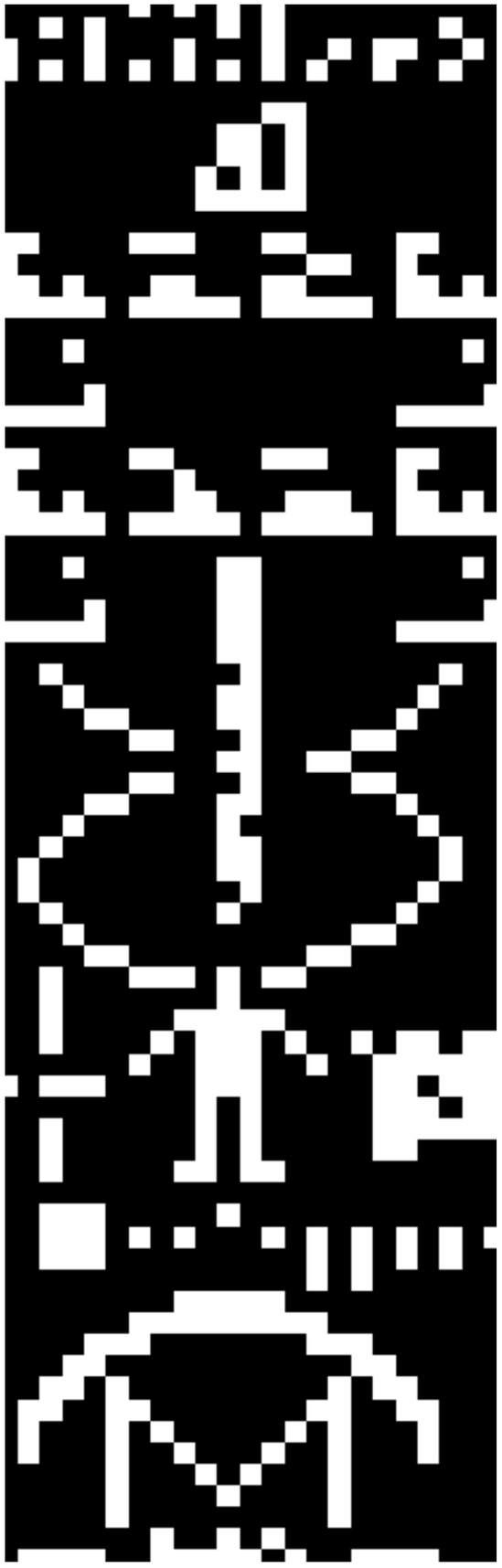
- Perhaps some “visual” representation of bits



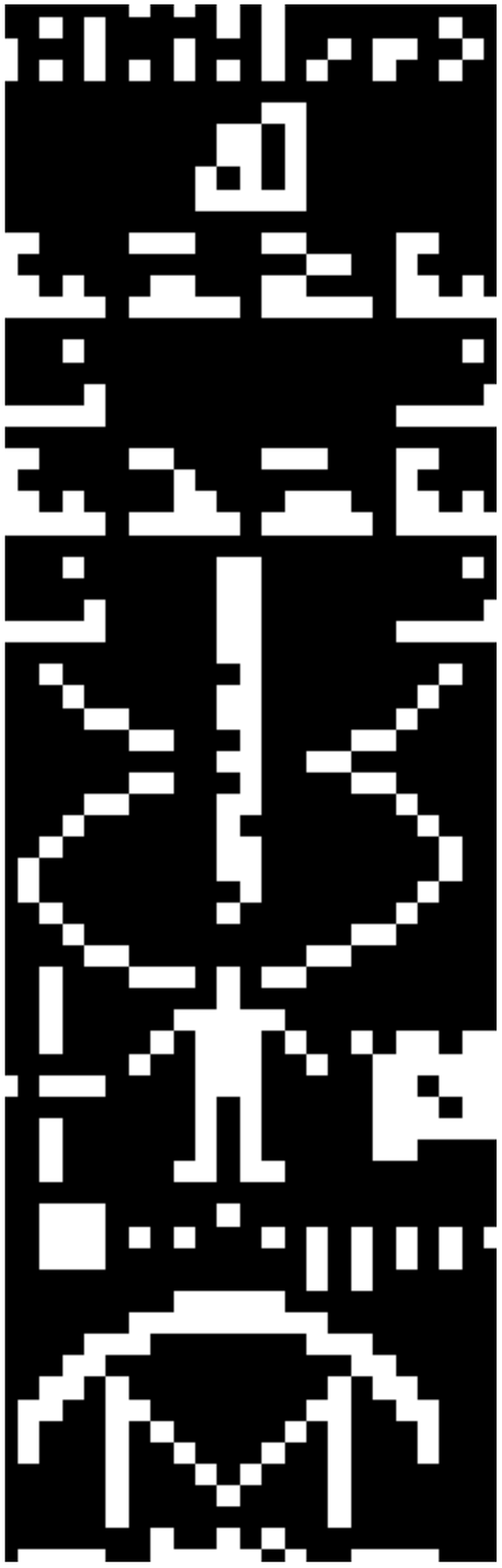
  - (what is black vs white?)
- Aliens notice 1679 = 23 x 73 (product of two primes)
- Perhaps its not a linear sequence: 2-D array
  - (try that)
- Two ways of sequencing values in 2D array
  - (try that)
- Various ways of laying them out in 2D space
  - (try that)
- Then: have to decipher it!

73 x 23

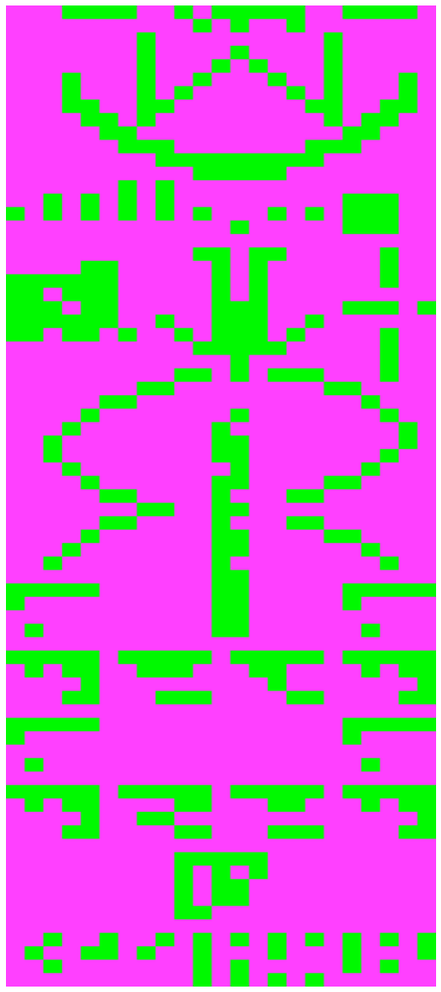
73 x 23



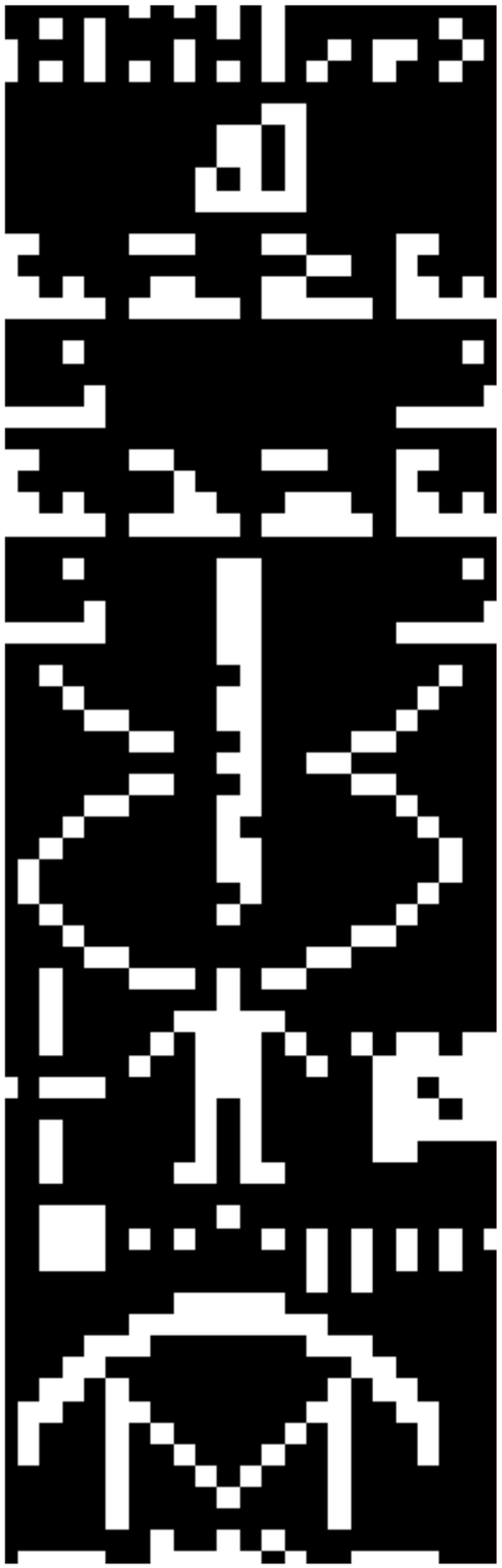
23 x 73: what was different?



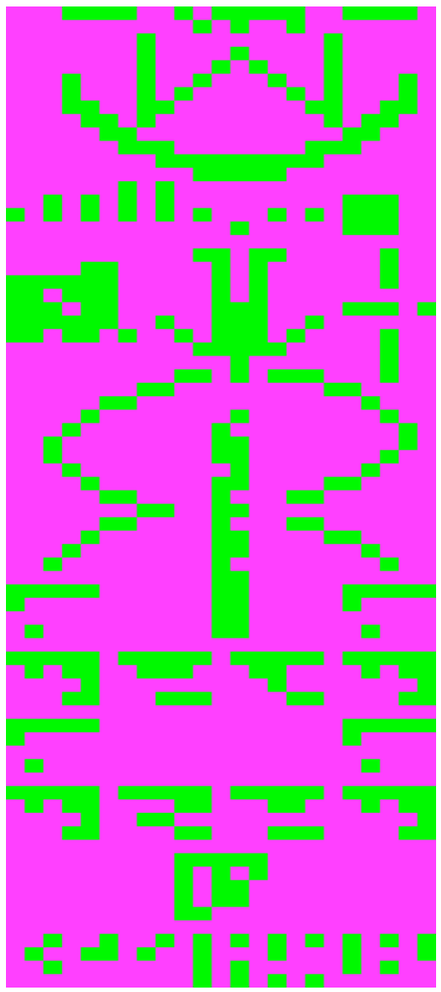
73 x 23

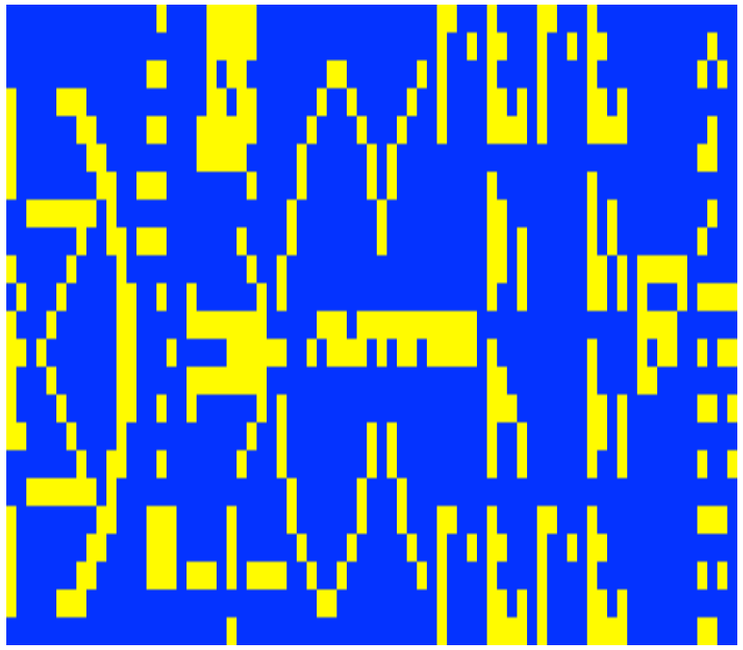


23 x 73: what was different?

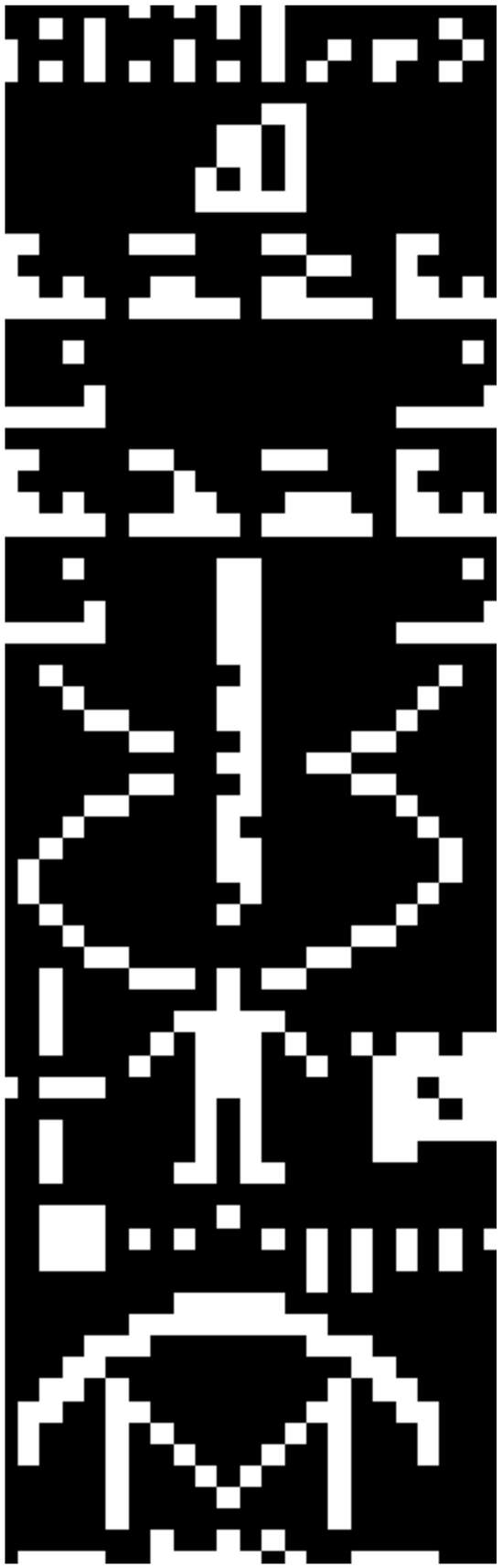


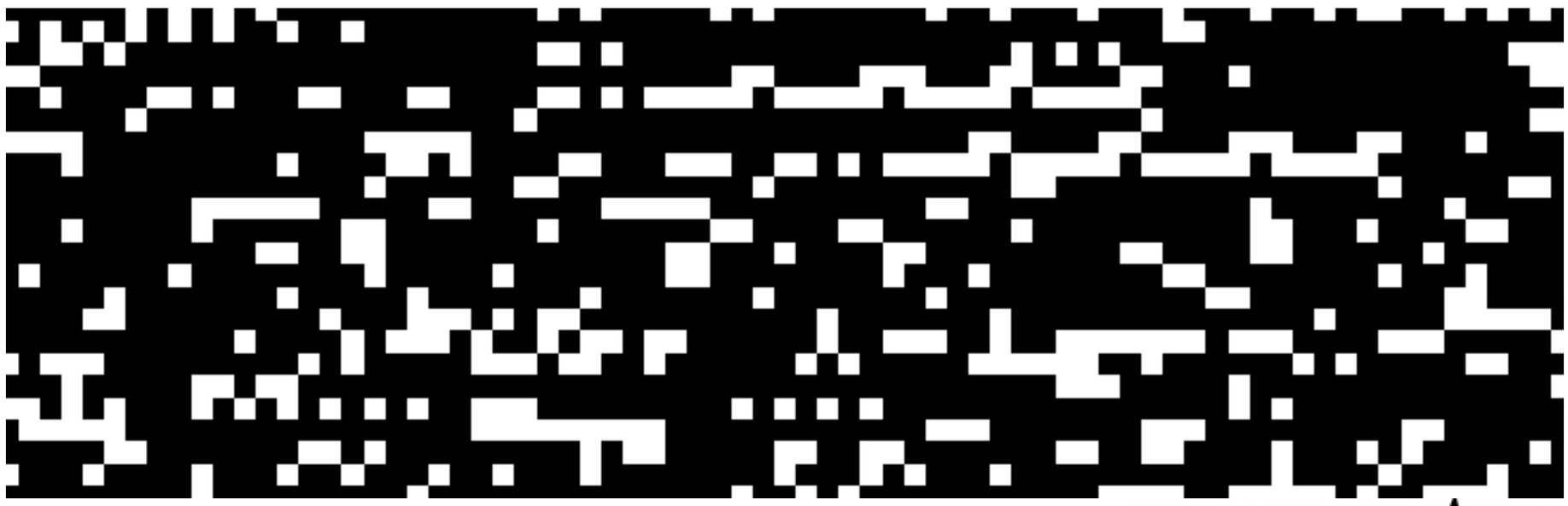
73 x 23



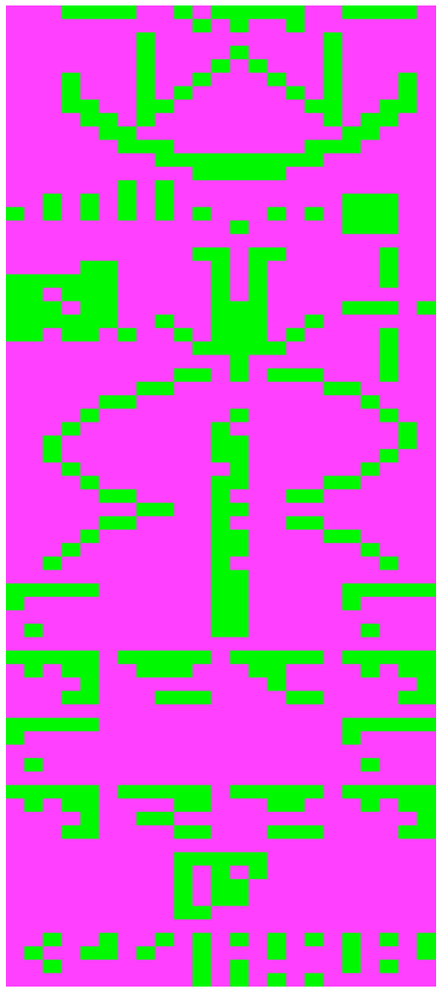


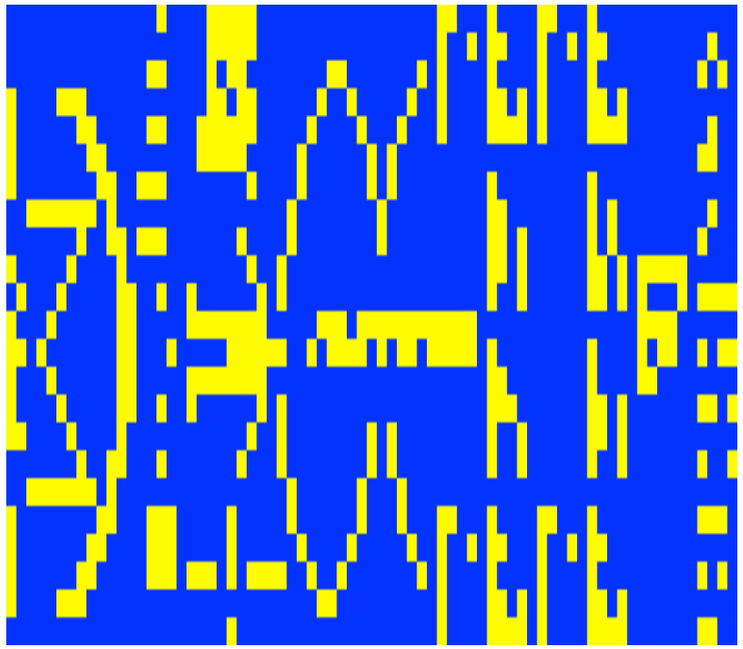
23 x 73: what was different?



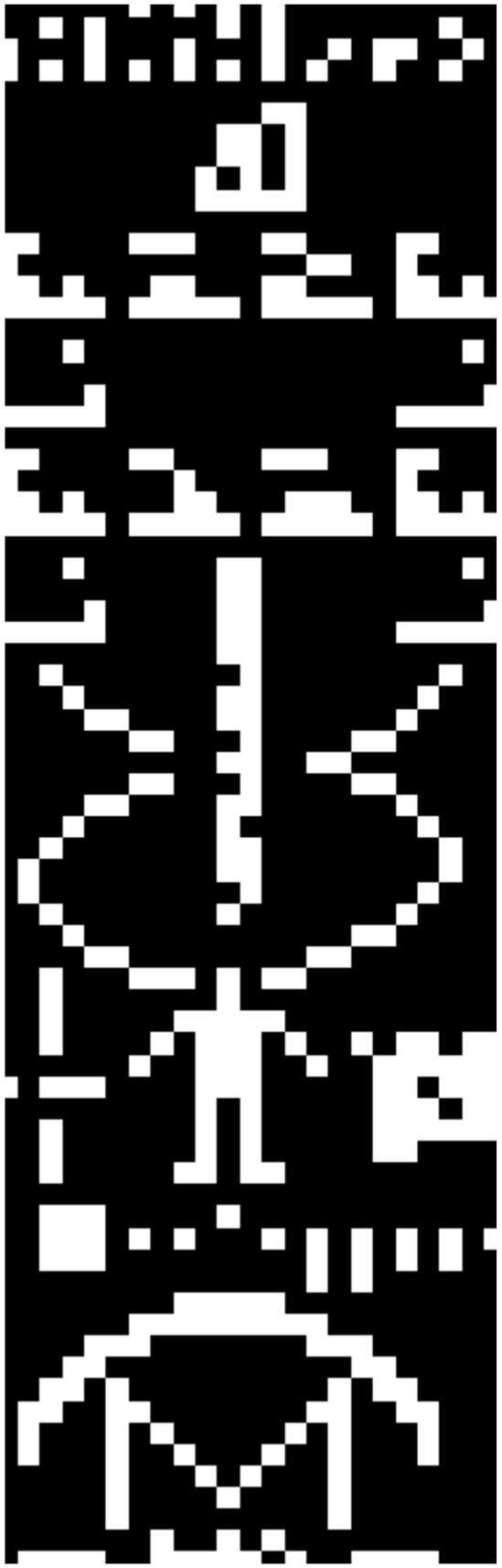


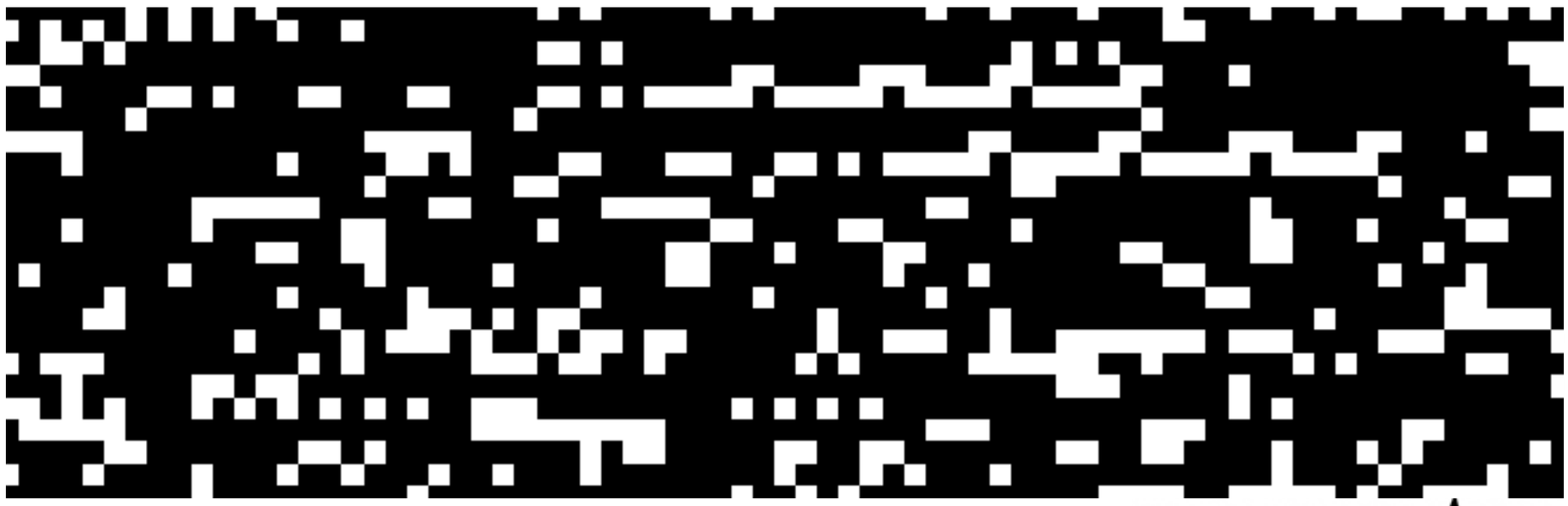
73 x 23



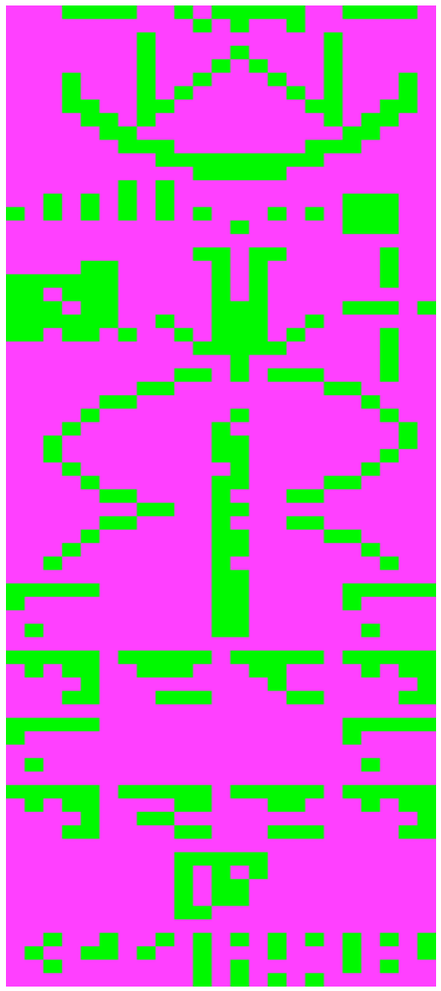


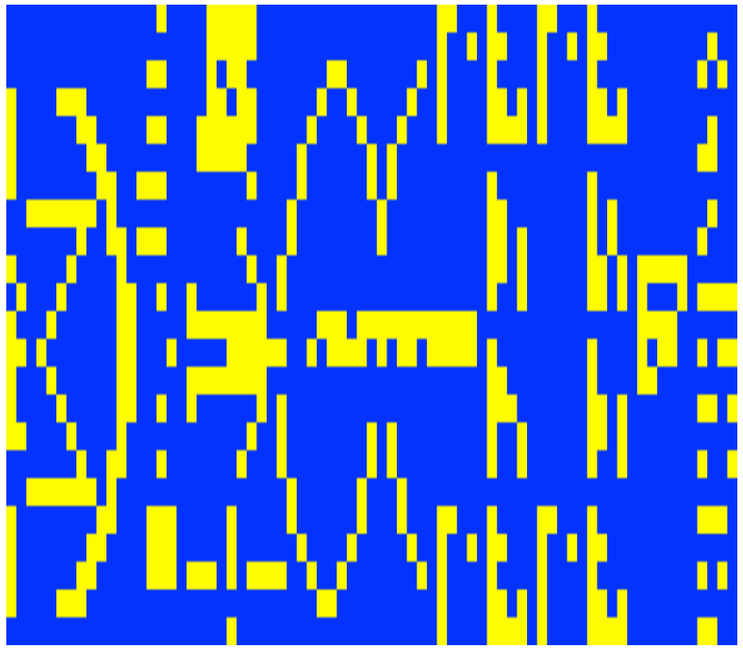
23 x 73: what was different?





73 x 23



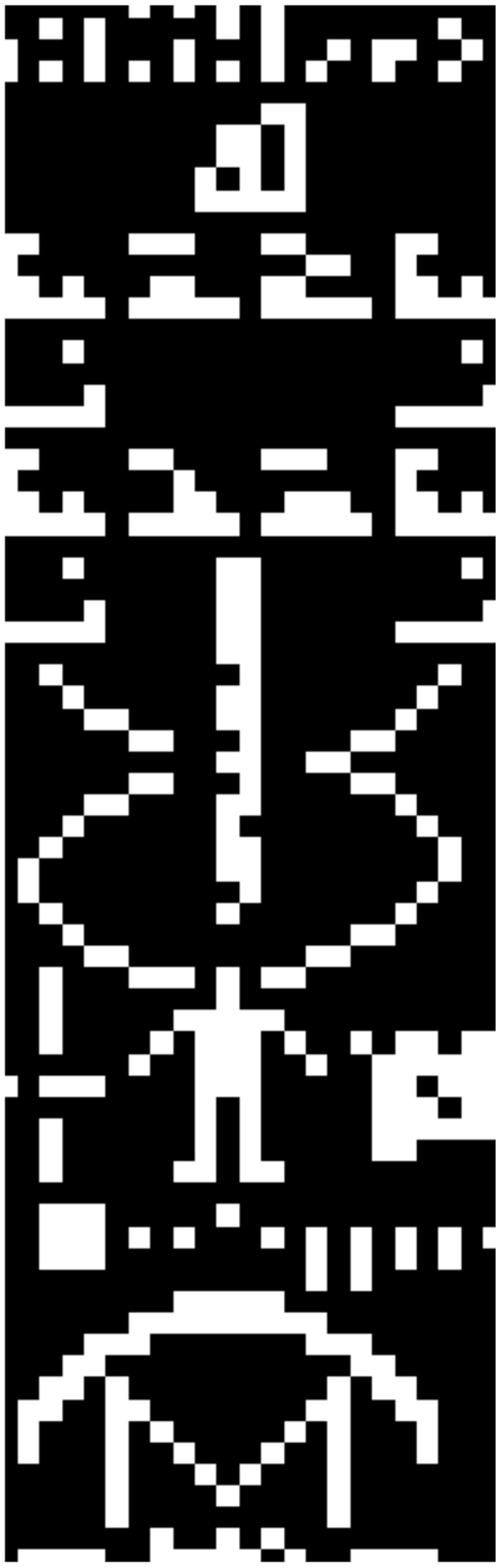


compare to:
http://en.wikipedia.org/wiki/Arecibo_message

23 x 73: what was different?

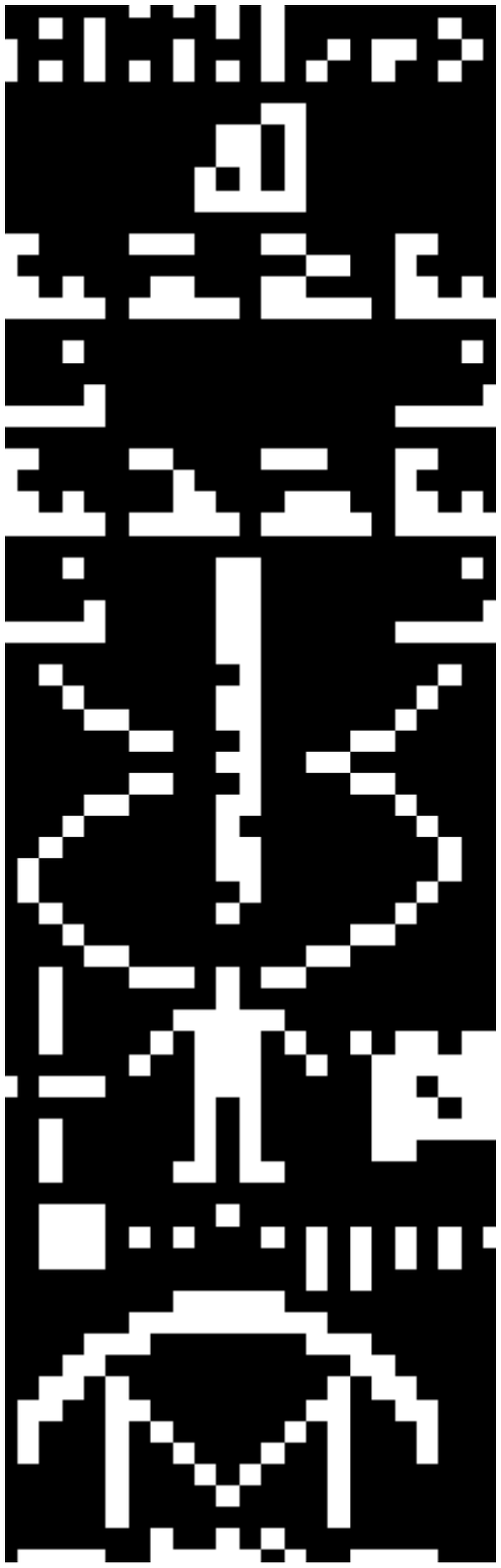
Numbers 1 to 10

73 x 23



compare to:
http://en.wikipedia.org/wiki/Arecibo_message

23 x 73: what was different?



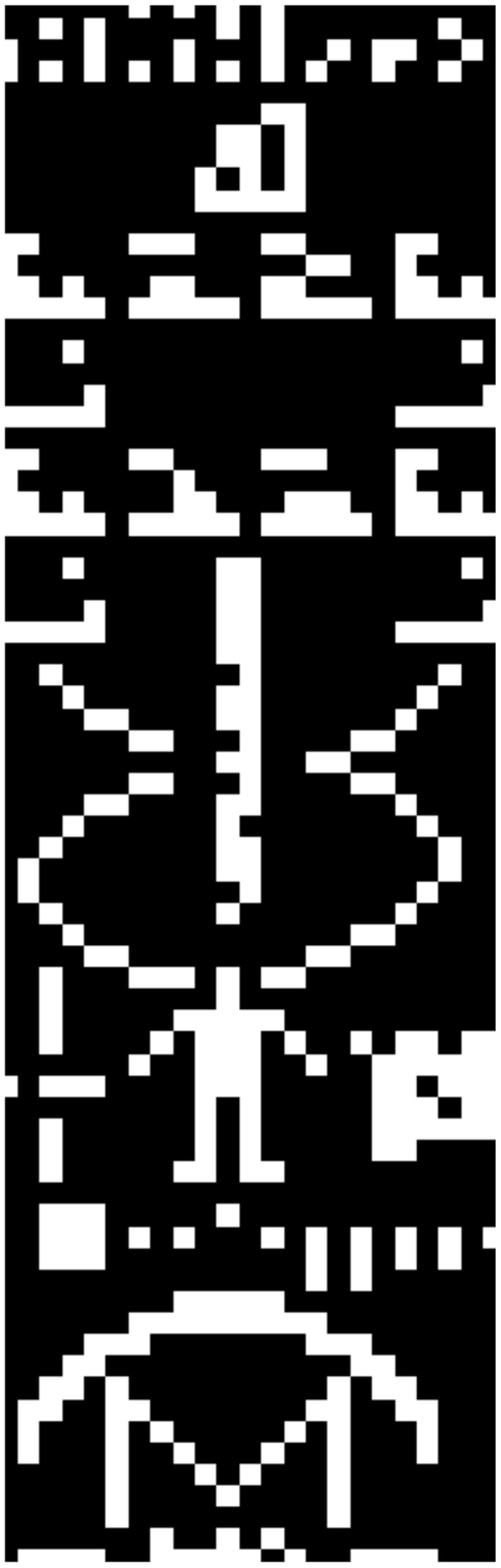
Numbers 1 to 10

Atomic Numbers of H,C,N,O,P (DNA)

73 x 23

compare to:
http://en.wikipedia.org/wiki/Arecibo_message

23 x 73: what was different?



Numbers 1 to 10

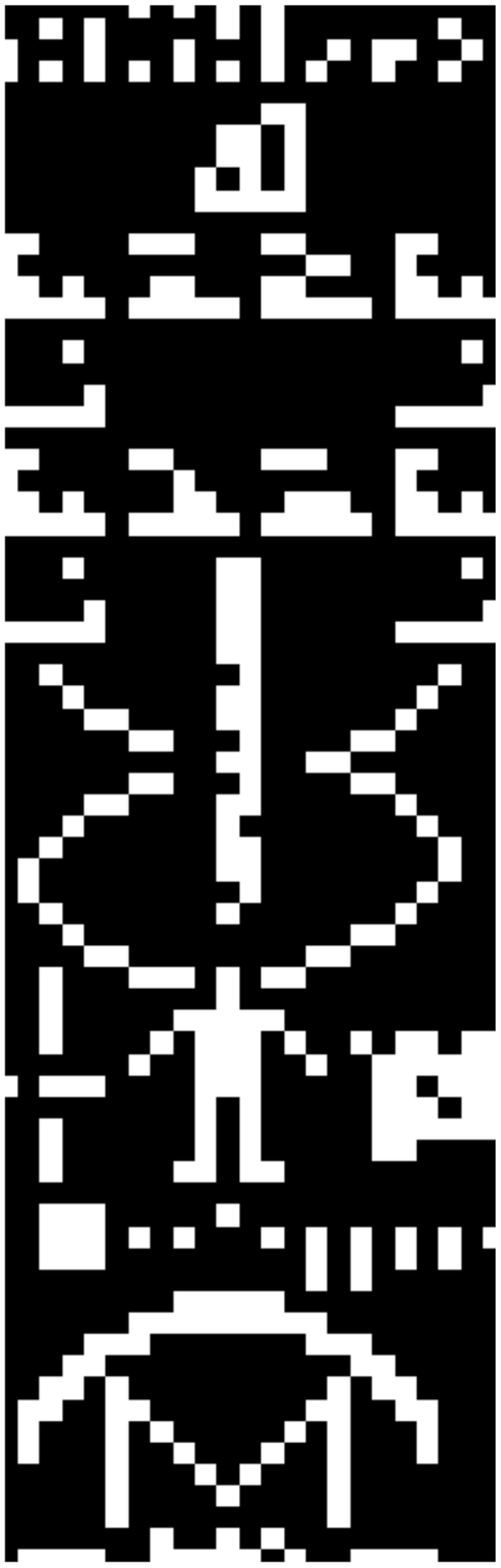
Atomic Numbers of H,C,N,O,P (DNA)

Formula for nucleotides in DNA

73 x 23

compare to:
http://en.wikipedia.org/wiki/Arecibo_message

23 x 73: what was different?

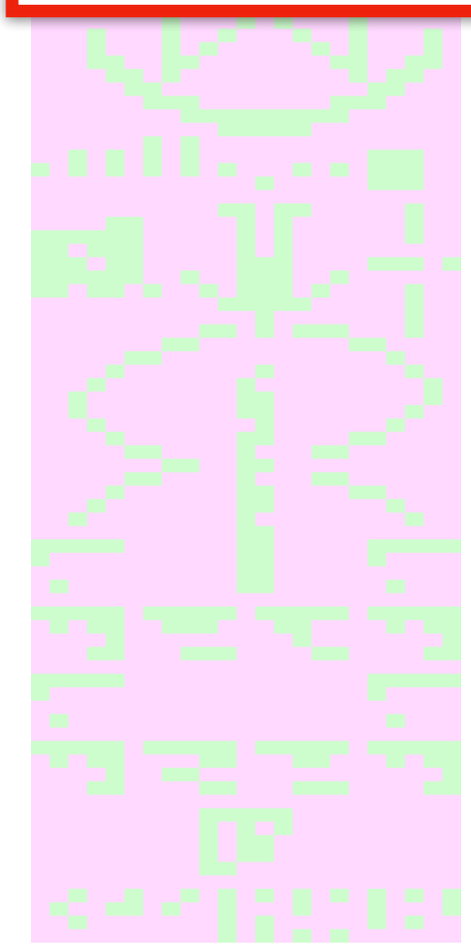


Numbers 1 to 10

Atomic Numbers of H,C,N,O,P (DNA)

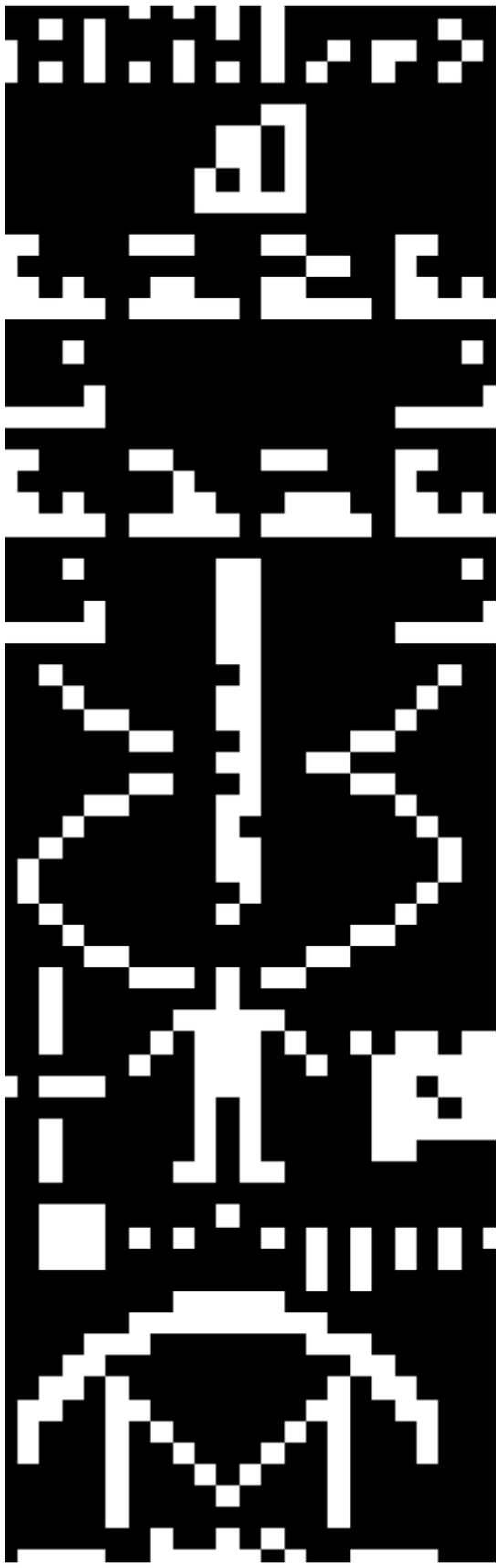
Formula for nucleotides in DNA

Estimated number of nucleotides in DNA



compare to:
http://en.wikipedia.org/wiki/Arecibo_message

23 x 73: what was different?



Numbers 1 to 10

Atomic Numbers of H,C,N,O,P (DNA)

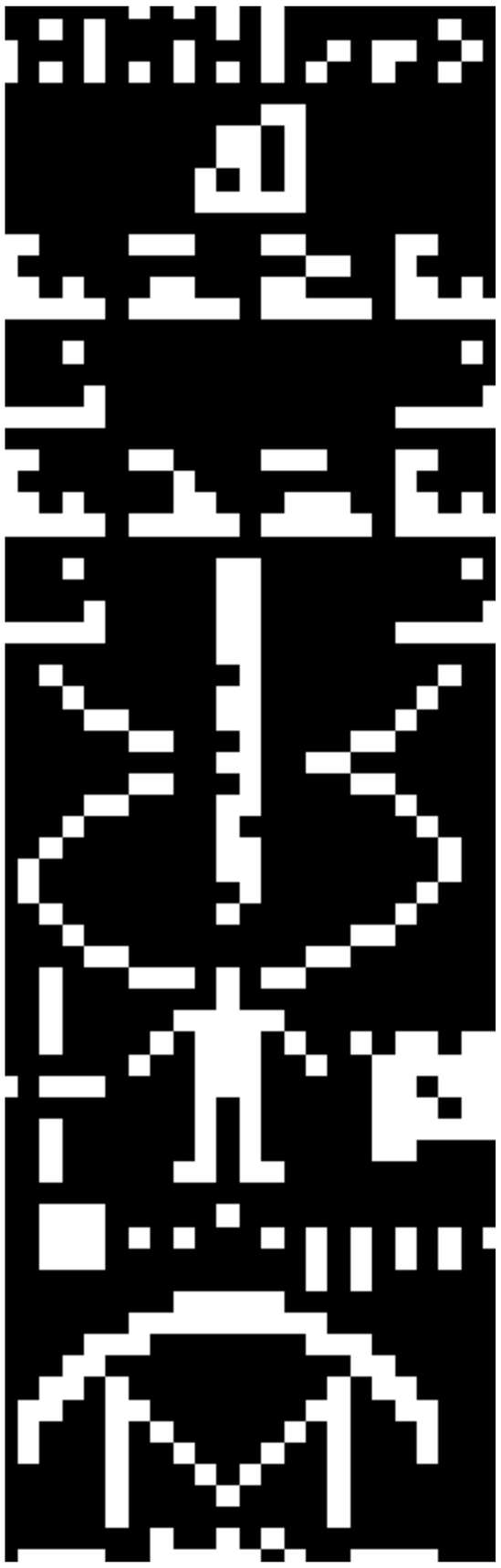
Formula for nucleotides in DNA

Estimated number of nucleotides in DNA

Graphic of Double Helix

compare to:
http://en.wikipedia.org/wiki/Arecibo_message

23 x 73: what was different?



Numbers 1 to 10

Atomic Numbers of H,C,N,O,P (DNA)

Formula for nucleotides in DNA

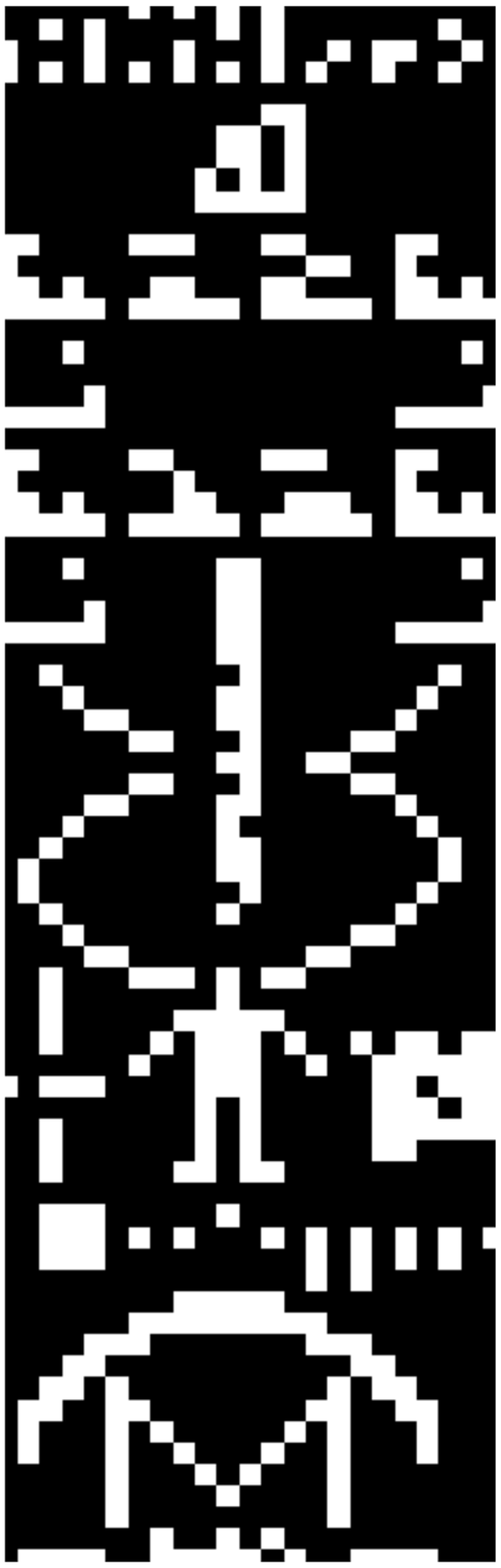
Estimated number of nucleotides in DNA

Graphic of Double Helix

Graphic of average human, dimensions, and population of Earth

compare to:
http://en.wikipedia.org/wiki/Arecibo_message

23 x 73: what was different?



Numbers 1 to 10

Atomic Numbers of H,C,N,O,P (DNA)

Formula for nucleotides in DNA

Estimated number of nucleotides in DNA

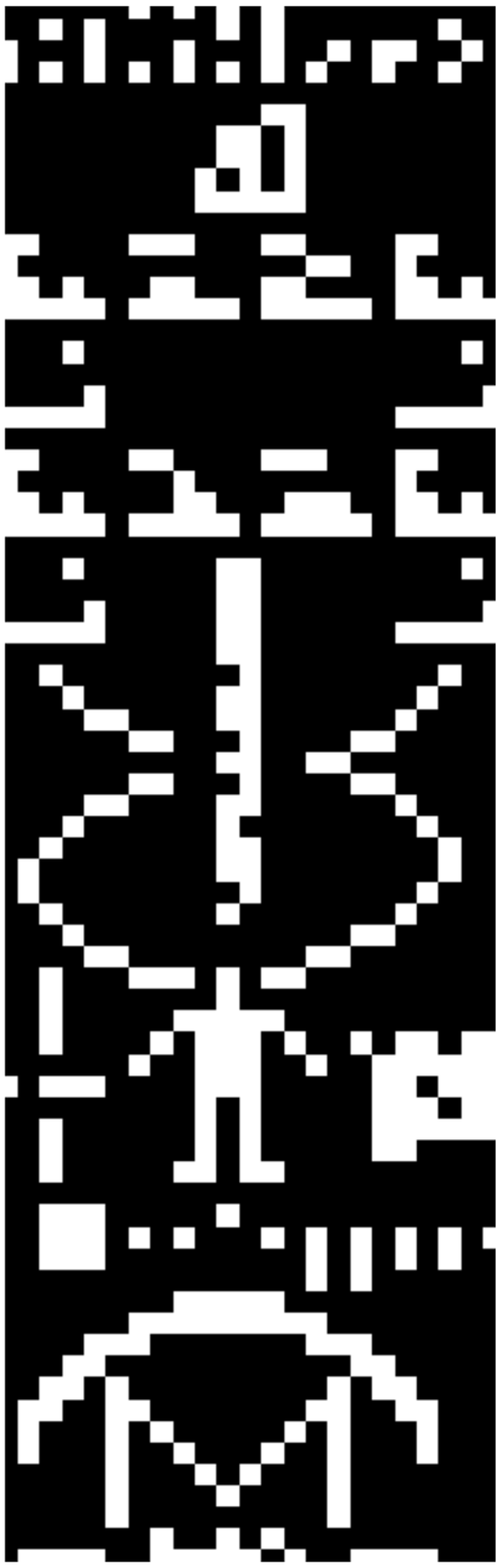
Graphic of Double Helix

Graphic of average human, dimensions, and population of Earth

Graphic of solar system + pluto

compare to:
http://en.wikipedia.org/wiki/Arecibo_message

23 x 73: what was different?



Numbers 1 to 10

Atomic Numbers of H,C,N,O,P (DNA)

Formula for nucleotides in DNA

Estimated number of nucleotides in DNA

Graphic of Double Helix

Graphic of average human, dimensions, and population of Earth

Graphic of solar system + pluto

Graphic of Arecibo Telescope

23 x 73: what was different?

5 basic pieces of image metadata

- Interpretation of individual values
  - units, scalars, vectors, tensors, measurement frame
- Dimension of array
  - dimension of domain sampled
  - # axes, or # indices for getting a single sample
- Choice of axis ordering (fast-to-slow, or slow-to-fast)
  - Culturally specific
- # samples along each axis
  - “640-by-480 image” or “N-by-M matrix”
  - FSV notes Section 3.1 ...
- Image location & Orientation of each Axis
  - Summarized by homogenous-coords affine transform
  - FSV notes Section 3.2: “index-space” “world-space”
  - Image Boundaries: node-centered vs cell-centered samples

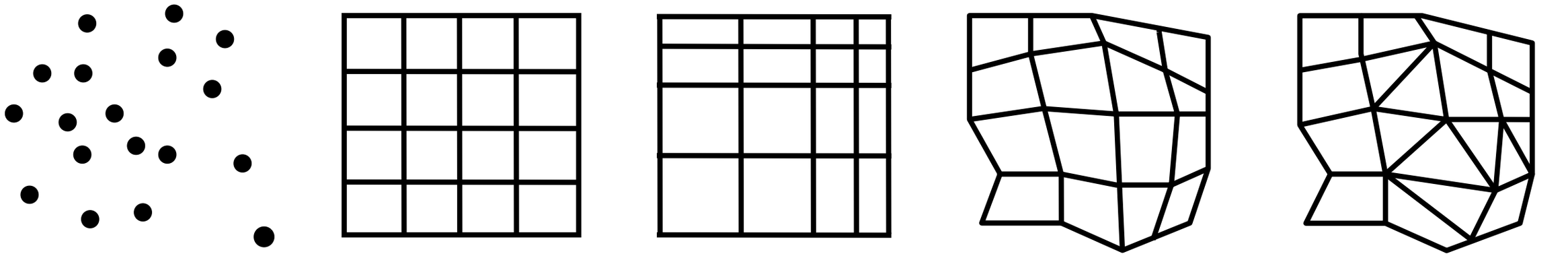
Discrete vs. Continuous

- In the abstract sense, domains are **continuous** objects
- We **discretize** a domain by
  - Taking a finite set of **samples** (positional information) living in some **embedding space**
  - Gluing them together with **connectivity** that defines the rest of the domain as a piecewise collection of **cells**

Grids (Meshes)

- Meshes combine positional information (geometry) with topological information (connectivity).

- Mesh type can differ substantial depending in the way mesh cells are formed.



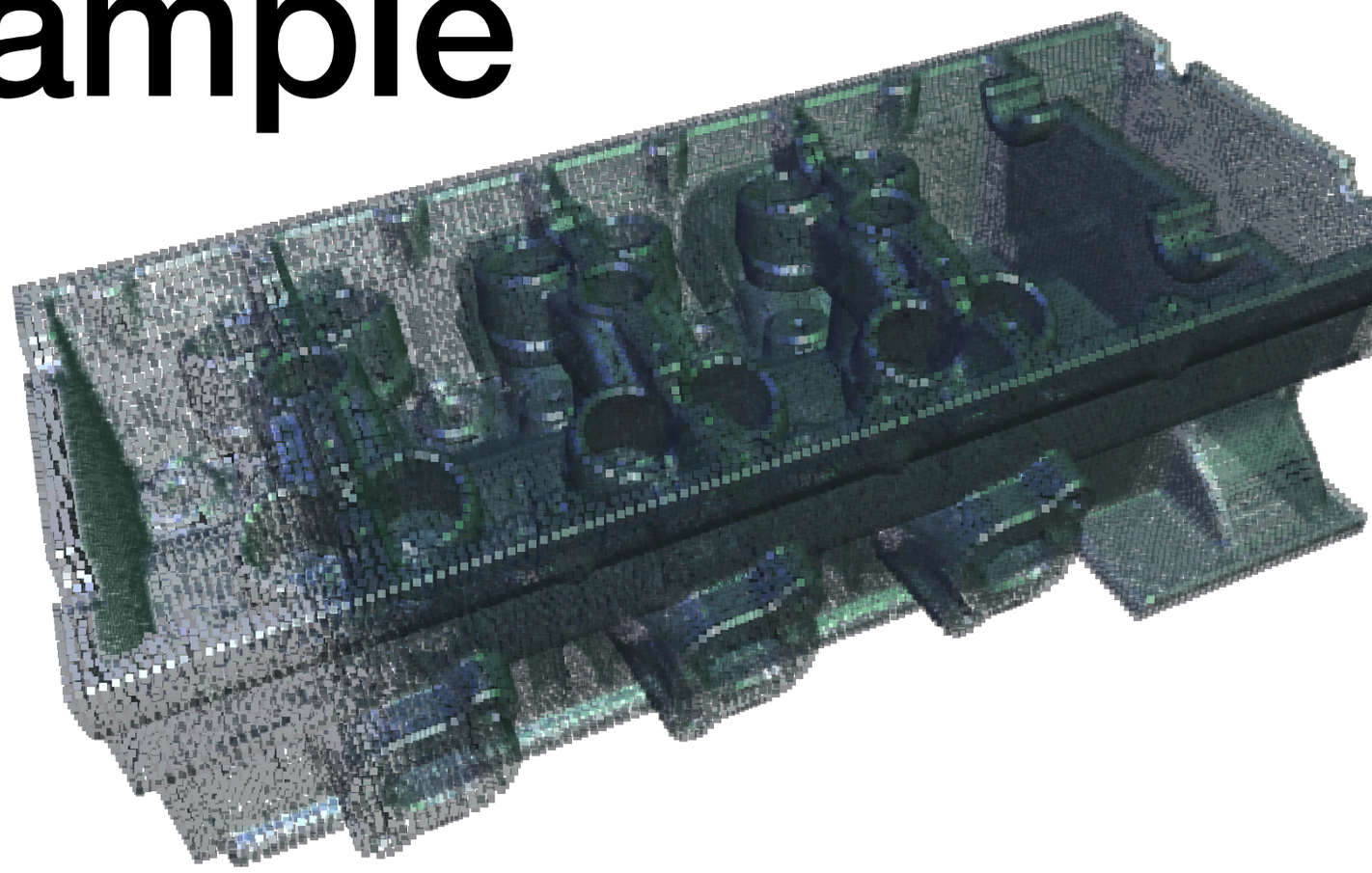
From Weiskopf, Machiraju, Möller

Grid Classification

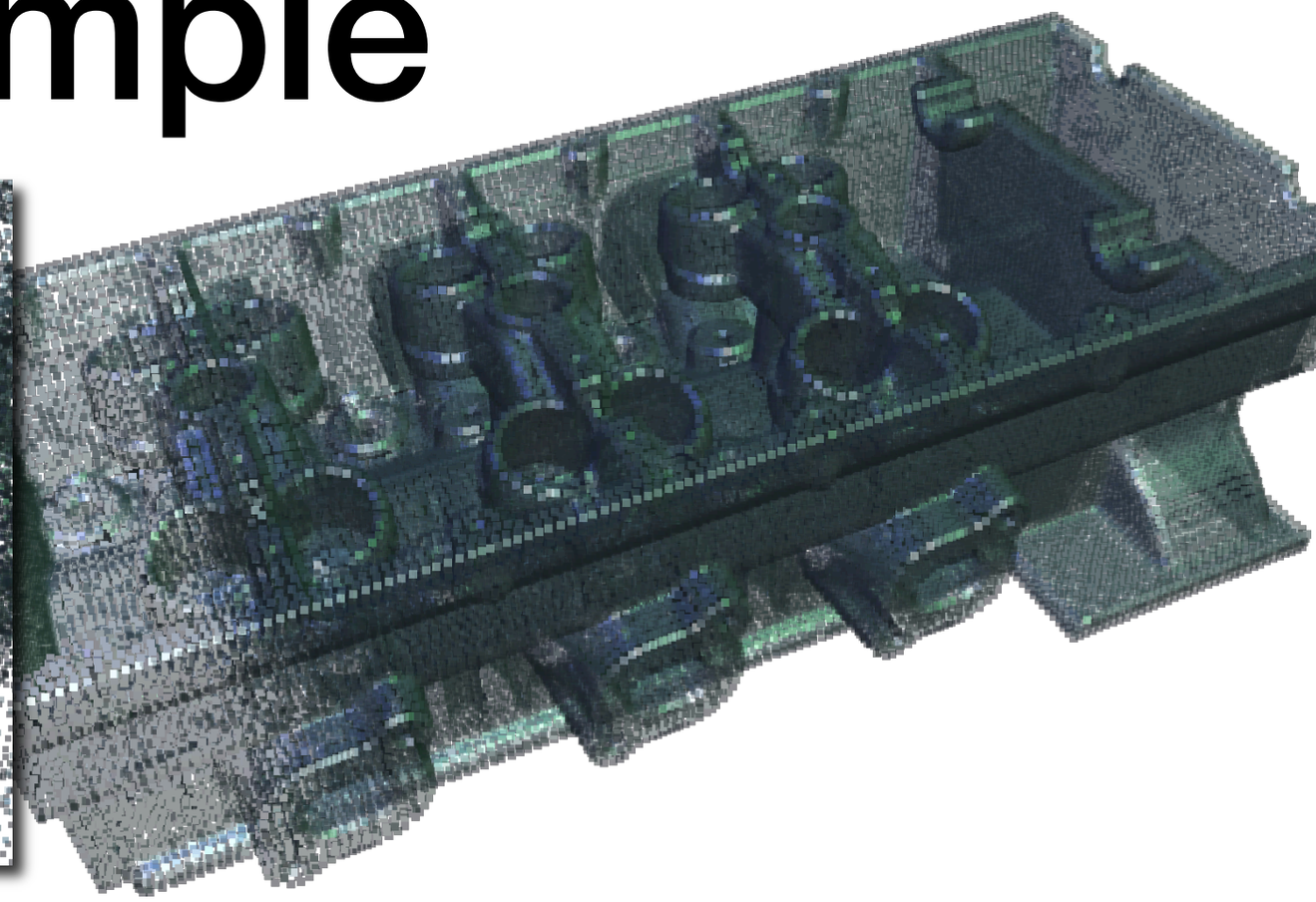
- Geometry
  - Positions of mesh **vertices** in Euclidean Space
  - Can follow uniform, aligned, on arbitrary patterns
- Topology
  - Collections of **cells** that decompose the domain
  - Define neighborhoods / locality
  - Can be connected in various ways: structured / unstructured

Example

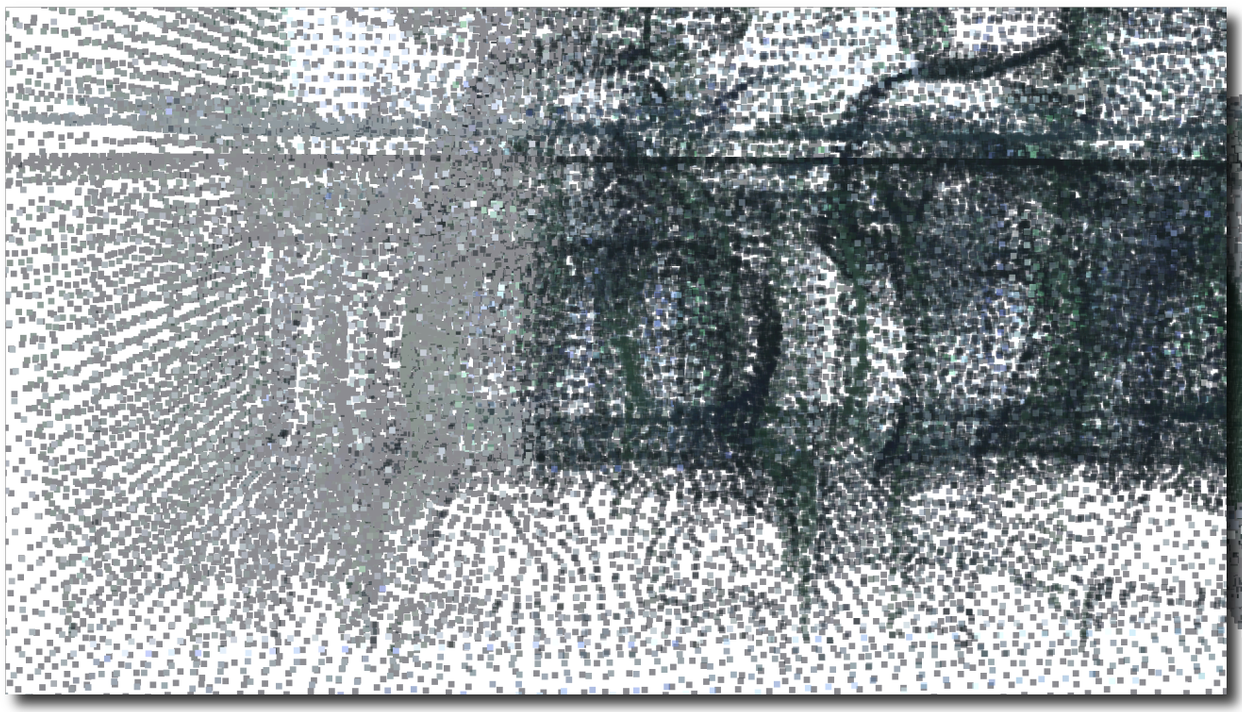
Example

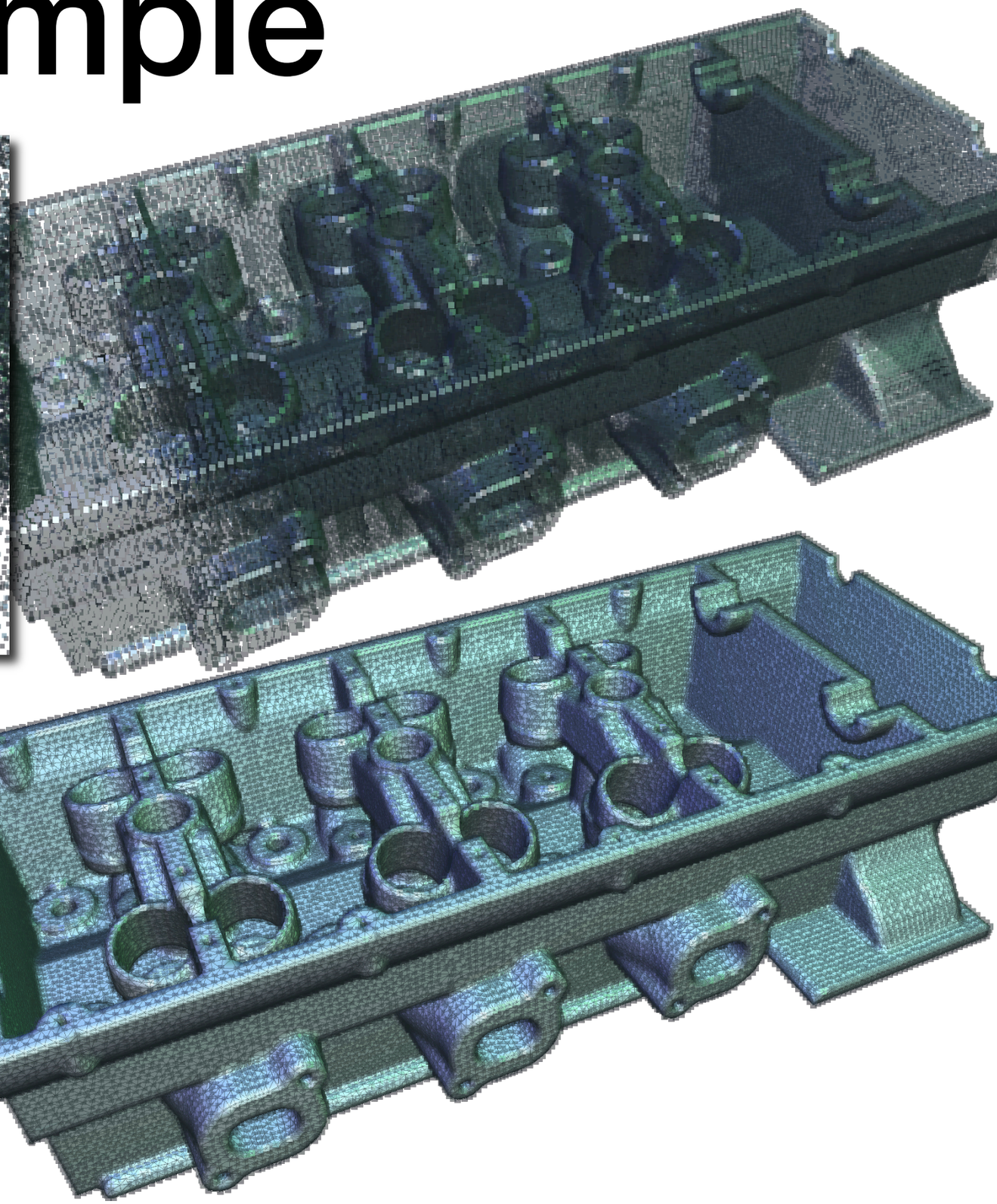


Example

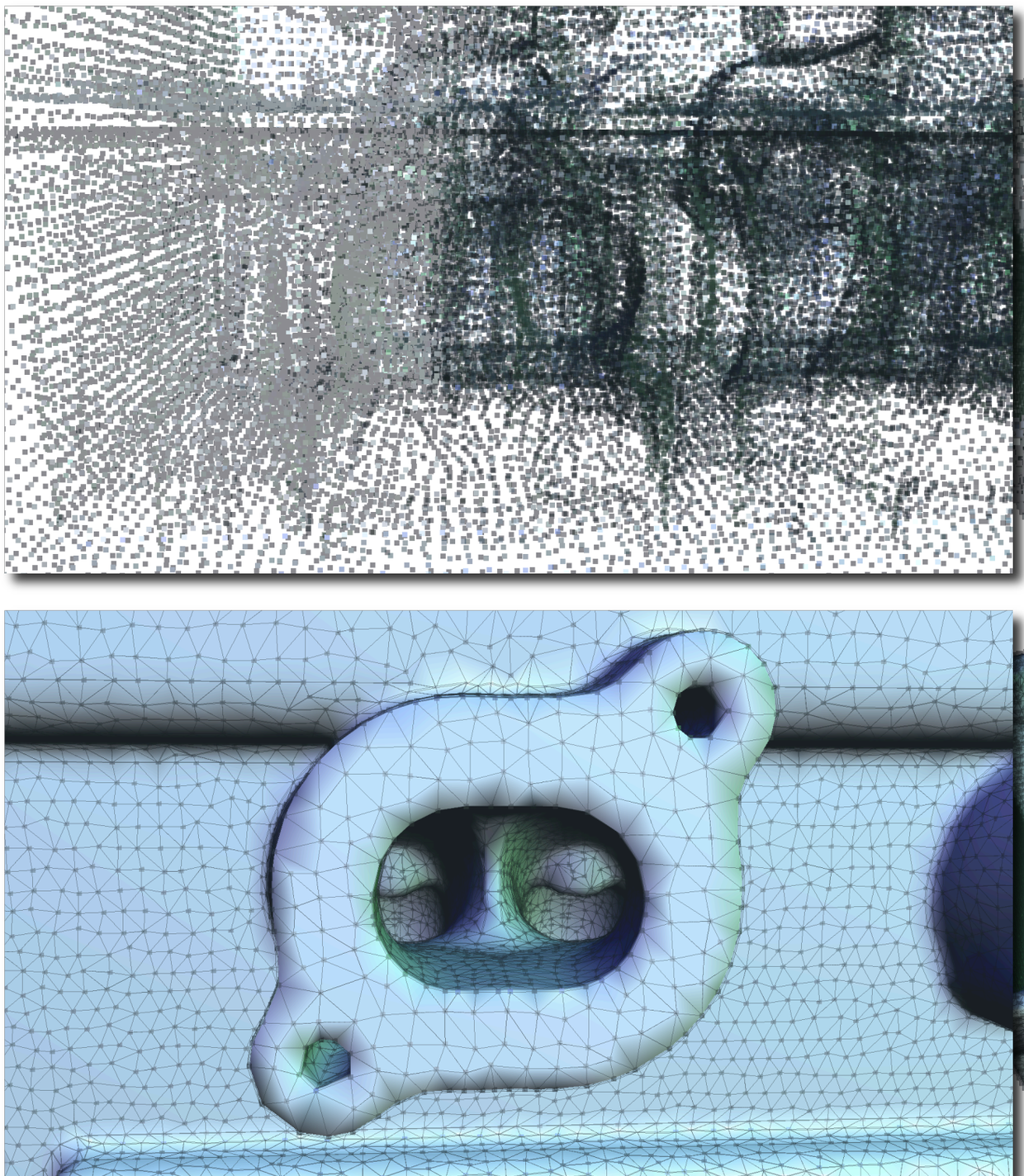


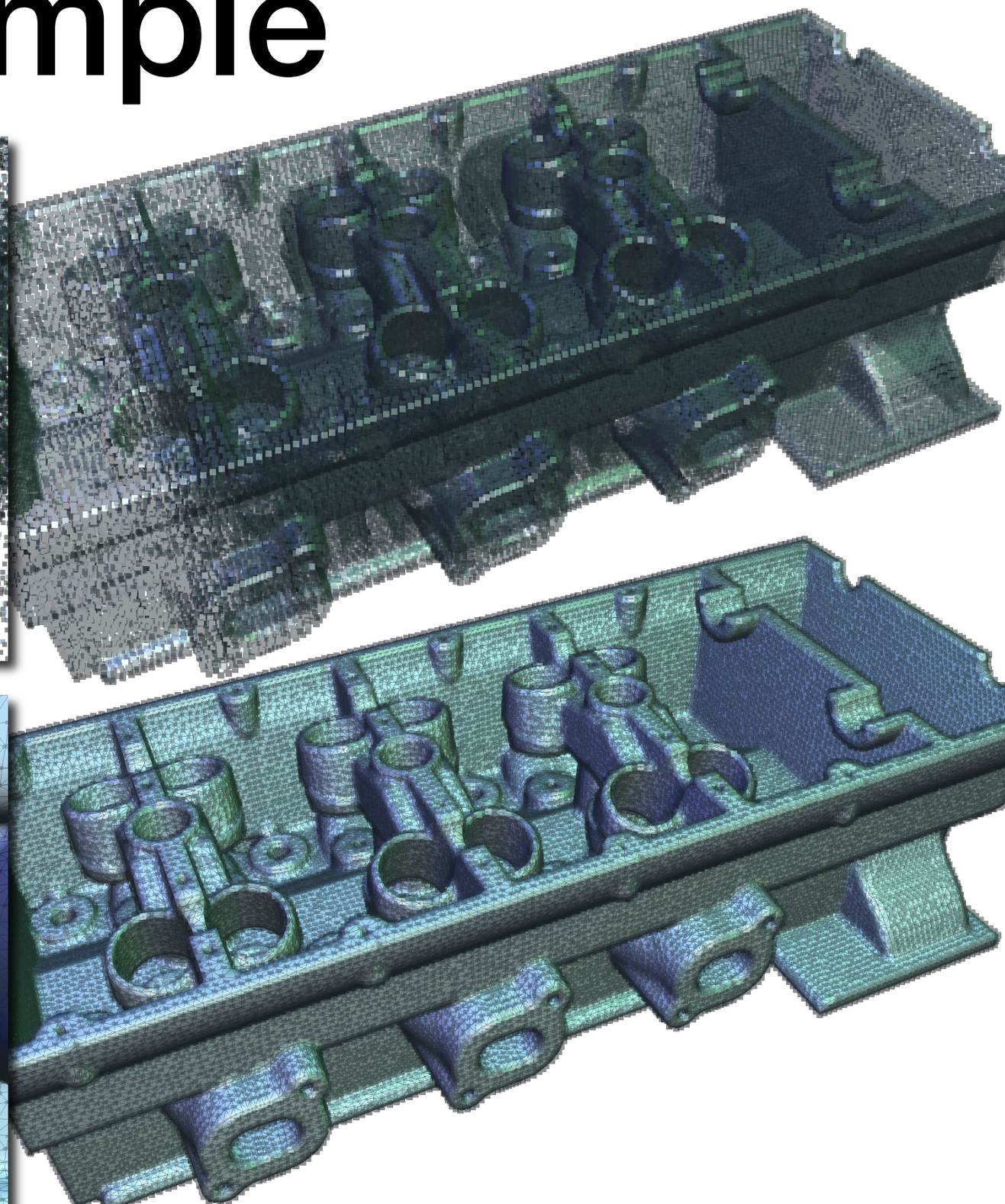
Example



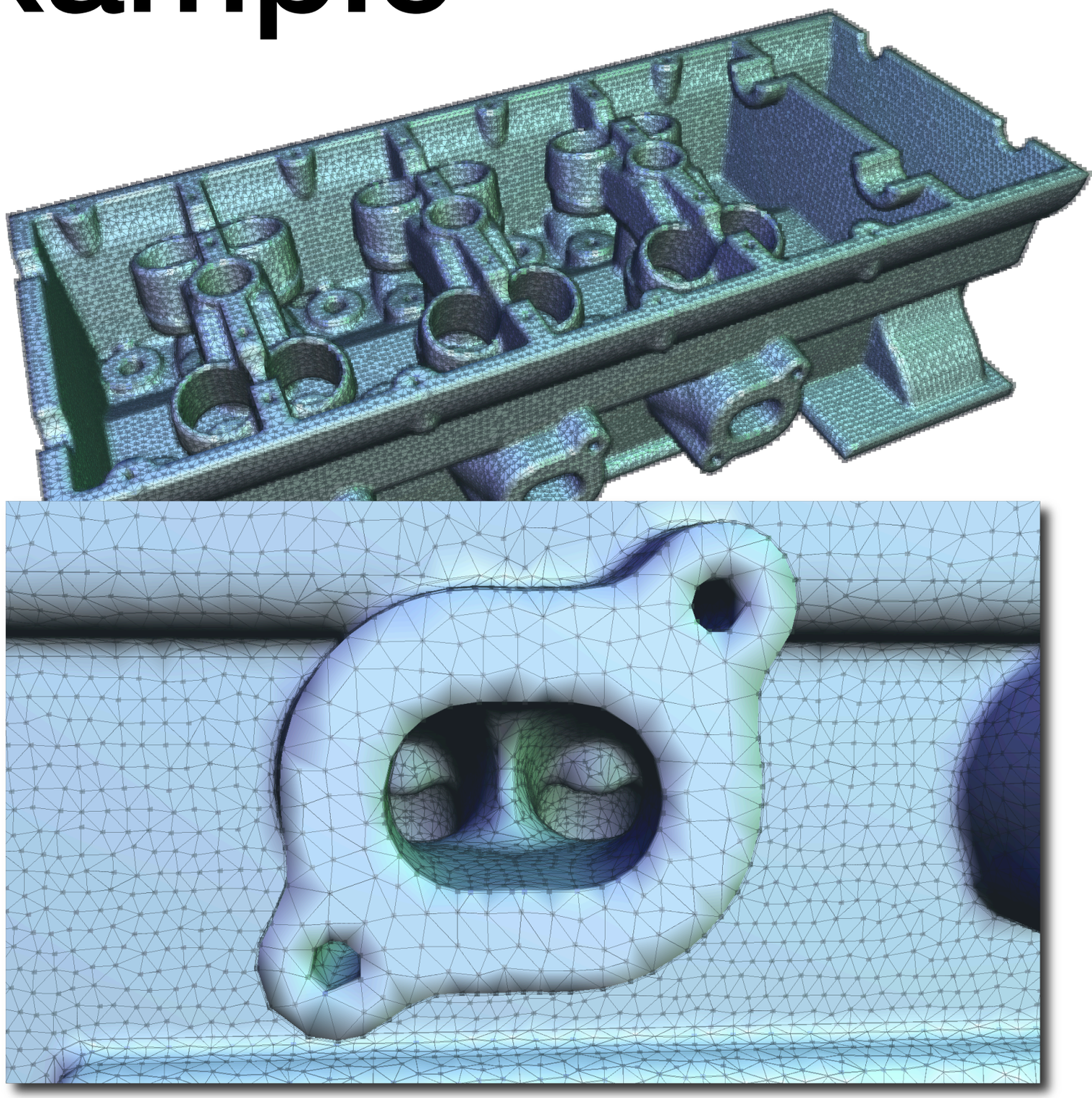


Example



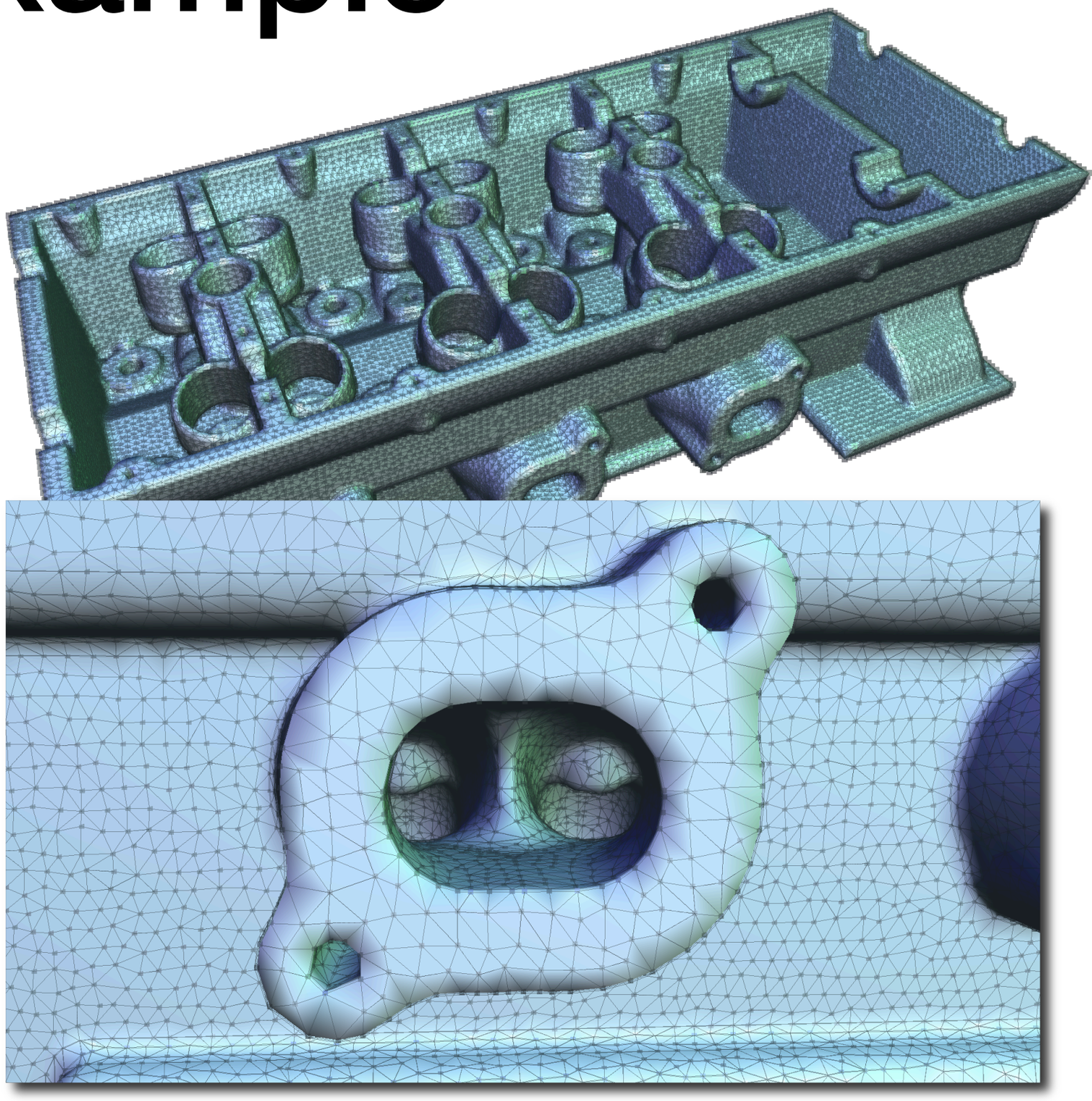


Example



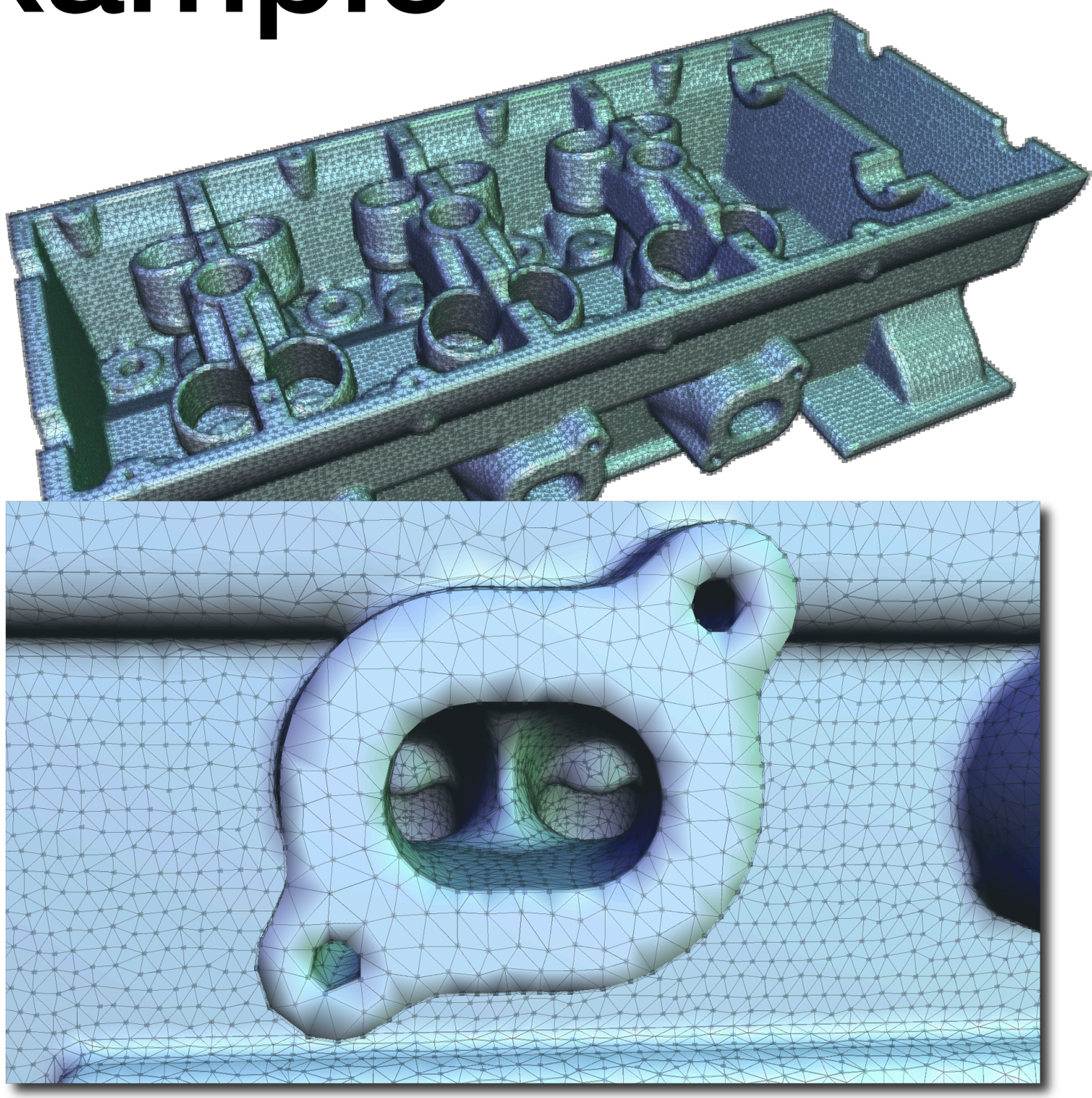
Example

- Embedding Space:



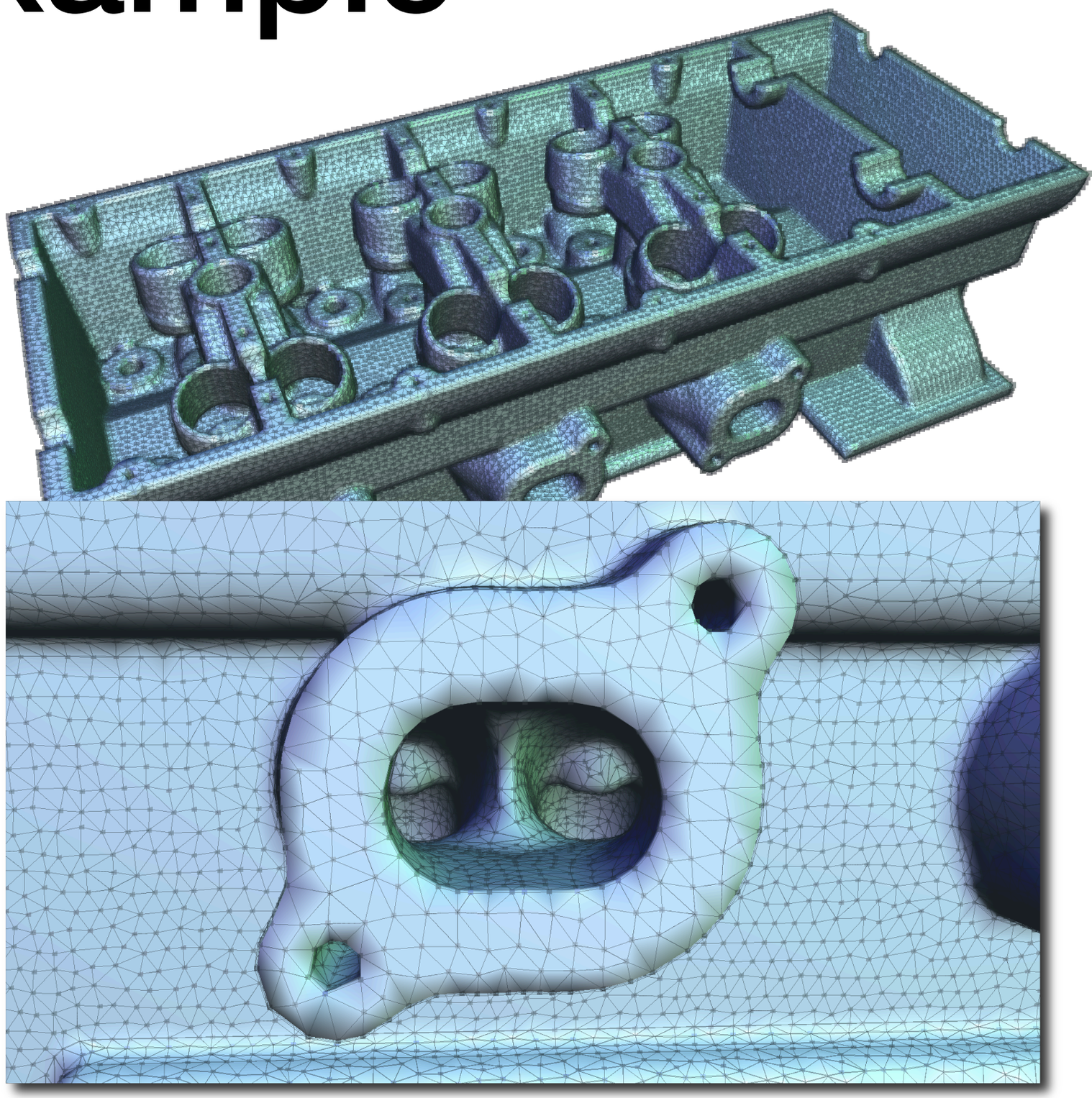
Example

- Embedding Space:
  - 3D Space



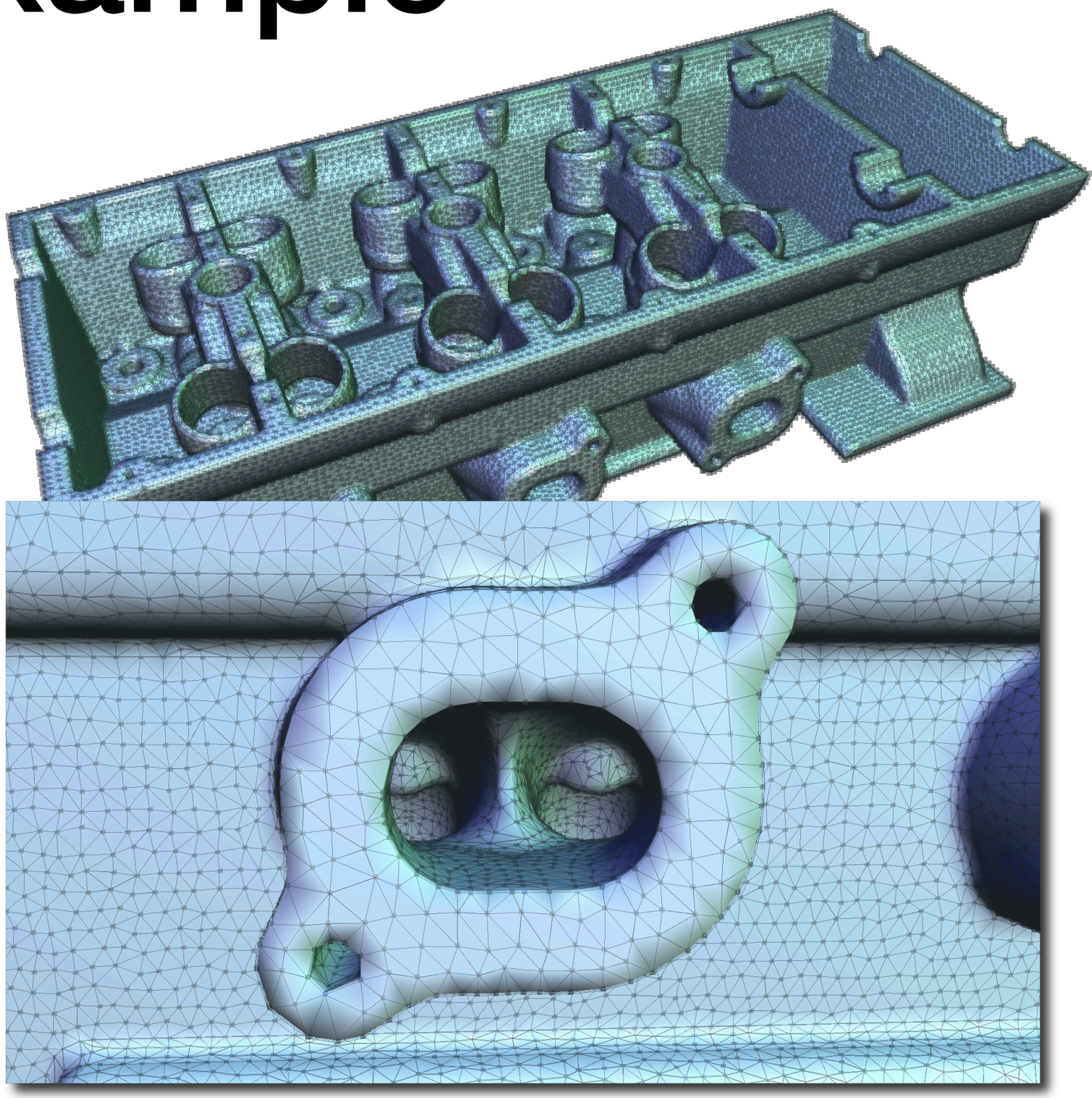
Example

- Embedding Space:
  - 3D Space
- Cellular Elements:



Example

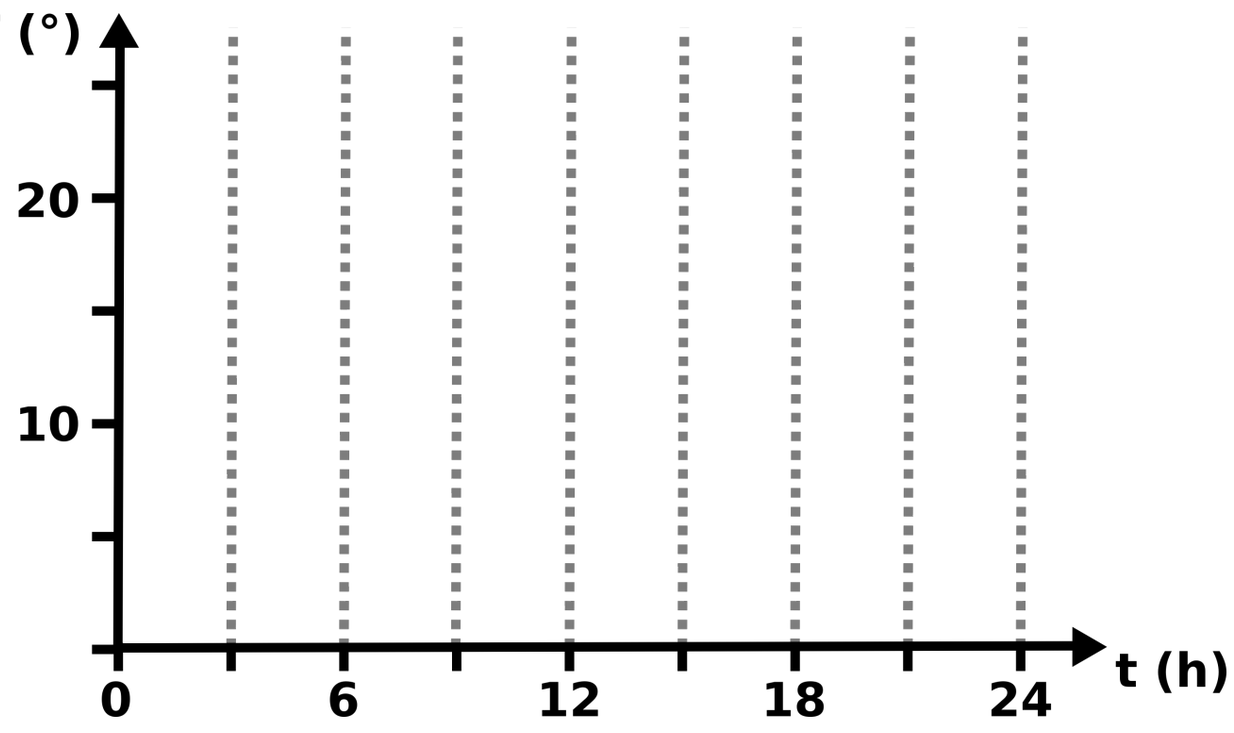
- Embedding Space:
  - 3D Space
- Cellular Elements:
  - Triangles



Encoding Functions Defined on Regular Grids of $\mathbb{R}$

$$f : [0, 24] \to \mathbb{R}$$

- Since our domain is 1-dimensional, we need a 1-dimensional data structure



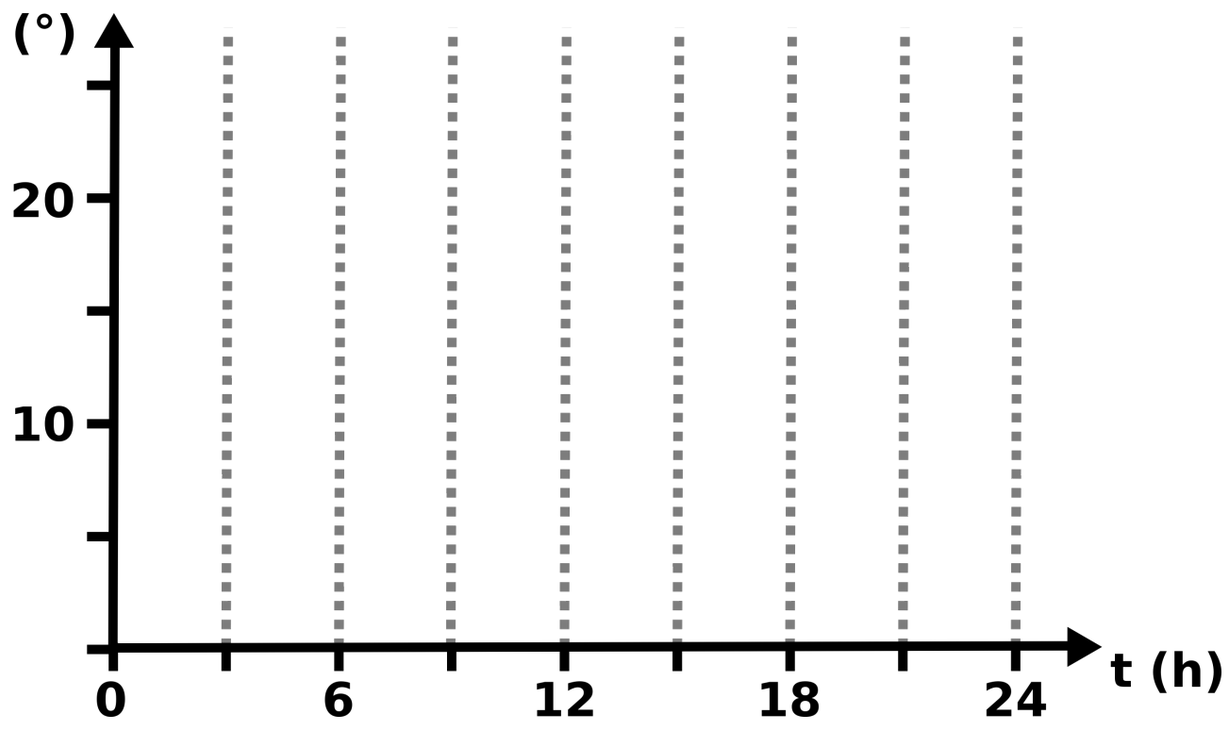
Encoding Functions Defined on Regular Grids of $\mathbb{R}$

$$f : [0, 24] \to \mathbb{R}$$

- Since our domain is 1-dimensional, we need a 1-dimensional data structure

- Arrays!

9	7	10	12	16	18	11	10	7



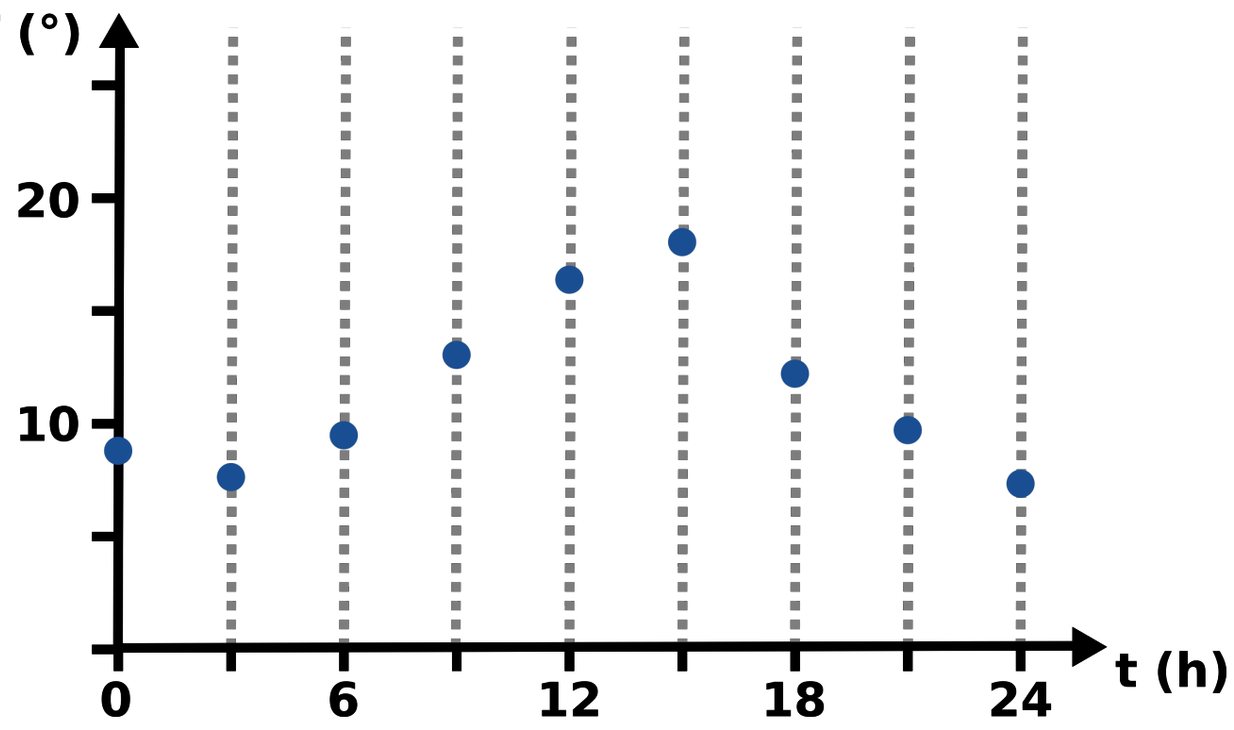
Encoding Functions Defined on Regular Grids of $\mathbb{R}$

$$f : [0, 24] \to \mathbb{R}$$

- Since our domain is 1-dimensional, we need a 1-dimensional data structure

- Arrays!

9	7	10	12	16	18	11	10	7



Encoding Functions Defined on Regular Grids of $\mathbb{R}$

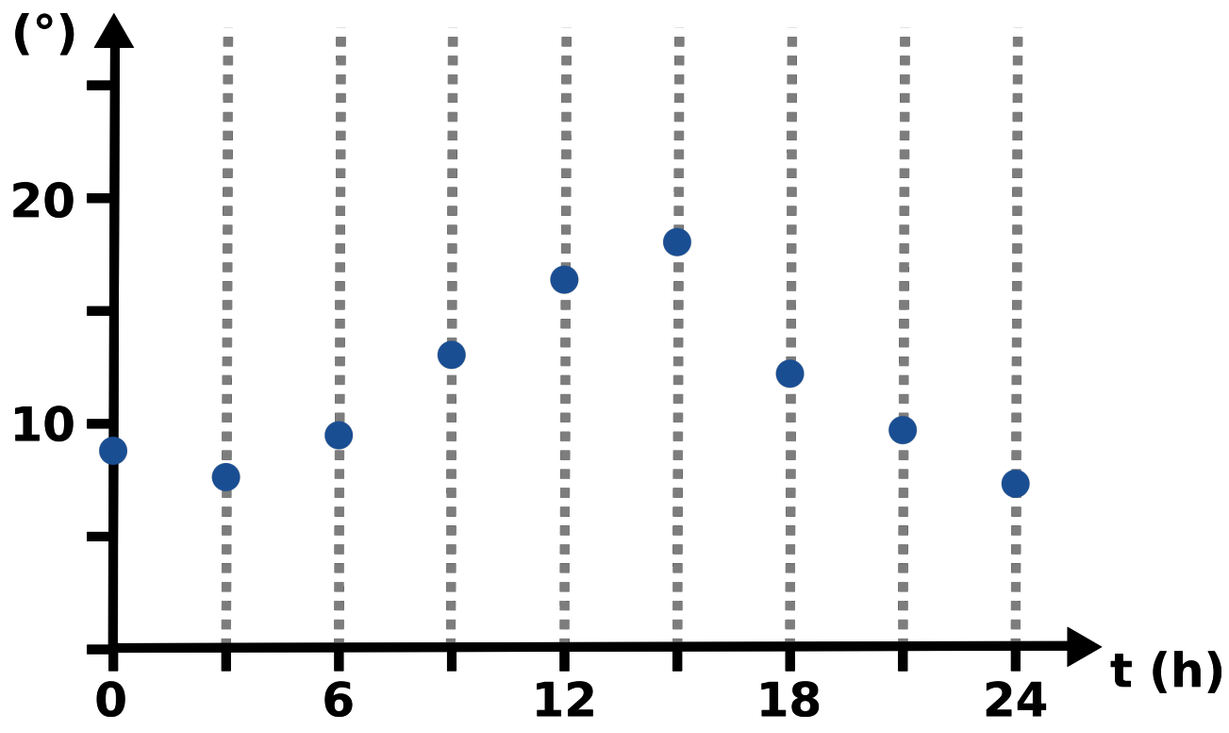
$$f : [0, 24] \to \mathbb{R}$$

- Since our domain is 1-dimensional, we need a 1-dimensional data structure

- Arrays!

9	7	10	12	16	18	11	10	7

- Need anything else?

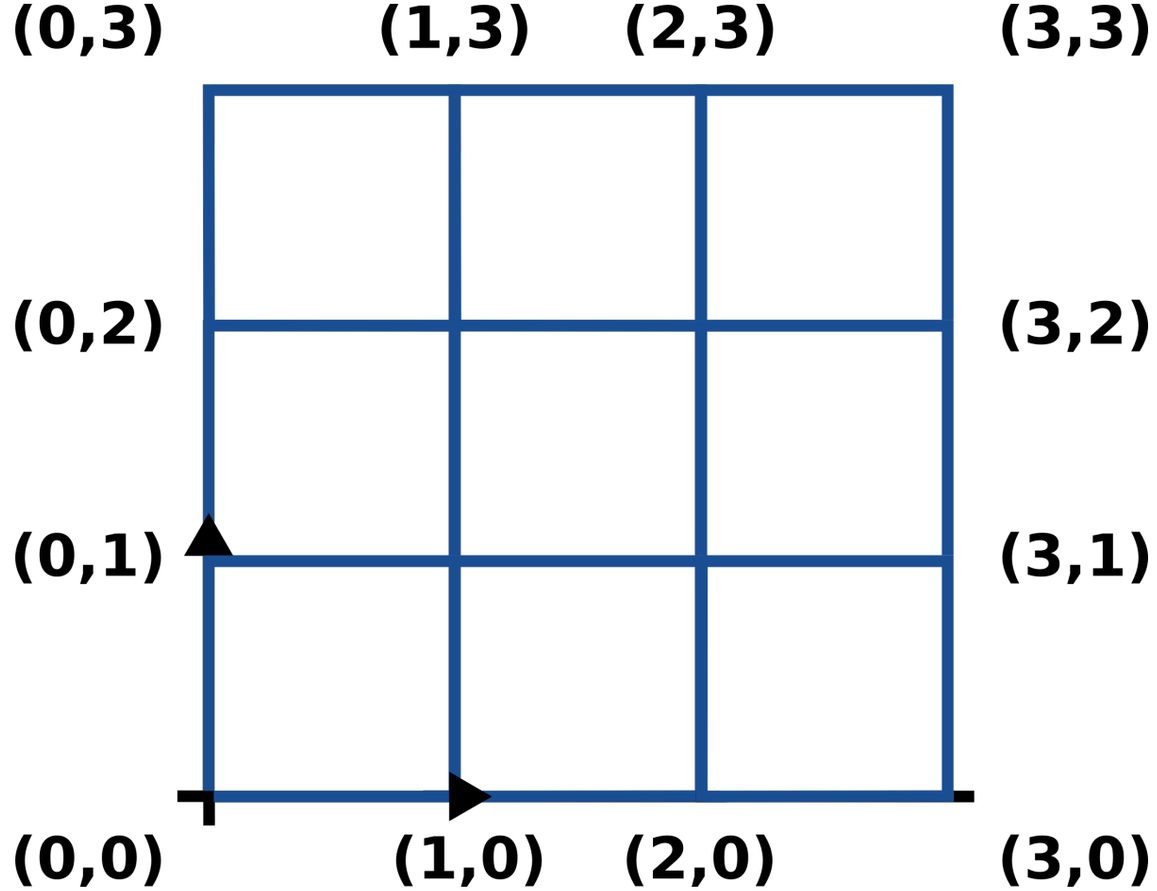


Encoding Functions Defined on Regular Grids of $\mathbb{R}^2$

- A Small Problem
  - Computer memory is one-dimensional!

$$f : [0,3] \times [0,3] \to \mathbb{R}$$

$$f(p) = \sqrt{p_x^2 + p_y^2}$$

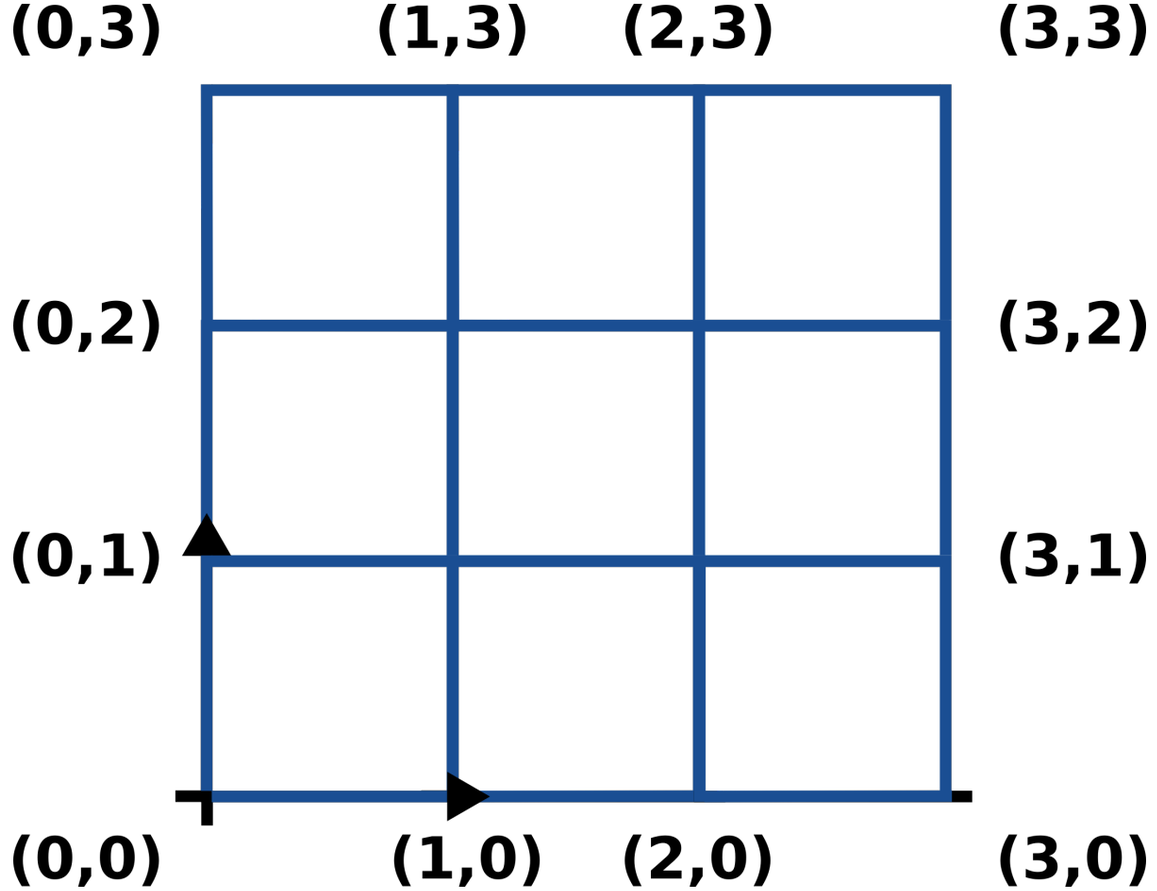


Encoding Functions Defined on Regular Grids of $\mathbb{R}^2$

- A Small Problem
  - Computer memory is one-dimensional!
- Solution
  - Order samples with a space-filling curve
  - Span one dimension at a time
  - Store metadata that describes the extent

$$f : [0,3] \times [0,3] \to \mathbb{R}$$

$$f(p) = \sqrt{p_x^2 + p_y^2}$$

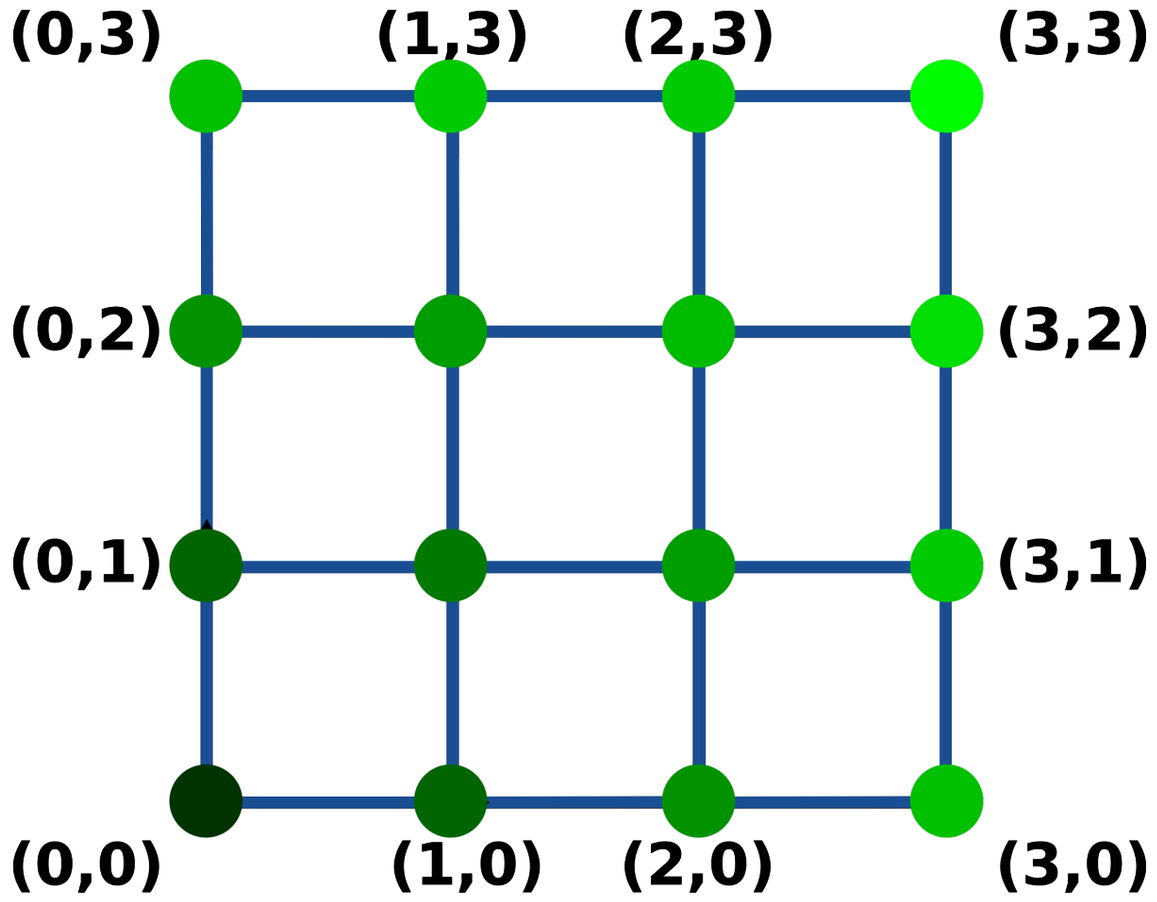


Encoding Functions Defined on Regular Grids of $\mathbb{R}^2$

- A Small Problem
  - Computer memory is one-dimensional!
- Solution
  - Order samples with a space-filling curve
  - Span one dimension at a time
  - Store metadata that describes the extent

$$f : [0,3] \times [0,3] \rightarrow \mathbb{R}$$

$$f(p) = \sqrt{p_x^2 + p_y^2}$$

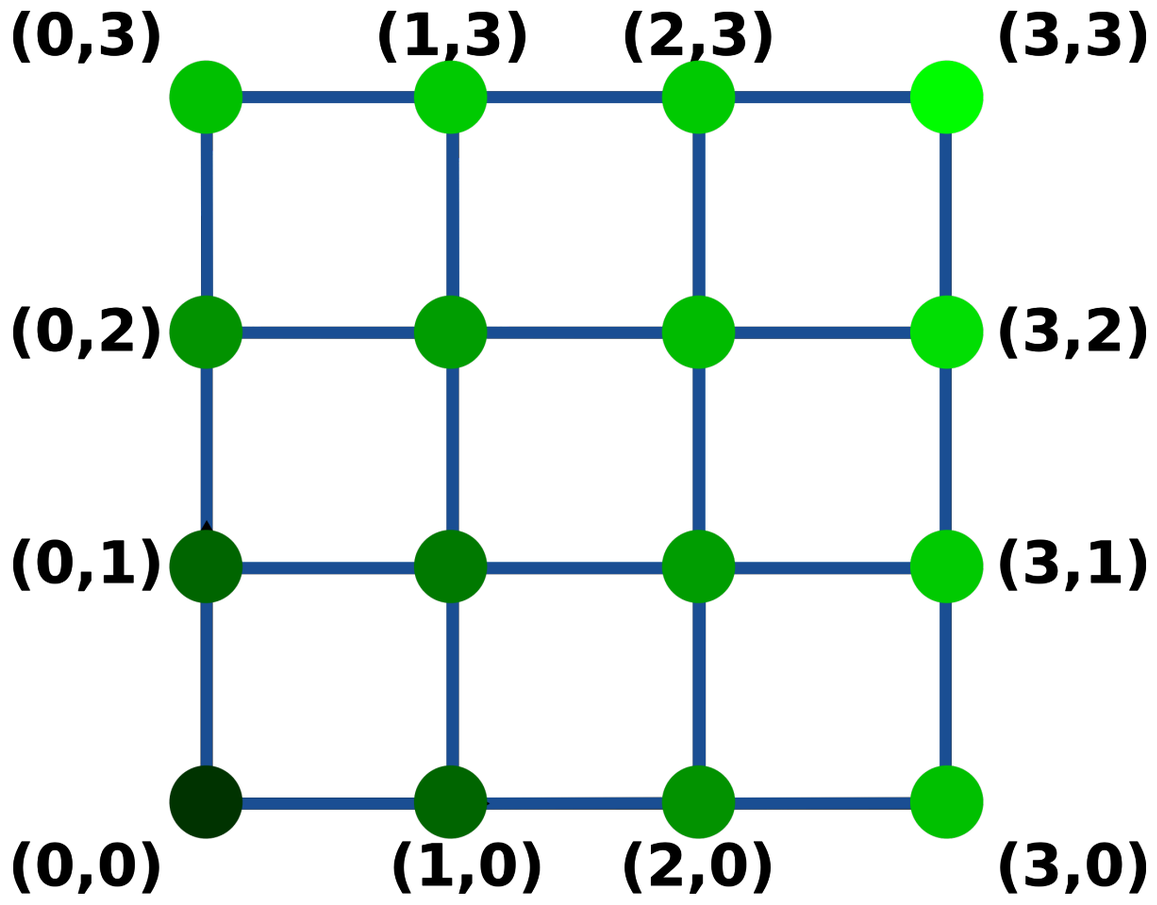


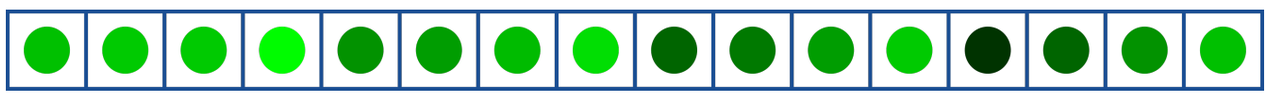
Encoding Functions Defined on Regular Grids of $\mathbb{R}^2$

- A Small Problem
  - Computer memory is one-dimensional!
- Solution
  - Order samples with a space-filling curve
  - Span one dimension at a time
  - Store metadata that describes the extent

$$f : [0,3] \times [0,3] \rightarrow \mathbb{R}$$

$$f(p) = \sqrt{p_x^2 + p_y^2}$$



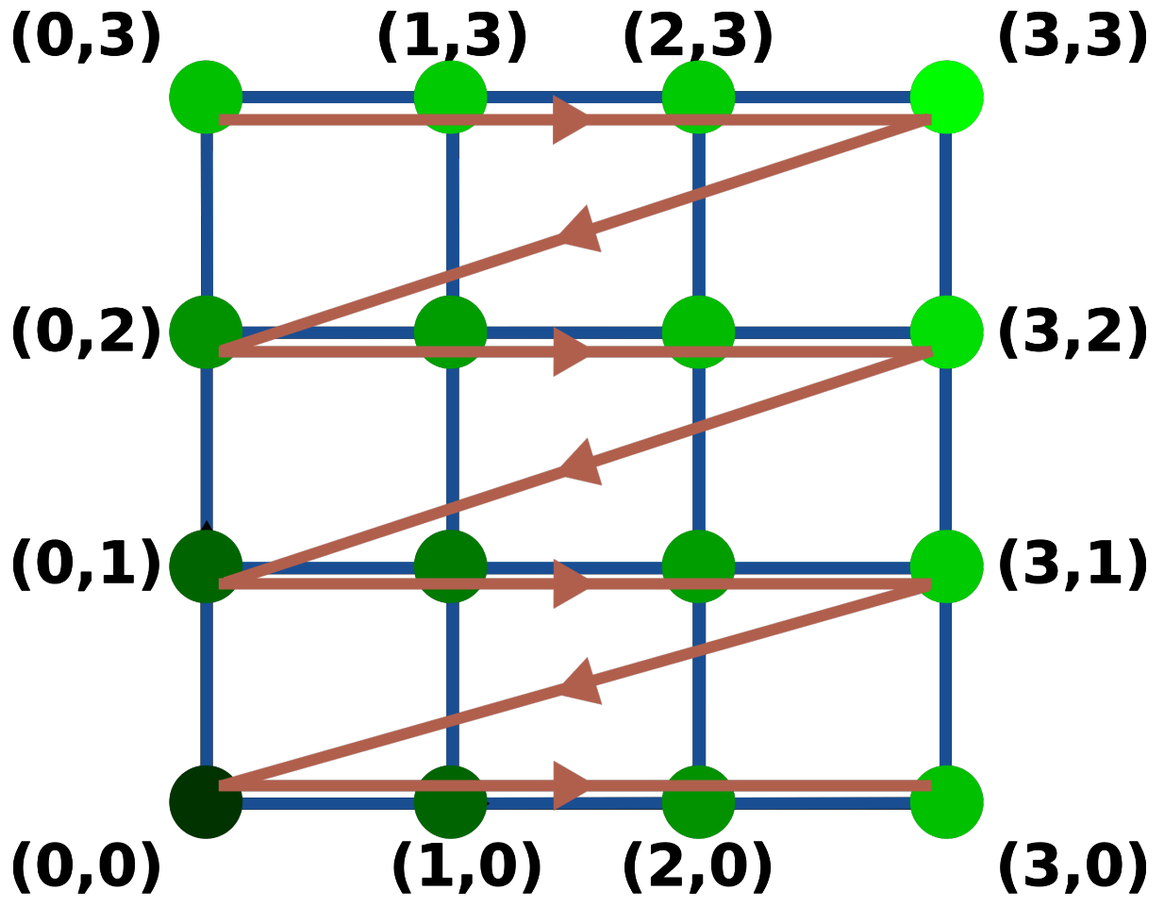


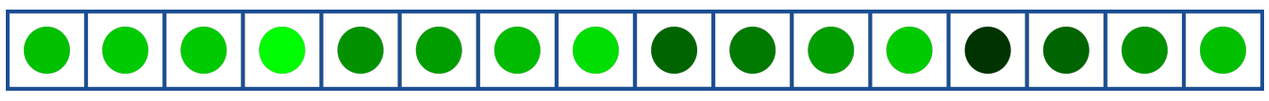
Encoding Functions Defined on Regular Grids of $\mathbb{R}^2$

- A Small Problem
  - Computer memory is one-dimensional!
- Solution
  - Order samples with a space-filling curve
  - Span one dimension at a time
  - Store metadata that describes the extent

$$f : [0,3] \times [0,3] \rightarrow \mathbb{R}$$

$$f(p) = \sqrt{p_x^2 + p_y^2}$$

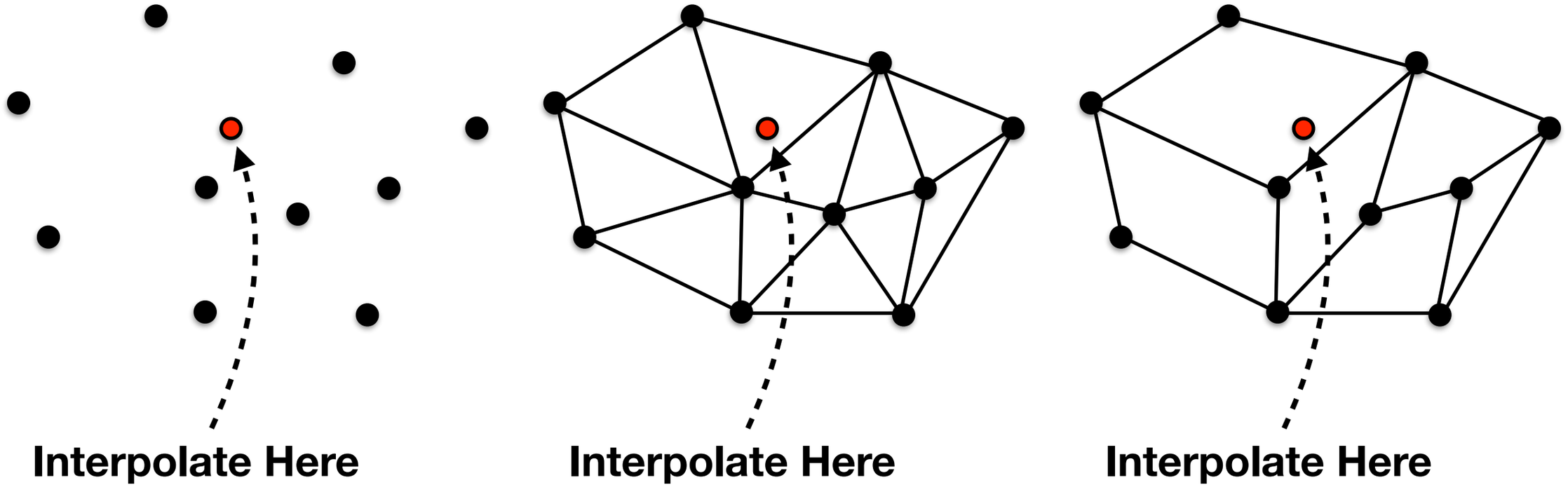




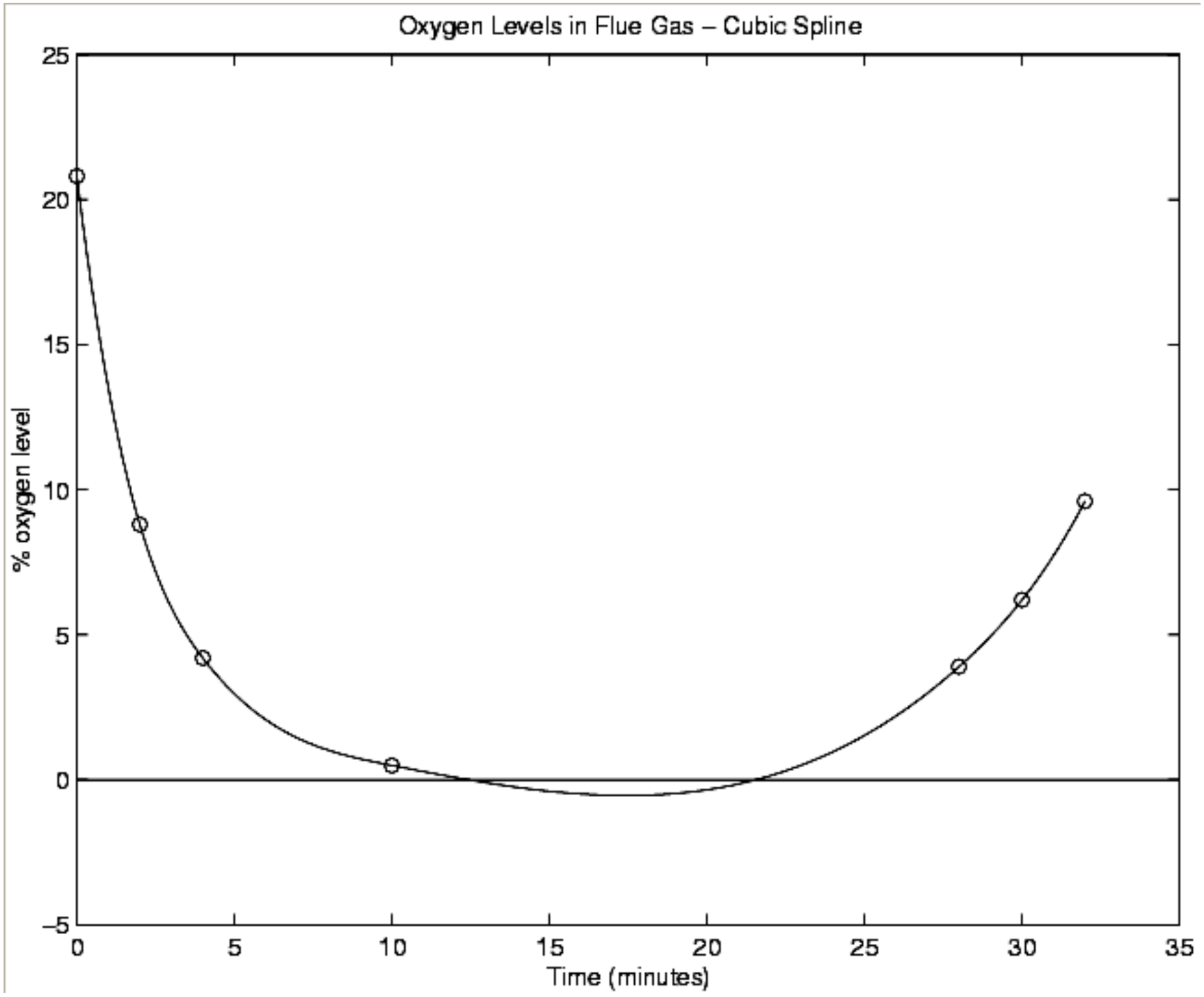
Interpolation

Mesh Choice Impacts How the Continuous Data is Interpreted

- Two key questions:
  - **Sampling**, or the choice of where attributes are measured
  - **Interpolation**, or how to model the attributes in the rest of space

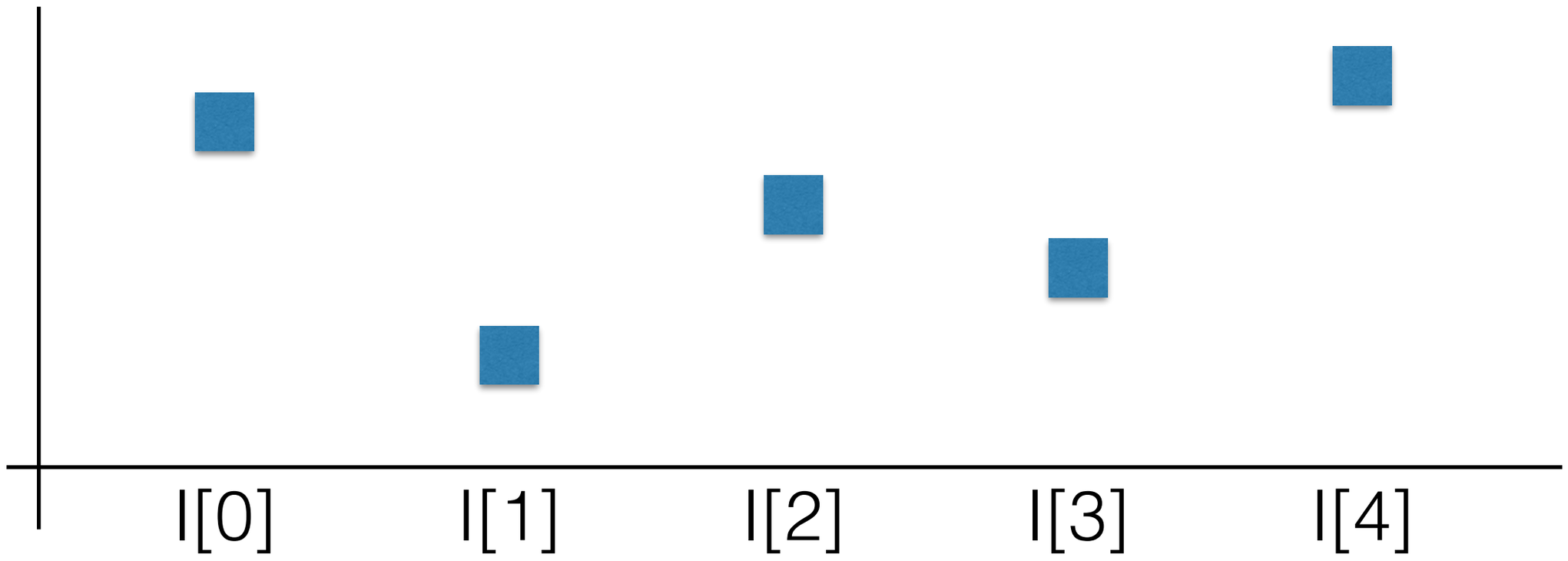


What is “Correct” Interpolation?



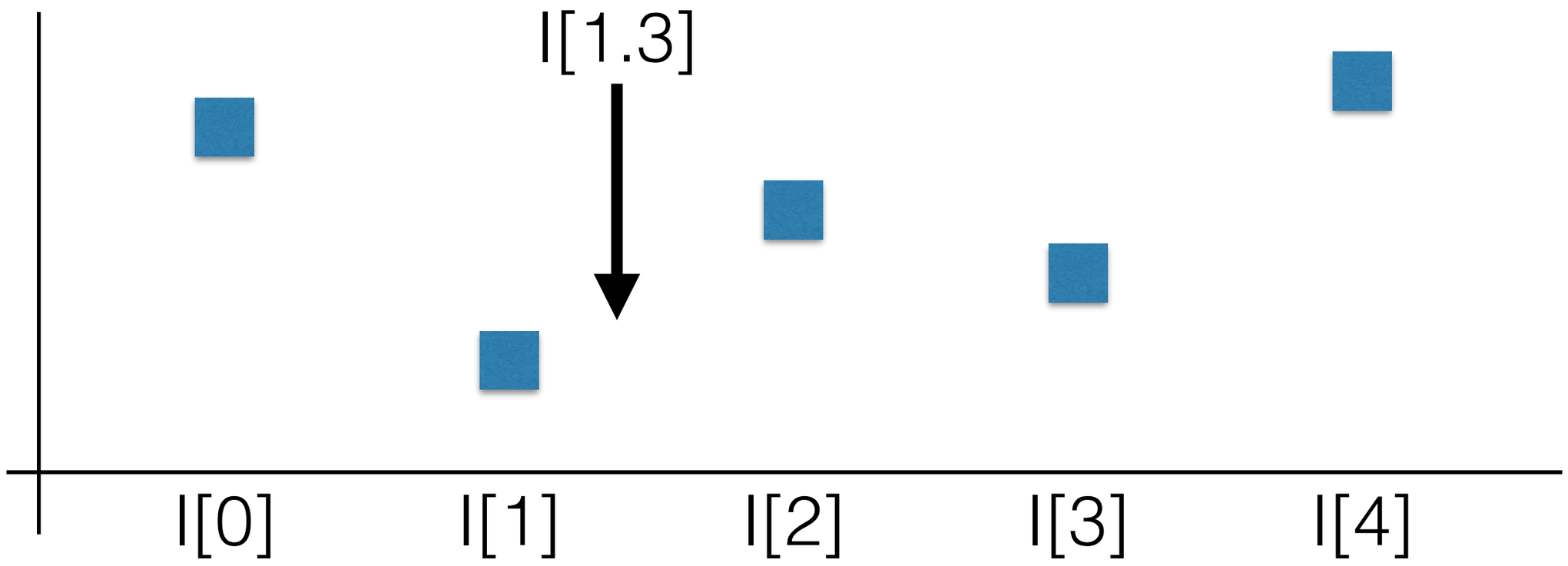
Nearest Neighbor Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?



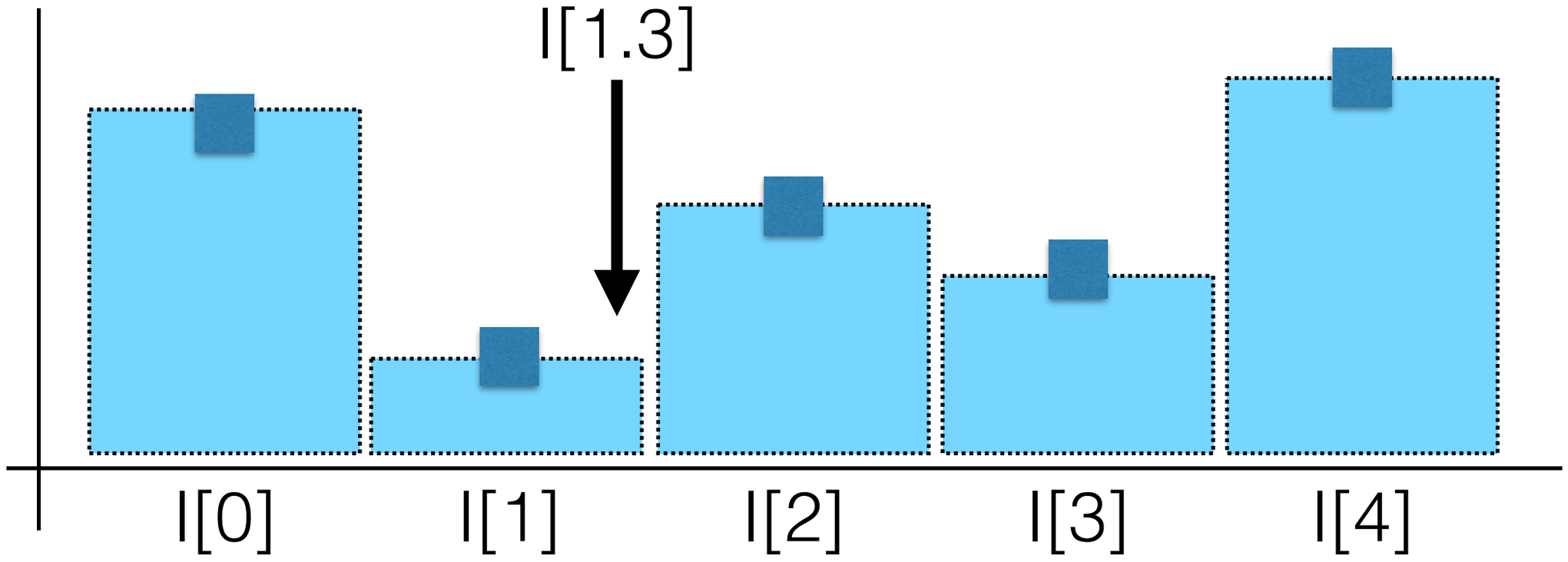
Nearest Neighbor Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?



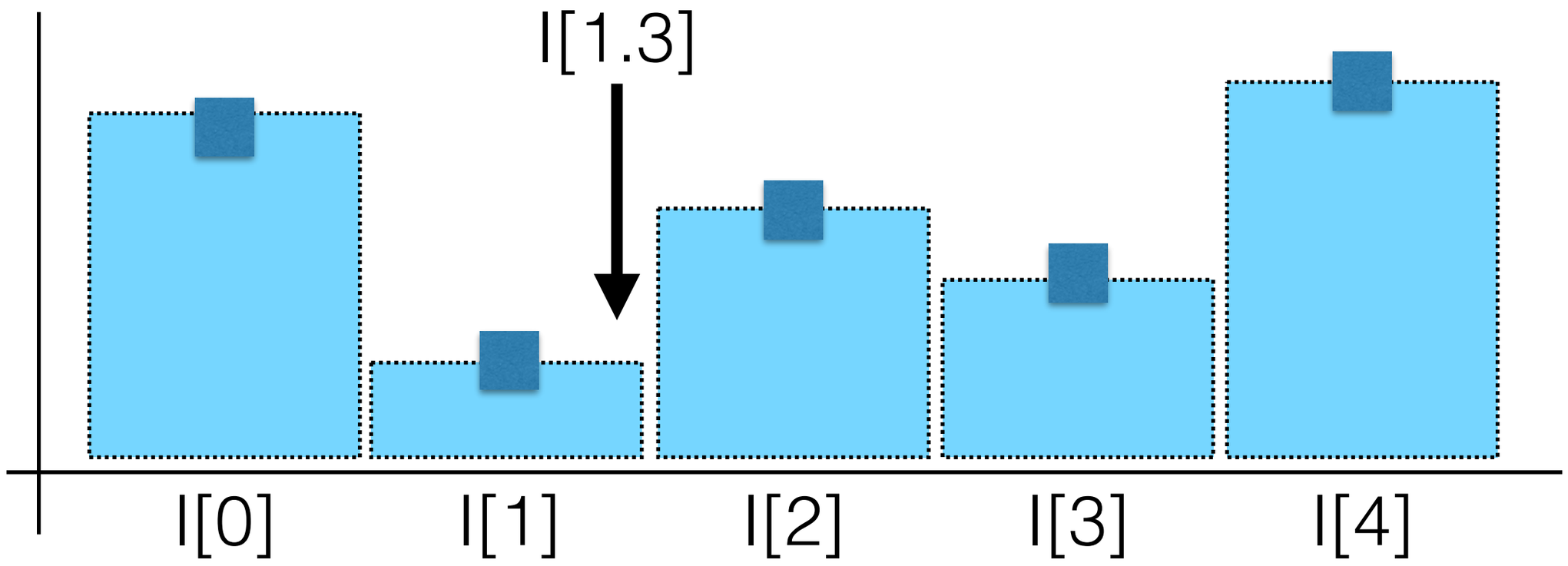
Nearest Neighbor Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?



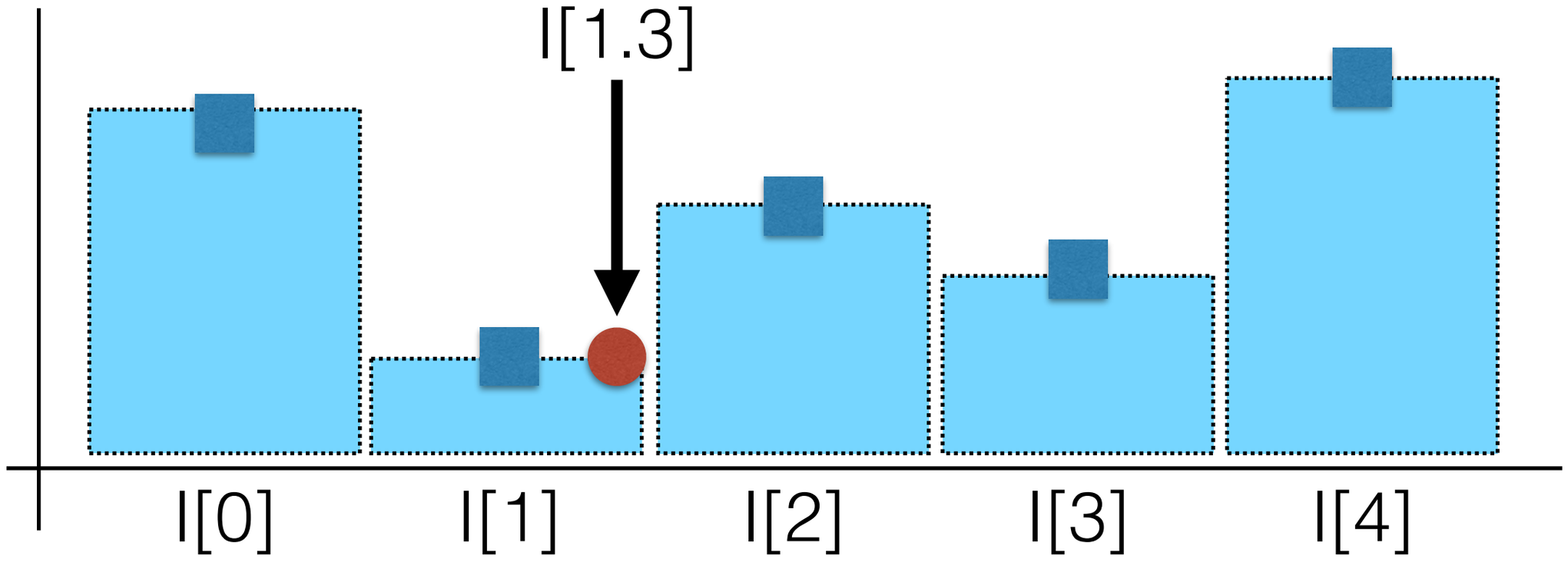
Nearest Neighbor Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?
  - I[1.3] = I[round(1.3)] = I[1]



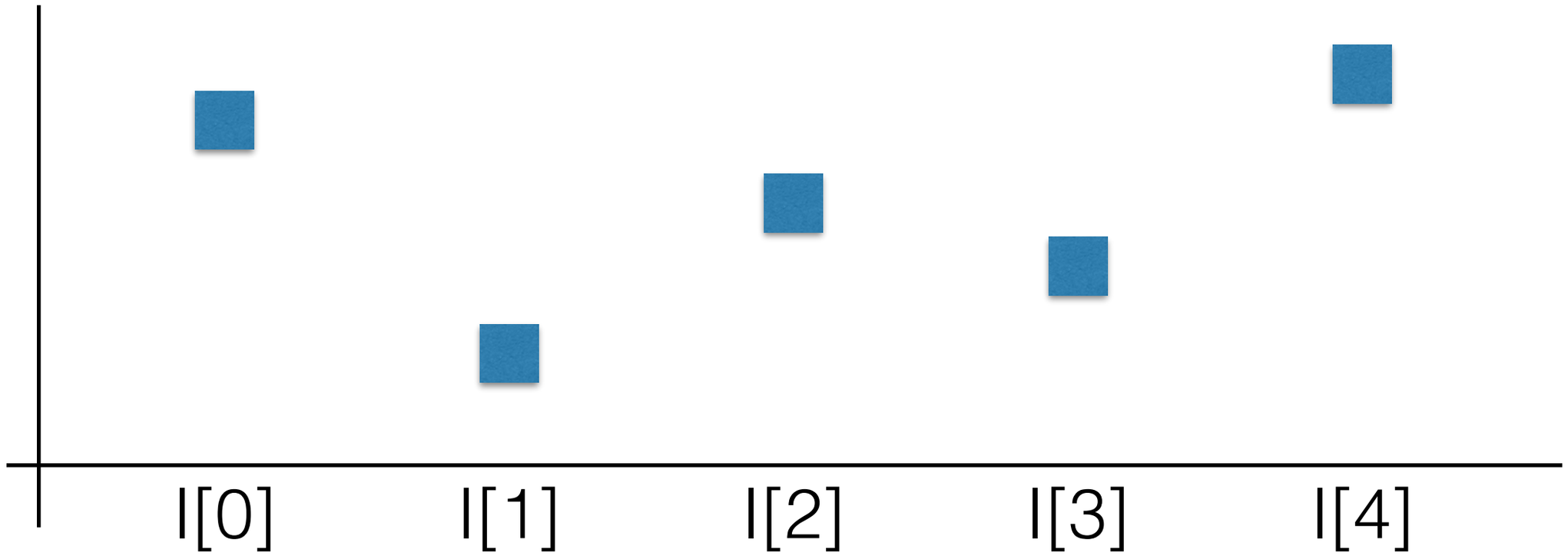
Nearest Neighbor Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?
  - I[1.3] = I[round(1.3)] = I[1]



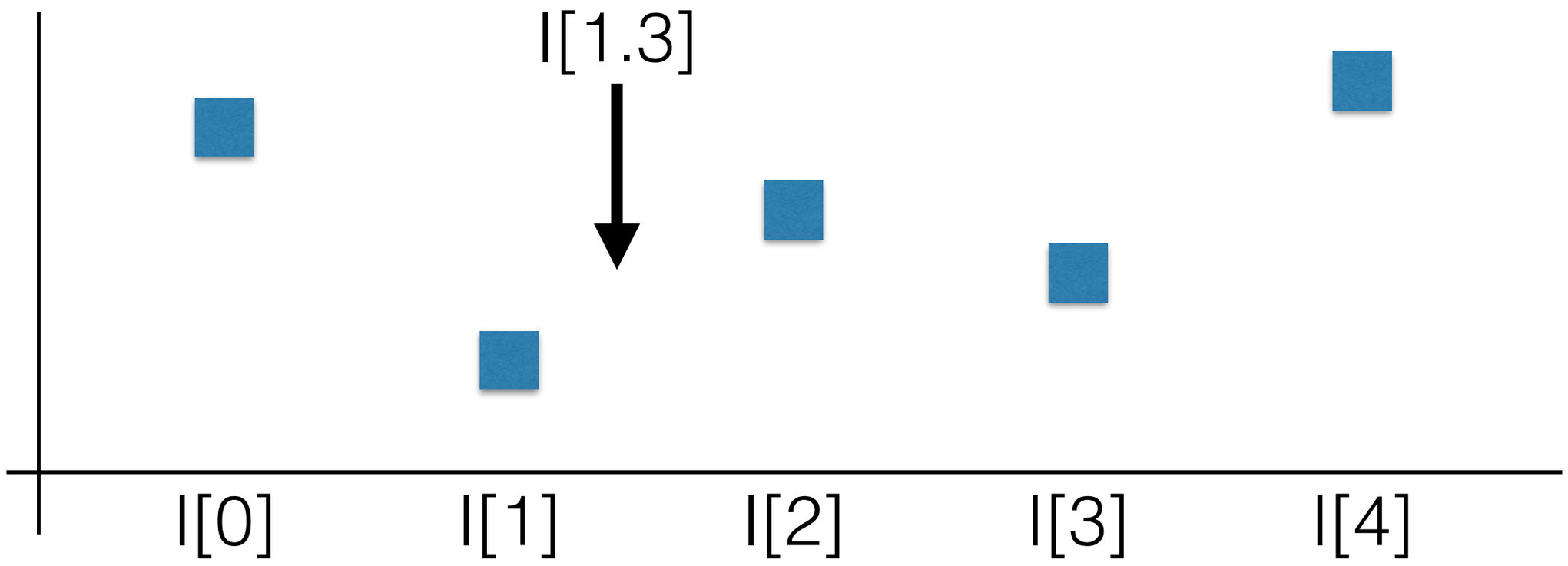
Linear Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?



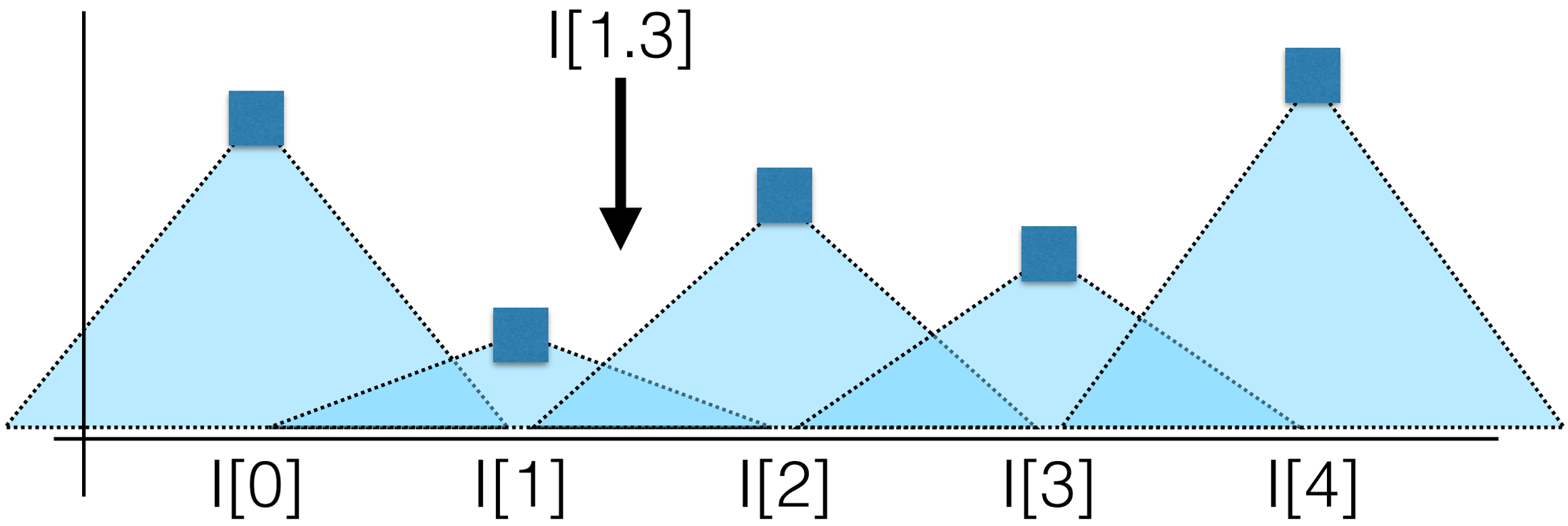
Linear Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?



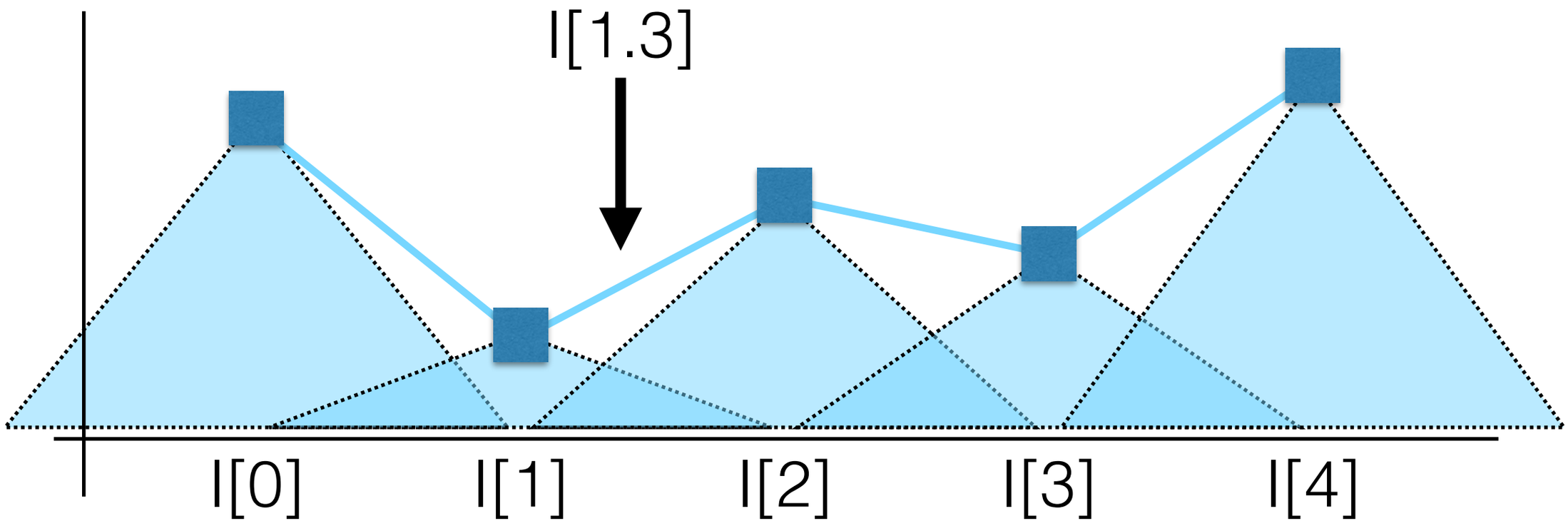
Linear Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?



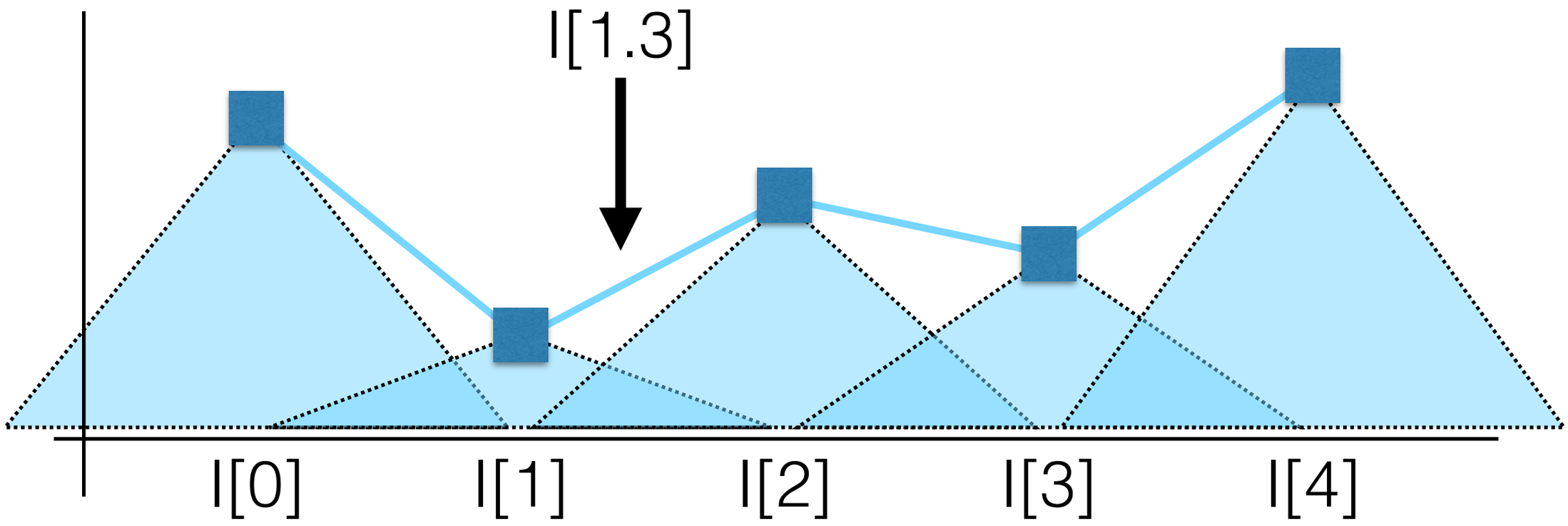
Linear Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?



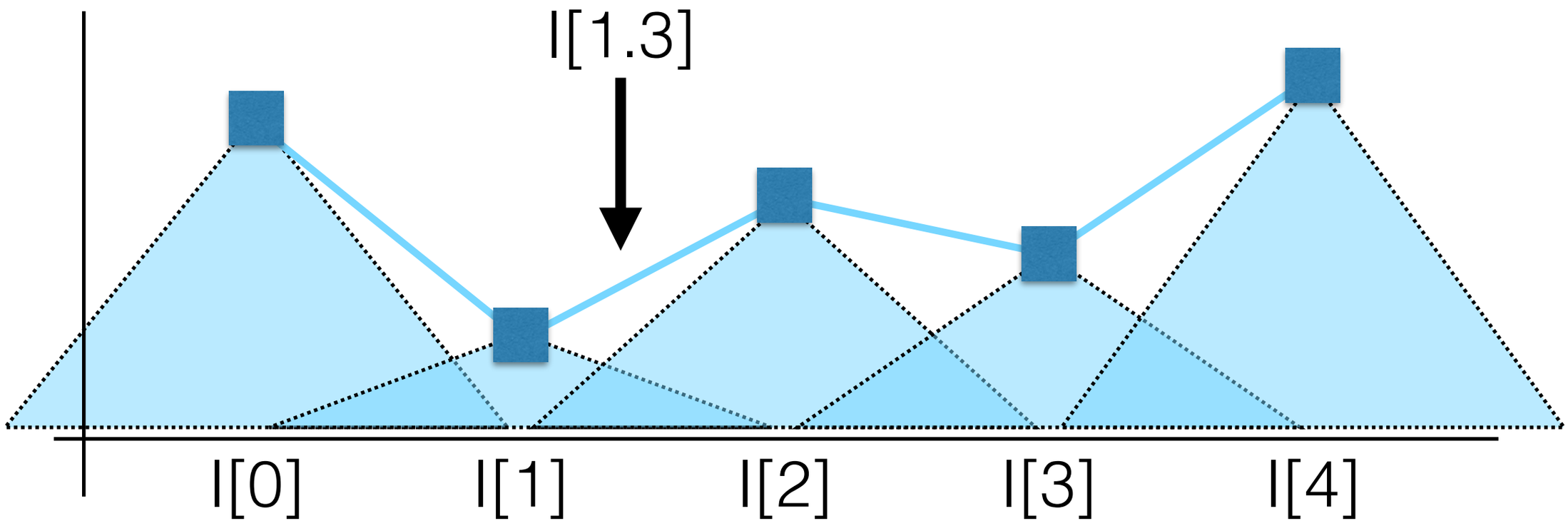
Linear Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?
  - Let s = 1.3 - round(1.3)



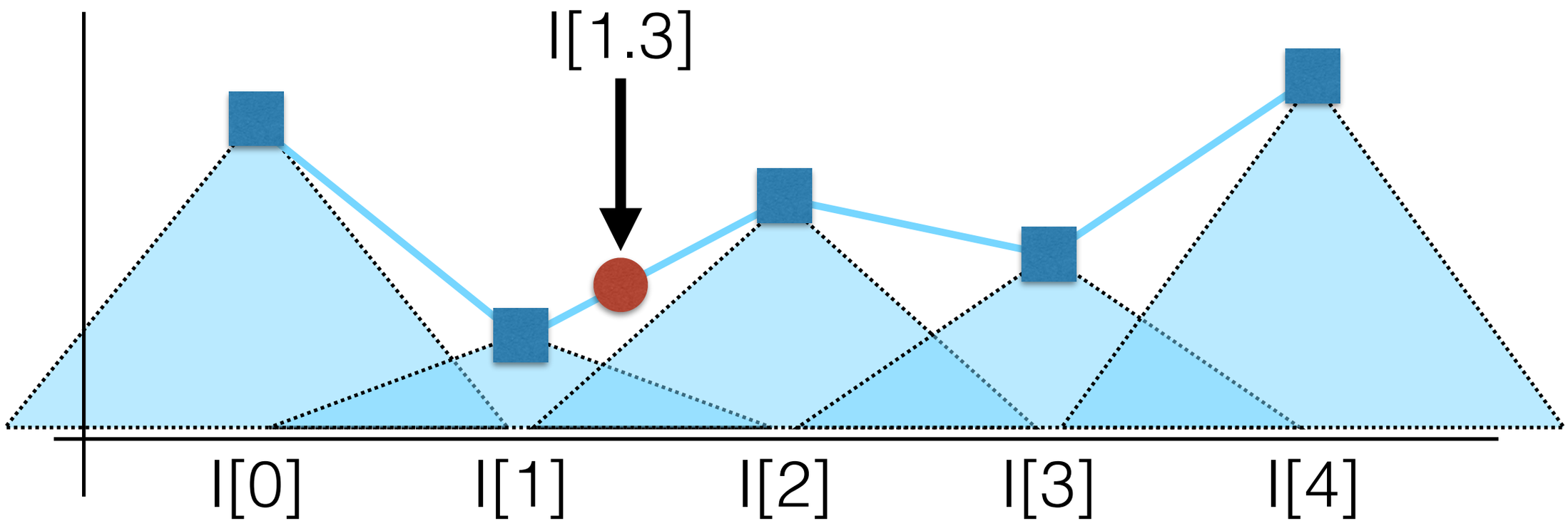
Linear Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?
  - Let s = 1.3 - round(1.3)
  - I[1.3] = 0.7*I[1] + 0.3*I[2] = (1-s)*I[1] + s*I[2]



Linear Interpolation

- Consider a 1-dimensional function sampled with five values
- What value is I[1.3] ?
  - Let s = 1.3 - round(1.3)
  - I[1.3] = 0.7*I[1] + 0.3*I[2] = (1-s)*I[1] + s*I[2]

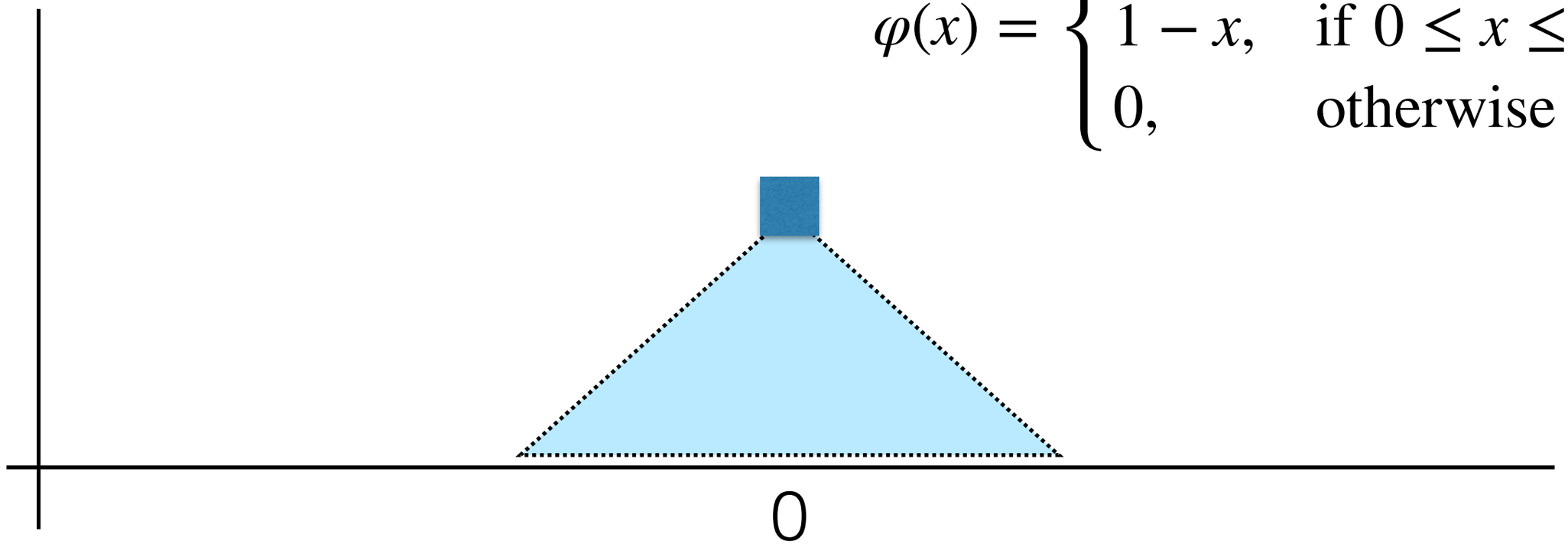


Finite-Dimensional Function Spaces

- More generally, we're building up functions from smaller units of other functions.

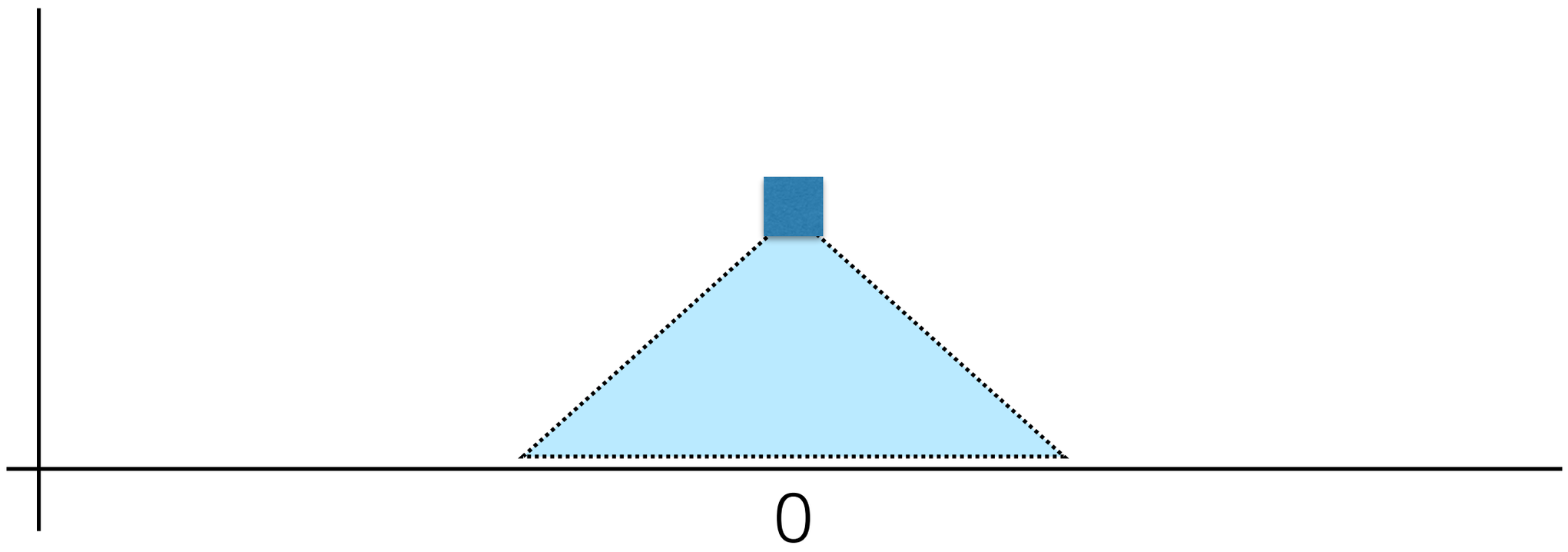
- These individual functions can be represented succinctly.

$$\varphi(x) = \begin{cases} 1 + x, & \text{if } -1 \le x \le 0 \\ 1 - x, & \text{if } 0 \le x \le 1 \\ 0, & \text{otherwise} \end{cases}$$



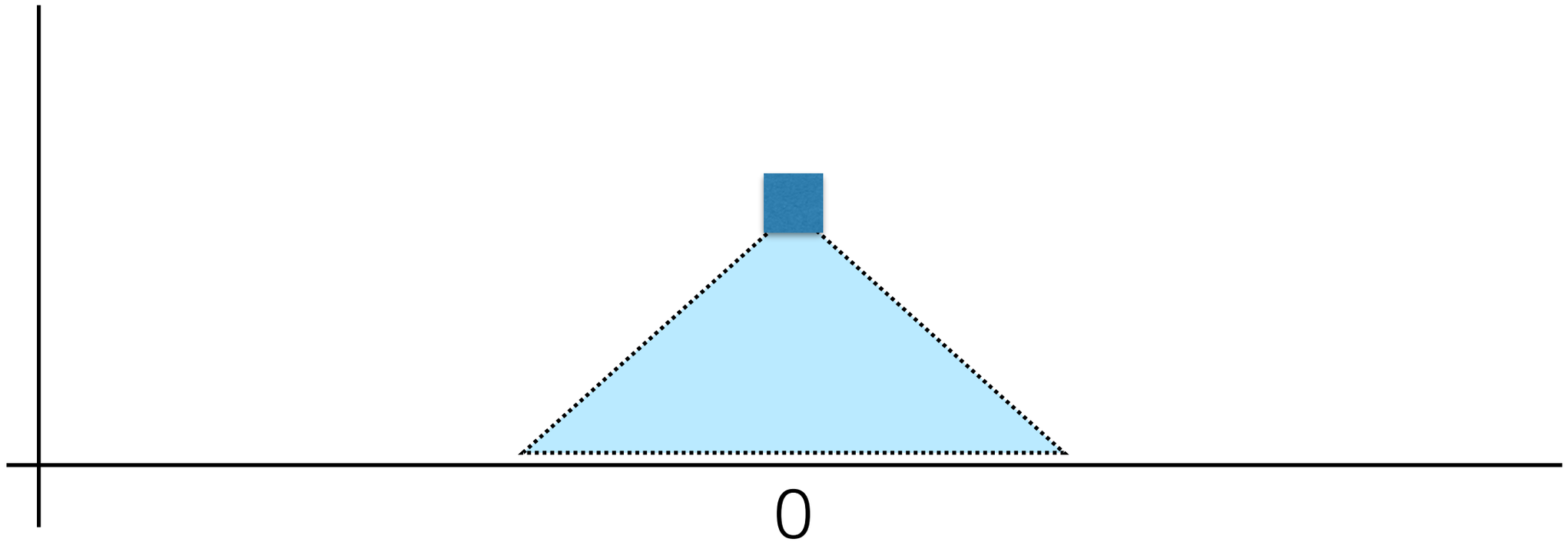
Finite-Dimensional Function Spaces

- Next, we build complicated functions by **sums** of the simple functions that **shifted** and **scaled**.



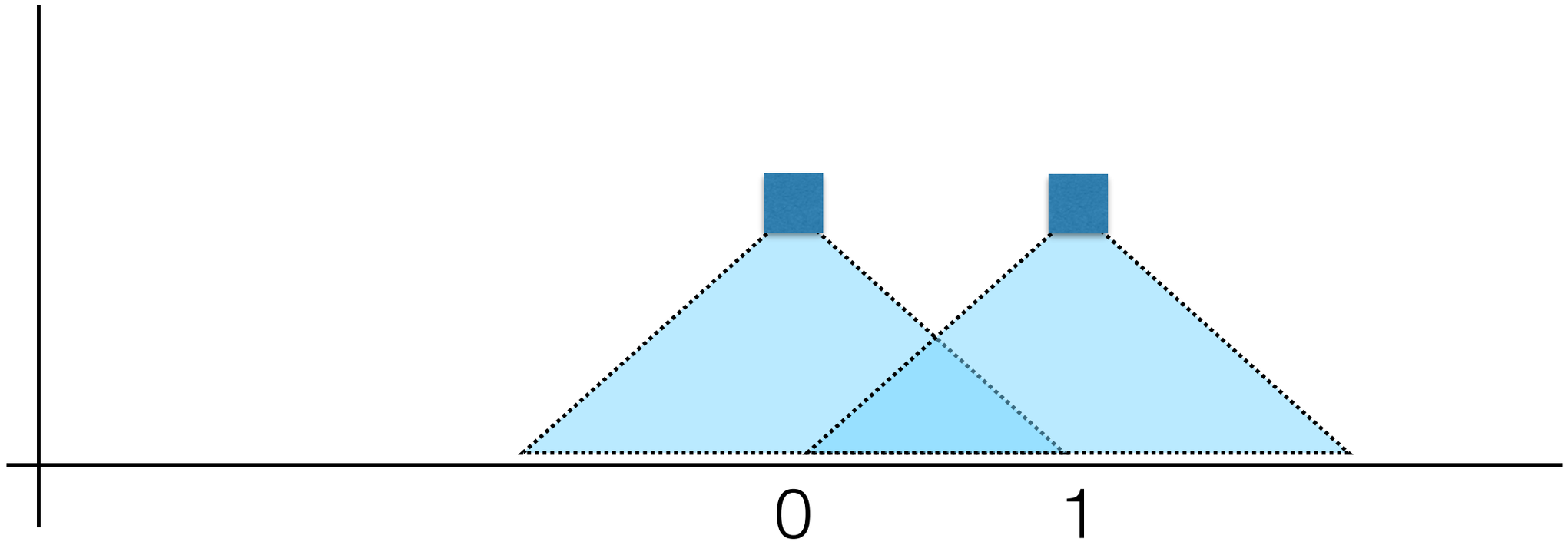
Finite-Dimensional Function Spaces

- Next, we build complicated functions by sums of the simple functions that **shifted** and scaled.



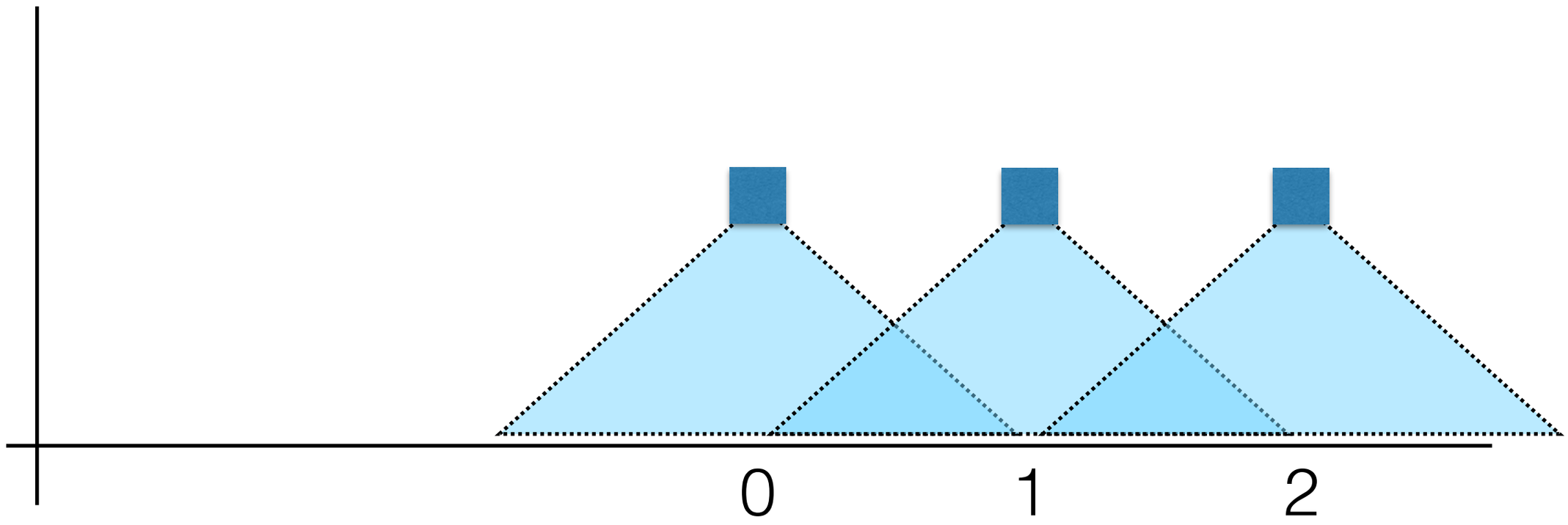
Finite-Dimensional Function Spaces

- Next, we build complicated functions by sums of the simple functions that **shifted** and scaled.



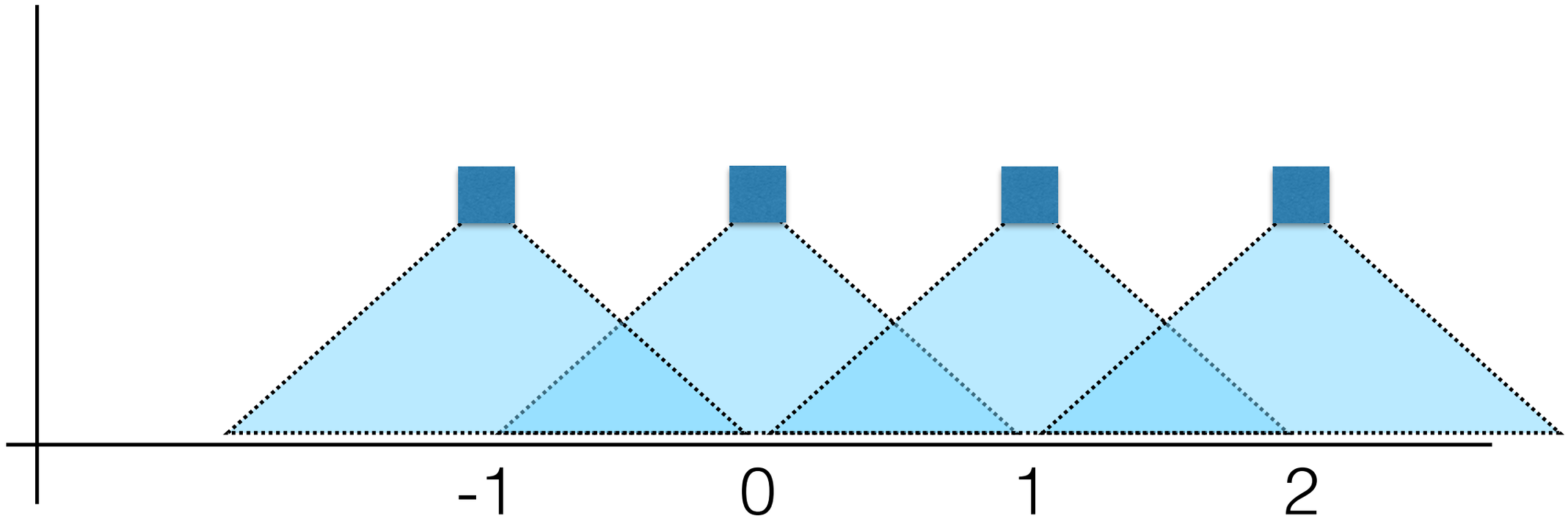
Finite-Dimensional Function Spaces

- Next, we build complicated functions by sums of the simple functions that **shifted** and scaled.



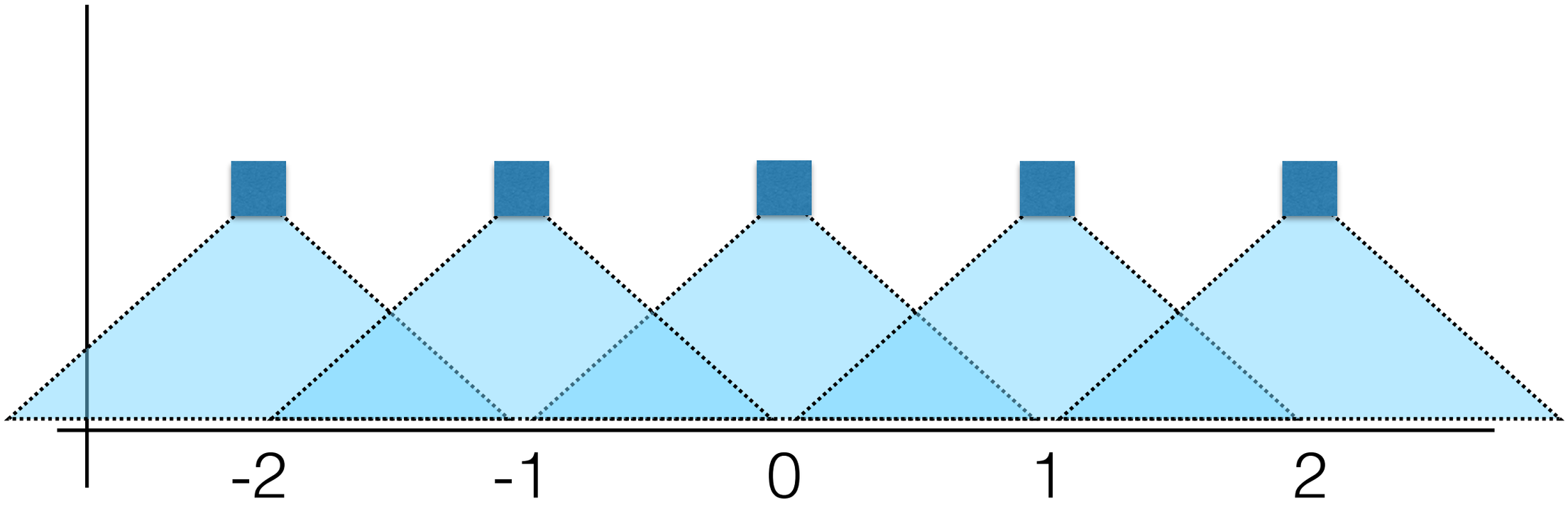
Finite-Dimensional Function Spaces

- Next, we build complicated functions by sums of the simple functions that **shifted** and scaled.



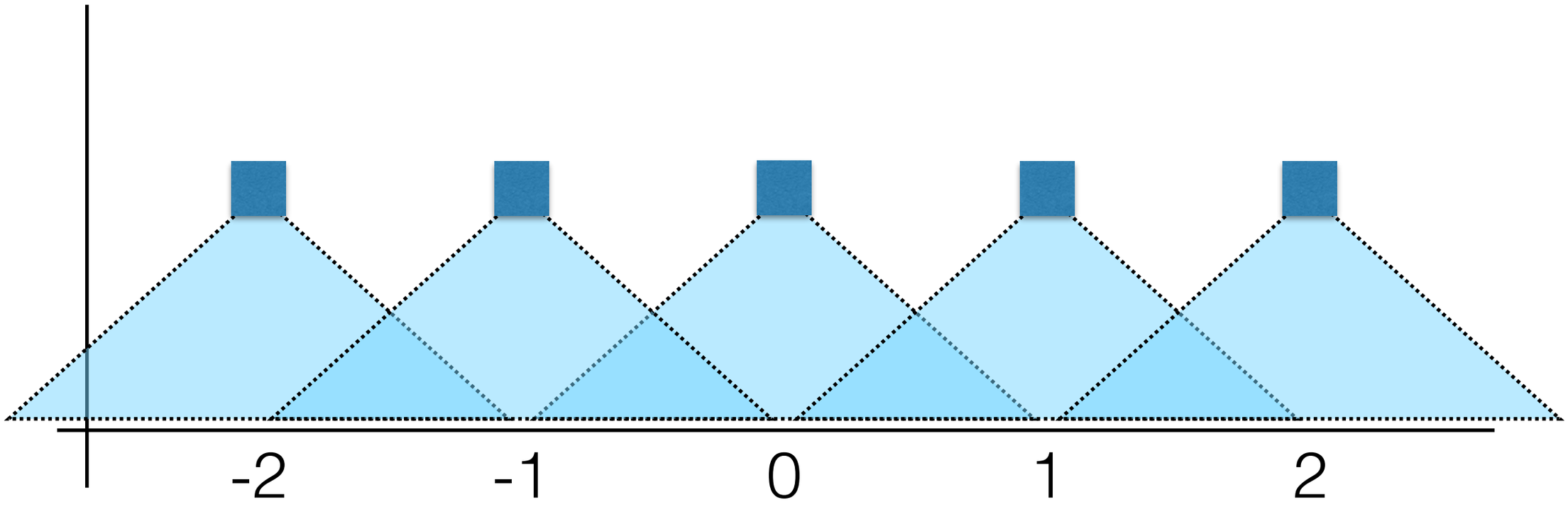
Finite-Dimensional Function Spaces

- Next, we build complicated functions by sums of the simple functions that **shifted** and scaled.



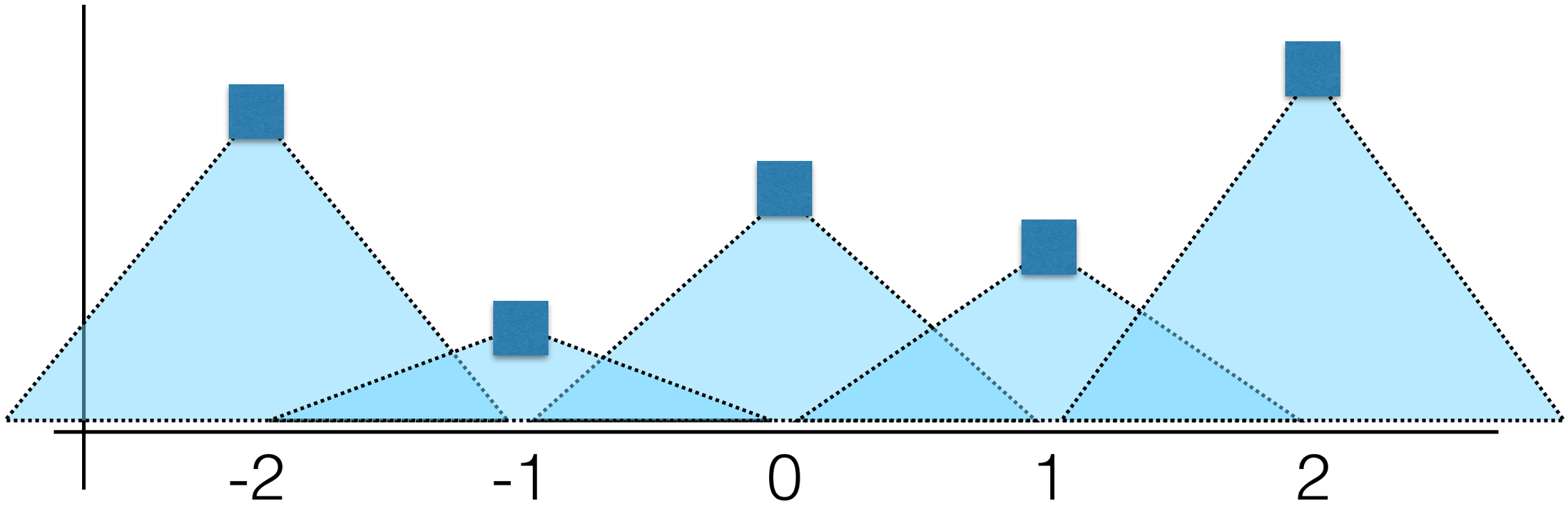
Finite-Dimensional Function Spaces

- Next, we build complicated functions by sums of the simple functions that shifted and **scaled**.



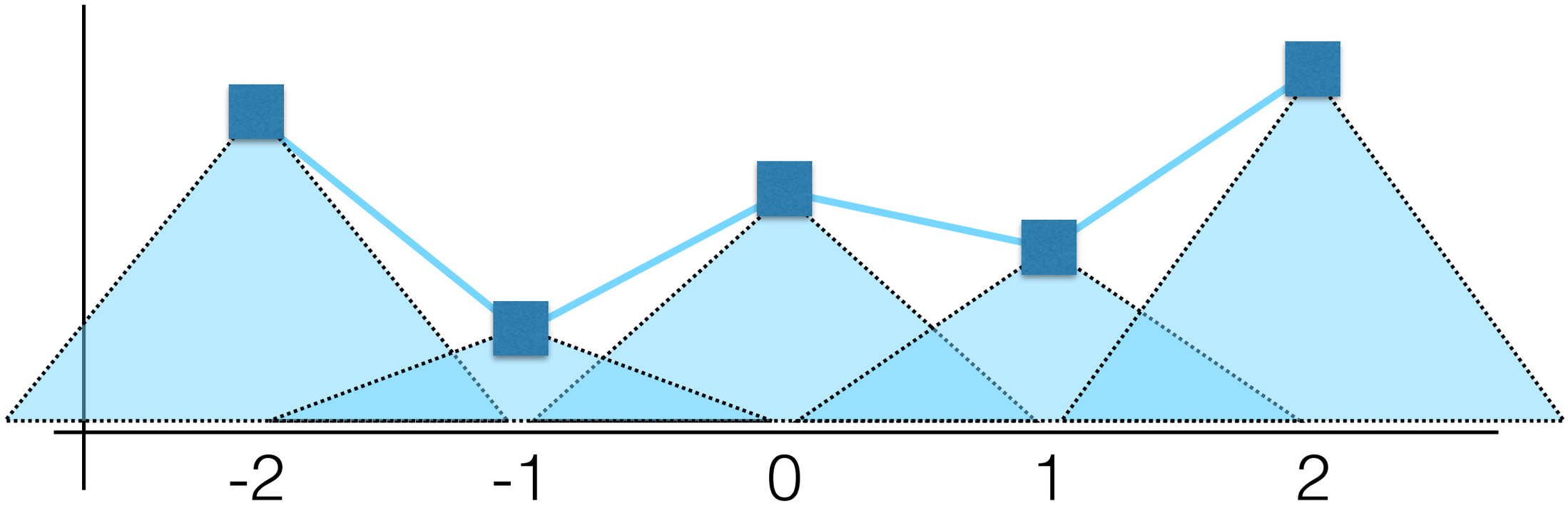
Finite-Dimensional Function Spaces

- Next, we build complicated functions by sums of the simple functions that shifted and **scaled**.

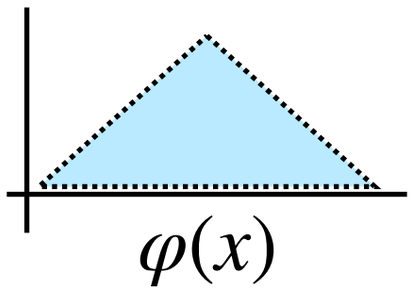


Finite-Dimensional Function Spaces

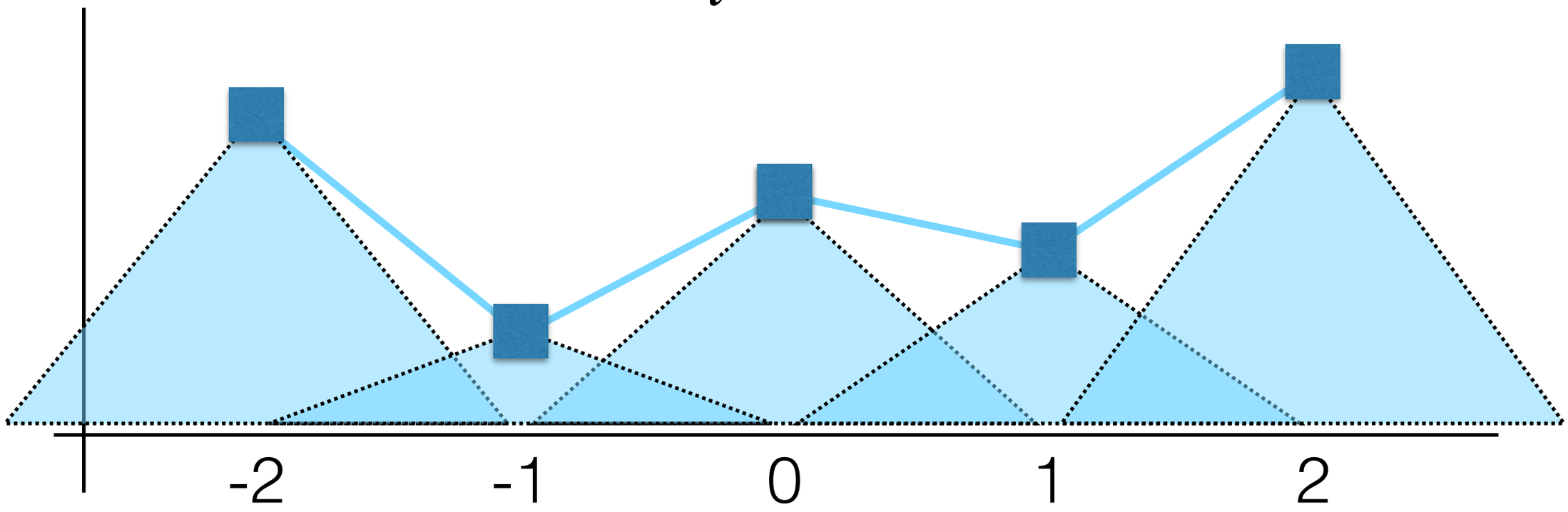
- Next, we build complicated functions by **sums** of the simple functions that shifted and scaled.



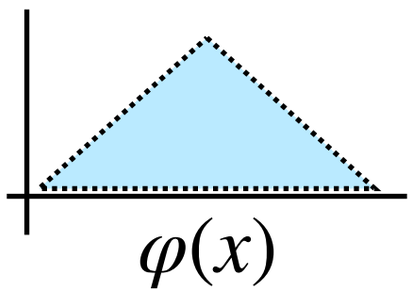
Finite-Dimensional Function Spaces



$$f(x) = \sum_i \varphi(x - i)c_i$$



Finite-Dimensional Function Spaces



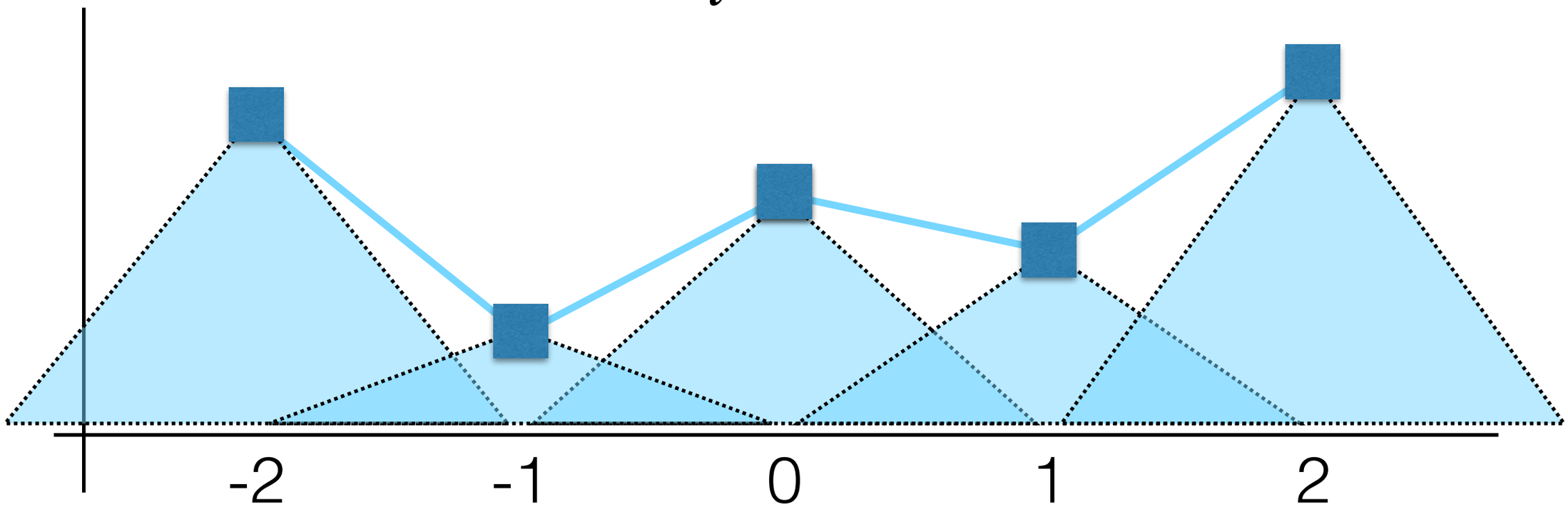
sums

simple functions

shifts

scales

$$f(x) = \sum_i \varphi(x - i)c_i$$



Why Go Through This Trouble?

- Why not just define these functions “procedurally”?
    - The function we’re computing is just arrays and if statements, there are multiple ways to implement.
- Reason: Because we can compute math on those sums more easily, e.g. derivatives (more on this next week).

Linear Interpolation In Code

- In one-dimension, this boils down to building a many functions f(t) such that:

```
function f(t) {
  return a * (1 - t) + b * t;
}
```

- For t between [0,1] this returns values between [a,b] alone the line connecting them.

- By rearranging terms, we can think of this as an equation that is linear in the variable t: `(b - a) * t + a`

Useful Links / Demo

- D3's interpolation routines:
  https://github.com/d3/d3-interpolate

- D3's use of interpolation for generating curves:
  https://github.com/d3/d3-shape#lines

Interpolation on Other Domains

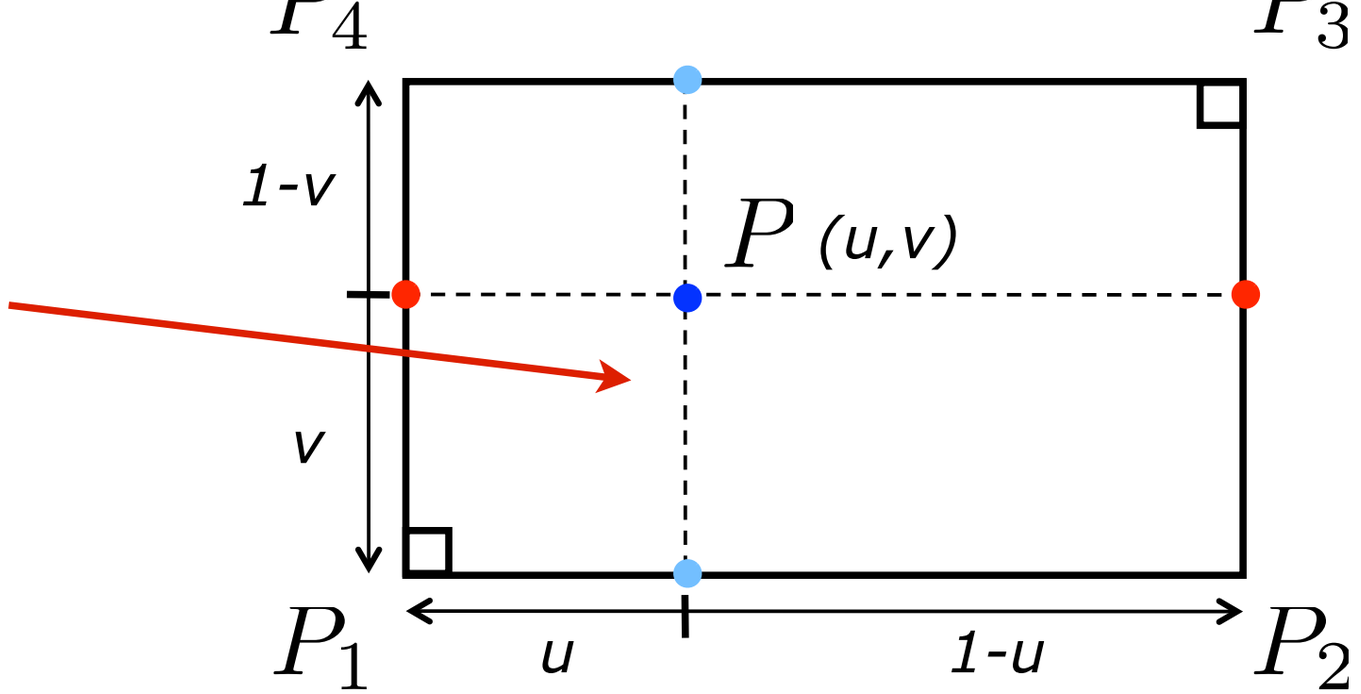
Bilinear Interpolation

- Bilinear interpolation

$$\phi(x, y) = axy + bx + cy + d$$

$$\forall i, \phi(P_i) = \phi_i$$

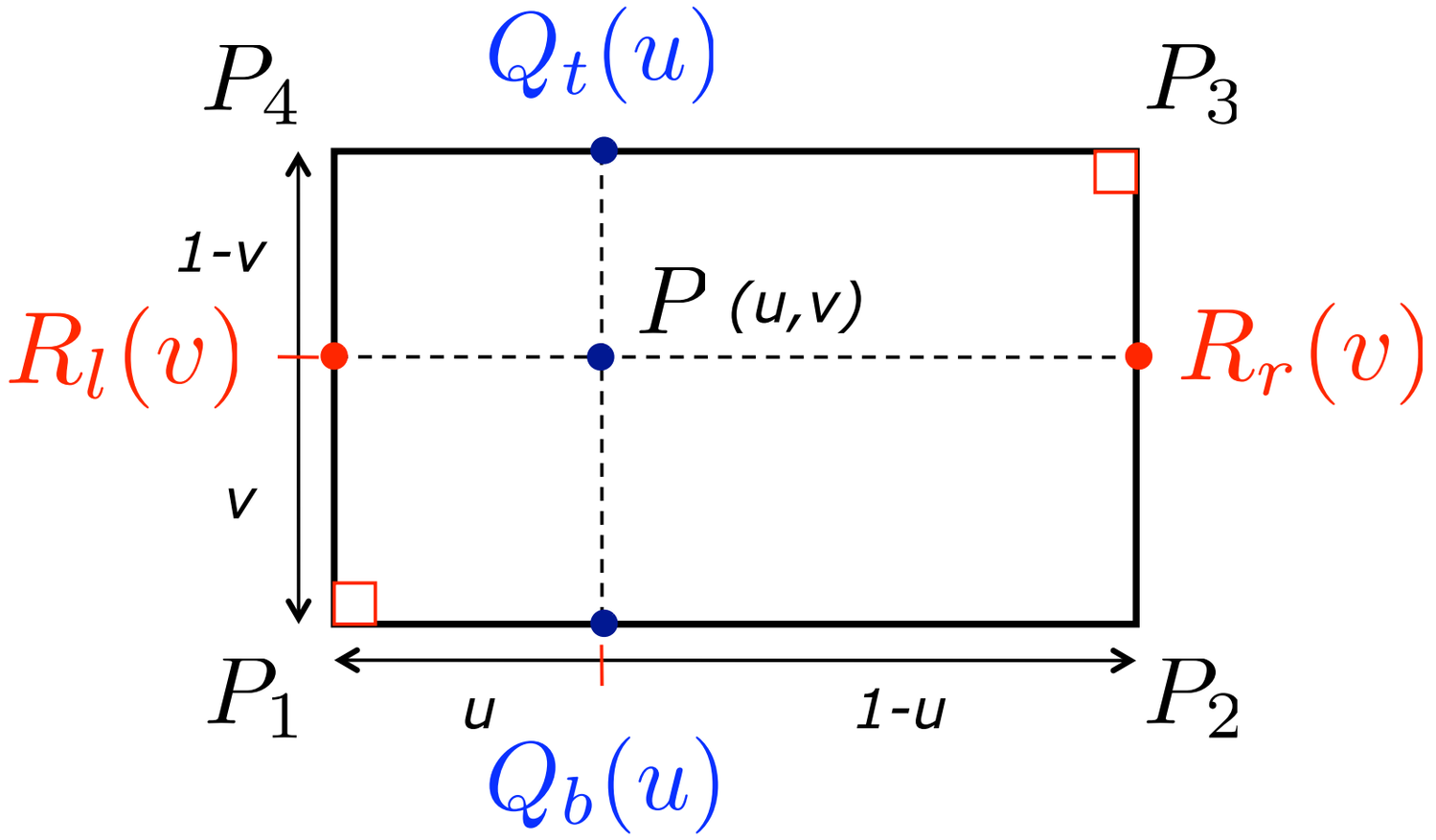
Combination of two consecutive linear interpolation



Bilinear Interpolation

- In rectangle

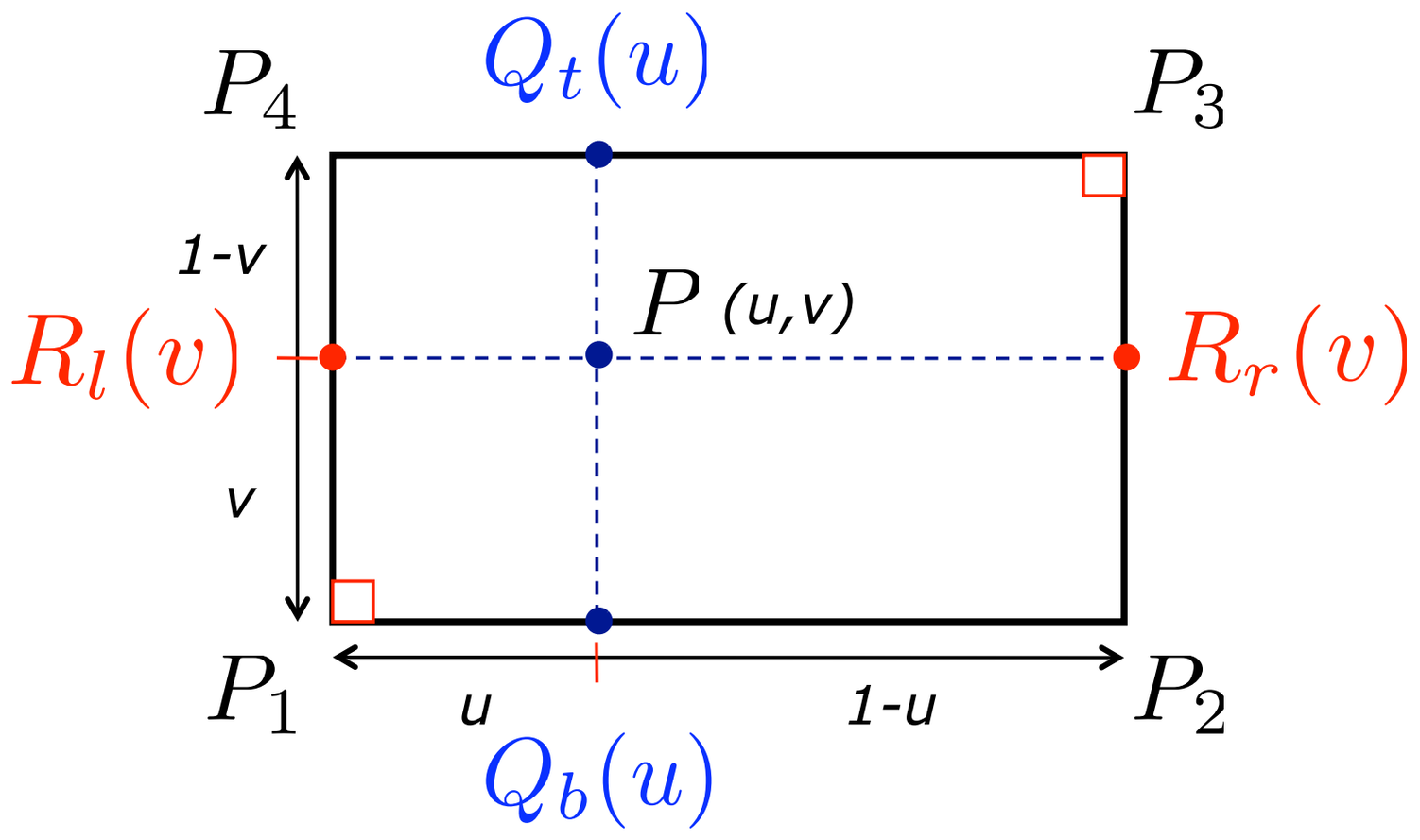
$$P = (1-v)Q_b(u) + vQ_t(u)$$
$$= (1-u)R_l(v) + uR_r(v)$$



Bilinear Interpolation

- In rectangle

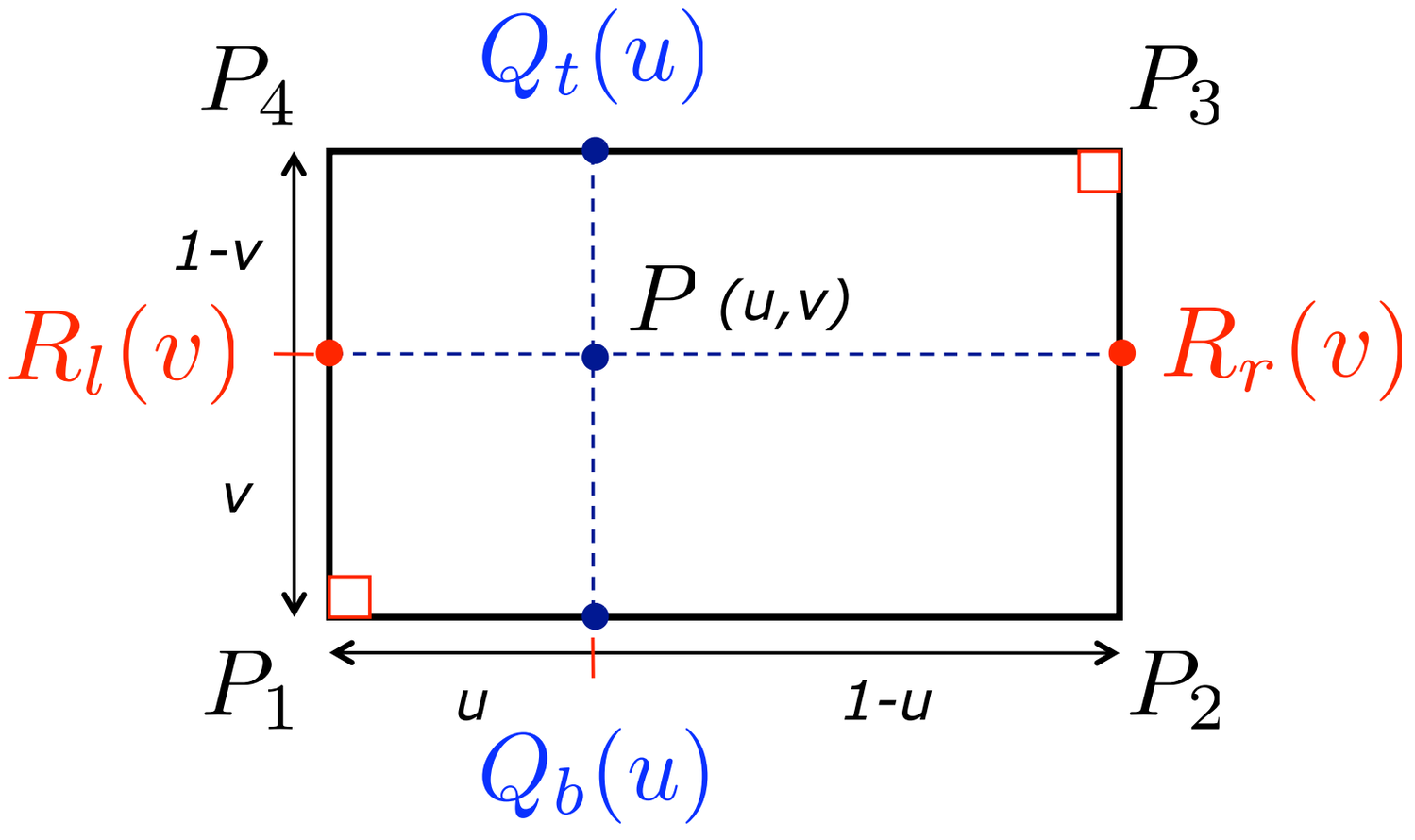
$$P = P_1 + u(P_2 - P_1) + v(P_4 - P_1) + uv(P_1 - P_2 + P_3 - P_4)$$



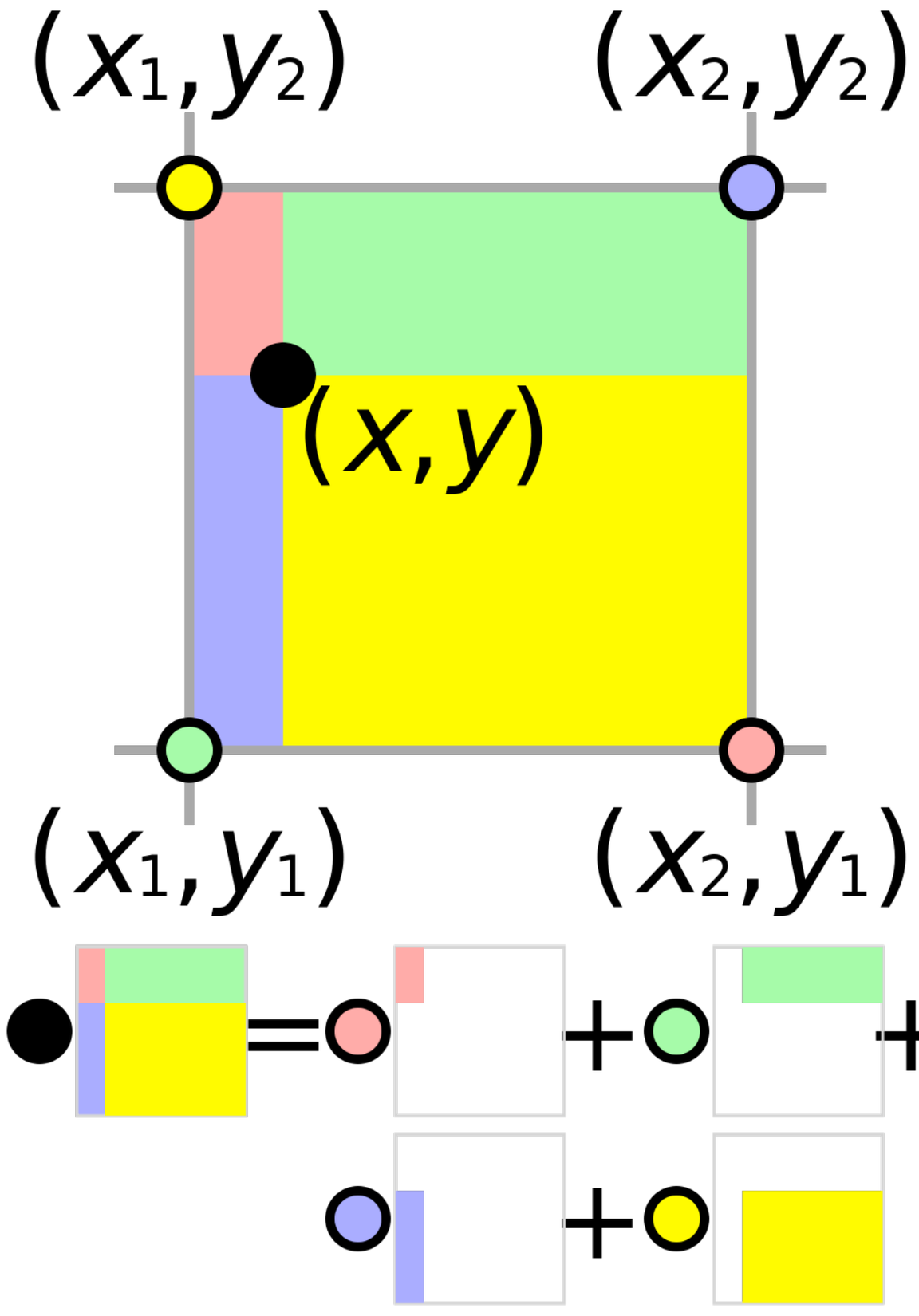
Bilinear Interpolation

- In rectangle

$$\phi(P) = \phi_1 + u(\phi_2 - \phi_1) + v(\phi_4 - \phi_1) + uv(\phi_1 - \phi_2 + \phi_3 - \phi_4)$$



Bilinear Interpolation



- Alternate interpretation is a weighted sum of the four pixel values

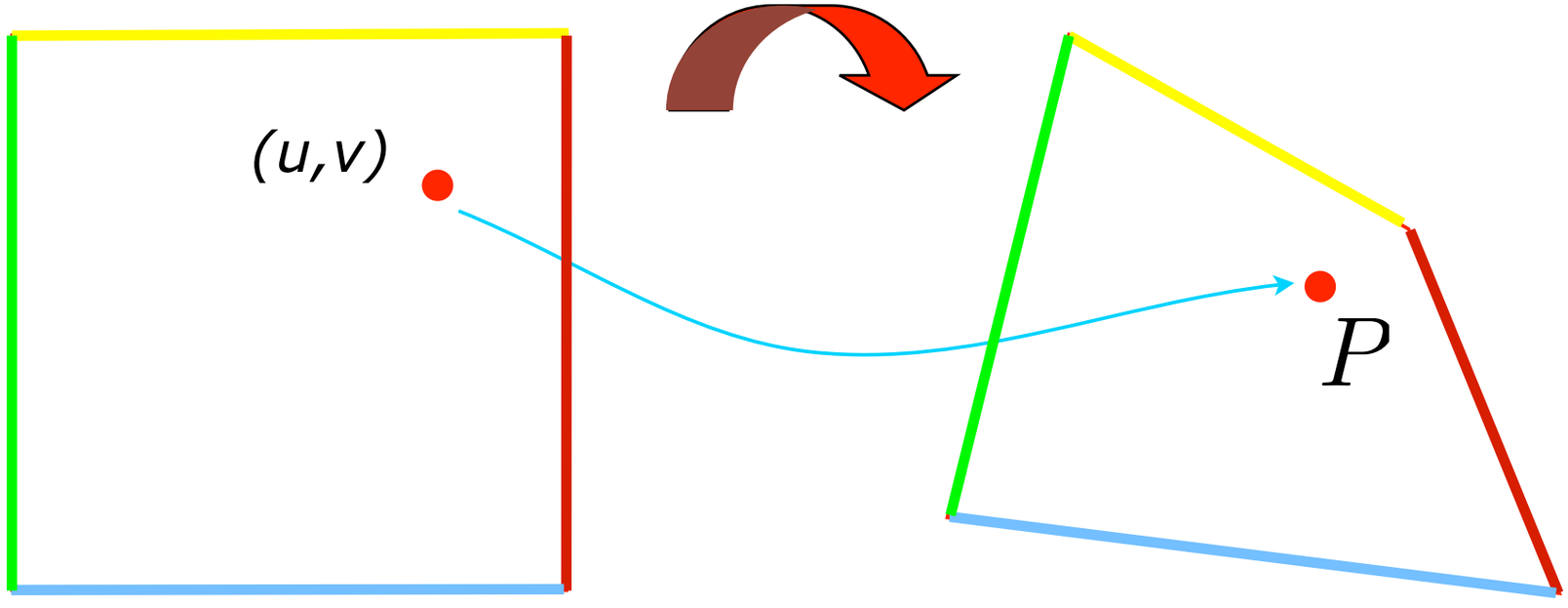
- Weights defined by the area opposite each corner

http://en.wikipedia.org/wiki/Bilinear_interpolation

Bilinear Interpolation

- In arbitrary quadrilateral

  i. Start from (u,v) coordinates (if available) in computational space and apply previous formula



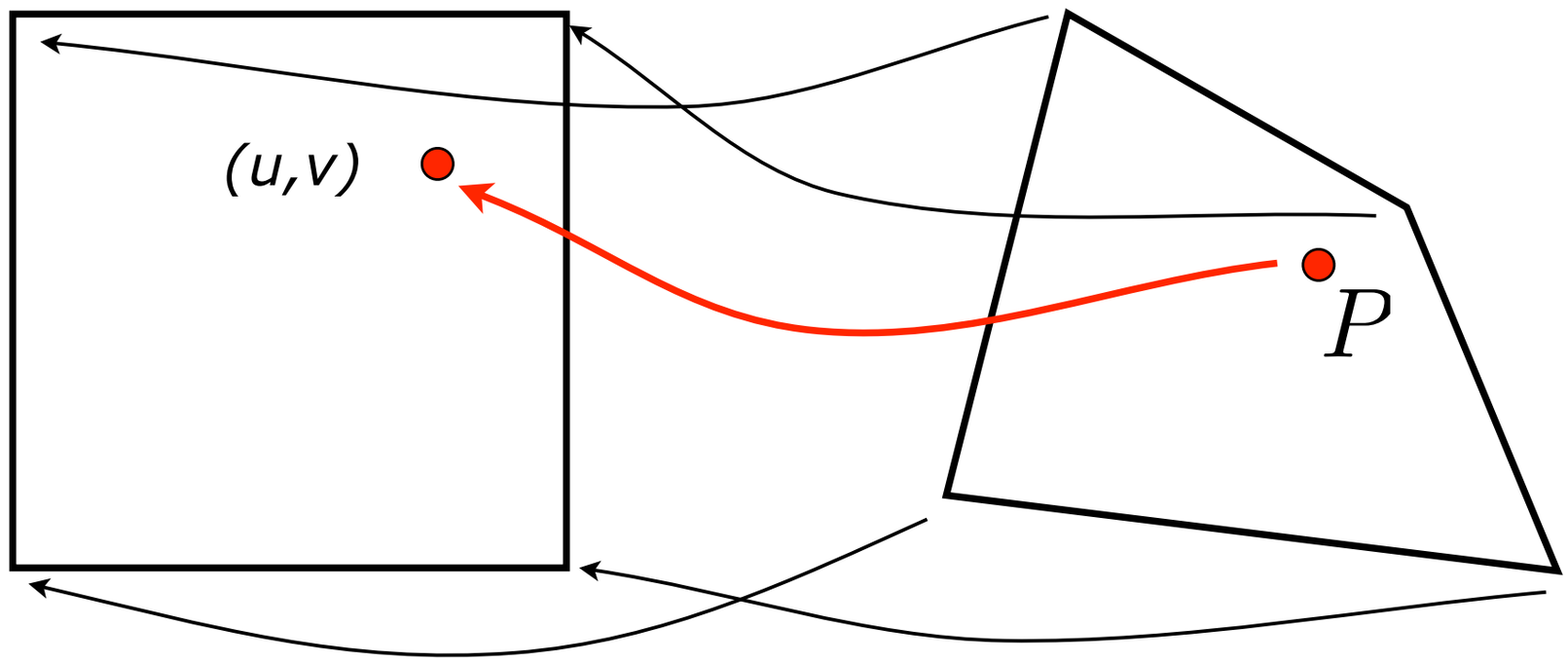
Important: axis-parallel straight lines are mapped to straight lines

Bilinear Interpolation

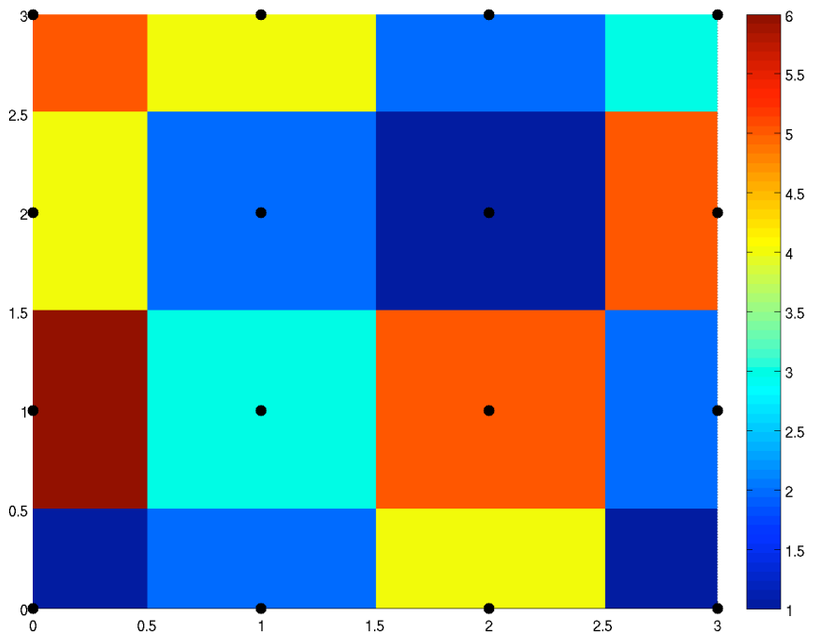
- In arbitrary quadrilateral

  ii. Otherwise

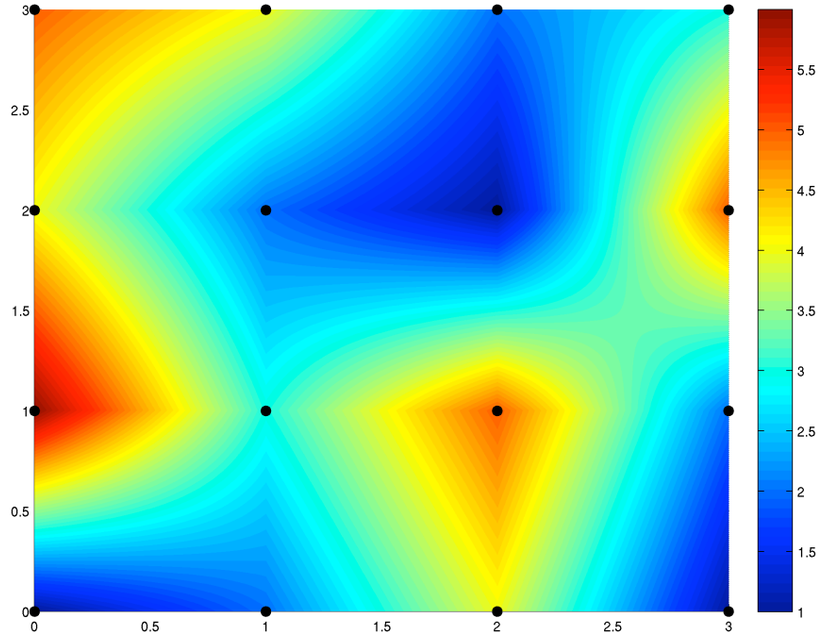
    1. Compute (inverse) mapping from physical space to computational space (unit square): quadratic eqns
    2. goto i.



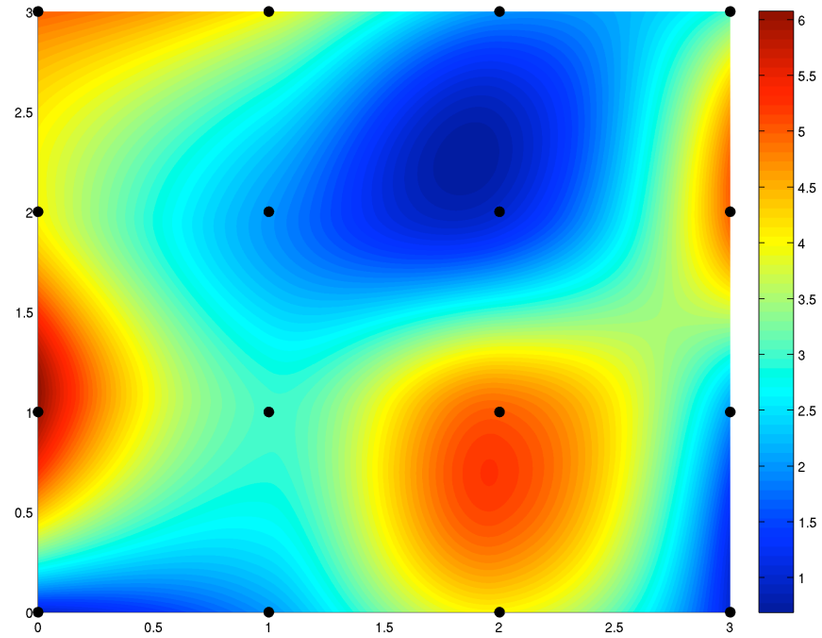
Interpolation



Nearest Neighbor



Bilinear



Bicubic

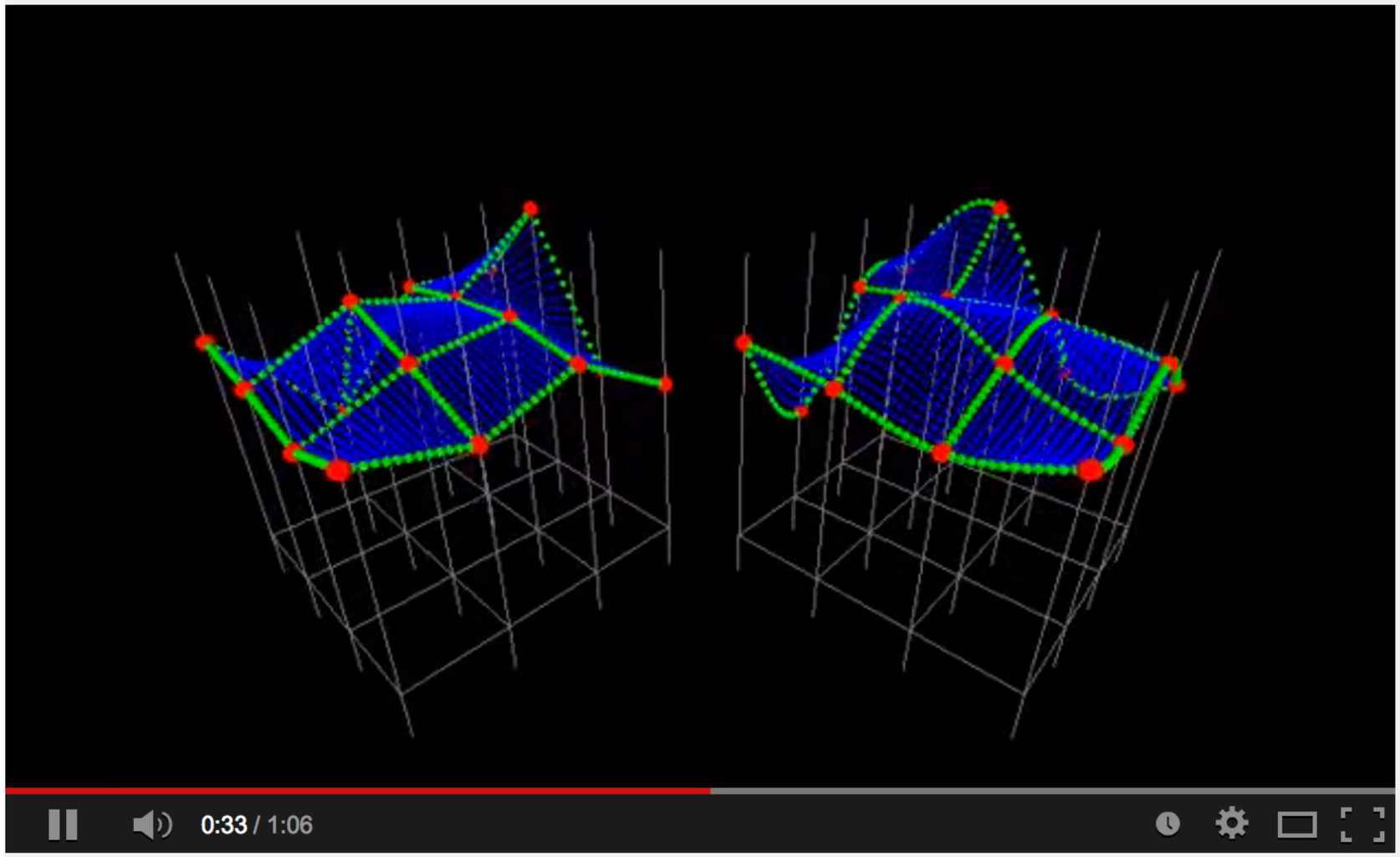
http://en.wikipedia.org/wiki/Bicubic_interpolation

https://www.youtube.com/watch?v=EMNpxfWo9go

Go back one page
Pull down to show history

ting Started | Configure default... | vis | gm | CoVis Information | shusen

YouTube



Visualization of Linear vs. Cubic Interpolation

Georg Hackenberg

Subscribe 86

1,654

Add to | Share | More

7 0

Uploaded on Feb 19, 2011

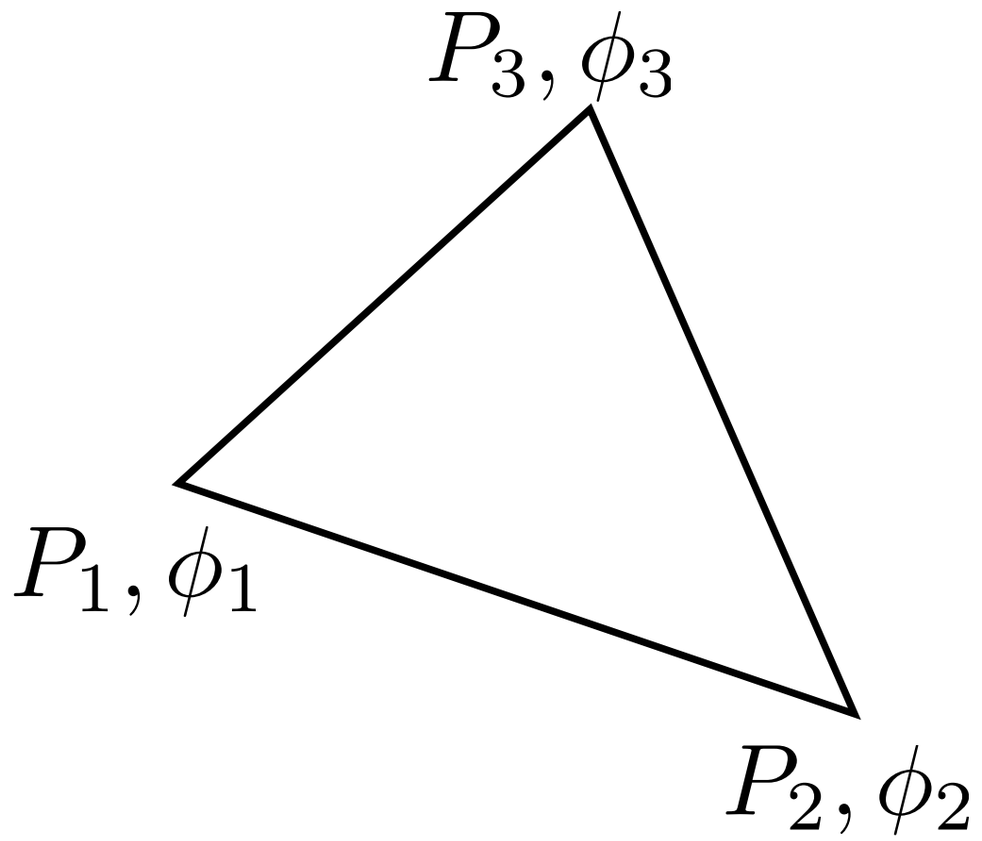
Interpolation is a technique to calculate unknown data points from known samples. There exist many

Linear Interpolation

- In triangle (2D **simplex**)

$$\phi(x, y) = a + bx + cy$$

$$\forall i, \phi(P_i) = \phi_i$$

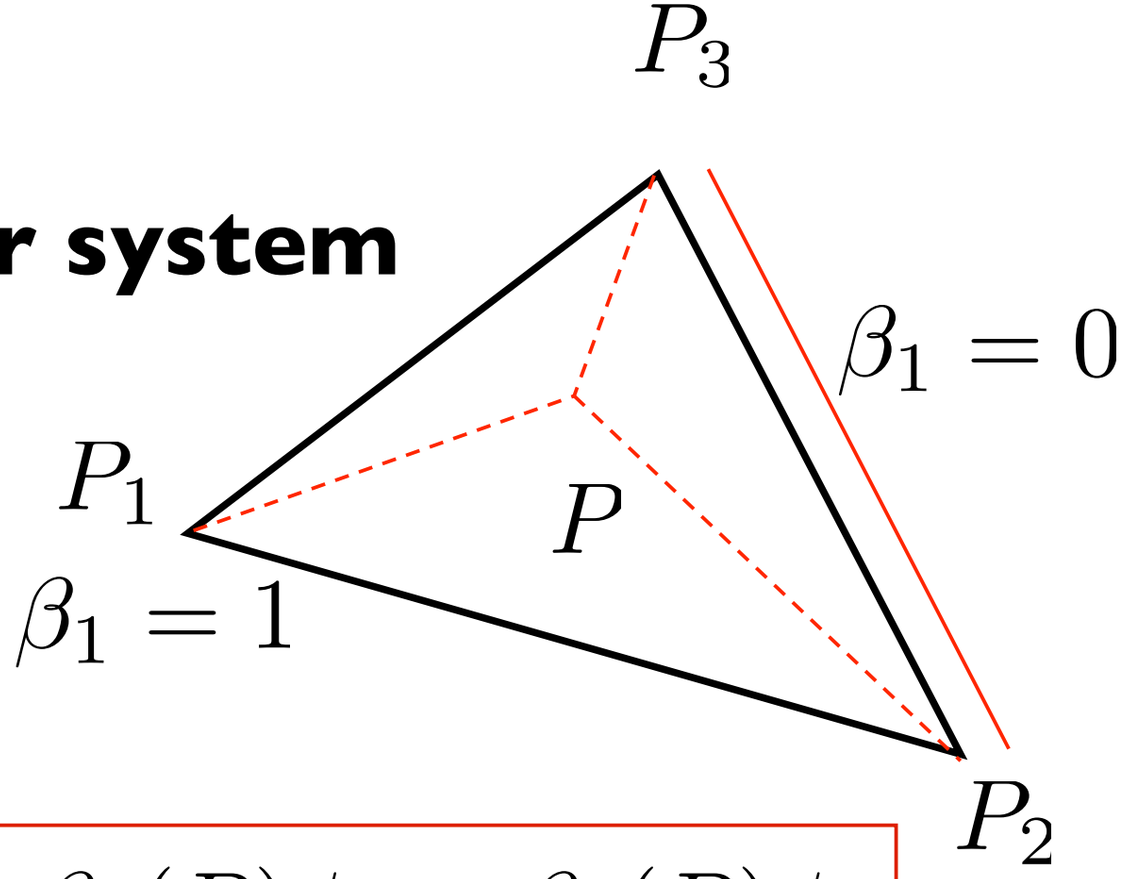


Linear Interpolation

Barycentric coordinates

$$\begin{cases} P = \sum_{i=1}^{3} \beta_i P_i \\ \sum_{i=1}^{3} \beta_i = 1 \end{cases}$$

Linear system

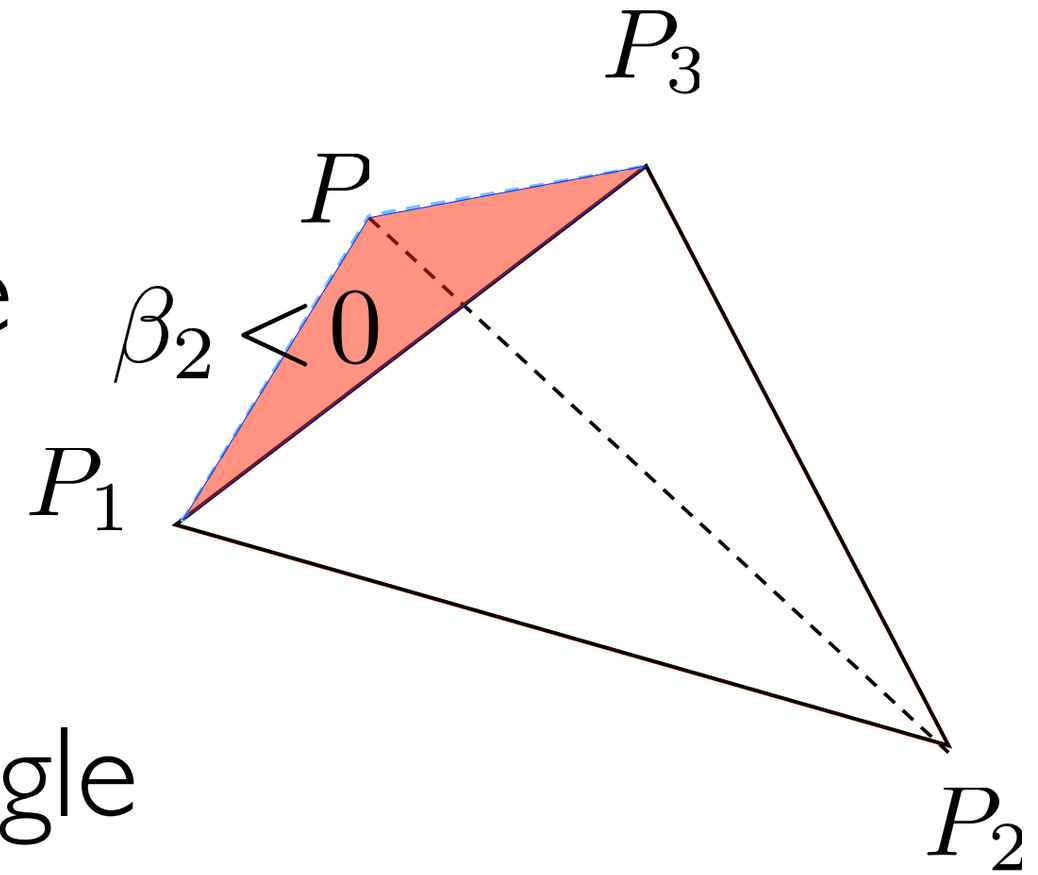


$$\phi(P) = \beta_1(P)\phi_1 + \beta_2(P)\phi_2 + \beta_3(P)\phi_3$$

Linear Interpolation

Barycentric coordinates

- P lies outside triangle ⇔ at least one β_i is negative

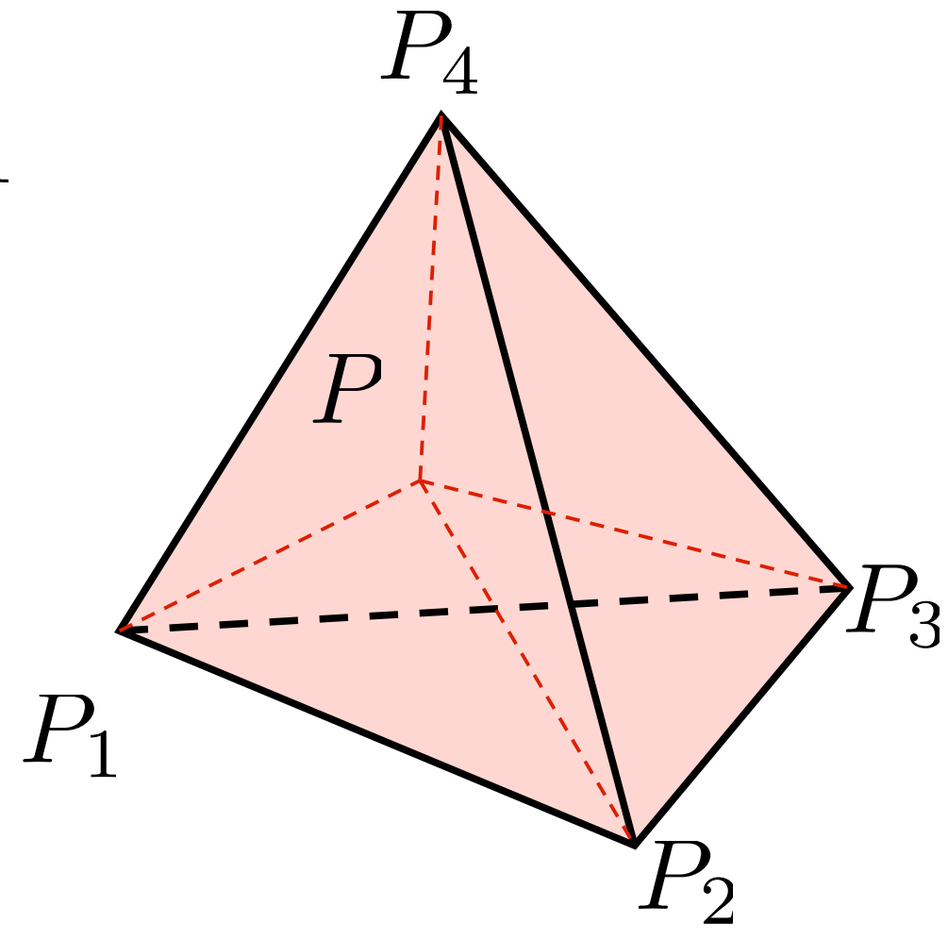


- Easy way to check if a position lies inside a triangle

Linear Interpolation

- In tetrahedron (3D simplex)
  - Barycentric coordinates

$$\left| \begin{array}{l} P = \sum_{i=1}^{4} \beta_i P_i, \sum_{i=1}^{4} \beta_i = 1 \\ \\ \phi(P) = \sum_{i=1}^{4} \beta_i(P) \phi_i \end{array} \right.$$



  - Same inclusion test

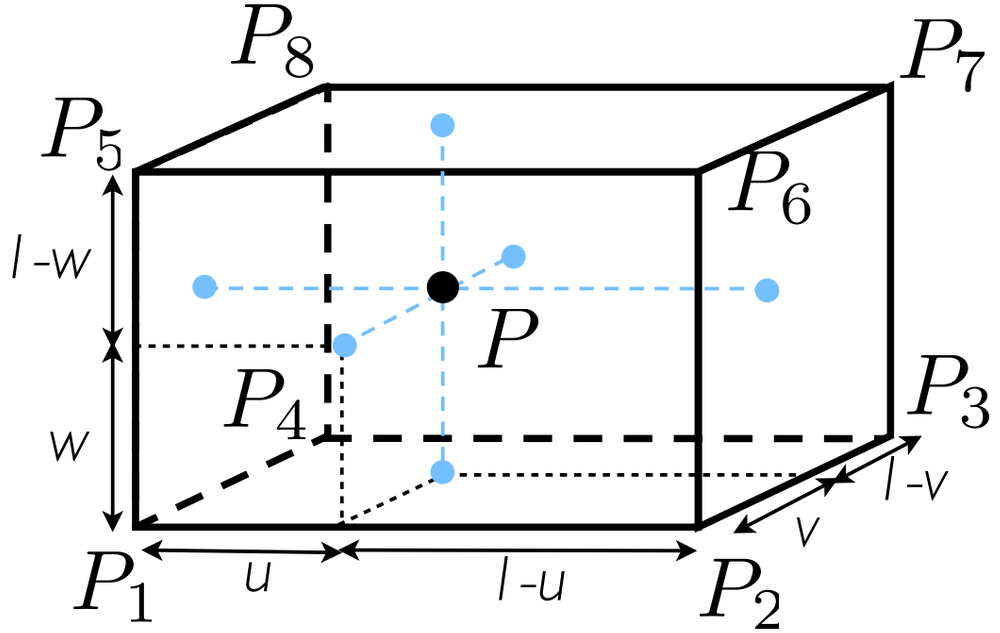
Trilinear Interpolation

- In a cuboid (axis parallel)
  - general formula

$$\phi(x, y, z) = axyz + bxy + cxz + dyz + ex + fy + gz + h$$

  - with local coordinates

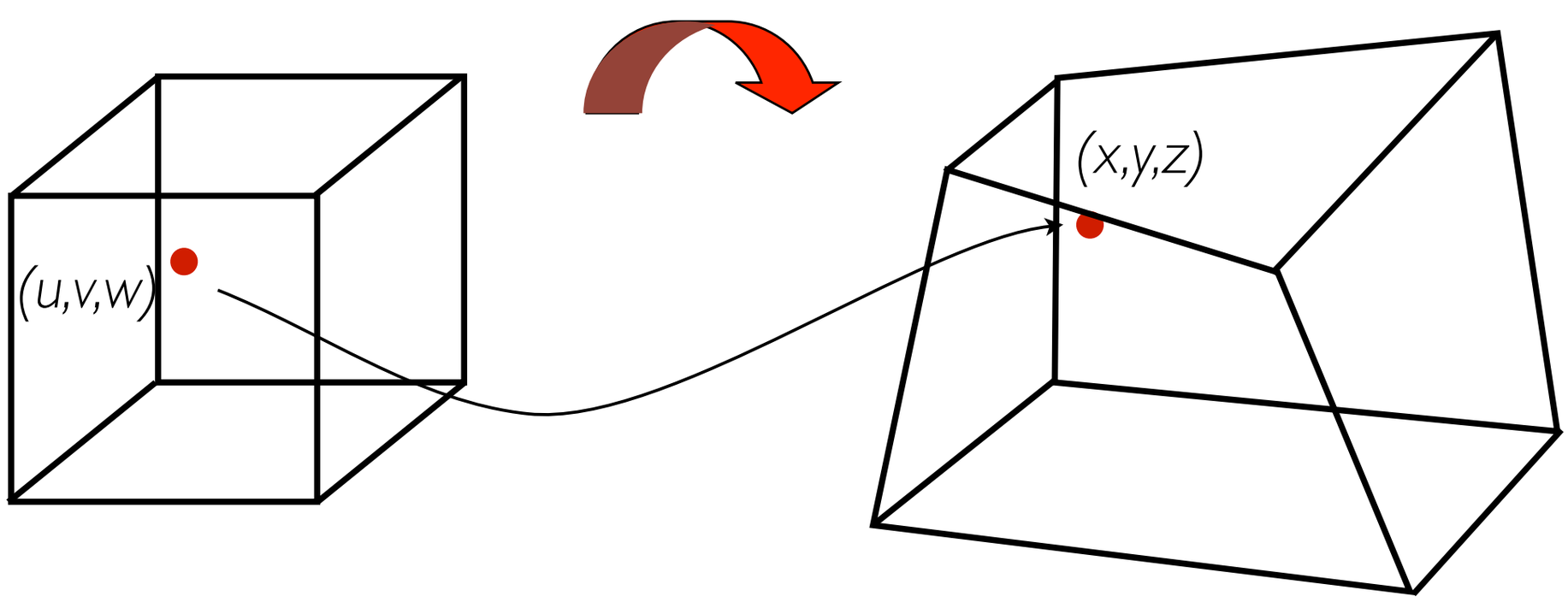
$$\begin{aligned} P = P_1 \\ &+u(P_2 - P_1) \\ &+v(P_4 - P_1) \\ &+w(P_5 - P_1) \\ &+uv(P_1 - P_2 + P_3 - P_4) \\ &+uw(P_1 - P_2 + P_6 - P_5) \\ &+vw(P_1 - P_4 + P_8 - P_5) \\ &+uvw(P_1 - P_2 + P_3 - P_4 + P_5 - P_6 + P_7 - P_8) \end{aligned}$$



Trilinear Interpolation

- In arbitrary hexahedron

  i. Start from (u,v) coordinates (if available) in computational space and apply previous formula



Trilinear Interpolation

- In arbitrary hexahedron

ii. Otherwise

1. compute (inverse) mapping from physical space to computational space (axis aligned unit cube)

I. nonlinear problem

$$(x, y, z)^T = F(u, v, w)^T$$

II. solved with iterative scheme

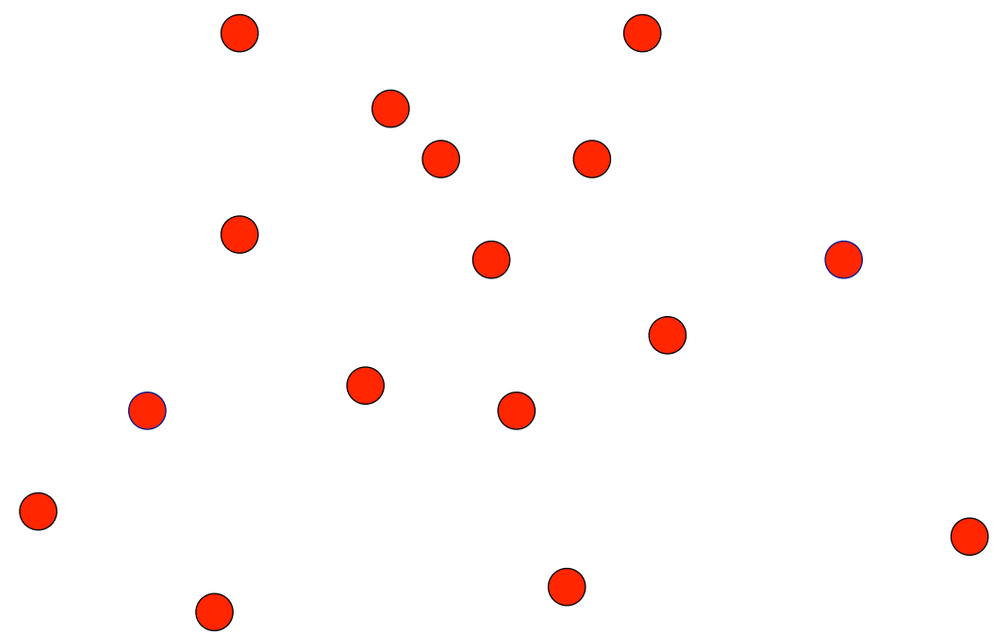
$$F(\mathbf{u} + \vec{\delta}) = F(\mathbf{u}) + J\vec{\delta} + ...$$

$$J\vec{\delta} = -F(\mathbf{u})$$

2. goto i.

Meshless Fields

- What if no topology is provided at all and we only have a geometric sample of some domain?

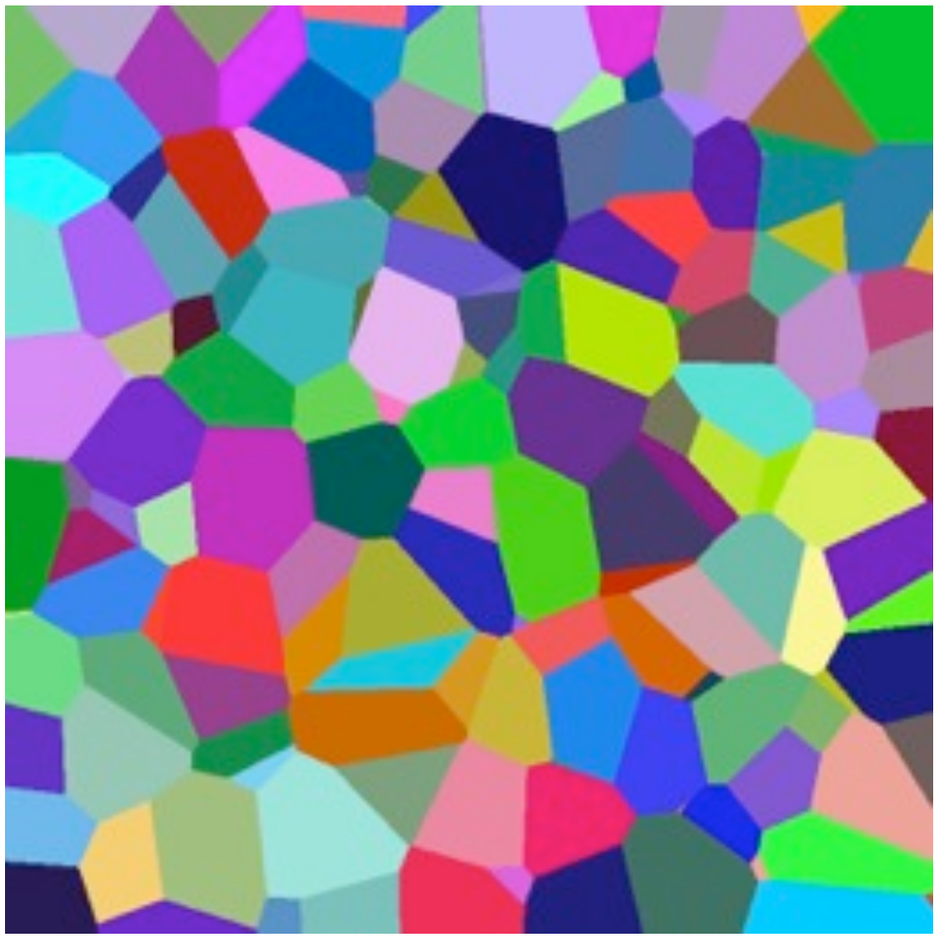


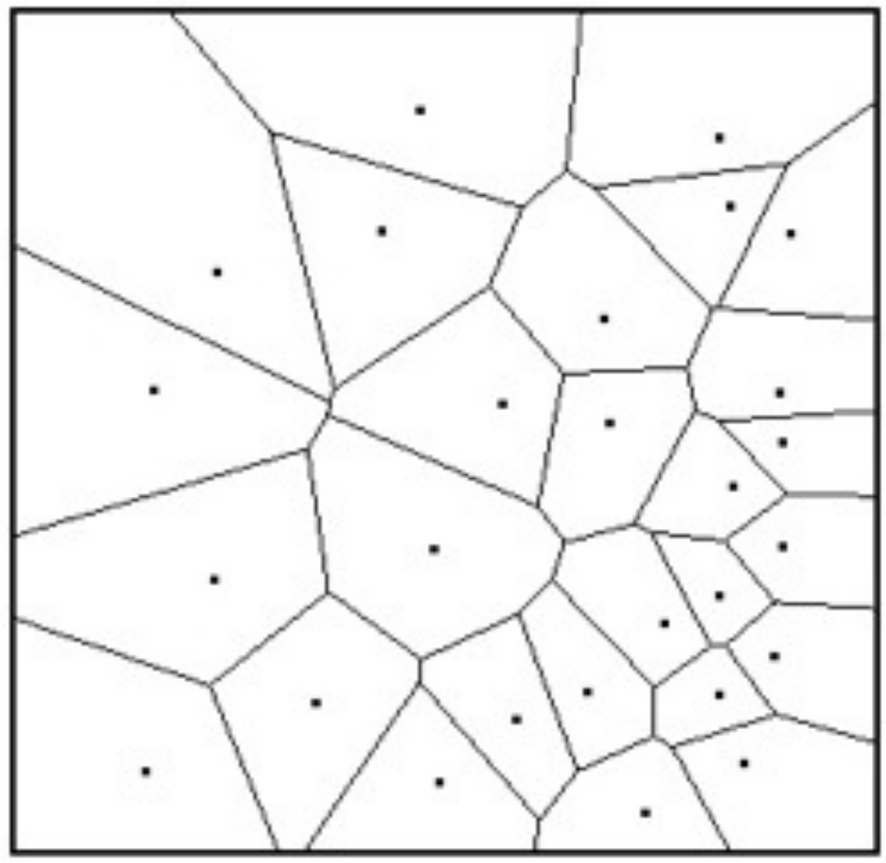
- No underlying structure, explicit encoding of vertices: (x_1,y_1,z_1), (x_2,y_2,z_{12}), …

- Meshless data common in a variety of simulation types: molecular dynamics, fracture, astrophysics, and more.

But Also...

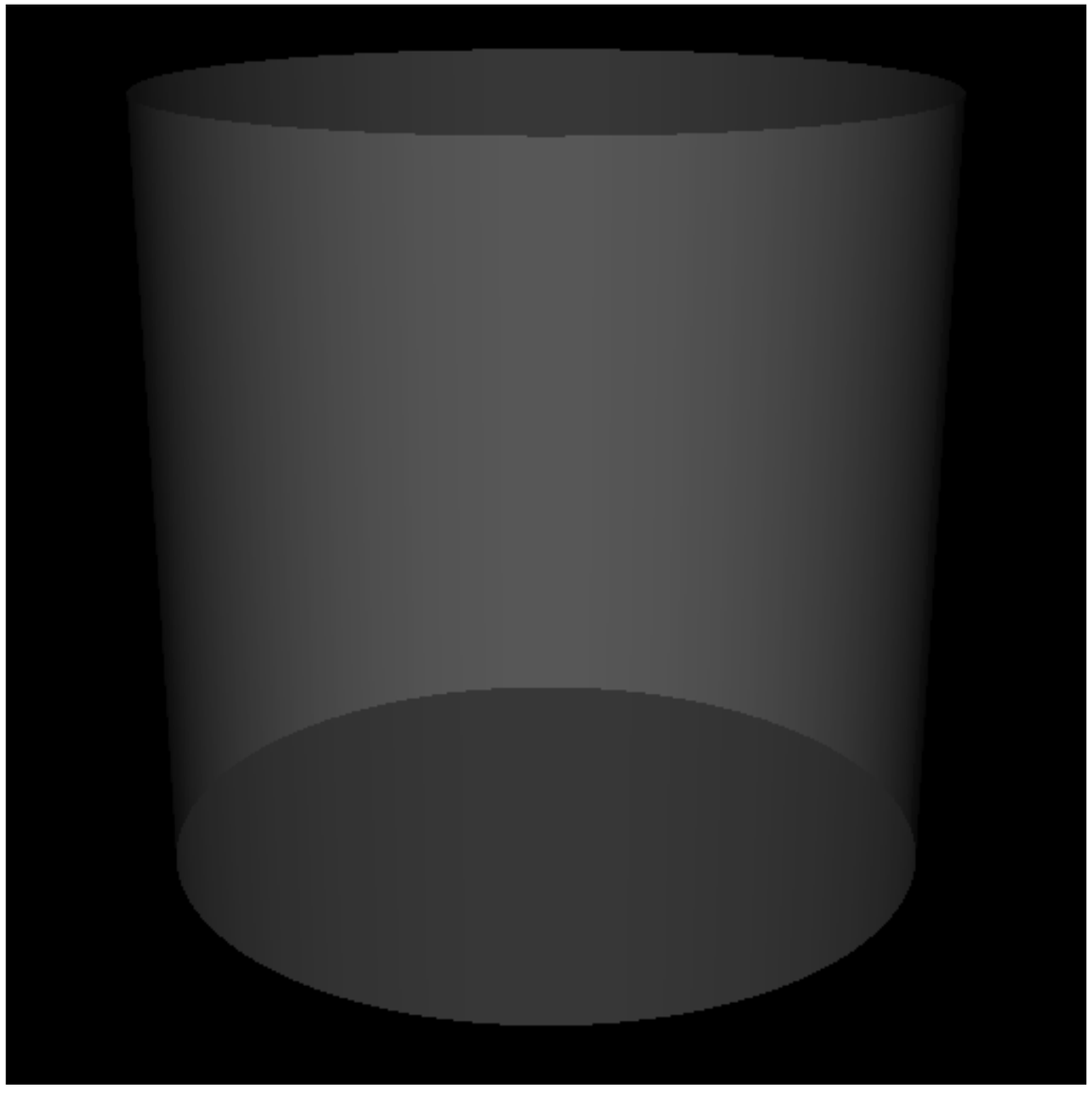
- Nearest Neighbor interpolation





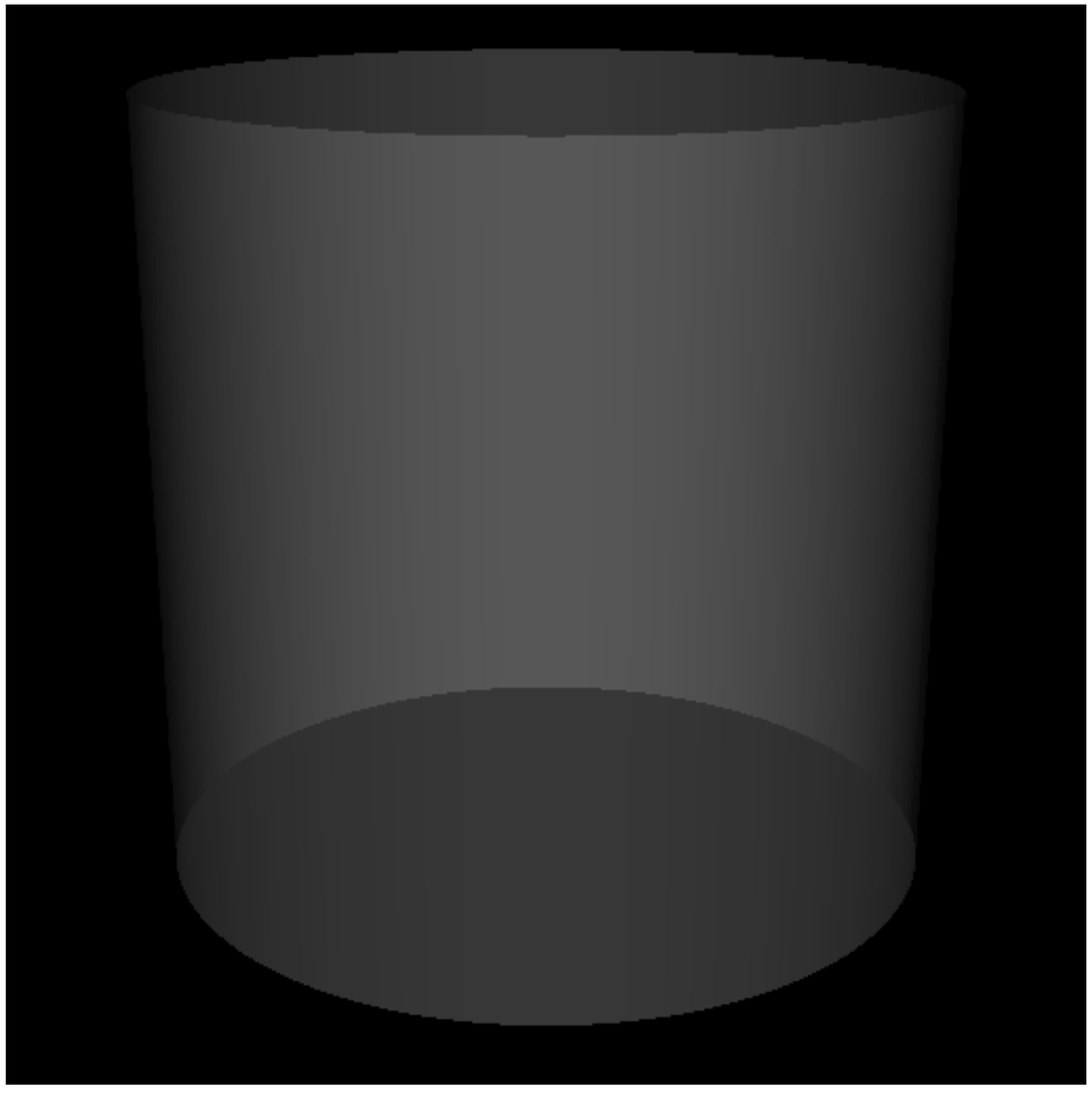
Voronoi diagram

http://www.uni-kl.de/sciviscontest/



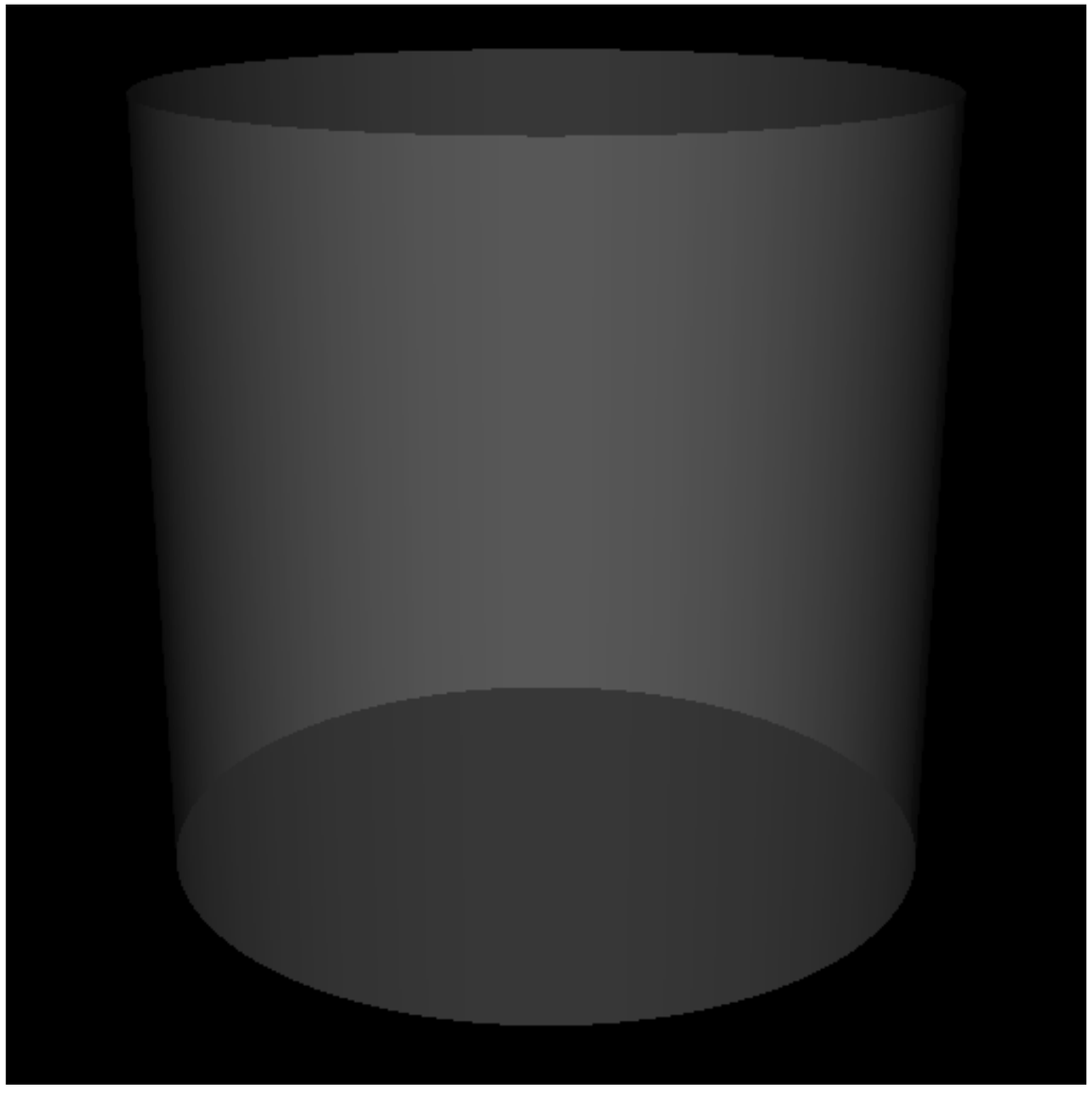
https://wci.llnl.gov/simulation/computer-codes/spheral

http://www.uni-kl.de/sciviscontest/



https://wci.llnl.gov/simulation/computer-codes/spheral

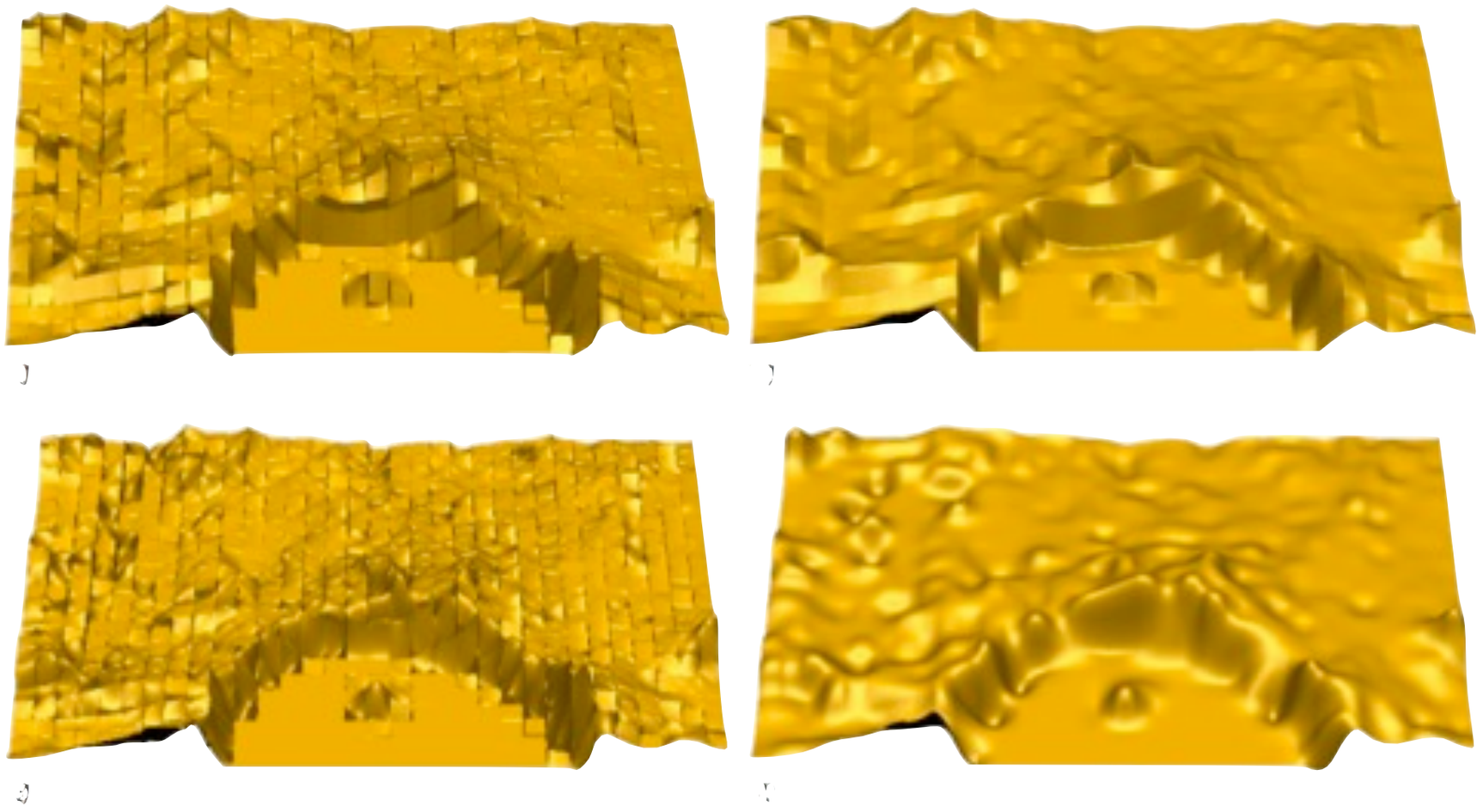
http://www.uni-kl.de/sciviscontest/



https://wci.llnl.gov/simulation/computer-codes/spheral

But Also...

- Higher-order interpolation schemes
  - splines, local polynomial fit (interpolation, least sq., ...)
  - smooth reconstruction kernels (on uniform grids)



Interpolation Affects Our Perception of Discrete Objects as Continuous

Human Factors in Visualization Research

Melanie Tory and Torsten Möller

Abstract—Visualization can provide valuable assistance for data analysis and decision making tasks. However, how people perceive and interact with a visualization tool can strongly influence their understanding of the data as well as the system's usefulness. Human factors therefore contribute significantly to the visualization process and should play an important role in the design and evaluation of visualization tools. Several research initiatives have begun to explore human factors in visualization, particularly in perception-based design. Nonetheless, visualization work involving human factors is in its infancy, and many potentially promising areas have yet to be explored. Therefore, this paper aims to 1) review known methodology for doing human factors research, with specific emphasis on visualization, 2) review current human factors research in visualization to provide a basis for future investigation, and 3) identify promising areas for future research.

Index Terms—Human factors, visualization, perception, cognitive support, methodology.

1 Introduction

Modern technology provides access to large quantities of data in many application domains, such as medical imaging, fluid flow simulation, and geographic information "information visualization." We introduce new terminology that is more precise. *Continuous model visualization* encom-

Lec17 Reading

- Munzner, Chapter 8.4.1
- Marching Cubes: A High Resolution 3D Surface Construction Algorithm. William E. Lorensen and Harvey E. Cline, SIGGRAPH 1987.

Reminder
Assignment 04

Assigned: Monday, March 13
Due: Monday, March 27, 4:59:59 pm

Reminder
Project Milestone 02

Assigned: Wednesday, February 22
Due: Wednesday, March 29, 4:59:59 pm