MESH MATH AND BEYOND

On creating, storing and using geometry

Silvia Sellán
@sellan_s

Fellow geometry people: In (the magnificent) "This is me trying", Taylor Swift says "I was so ahead of the curve, the curve became a sphere". What does this mean?

MESH MATH AND BEYOND

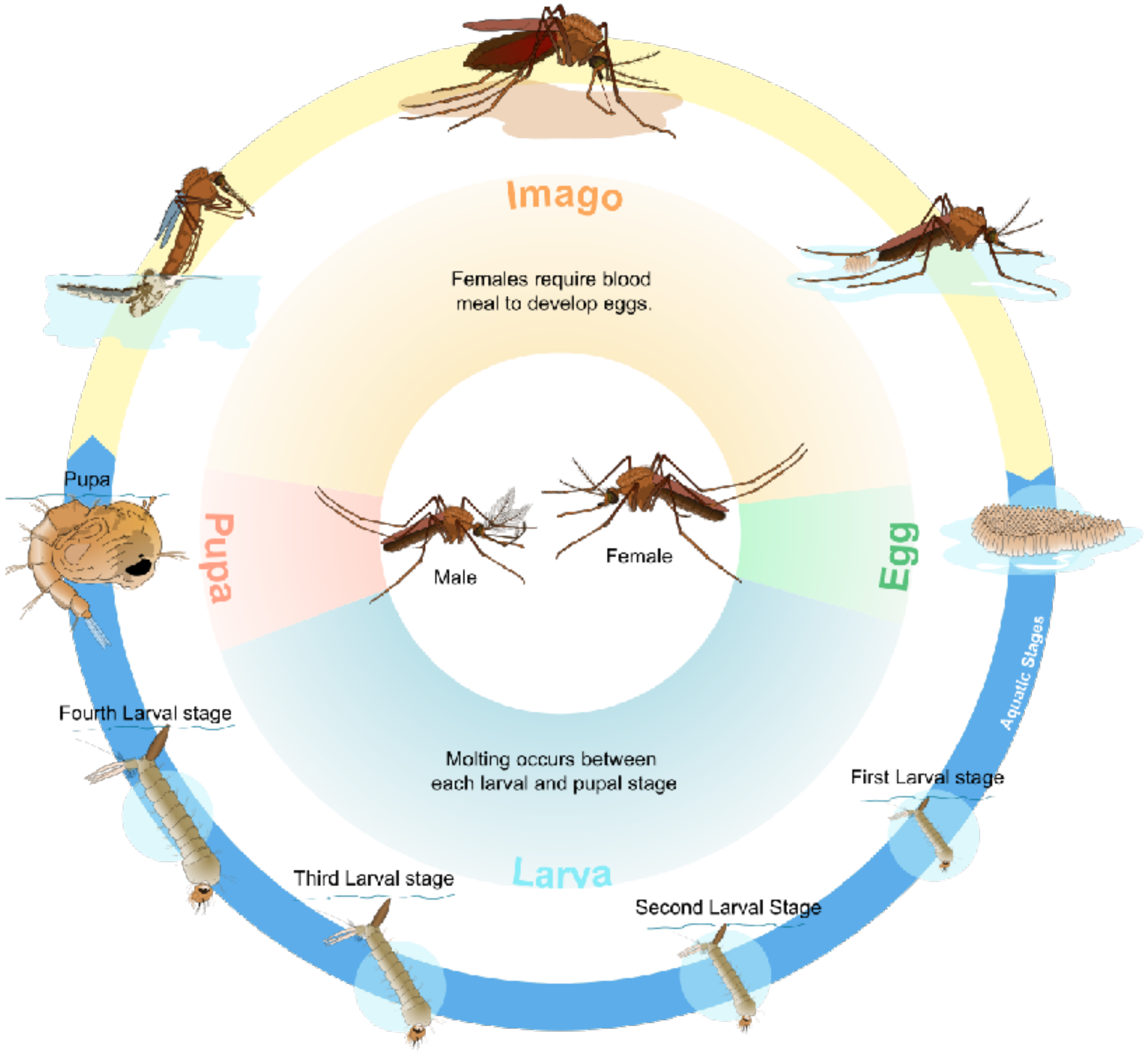
On creating, storing and using geometry

NICE TO MEET YOU!

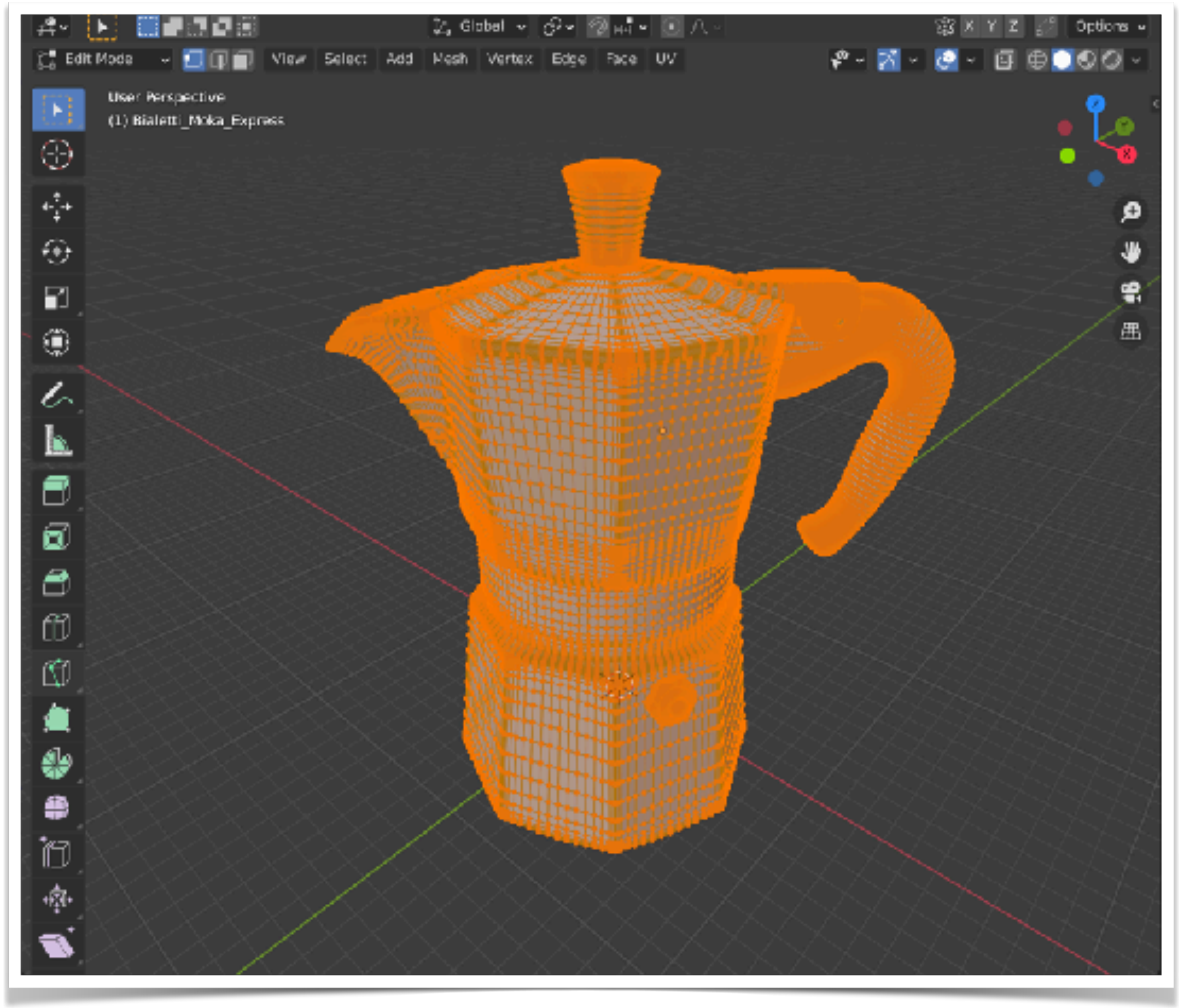
“Geometry Processing is a subfield of biology”

- Alec Jacobson

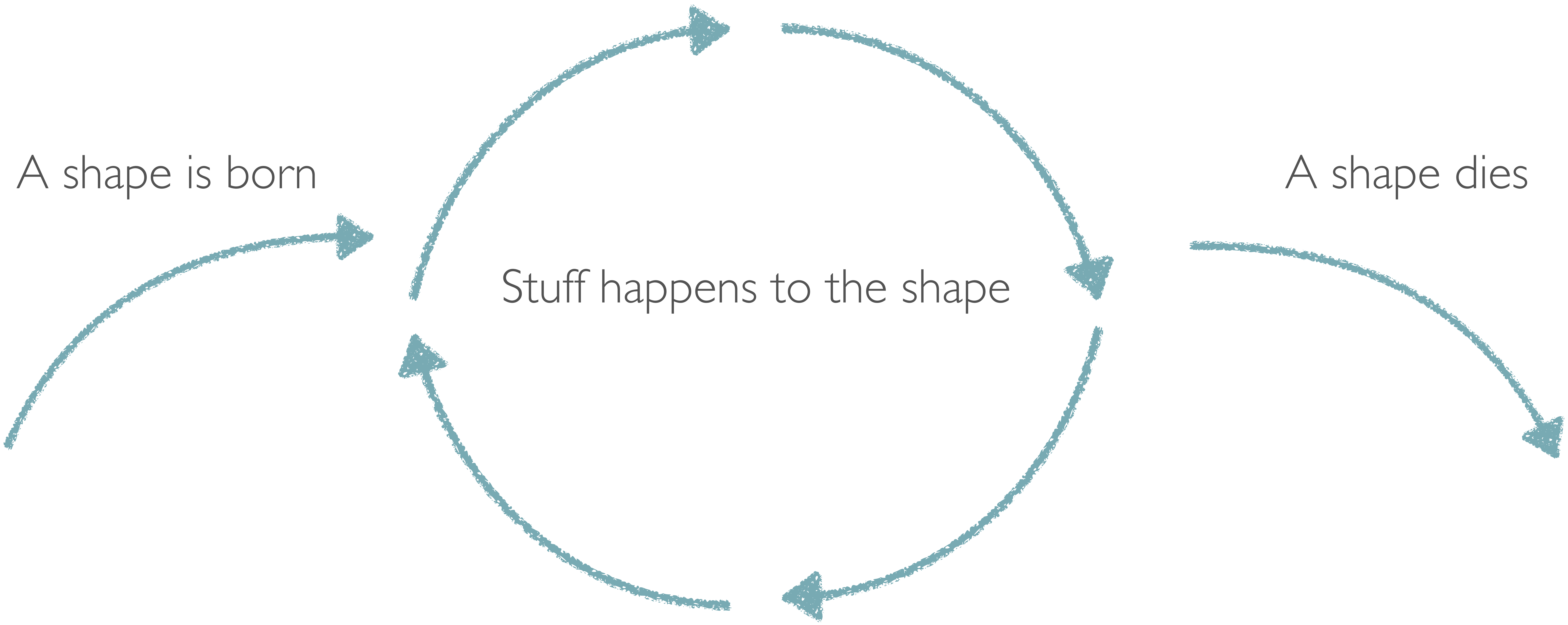
ORGANISM HAVE A LIFE CYCLE



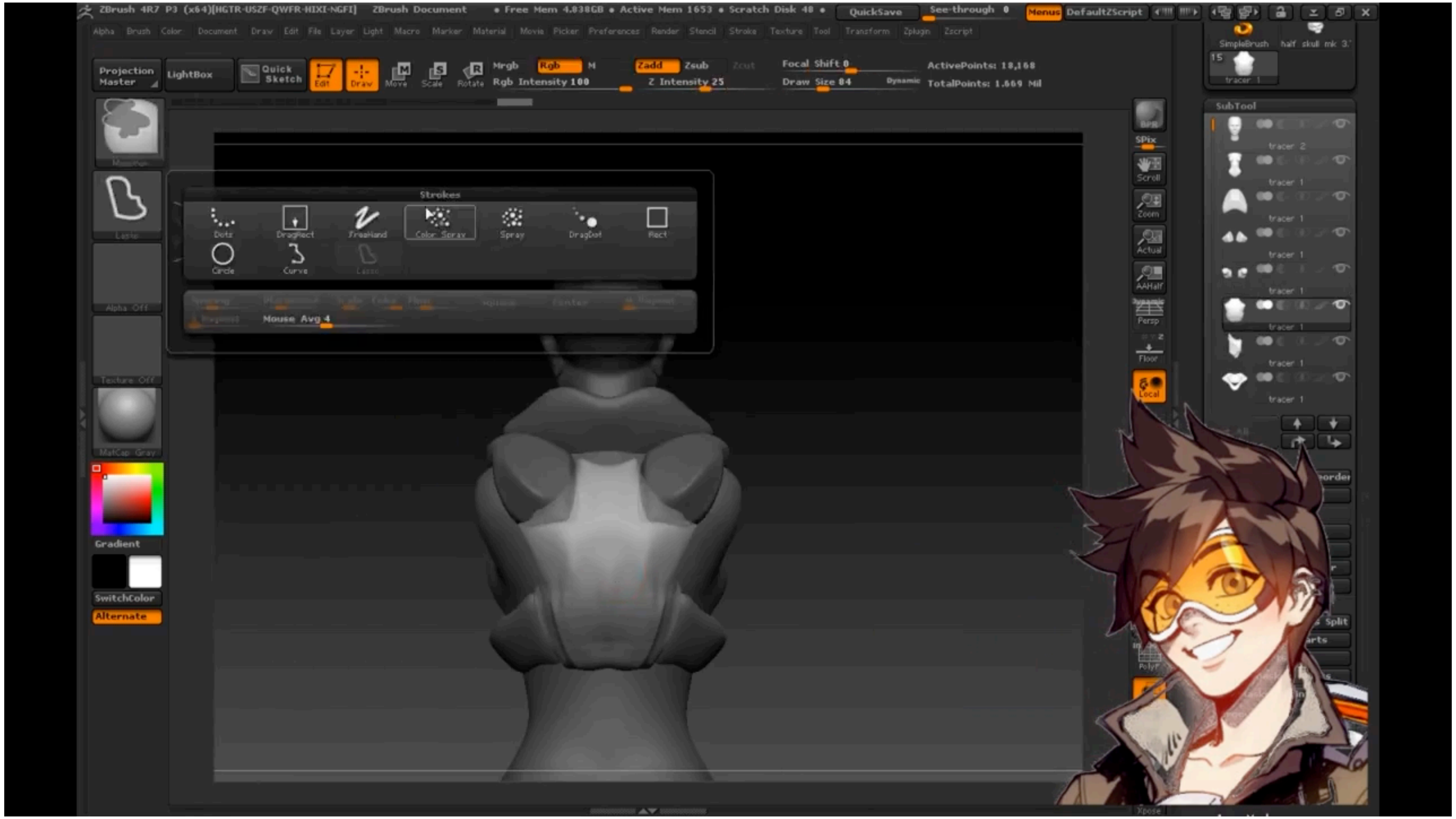
SHAPES HAVE A LIFE CYCLE TOO



SHAPES HAVE A LIFE CYCLE TOO



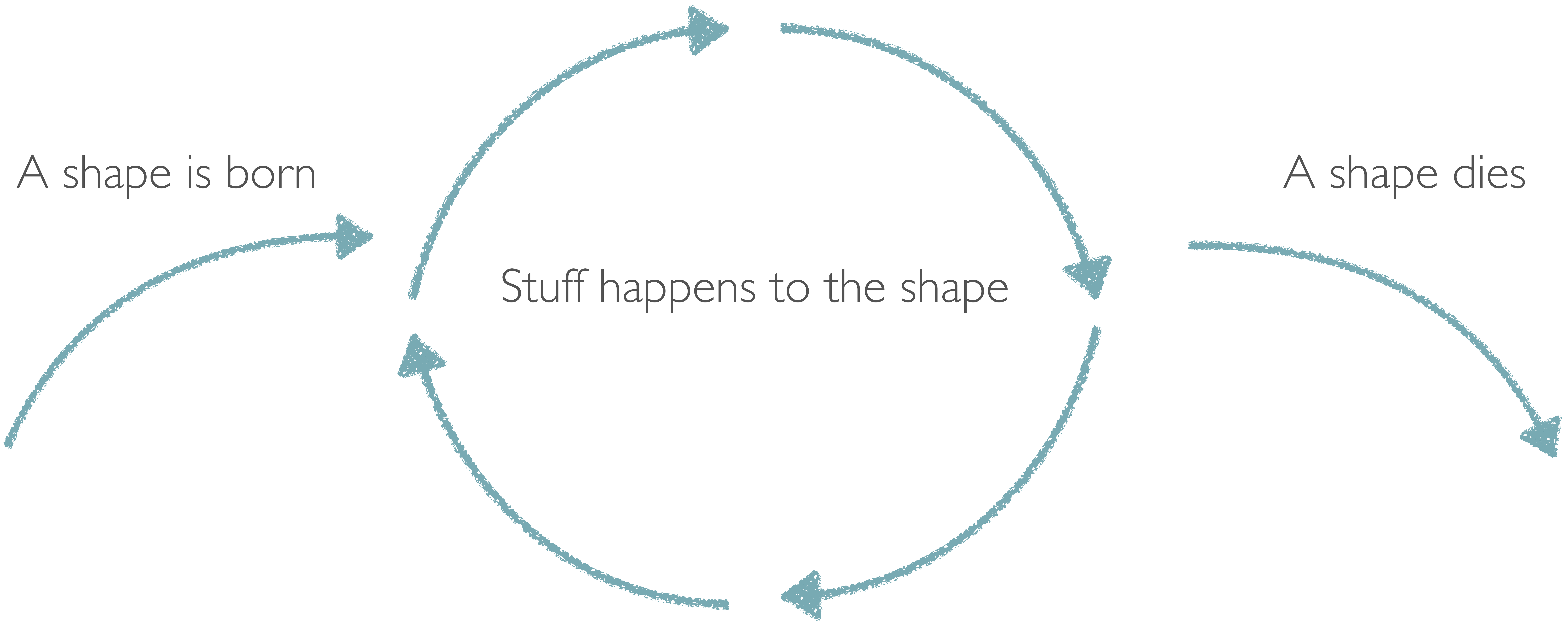
SHAPES HAVE A LIFE CYCLE TOO



Overwatch - Tracer 3D print model - time-lapse sculpt, by "Printed Obsession", Youtube

A shape dies

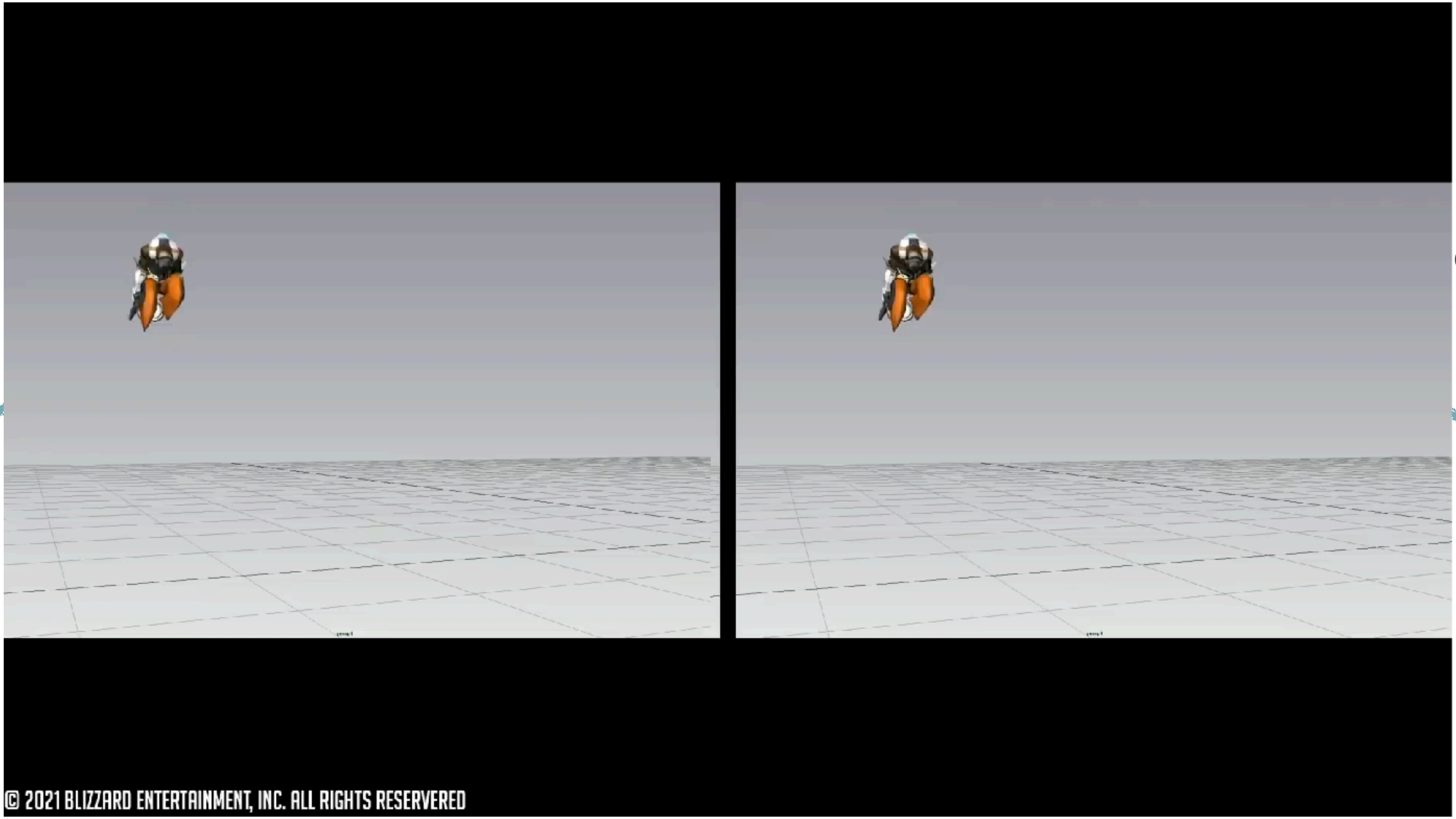
SHAPES HAVE A LIFE CYCLE TOO



SHAPES HAVE A LIFE CYCLE TOO

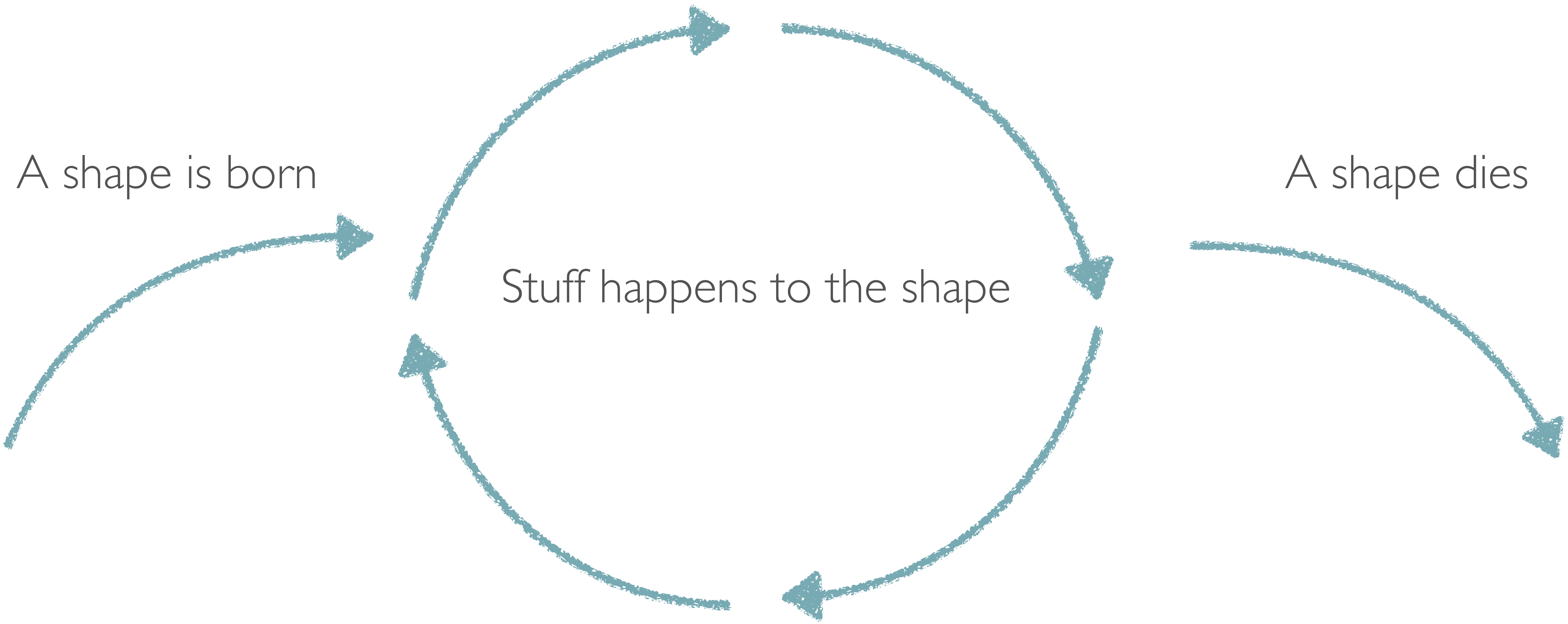
A sh

dies



How devs break bones to make animation feel right, by "Jenna Stoeber (Polygon)", Youtube

SHAPES HAVE A LIFE CYCLE TOO



SHAPES HAVE A LIFE CYCLE TOO

A shape is born

How devs break bones to make animation feel right, by "Jenna Stoeber (Polygon)", Youtube

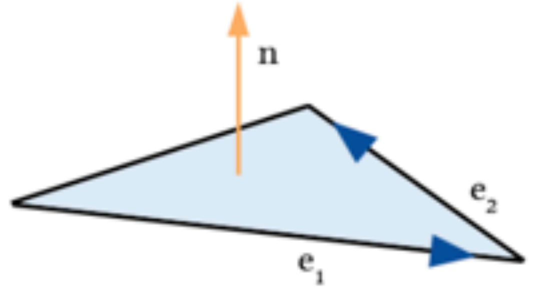
WHAT YOU'VE SEEN SO FAR

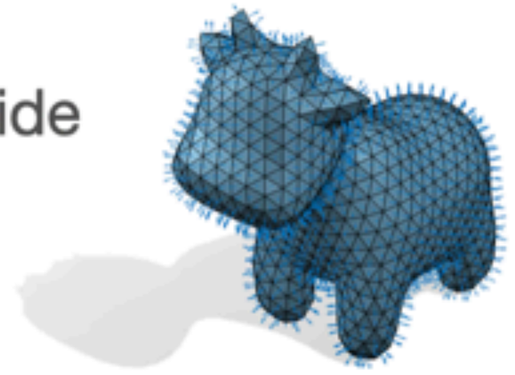
A shape is born

29

NORMAL VECTOR

- The normal vector $\mathbf{n}$ is the unit-length perpendicular vector to a triangle and positively oriented.
- $\tilde{\mathbf{n}} = \mathbf{e}_1 \times \mathbf{e}_2, \quad \mathbf{n} = \tilde{\mathbf{n}}/||\tilde{\mathbf{n}}||$
- normals point outside





ex 008

A shape dies

WHAT YOU'VE SEEN SO FAR

So far, you've seen **stuff happening to shapes**

And so far, **shape has meant "triangle mesh"**

Reality is much more complicated than that

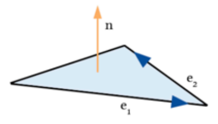
SHAPE REPRESENTATIONS

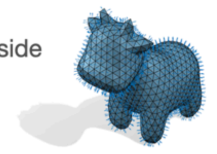
Triangle meshes

29

NORMAL VECTOR

- The normal vector $\mathbf{n}$ is the unit-length perpendicular vector to a triangle and positively oriented.
- $\tilde{\mathbf{n}} = \mathbf{e}_1 \times \mathbf{e}_2, \quad \mathbf{n} = \tilde{\mathbf{n}} / \|\tilde{\mathbf{n}}\|$
- normals point outside

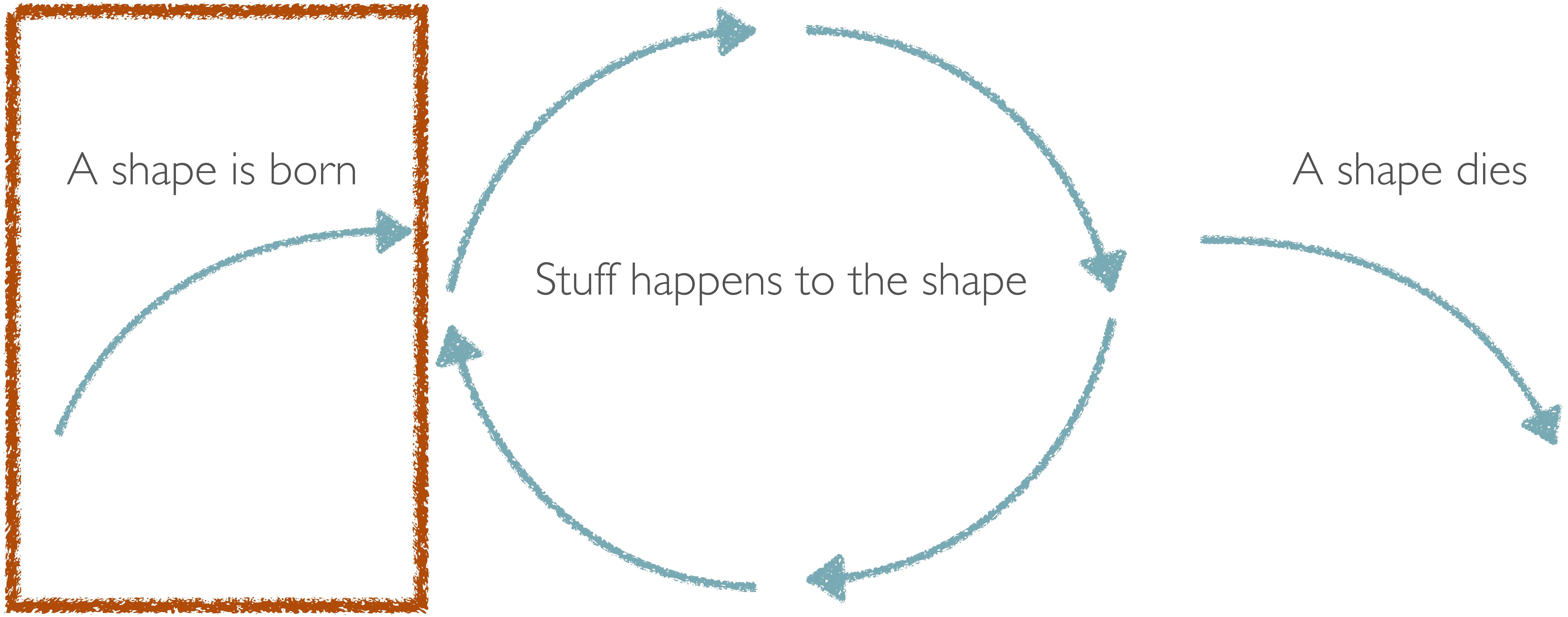




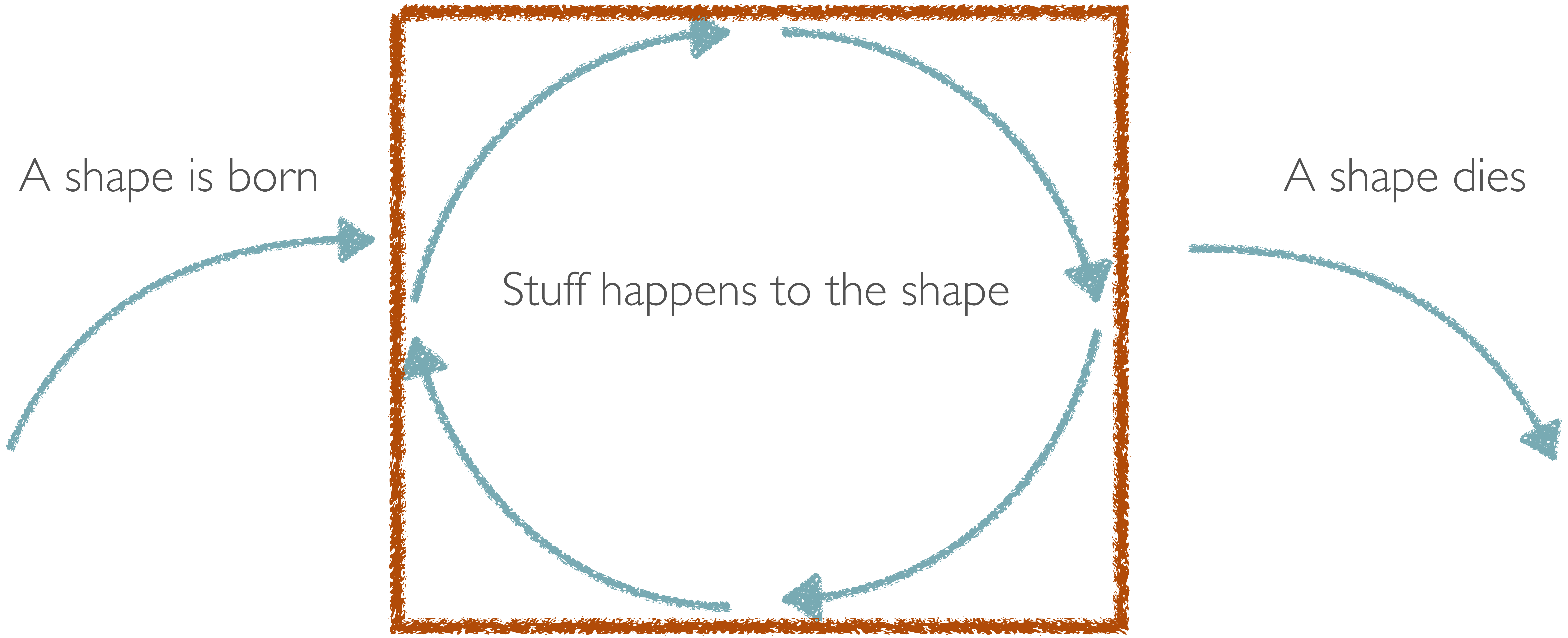
ex 008

WHAT’S WRONG WITH MESHES?

REASON 1: NOT EVERY SHAPE IS BORN AS A MESH

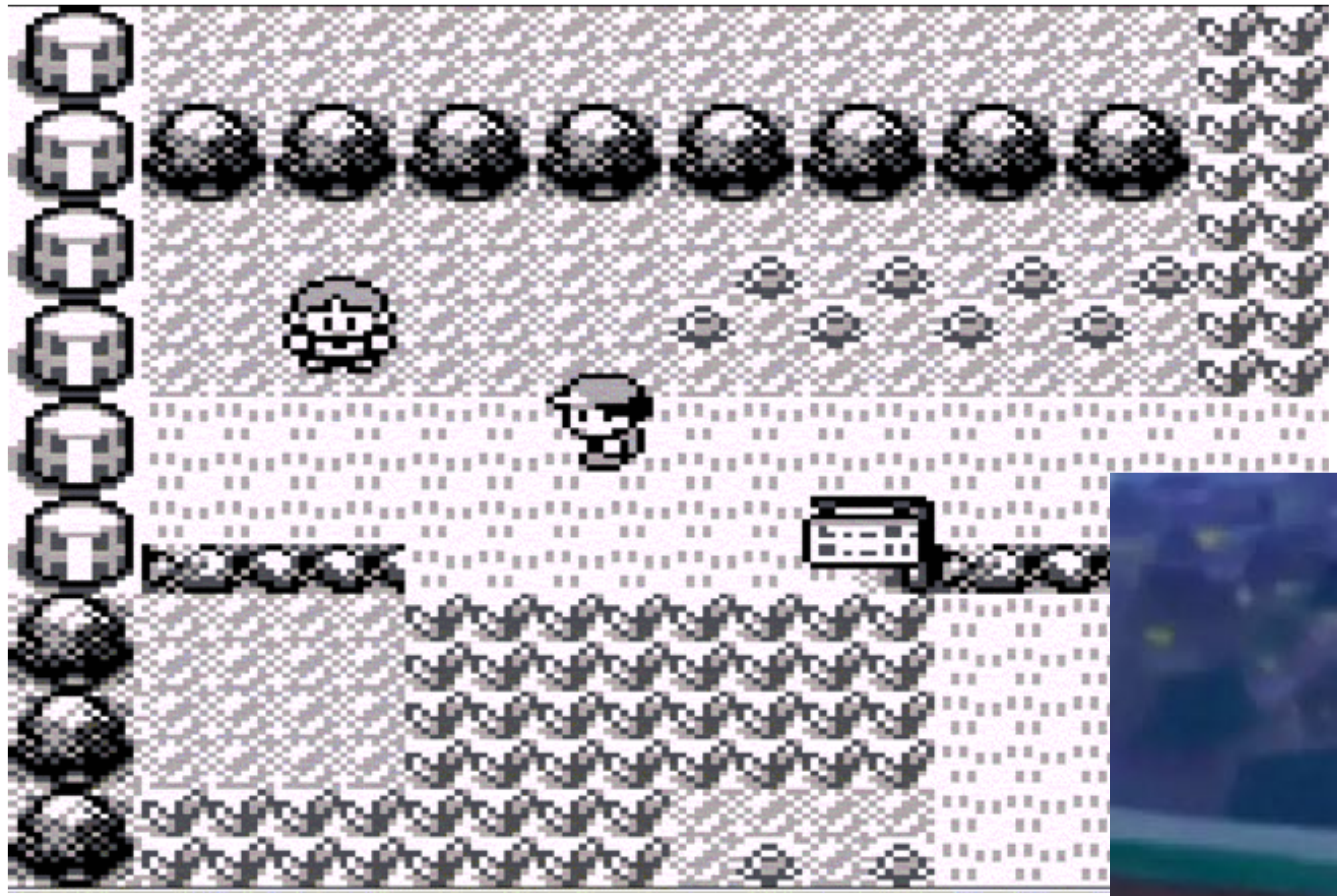


REASON 11: “STUFF” CAN BE HARD TO DO ON MESHES



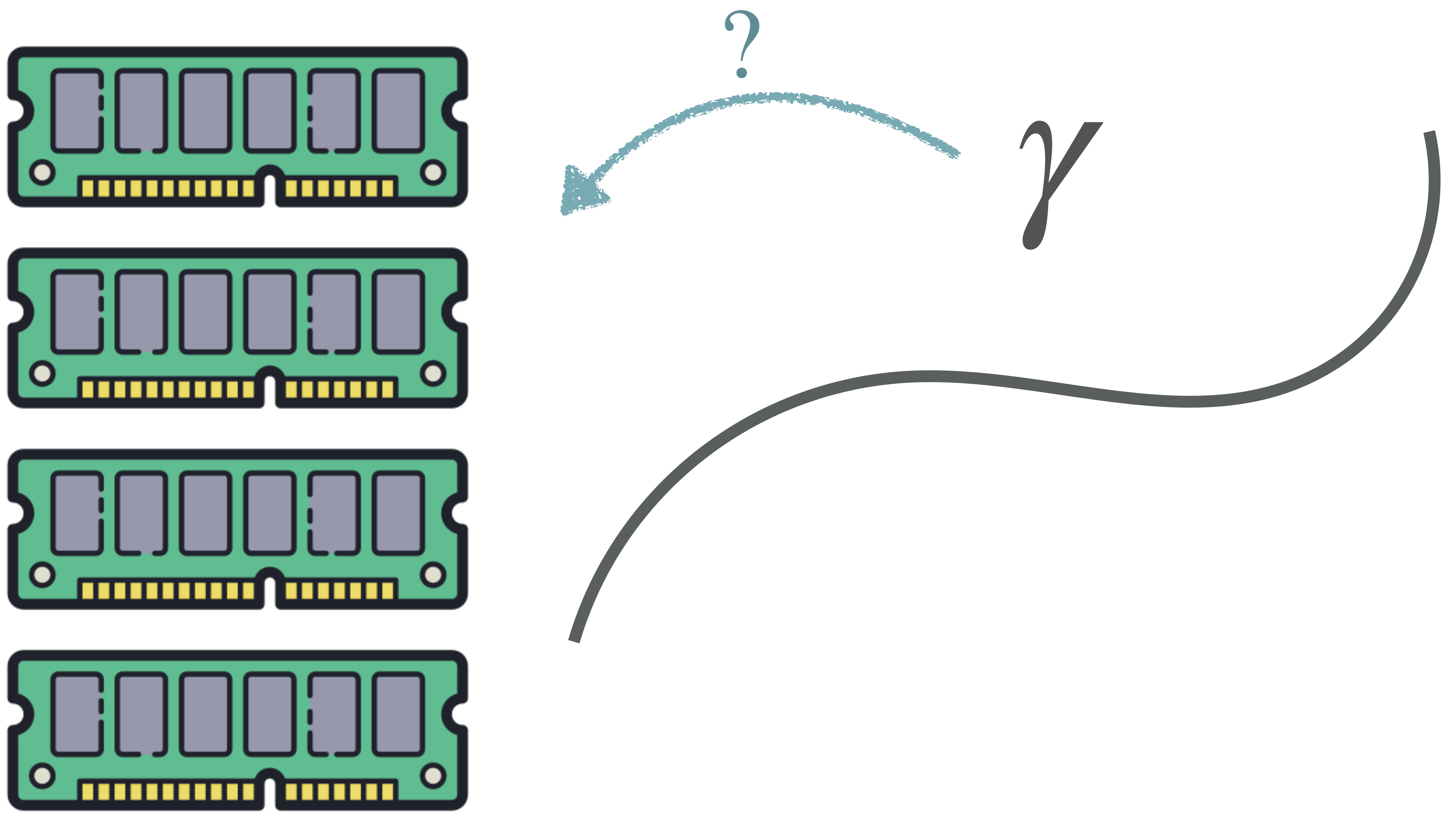
SHAPE REPRESENTATIONS IN 2D

SHAPE REPRESENTATIONS IN 2D

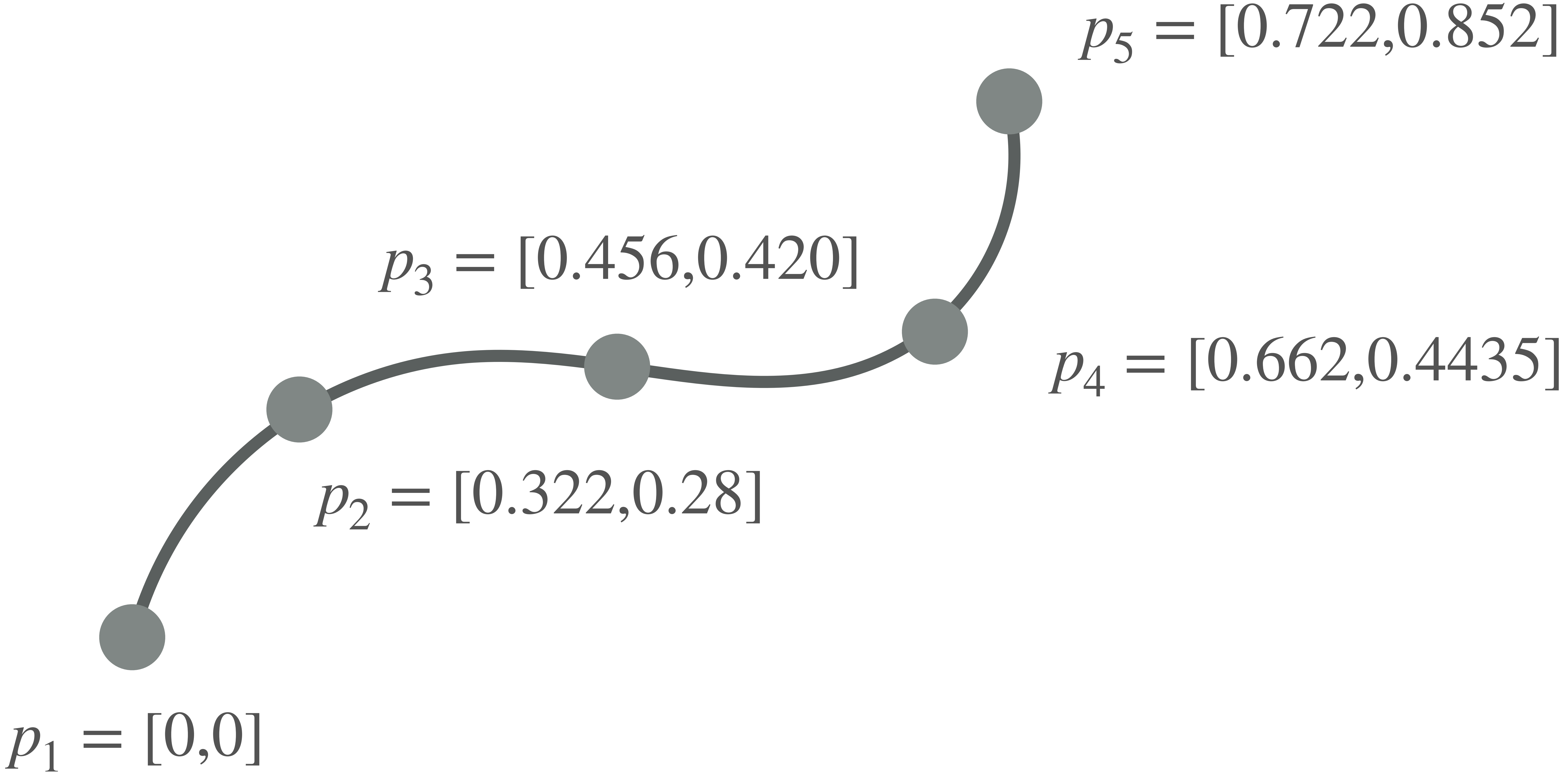


2D'S VERSION OF "SURFACE" IS A CURVE

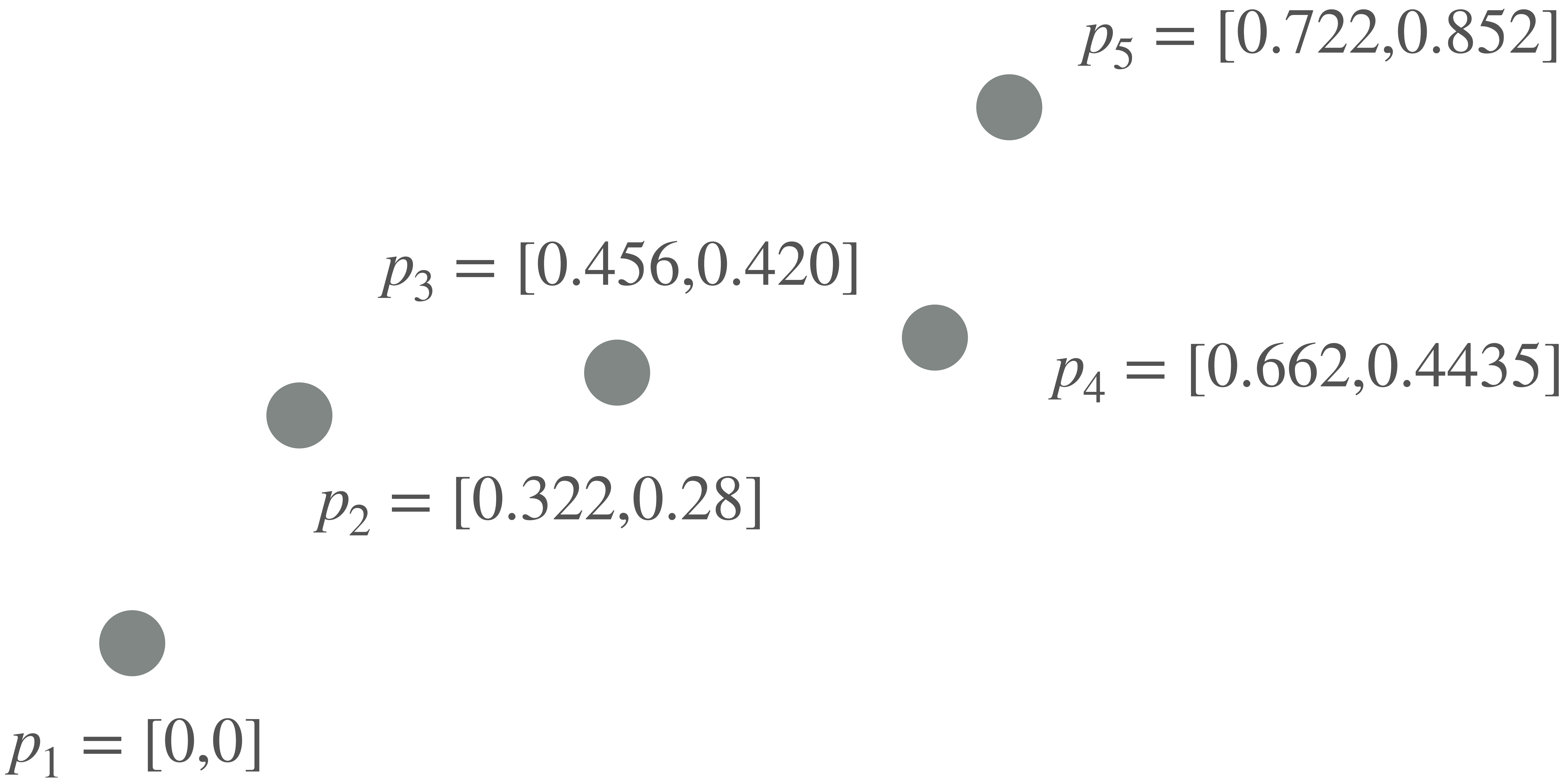
…HOW DO I STORE A CURVE ON A COMPUTER?



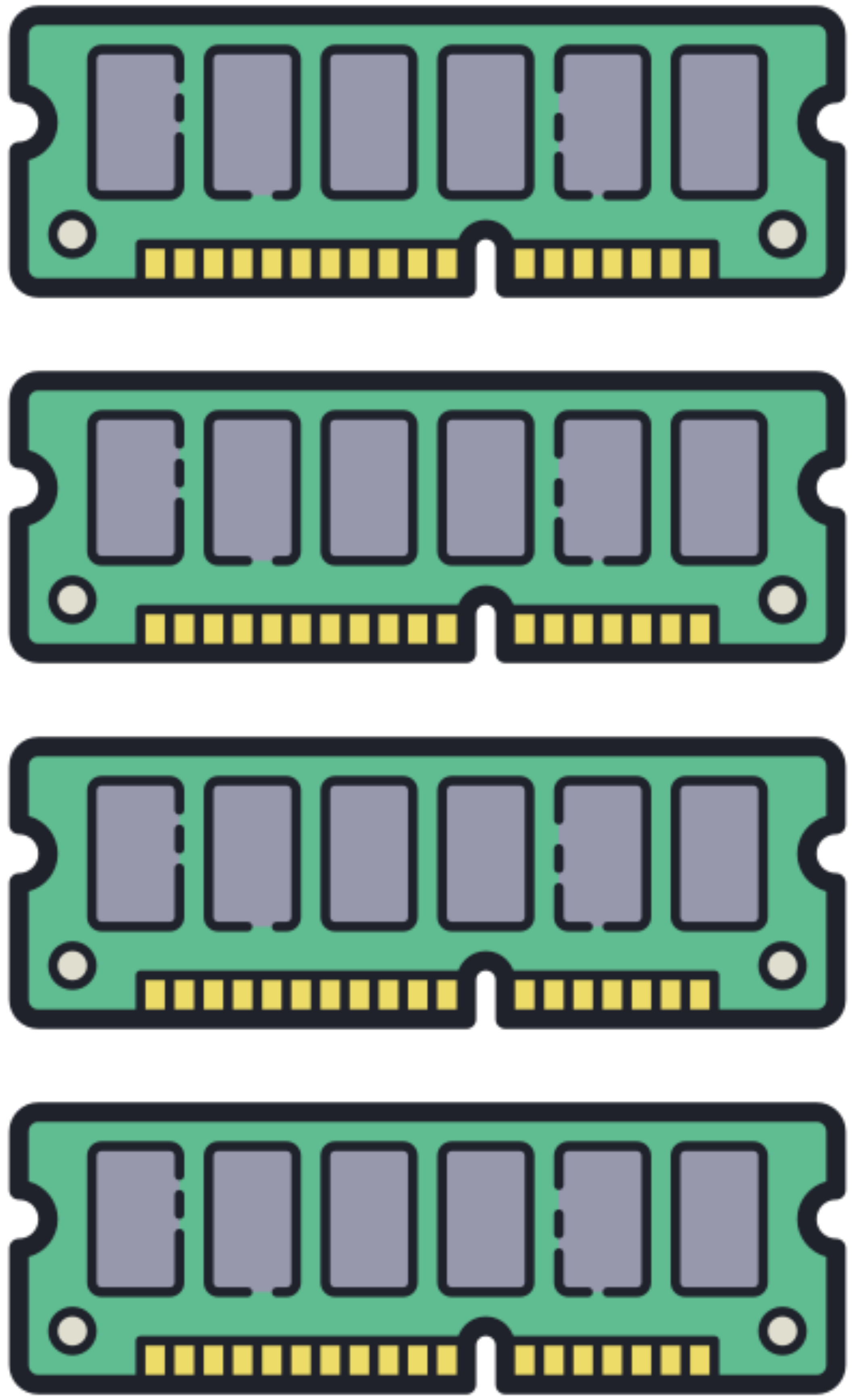
AN OPTION: FINITE SET OF POINTS



AN OPTION: FINITE SET OF POINTS



AN OPTION: FINITE SET OF POINTS



[0,0]

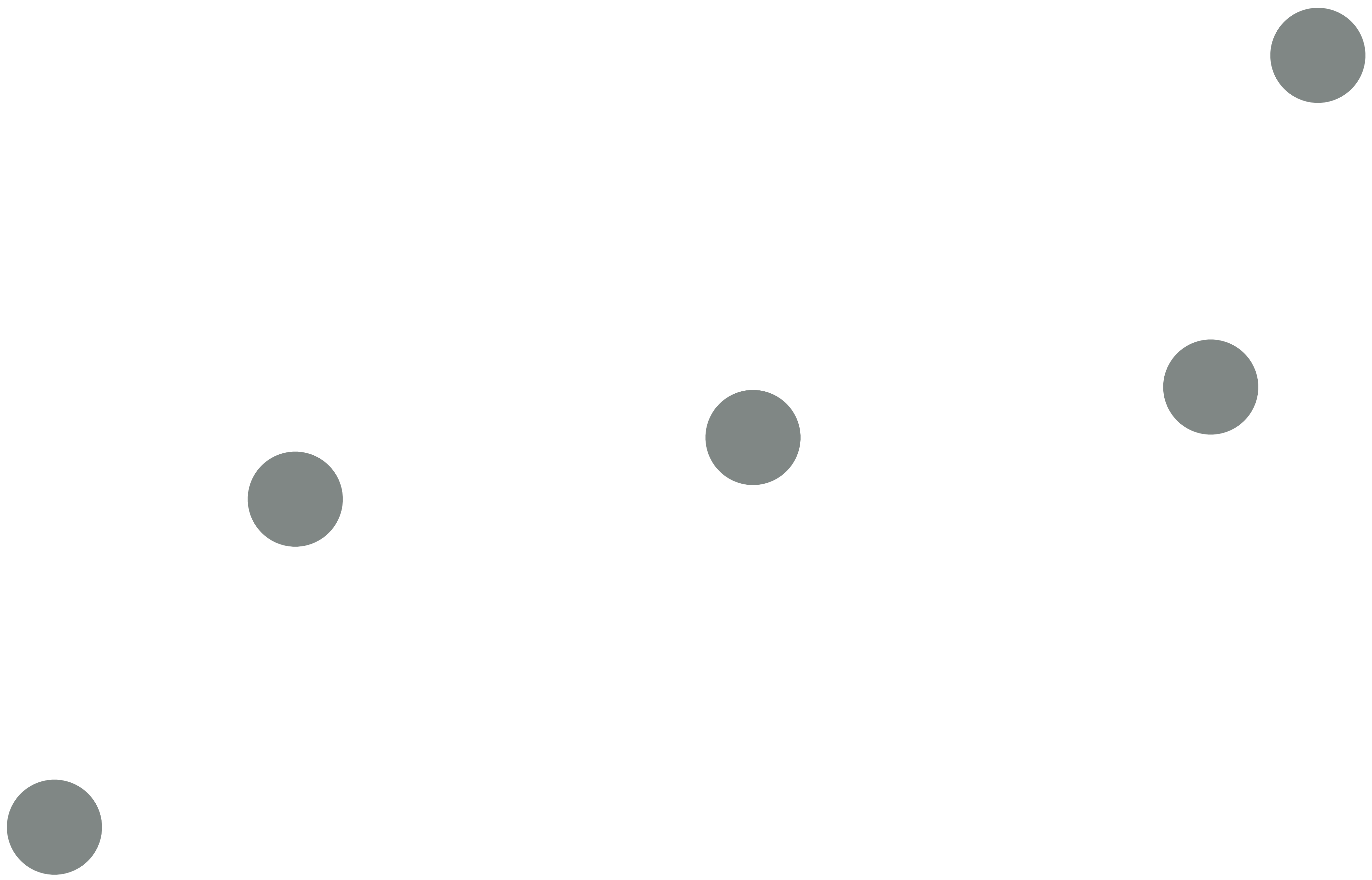
[0.322,0.28]

[0.456,0.420]

[0.662,0.4435]

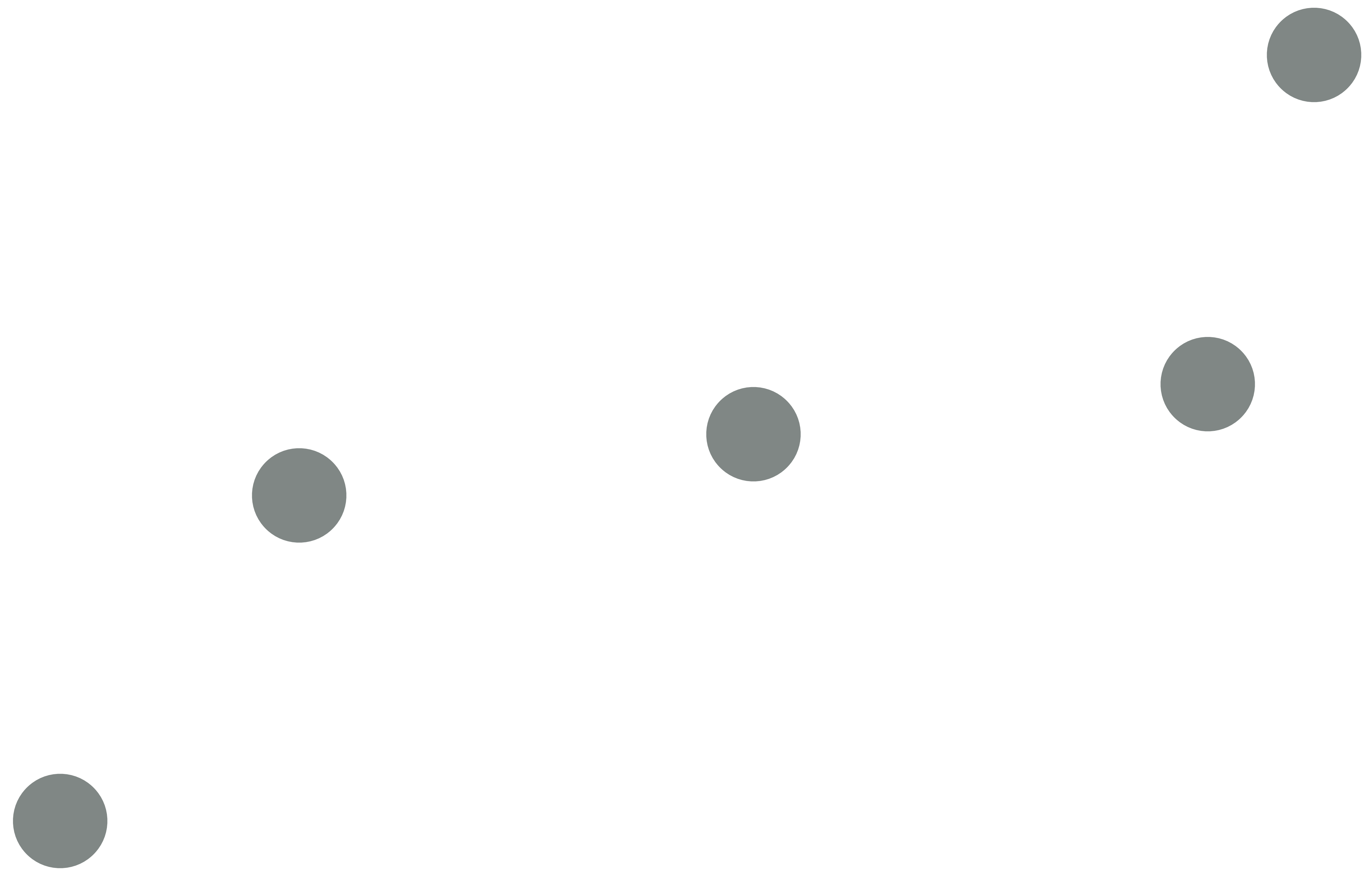
[0.722,0.852]

AN OPTION: FINITE SET OF POINTS



✓ Easy to store

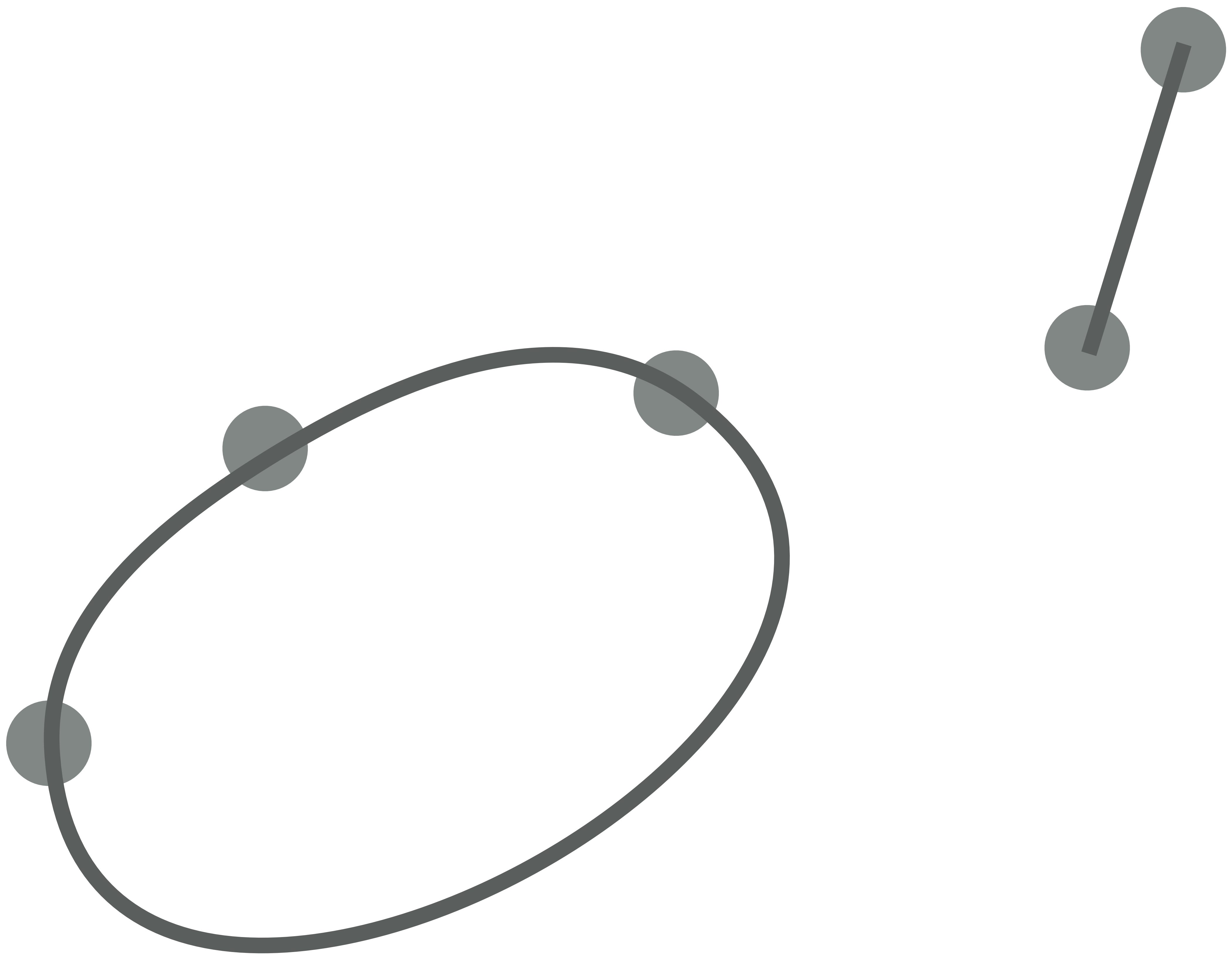
AN OPTION: FINITE SET OF POINTS



✓ Easy to store

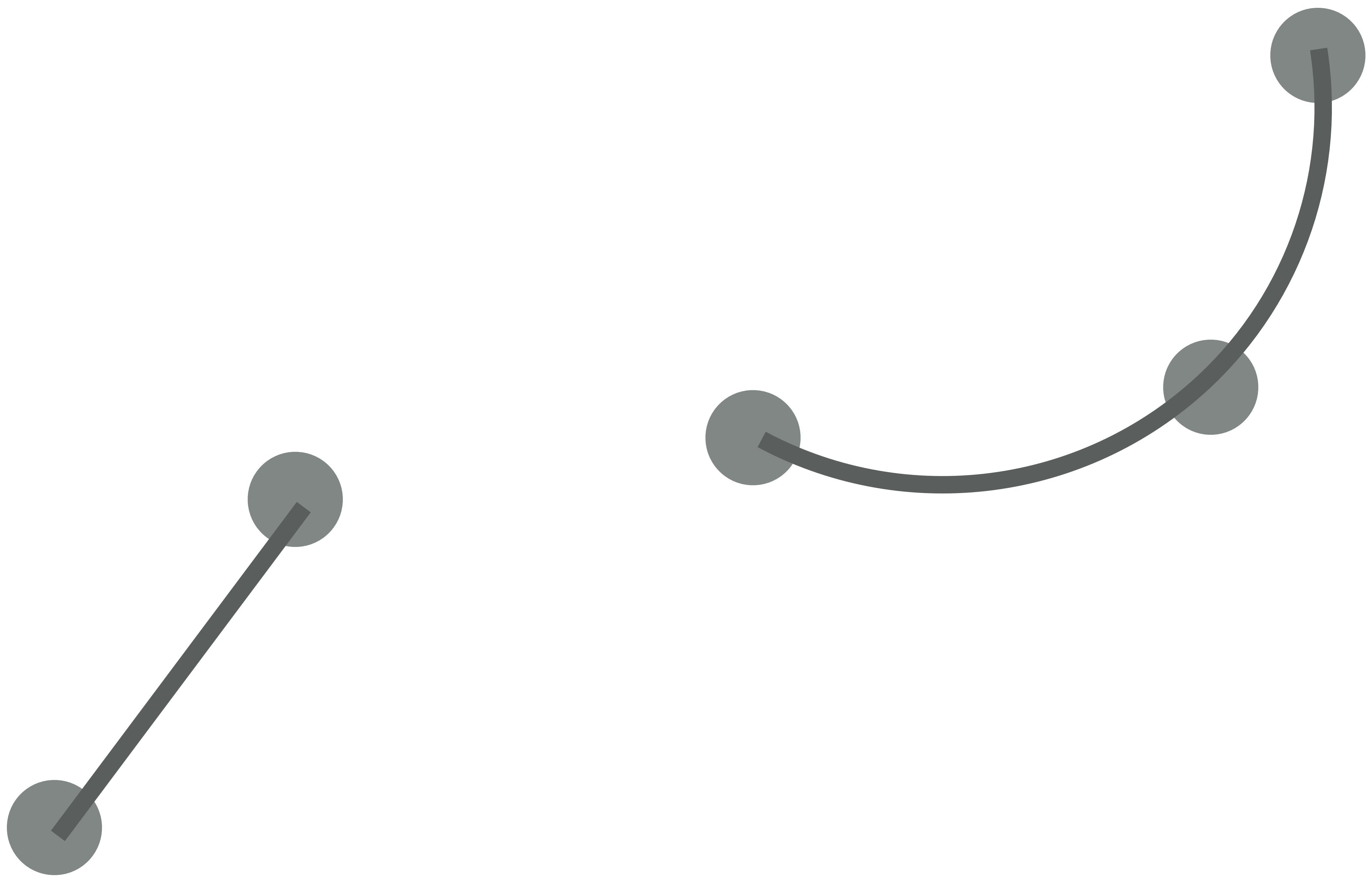
✗ Connectivity?

AN OPTION: FINITE SET OF POINTS



- ✓ Easy to store
- ✗ Connectivity?

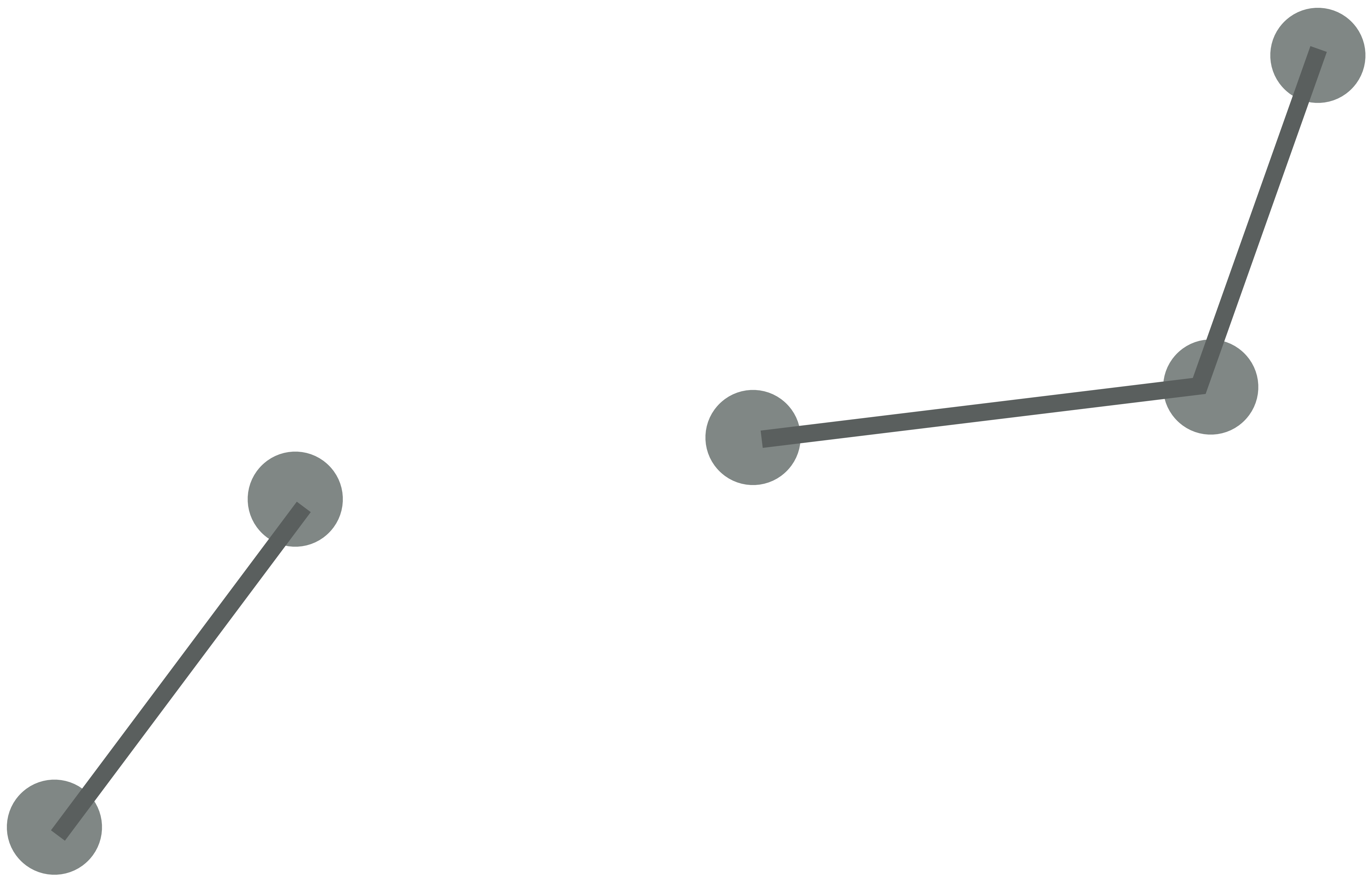
AN OPTION: FINITE SET OF POINTS



✓ Easy to store

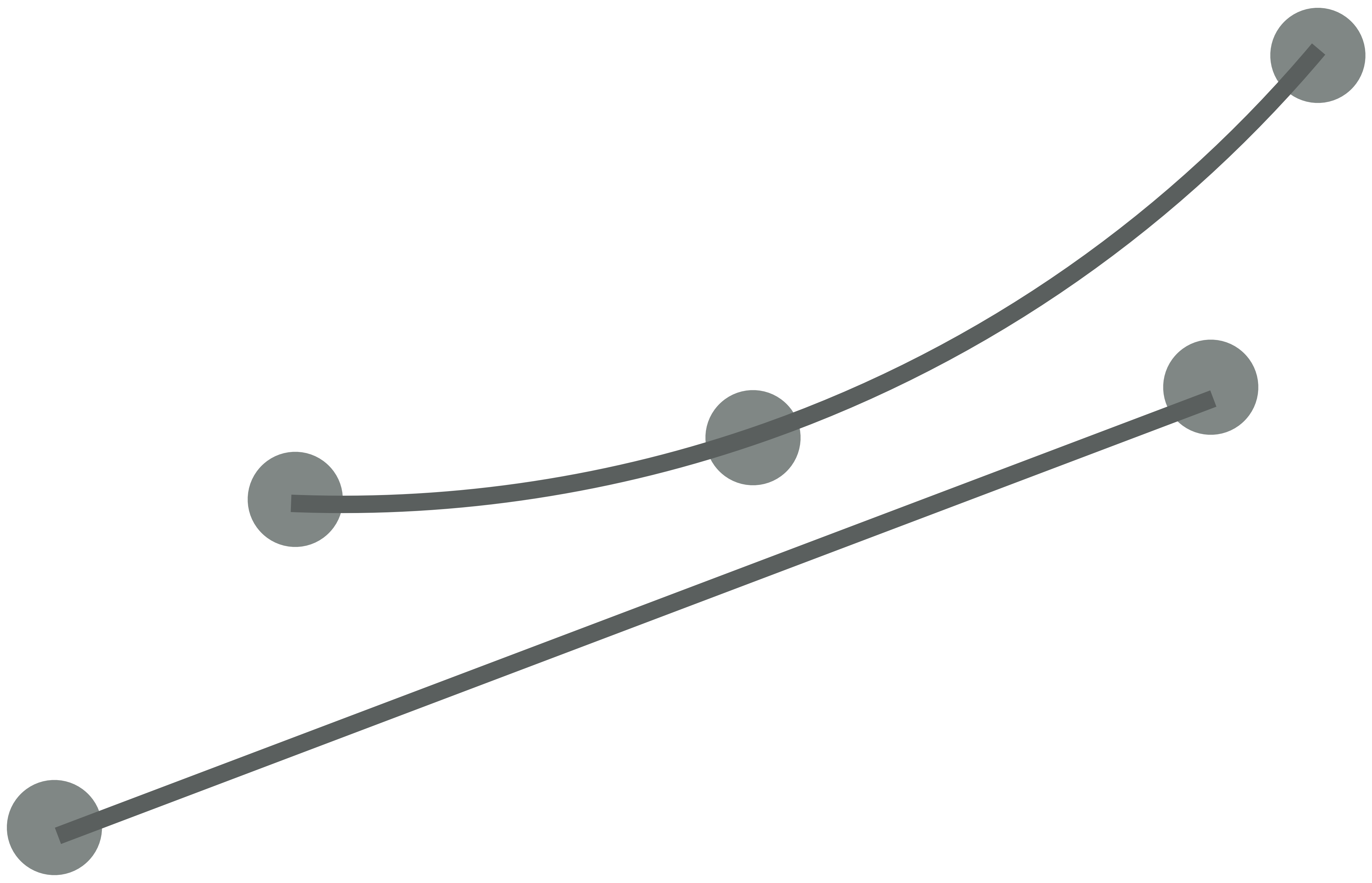
✗ Connectivity?

AN OPTION: FINITE SET OF POINTS



- ✓ Easy to store
- ✗ Connectivity?

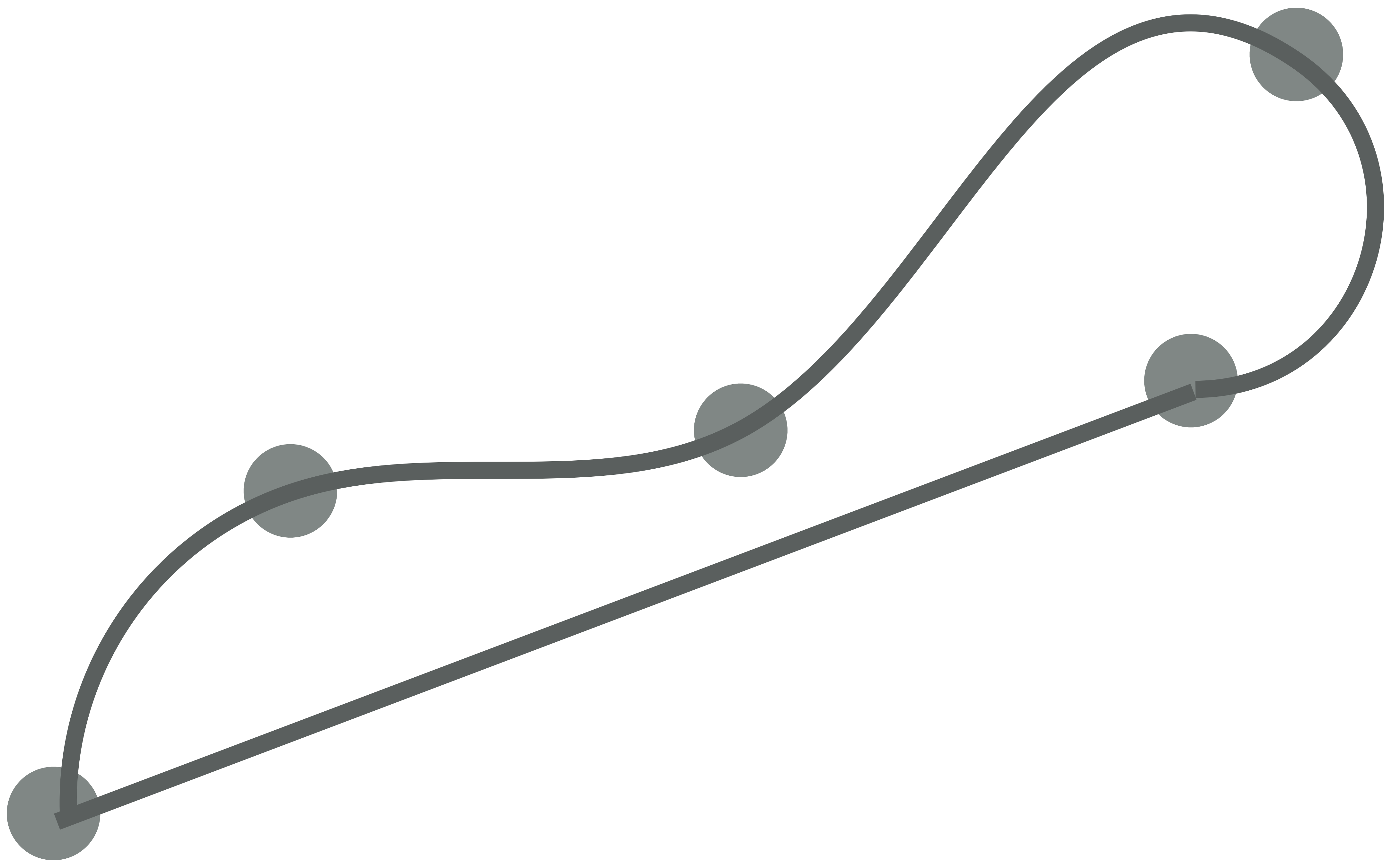
AN OPTION: FINITE SET OF POINTS



✓ Easy to store

✗ Connectivity?

AN OPTION: FINITE SET OF POINTS

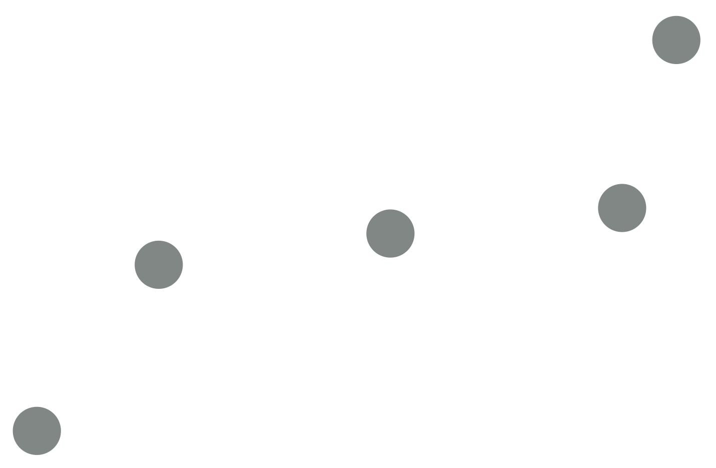


✓ Easy to store

✗ Connectivity?

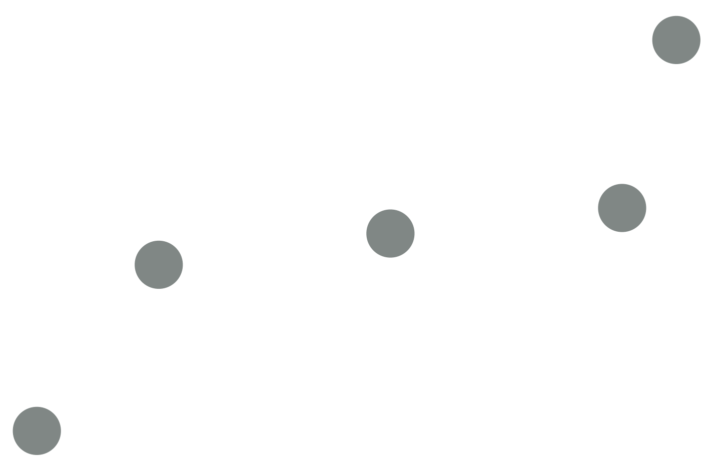
SHAPE REPRESENTATIONS

Set of points

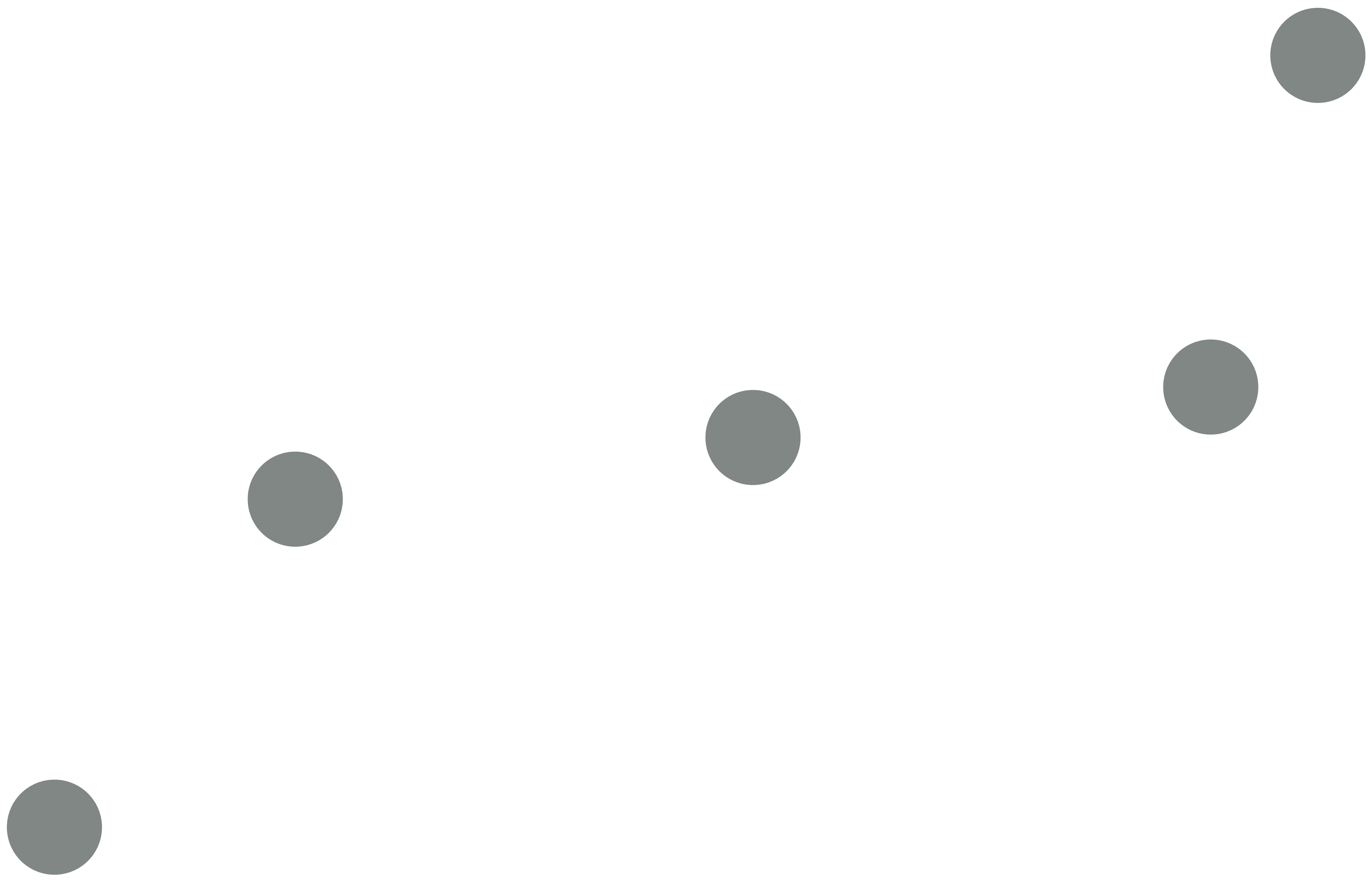


SHAPE REPRESENTATIONS

Point cloud



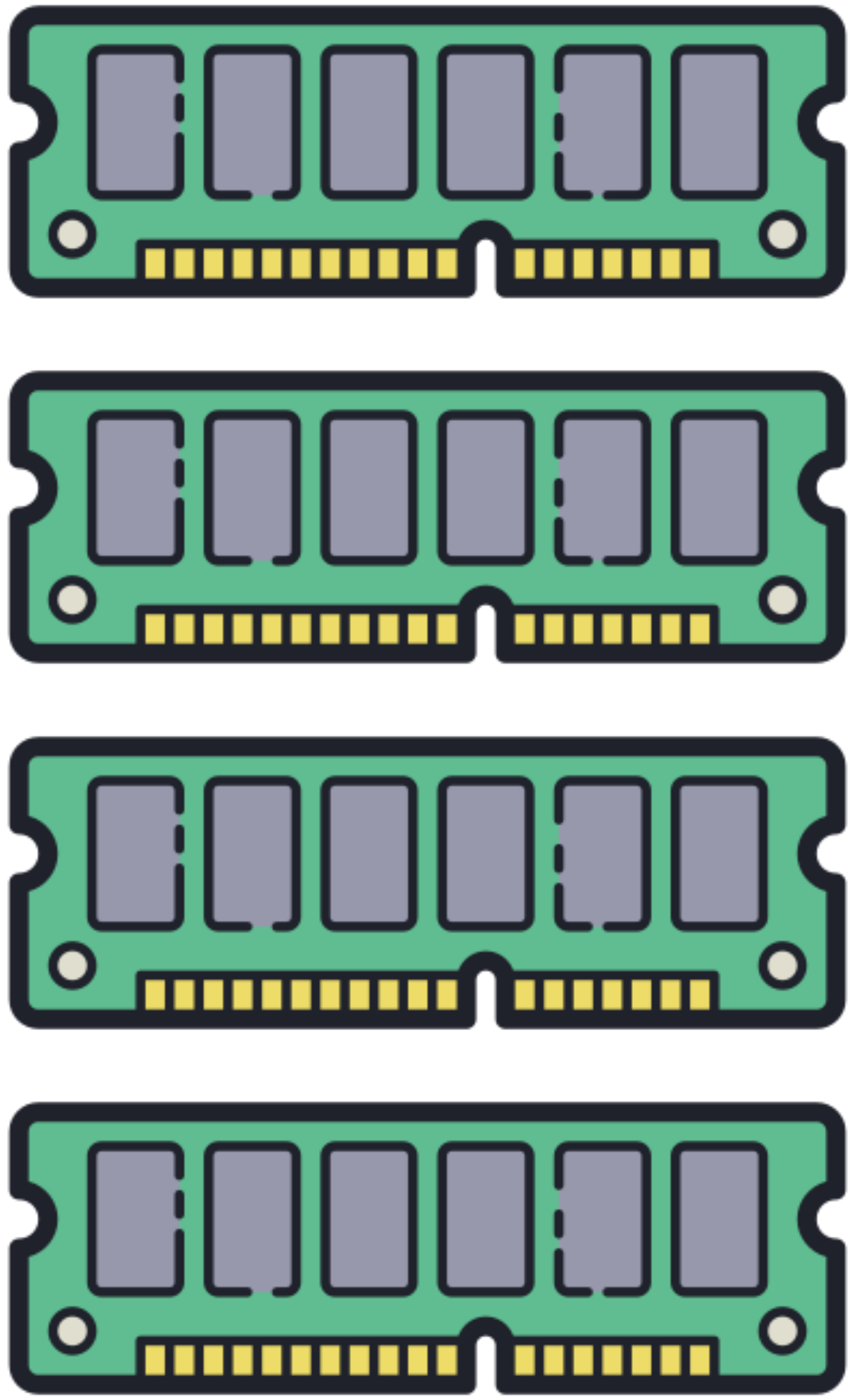
IMPROVING ON A POINT CLOUD



✓ Easy to store

✗ Connectivity?

AN OPTION: POINTS + CONNECTIVITY



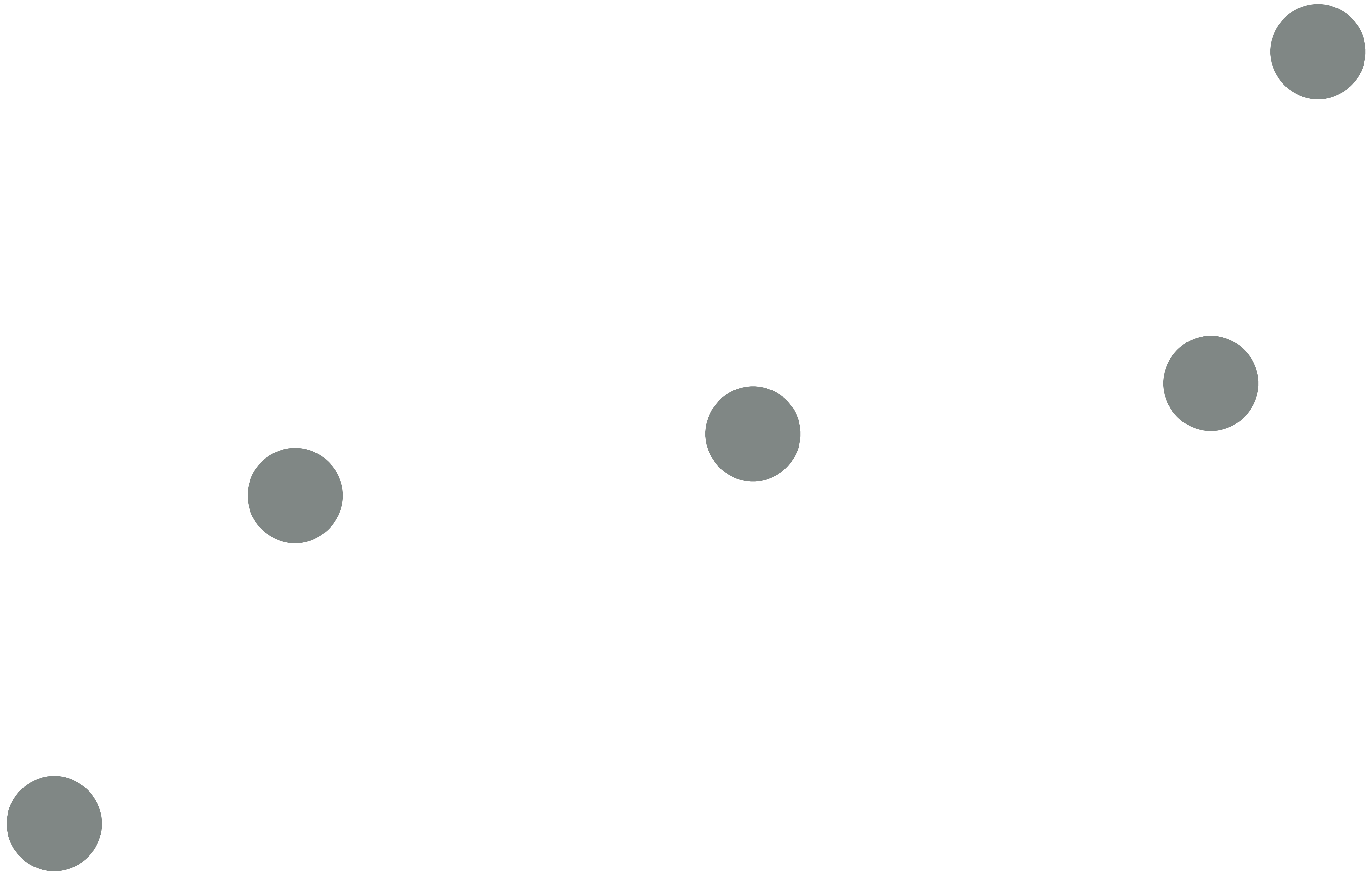
[0,0]

[0.322,0.28]

[0.456,0.420]

[0.662,0.4435]

[0.722,0.852]



AN OPTION: POINTS + CONNECTIVITY

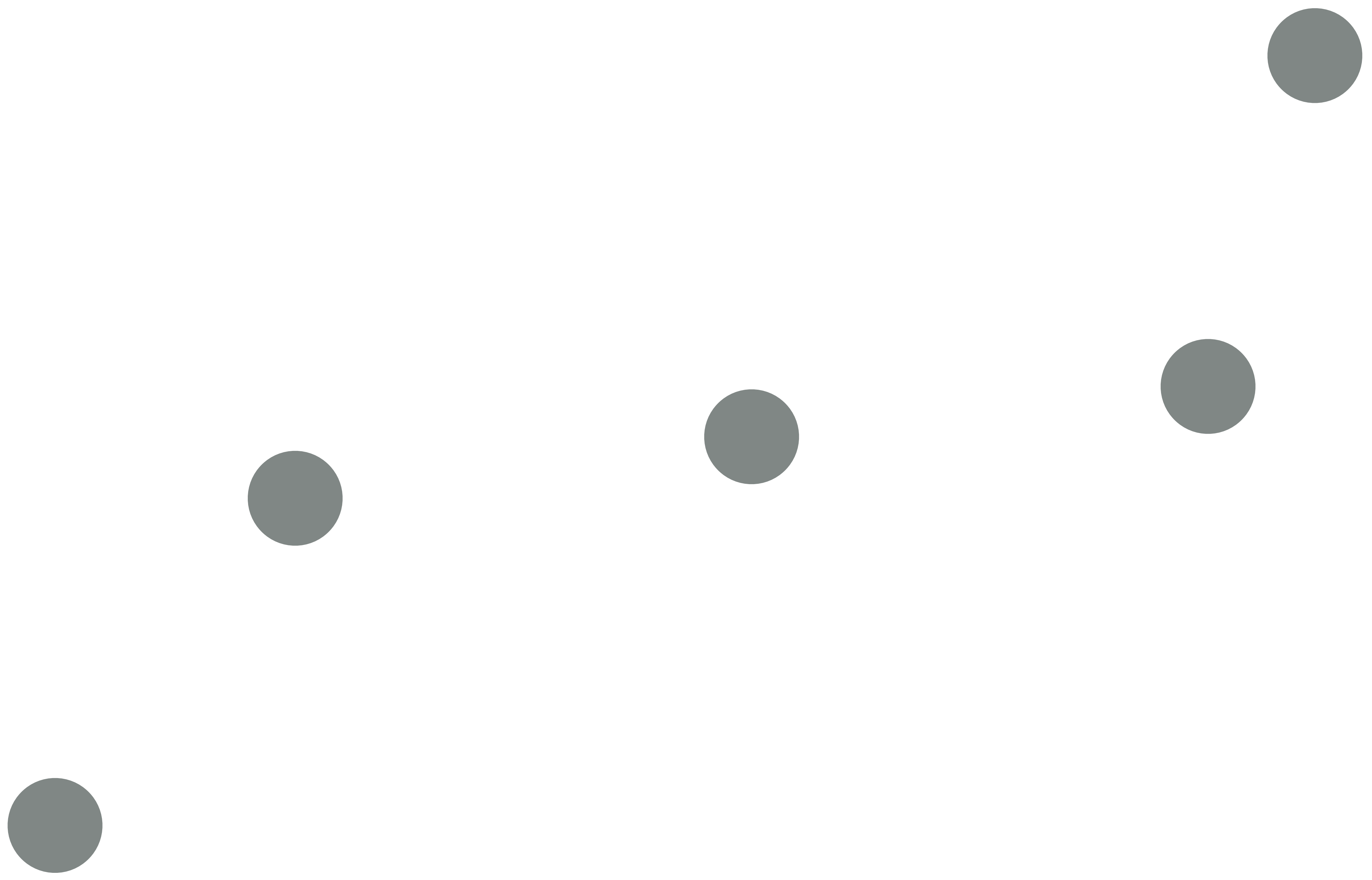
1[0,0]

2[0.322,0.28]

3[0.456,0.420]

4[0.662,0.4435]

5[0.722,0.852]



AN OPTION: POINTS + CONNECTIVITY

1[0,0]

2[0.322,0.28]

3[0.456,0.420]

4[0.662,0.4435]

5[0.722,0.852]

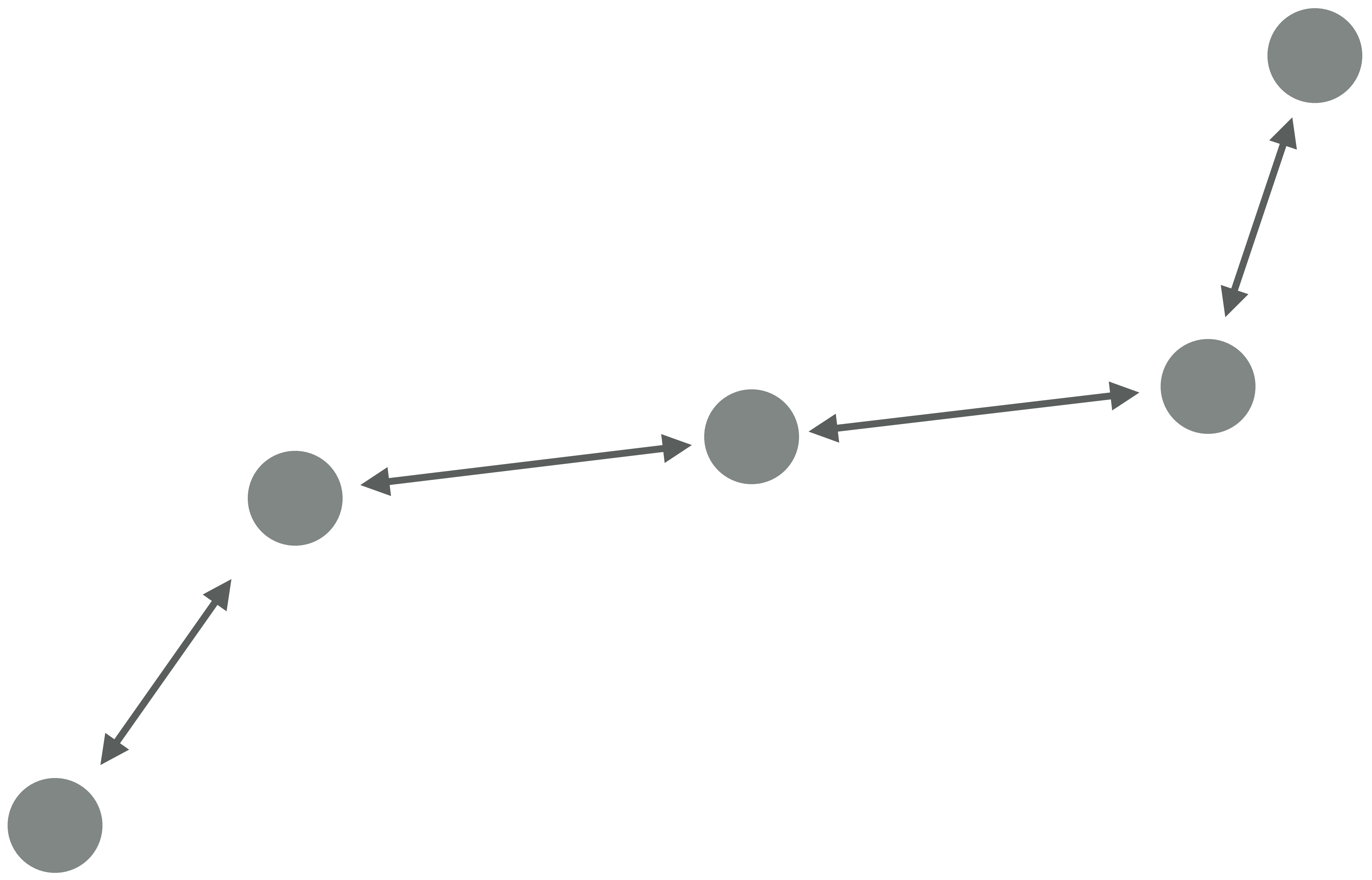
+

$1 \leftrightarrow 2$

$2 \leftrightarrow 3$

$3 \leftrightarrow 4$

$4 \leftrightarrow 5$



AN OPTION: POINTS + CONNECTIVITY

1[0,0]

2[0.322,0.28]

3[0.456,0.420]

4[0.662,0.4435]

5[0.722,0.852]

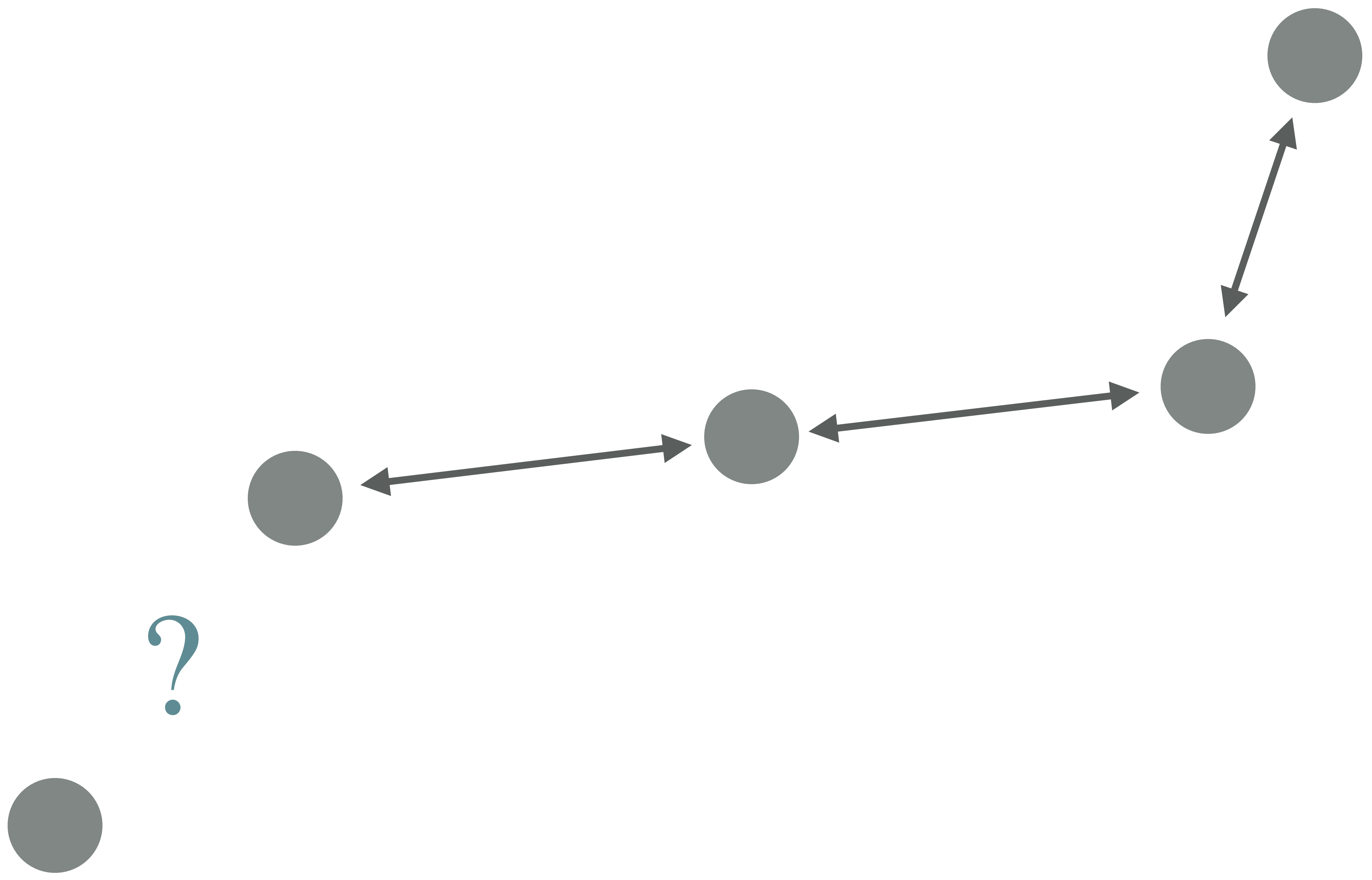
+

$1 \leftrightarrow 2$

$2 \leftrightarrow 3$

$3 \leftrightarrow 4$

$4 \leftrightarrow 5$



AN OPTION: POINTS + CONNECTIVITY

1[0,0]

2[0.322,0.28]

3[0.456,0.420]

4[0.662,0.4435]

5[0.722,0.852]

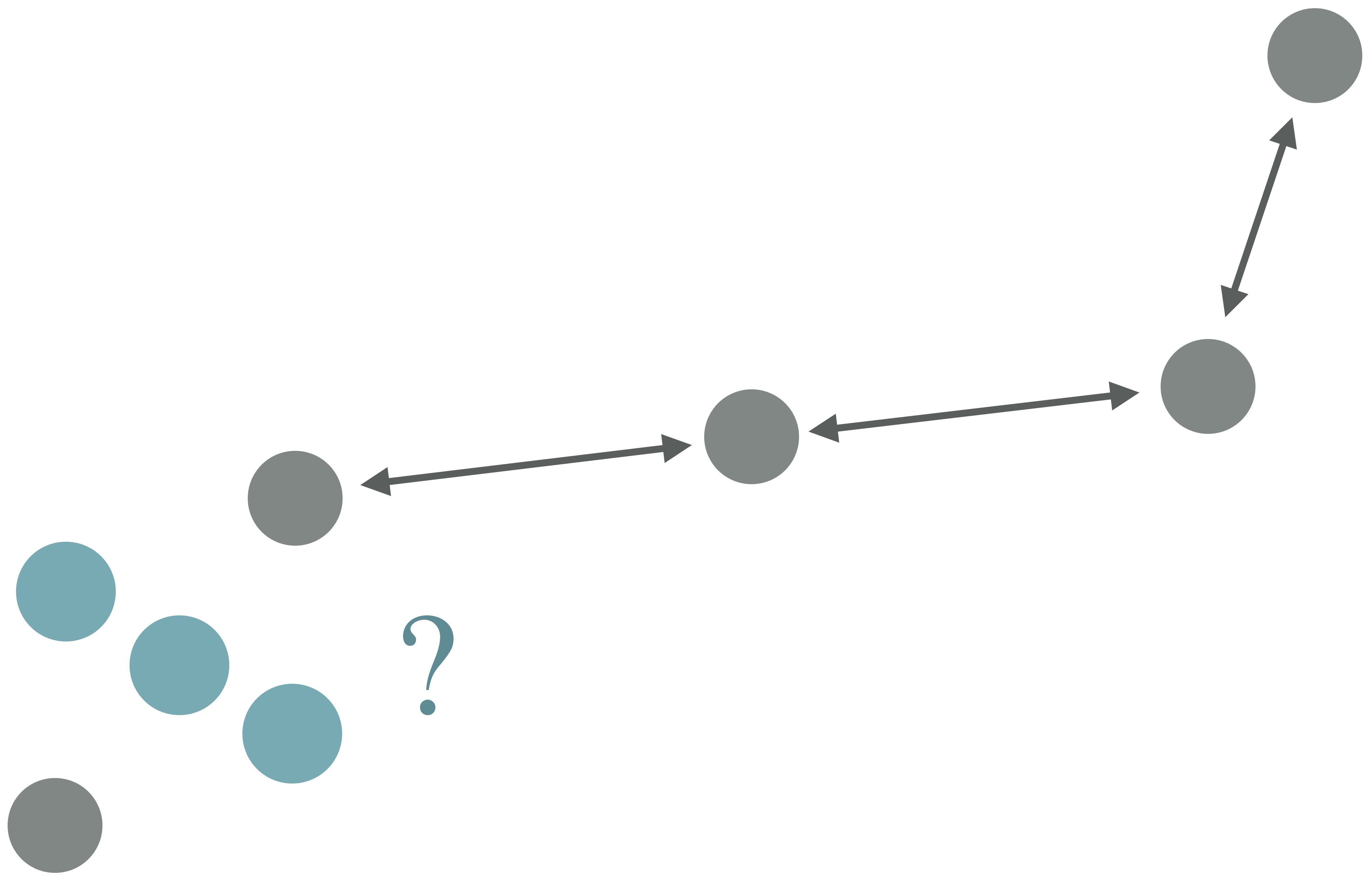
+

$1 \leftrightarrow 2$

$2 \leftrightarrow 3$

$3 \leftrightarrow 4$

$4 \leftrightarrow 5$



AN OPTION: POINTS + CONNECTIVITY

1[0,0]

2[0.322,0.28]

3[0.456,0.420]

4[0.662,0.4435]

5[0.722,0.852]

+

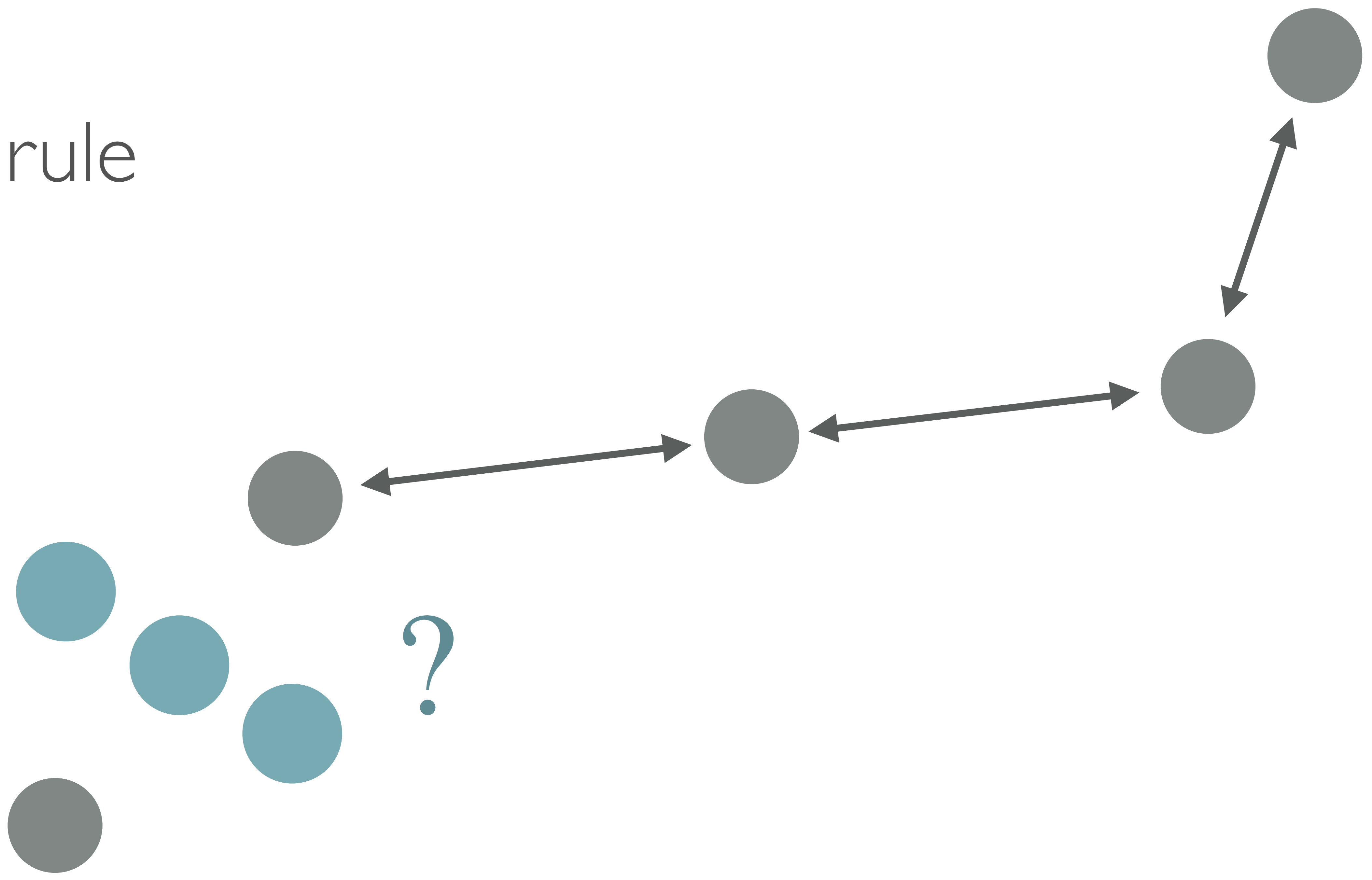
$1 \leftrightarrow 2$

$2 \leftrightarrow 3$

$3 \leftrightarrow 4$

$4 \leftrightarrow 5$

+ Interpolation rule



AN OPTION: POINTS + CONNECTIVITY

1[0,0]

2[0.322,0.28]

3[0.456,0.420]

4[0.662,0.4435]

5[0.722,0.852]

+

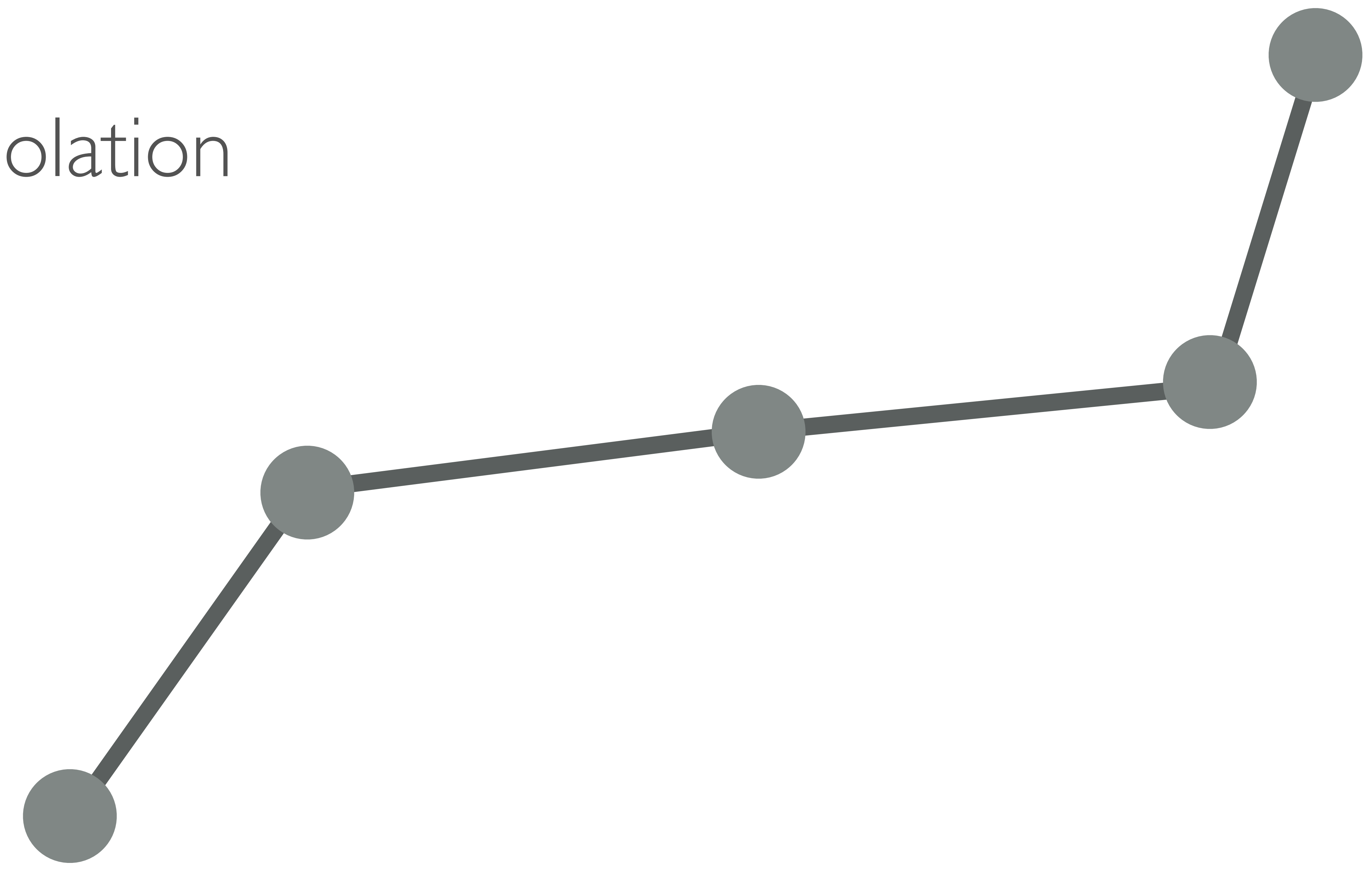
$1 \leftrightarrow 2$

$2 \leftrightarrow 3$

$3 \leftrightarrow 4$

$4 \leftrightarrow 5$

+ Linear interpolation



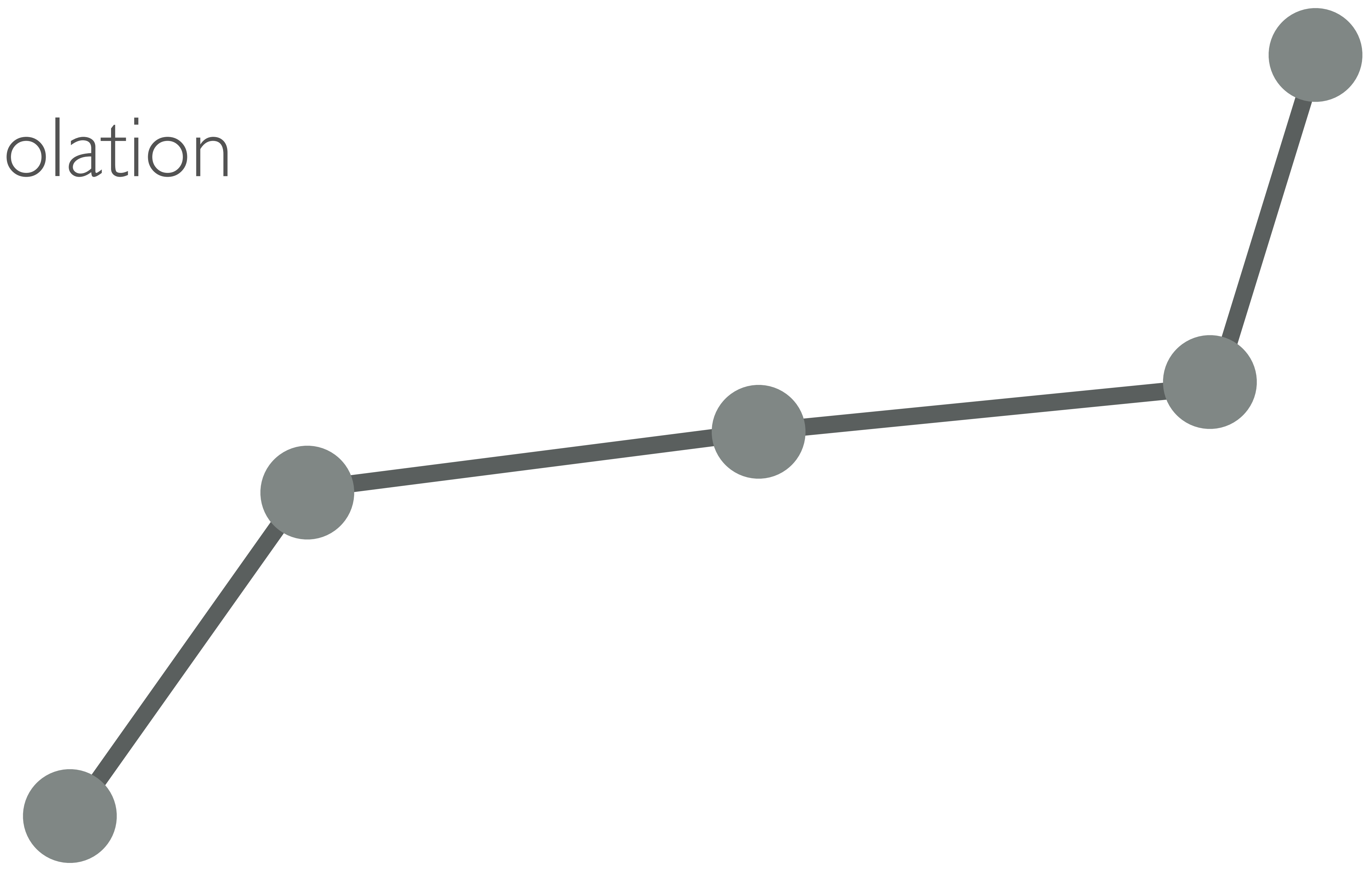
AN OPTION: POLYLINE

1[0,0]
2[0.322,0.28]
3[0.456,0.420]
4[0.662,0.4435]
5[0.722,0.852]

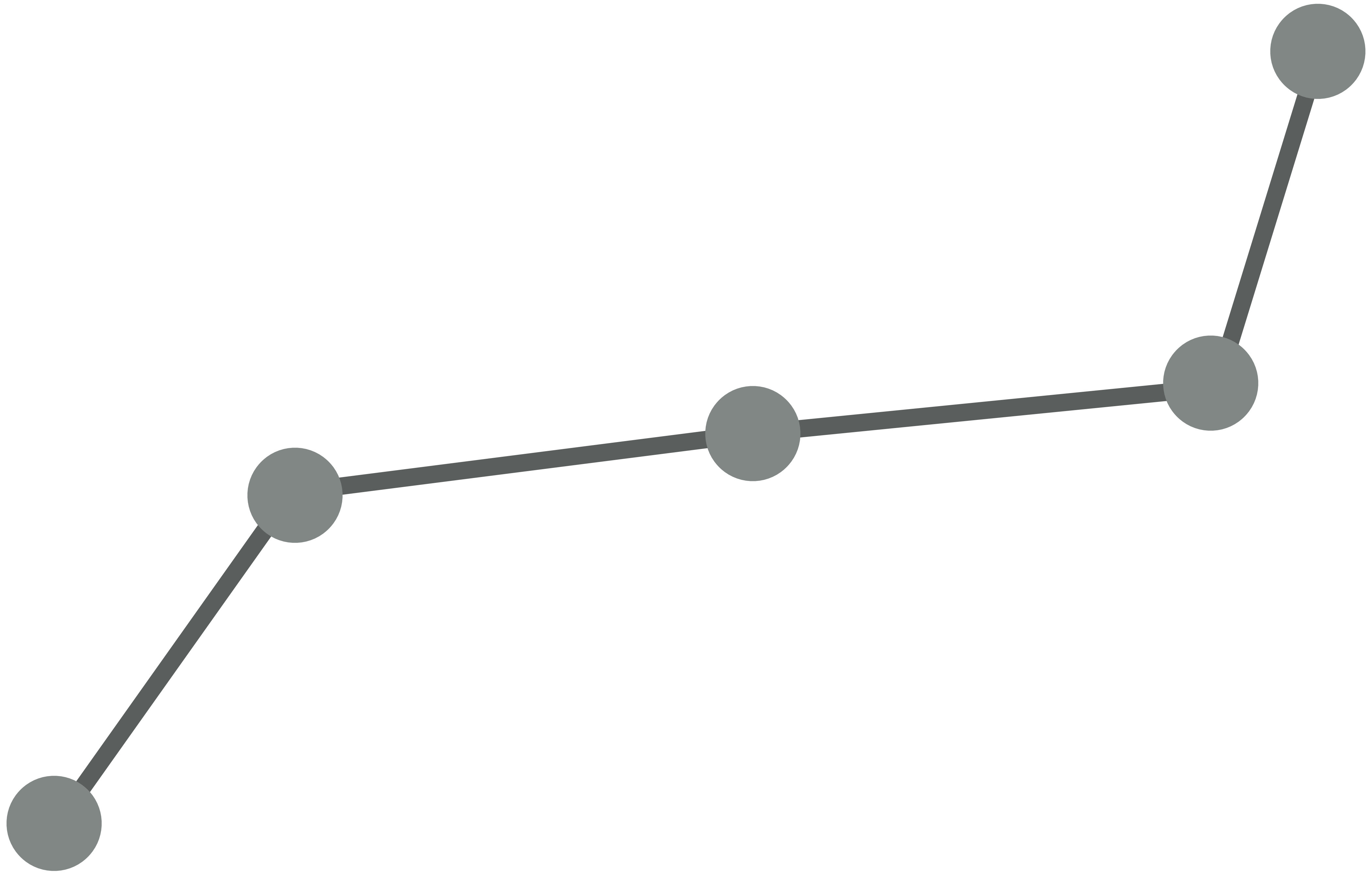
+

1 ↔ 2
2 ↔ 3
3 ↔ 4
4 ↔ 5

+ Linear interpolation

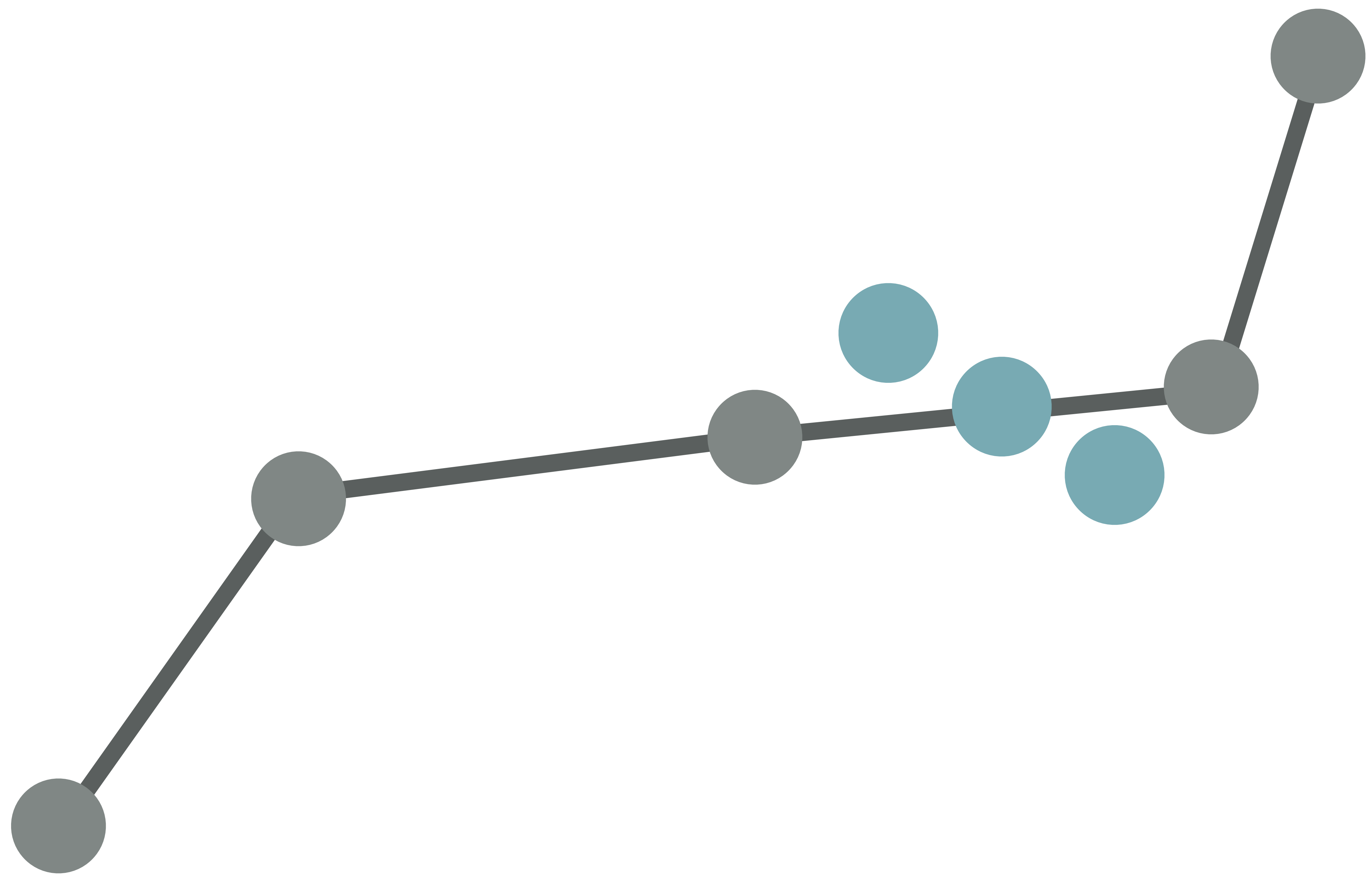


AN OPTION: POLYLINE



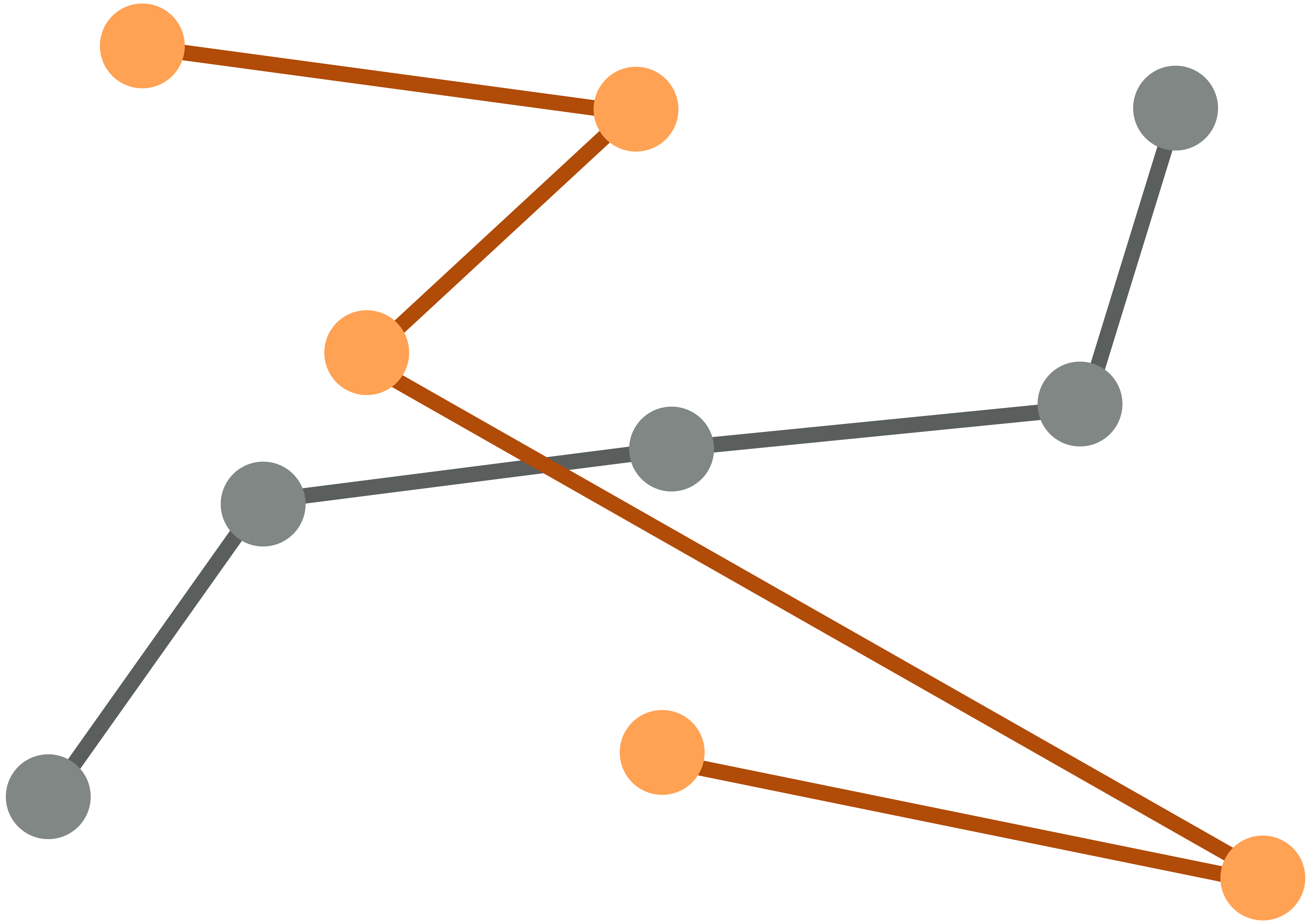
✓ Easy to store and plot

AN OPTION: POLYLINE



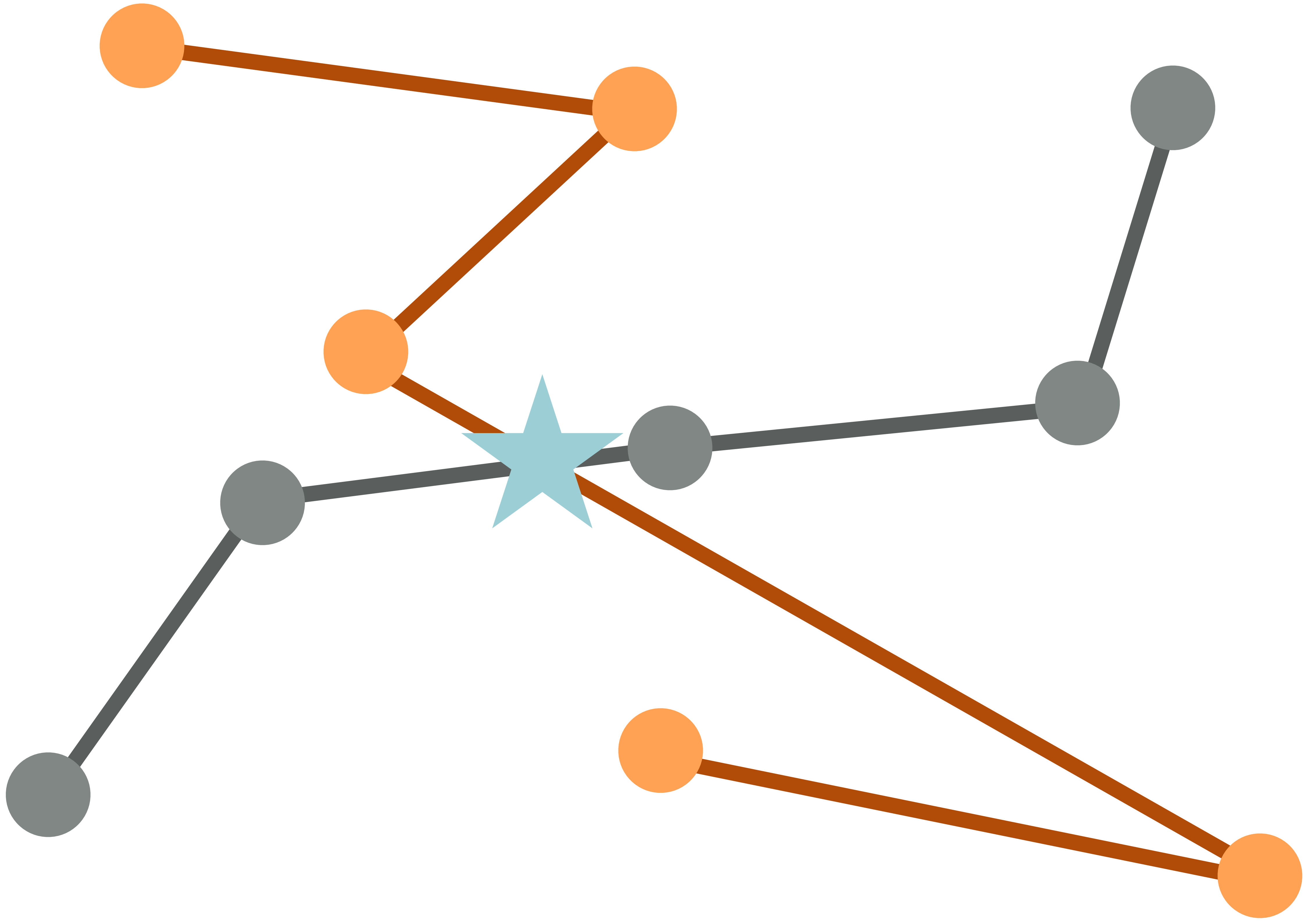
- ✓ Easy to store and plot
- ✓ Easy to query

AN OPTION: POLYLINE



- Easy to store and plot
- Easy to query
- Easy intersections

AN OPTION: POLYLINE



- Easy to store and plot
- Easy to query
- Easy intersections

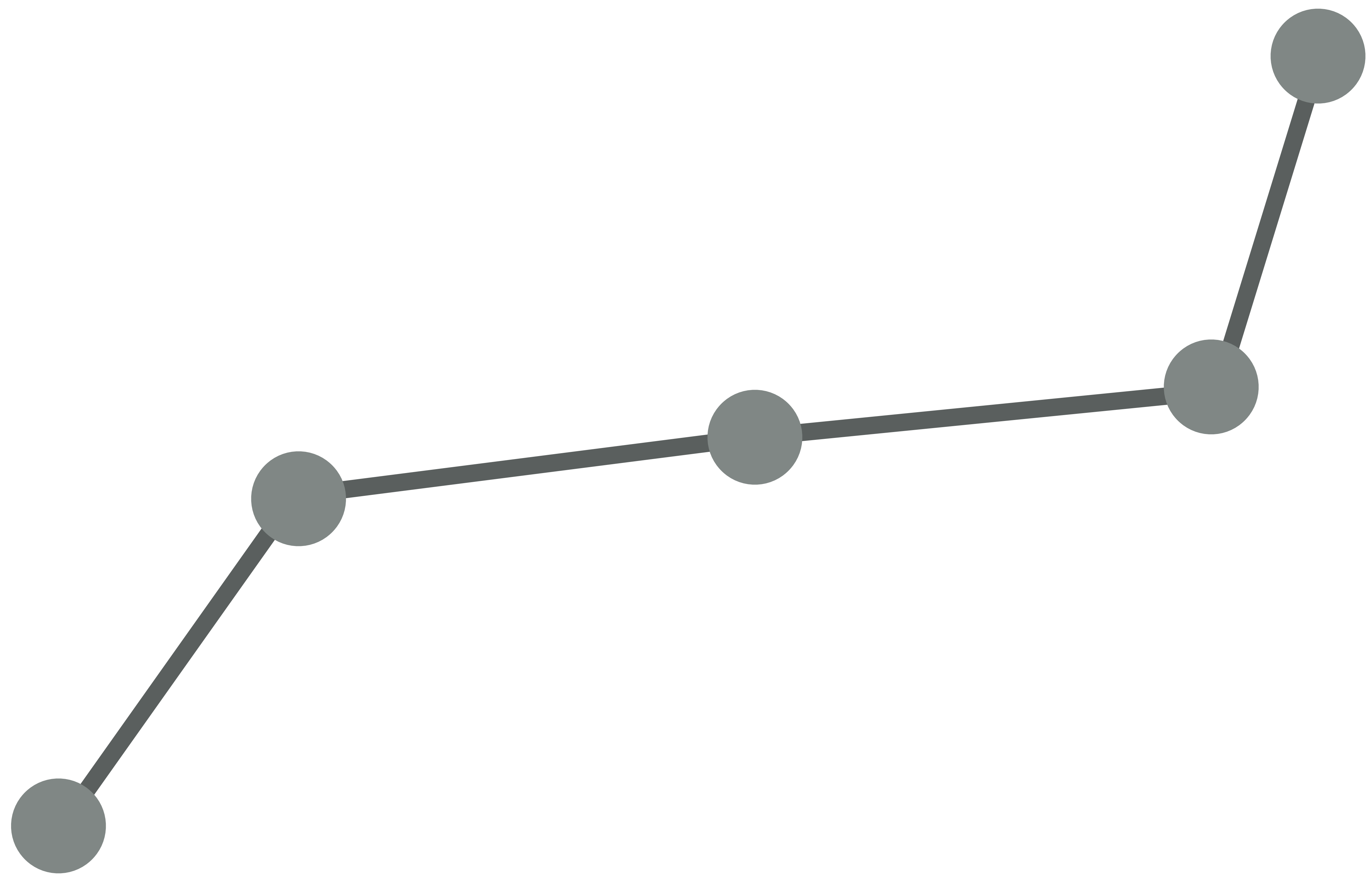
AN OPTION: POLYLINE

- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections

AN OPTION: POLYLINE

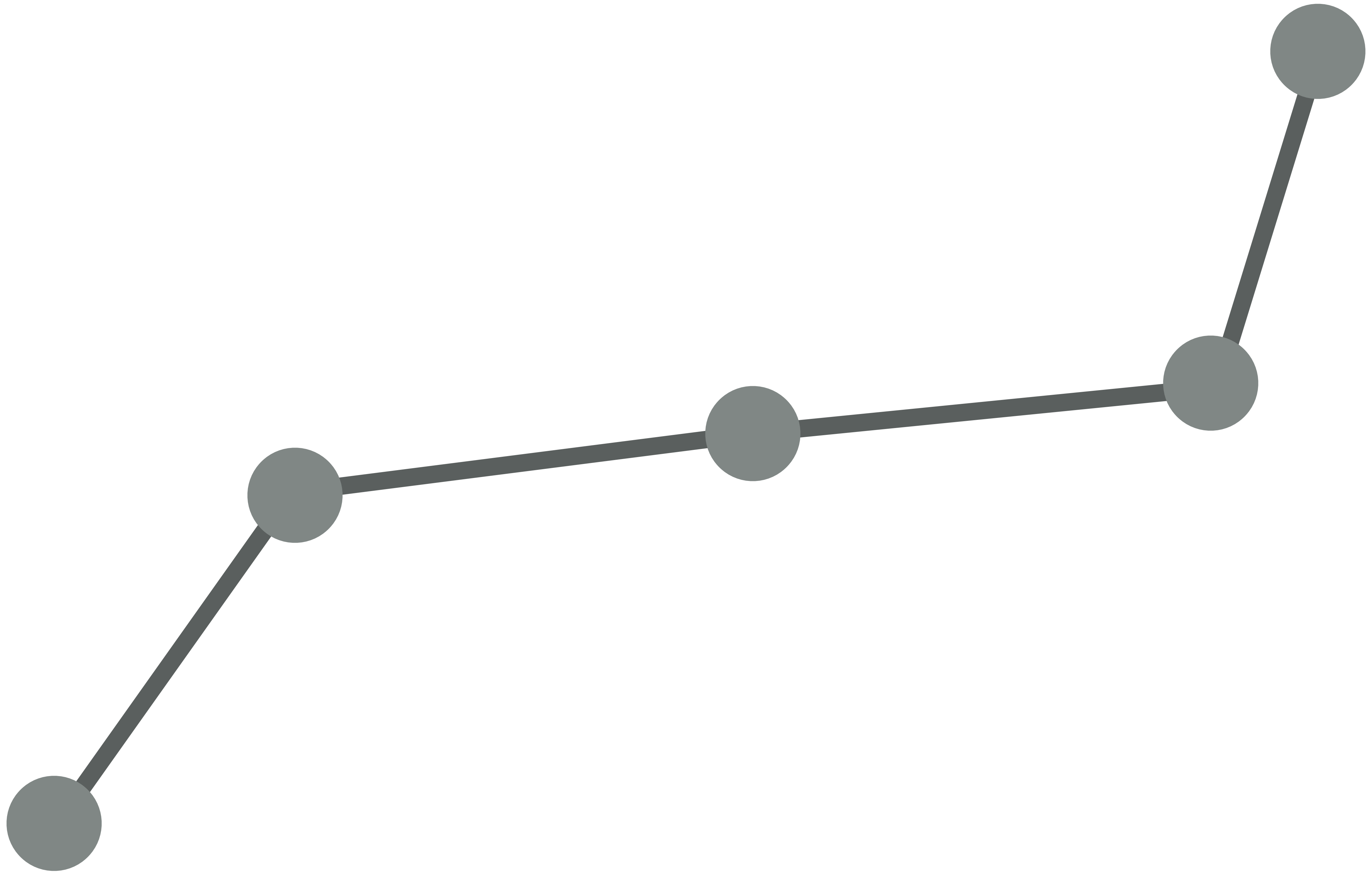
- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections

AN OPTION: POLYLINE



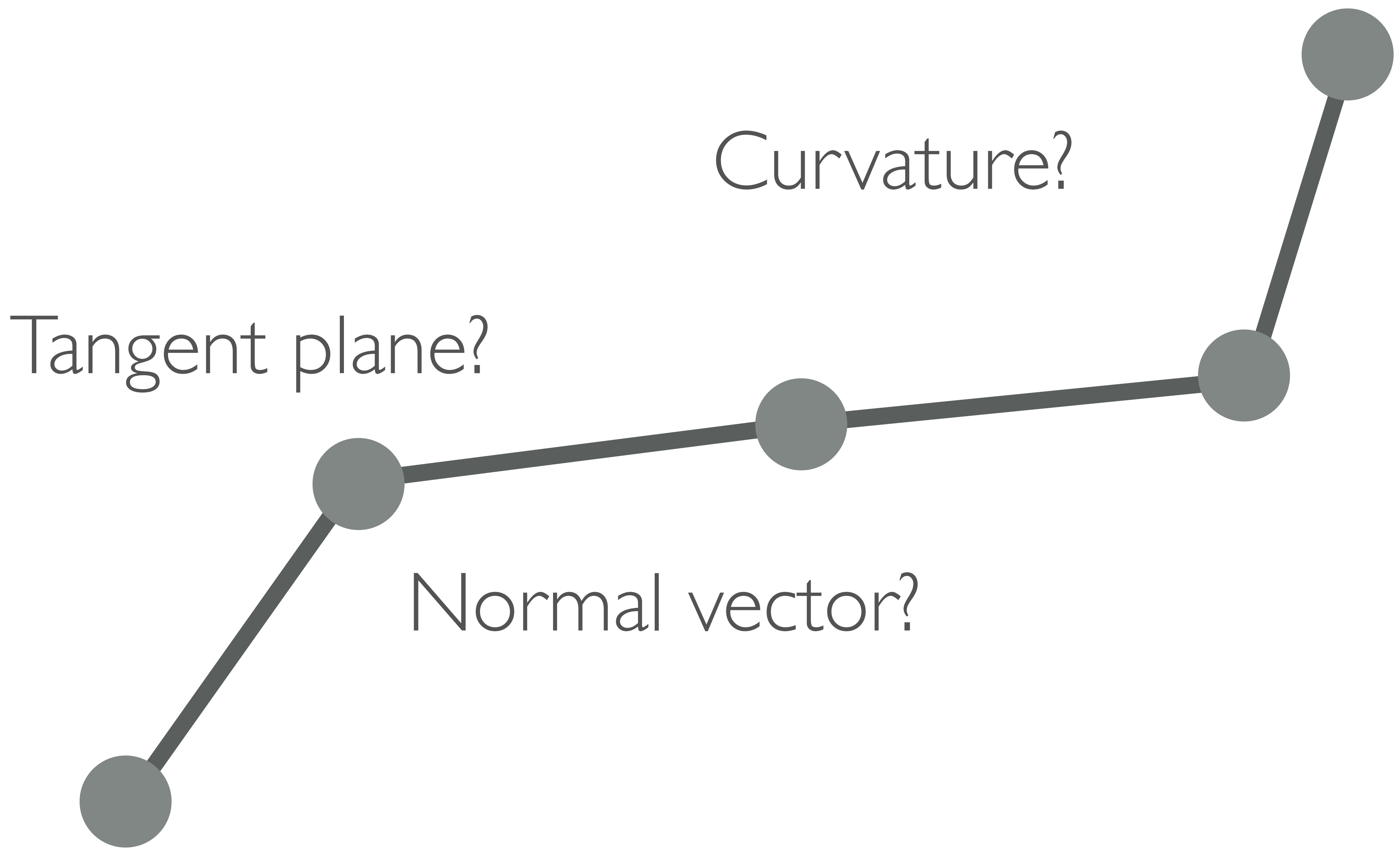
- Easy to store and plot
- Easy to query
- Easy intersections

AN OPTION: POLYLINE



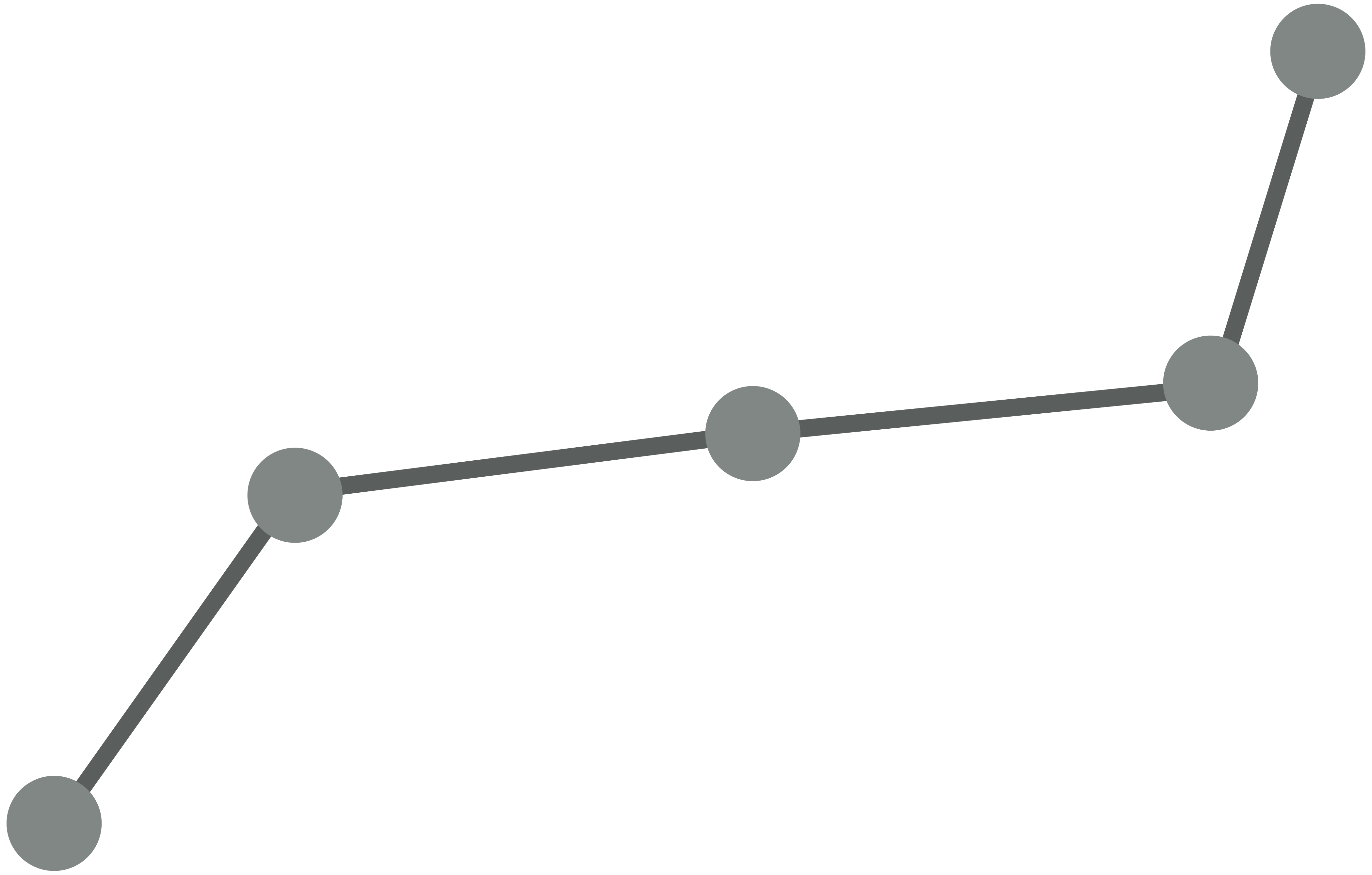
- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?

AN OPTION: POLYLINE



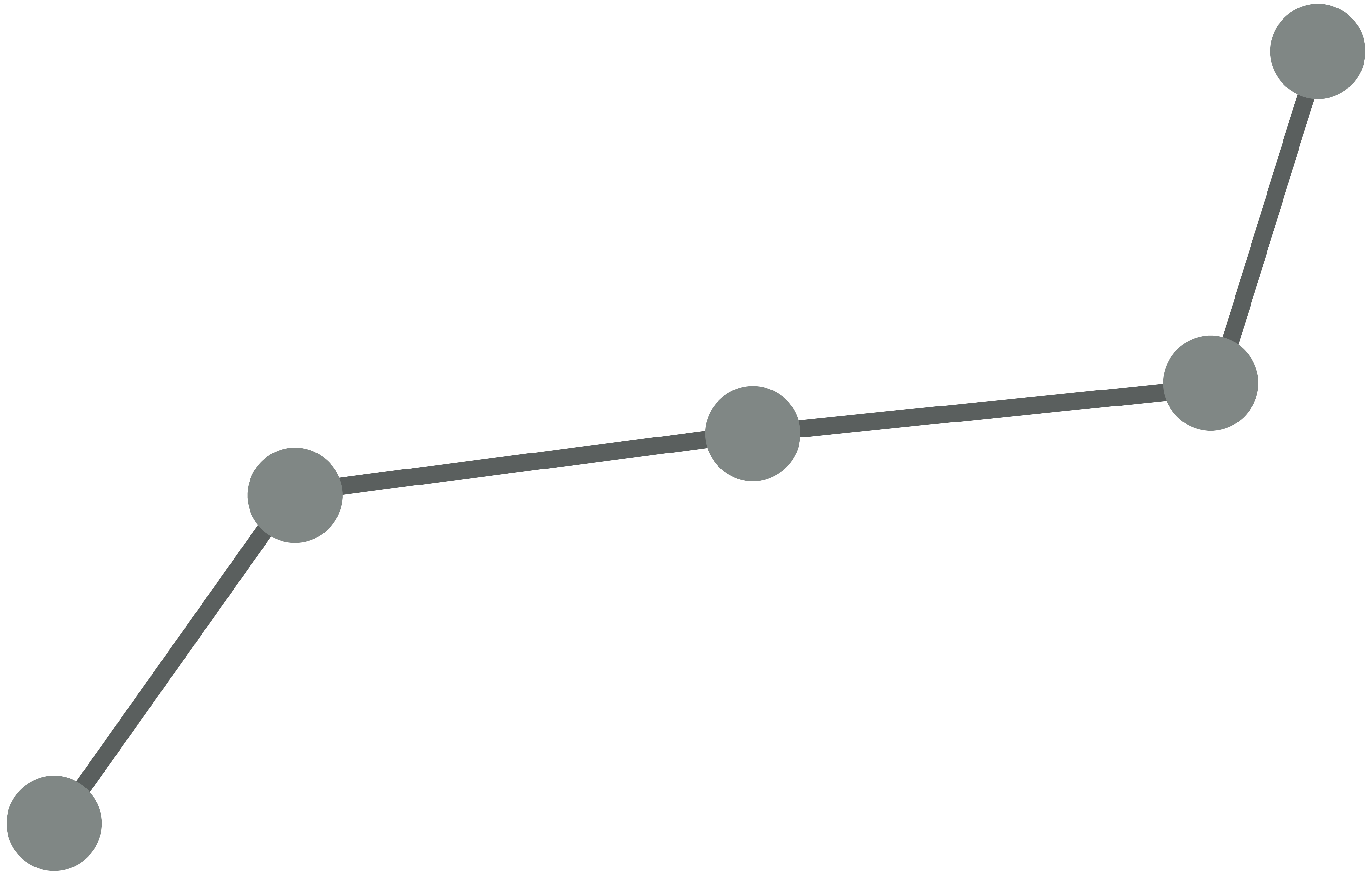
- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?

AN OPTION: POLYLINE



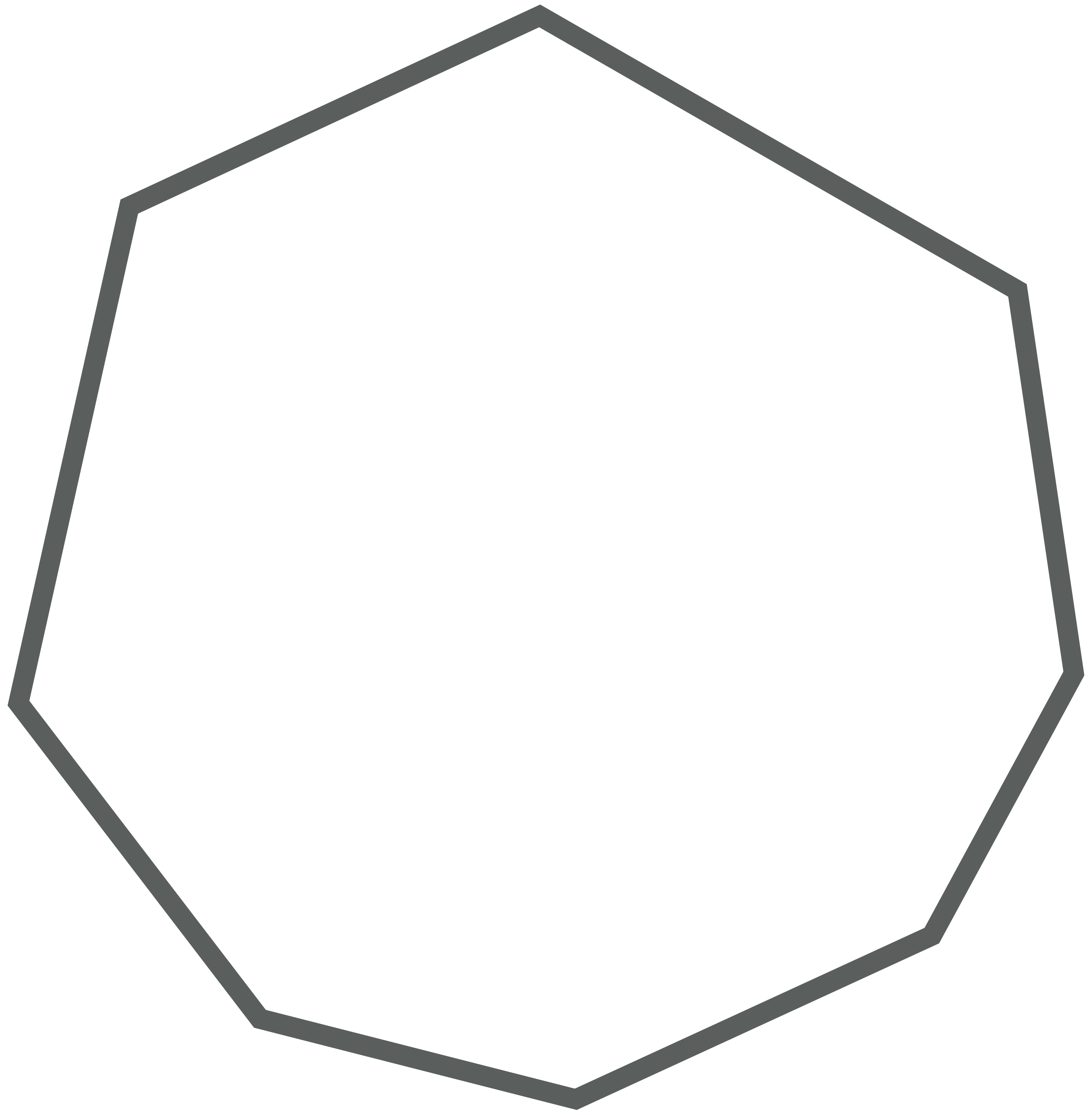
- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?

AN OPTION: POLYLINE



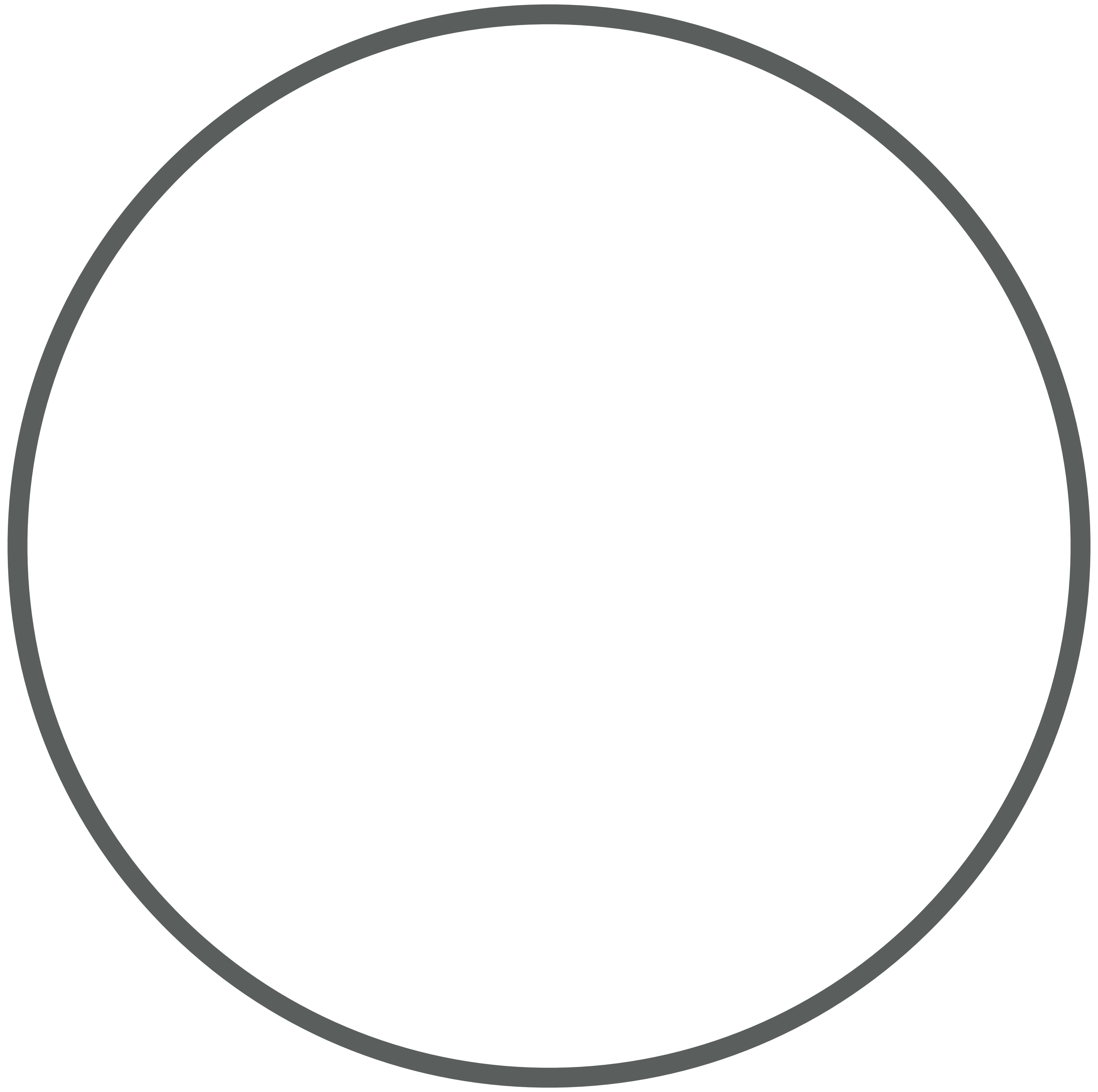
- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad!

AN OPTION: POLYLINE



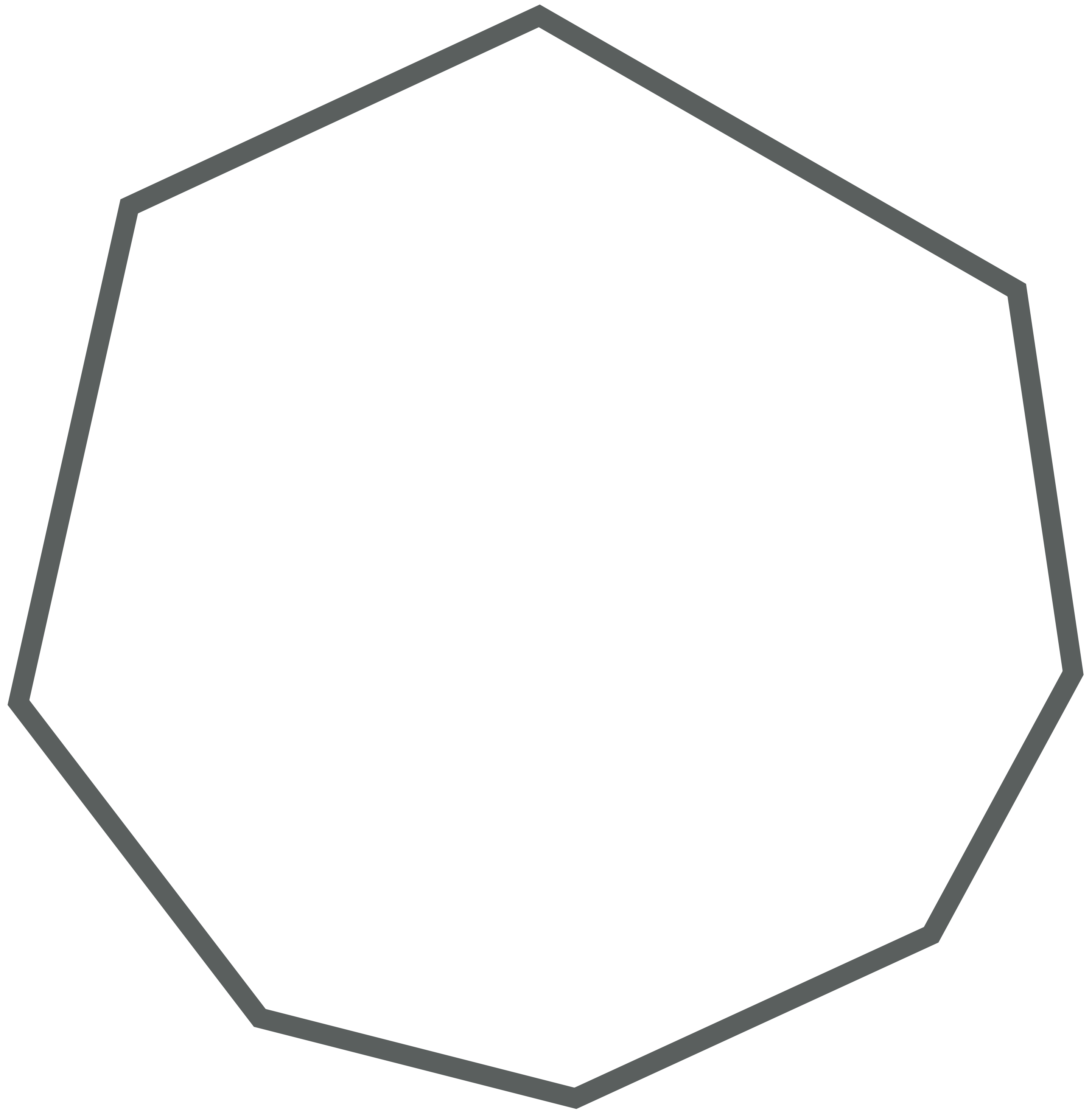
- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad!

AN OPTION: POLYLINE



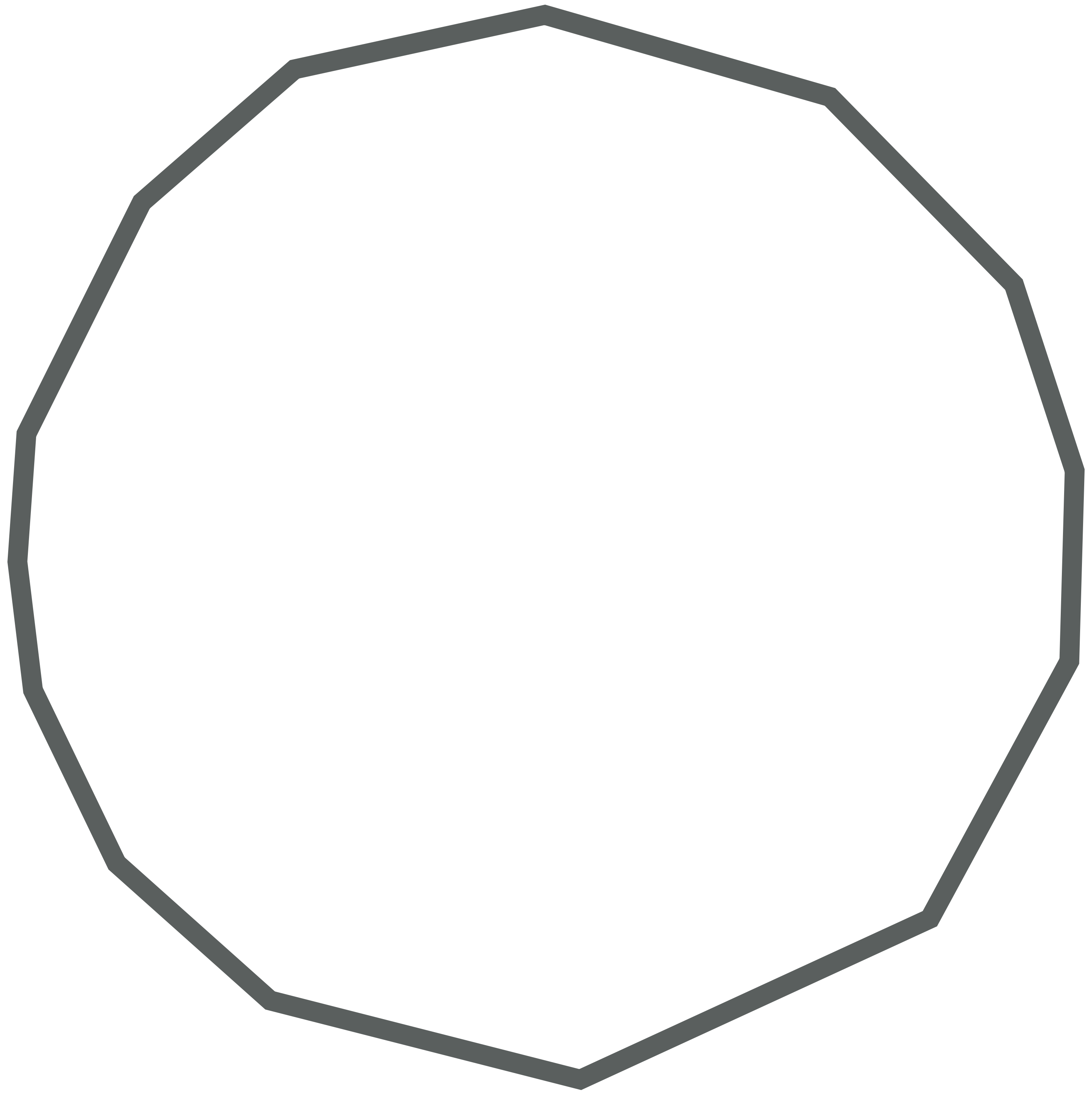
- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad!

AN OPTION: POLYLINE



- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad!

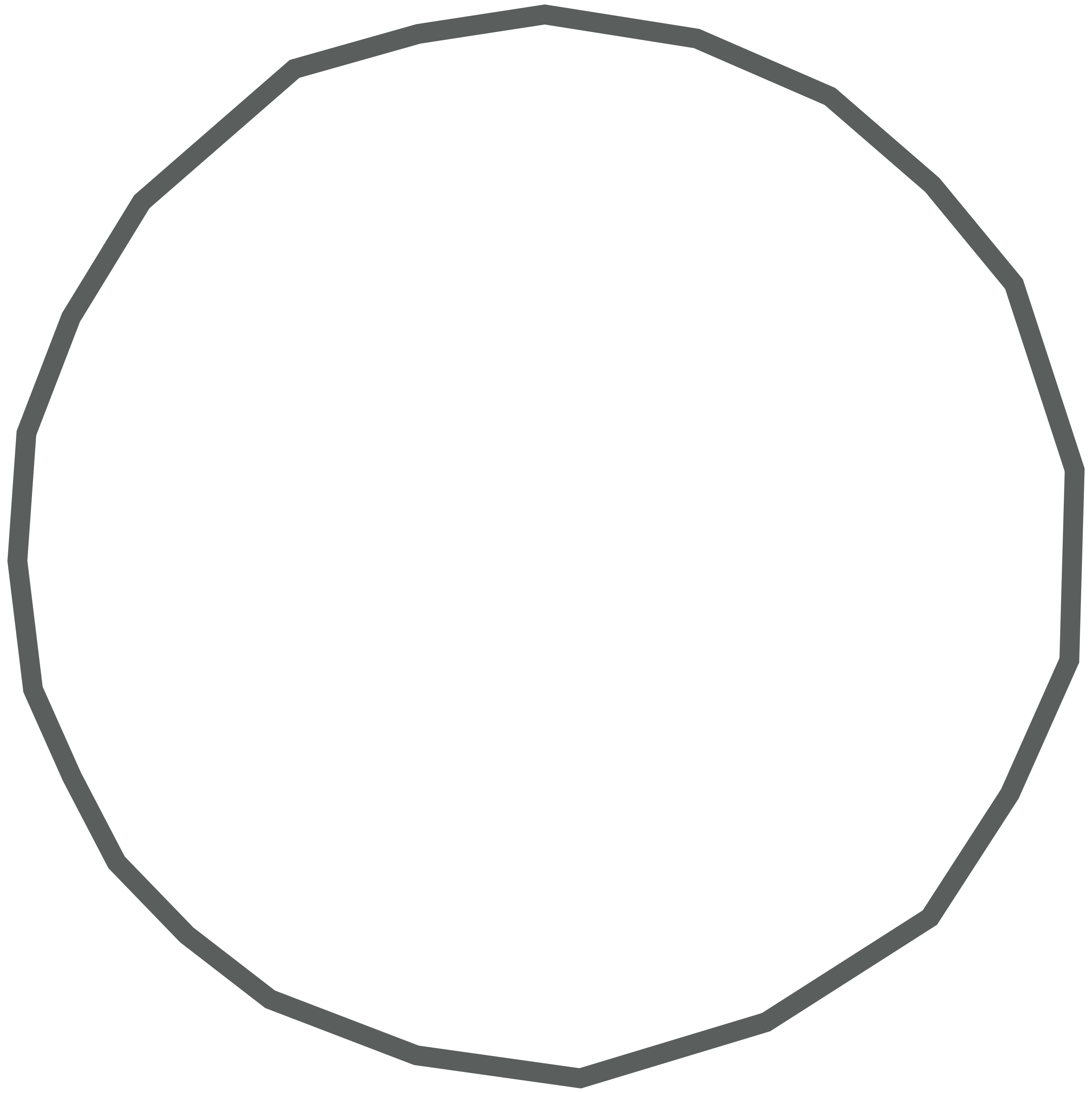
AN OPTION: POLYLINE



- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad!

AN OPTION: POLYLINE

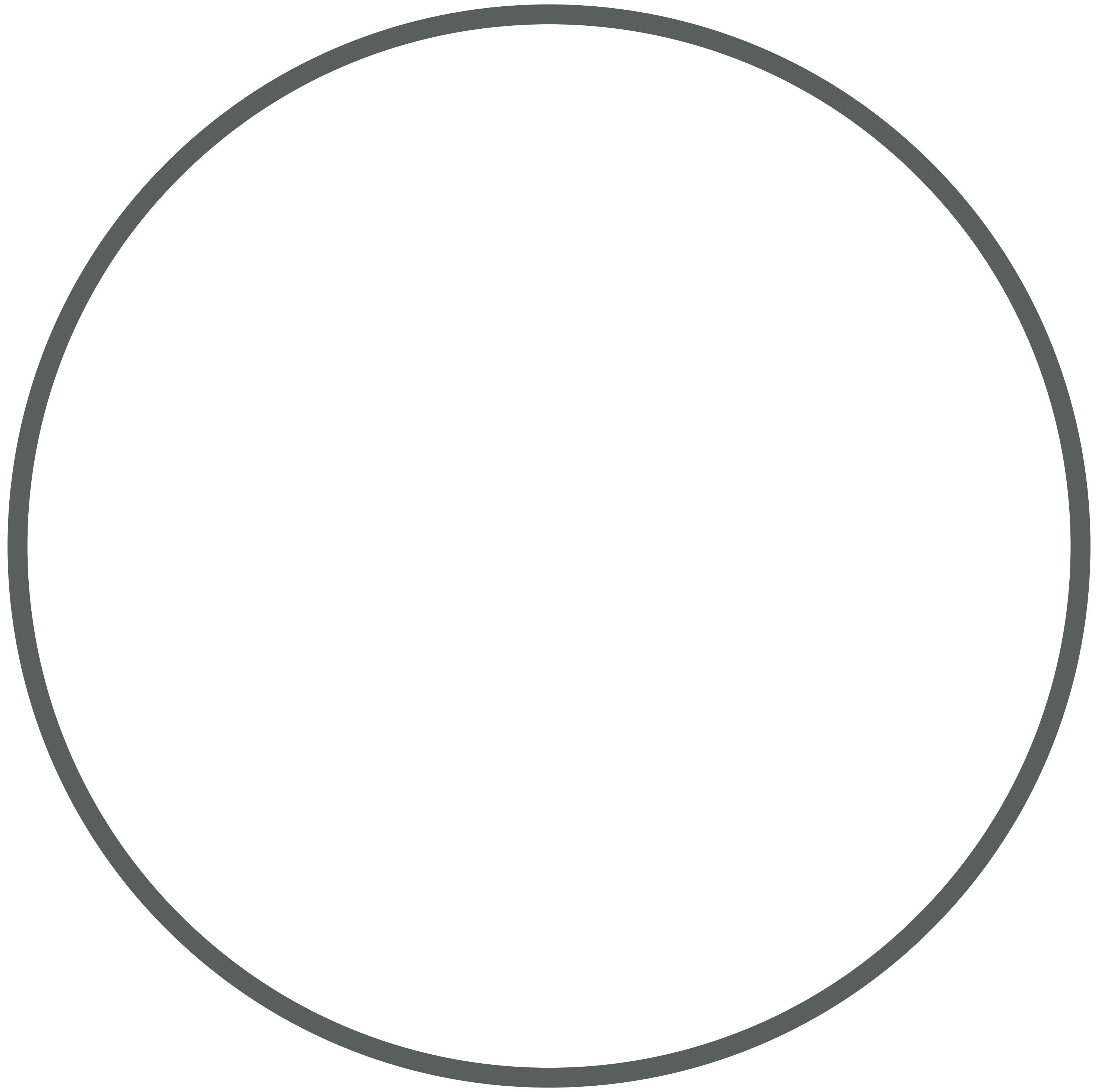
100s of points



- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad!

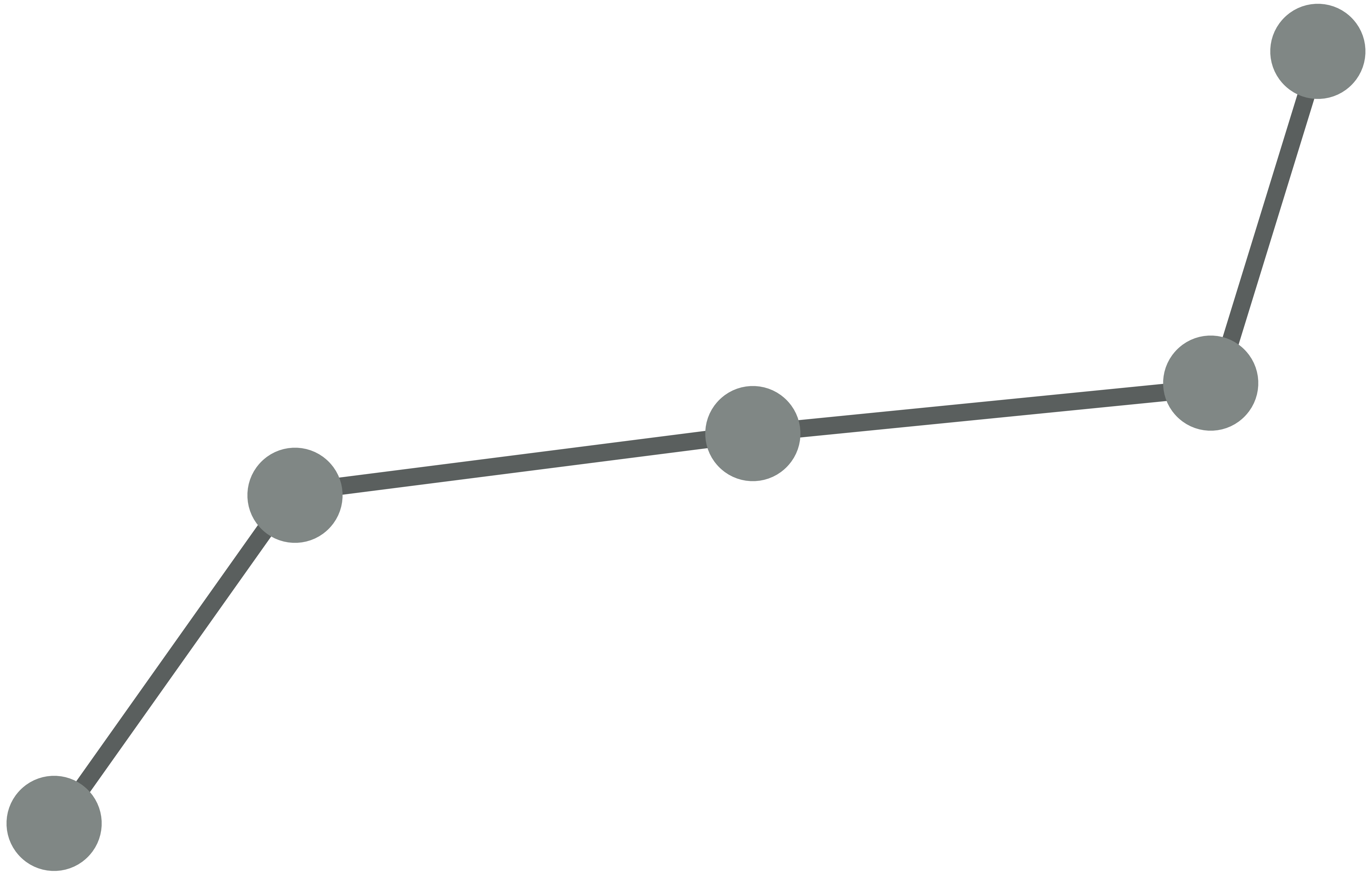
AN OPTION: POLYLINE

$$\gamma(t) = (cos(t), sin(t)) \quad t \in [0, 2\pi)$$



- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad!

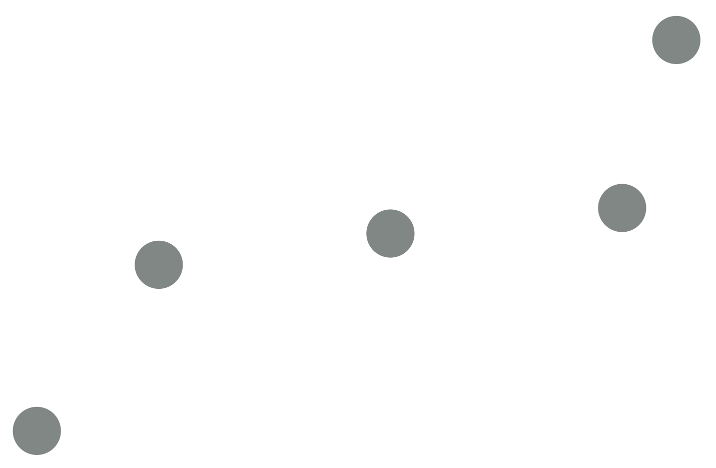
AN OPTION: POLYLINE



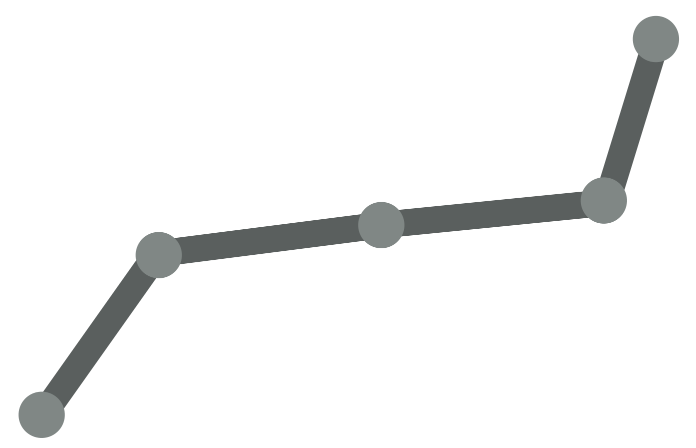
- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad! or Needs many points!

SHAPE REPRESENTATIONS

Point cloud

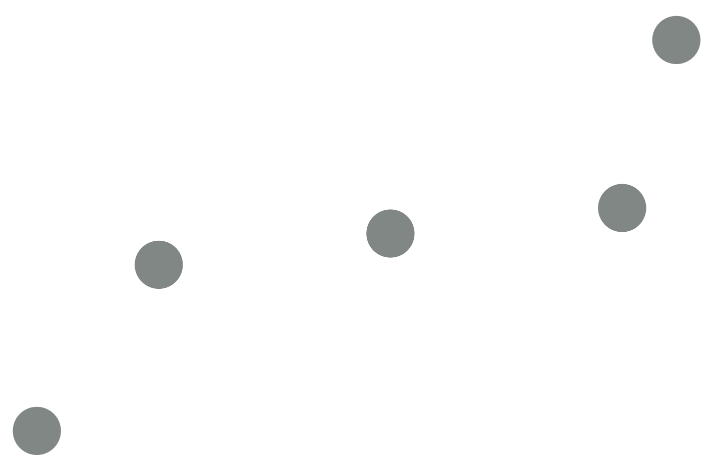


Polyline

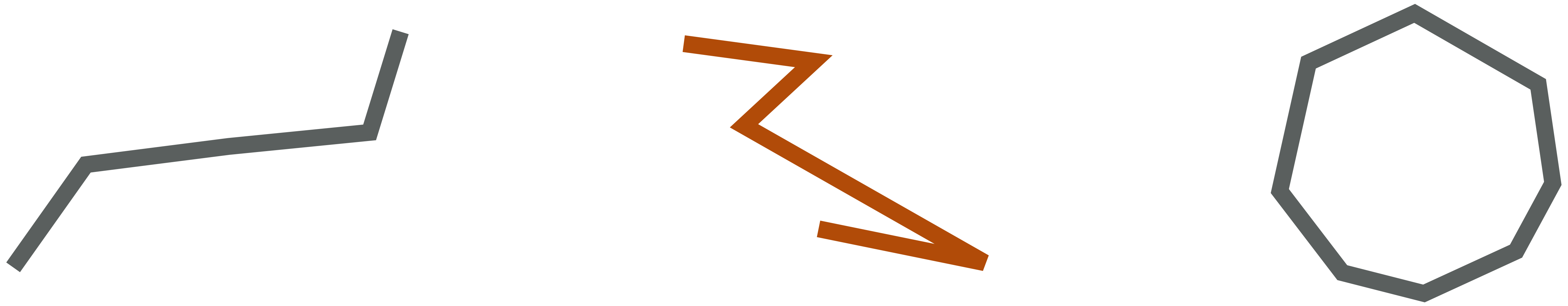


SHAPE REPRESENTATIONS

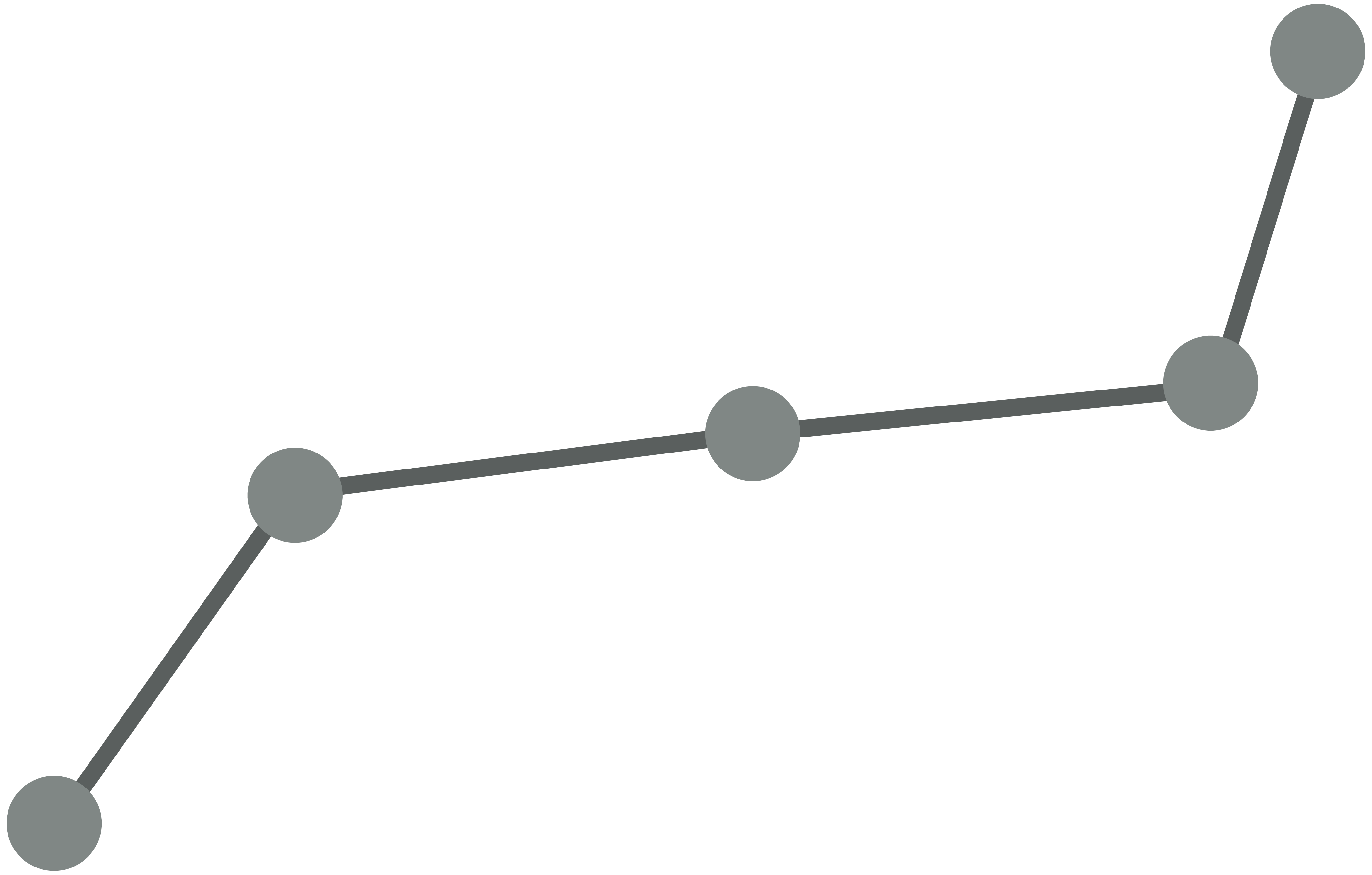
Point cloud



Points + Connectivity + Piecewise flat interpolation

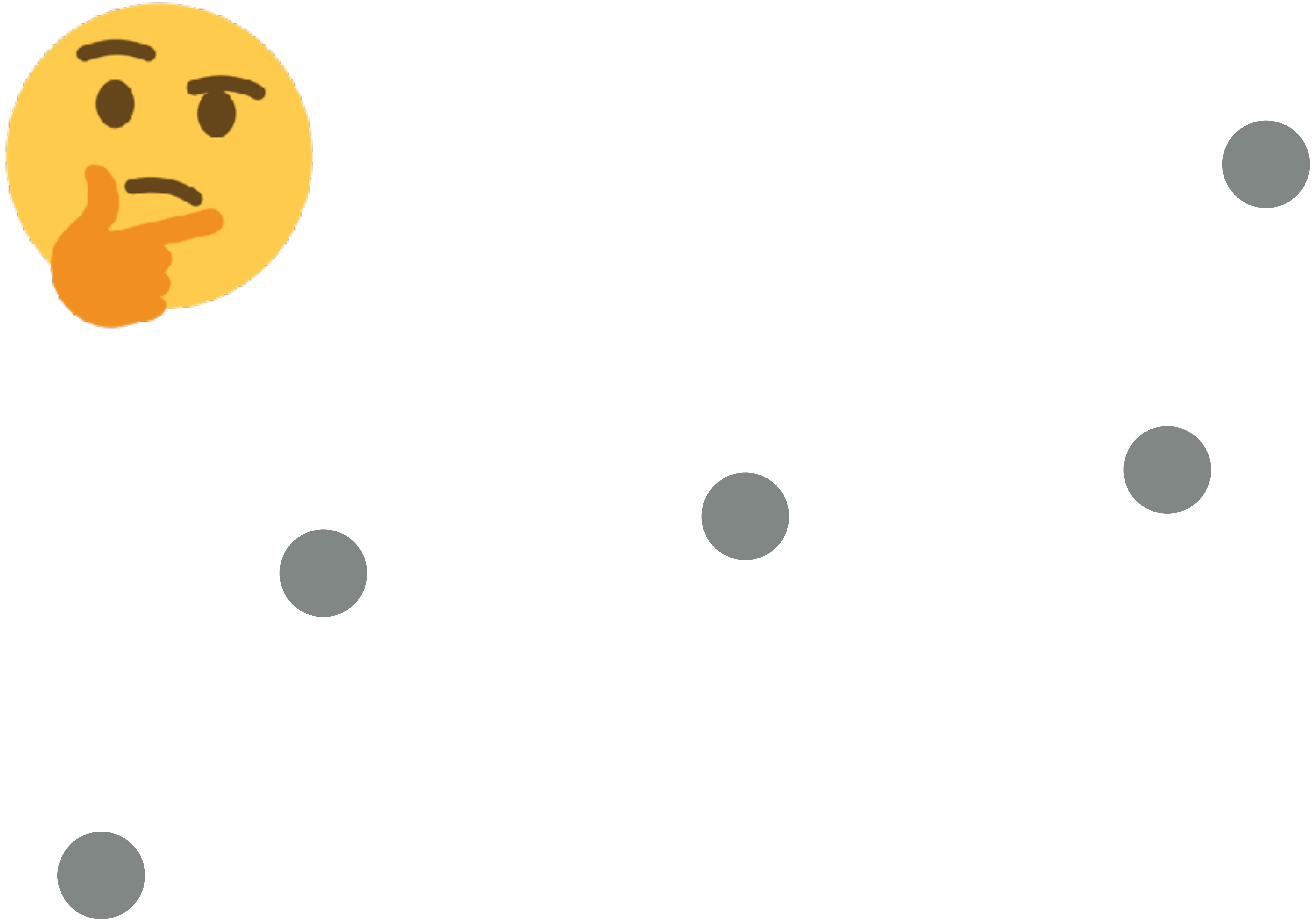


BEYOND POLYLINES



- ✓ Easy to store and plot
- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad! or
  Needs many points!

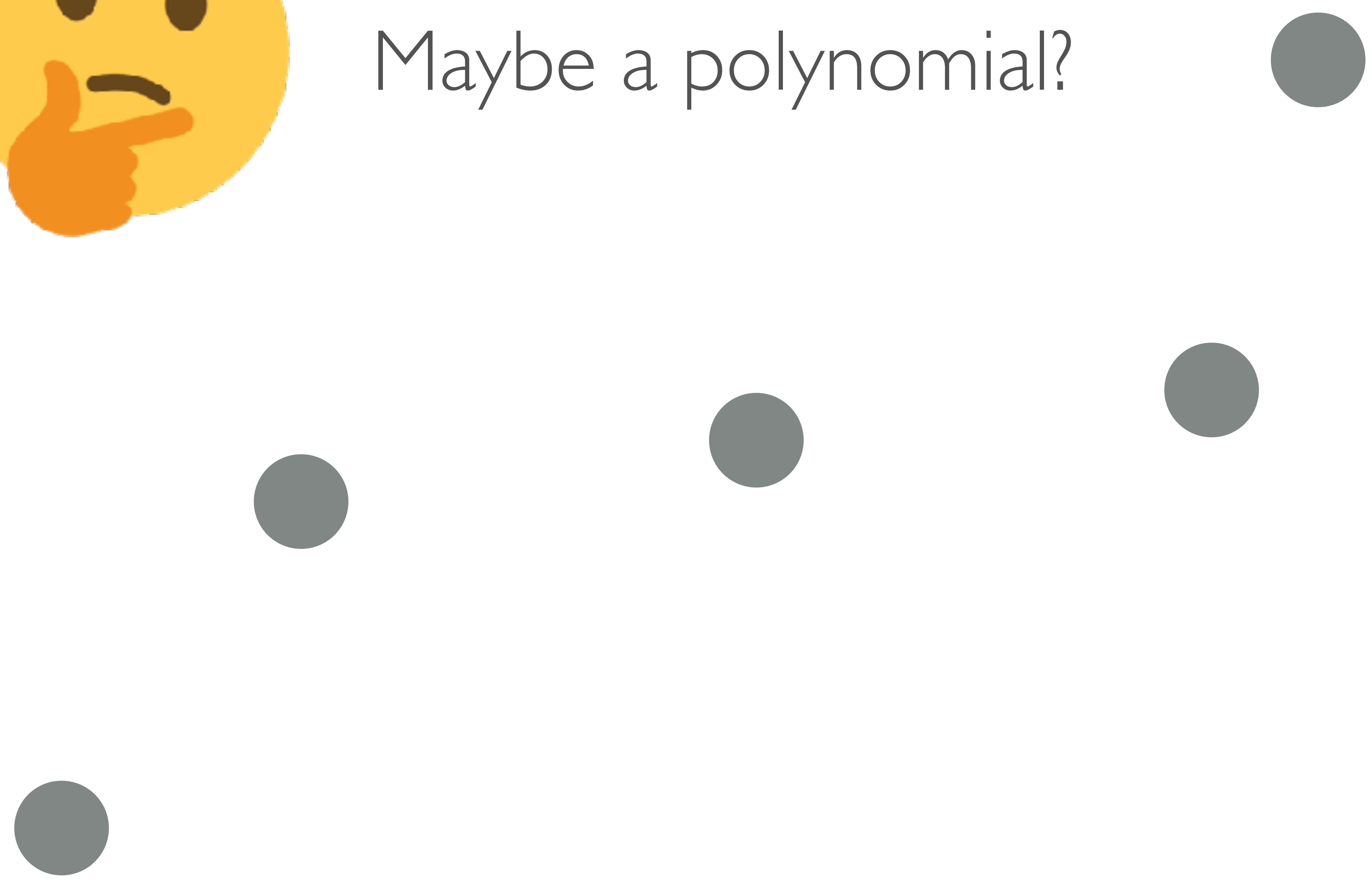
BEYOND POLYLINES



- ✗ Differential quantities?
- ✗ Looks bad! or
  Needs many points!

BEYOND POLYLINES

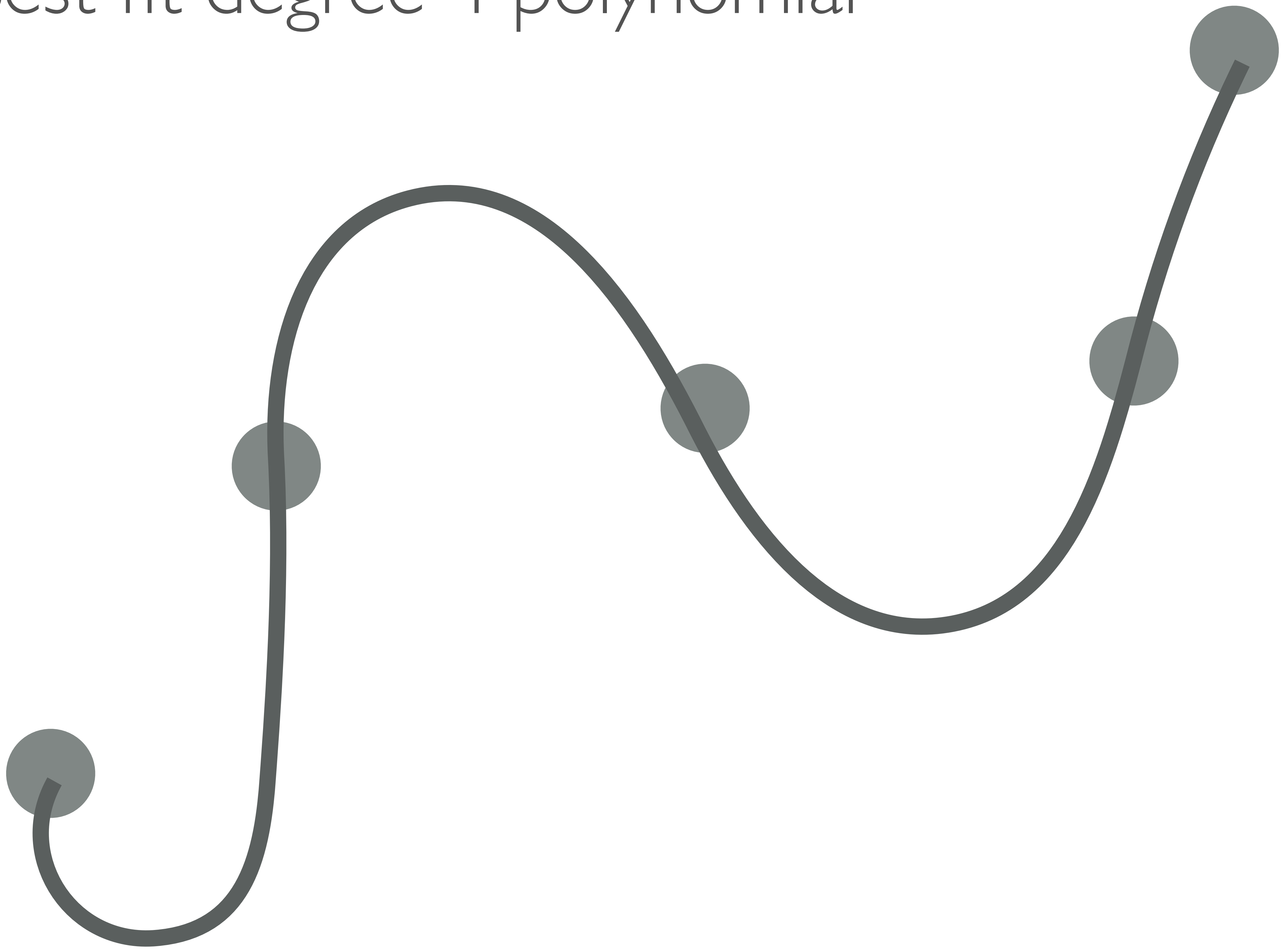
Maybe a polynomial?



- Differential quantities?
- Looks bad! or Needs many points!

BEYOND POLYLINES

Best fit degree 4 polynomial

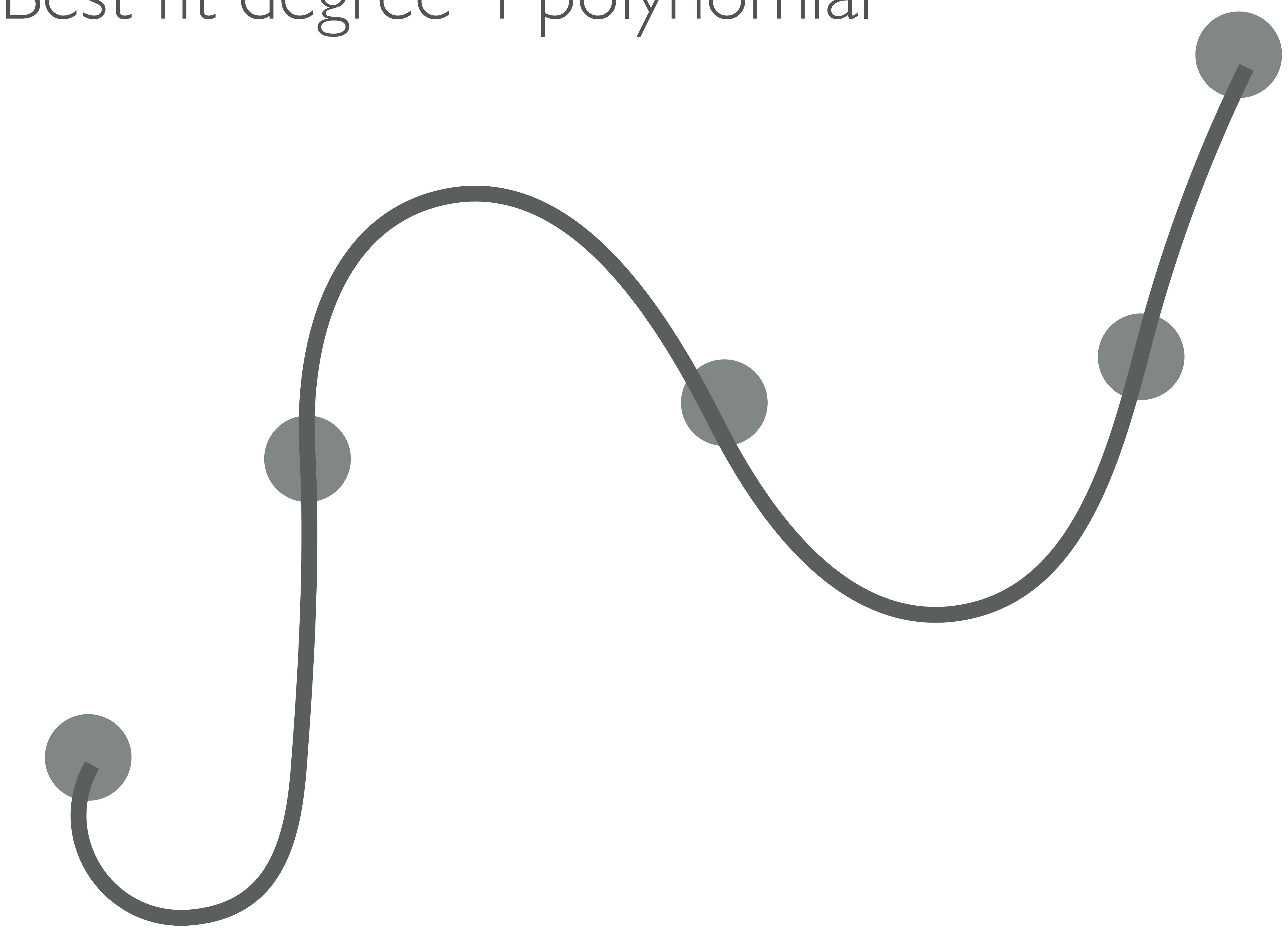


✗ Differential quantities

✗ Looks bad! or
Needs many points!

BEYOND POLYLINES

Best fit degree 4 polynomial

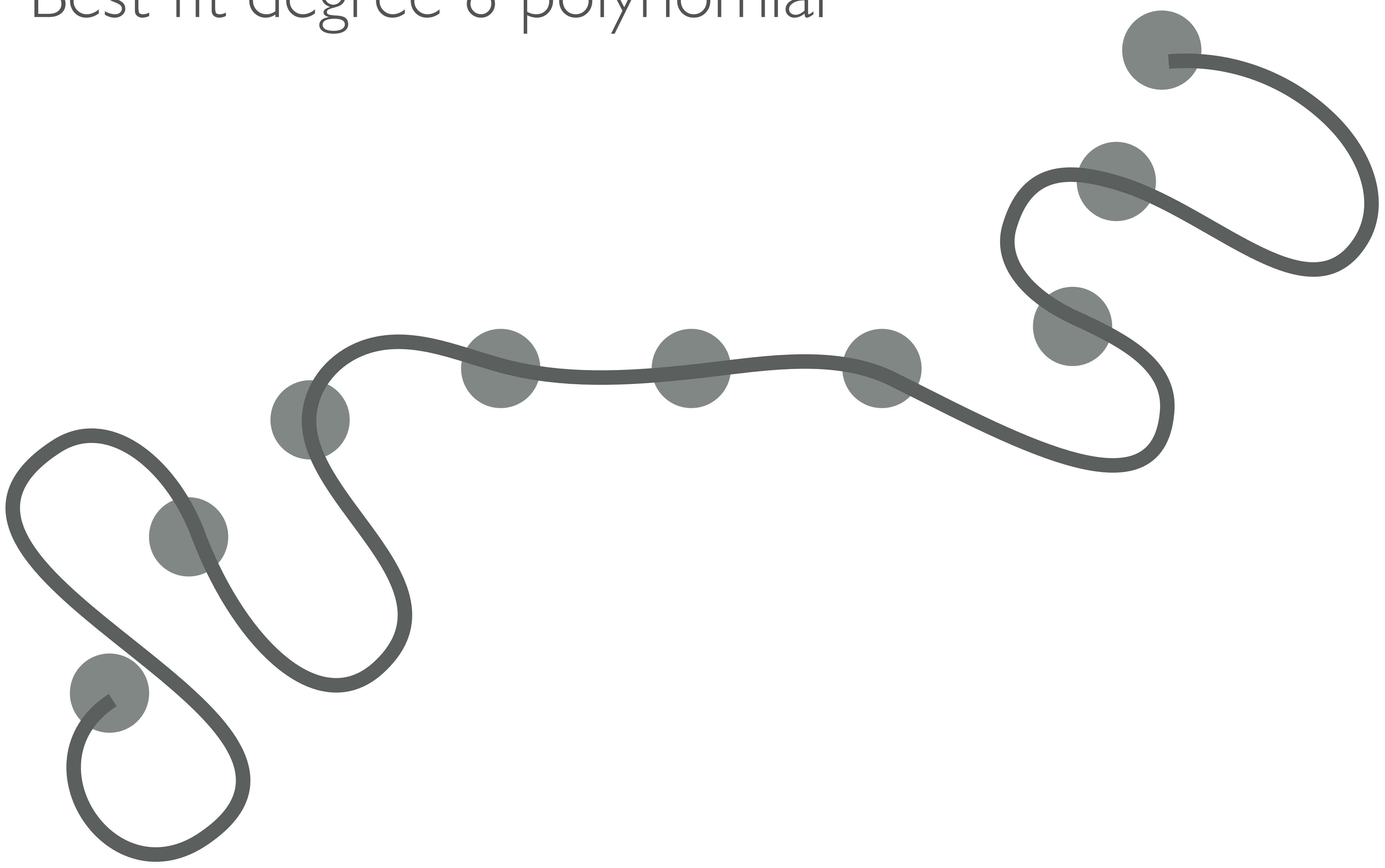


✓ Differential quantities

✗ Looks bad! or
Needs many points!

BEYOND POLYLINES

Best fit degree 8 polynomial



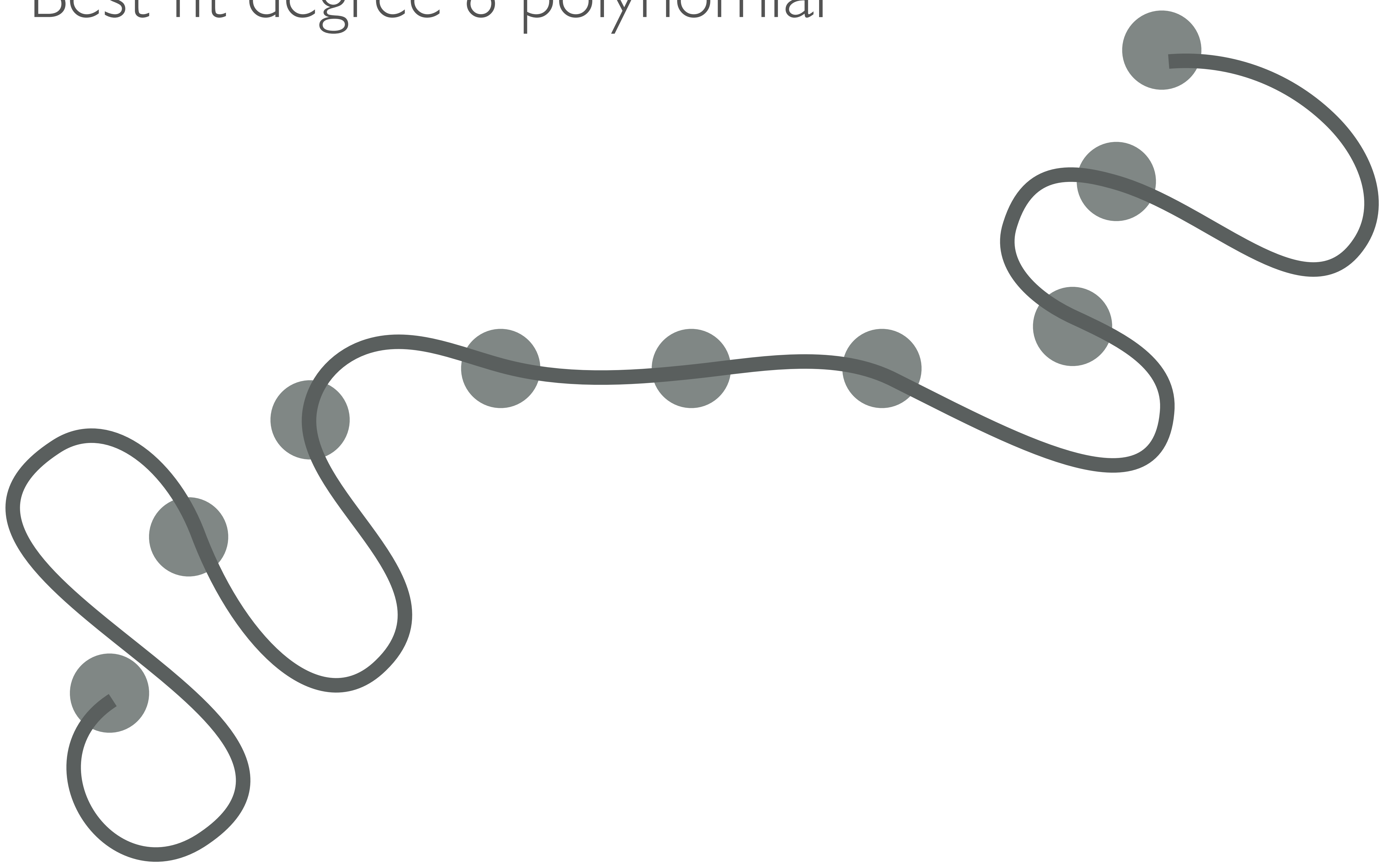
Differential quantities

Looks bad! or

Needs many points!

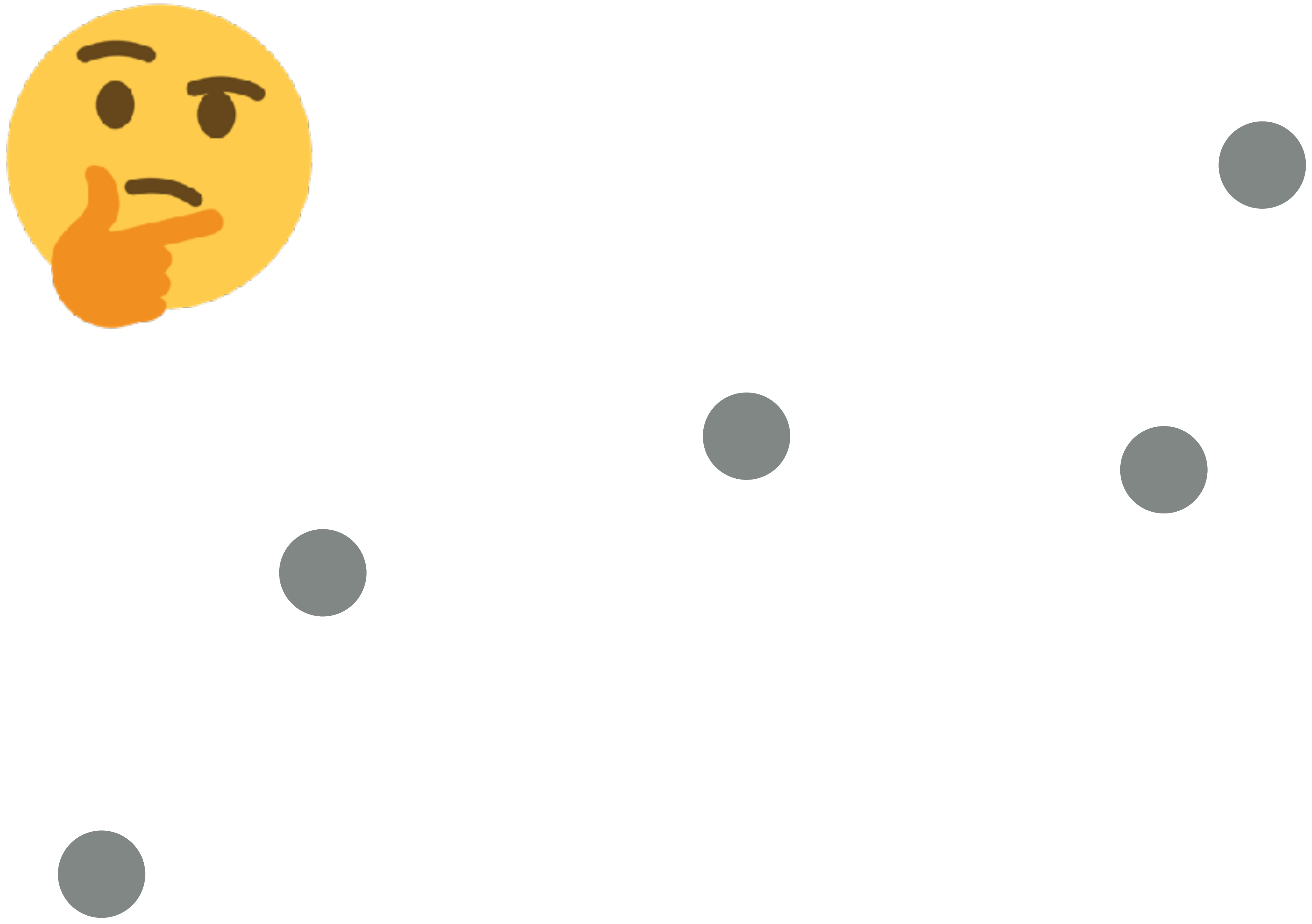
BEYOND POLYLINES

Best fit degree 8 polynomial



- ✓ Differential quantities
- ✗ Runge's phenomenon
- ✗ Degree grows indefinitely

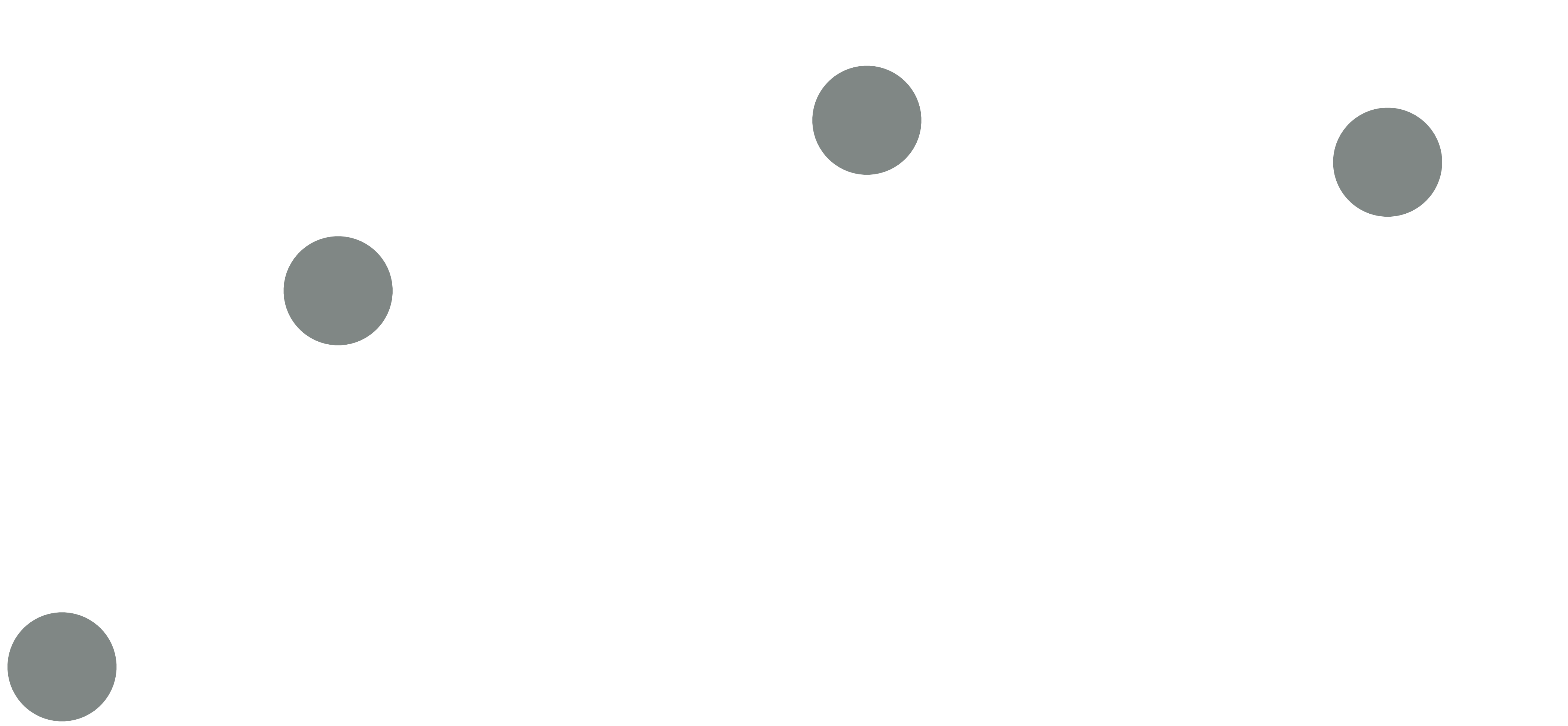
BEYOND POLYLINES



- ✗ Differential quantities?
- ✗ Looks bad! or Needs many points!

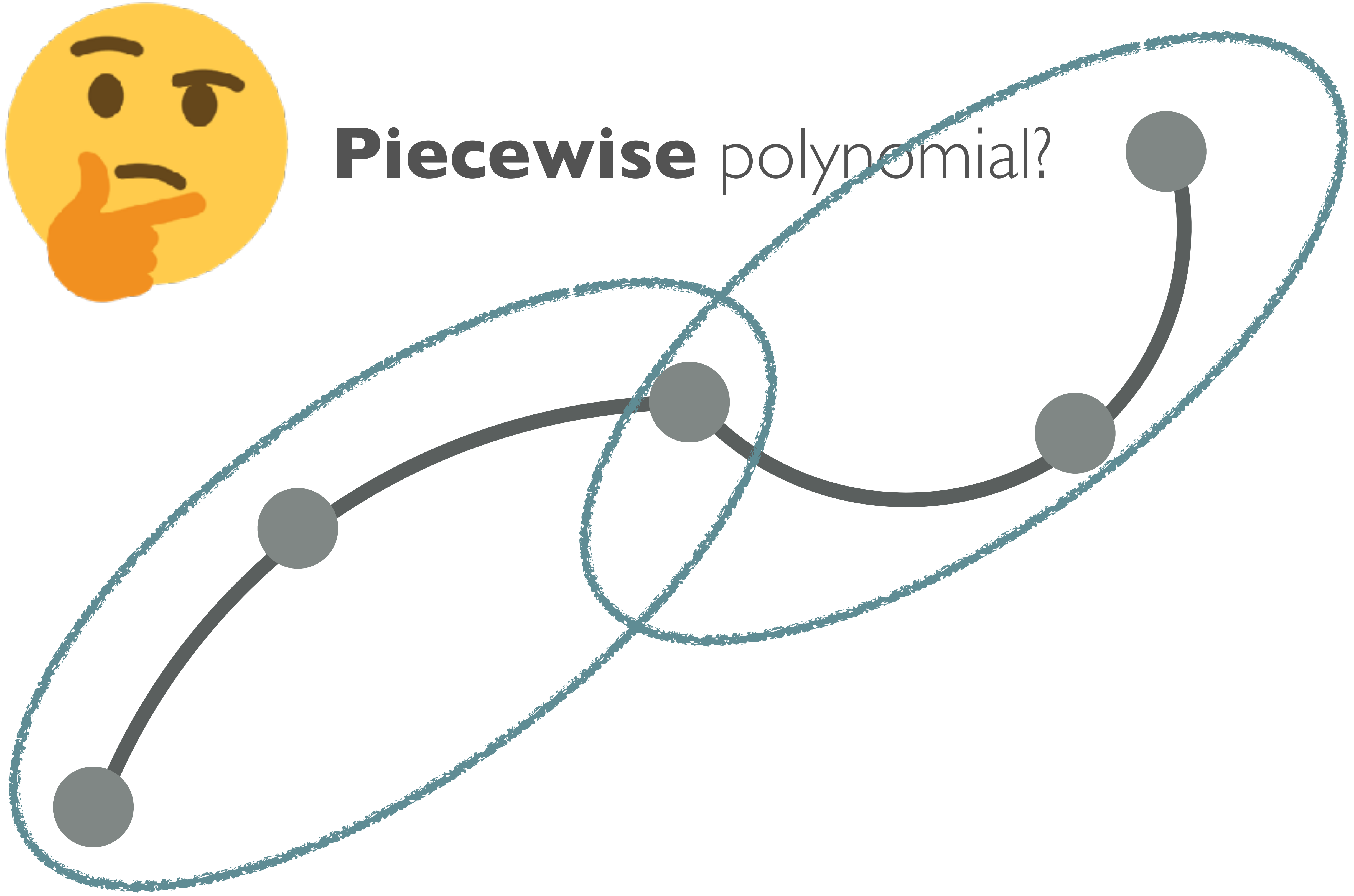
BEYOND POLYLINES

Piecewise polynomial?



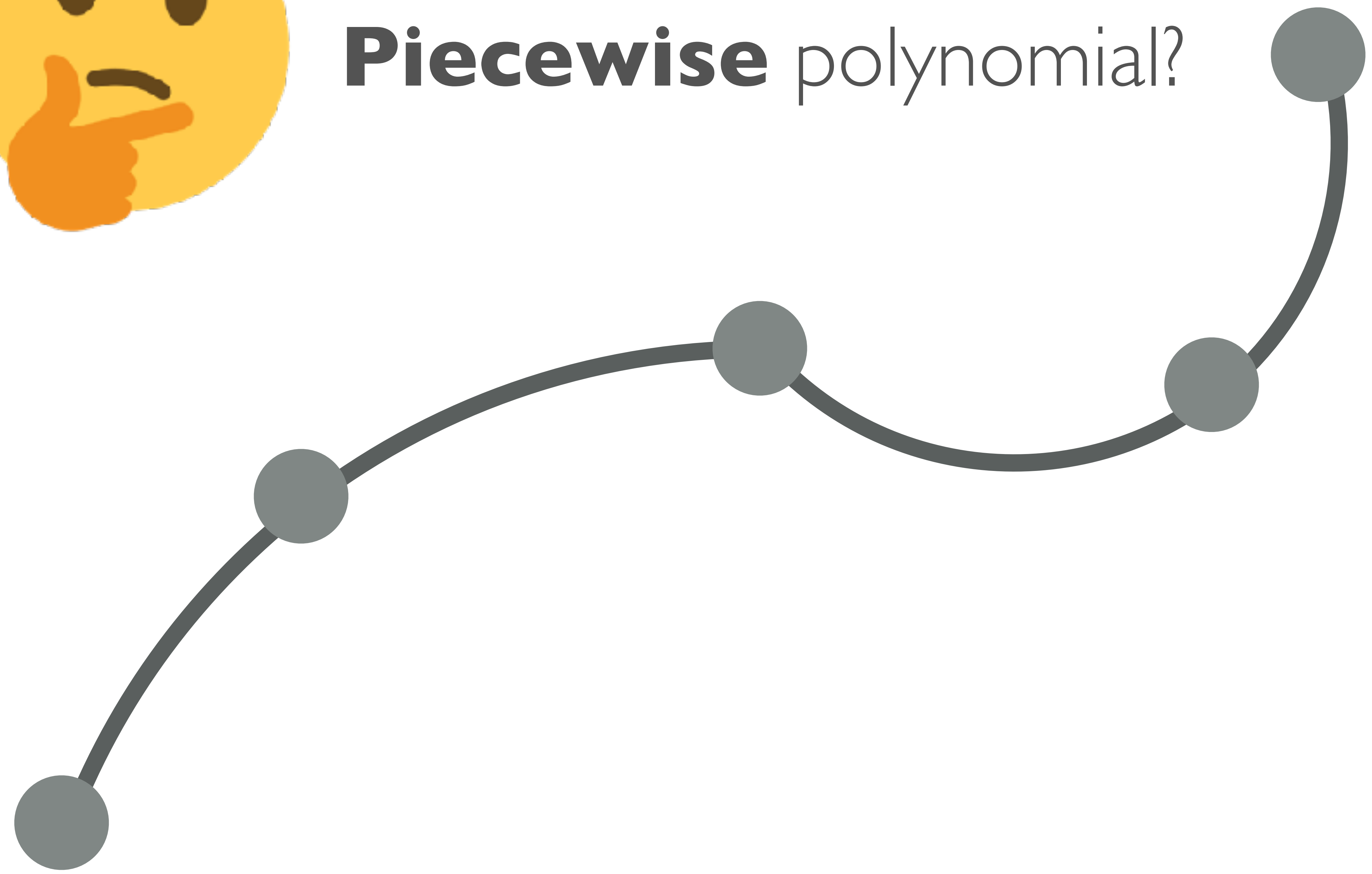
BEYOND POLYLINES

Piecewise polynomial?



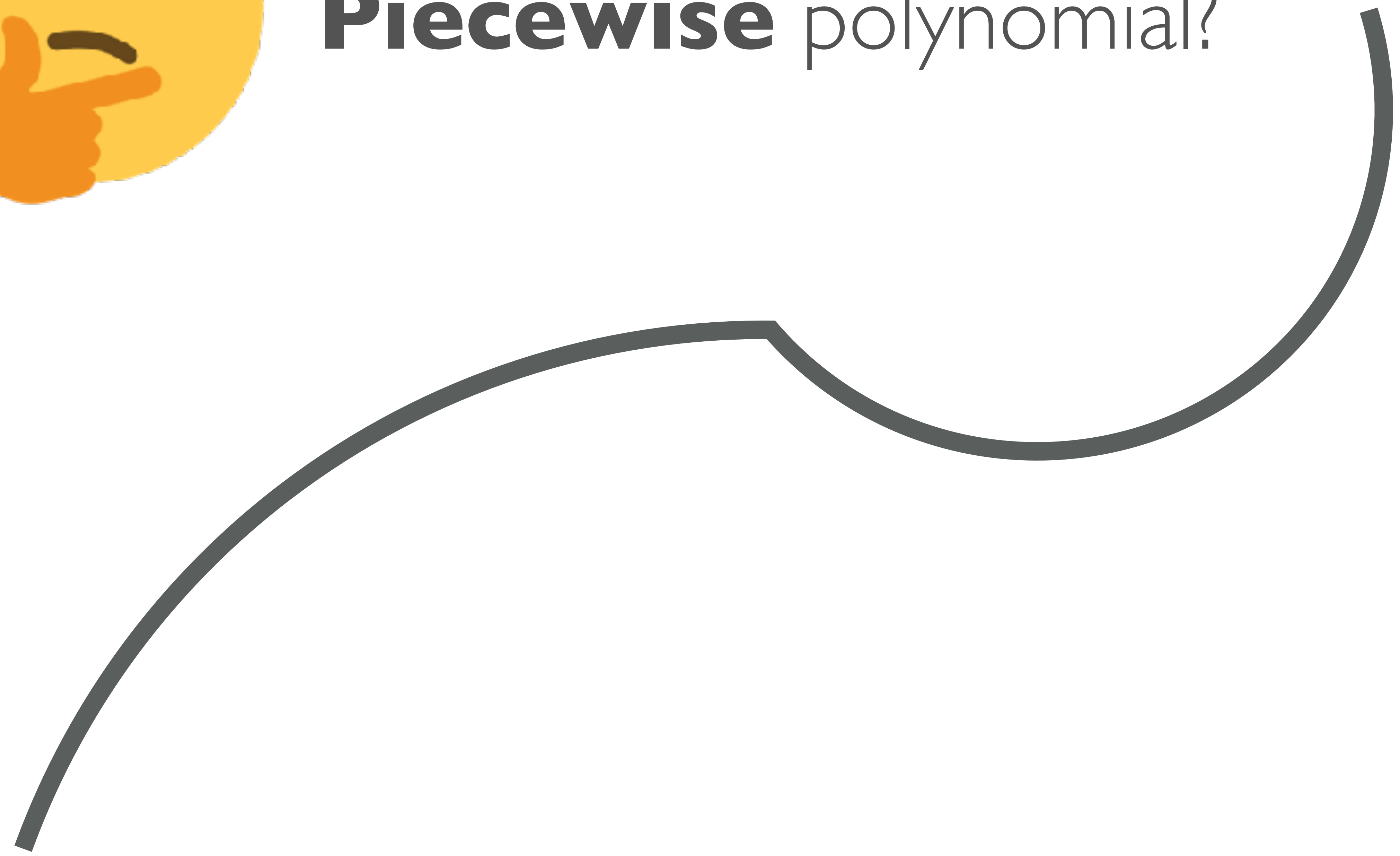
BEYOND POLYLINES

Piecewise polynomial?

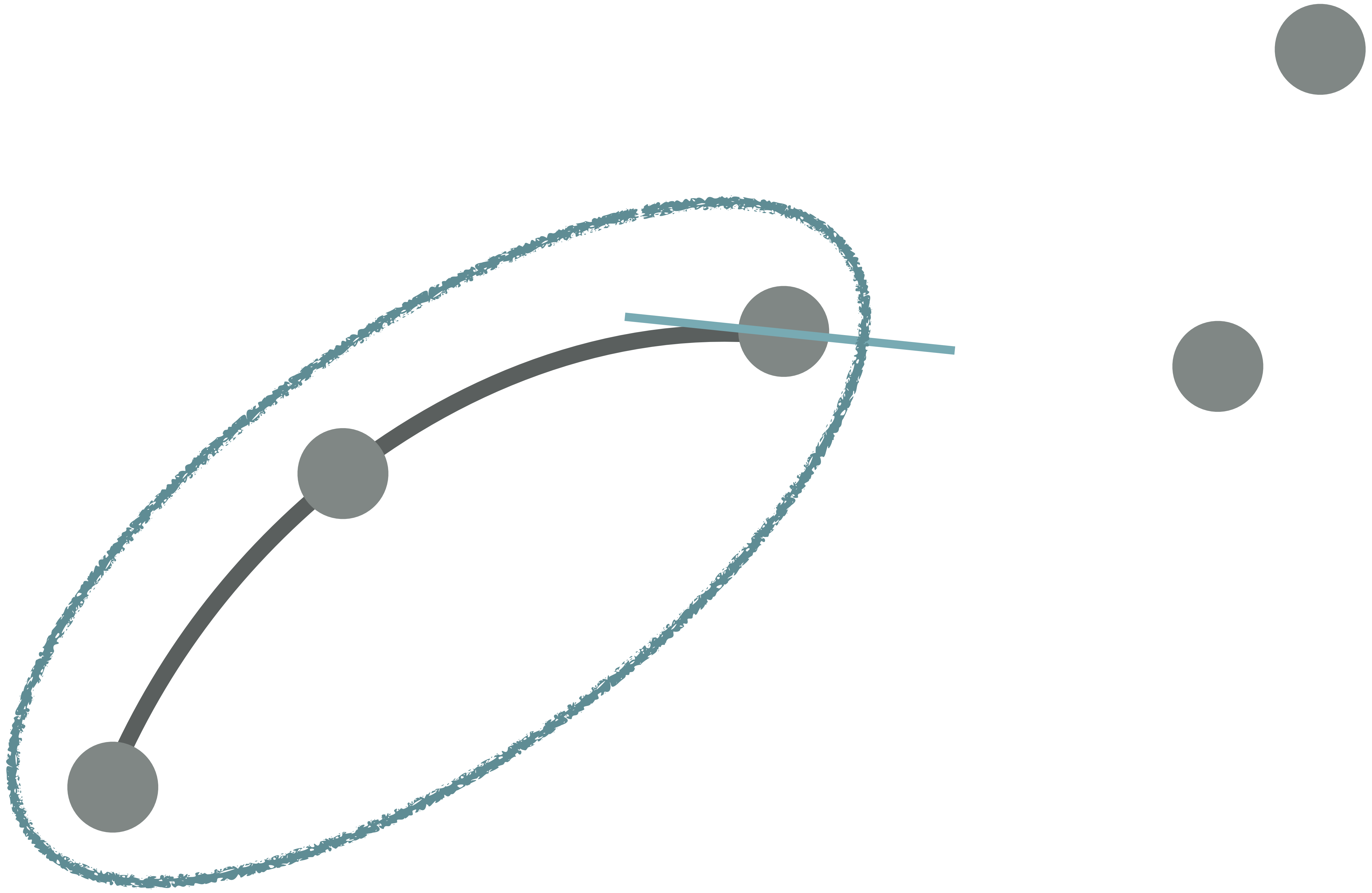


BEYOND POLYLINES

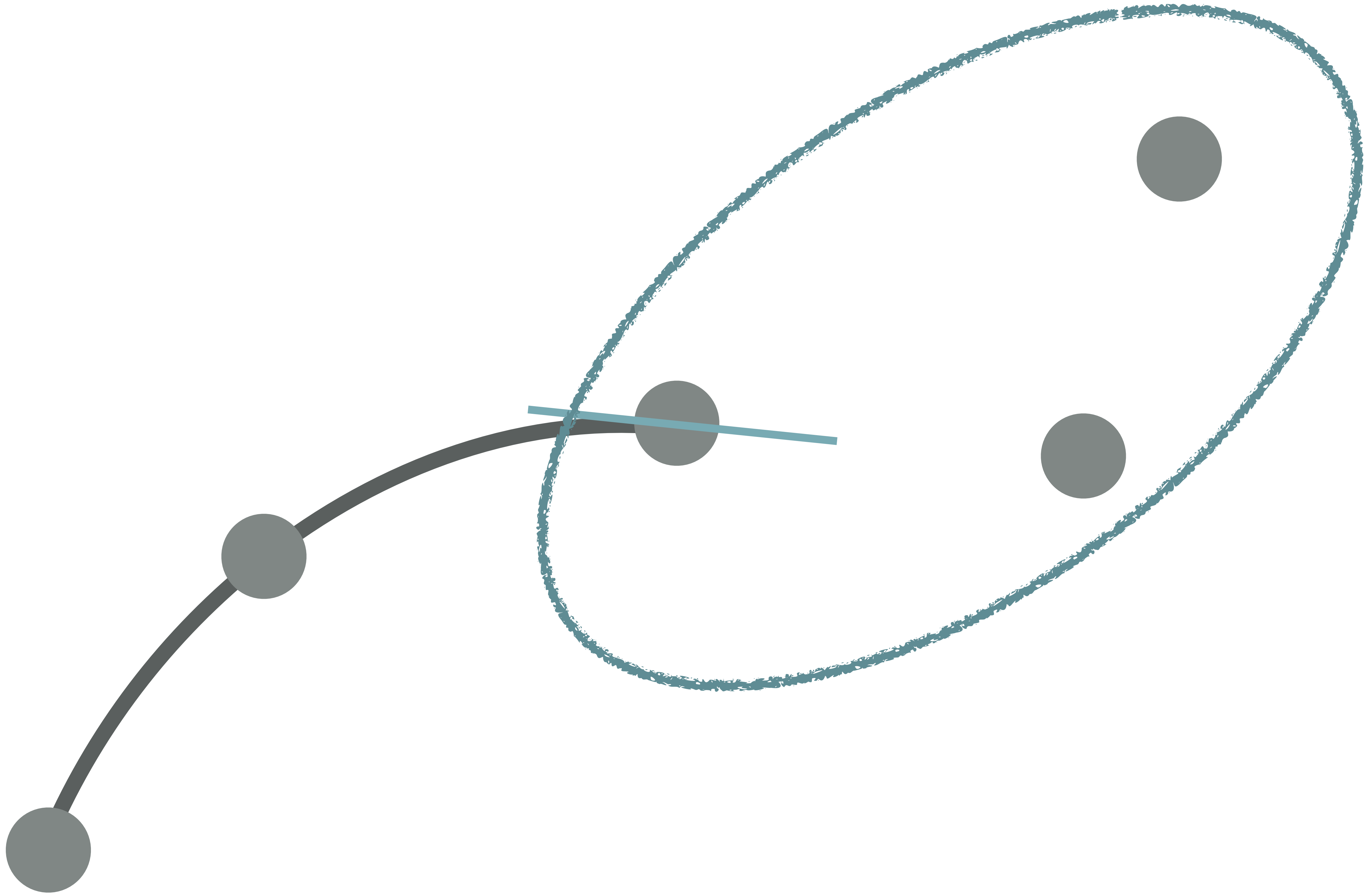
Piecewise polynomial?



BEYOND POLYLINES

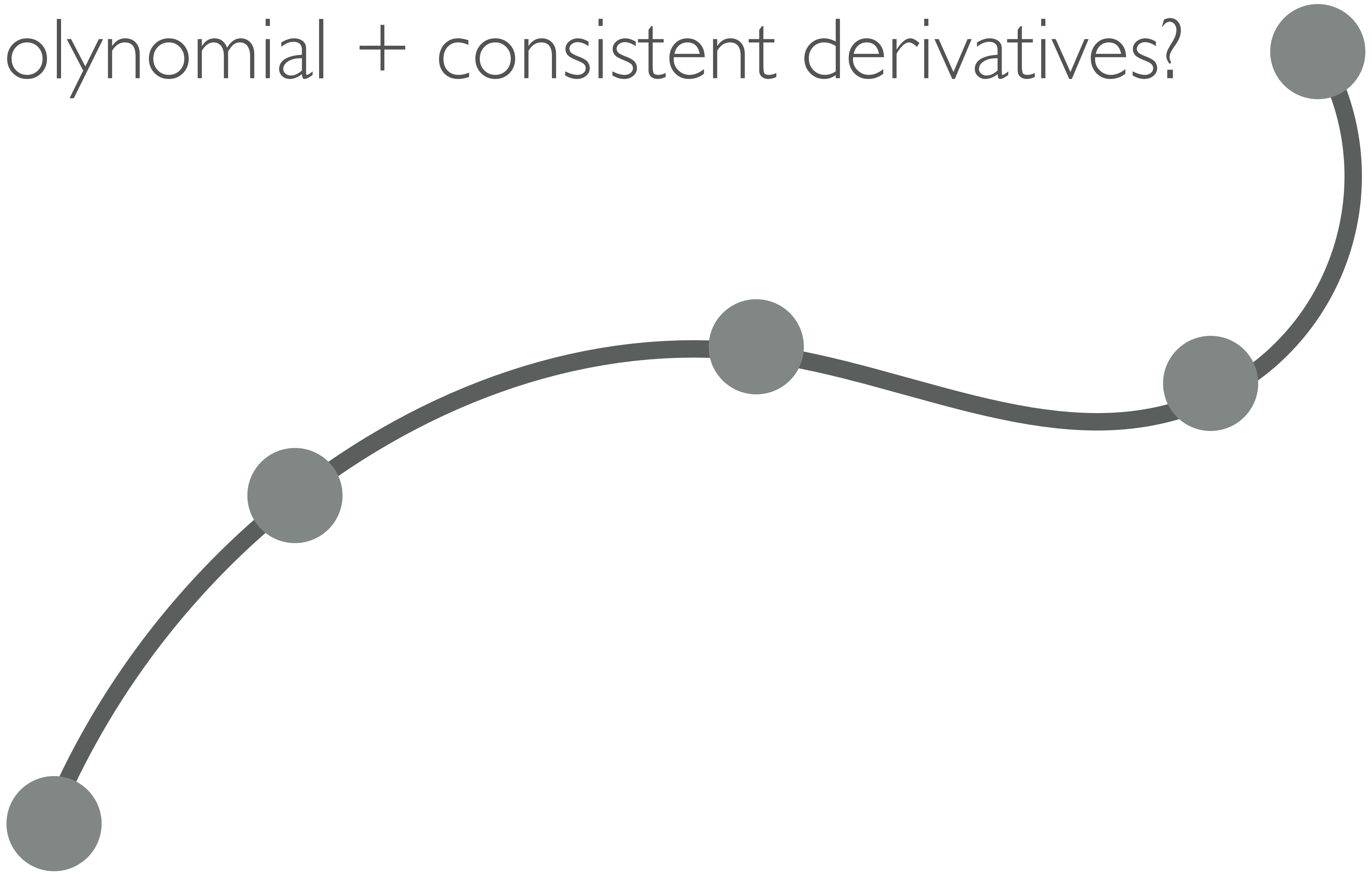


BEYOND POLYLINES



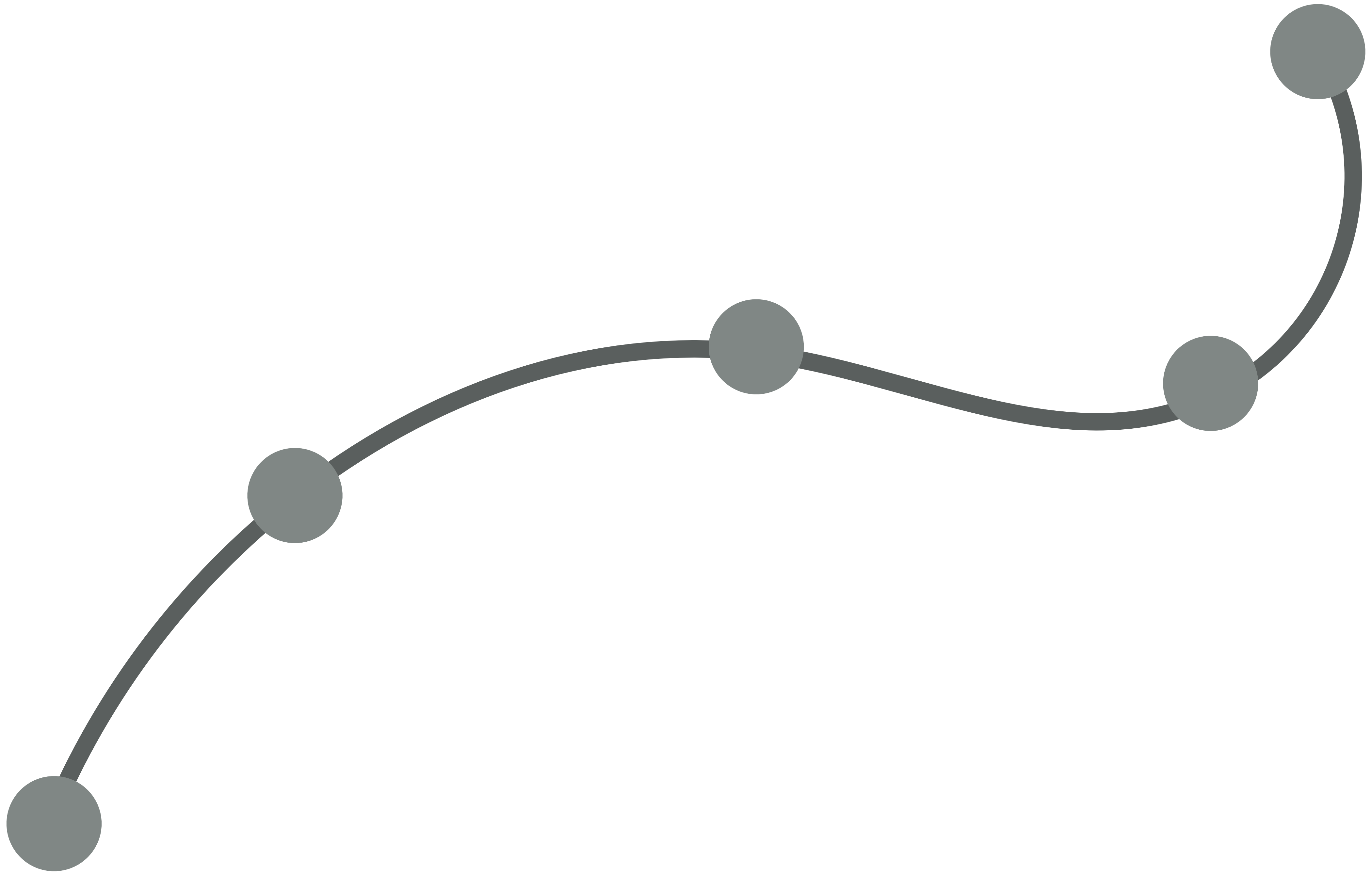
BEYOND POLYLINES

Piecewise polynomial + consistent derivatives?



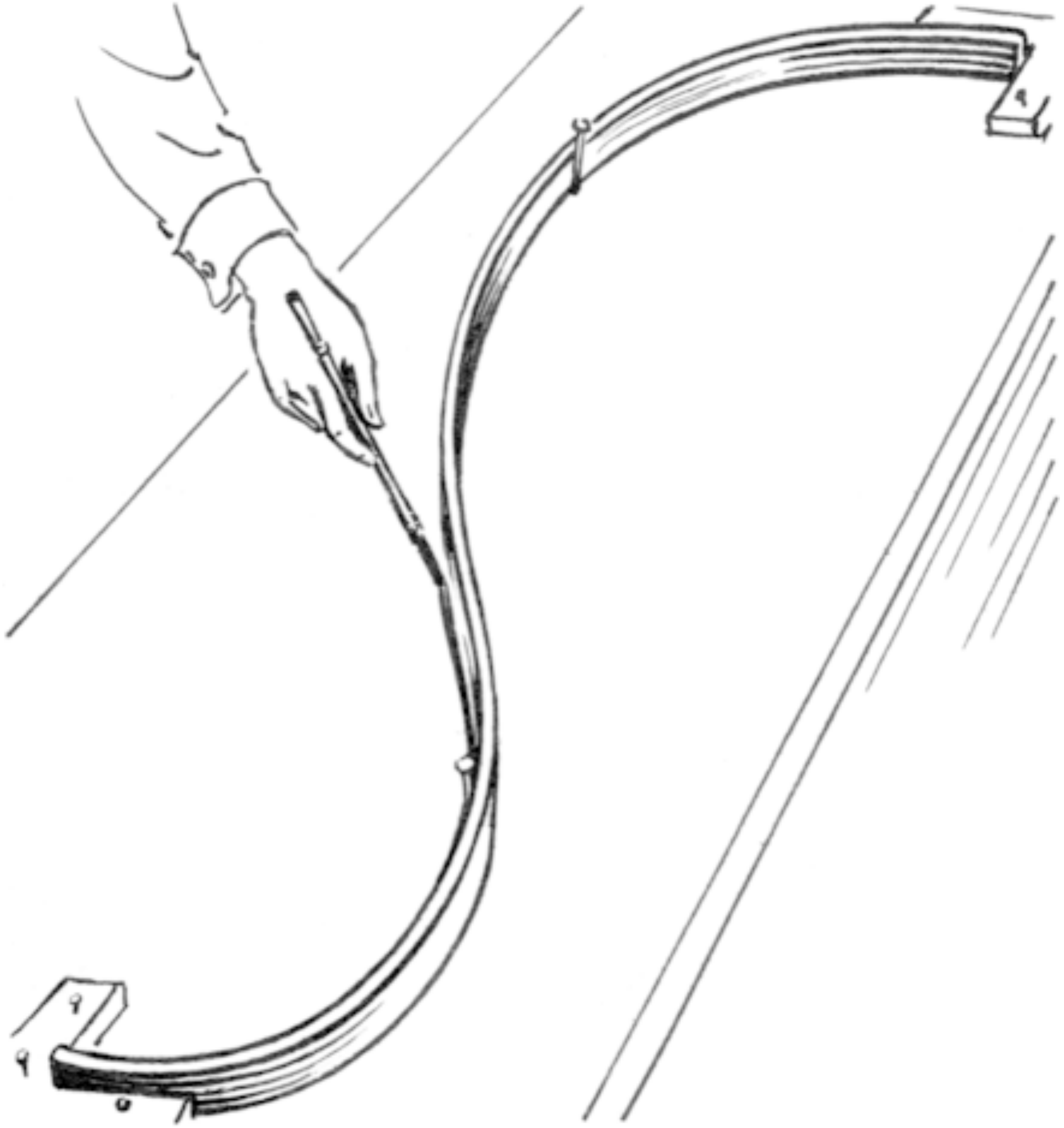
SPLINES

A spline!

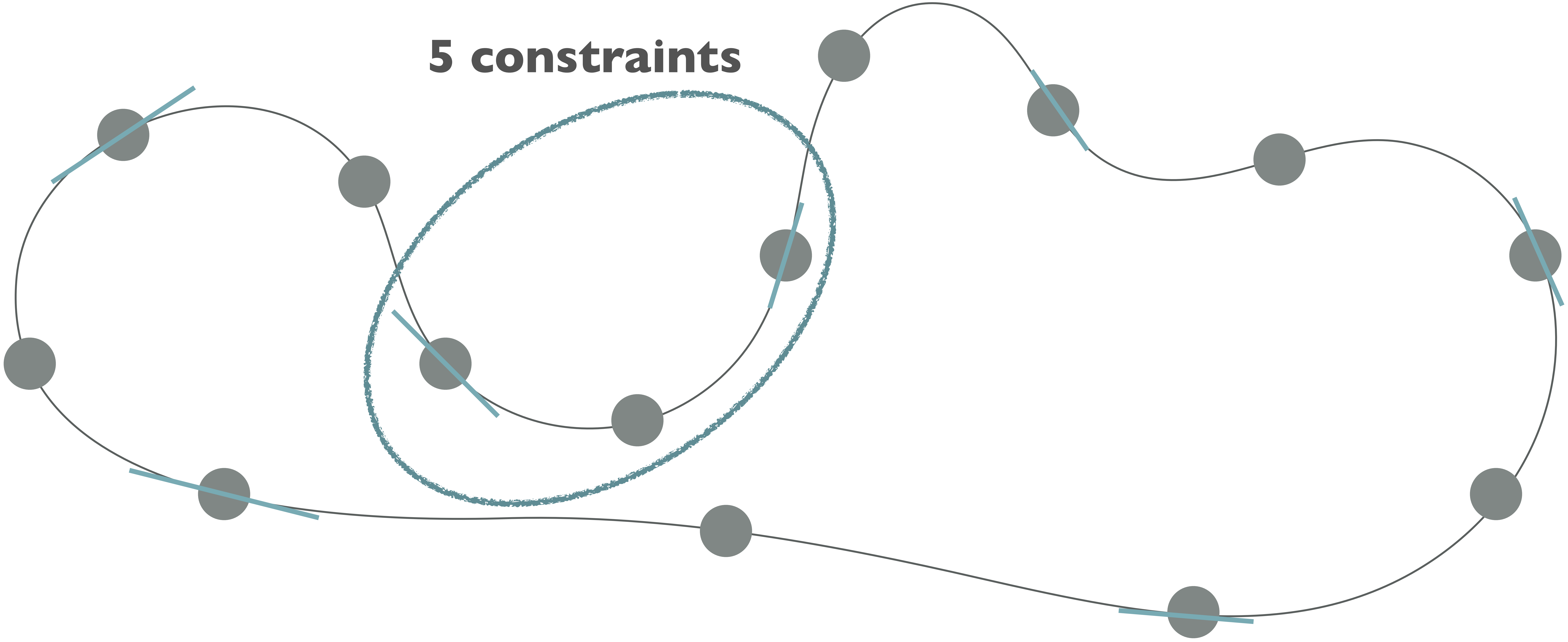


SPLINES

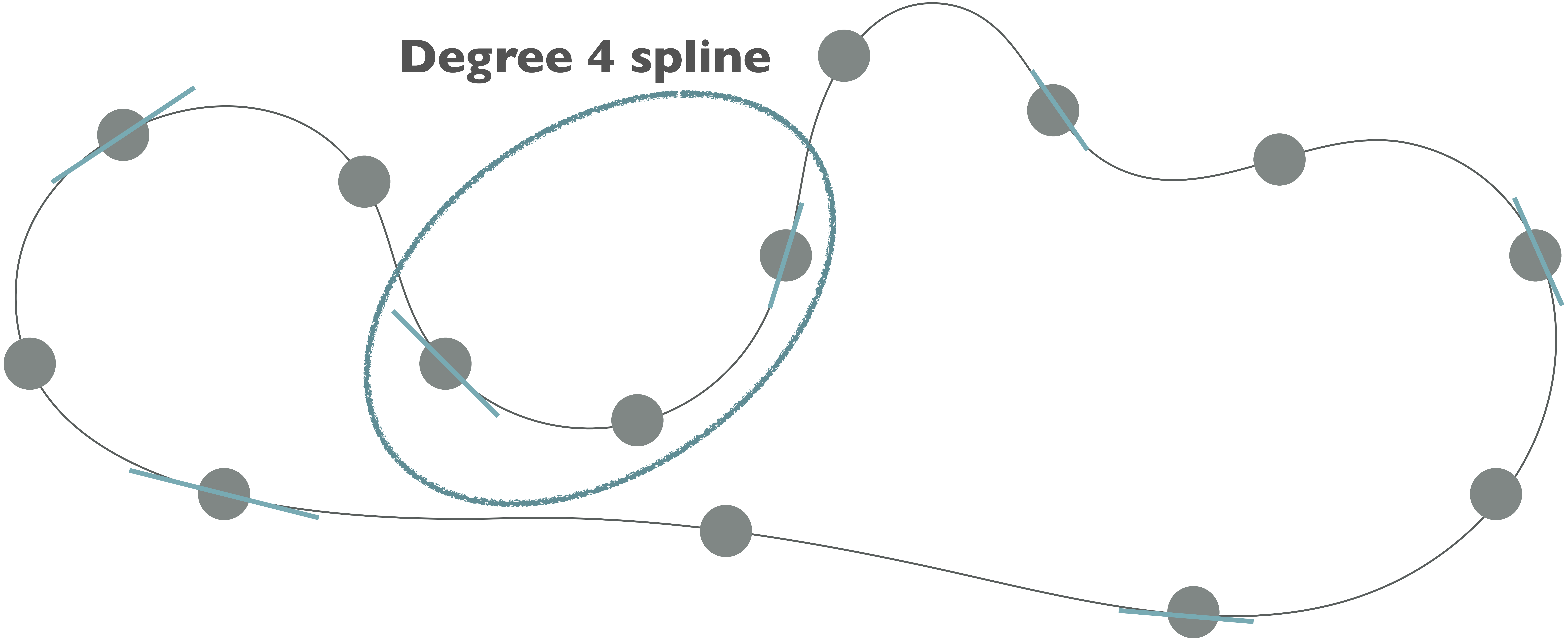
A spline!



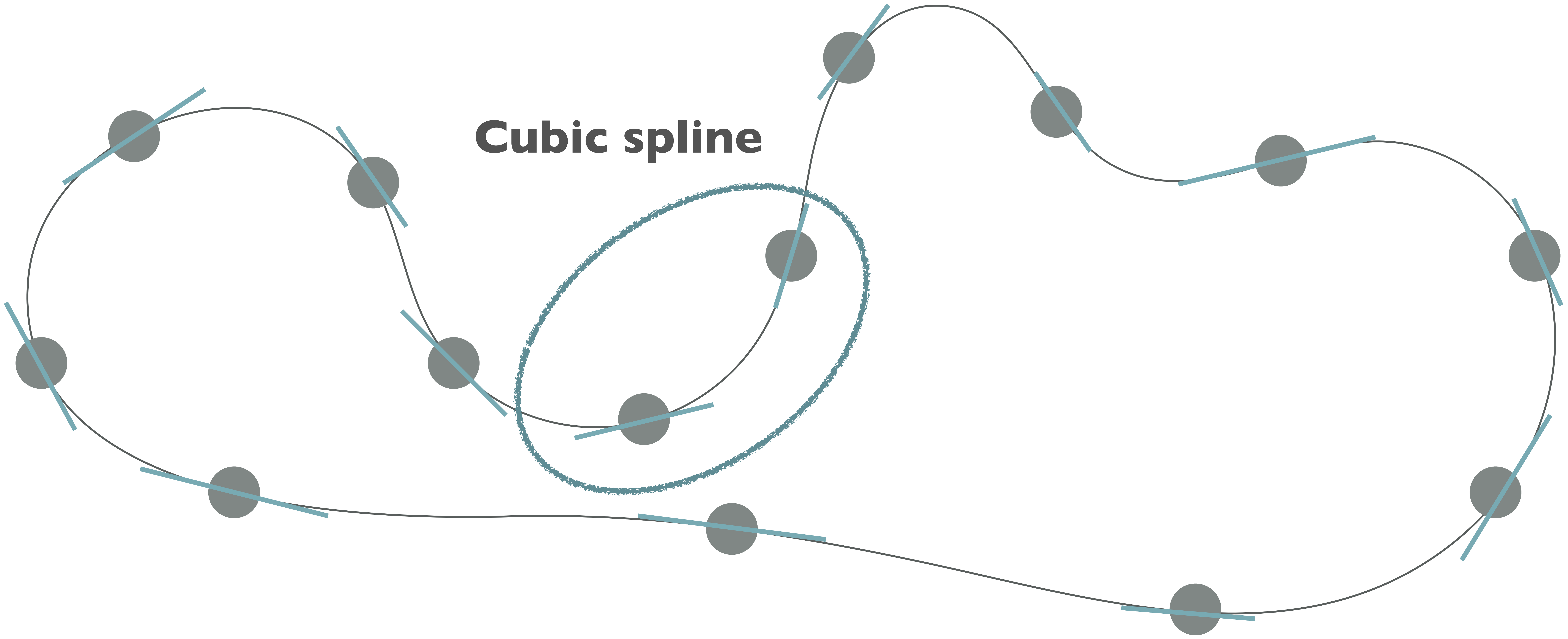
SPLINES



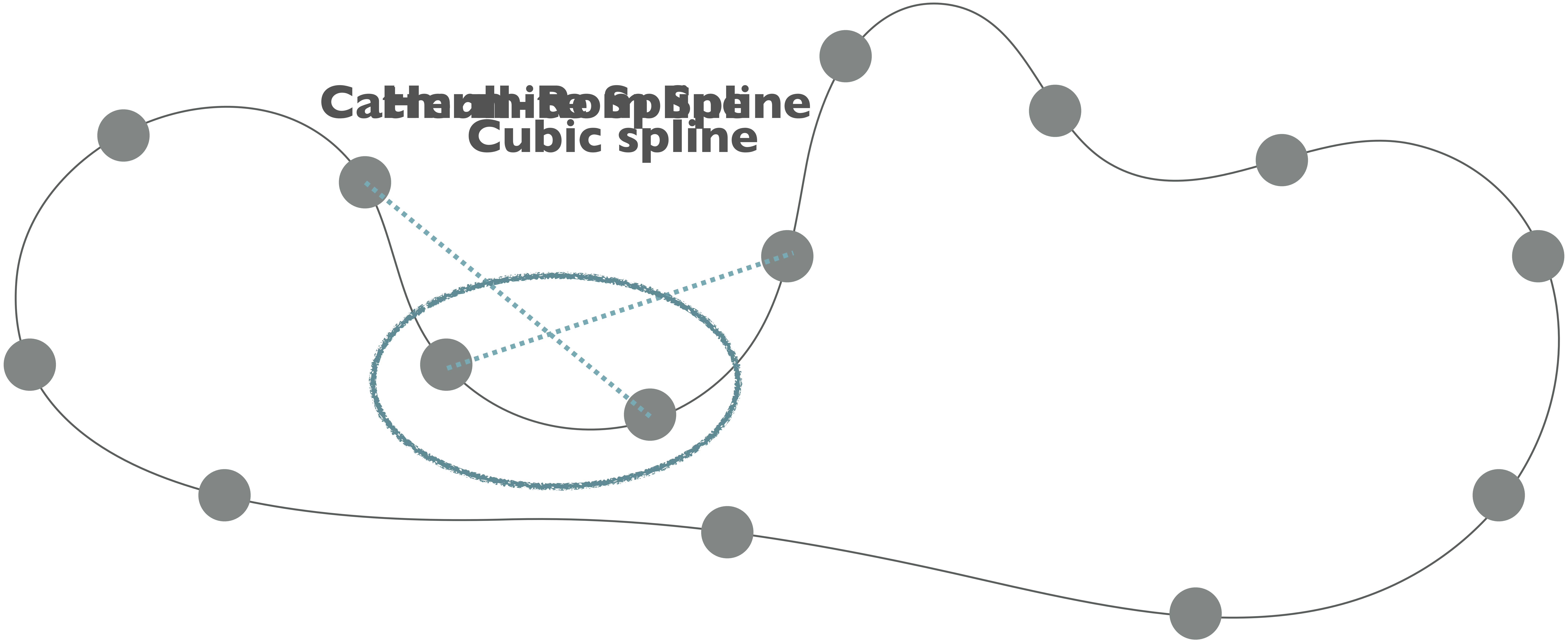
SPLINES



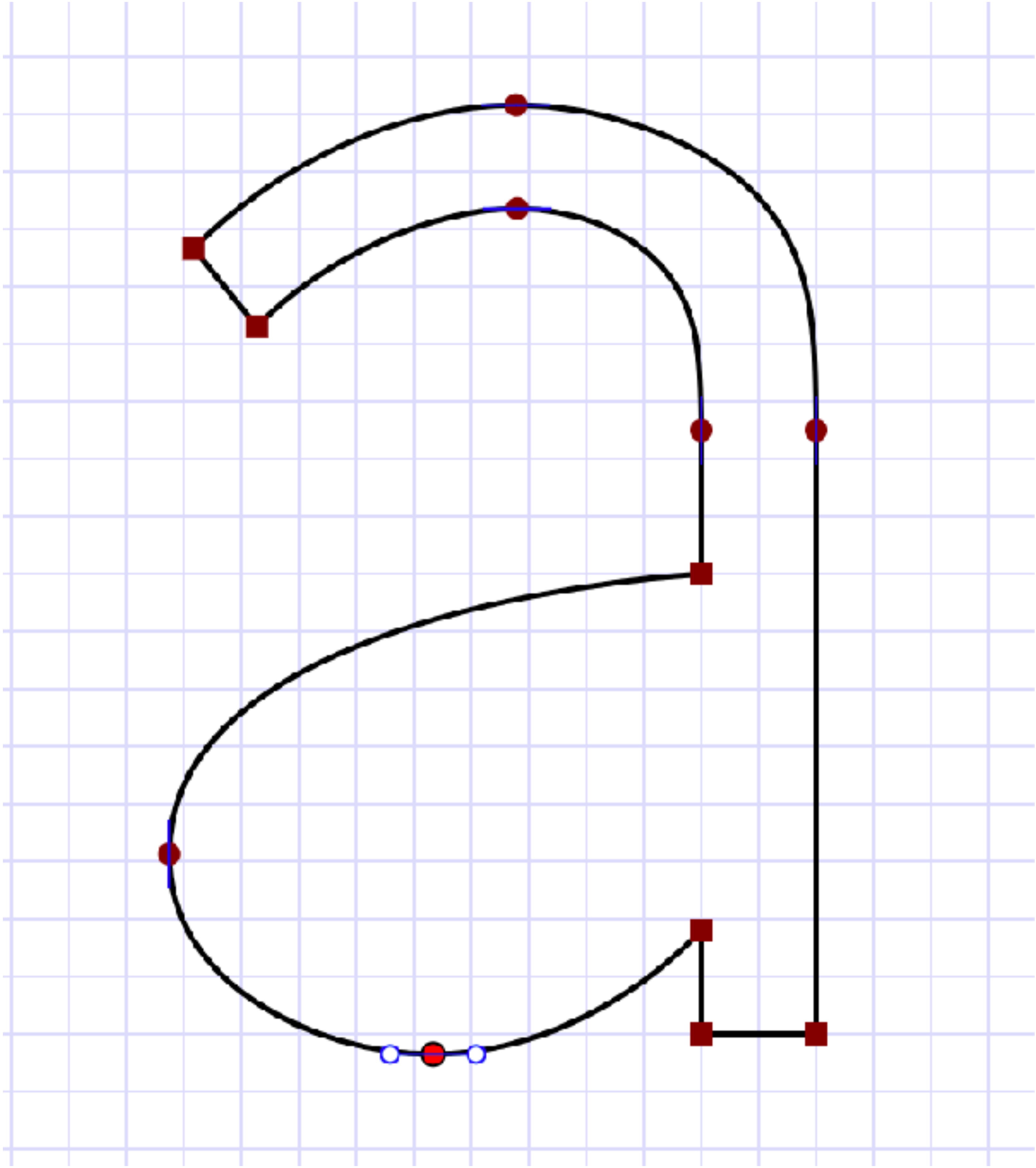
SPLINES



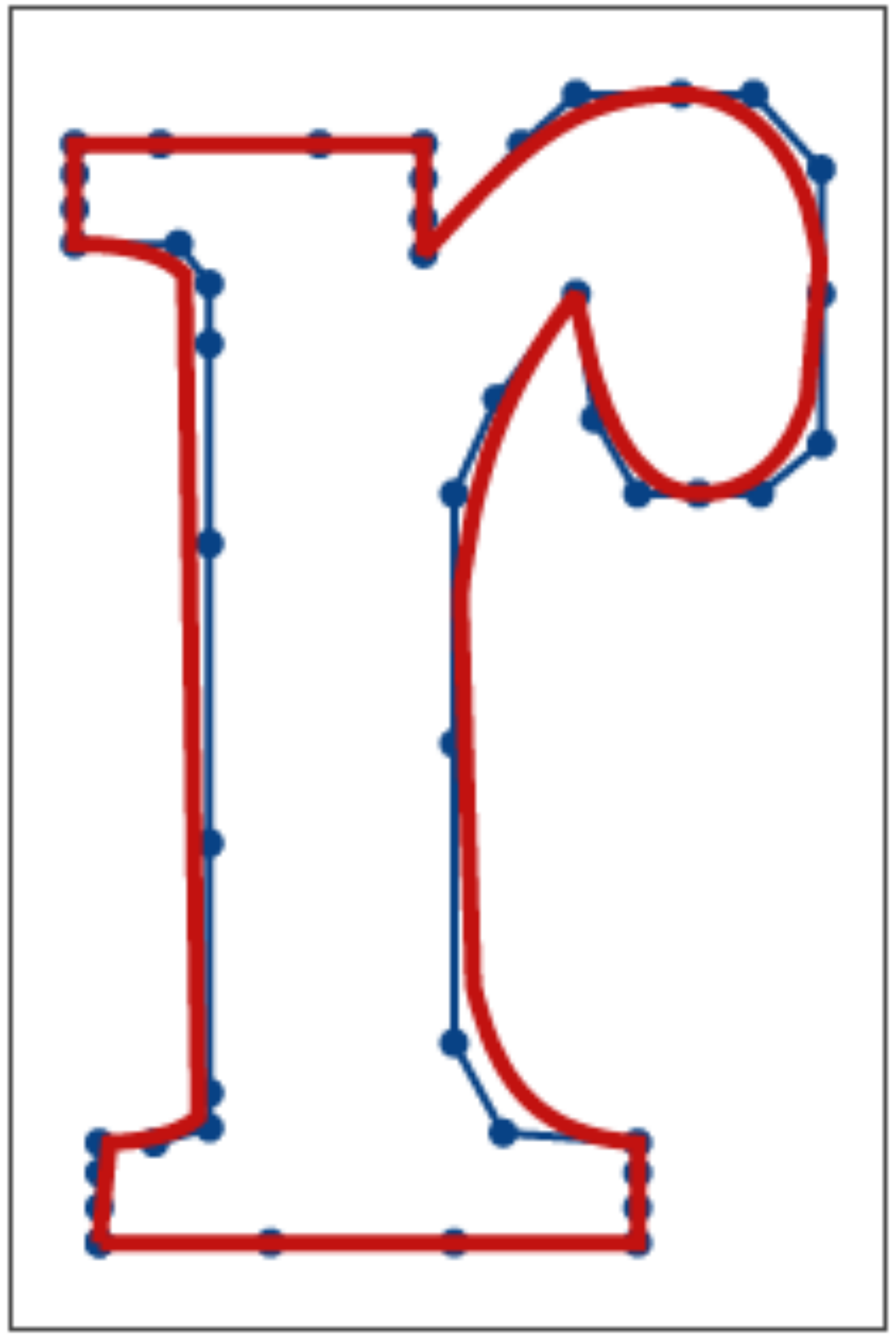
SPLINES



SPLINES

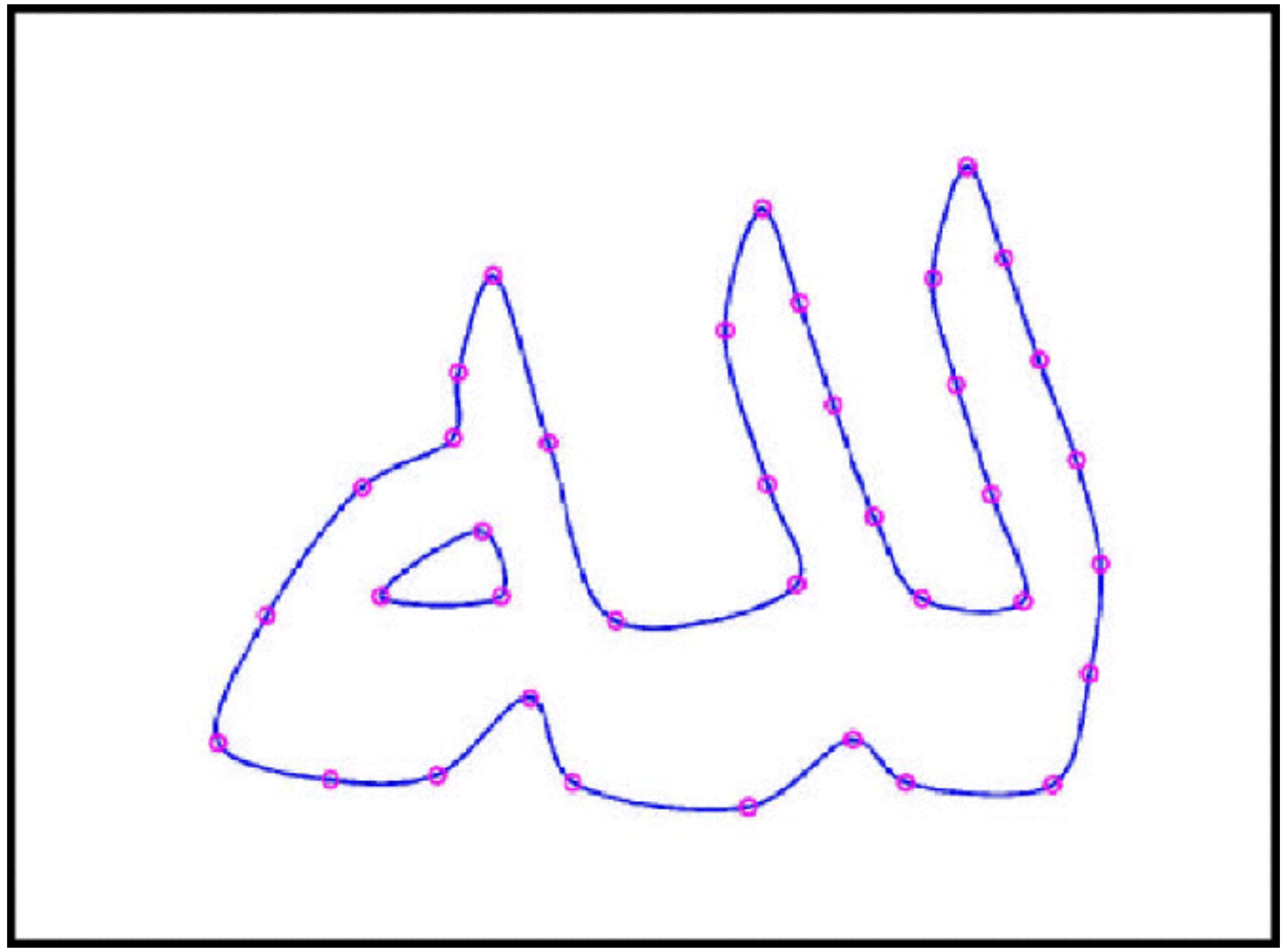
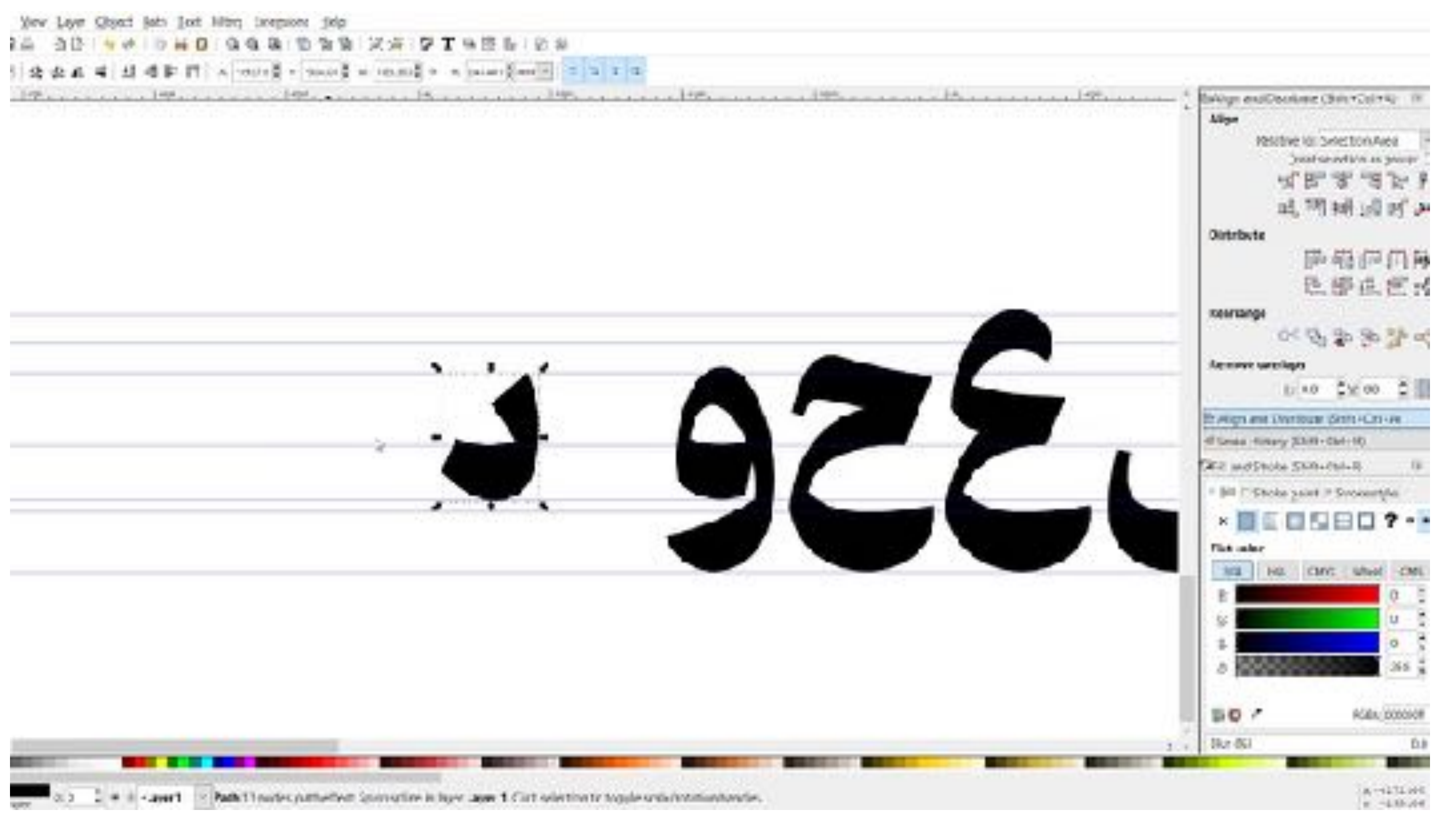


Font as a B-spline curve

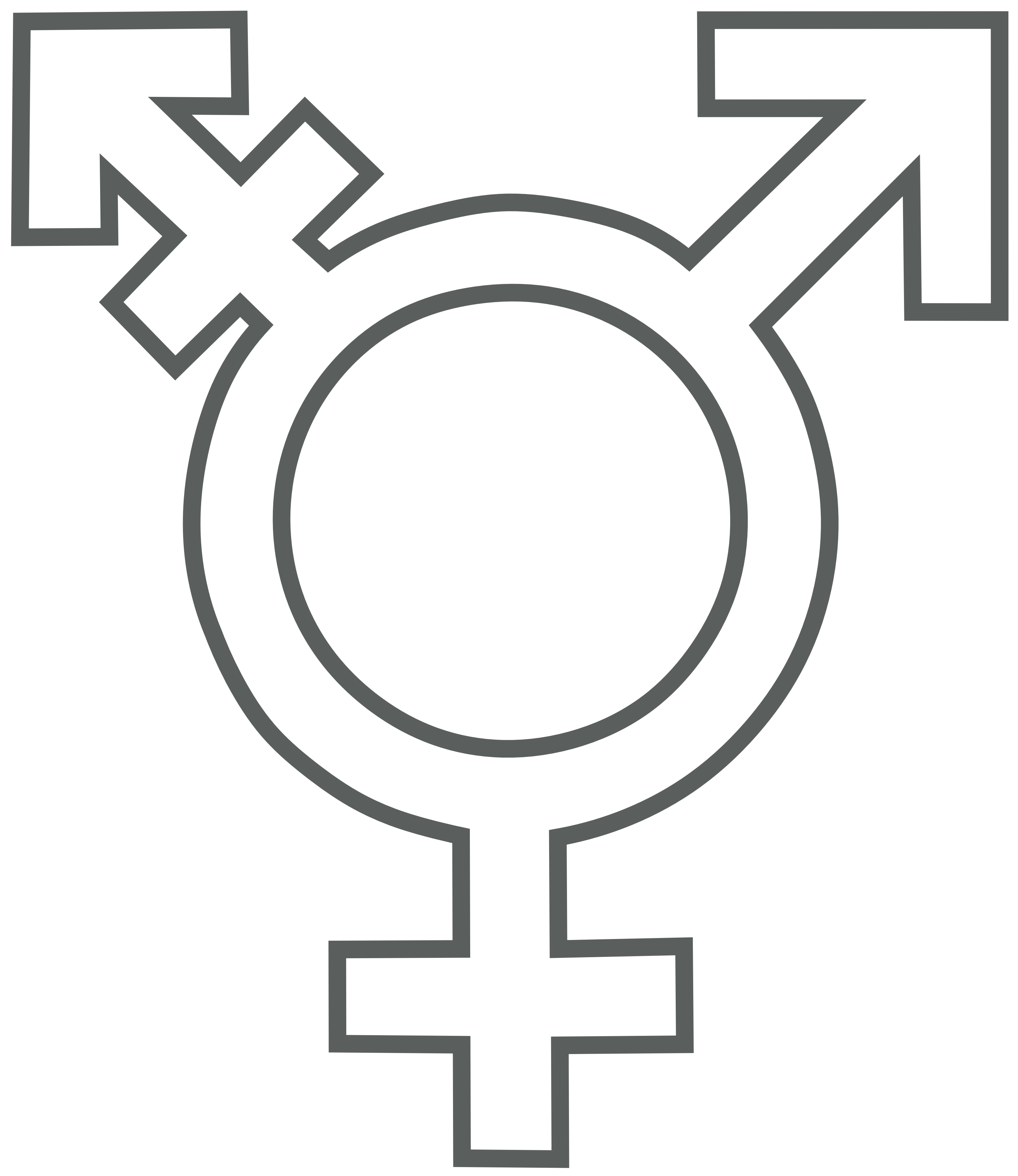


Data: G.Farin, Curves and Surfaces for Computer Aided Geometric Design

SPLINES



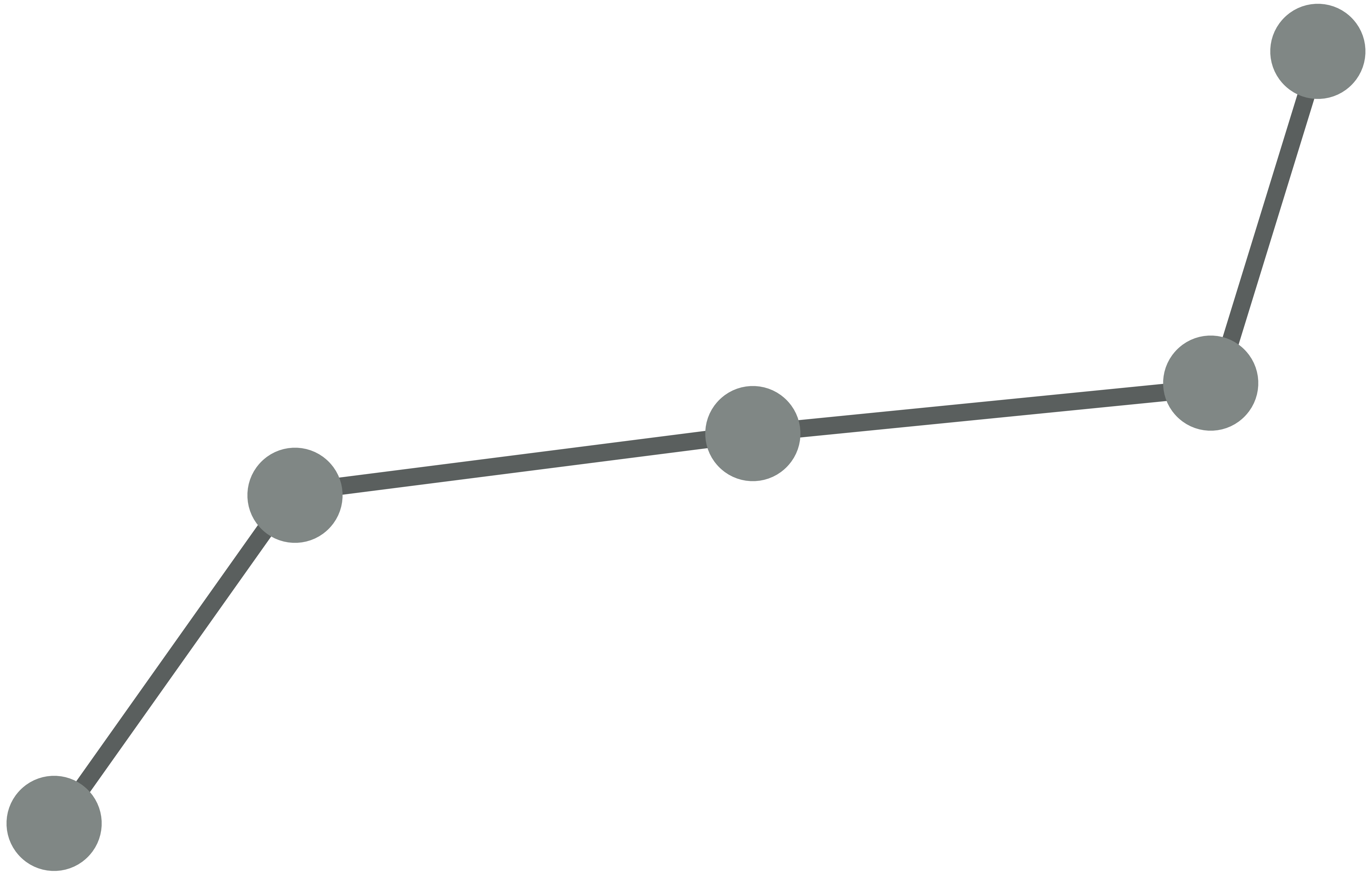
SPLINES



SPLINES

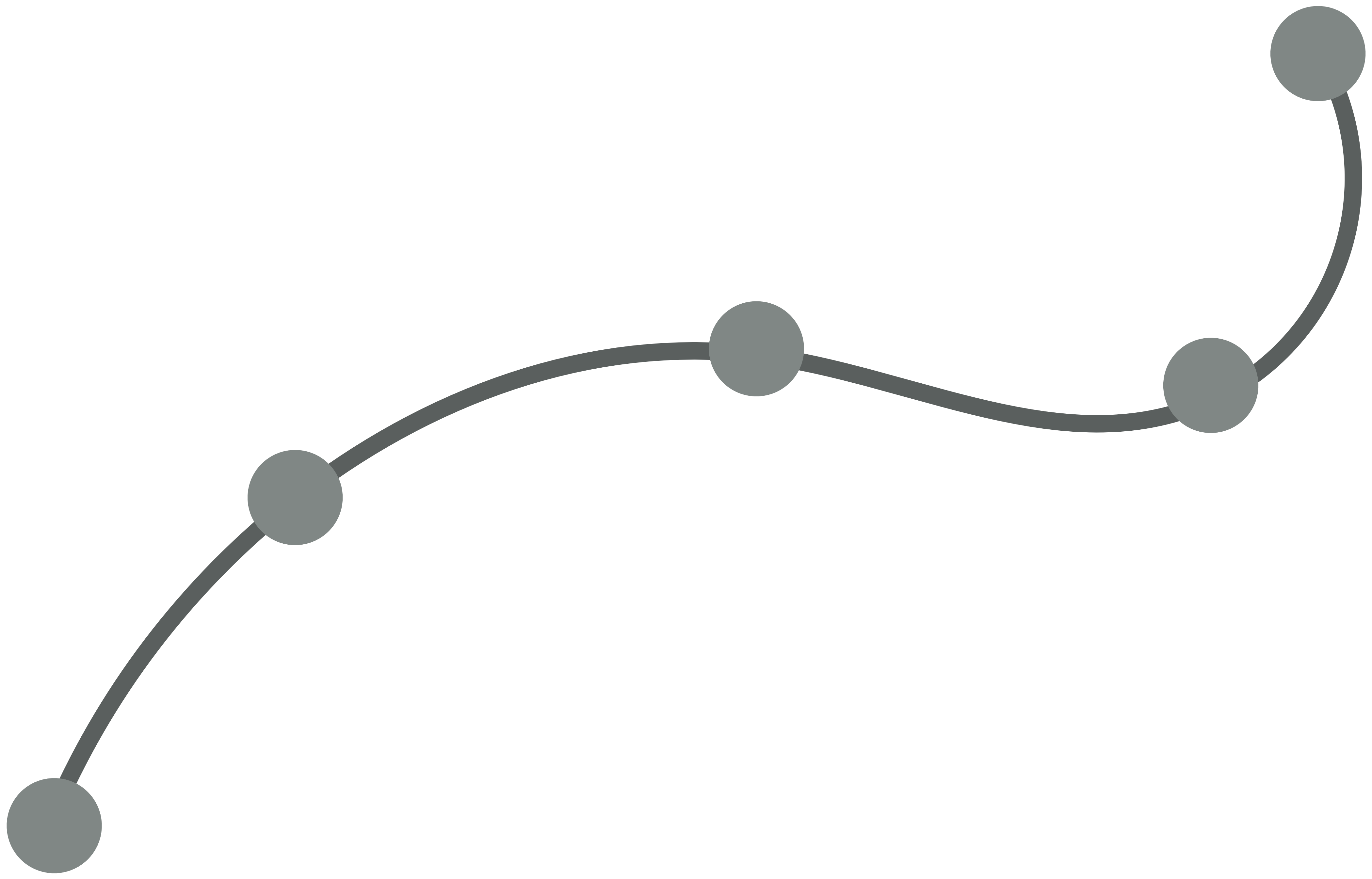
A Class of C2 Interpolating Splines - Paper Presentation at SIGGRAPH 2020

SPLINES



- ✓ Easy to query
- ✓ Easy intersections
- ✗ Differential quantities?
- ✗ Looks bad! or Needs many points!

SPLINES



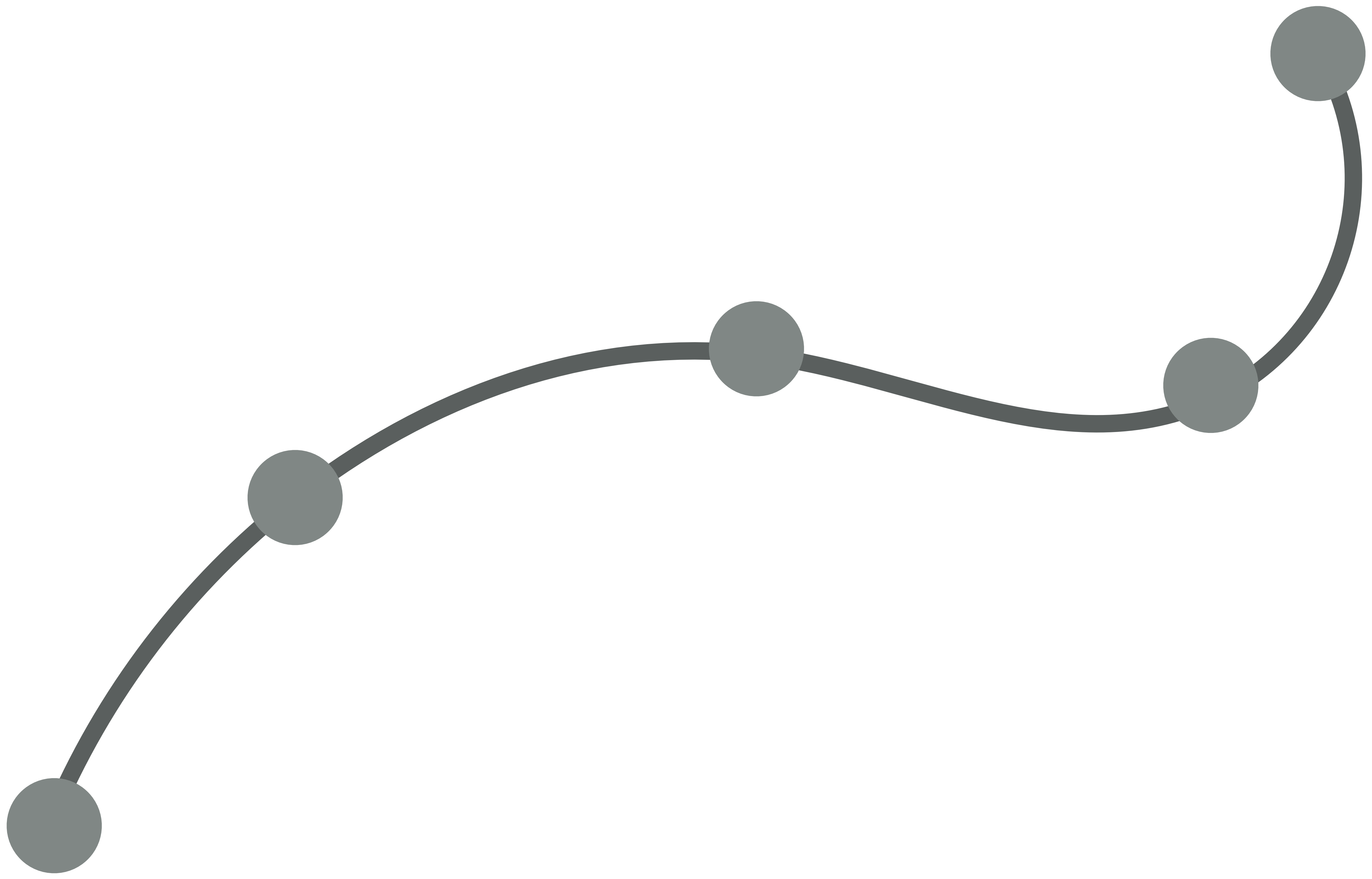
Easy to query

Easy intersections

Differential quantities?

Looks bad! or
Needs many points!

SPLINES



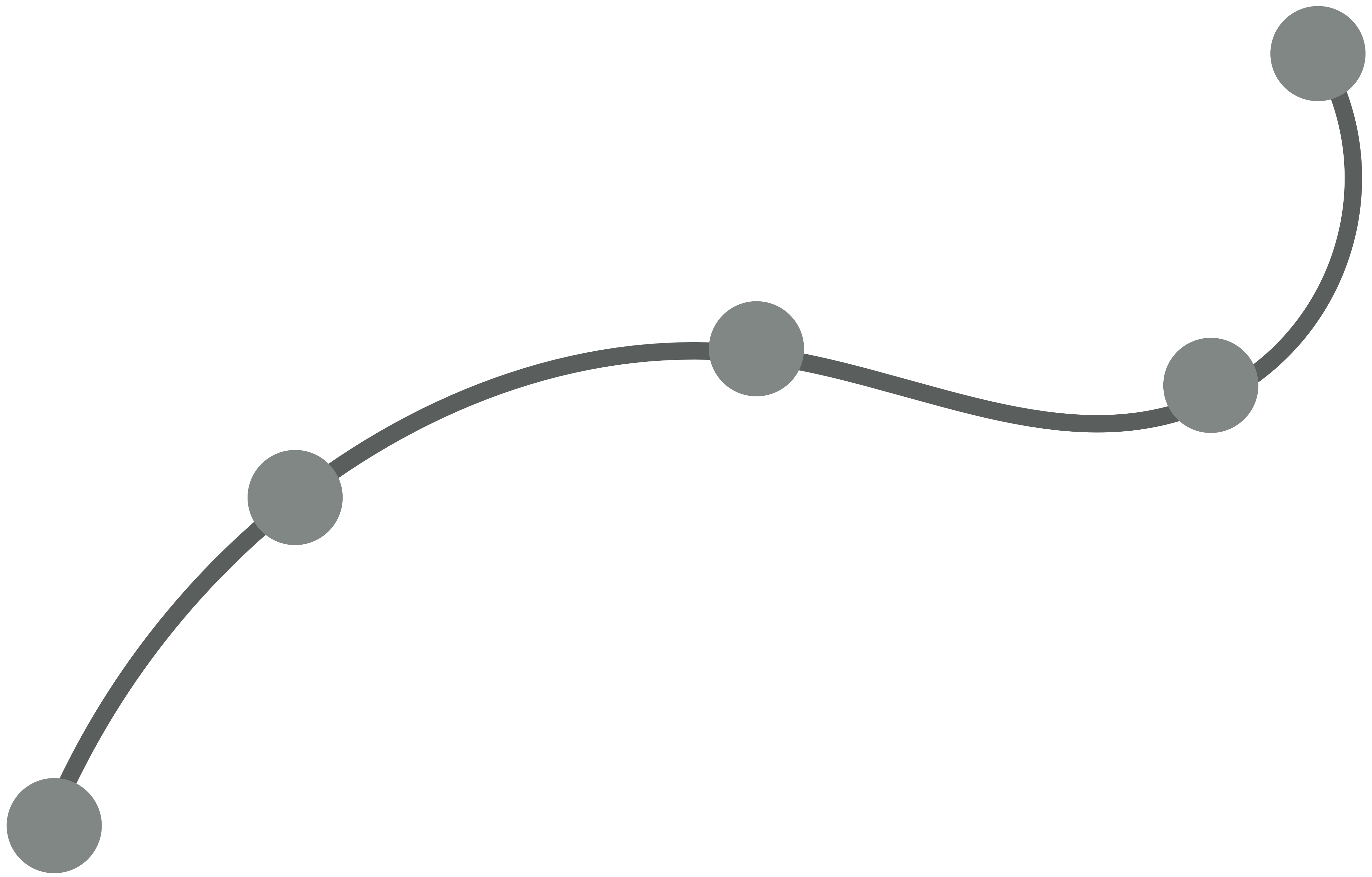
Easy to query

Easy intersections

✓ Differential quantities?

Looks bad! or
Needs many points!

SPLINES



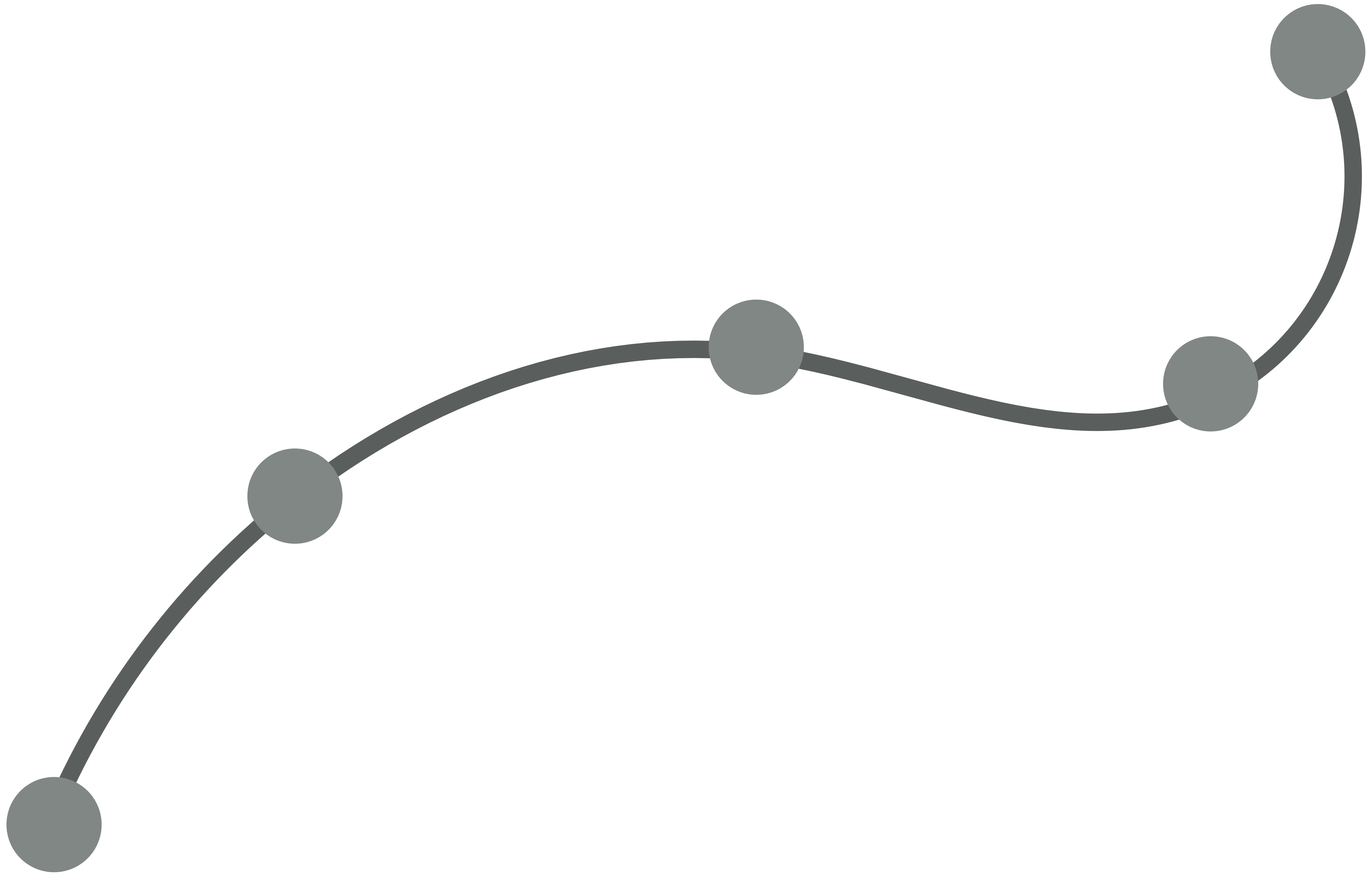
Easy to query

Easy intersections

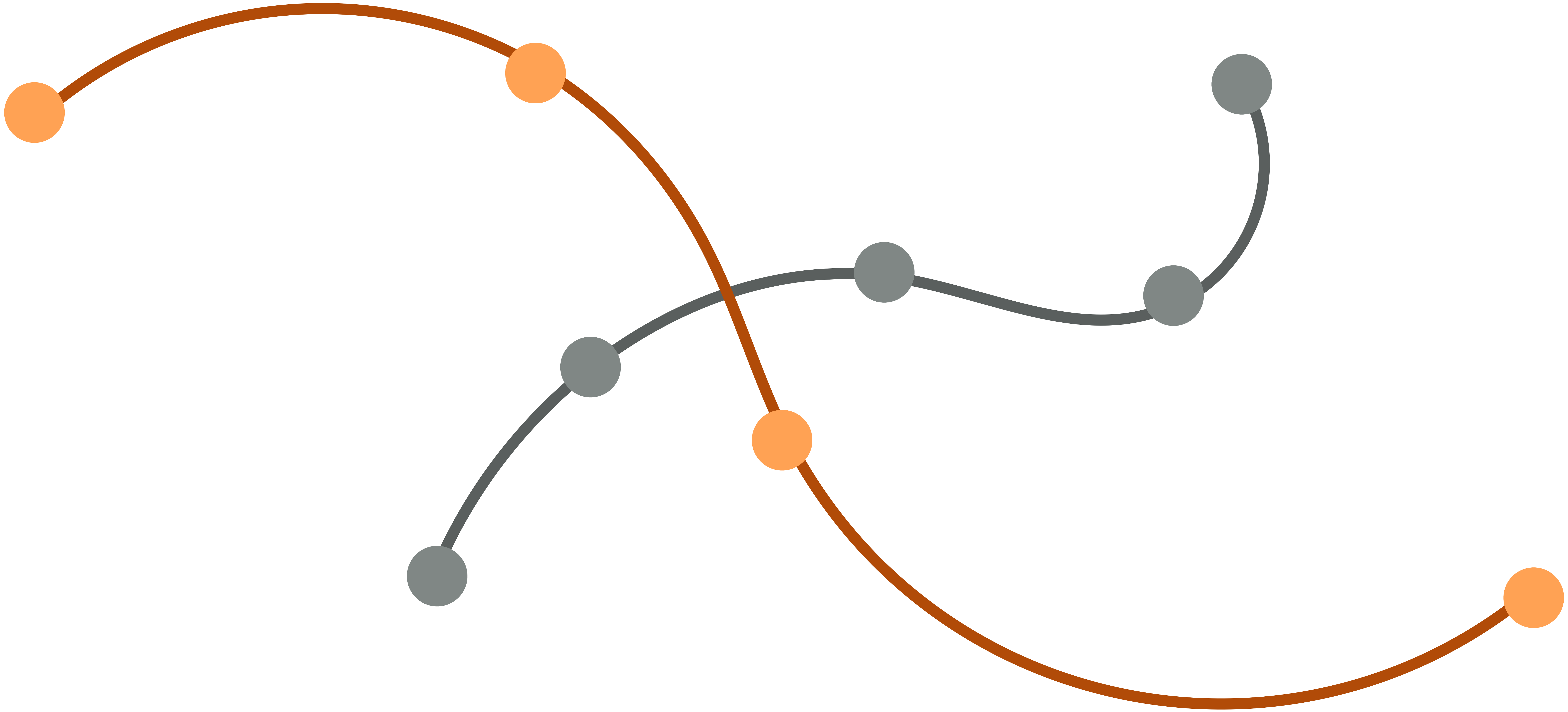
✓ Differential quantities?

✓ Looks great
With few points

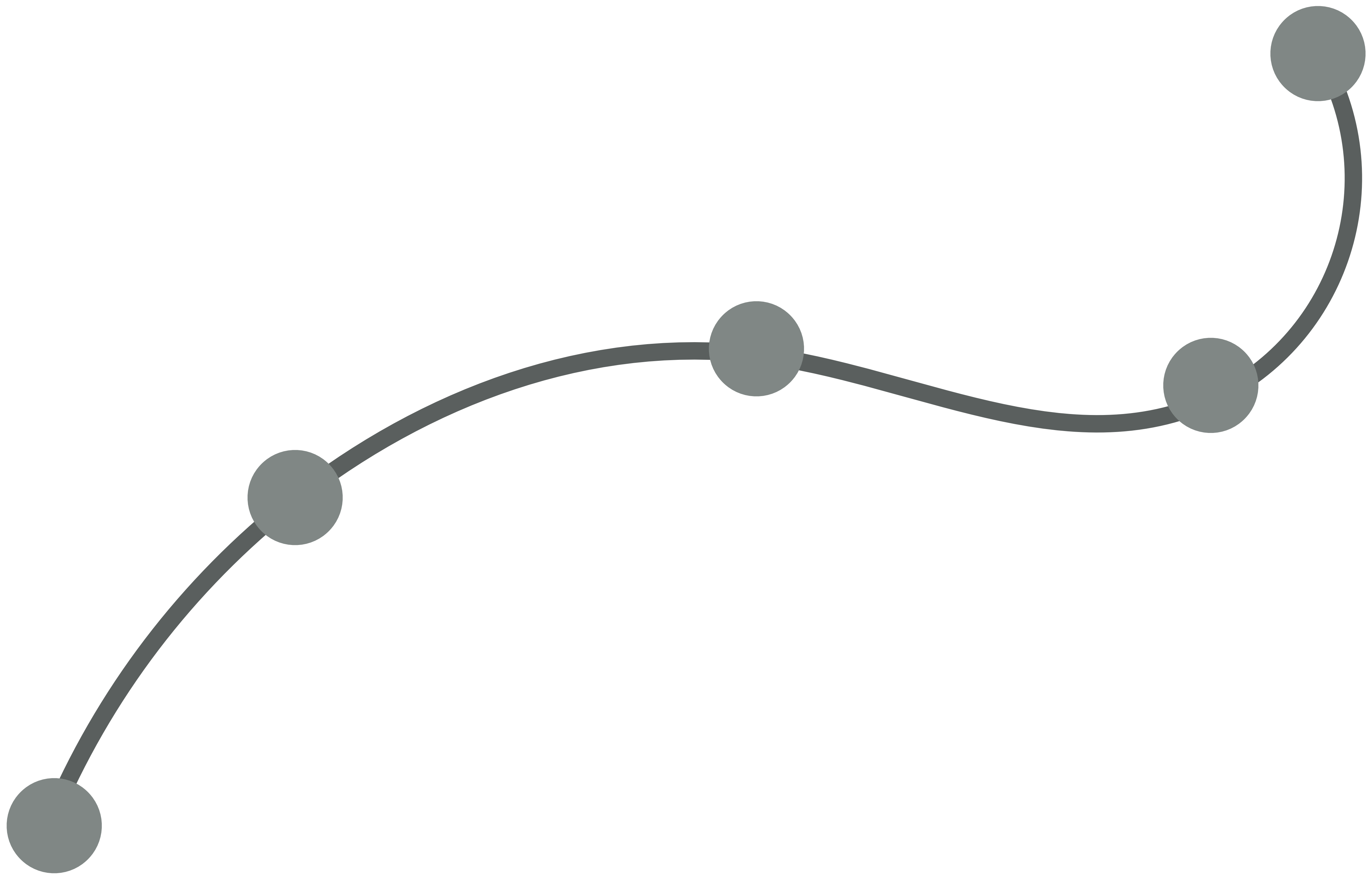
SPLINES



SPLINES



SPLINES



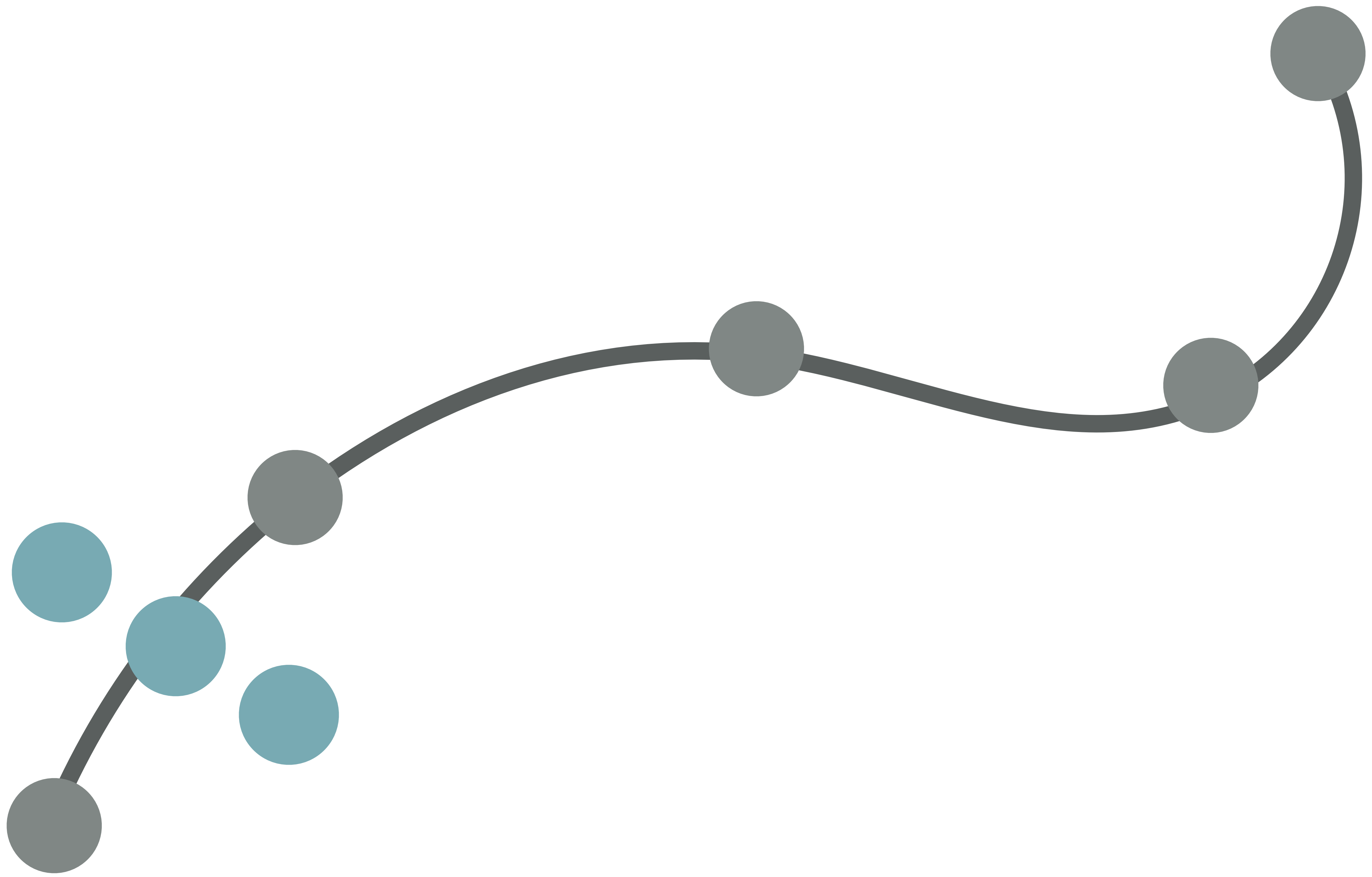
Easy to query

✗ Easy intersections

✓ Differential quantities?

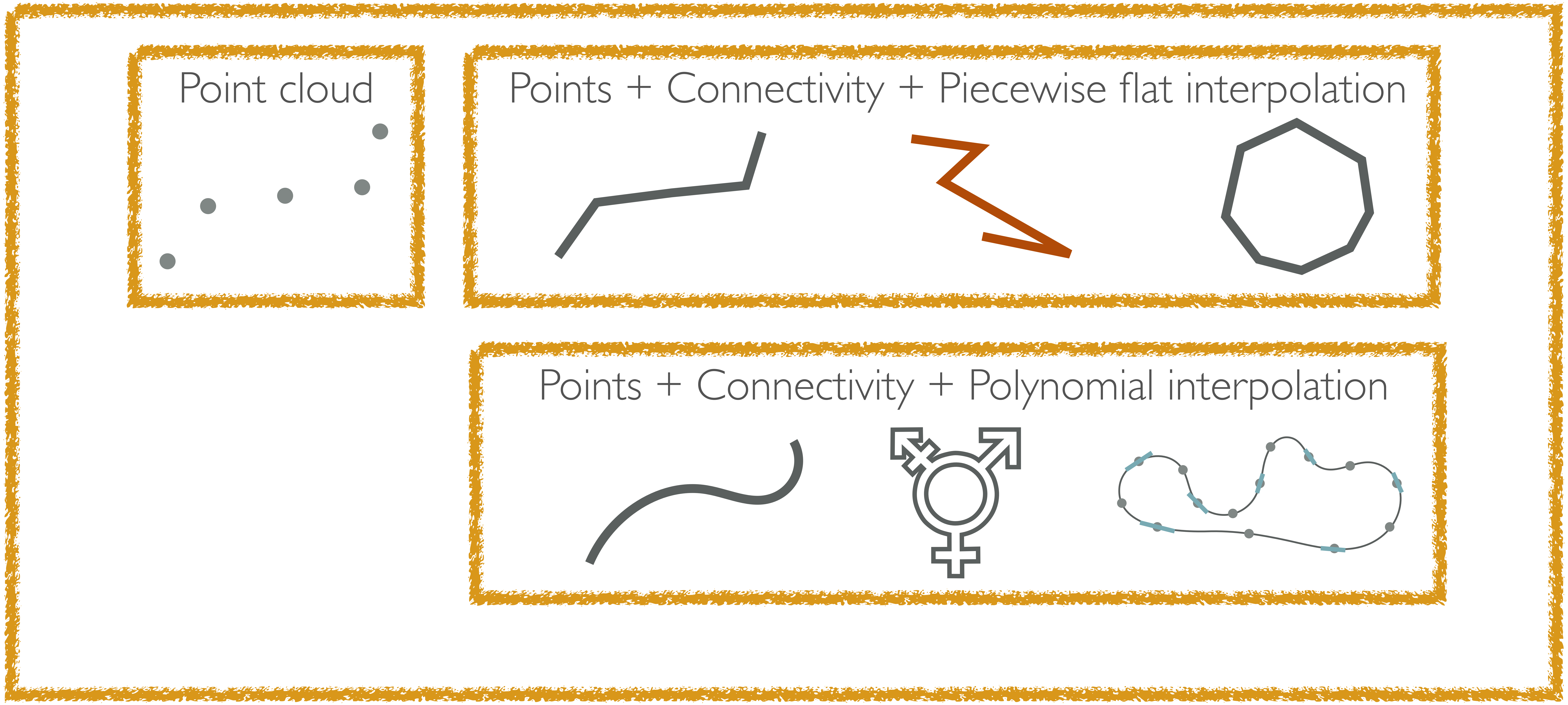
✓ Looks great
With few points

SPLINES



- ✗ Easy to query
- ✗ Easy intersections
- ✓ Differential quantities?
- ✓ Looks great With few points

SHAPE REPRESENTATIONS



CLIPasso: Semantically-Aware Object Sketching

Yael Vinker[2,1] Ehsan Pajouheshgar[1] Jessica Y. Bo[1] Roman Christian Bachmann[1]
Amit Haim Bermano[2] Daniel Cohen-Or[2] Amir Zamir[1] Ariel Shamir[3]

[1]Swiss Federal Institute of Technology (EPFL) [2]Tel-Aviv University [3]Reichman University

https://clipasso.github.io/clipasso/

Abstract is at the heart of sketching due to the simple and minimal nature of line drawings. Abstraction entails identifying the essential visual properties of an object or scene, which requires semantic understanding and prior knowledge of high-level concepts. Abstract depictions are therefore challenging for artists, and even more so for machines. We present CLIPasso, an object sketching method that can achieve different levels of abstraction, guided by geometric and semantic simplifications. While sketch generation methods often rely on explicit sketch datasets for training, we utilize the remarkable ability of CLIP (Contrastive-Language-Image-Pretraining) to distill semantic concepts from sketches and images alike. We define a sketch as a set of Bézier curves and use a differentiable rasterizer to optimize the parameters of the curves directly with respect to a CLIP-based perceptual loss. The abstraction degree is controlled by varying the number of strokes. The generated sketches demonstrate multiple levels of abstraction while maintaining recognizability, underlying structure, and essential visual components of the subject drawn.

Free-hand sketching is a valuable visual tool for expressing ideas, concepts, and actions [10,15,18,43]. As sketches consist of only strokes, and often only a limited number of strokes, the process of *abstraction* is central to sketching. An artist must make representational decisions to choose key visual features of the subject drawn to capture the relevant information she wishes to express, while omitting (many) others [6]. For example, in the famous "Le Taureau" series (Figure 2), Picasso depicts the progressive abstraction of a bull. In this series of lithographs, the artist transforms a bull from a concrete, fully rendered, anatomical drawing, into a sketch composition of a few lines that still manages to capture the essence of the bull.

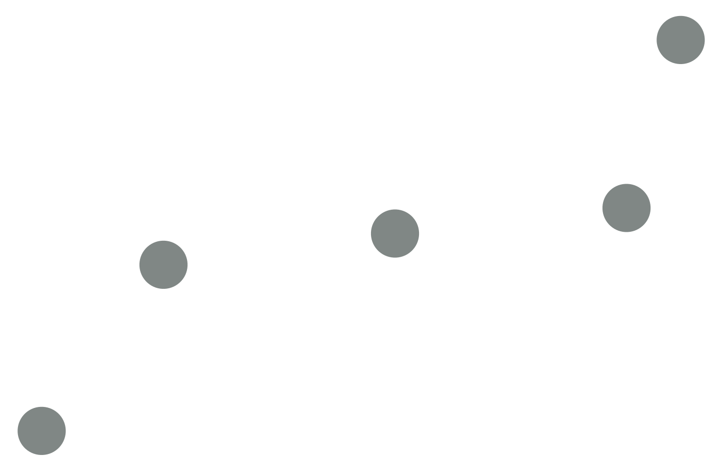
In this paper, we pose the question — can computer renderings imitate such a process of sketching abstraction, converting a photograph from a concrete depiction to an abstract one?

Today, machines can render realistic sketches simply by applying mathematical and geometric operations to an input photograph [5,42]. However, creating abstractions is more difficult for machines to achieve. The abstraction process suggests that the artist selects visual features that capture

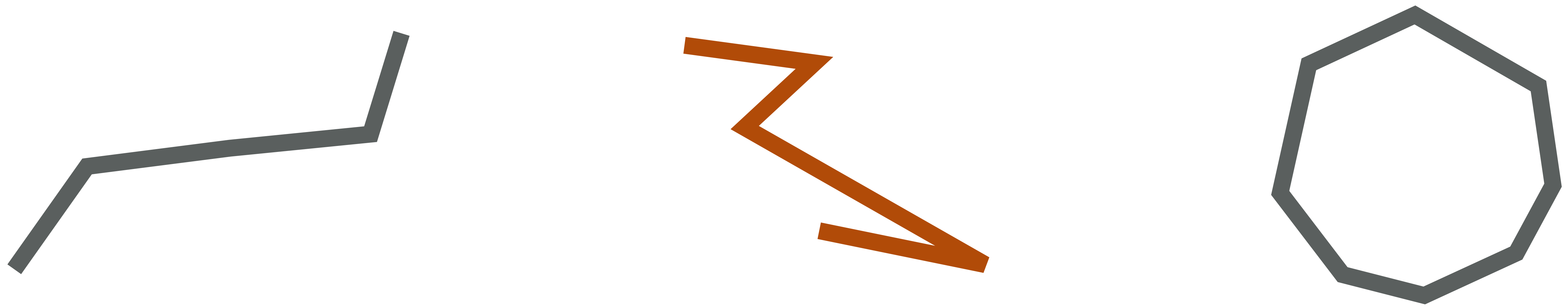
https://replicate.com/yael-vinker/clipasso

EXPLICIT SHAPE REPRESENTATIONS

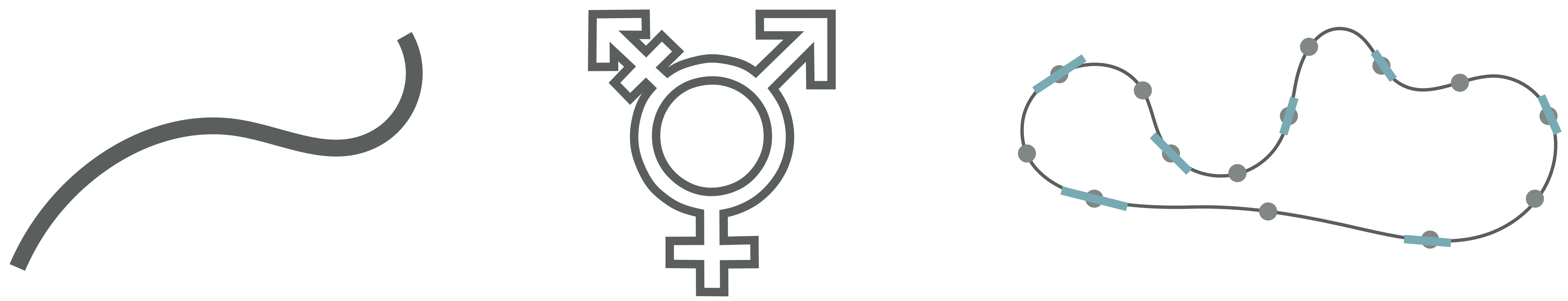
Point cloud



Points + Connectivity + Piecewise flat interpolation



Points + Connectivity + Polynomial interpolation



IMPLICIT SHAPE REPRESENTATIONS

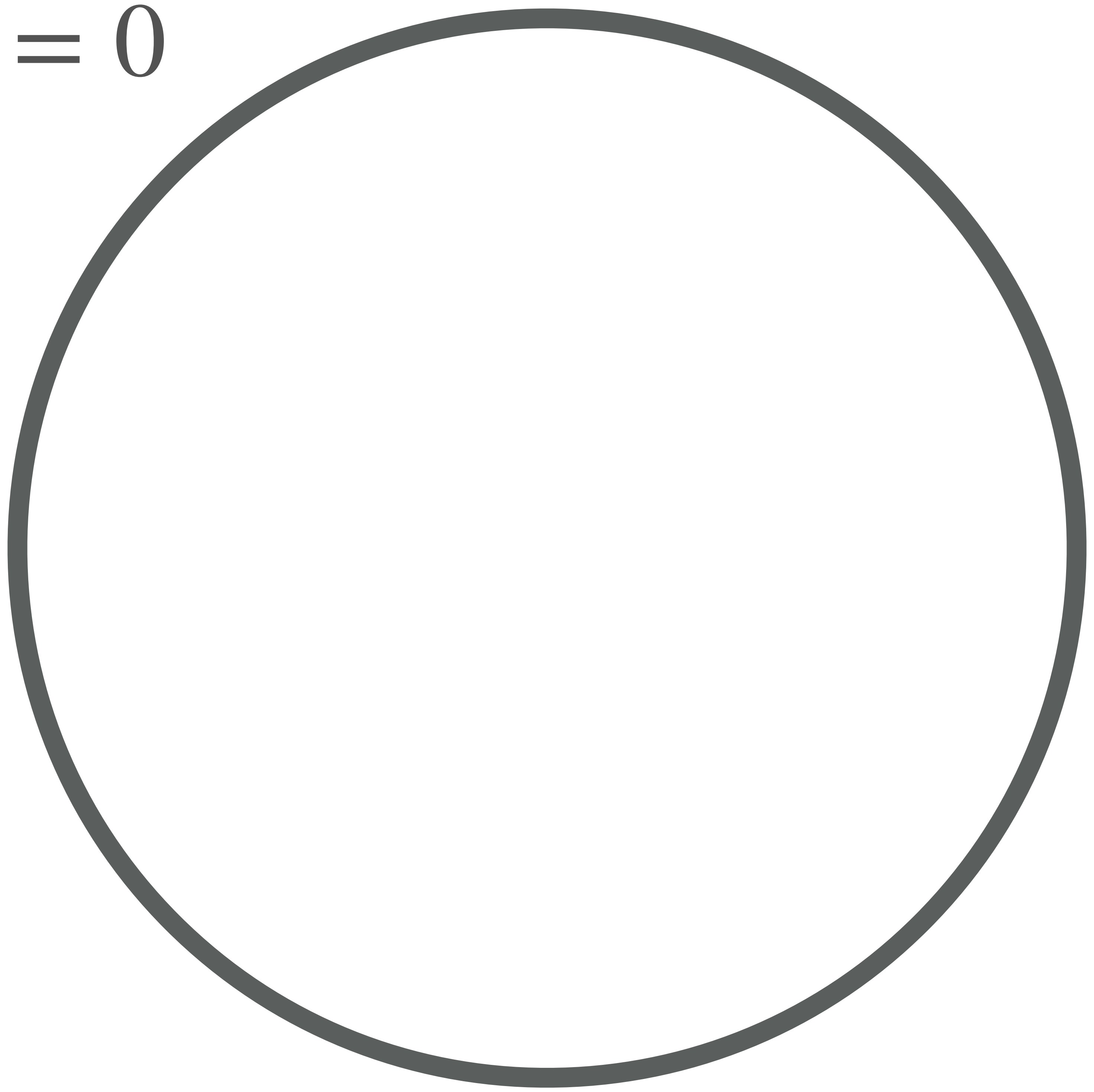
IMPLICIT SHAPE REPRESENTATIONS

$$f(x, y) = x^2 + y^2 - 1$$

IMPLICIT SHAPE REPRESENTATIONS

$$f(x, y) = x^2 + y^2 - 1$$

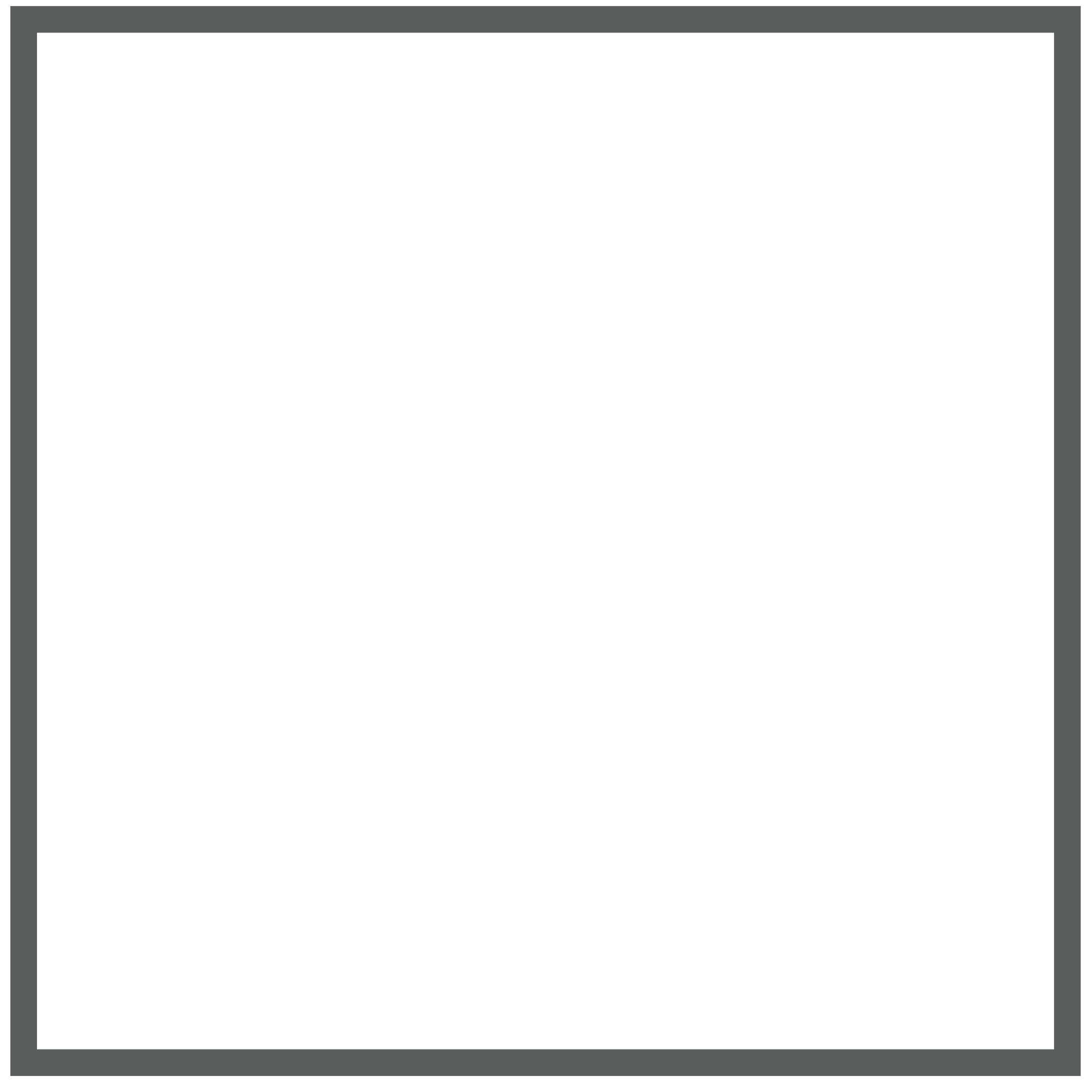
$$f(x, y) = 0$$



IMPLICIT SHAPE REPRESENTATIONS

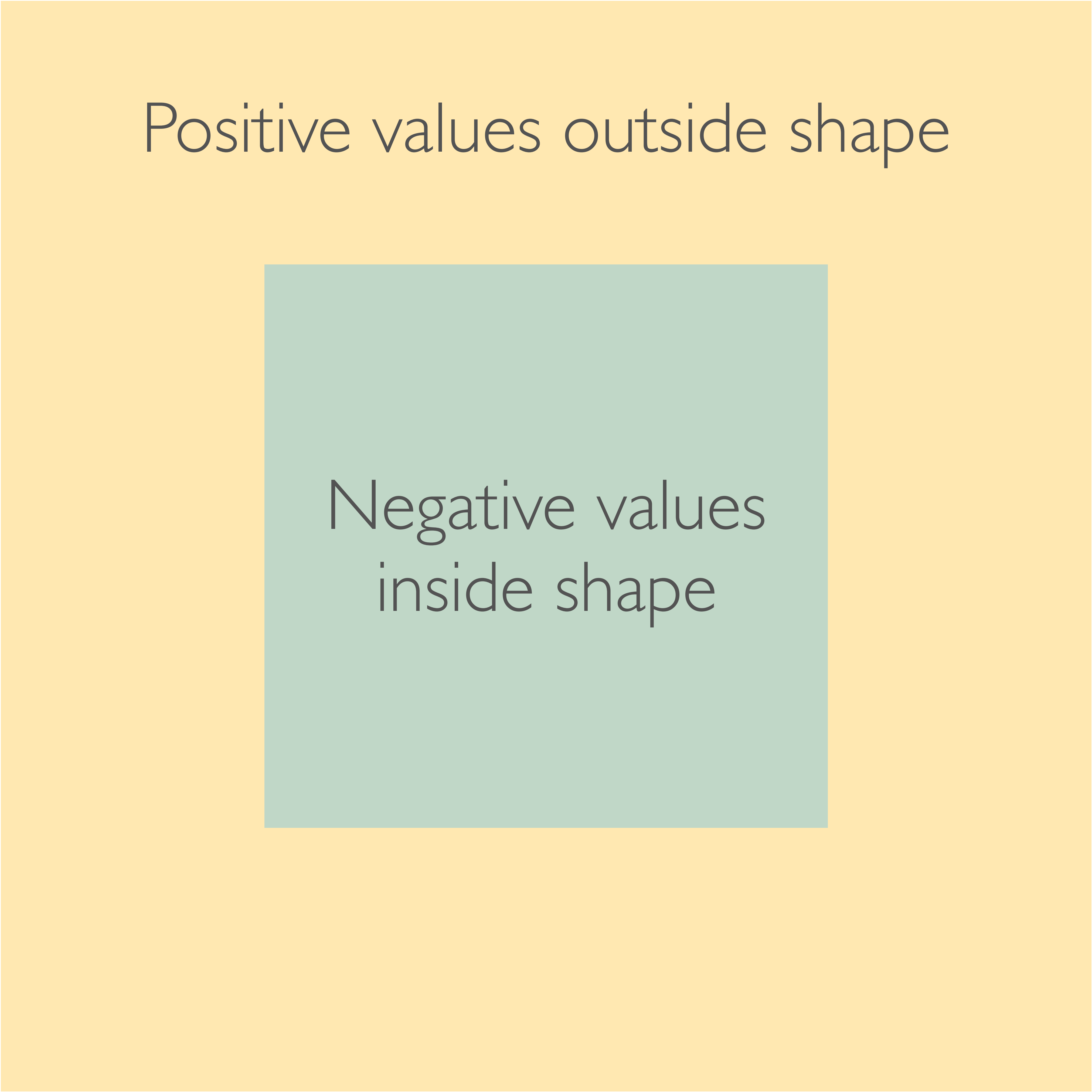
$f(x, y) = 0$

$f(x, y) = \max(\ \ x\ \ ,\ \ y\ \ ) - 1$



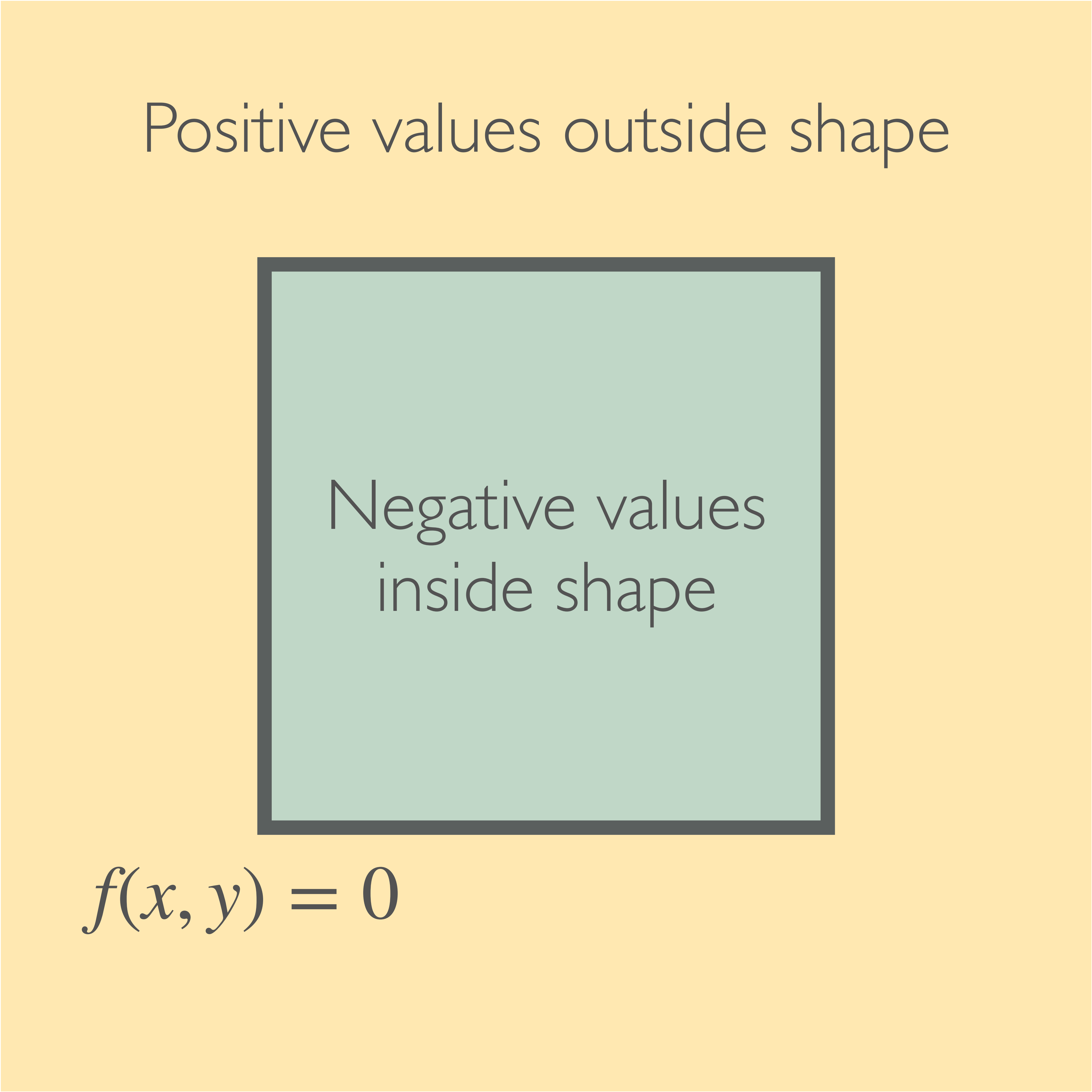
IMPLICIT SHAPE REPRESENTATIONS

$$f(x, y) = \max(\quad x \quad , \quad y \quad) - 1$$



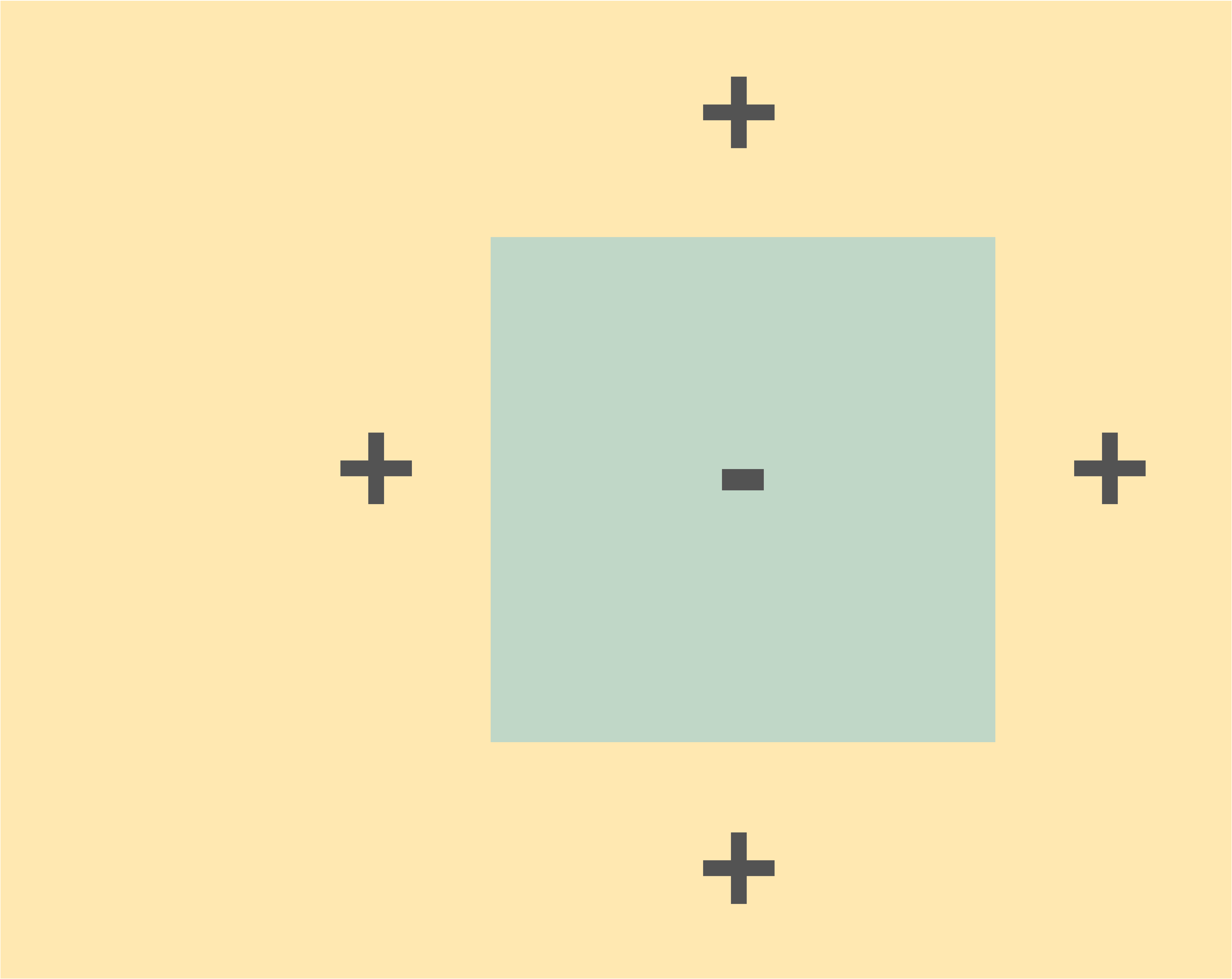
IMPLICIT SHAPE REPRESENTATIONS

$$f(x, y) = \max(\ \ x\ \ ,\ \ y\ \ ) - 1$$



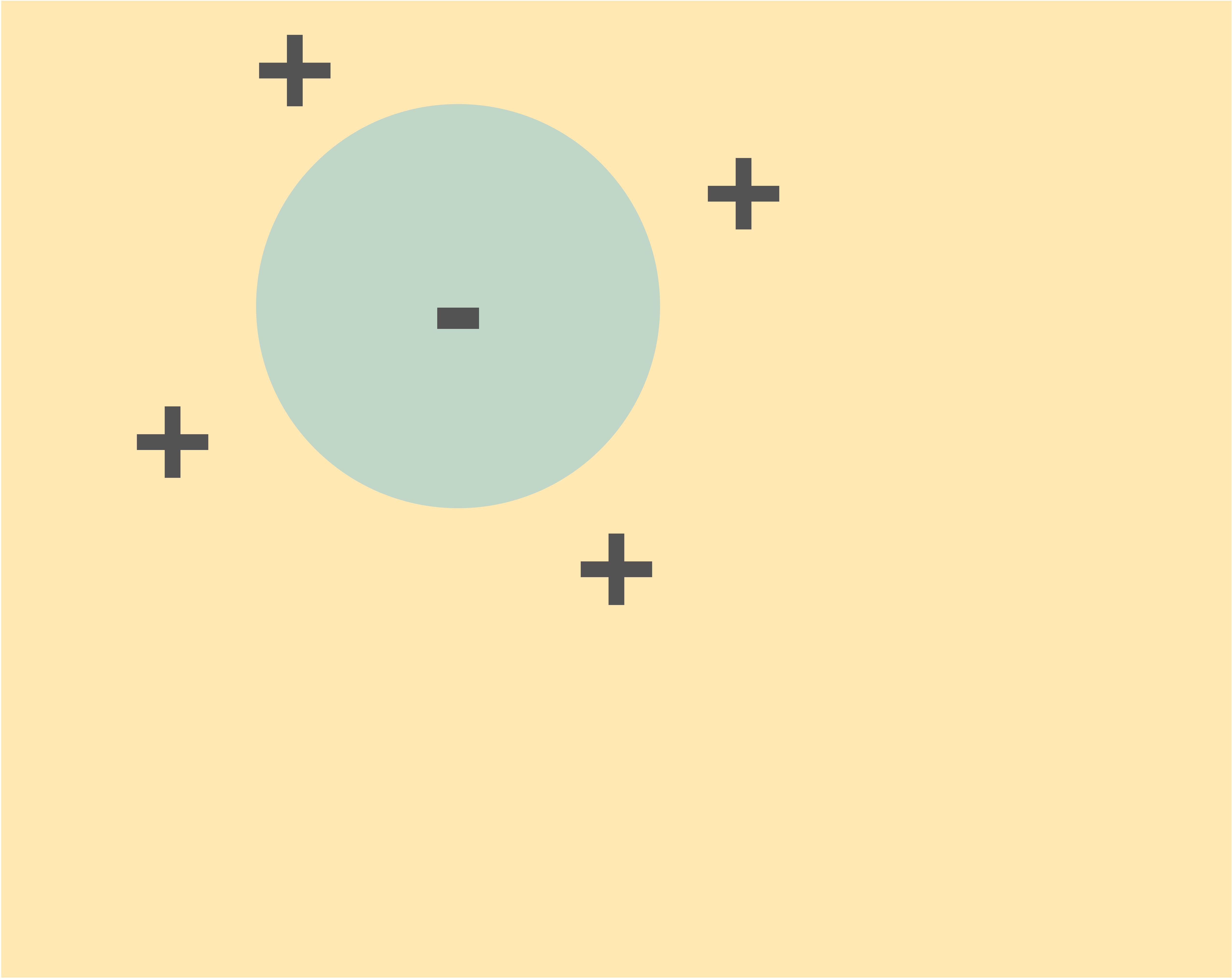
IMPLICIT SHAPE REPRESENTATIONS

$$f_1(x, y) = \max(\ x\ ,\ y\ ) - 1$$



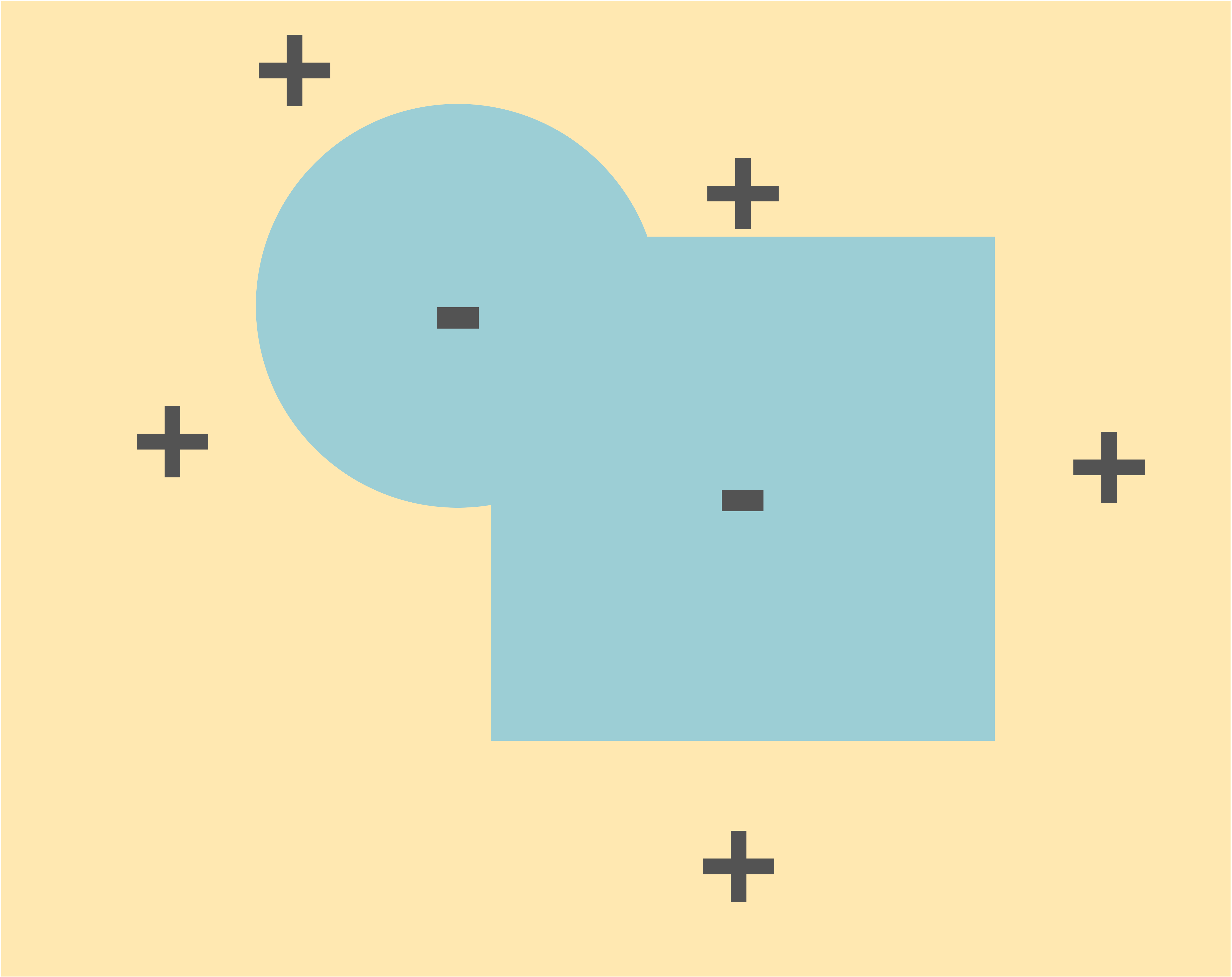
IMPLICIT SHAPE REPRESENTATIONS

$$f_2(x, y) = x^2 + y^2 - 1$$



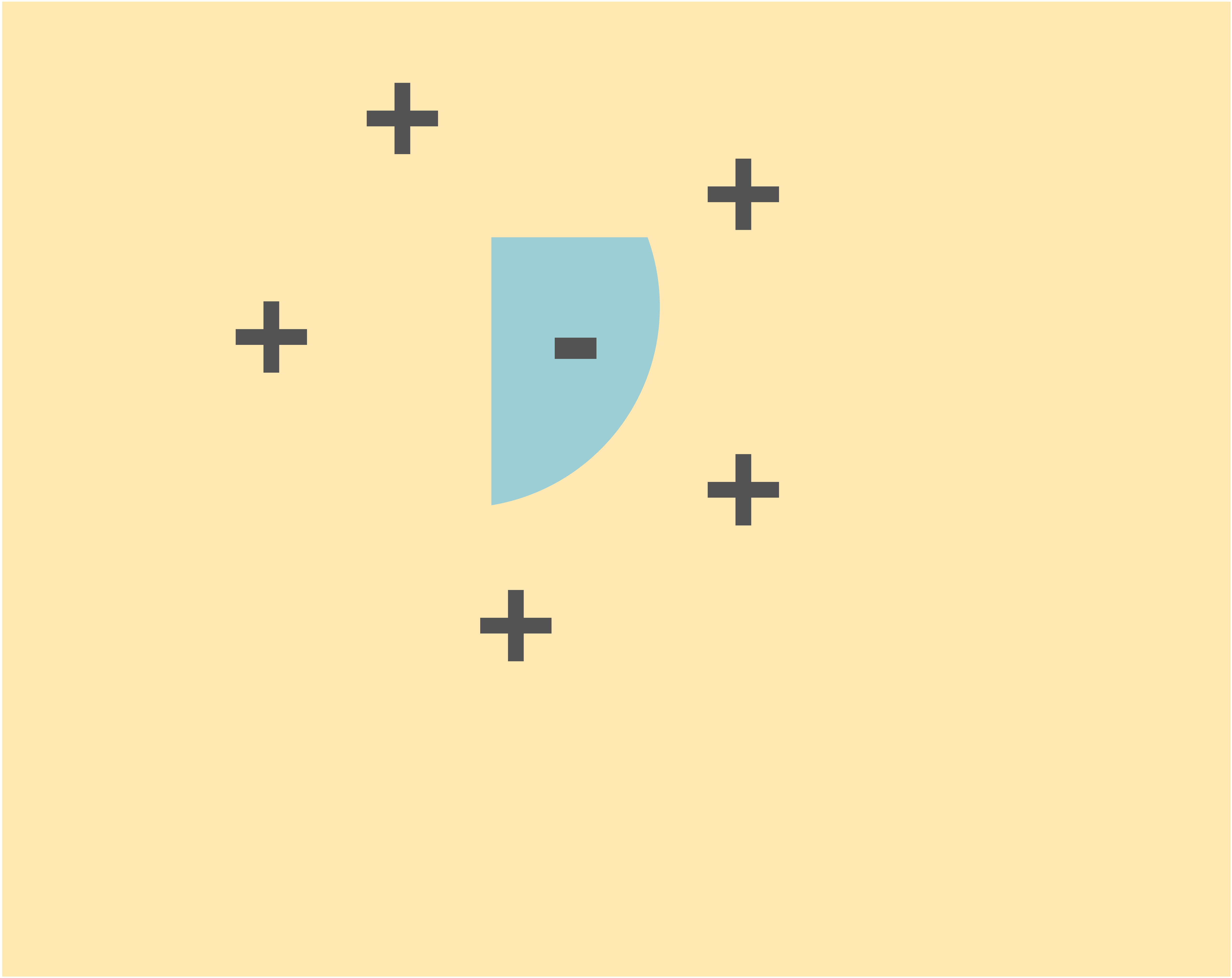
IMPLICIT SHAPE REPRESENTATIONS

$$f_{union} = \min(f_1, f_2)$$



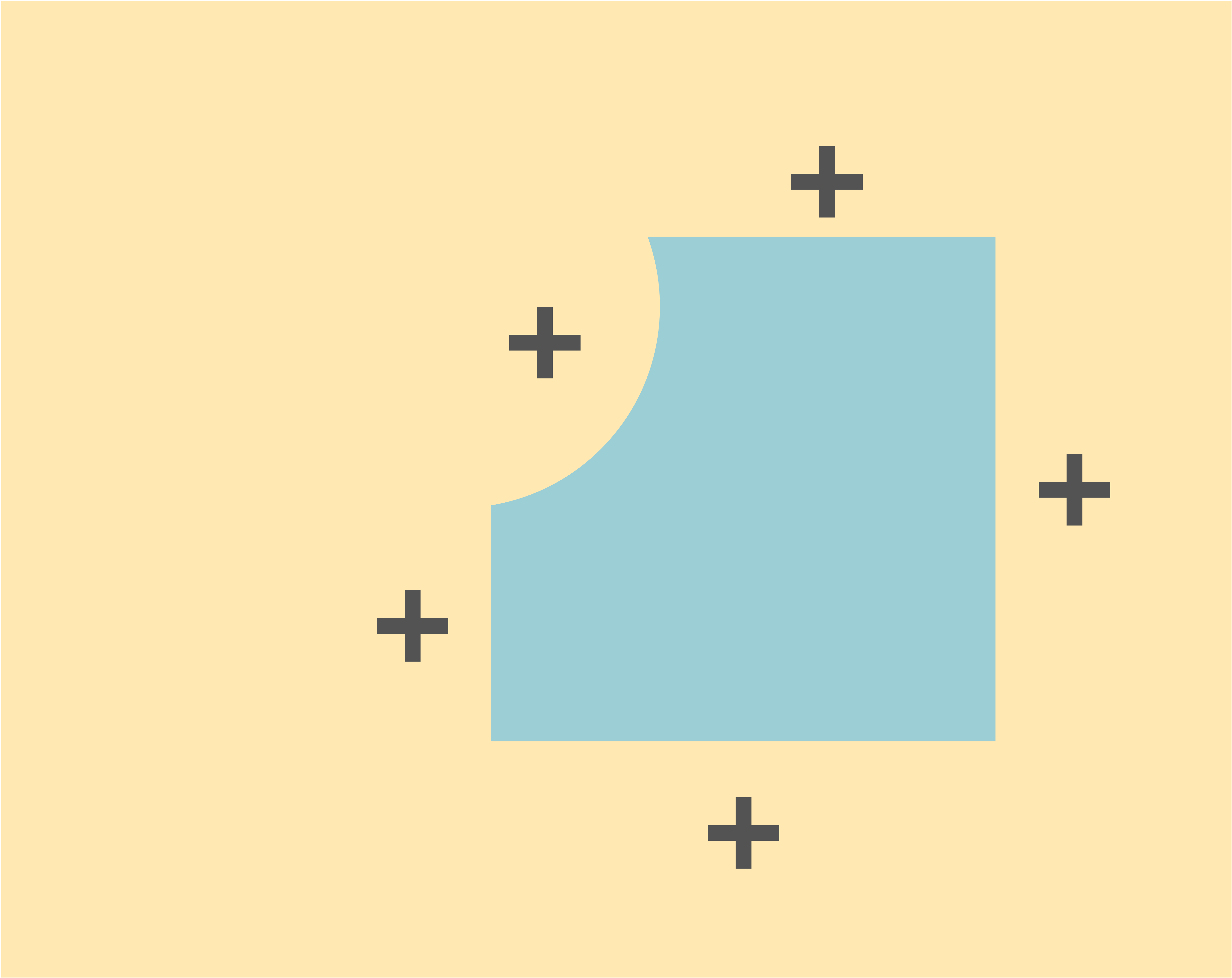
IMPLICIT SHAPE REPRESENTATIONS

$$f_{intersection} = \max(f_1, f_2)$$



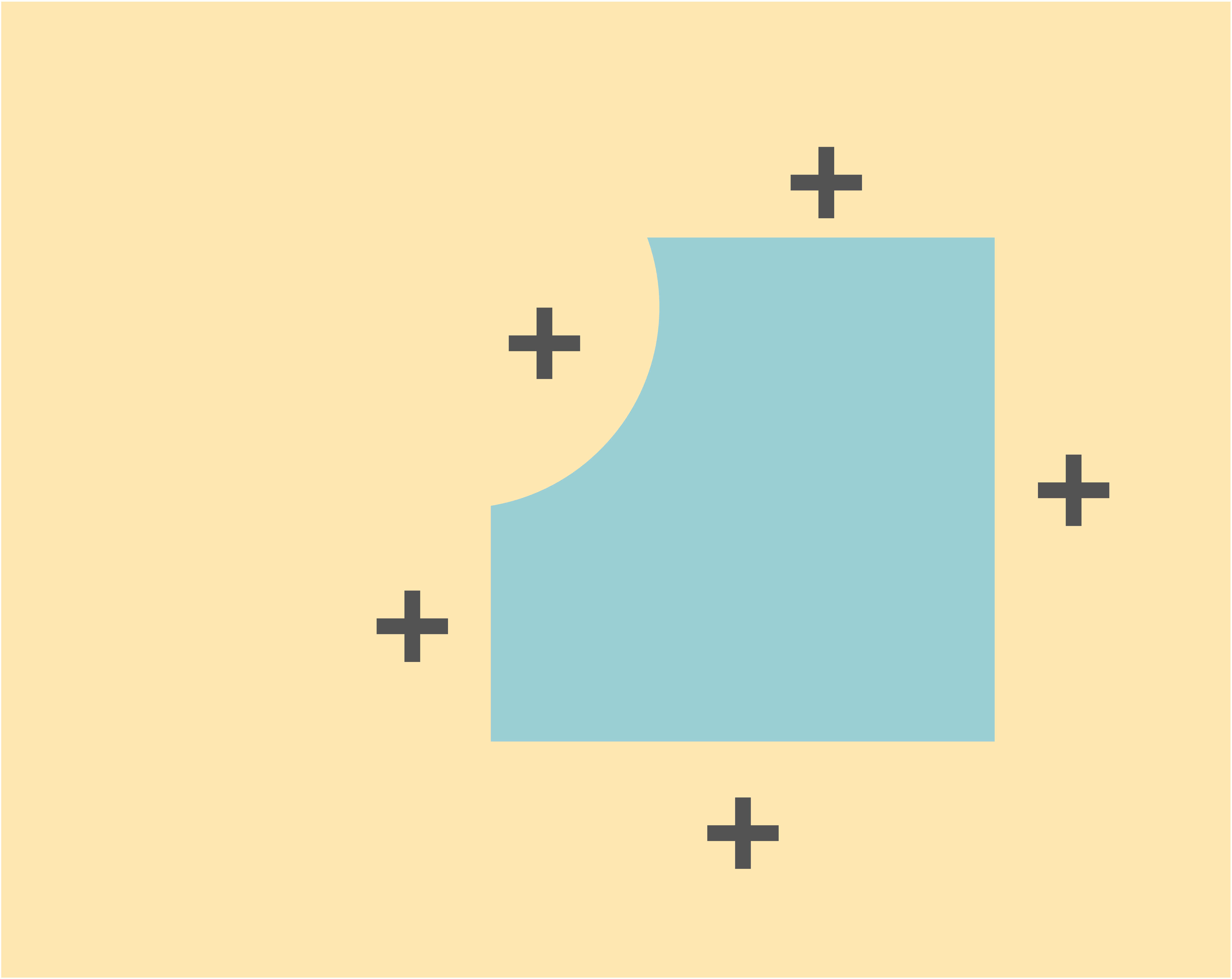
IMPLICIT SHAPE REPRESENTATIONS

$$f_{subtraction} = \max(f_1, -f_2)$$

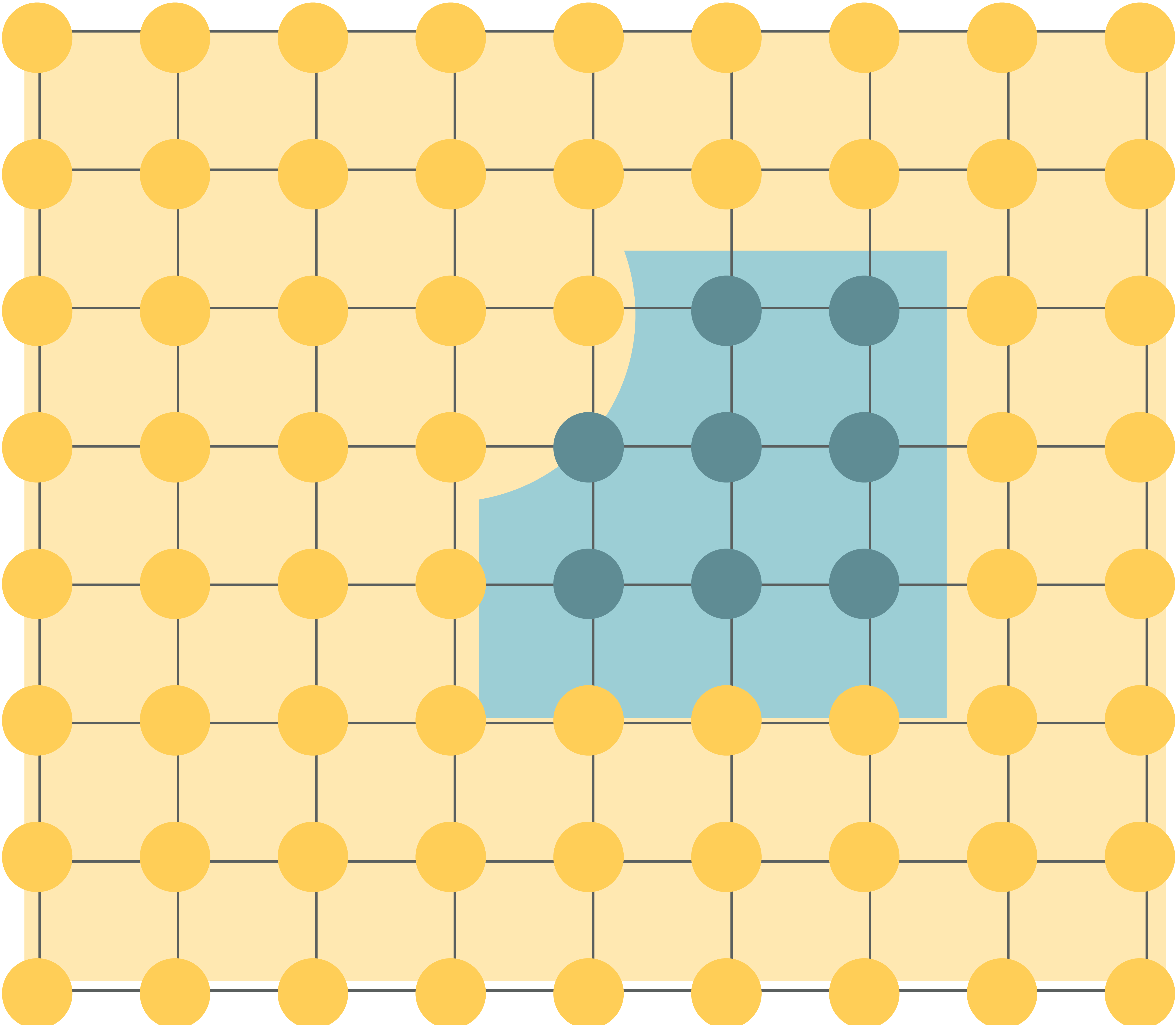


IMPLICIT SHAPE REPRESENTATIONS

✓ Easy boolean operations

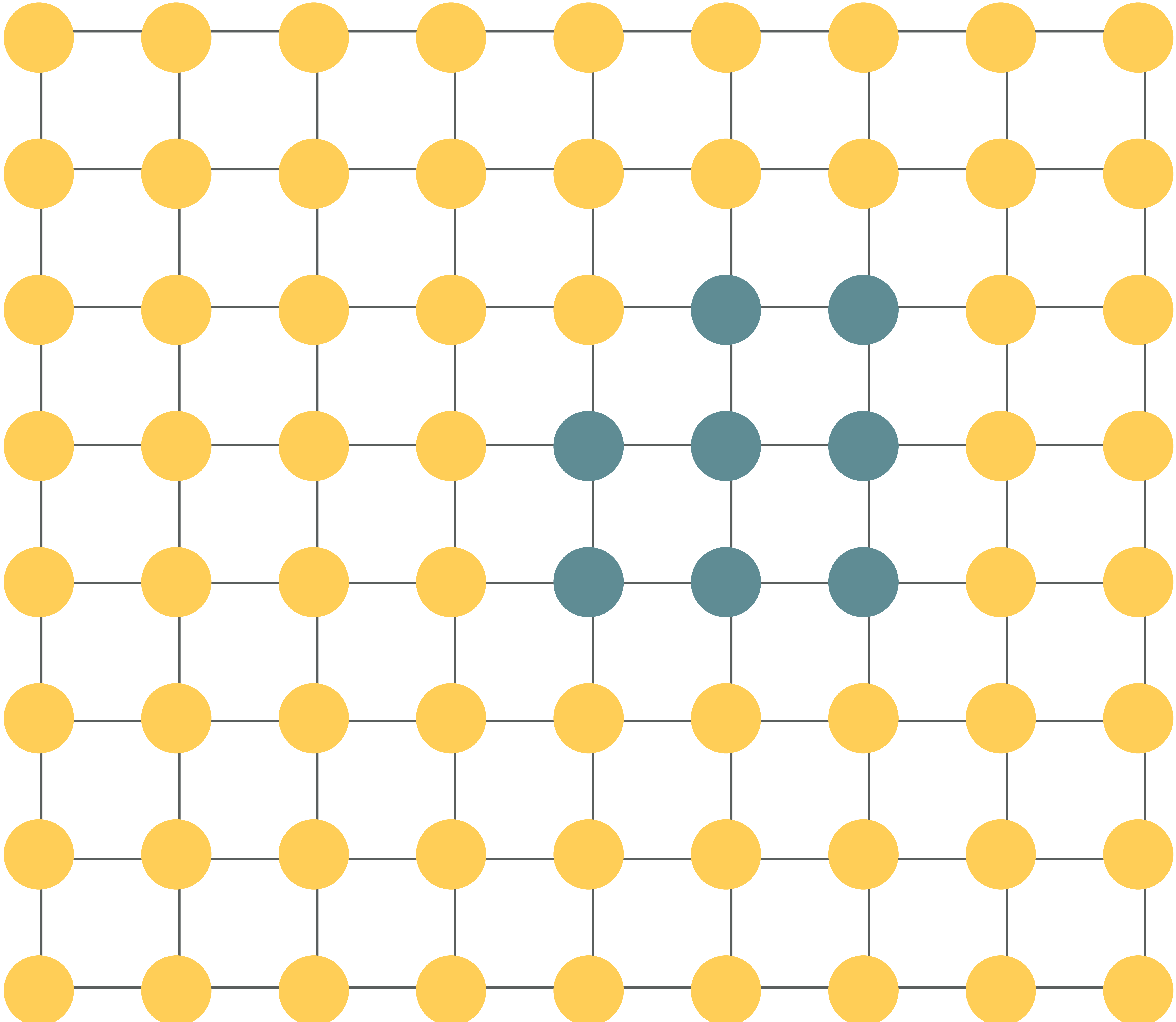


IMPLICIT SHAPE REPRESENTATIONS



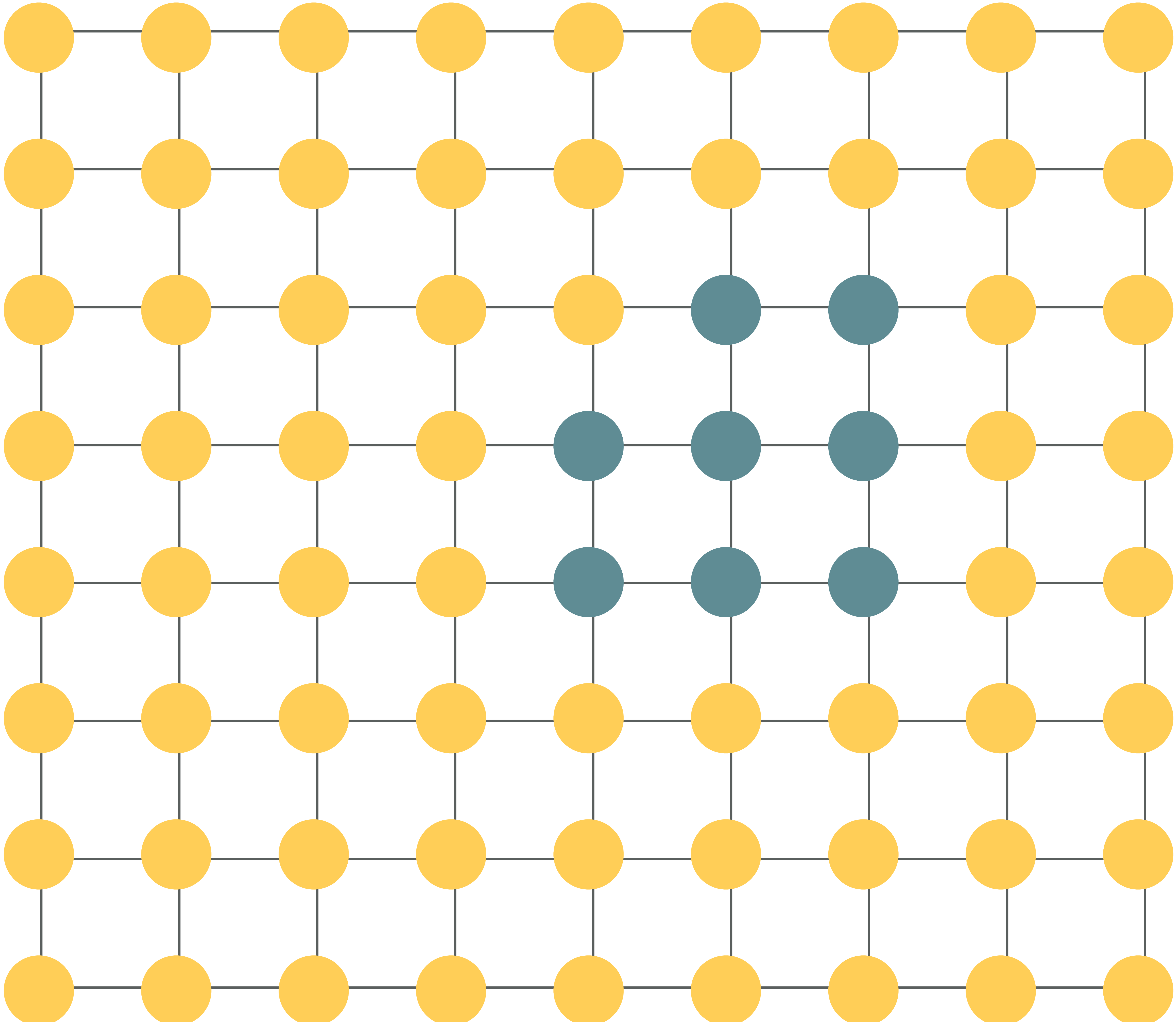
IMPLICIT SHAPE REPRESENTATIONS

Turn geometry into image



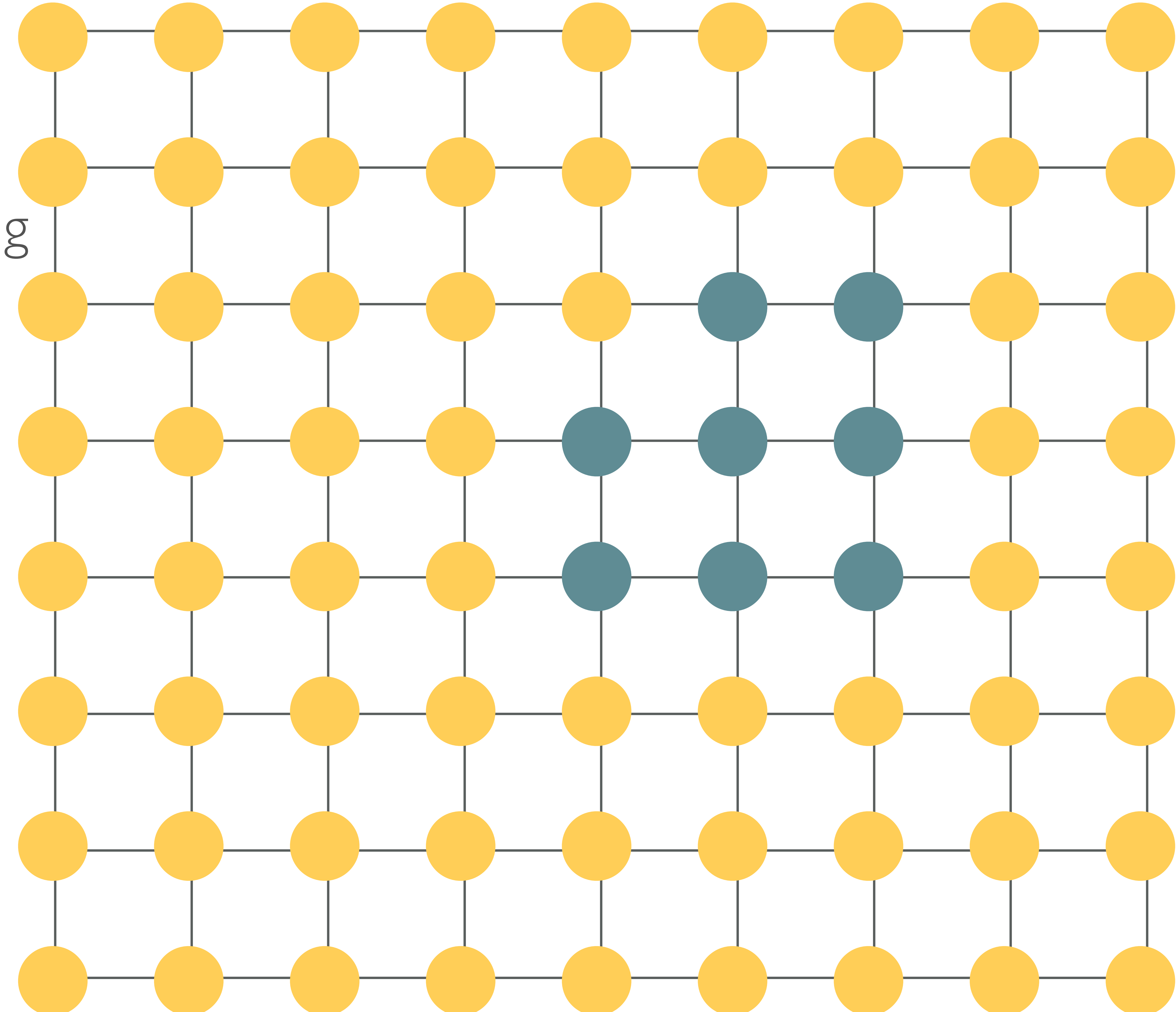
IMPLICIT SHAPE REPRESENTATIONS

Use *image* processing
instead of *geometry* processing



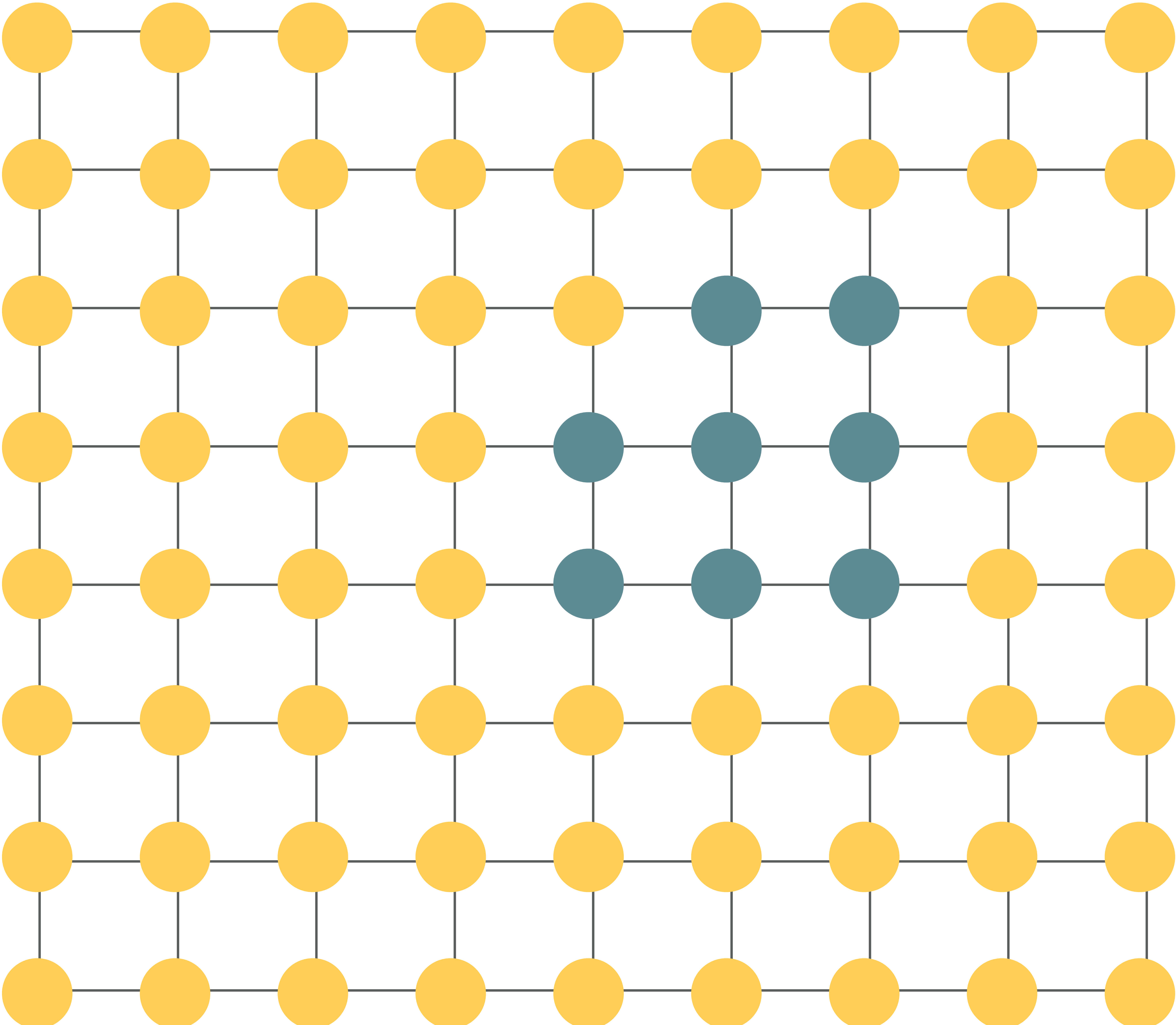
IMPLICIT SHAPE REPRESENTATIONS

- Use *image* processing instead of *geometry* processing



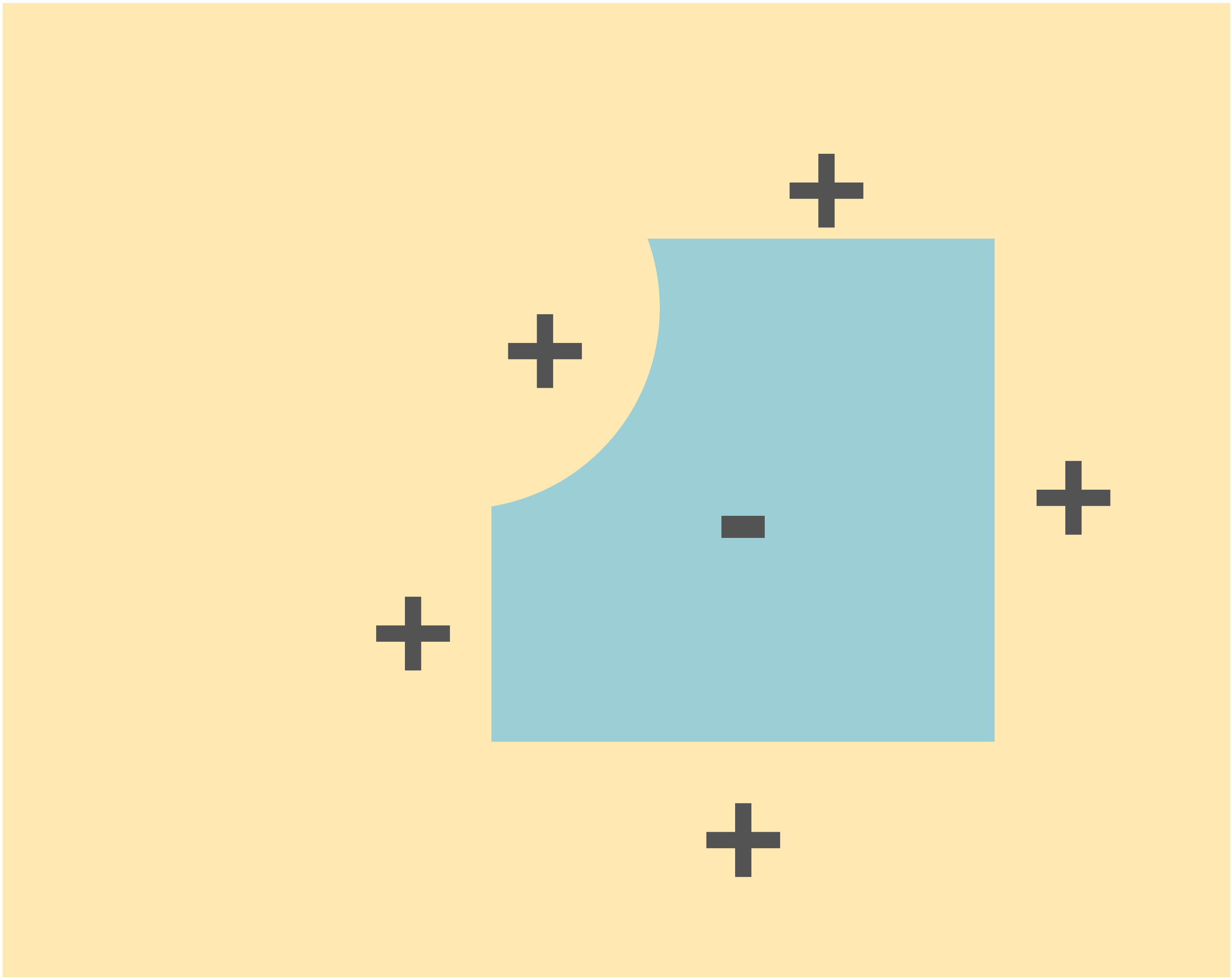
IMPLICIT SHAPE REPRESENTATIONS

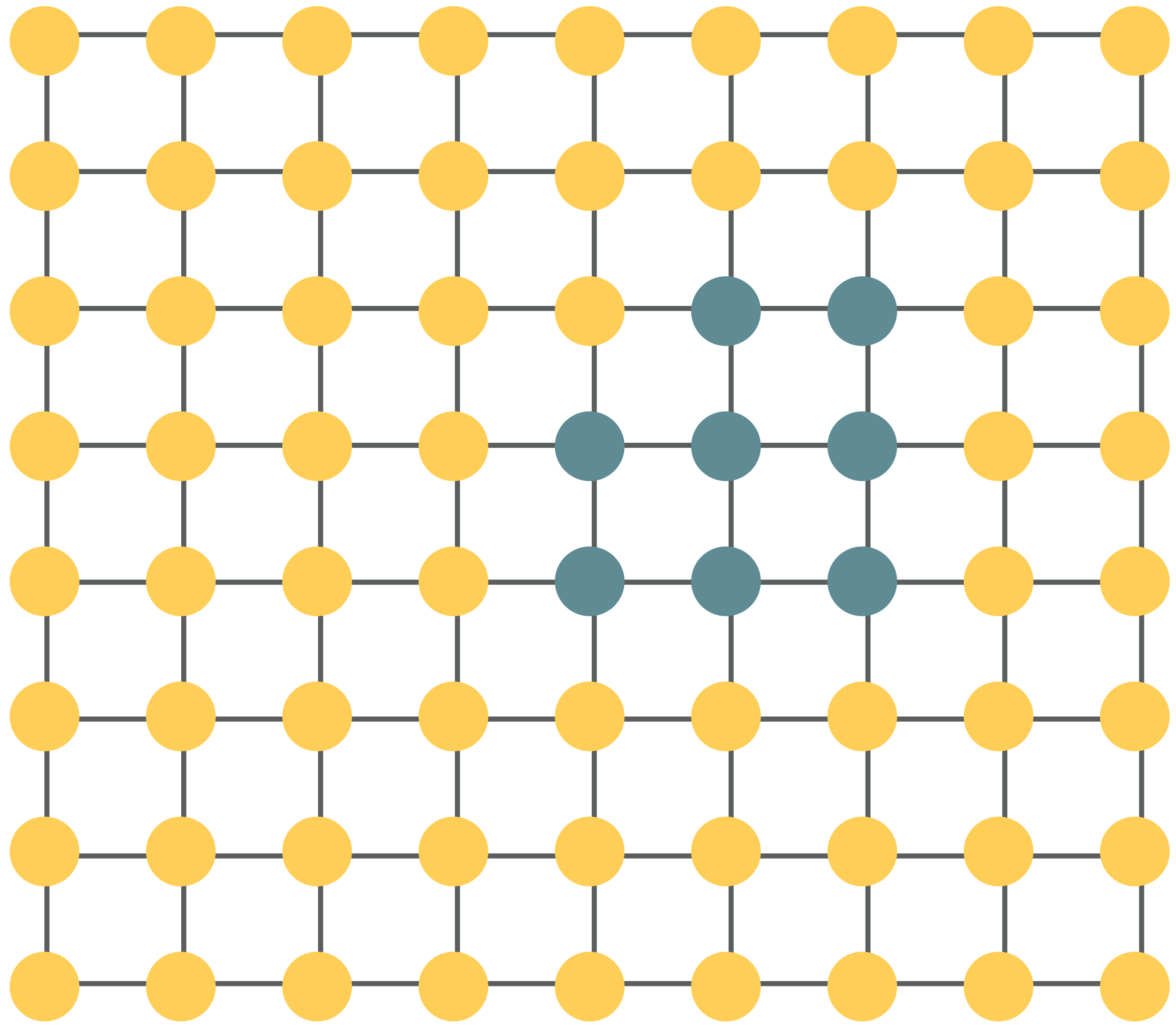
✓ Use *machine learning*



IMPLICIT SHAPE REPRESENTATIONS

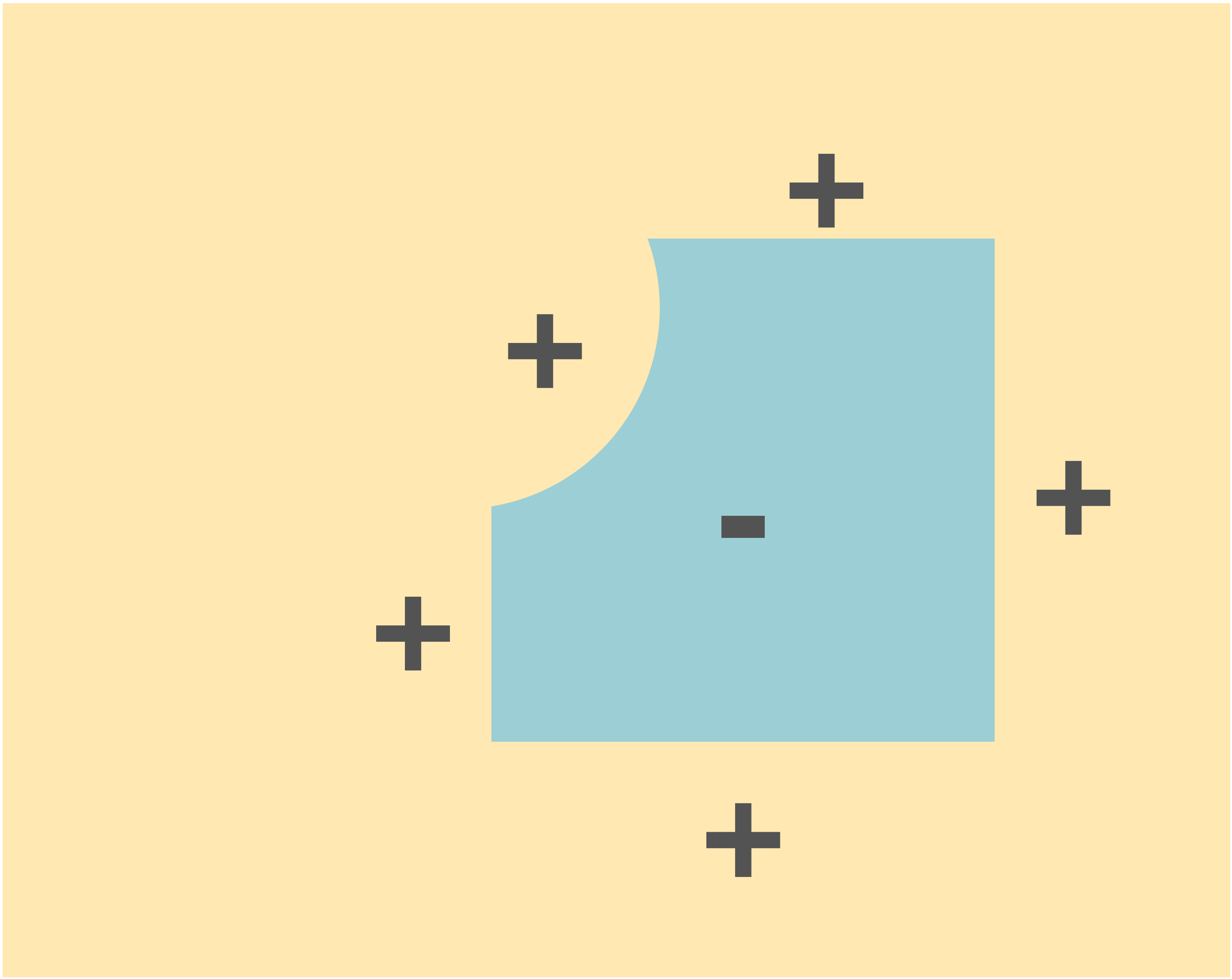
- ✓ Use *machine learning*
- ✓ Easy boolean operations

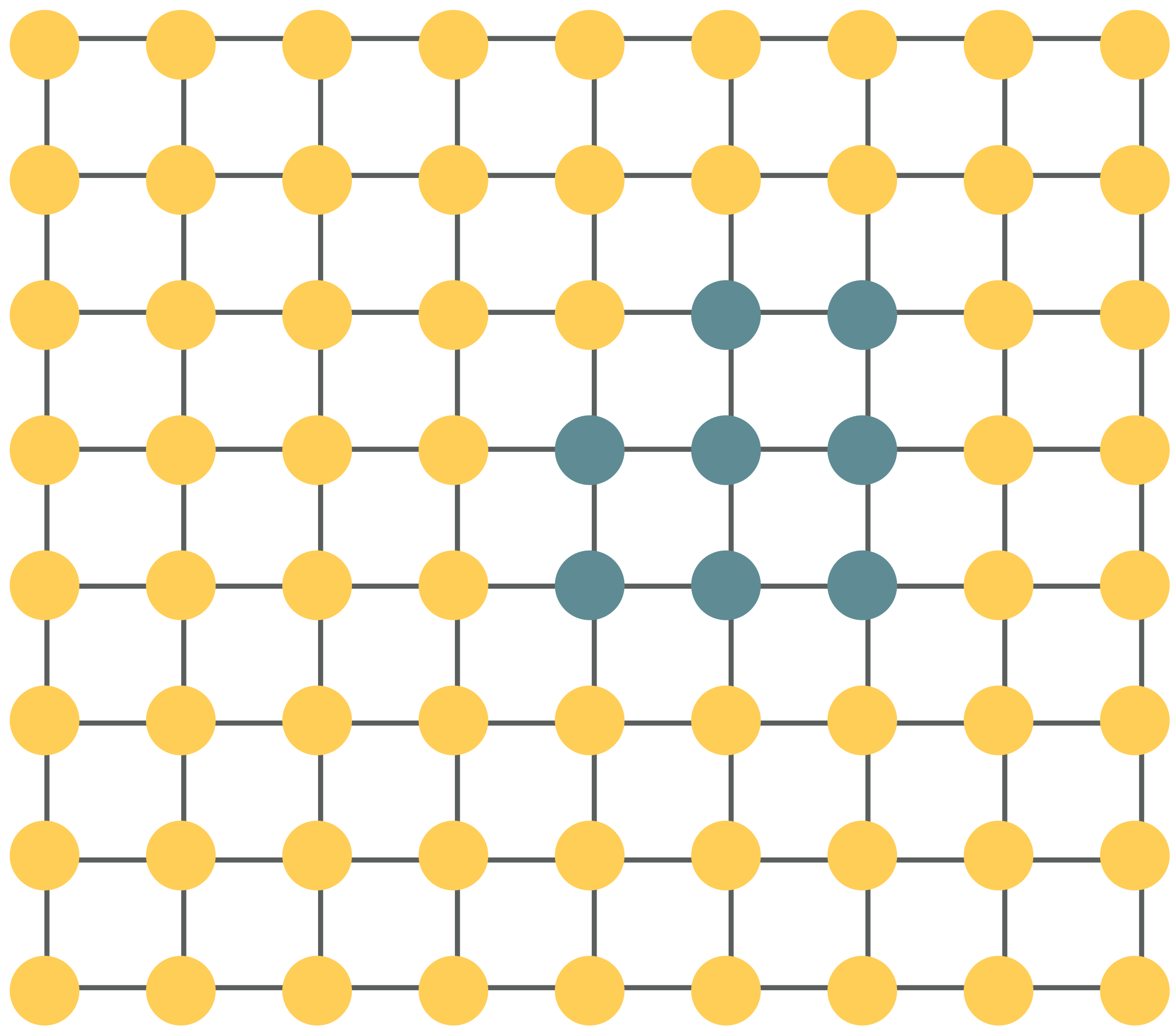




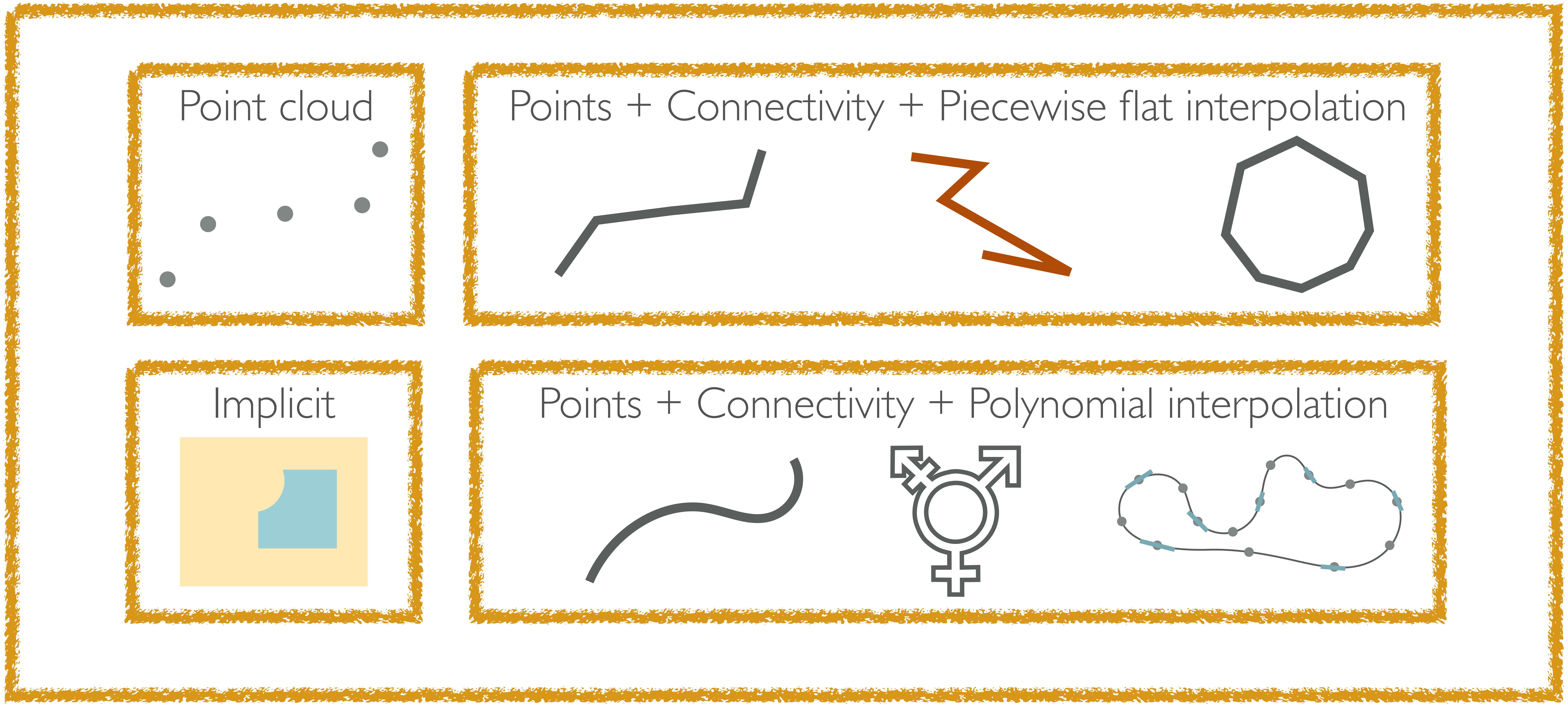
IMPLICIT SHAPE REPRESENTATIONS

- ✓ Use *machine learning*
- ✓ Easy boolean operations
- ✗ …almost everything else

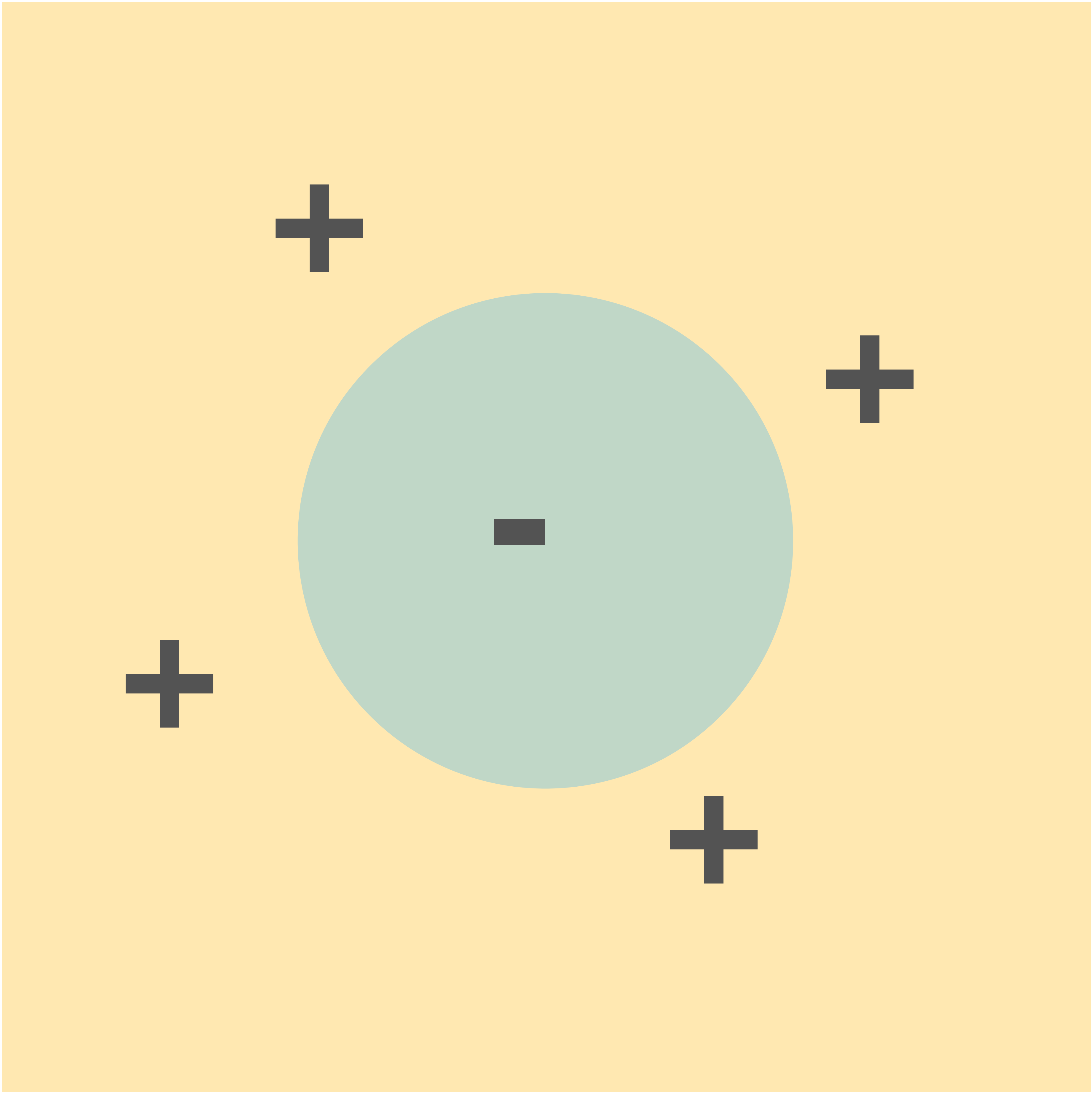




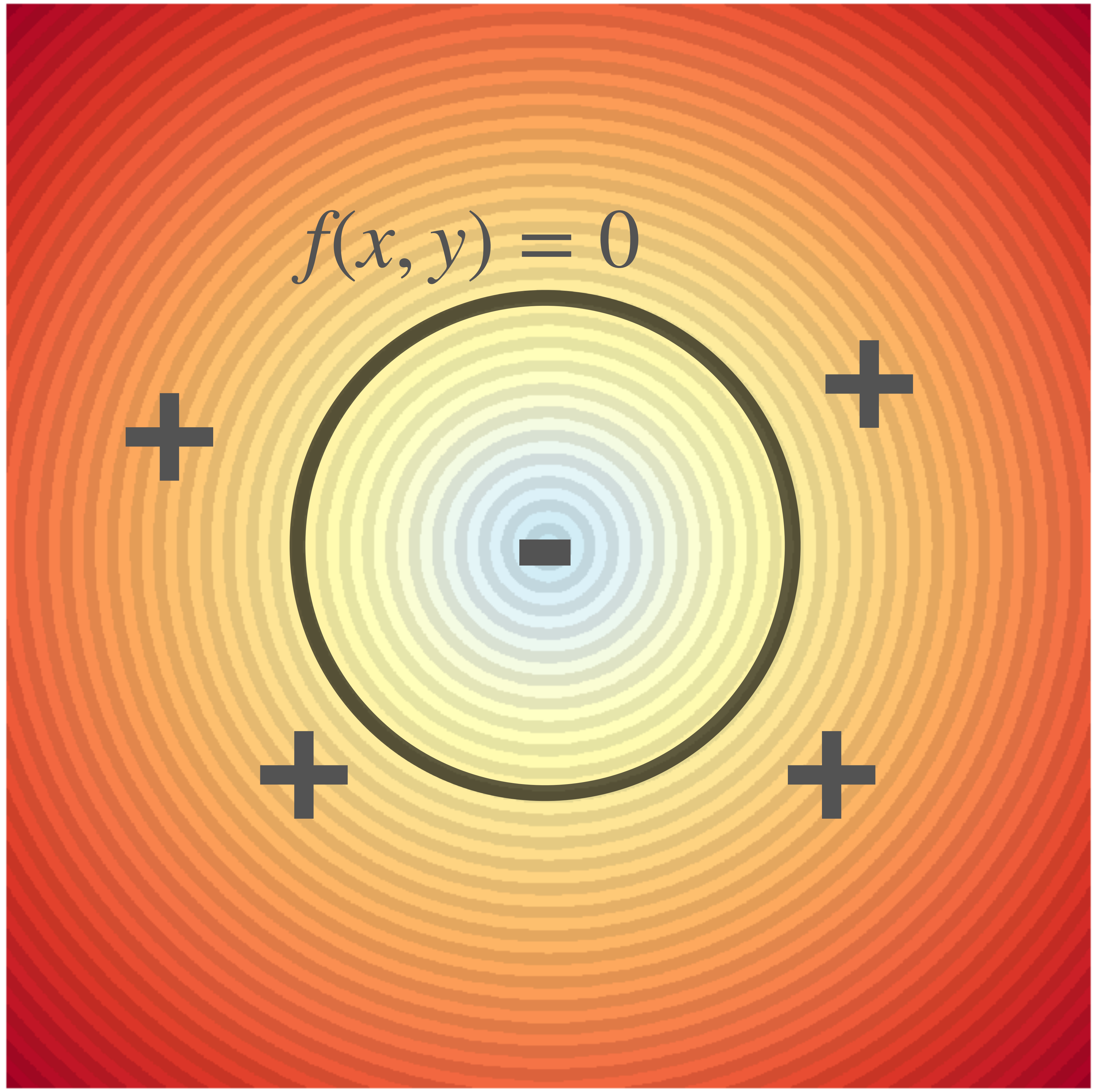
SHAPE REPRESENTATIONS



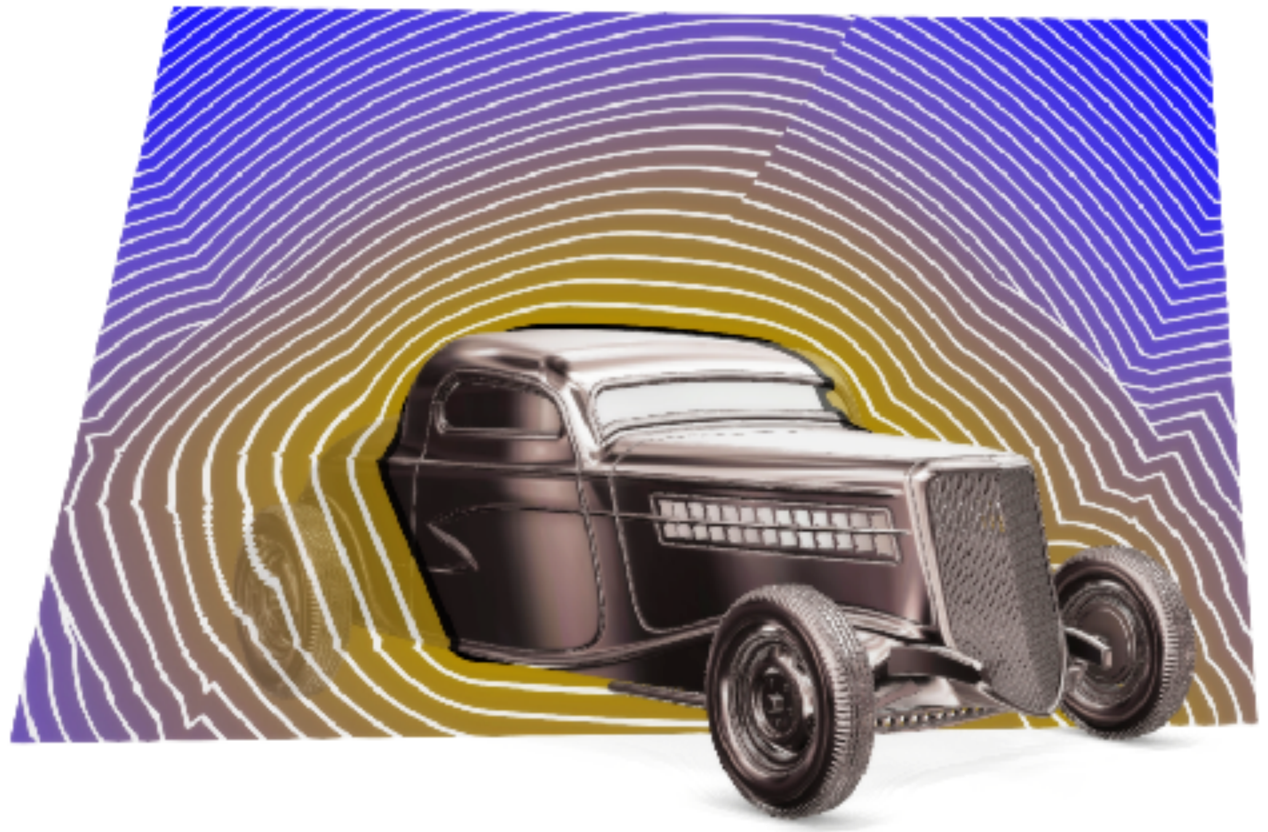
Implicit function



Signed distance function

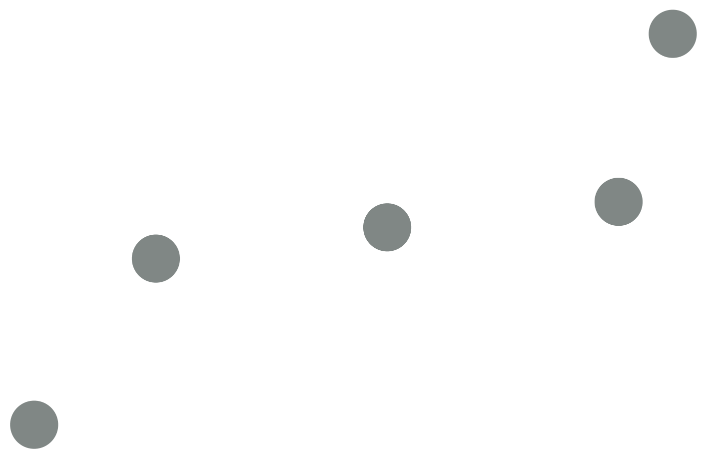


Later today…

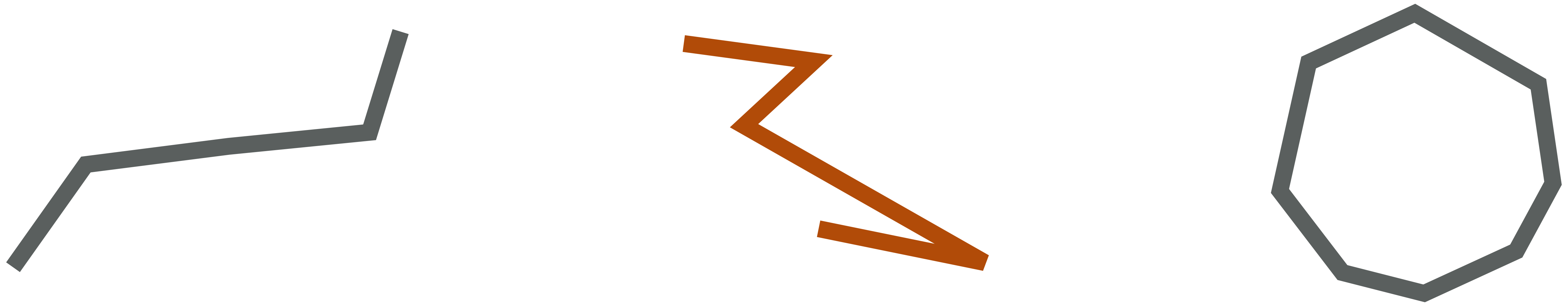


SHAPE REPRESENTATIONS IN 2D

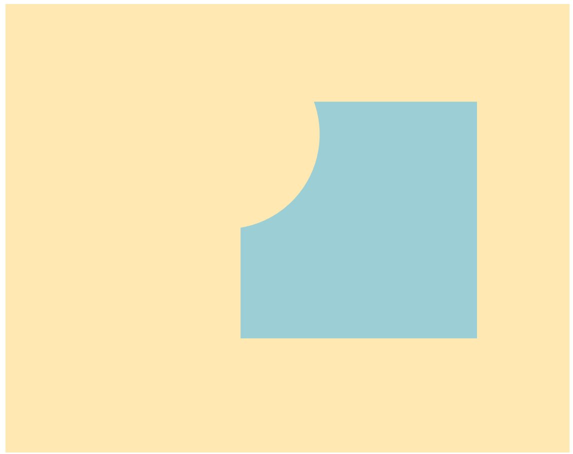
Point cloud



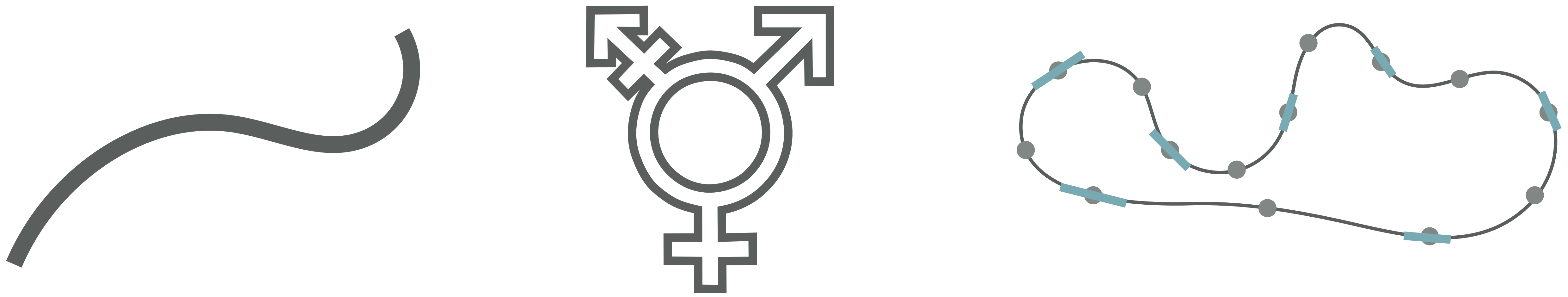
Points + Connectivity + Piecewise flat interpolation



Implicit



Points + Connectivity + Polynomial interpolation



WHAT'S THE PLAN NOW?

WHAT'S THE PLAN NOW?

- 201_polylines
- 202_wrappers
- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_mesh_booleans
- 207_implicit_surfaces

Same repo as yesterday

WHAT'S THE PLAN NOW?

- 201_polylines
- 202_wrappers

Now

- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_mesh_booleans
- 207_implicit_surfaces

Later in the day

WHAT'S THE PLAN NOW?

- 201_polylines
- 202_wrappers ← The boring one
- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_mesh_booleans
- 207_implicit_surfaces

WHAT'S THE PLAN NOW?

- 201_polylines
- 202_wrappers
- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_mesh_booleans
- 207_implicit_surfaces

The boring one

Requires installing stuff

WHAT'S THE PLAN NOW?

- 201_polylines
- 202_wrappers
- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_mesh_booleans
- 207_implicit_surfaces

The boring one

Requires installing stuff

For reference, no need to do it today

WHAT'S THE PLAN NOW?

- 201_polylines ← The interesting one
- 202_wrappers
- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_mesh_booleans
- 207_implicit_surfaces

WHAT'S THE PLAN NOW?

201_polylines

202_wrappers

203_basic_mesh_modeling

204_timing

205_mex

206_mesh_booleans

207_implicit_surfaces

The interesting one

It's meant to be hard!

```
>> solve_inplane(v,
Error using error
```

WHAT TO DO V … HARD?

silviasellan 🏳️‍🌈 09:16
ok now
just between us
did you know that “Loop subdivision” was named after a Charles Loop, and not just a loop

odedstein 10:33
Yes of course

---

silviasellan 🏳️‍🌈 16:39
help

odedstein 16:40
In general, or you need help now?

silviasellan 🏳️‍🌈 16:40
now, if youre not busy

---

silviasellan 🏳️‍🌈 12:28
want to proofread?

odedstein 12:28
Your asia submission?
sure

---

silviasellan 🏳️‍🌈 18:36

> silviasellan
> so
> Direct Message | Jan 12th, 2020 | View conversatio

> silviasellan
> much
> Direct Message | Jan 12th, 2020 | View conversatio

> silviasellan
> stress
> Direct Message | Jan 12th, 2020 | View conversation

odedstein 18:36
Do you want me to get you something

---

silviasellan 🏳️‍🌈 17:04
WHY IS MY BLENDER NOT WORKING

silviasellan 🏳️‍🌈 17:11
lincoln.png ▾

why are my shadows so ugly
they shouldnt be so sharp with environment lighting

odedstein 17:21
Is it too bright?
Is your material overtly shiny?

May 9th, 2020 ⌄

silviasellan 🏳️‍🌈 11:28
can I ask for your advice on something?

odedstein 11:28
Sure

---

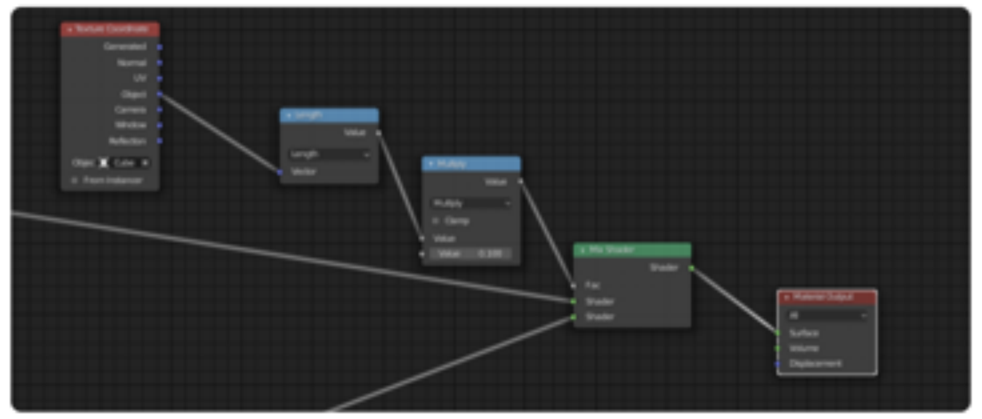
silviasellan 🏳️‍🌈 11:07
i'd love to know your thoughts
Keynote Document ▾
shape-representations.k
88 MB Keynote Document

odedstein 11:07
will look at it in the evening

---

silviasellan 🏳️‍🌈 09:35
ok so I am gonna ask a crazy thing
say I have two meshes
meshA and meshB
I want to render meshB
with a shader mix
“at points of meshB that are at a distance higher than tol f
does that sound like something blendable?

odedstein 09:46
image.png ▾



---

silviasellan 🏳️‍🌈 11:06
I have a scene with, say, 20 objects
and I want to render 15 of them with wireframe, but 5 of them without it
is there an easy way of doing this?

odedstein 11:07
There is a wireframe node in the cycles sha
Like when you make a material
play around with that

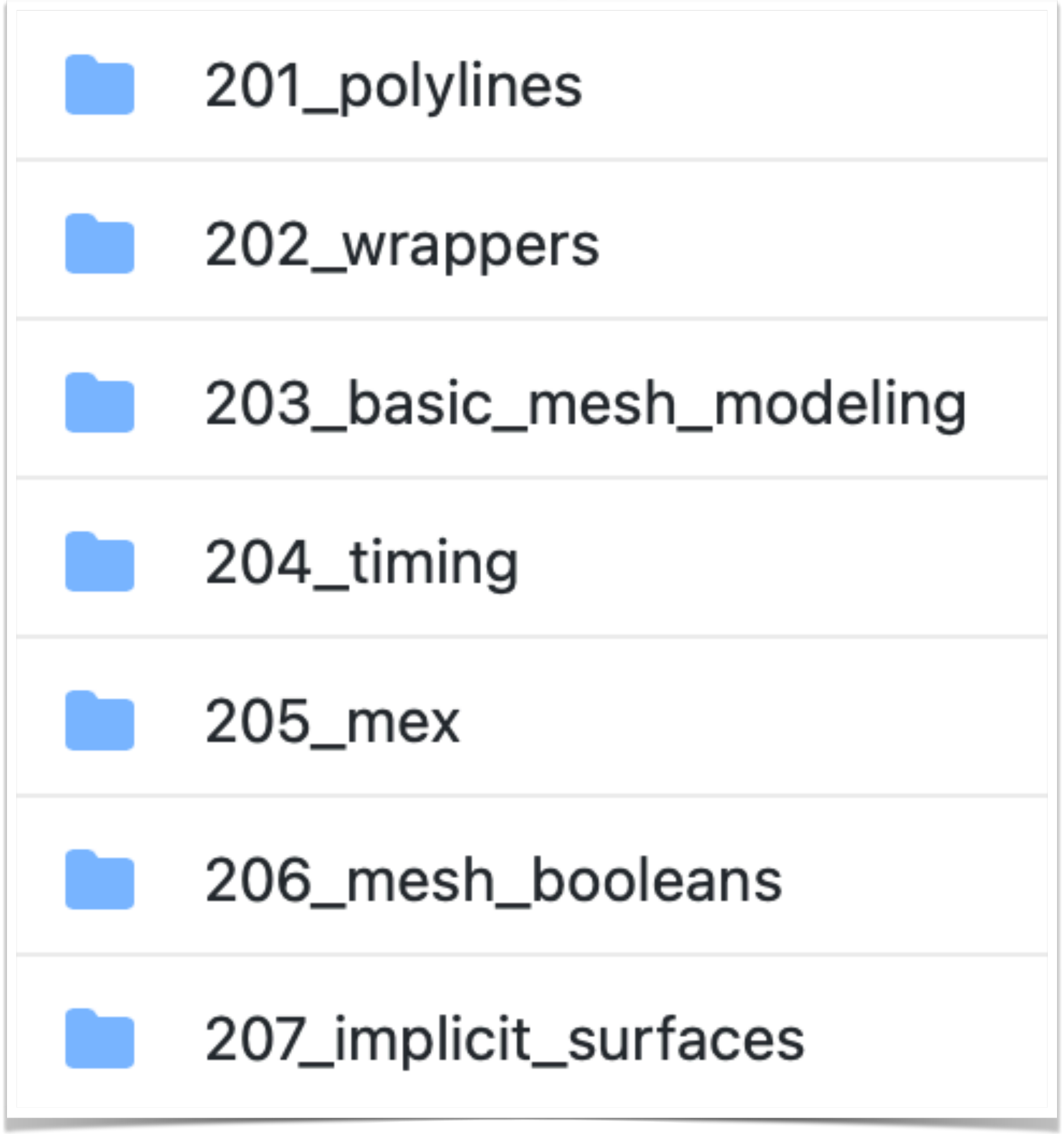
silviasellan 🏳️‍🌈 11:07
there you go
thank you oded

---

silviasellan 🏳️‍🌈 15:17
maybe you know how to solve this
if I have a .blend file
with like massive objs
and then I delete the objs from the scene
the .blend file size does not go down

odedstein 15:18
have you saved and quit?

GO!


201_polylines
202_wrappers
203_basic_mesh_modeling
204_timing
205_mex
206_mesh_booleans
207_implicit_surfaces

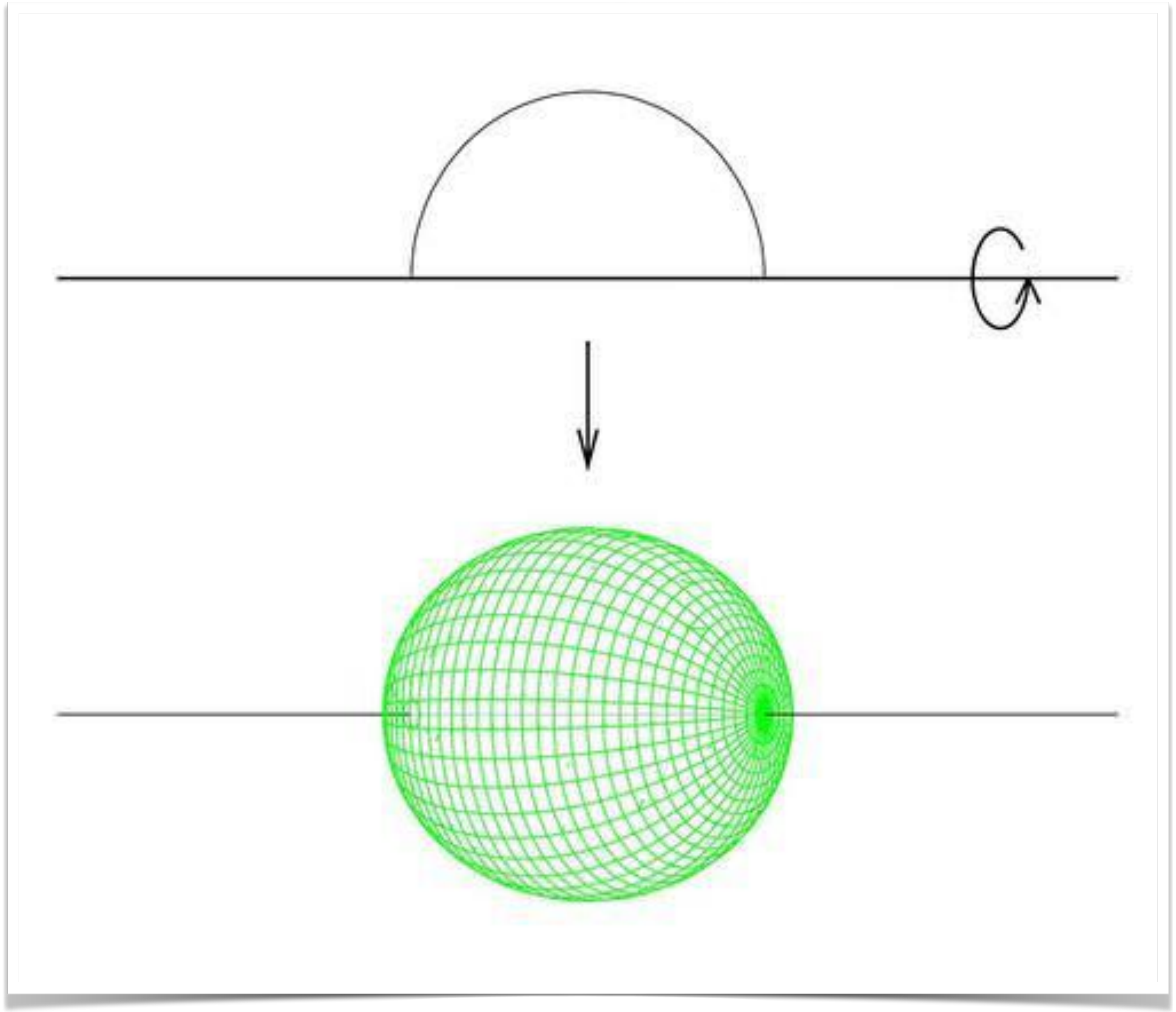
MESH MATH AND BEYOND

On creating, storing and using geometry

“I was so ahead of the curve, the curve became a sphere”

- Taylor Swift, *This is me trying*

Option 1: Surface of revolution



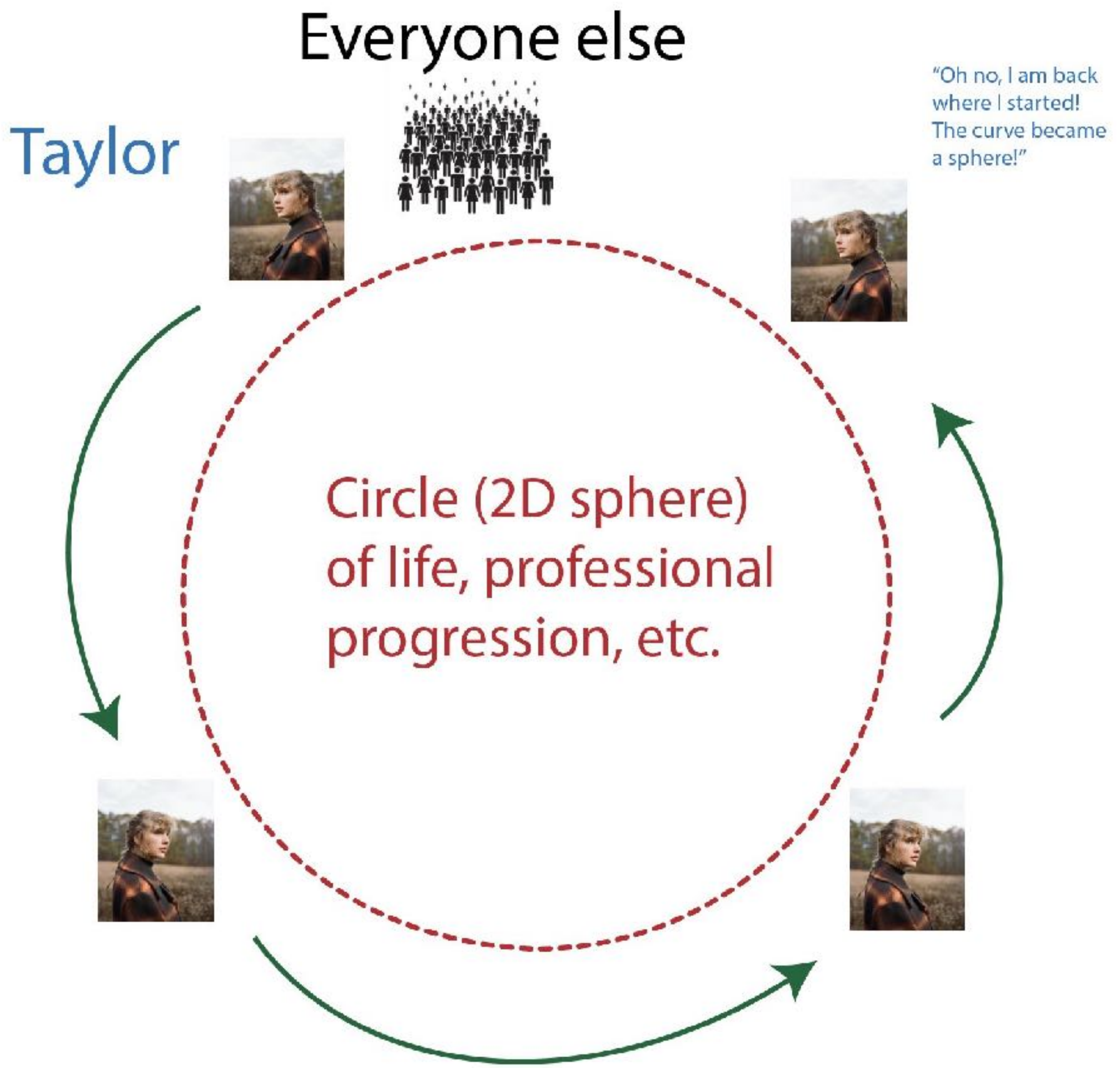

cleverlittleteapot · 2 yr. ago

I believe that she is saying was ahead of the curve. She was a mover and shaker. Remember when she was described as the music industry and 1989 was so highly regarded.

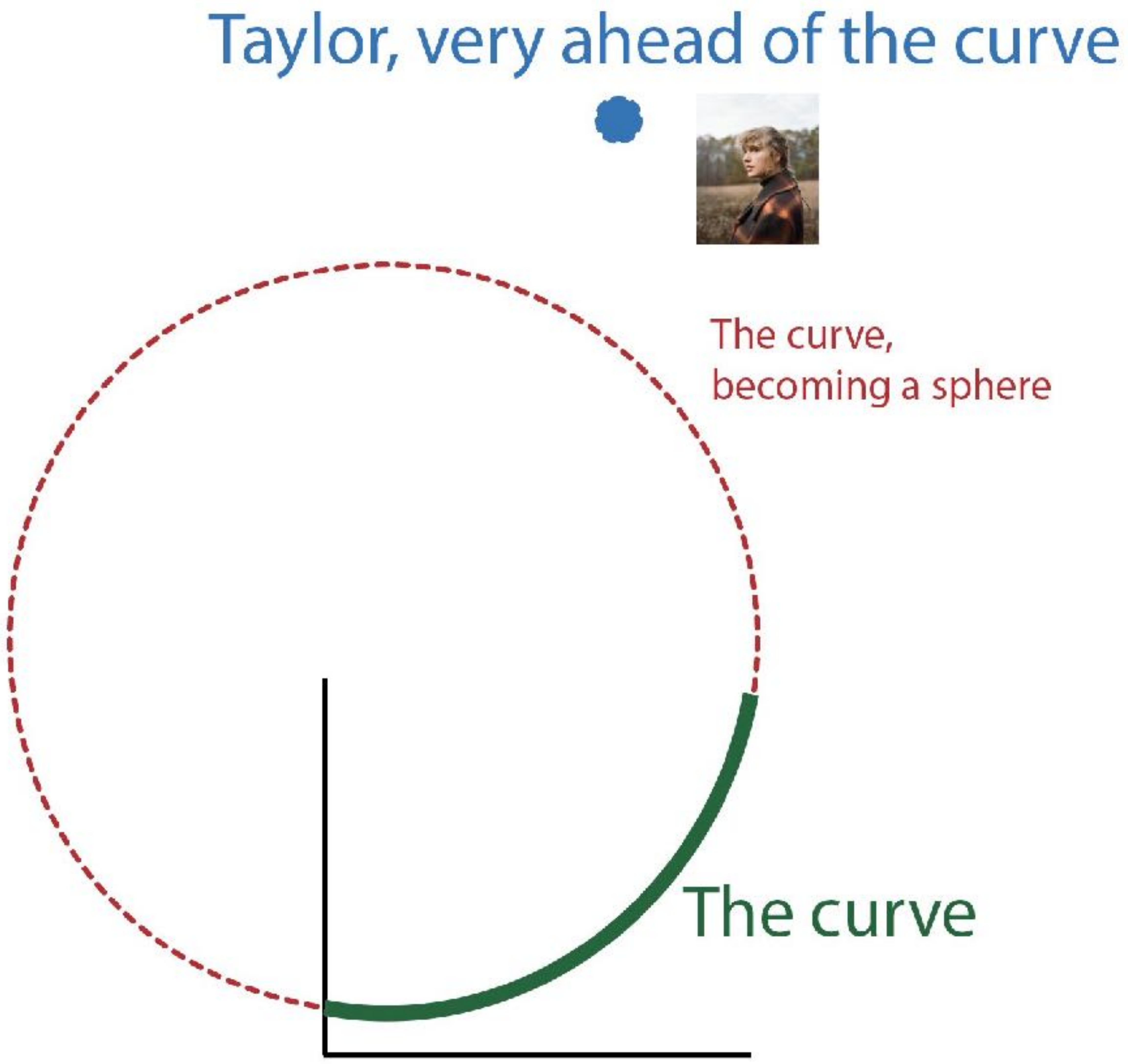
Things then changed. They stopped being 2 dimensional. A sphere is 3 dimensional. She is acknowledging that rather than continuing to be the mover and shaker, she was unable to keep up. The entire world changed, and left her behind.

This is her trying. This is her new avenue. Experimentation is new genres. Stepping away from the pop which has not served her well in the last two eras. No Grammy etc.

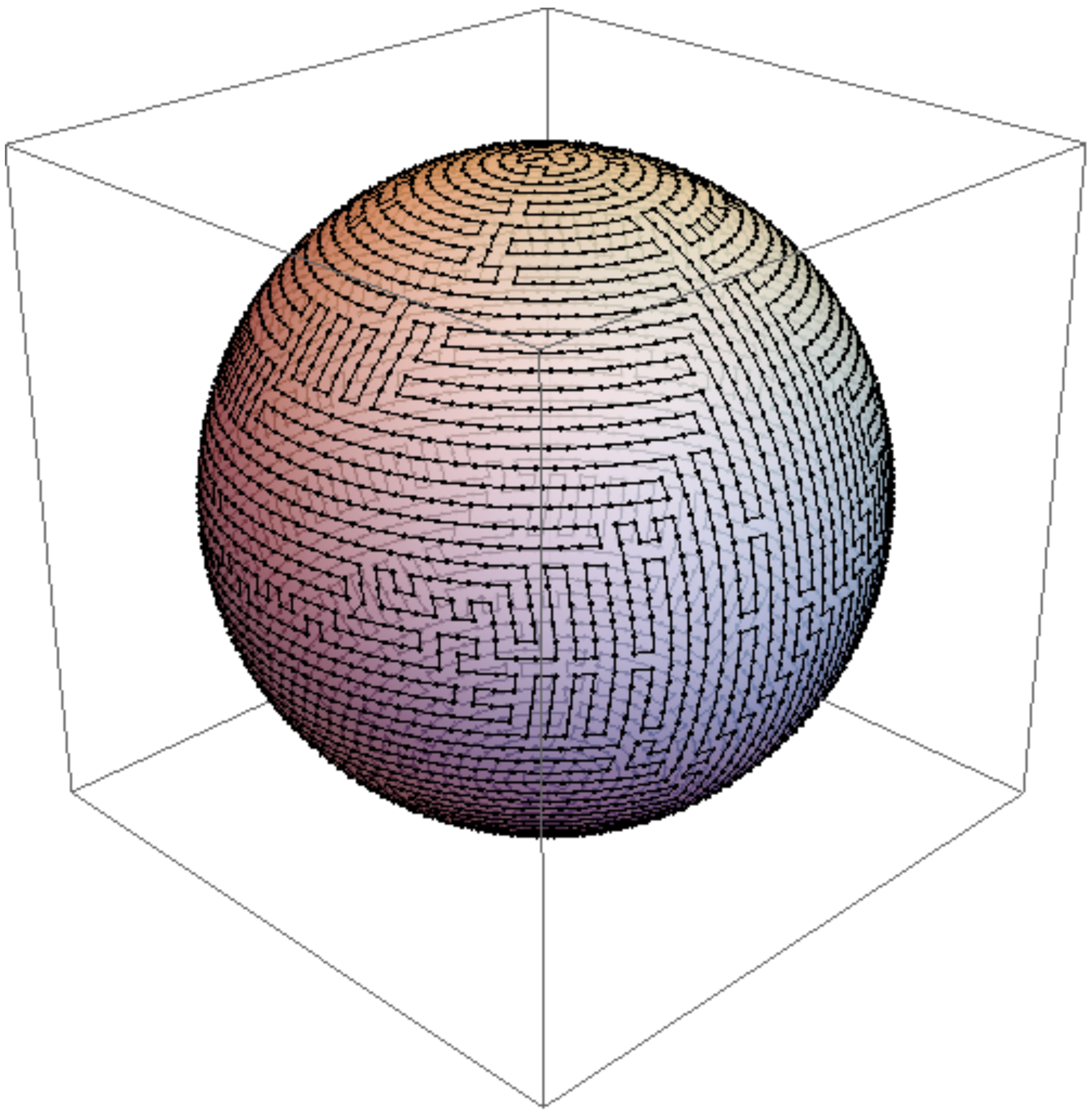
Option 2: "The cycle"



Option 3: "The graph"

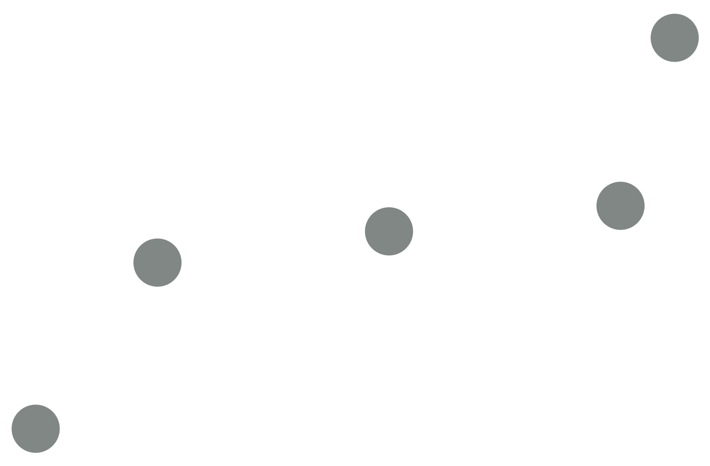


Option 4: "The space filling curve"

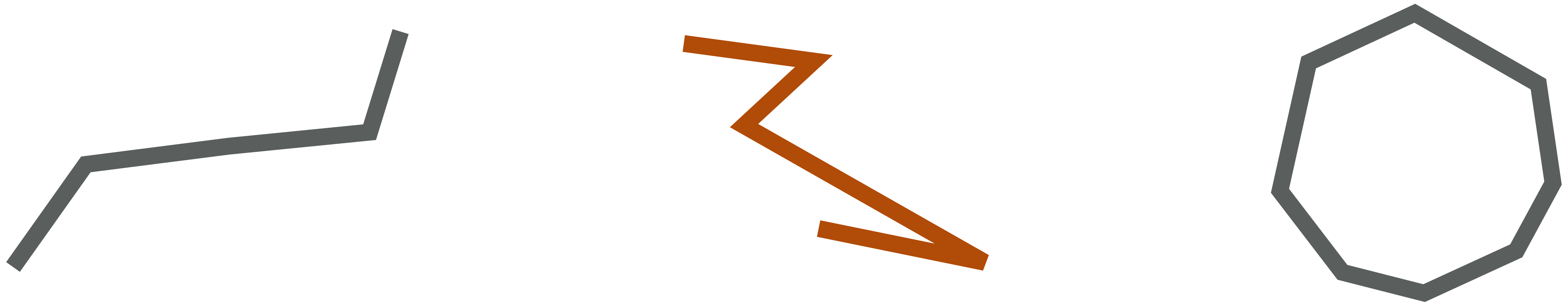


SHAPE REPRESENTATIONS IN 2D

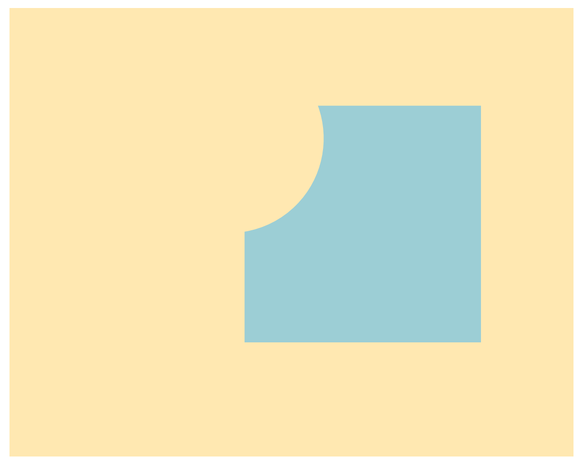
Point cloud



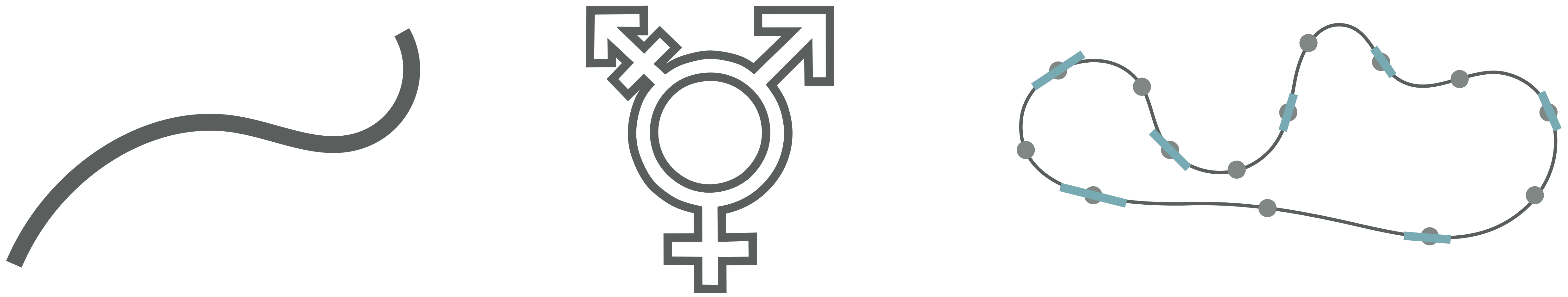
Points + Connectivity + Piecewise flat interpolation



Implicit



Points + Connectivity + Polynomial interpolation



SHAPE REPRESENTATIONS IN 3D

Point cloud

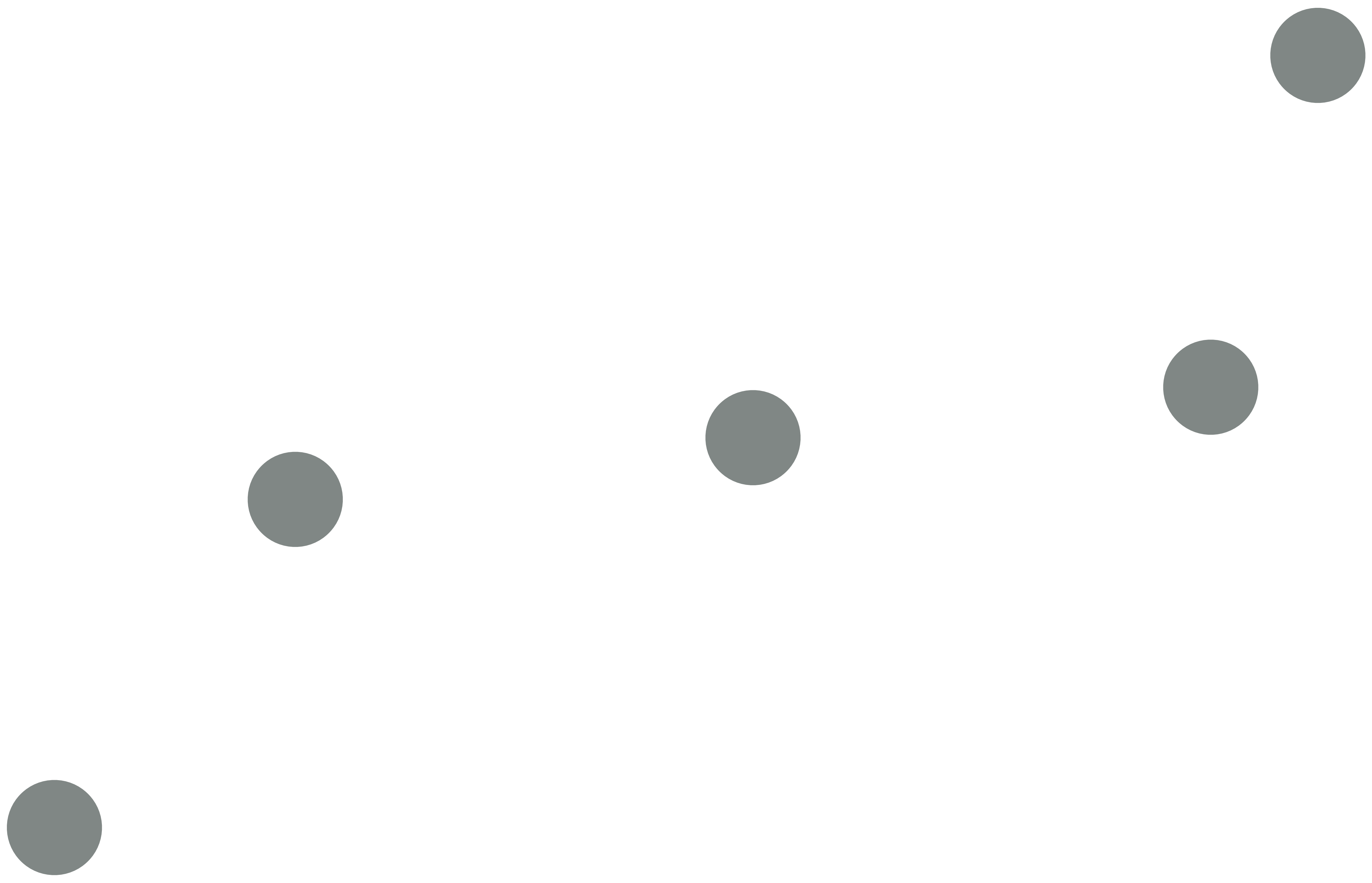
Points + Connectivity + Piecewise flat interpolation

Implicit

Points + Connectivity + Polynomial interpolation

Point cloud

Point cloud



Connectivity?

Point cloud

Connectivity?

Point cloud

Why use point clouds at all?

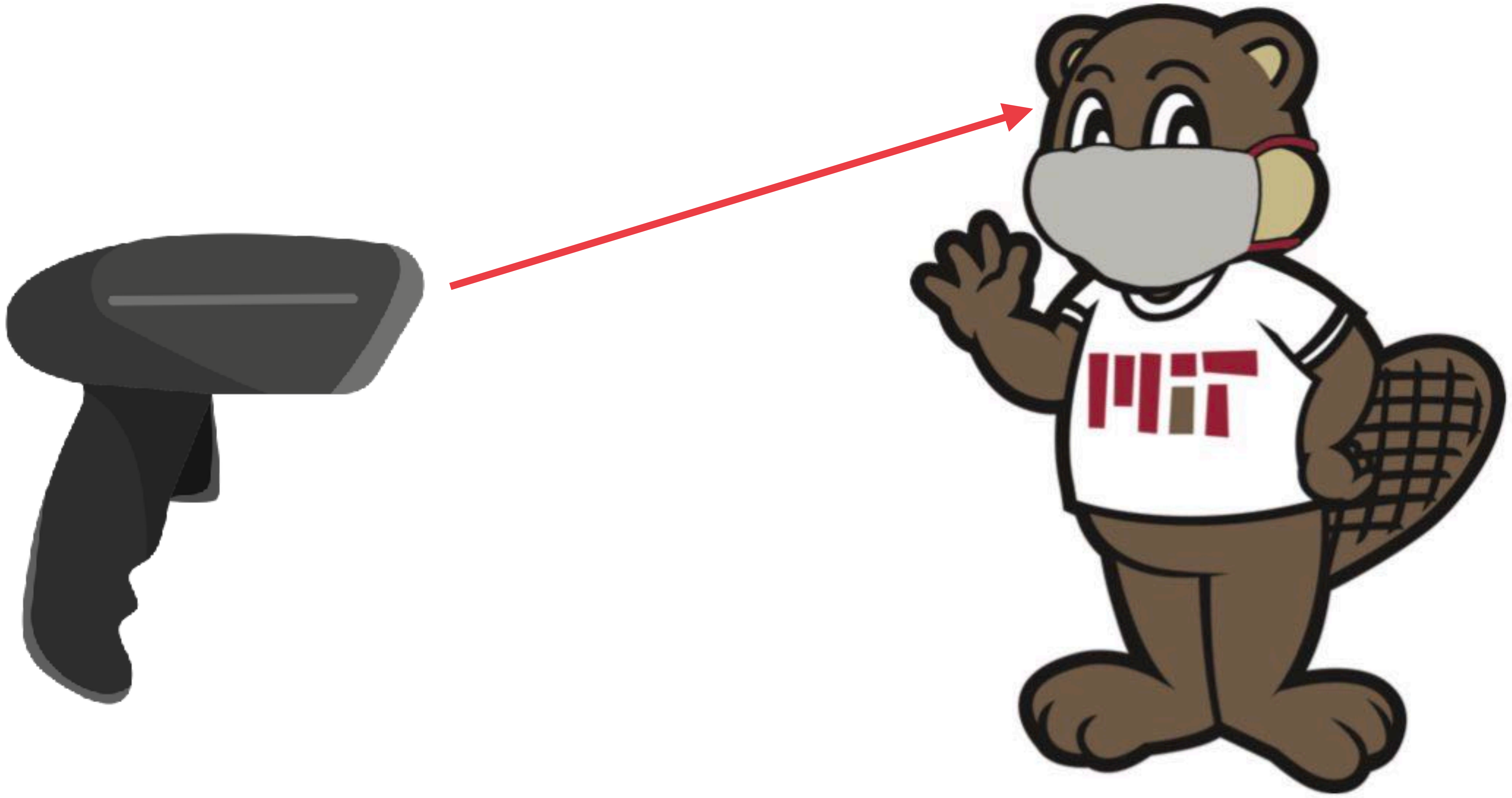
Point cloud

3D scanning

Point cloud

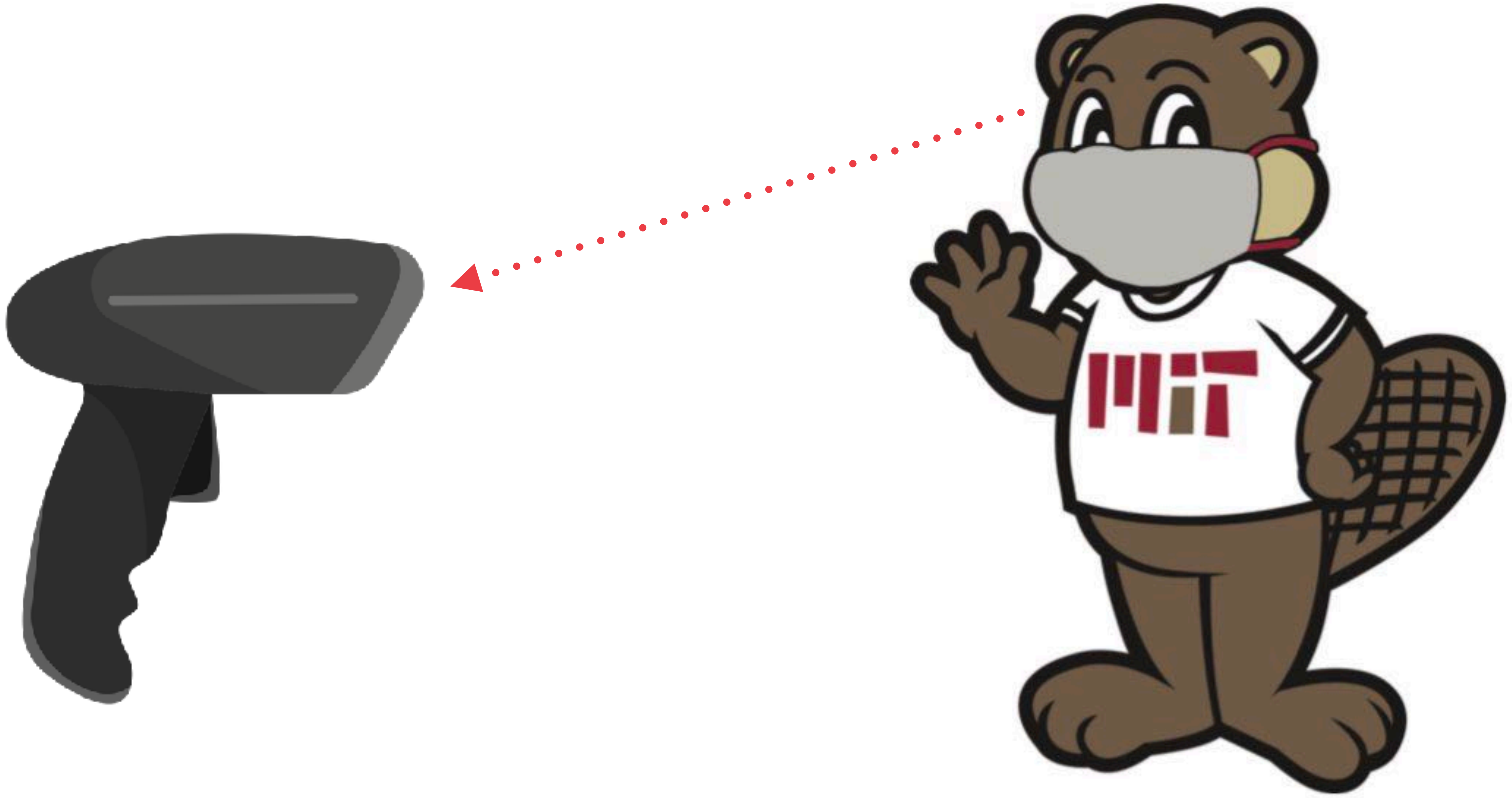
3D scanning

Point cloud



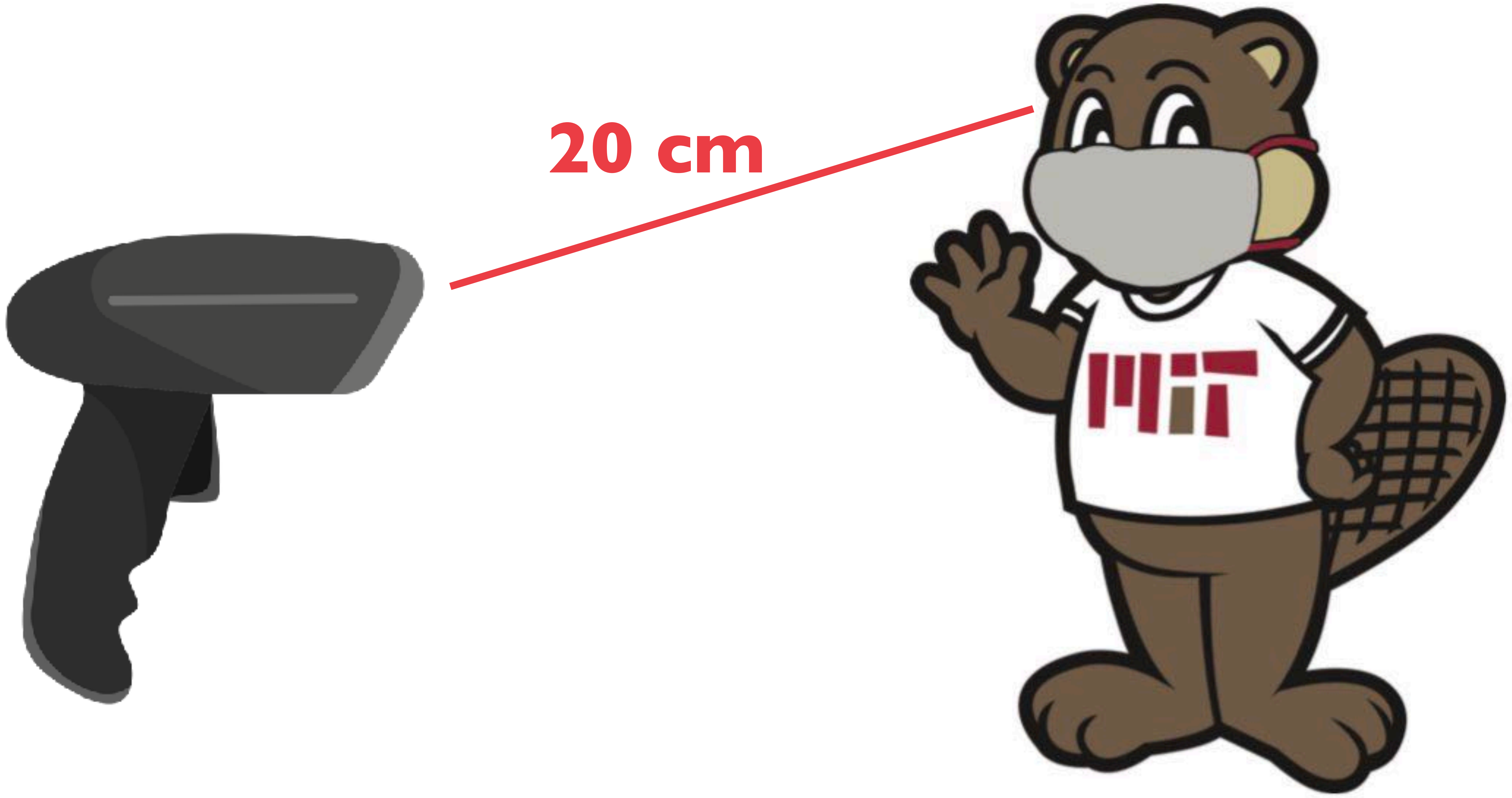
3D scanning

Point cloud



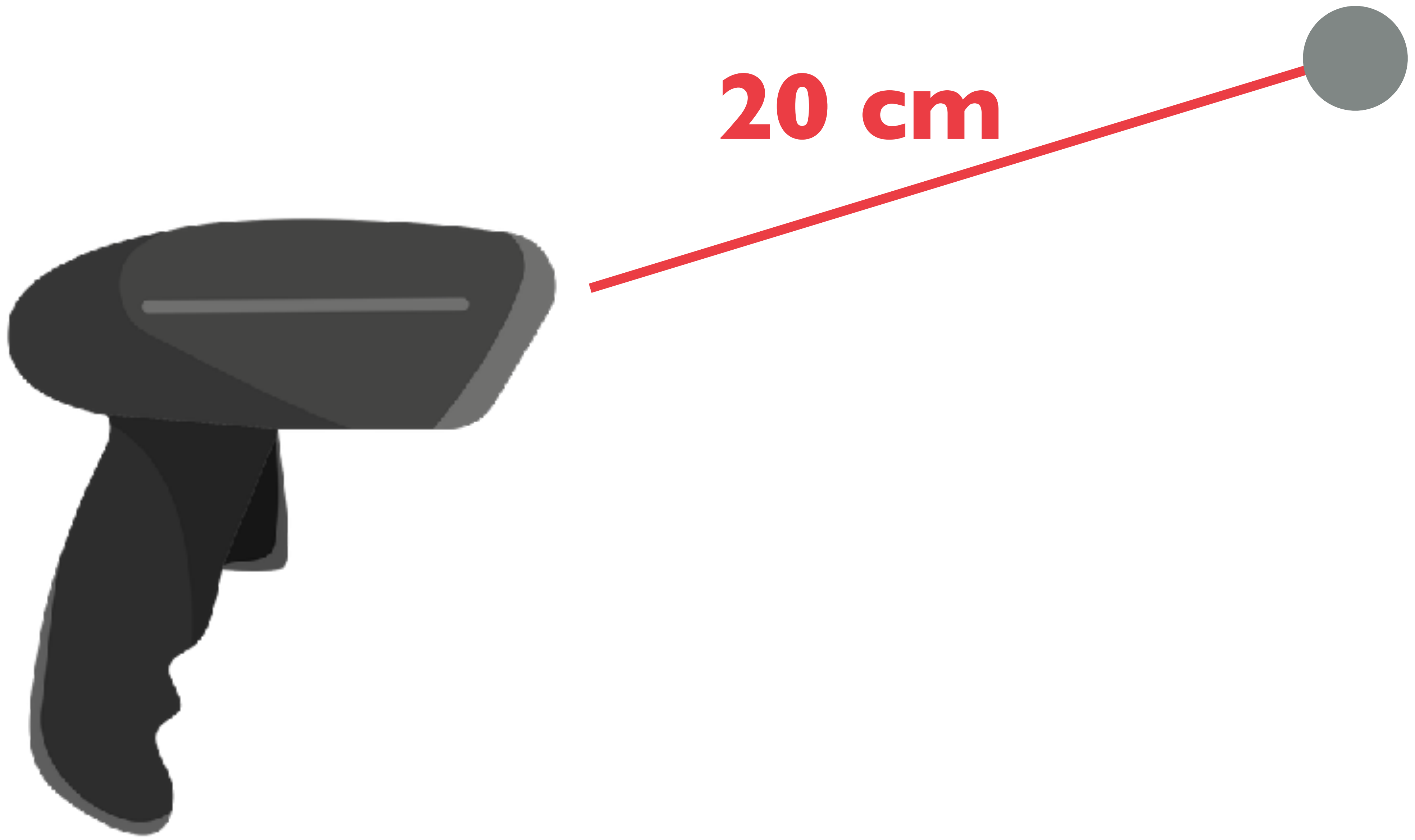
3D scanning

Point cloud



3D scanning

Point cloud

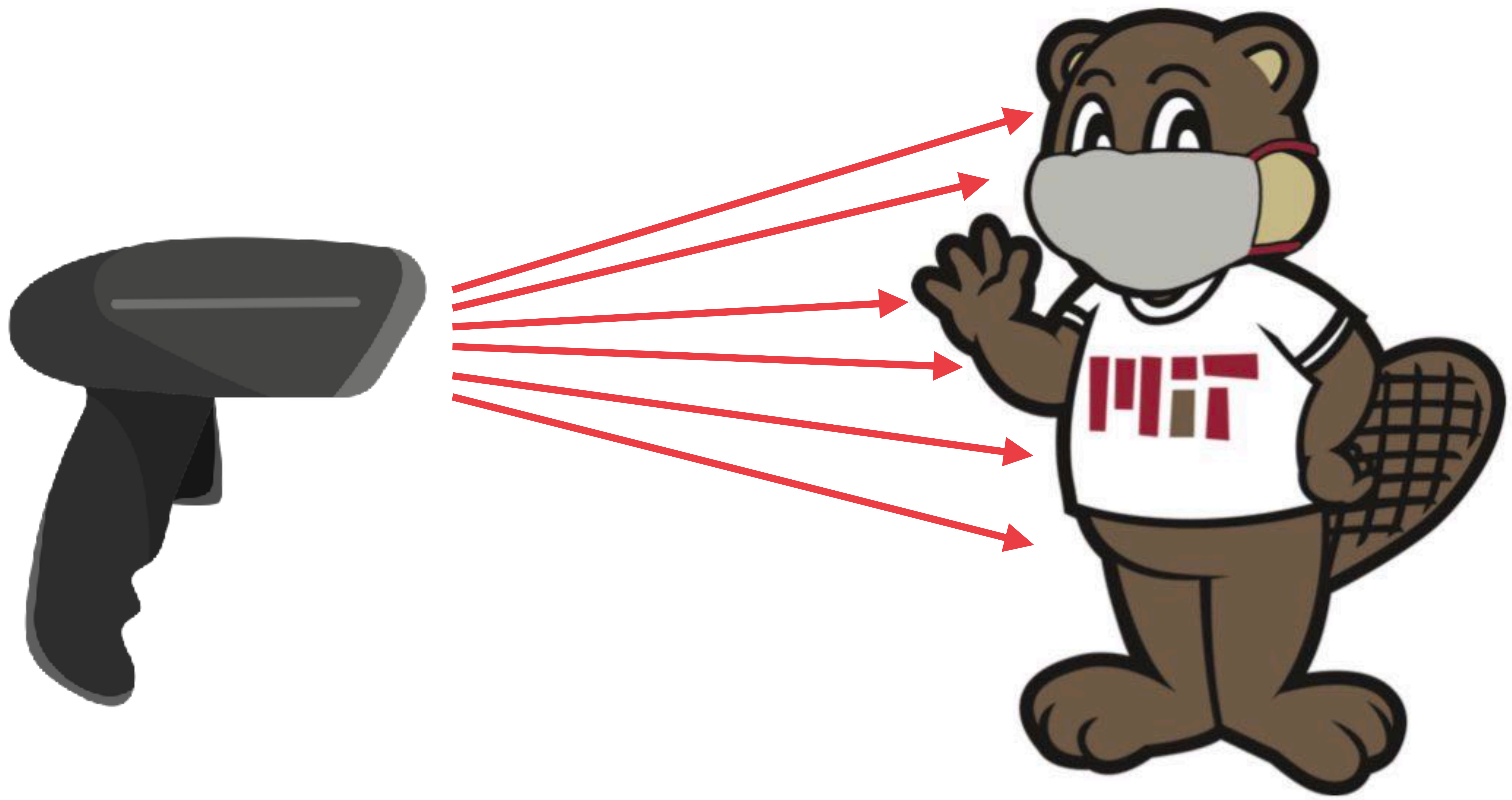


3D scanning

Point cloud

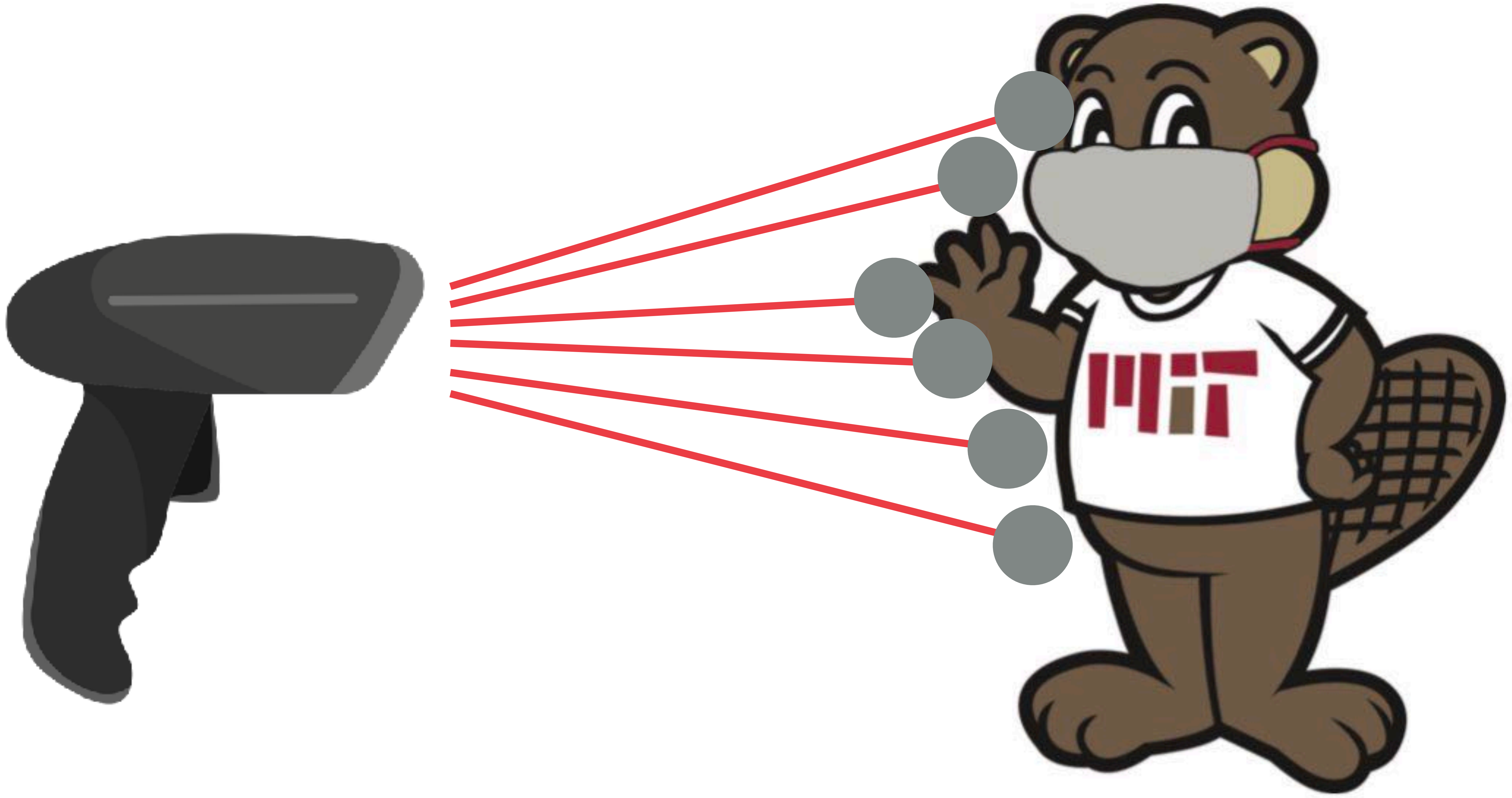
3D scanning

Point cloud



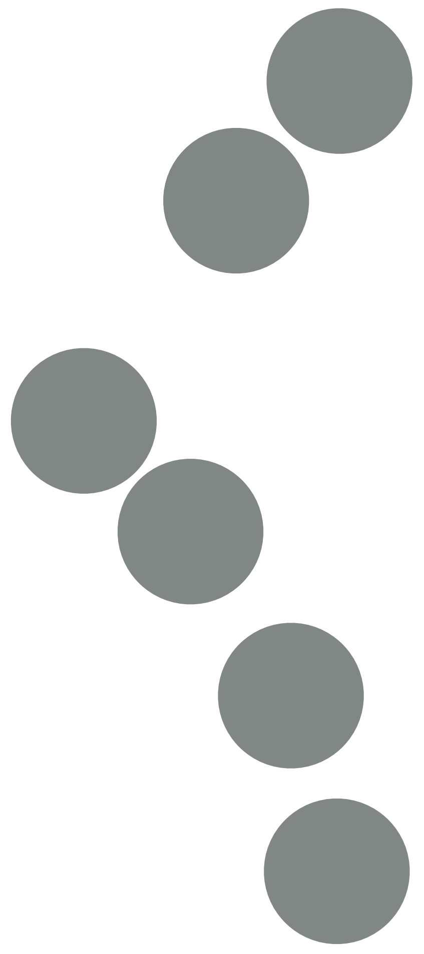
3D scanning

Point cloud



3D scanning

Point cloud



An autonomous car only sees point clouds

Your phone only sees point clouds

Your phone only sees point clouds

Art scanners only see point clouds

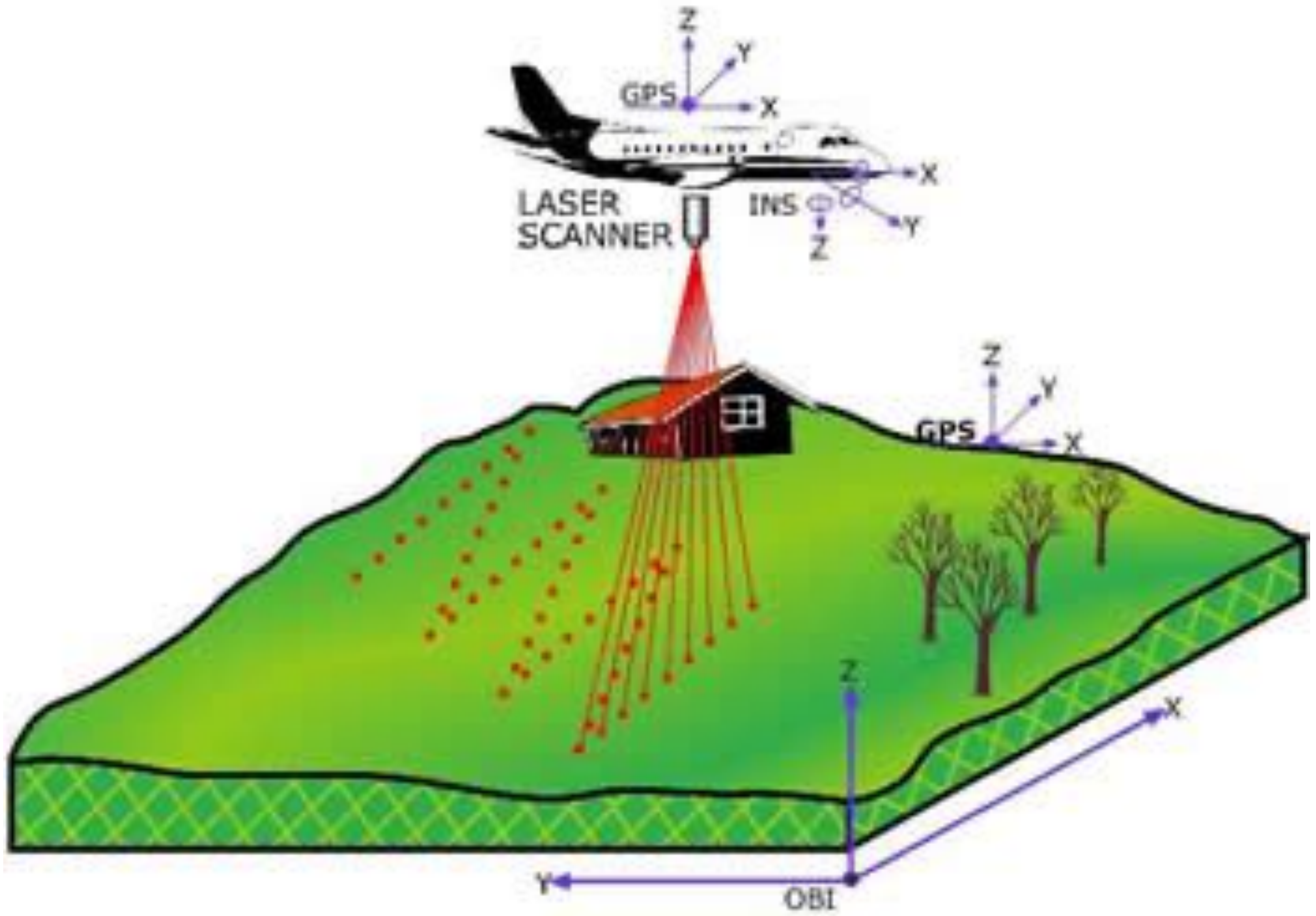
Art scanners only see point clouds

Art scanners only see point clouds

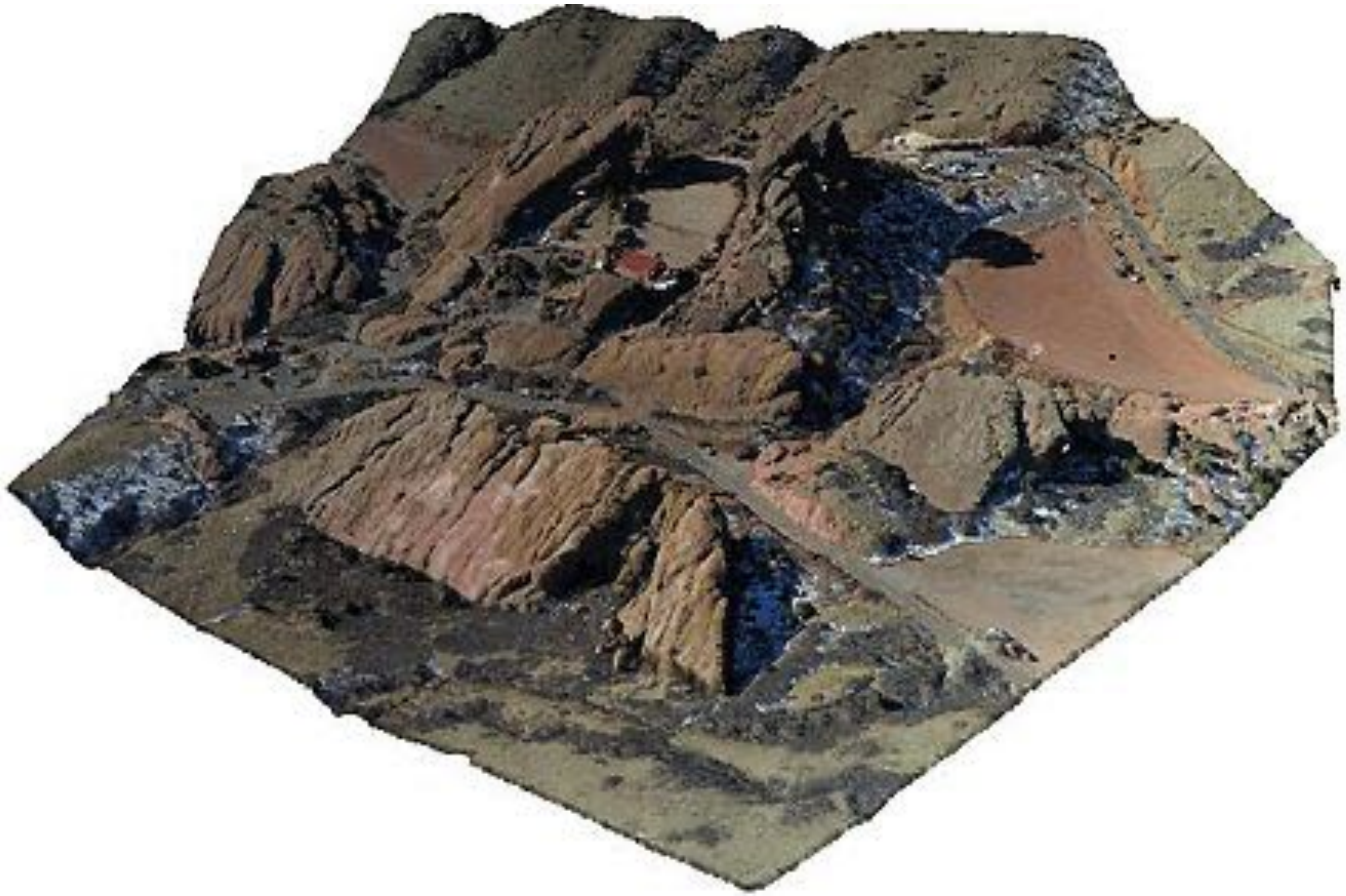
Welcome to the 3D Scanning Frontier

The 3D Program is a small group of technologists working within the Smithsonian Institution Digitization Program Office. We focus on developing solutions to further the Smithsonian's mission of "the increase and diffusion of knowledge" through the use of three-dimensional capture technology, analysis tools, and our distribution platform.

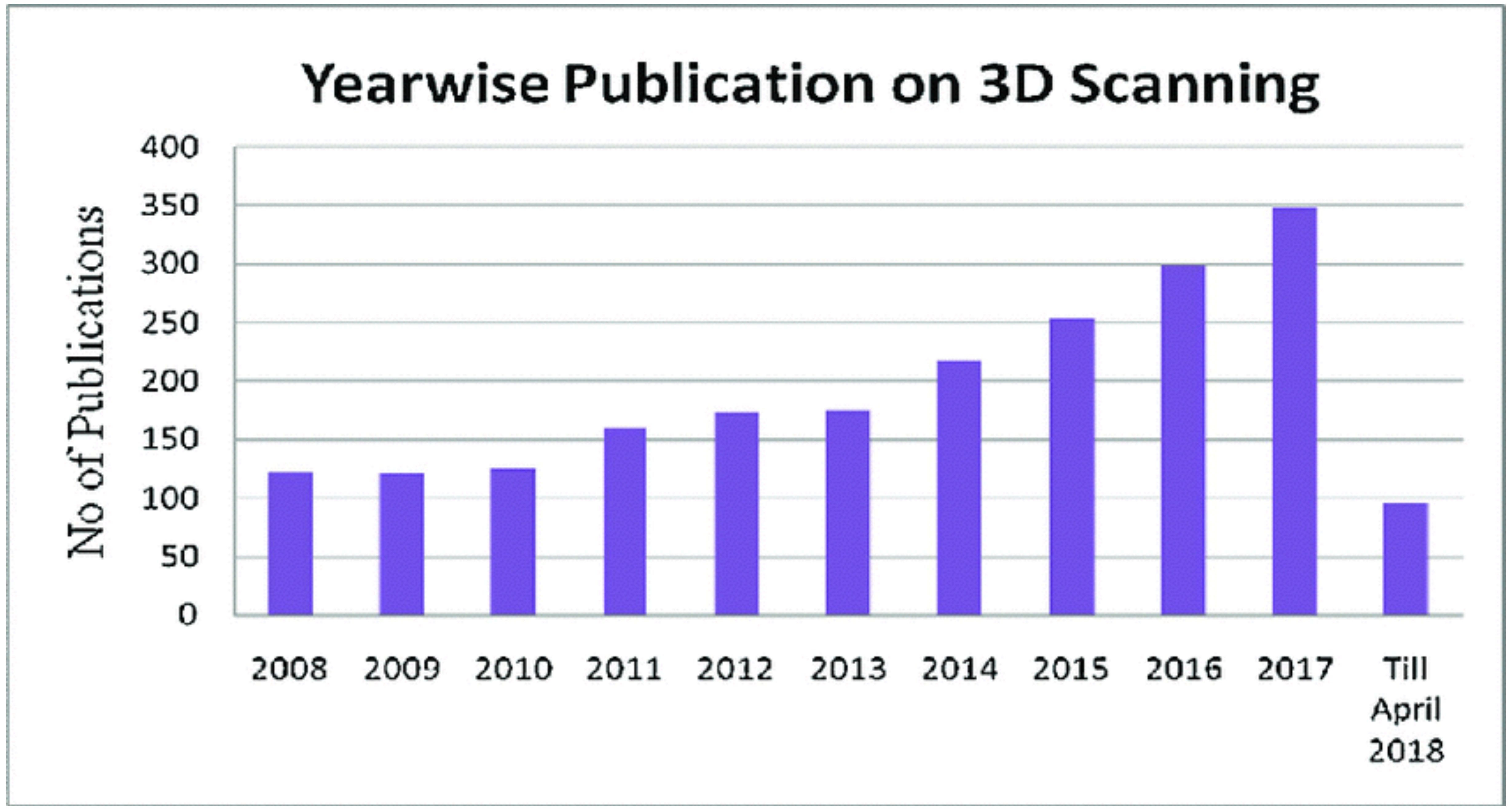
Ground surveyors only see point clouds



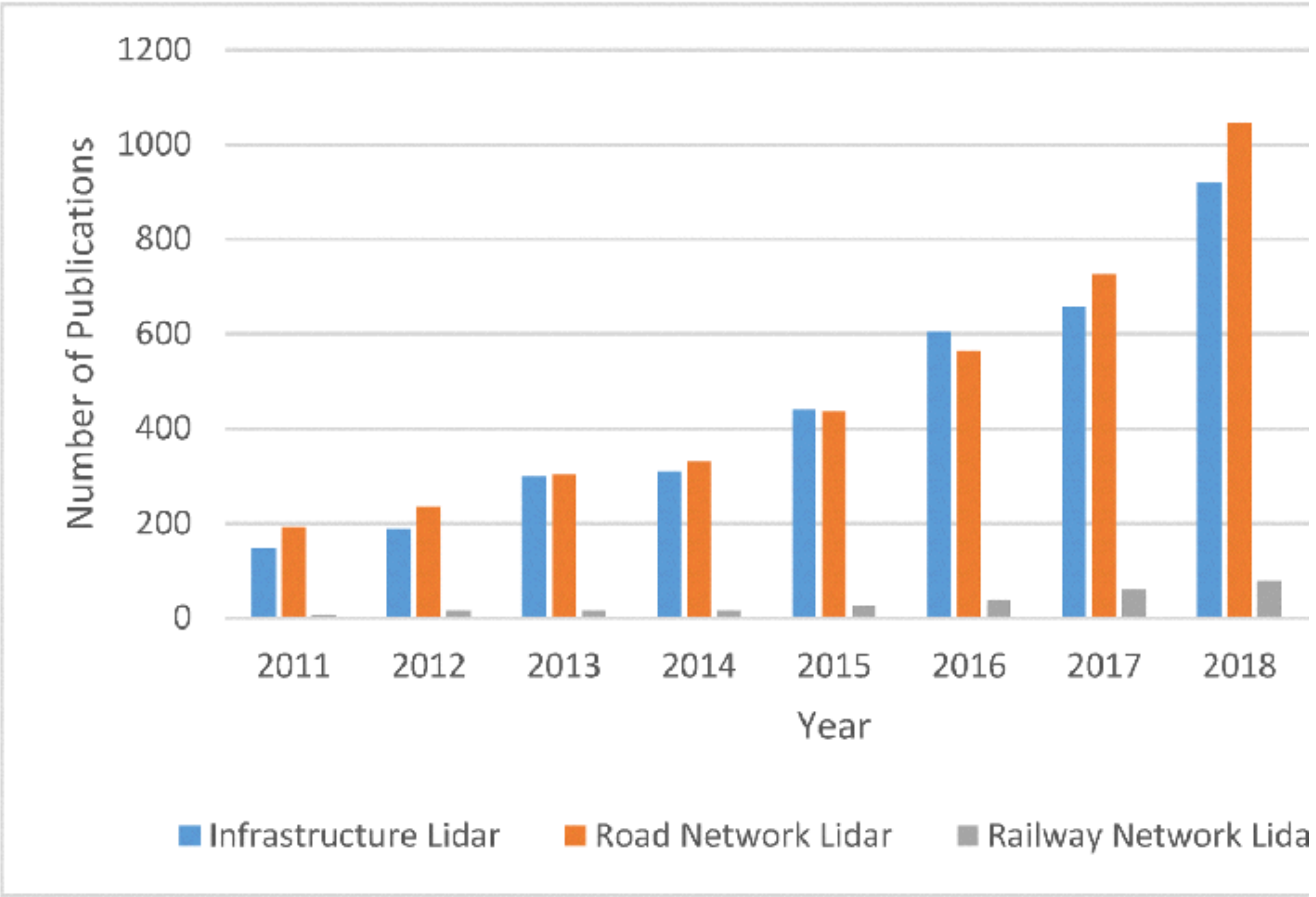
Ground surveyors only see point clouds



3D scanning has become increasingly popular



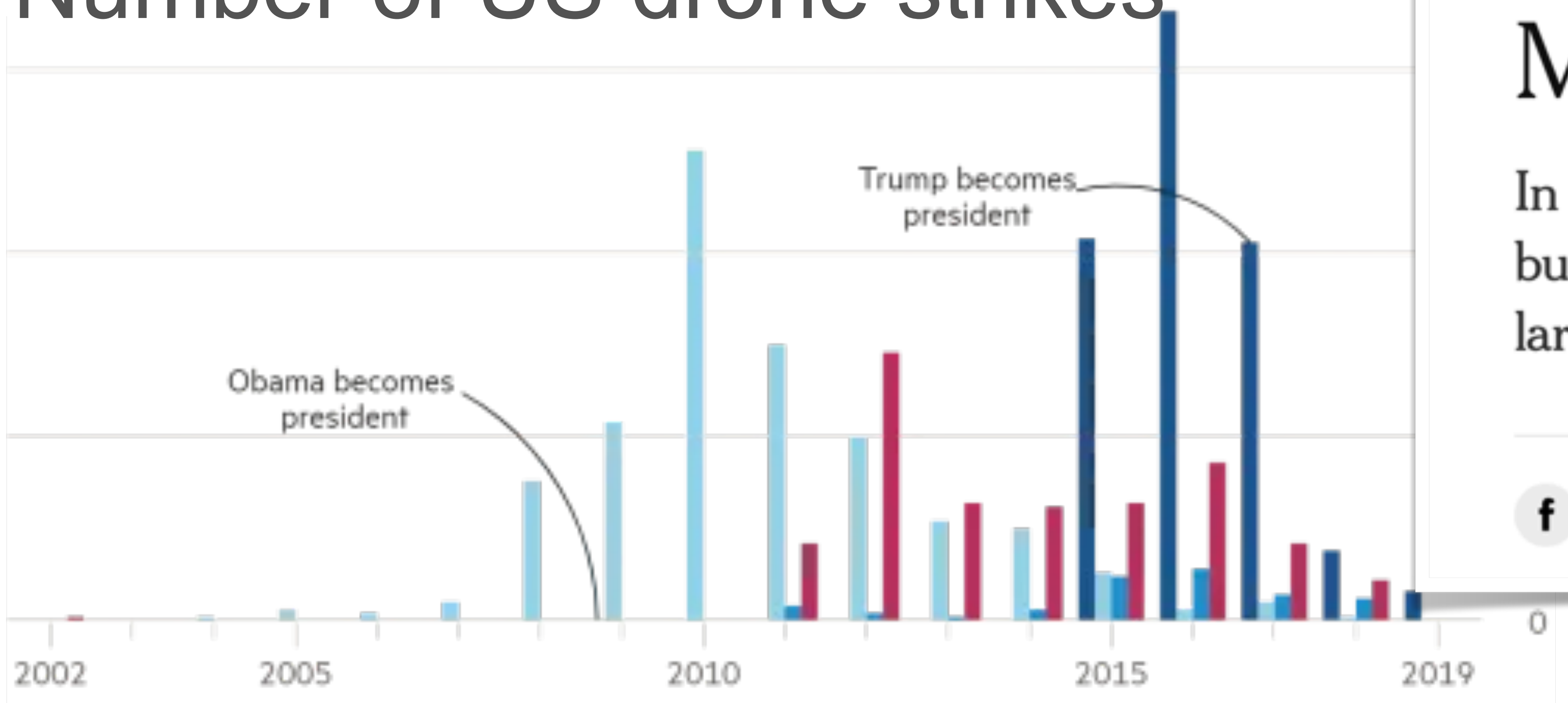
"3D Scanning Applications in Medical field: A Literature-based Review" Haleem et al. 2018



"Review of Laser Scanning Technologies and Applications" Soilán et al. 2019

…always ask yourselves why

Number of US drone strikes



*This is a conservative minimum strike count. Data is cross-referenced against local media reports, however many strikes are un- or under-reported
Source: The Bureau of Investigative Journalism
© FT

One Month, 500,000 Face Scans: How China Is Using A.I. to Profile a Minority

In a major ethical leap for the tech world, Chinese start-ups have built algorithms that the government uses to track members of a largely Muslim minority group.

QUESTIONABLE ETHICS ON 3D SCANNING SERVICES FOR BONE PRINTING

3D SCANNING SERVICES HELP IN A DIFFICULT SITUATION

The Horrible Things That Happen to Trans People Going Through Airport Security

"Nine out of ten times I've gone through a body scanner, I've been flagged for having an 'abnormality in the groin region.'"

Driver Charged in Uber's Fatal 2018 Autonomous Car Crash

Point cloud

Main representation for captured geometry

Point cloud

Main representation for captured geometry

Research questions include:

Point cloud

Main representation for captured geometry

Research questions include:

How to segment a point cloud?

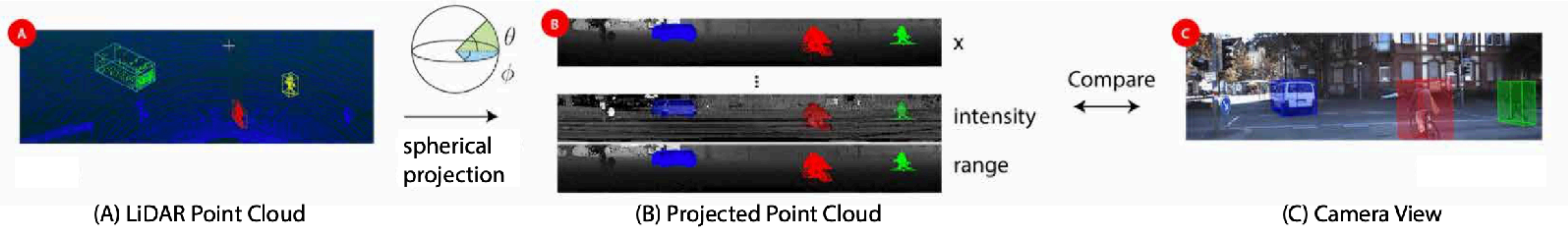


Fig. 2: LiDAR Projections. Note that each channel reflects structural information in the camera-view image.

"SqueezeSeg: […]" Wu et al. 2017 CVPR

Point cloud

Main representation for captured geometry

Research questions include:

How to segment a point cloud?

How to convert a point cloud?

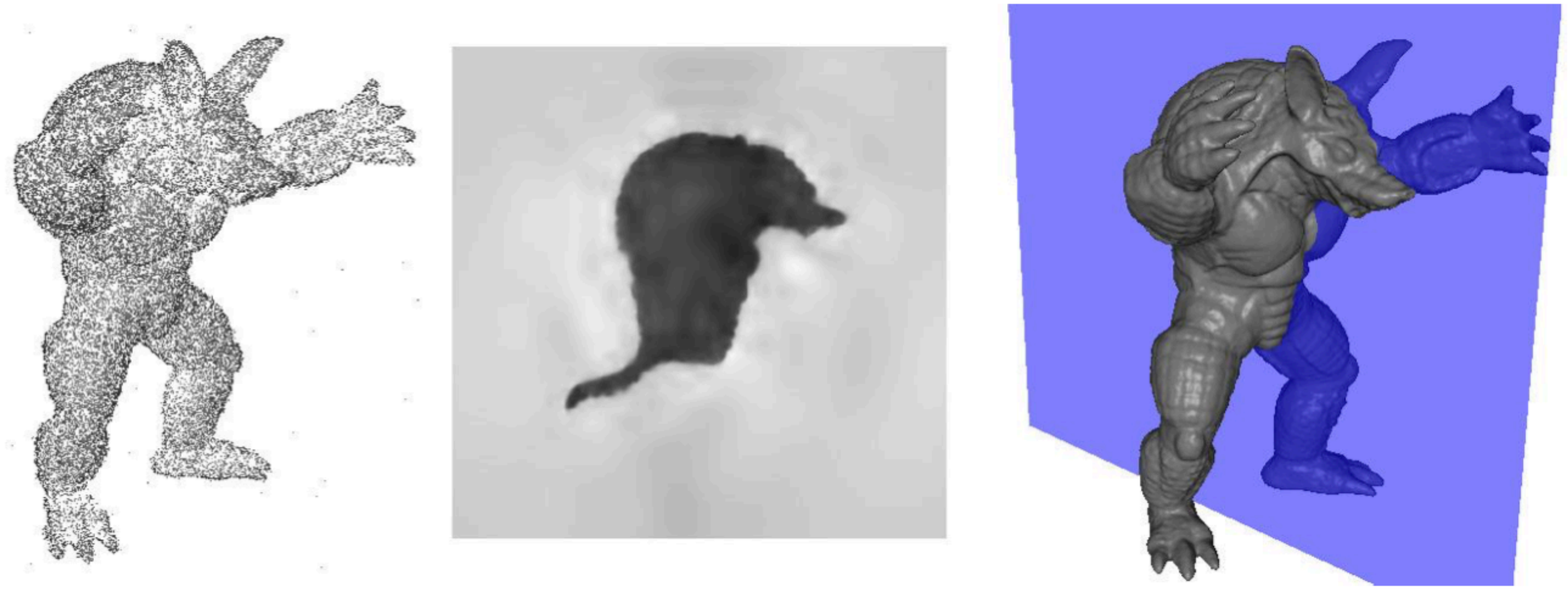
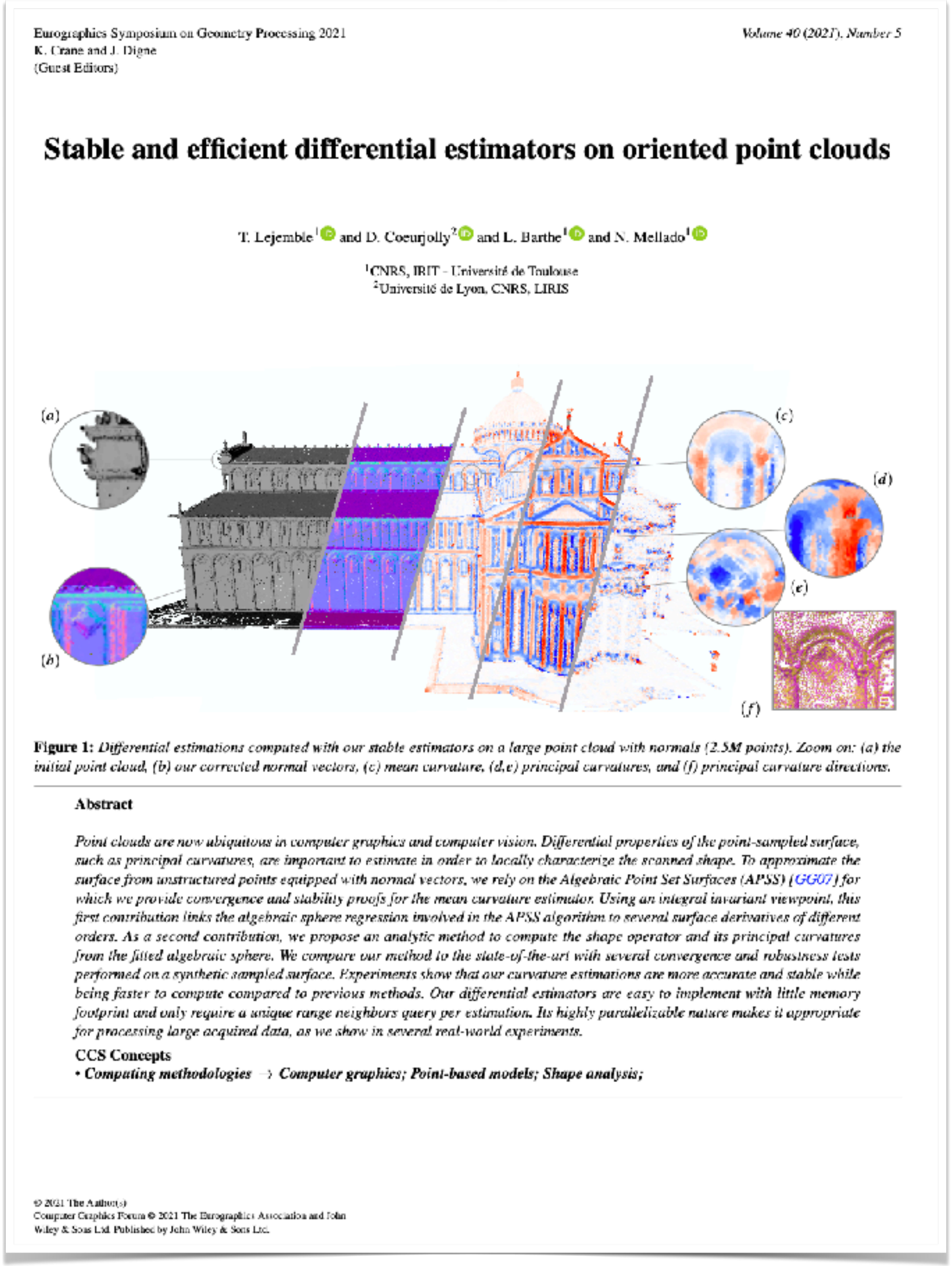


Figure 2: *Points from scans of the "Armadillo Man" model (left), our Poisson surface reconstruction (right), and a visualization of the indicator function (middle) along a plane through the 3D volume.*

"Poisson Surface Reconstruction" Kazhdan et al. 2006 SGP



Eurographics Symposium on Geometry Processing 2021
K. Crane and J. Digne
(Guest Editors)

Volume 40 (2021), Number 5

Stable and efficient differential estimators on oriented point clouds

T. Lejemble[1] and D. Coeurjolly[2] and L. Barthe[1] and N. Mellado[1]

[1]CNRS, IRIT - Université de Toulouse
[2]Université de Lyon, CNRS, LIRIS

Figure 1: *Differential estimations computed with our stable estimators on a large point cloud with normals (2.5M points). Zoom on: (a) the initial point cloud, (b) our corrected normal vectors, (c) mean curvature, (d,e) principal curvatures, and (f) principal curvature directions.*

Abstract

Point clouds are now ubiquitous in computer graphics and computer vision. Differential properties of the point-sampled surface, such as principal curvatures, are important to estimate in order to locally characterize the scanned shape. To approximate the surface from unstructured points equipped with normal vectors, we rely on the Algebraic Point Set Surfaces (APSS) (GG07) for which we provide convergence and stability proofs for the mean curvature estimator. Using an integral invariant viewpoint, this first contribution links the algebraic sphere regression involved in the APSS algorithm to several surface derivatives of different orders. As a second contribution, we propose an analytic method to compute the shape operator and its principal curvatures from the fitted algebraic sphere. We compare our method to the state-of-the-art with several convergence and robustness tests performed on a synthetic sampled surface. Experiments show that our curvature estimations are more accurate and stable while being faster to compute compared to previous methods. Our differential estimators are easy to implement with little memory footprint and only require a unique range neighbors query per estimation. Its highly parallelizable nature makes it appropriate for processing large acquired data, as we show in several real-world experiments.

CCS Concepts
• *Computing methodologies* → *Computer graphics; Point-based models; Shape analysis;*

Point cloud

Main representation for captured geometry

Research questions include:

- How to segment a point cloud?
- How to convert a point cloud?
- How to define operators on a point cloud?

SHAPE REPRESENTATIONS IN 3D

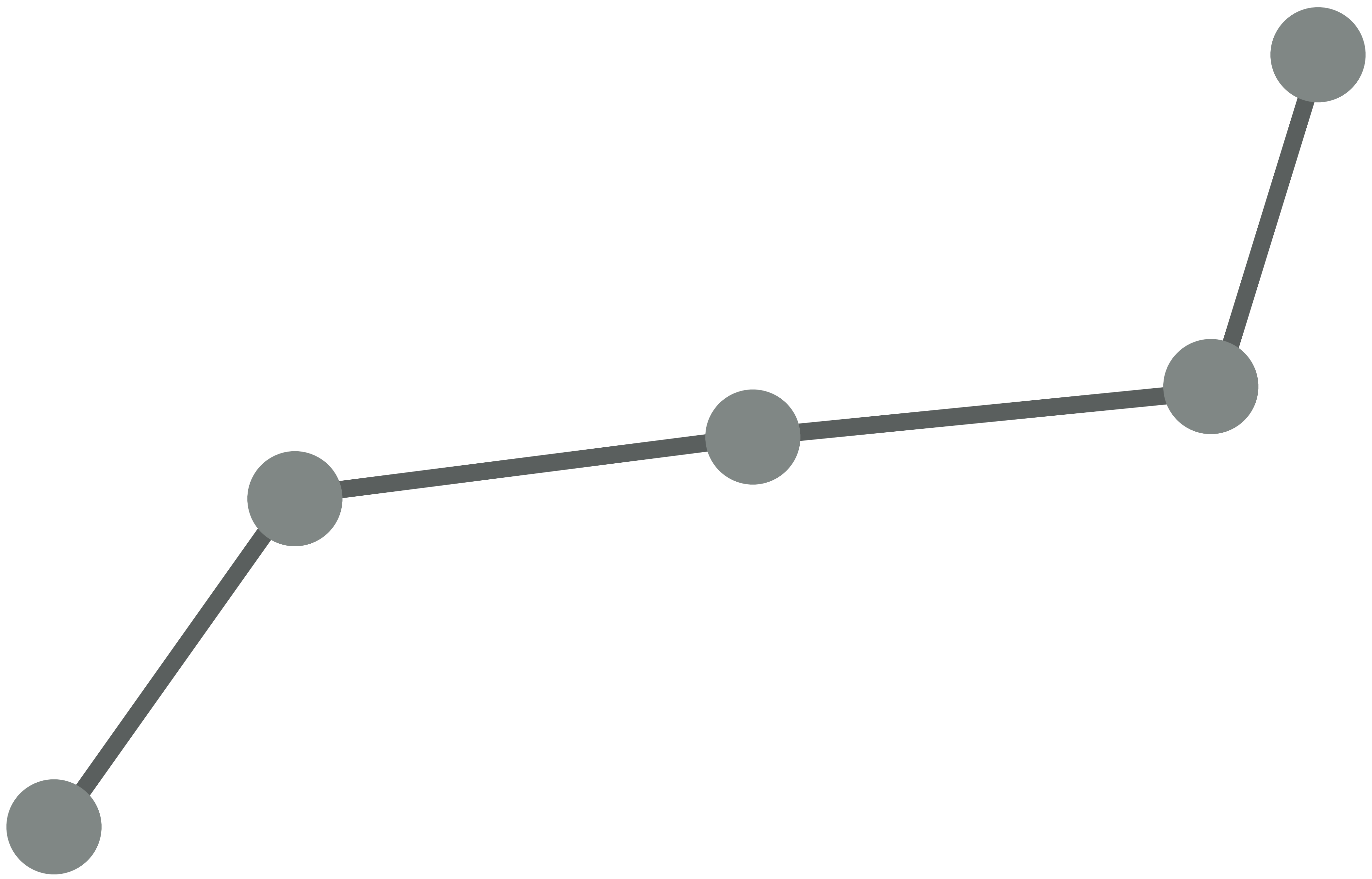
Point cloud

Points + Connectivity + Piecewise flat interpolation

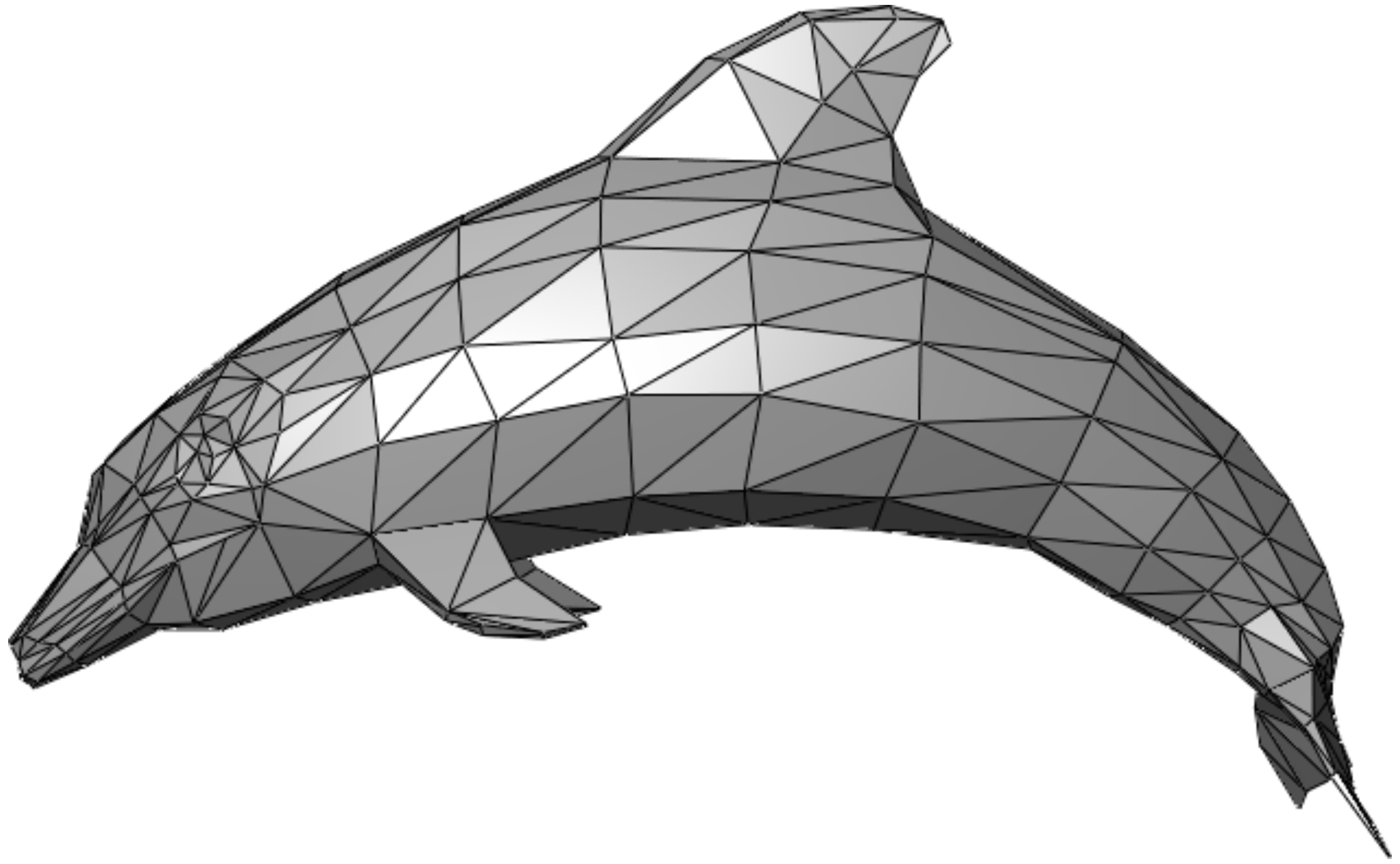
Implicit

Points + Connectivity + Polynomial interpolation

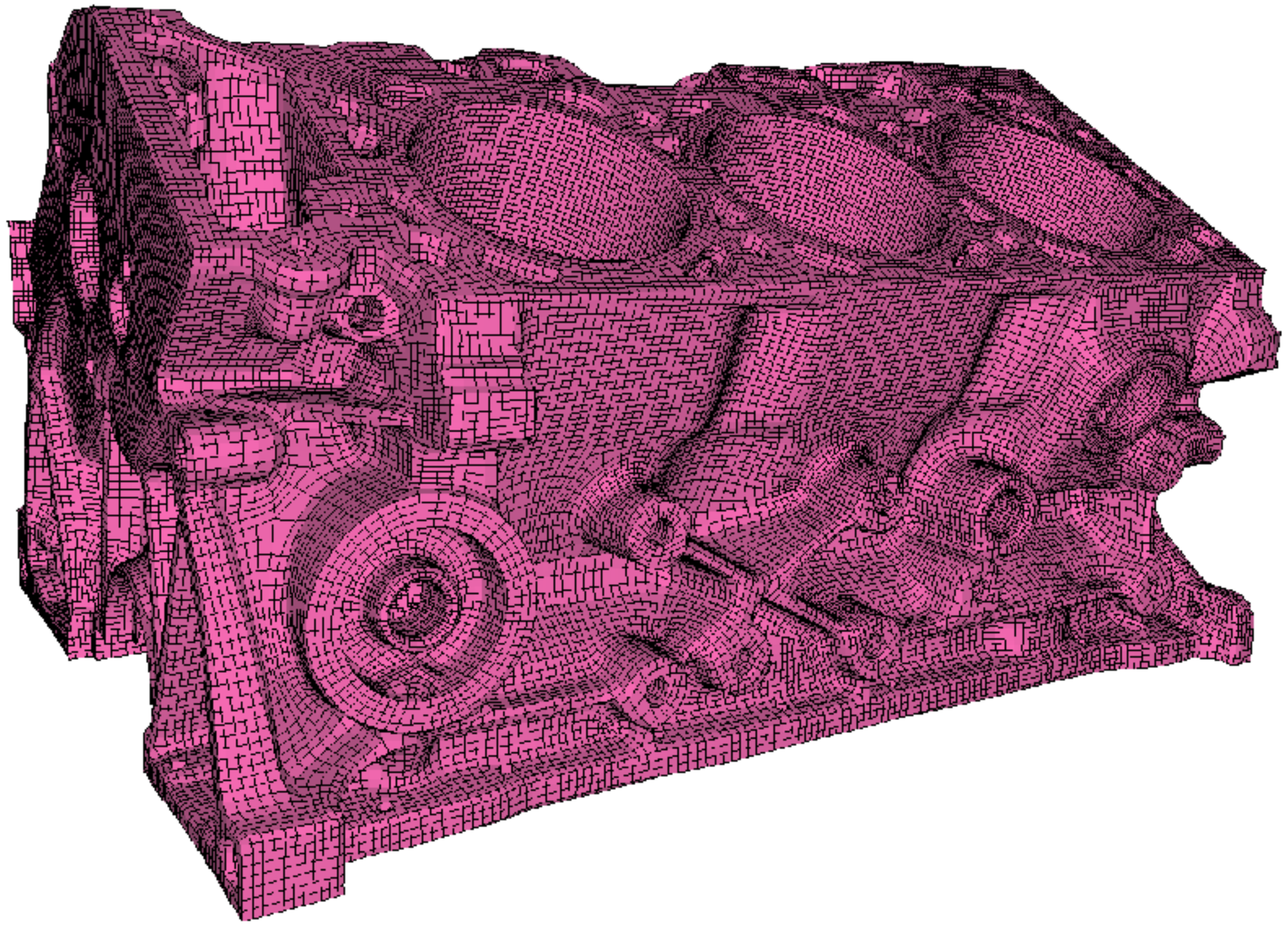
Points + Connectivity + Piecewise flat interpolation



Mesh



Triangle mesh

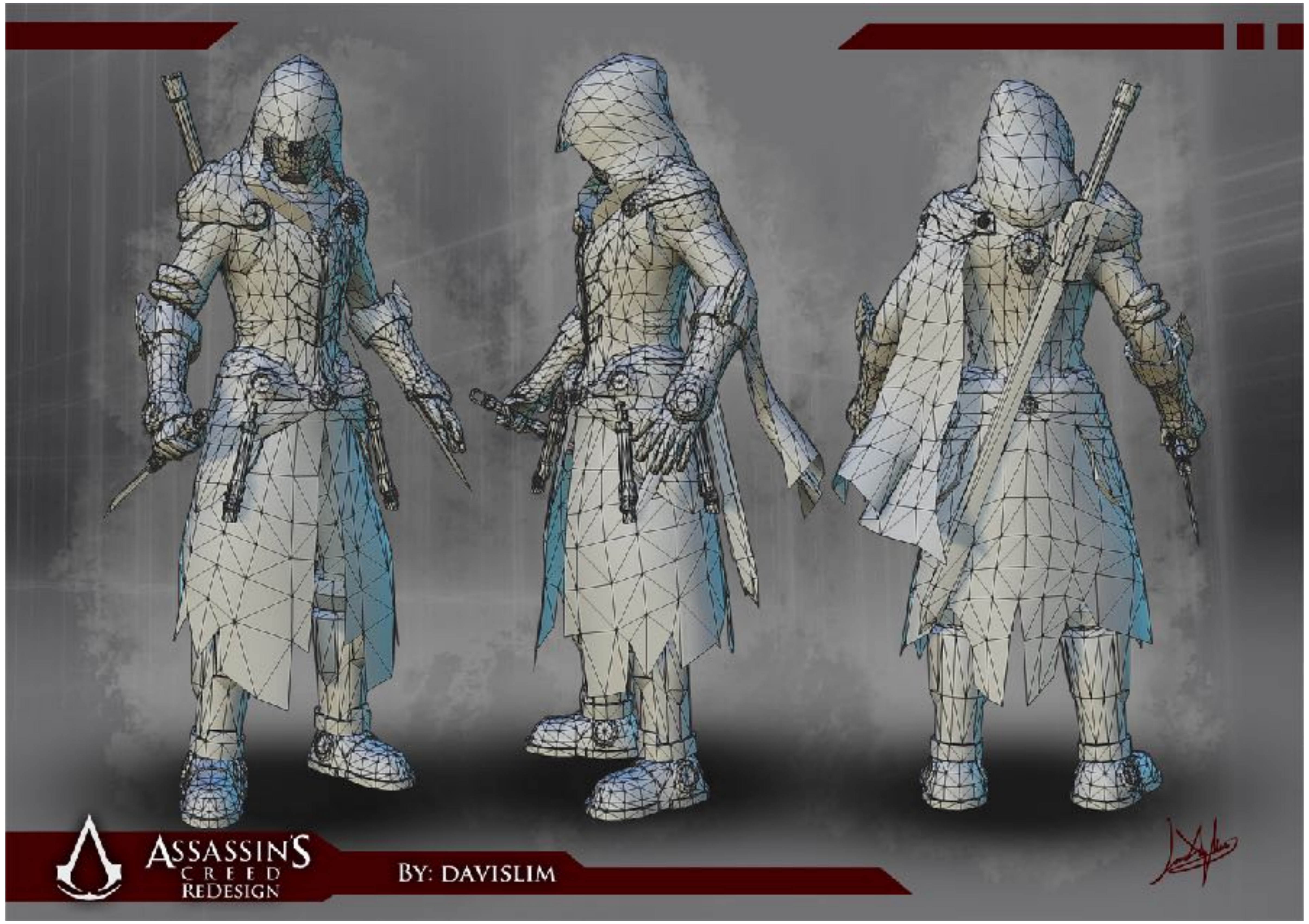


Quad mesh

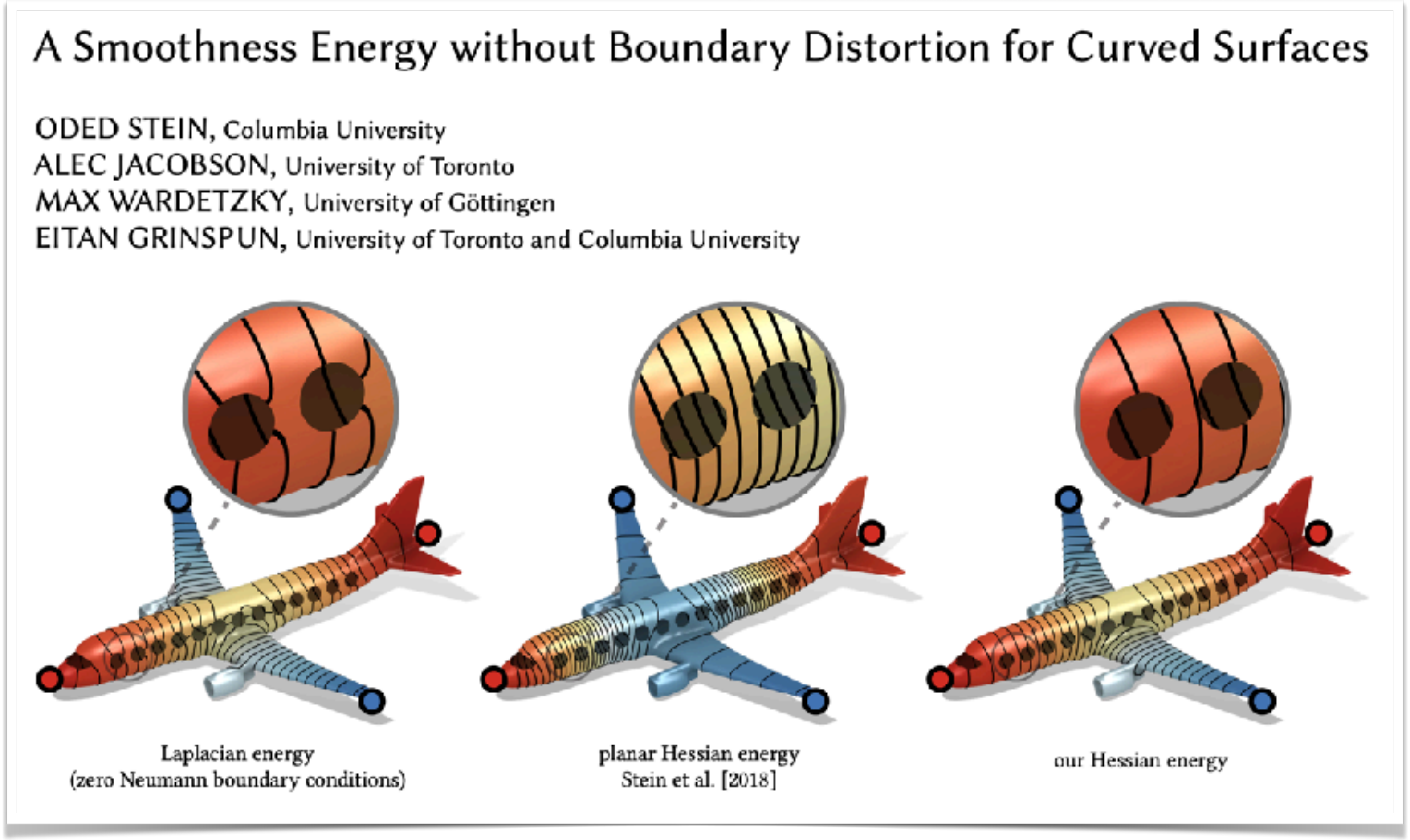
Mesh

Most common shape representation

Mesh



Most common shape representation

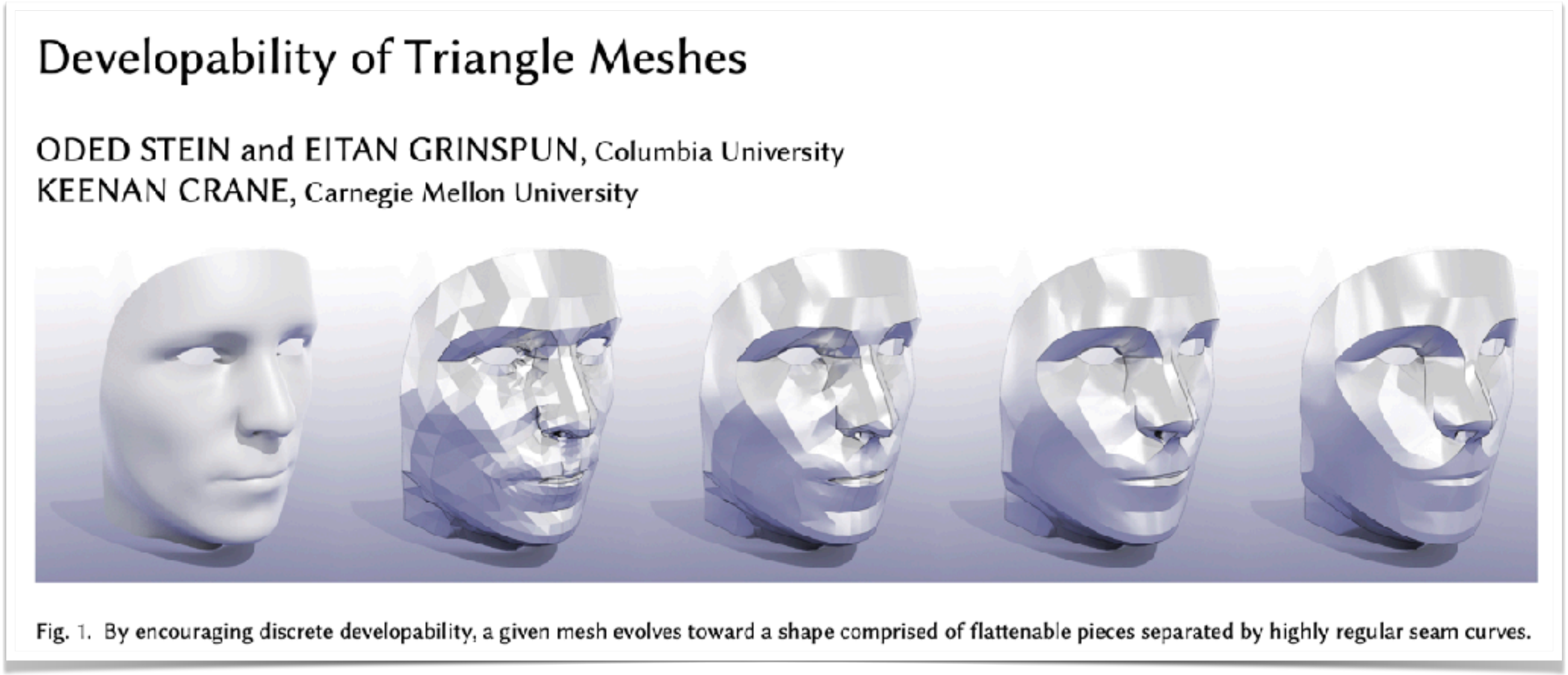

A Smoothness Energy without Boundary Distortion for Curved Surfaces

ODED STEIN, Columbia University
ALEC JACOBSON, University of Toronto
MAX WARDETZKY, University of Göttingen
EITAN GRINSPUN, University of Toronto and Columbia University

Mesh

Most common shape representation

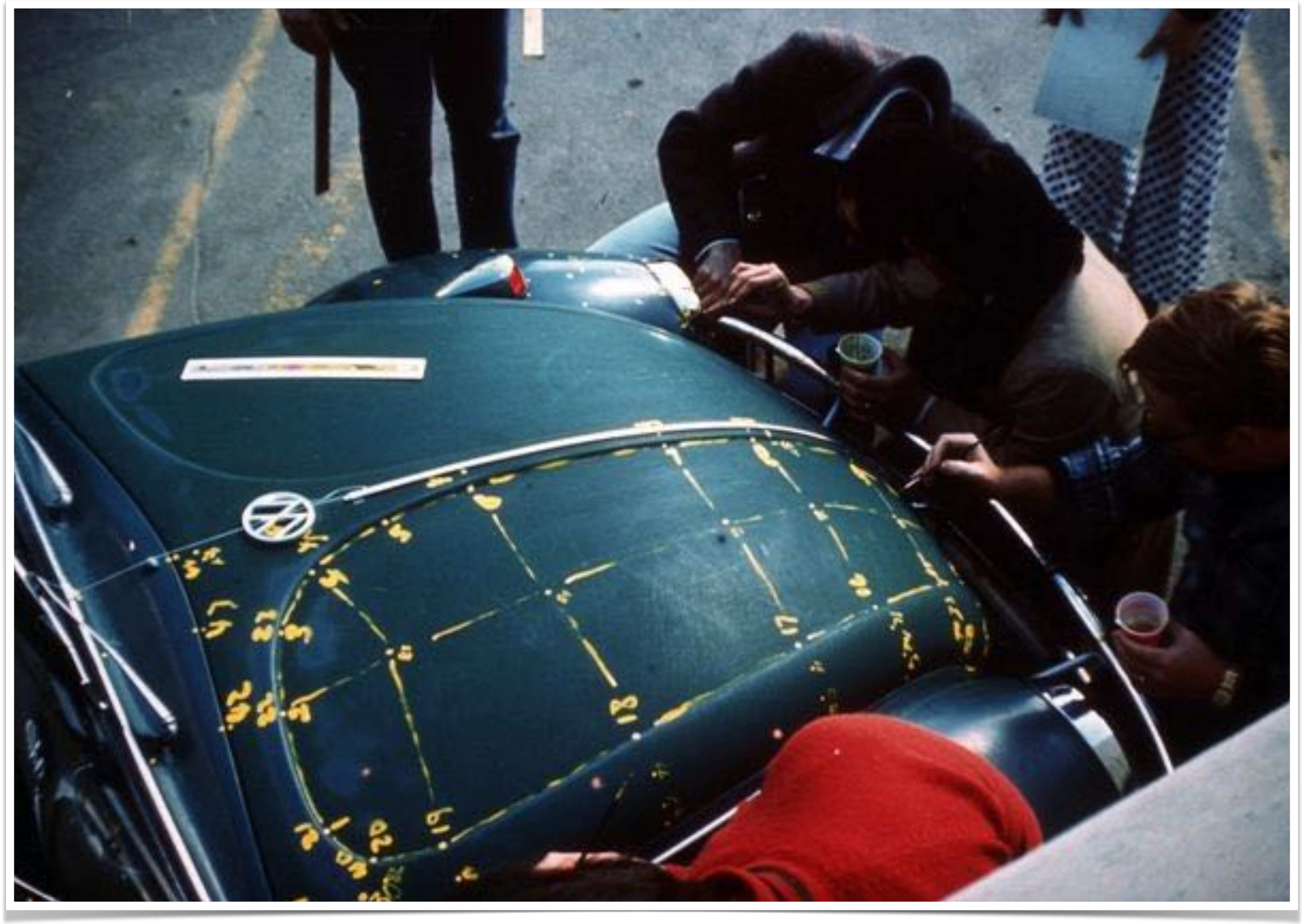
Easiest to work with for *most* applications


Developability of Triangle Meshes

ODED STEIN and EITAN GRINSPUN, Columbia University
KEENAN CRANE, Carnegie Mellon University

Fig. 1. By encouraging discrete developability, a given mesh evolves toward a shape comprised of flattenable pieces separated by highly regular seam curves.

Mesh



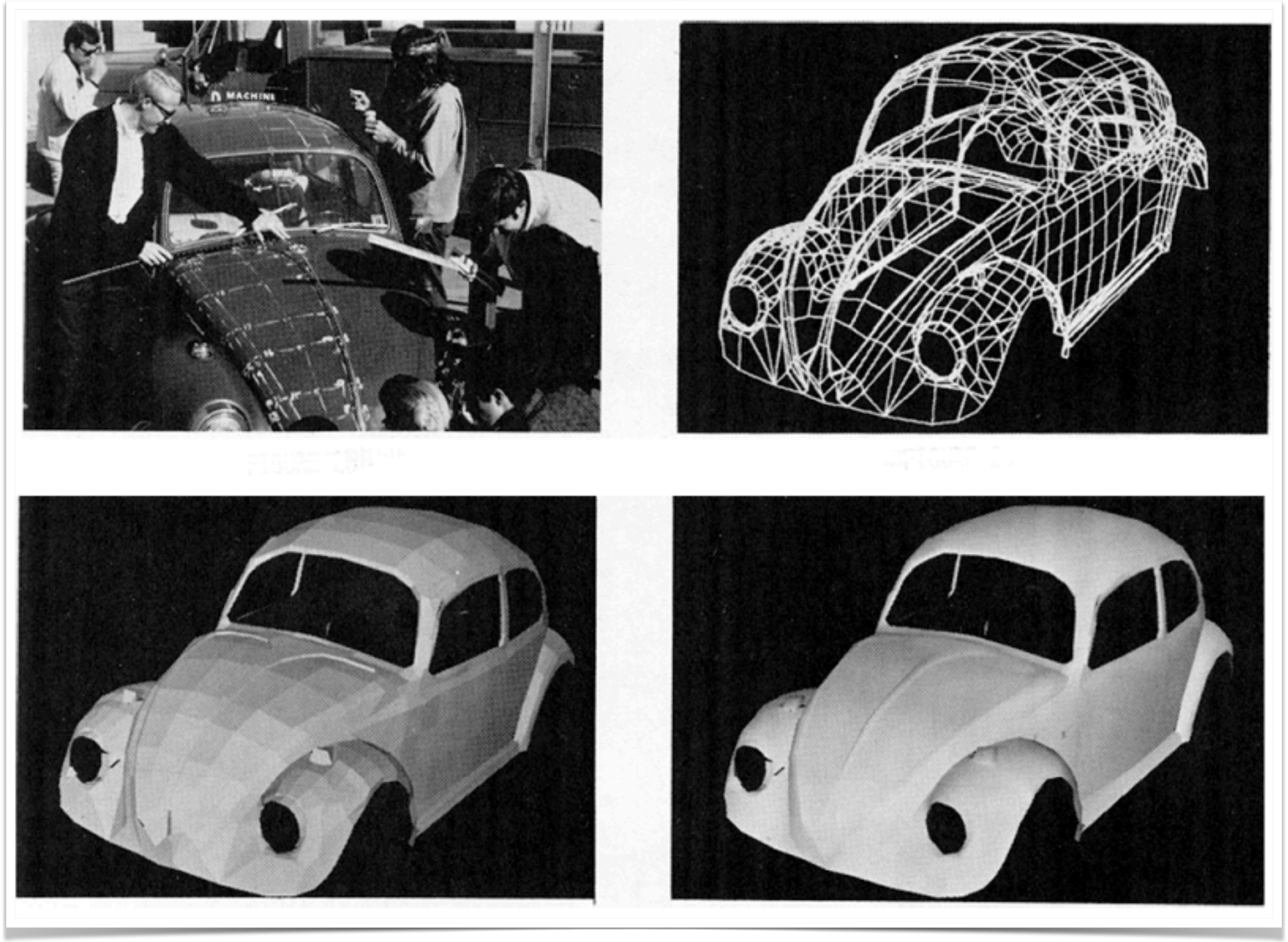
“The first real object ever 3D scanned and rendered was a WV Beetle” by Jason Torchinsky

Most common shape representation

Easiest to work with for *most* applications

Very hard/impossible to capture directly

Mesh



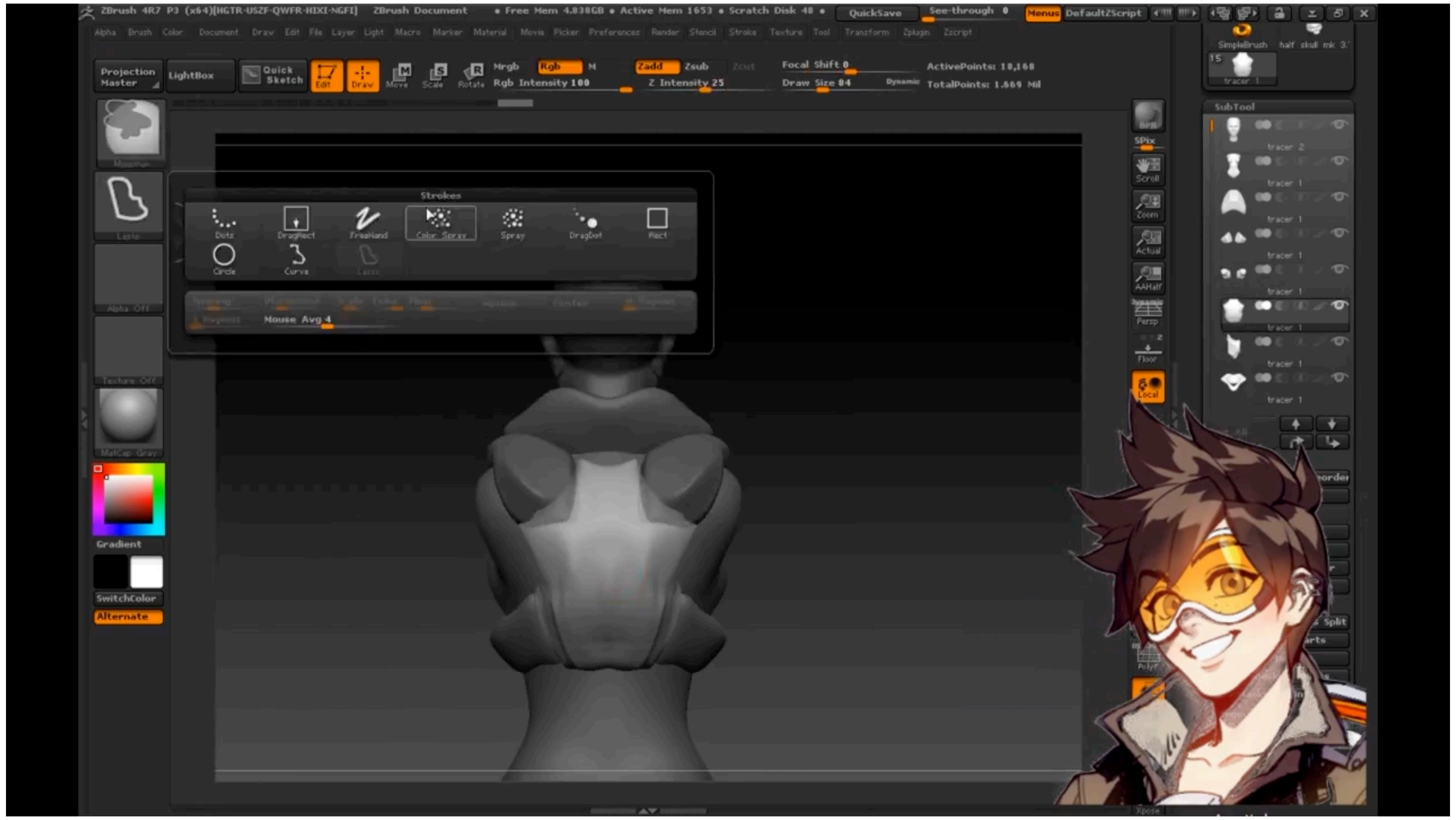
"The first real object ever 3D scanned and rendered was a VW Beetle" by Jason Torchinsky

Most common shape representation

Easiest to work with for *most* applications

Very hard/impossible to capture directly

Mesh



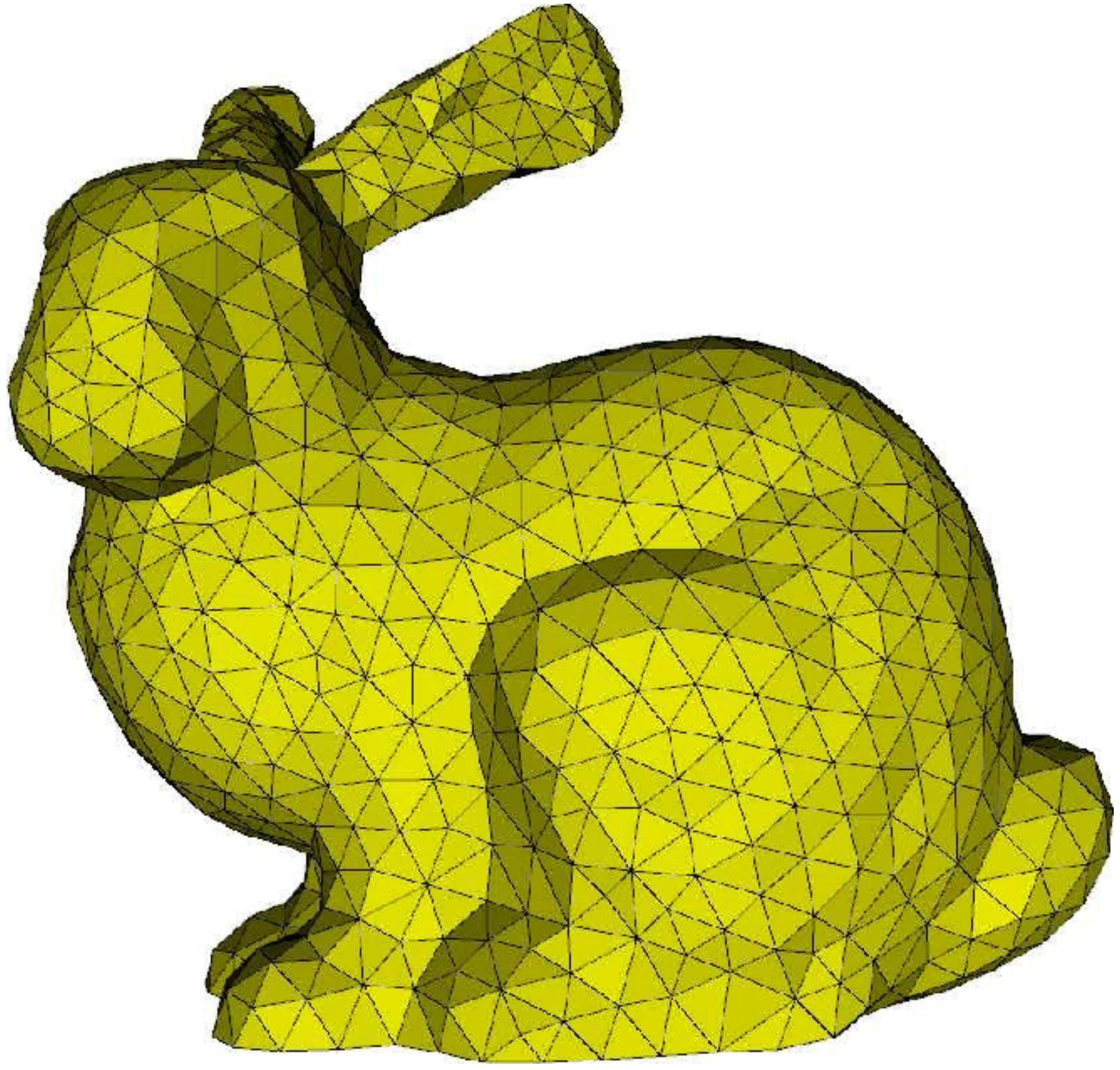
Most common shape representation

Easiest to work with for *most* applications

Very hard/impossible to capture directly

Digital design or converted point clouds

Mesh



Most common shape representation

Easiest to work with for *most* applications

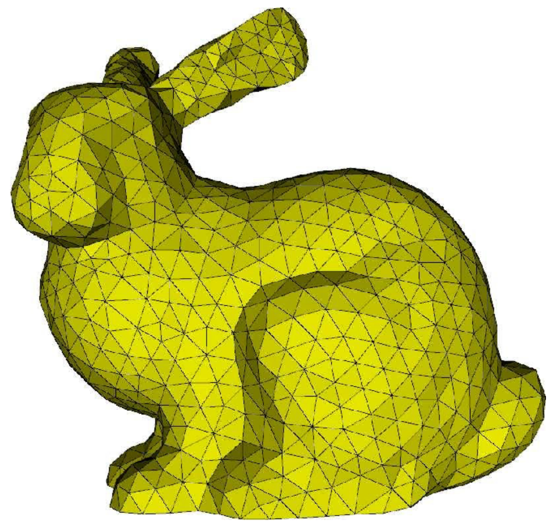
Very hard/impossible to capture directly

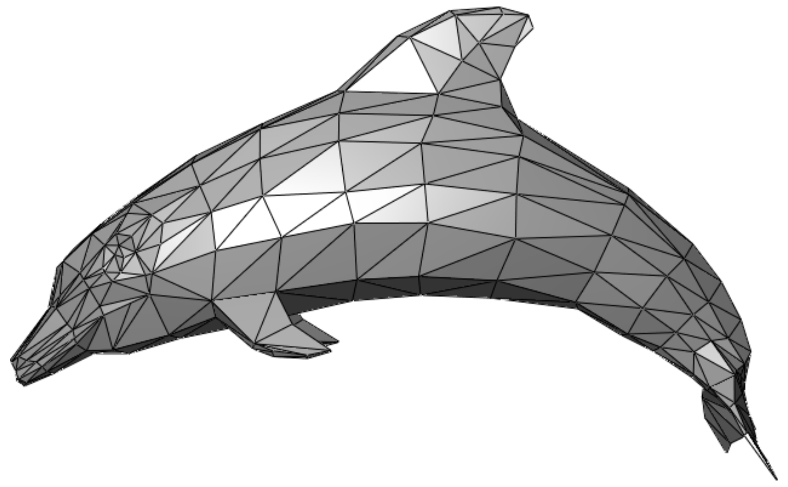
Digital design or converted point clouds

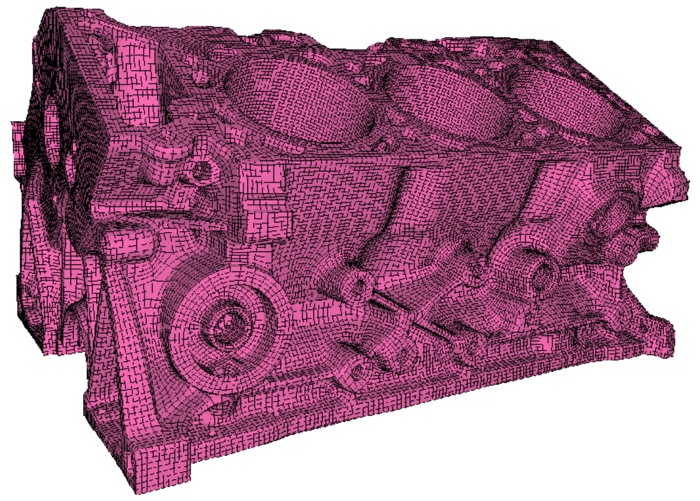
SHAPE REPRESENTATIONS IN 3D

Point cloud

Meshes



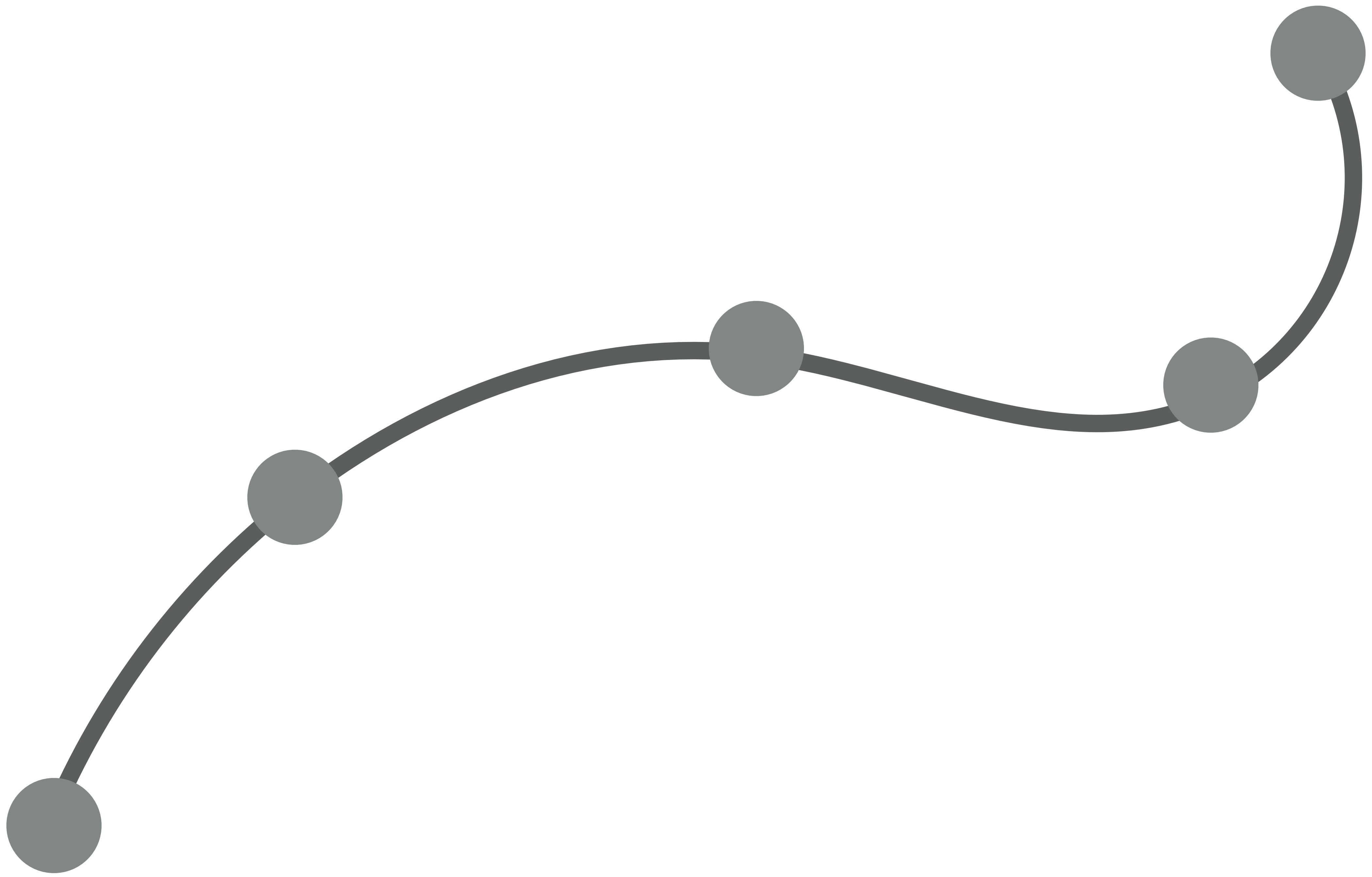




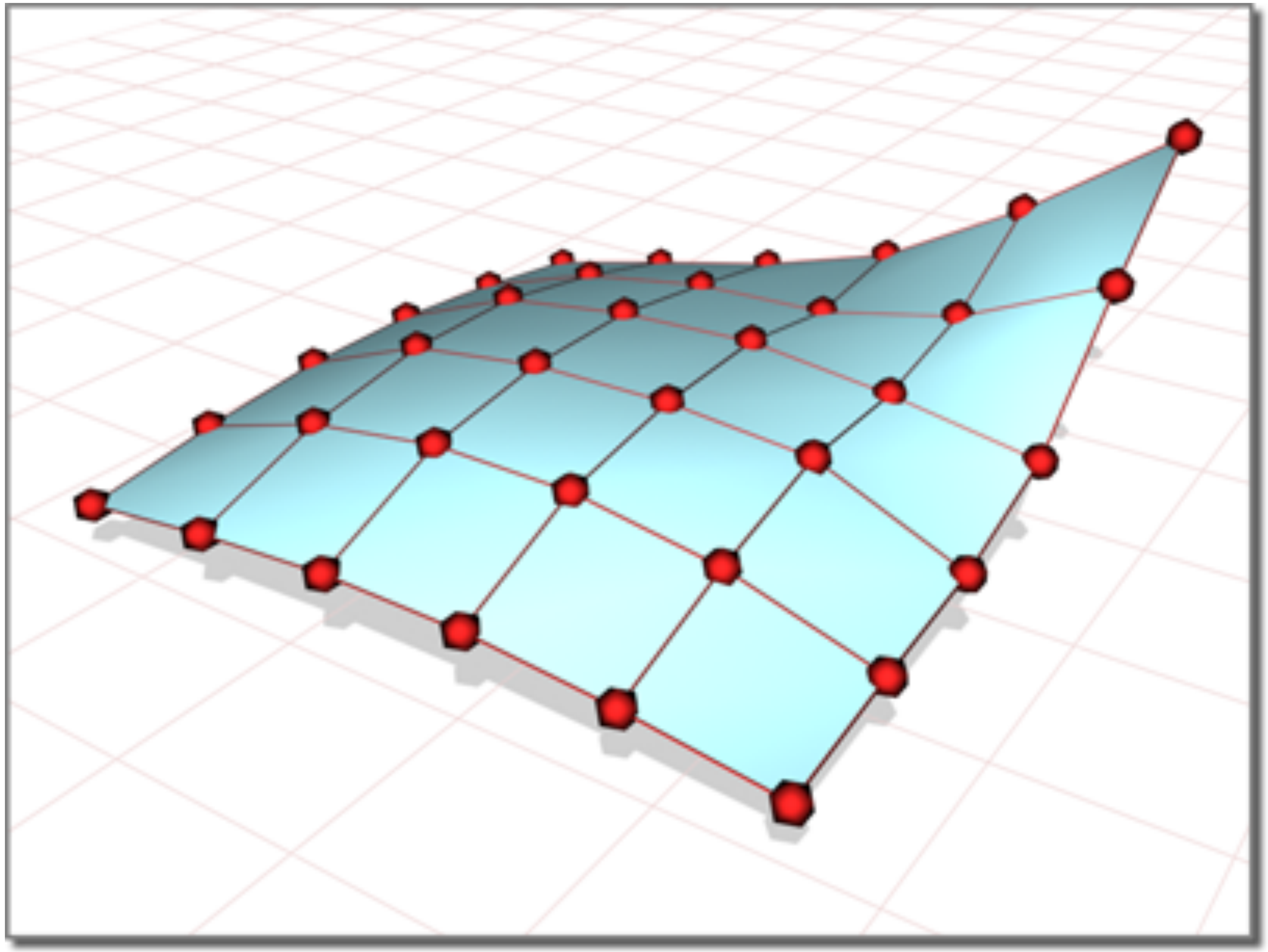
Implicit

Points + Connectivity + Polynomial interpolation

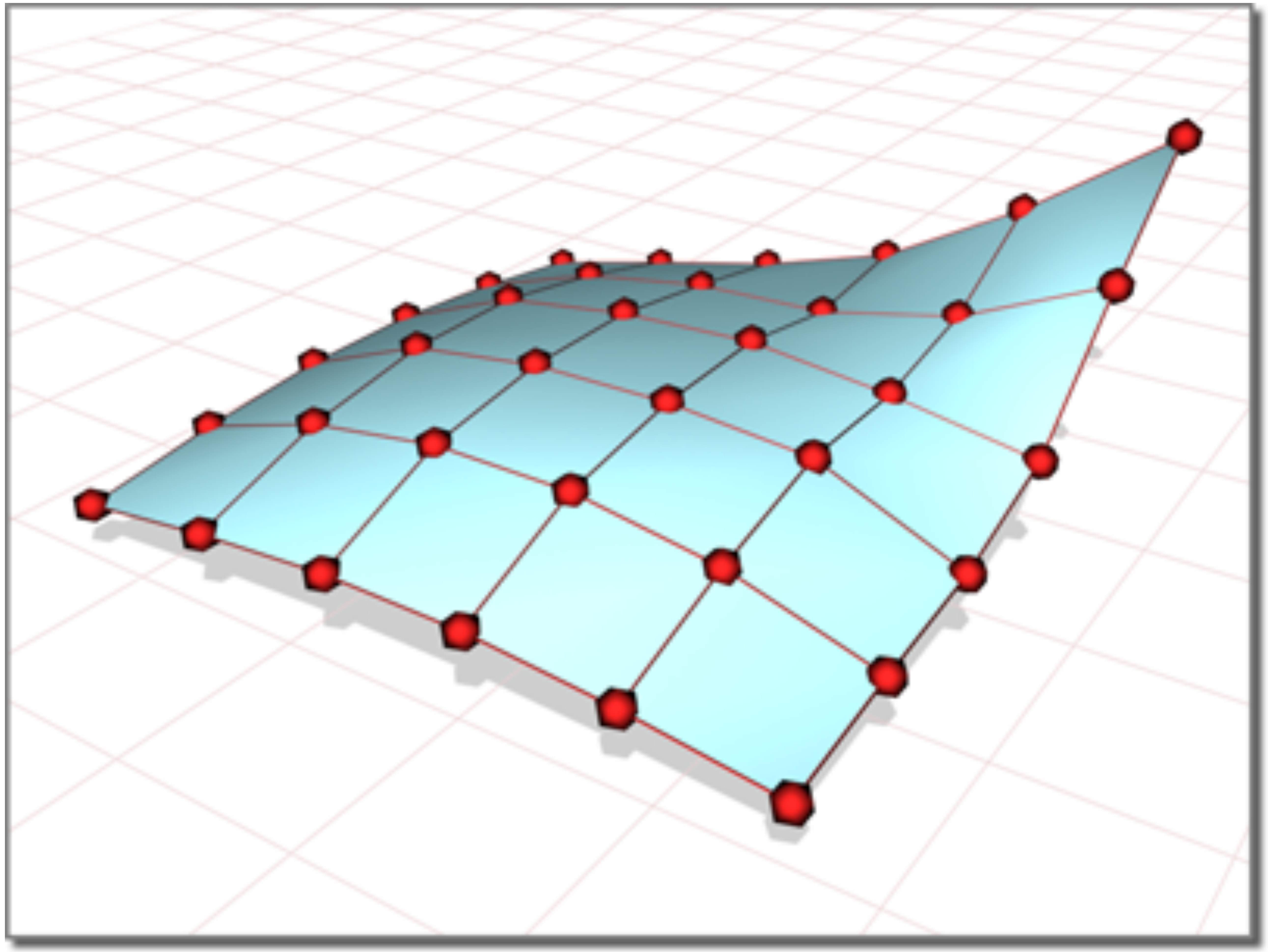
Points + Connectivity + Piecewise polynomial interpolation



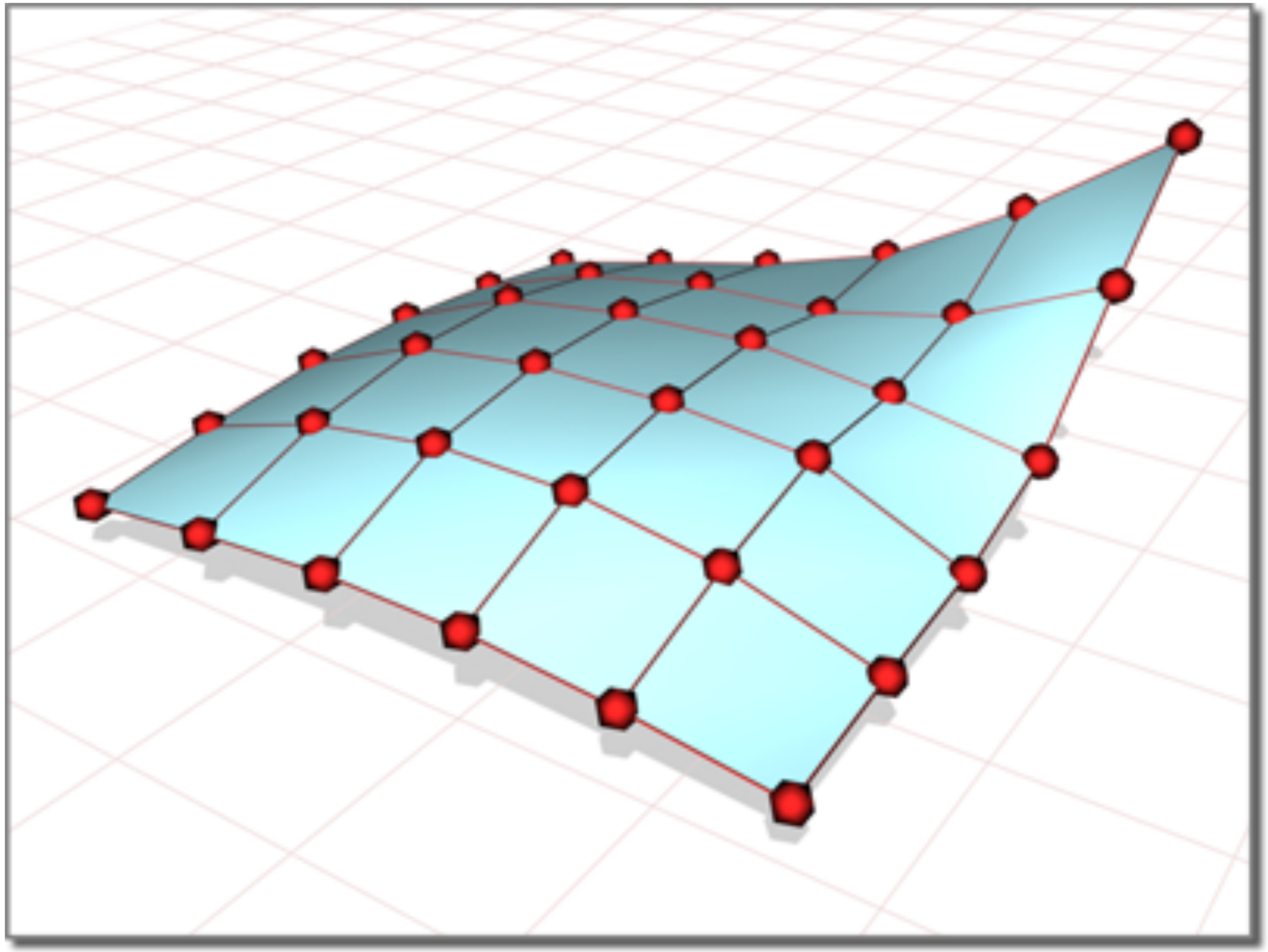
Points + Connectivity + Piecewise polynomial interpolation



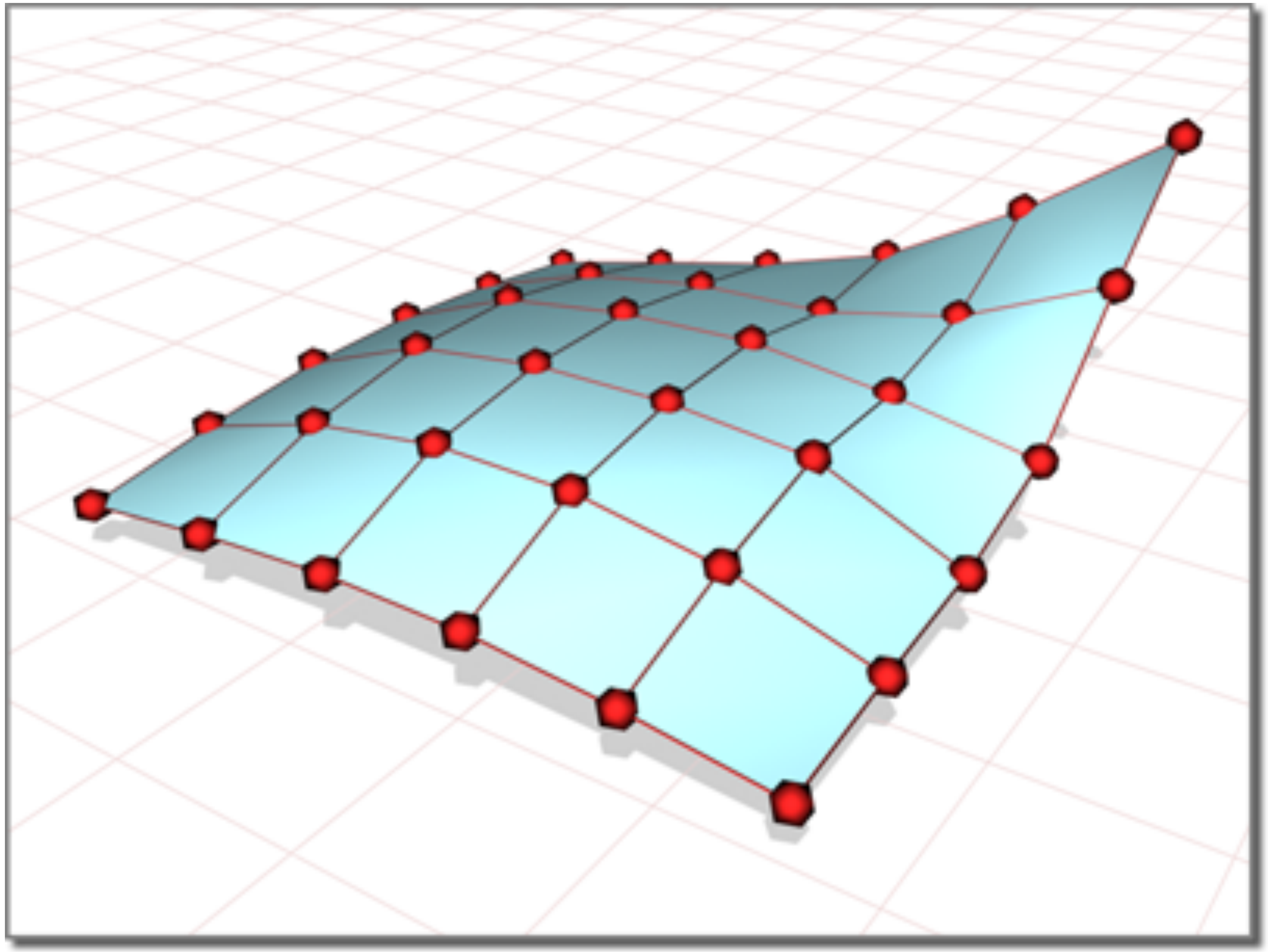
Parametric surface



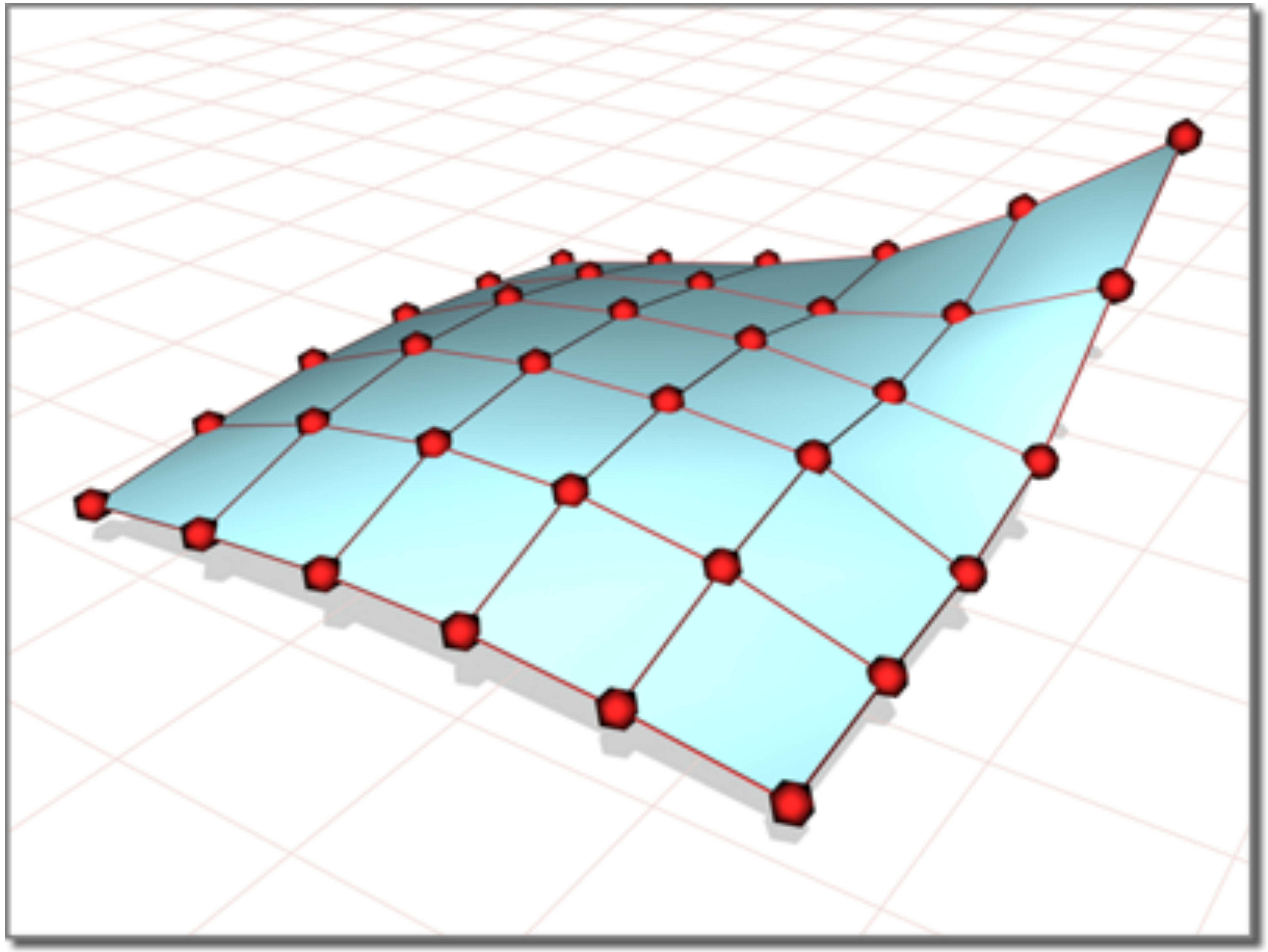
Curved surface



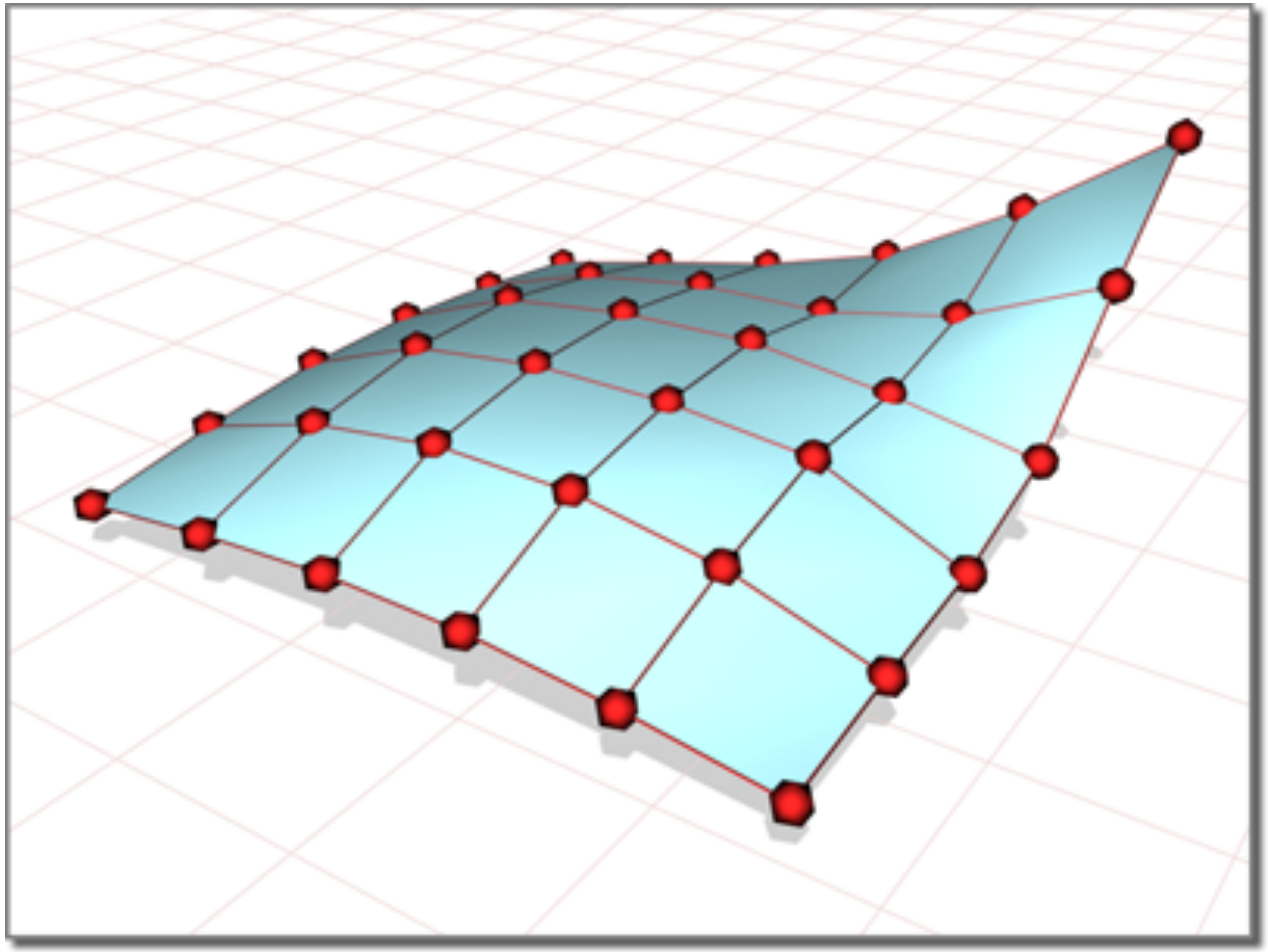
Freeform surface



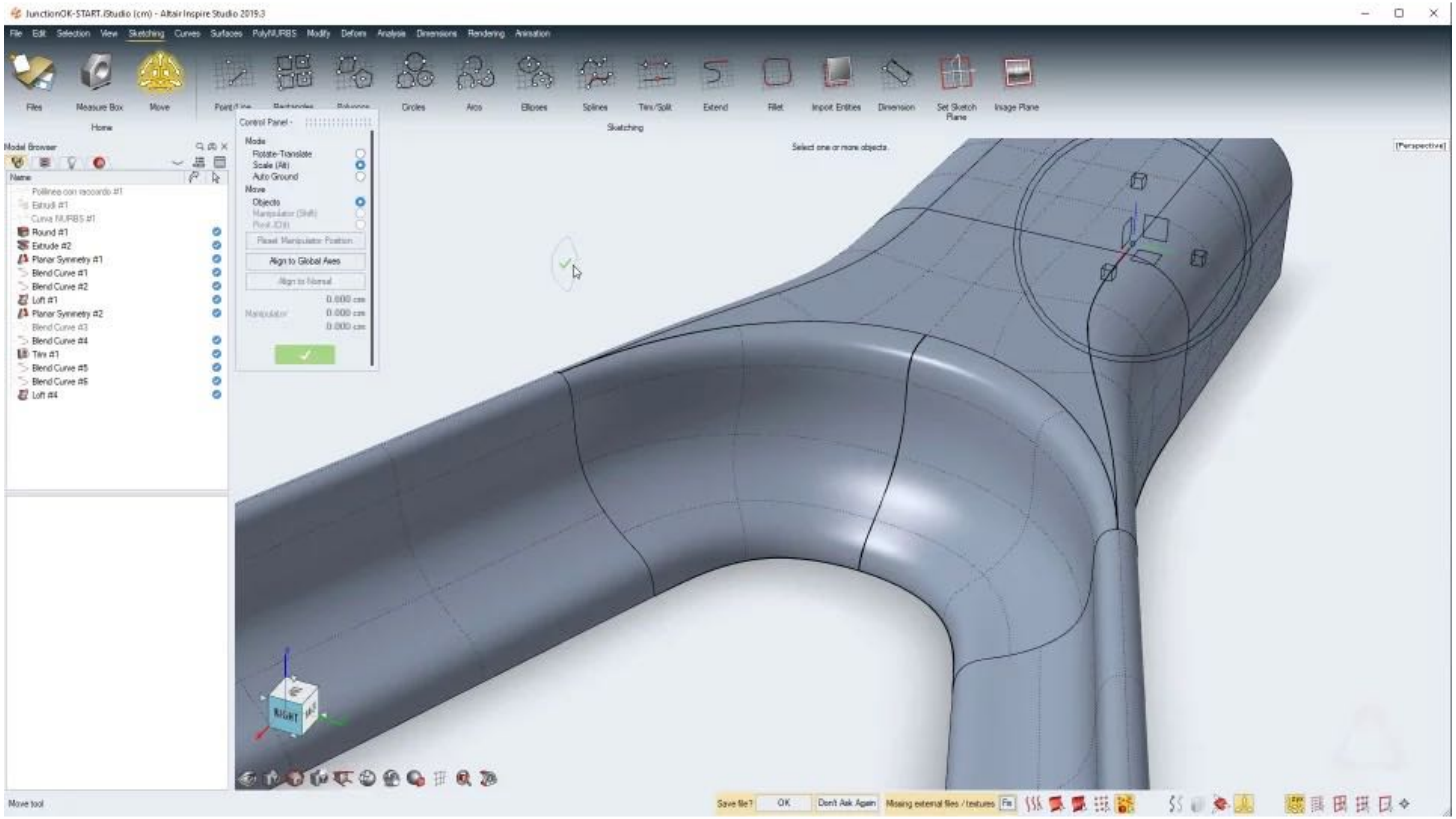
Surface spline

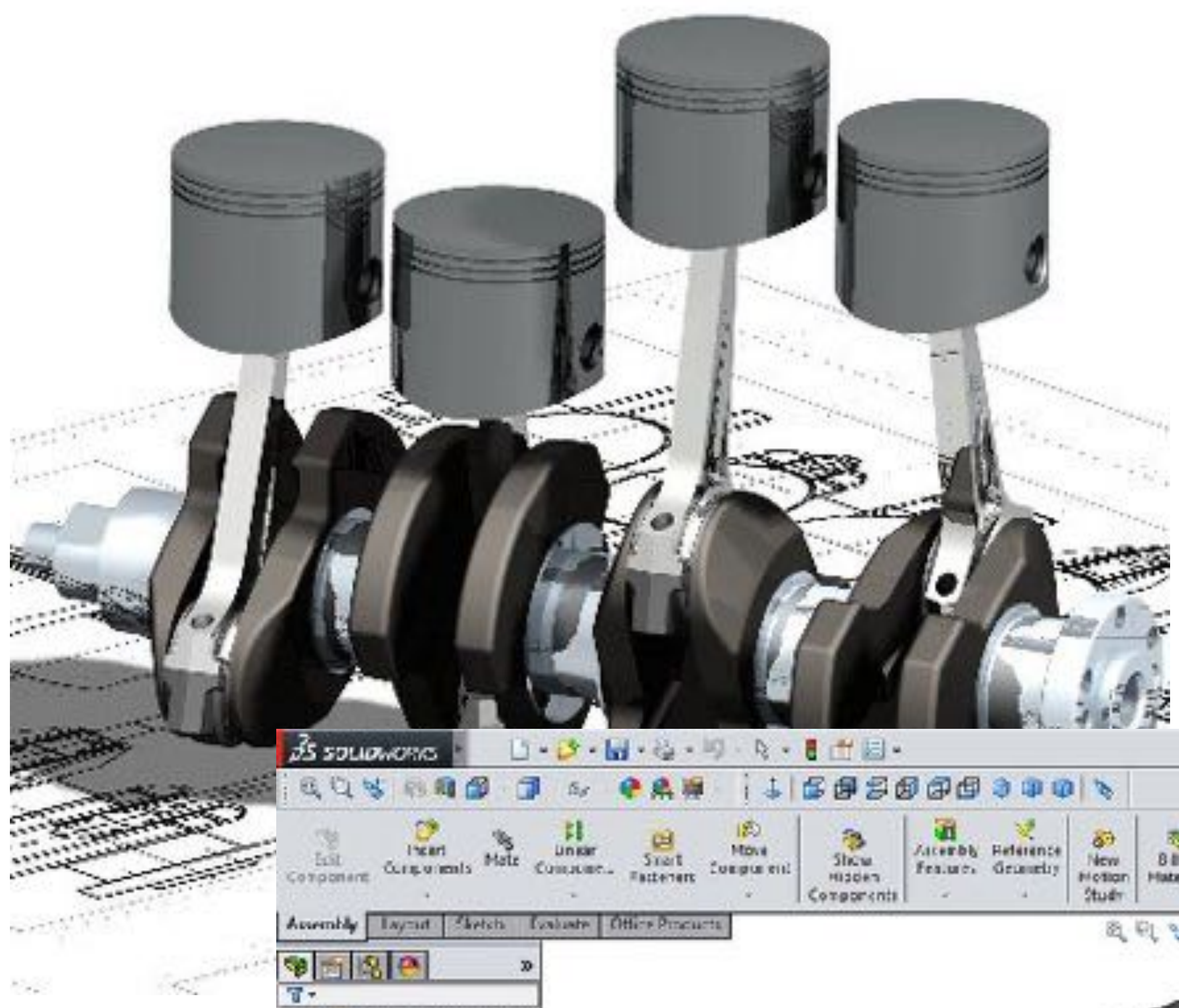


CAD surface

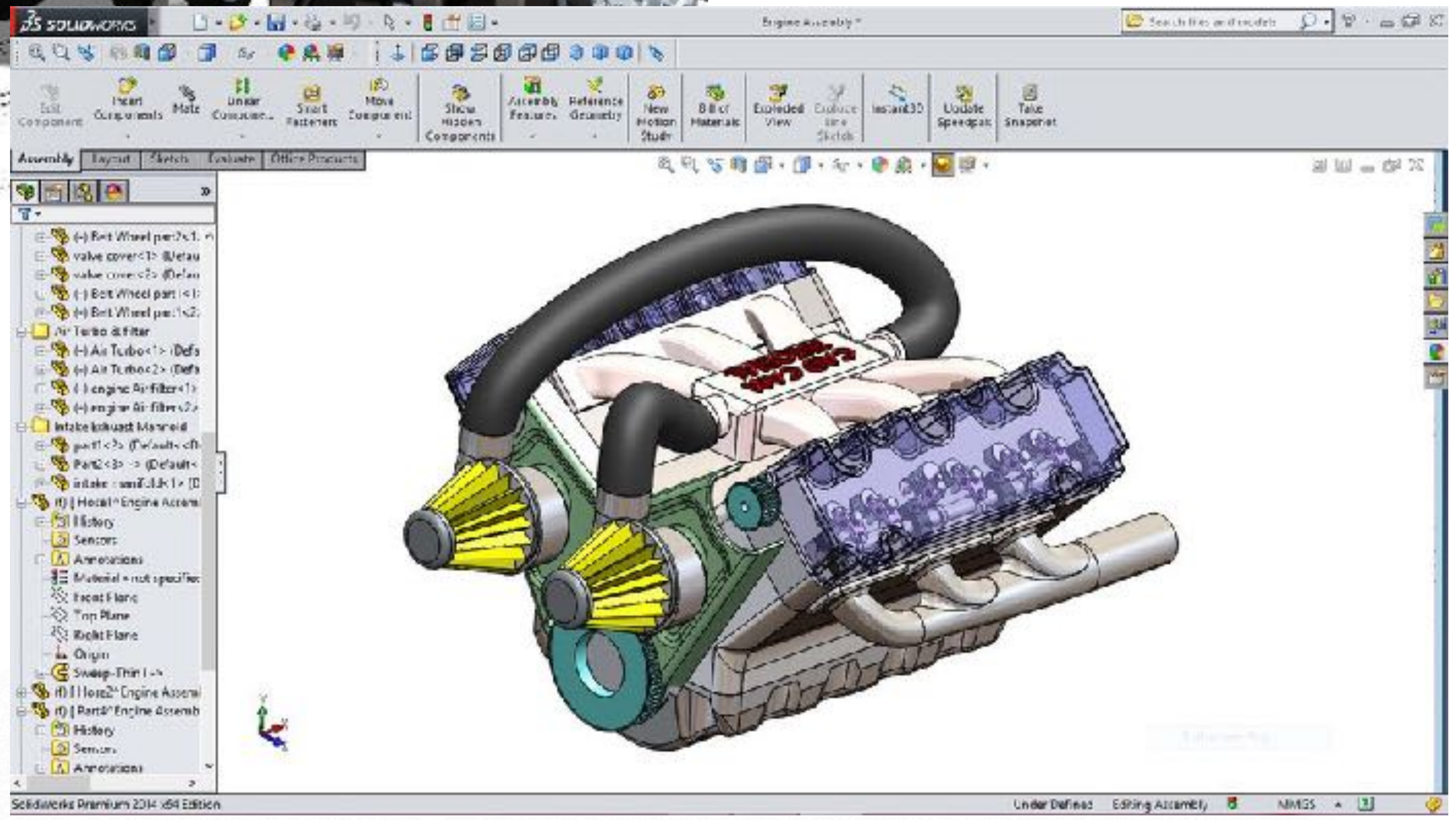


NURBS

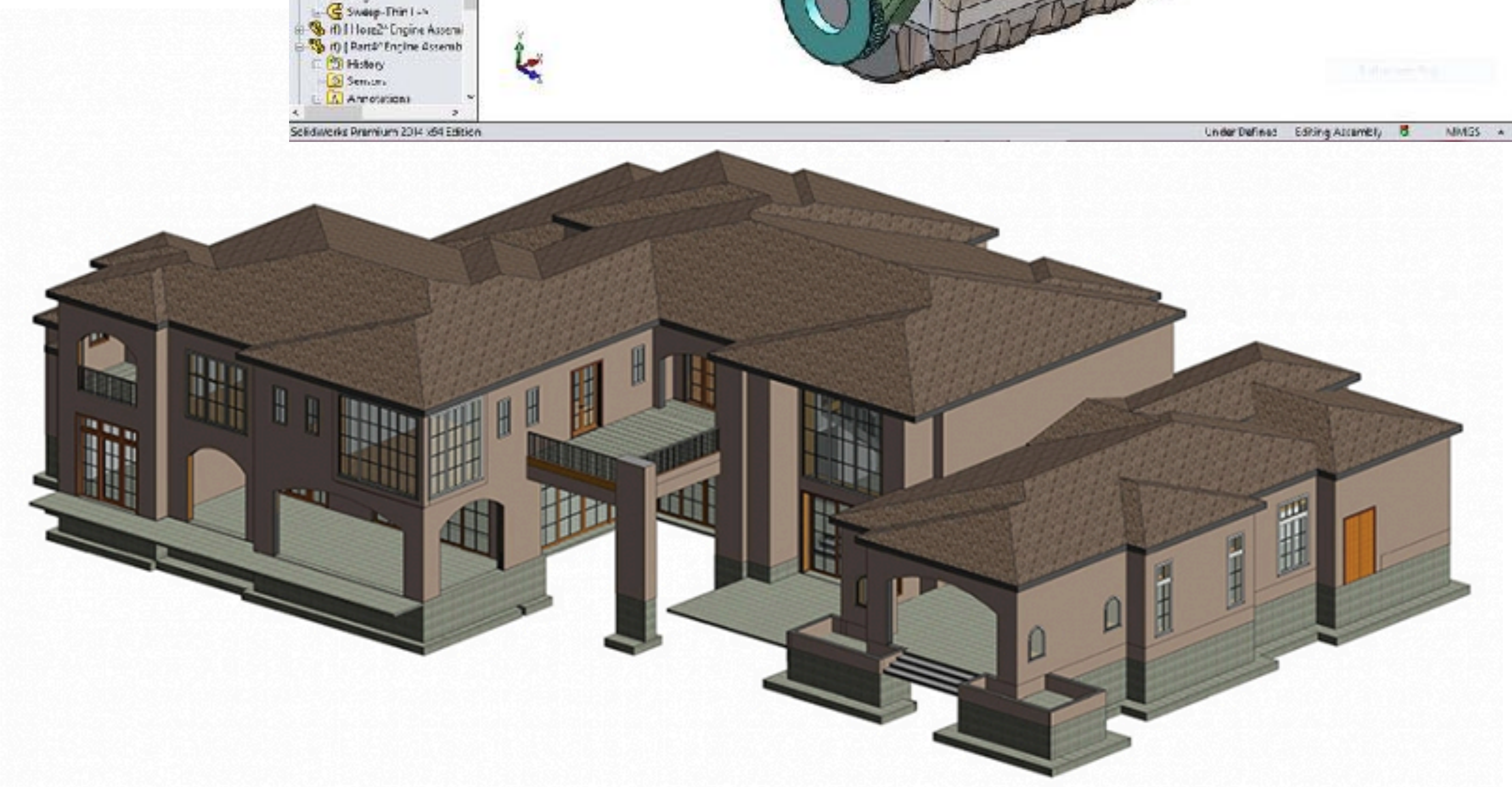

JunctionOK-START.iStudio (cm) - Altair Inspire Studio 2019.3
File Edit Selection View Sketching Curves Surfaces PolyNURBS Modify Deform Analysis Dimensions Rendering Animation
Files Measure Box Move
Home
Circles Arcs Ellipses Splines Trim/Split Extend Fillet Import Entities Dimension Set Sketch Plane Image Plane
Sketching
Control Panel
Mode
Rotate-Translate
Scale (Alt)
Auto Ground
Move
Objects
Align to Global Axes
Model Browser
Name
Round #1
Extrude #2
Planar Symmetry #1
Blend Curve #1
Blend Curve #2
Loft #1
Planar Symmetry #2
Blend Curve #4
Trim #1
Blend Curve #5
Blend Curve #6
Loft #4
Select one or more objects.
[Perspective]
Move tool
Save file? OK Don't Ask Again Missing external files / textures



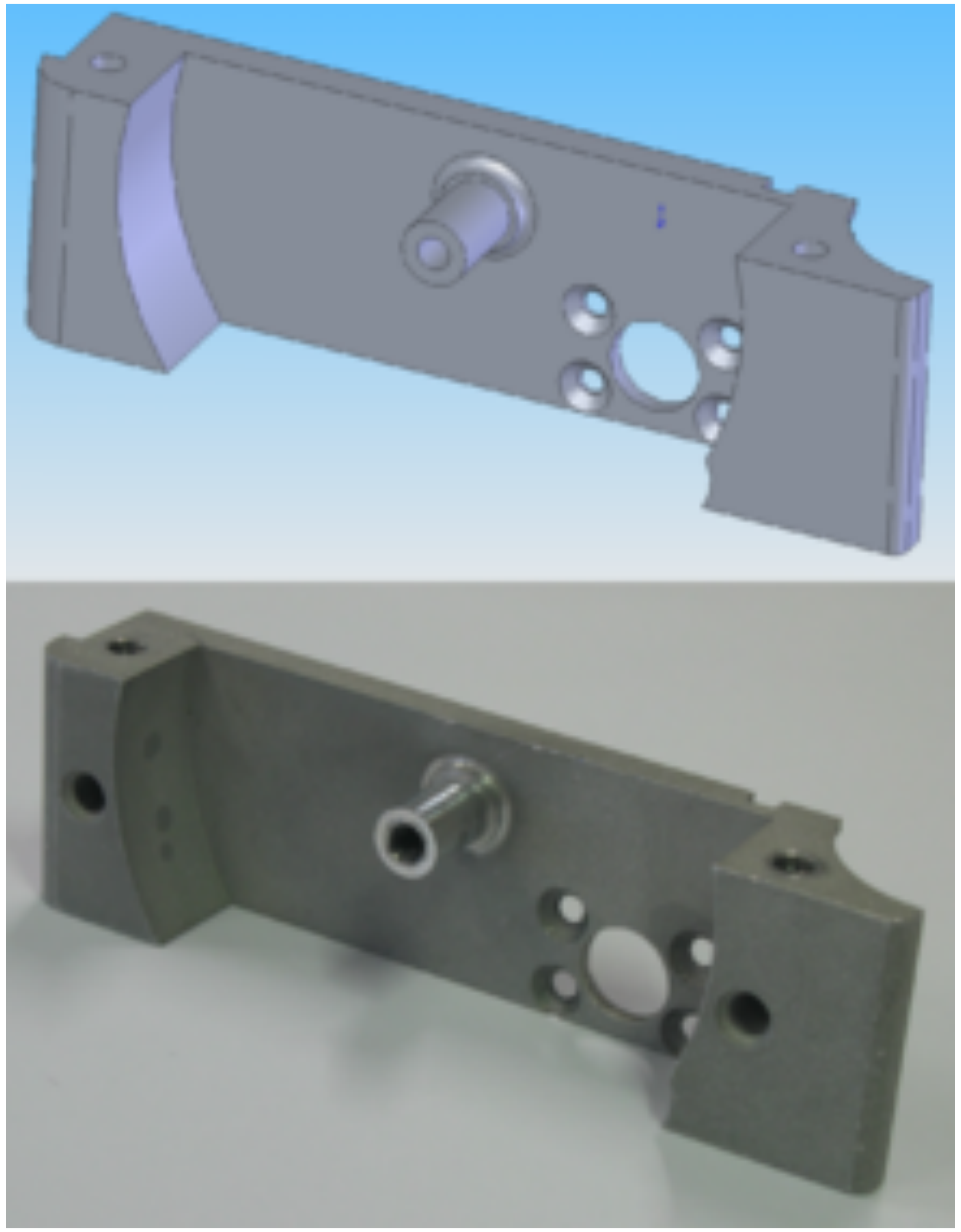
Parametric surface



Used in discipline that value *extreme* precision

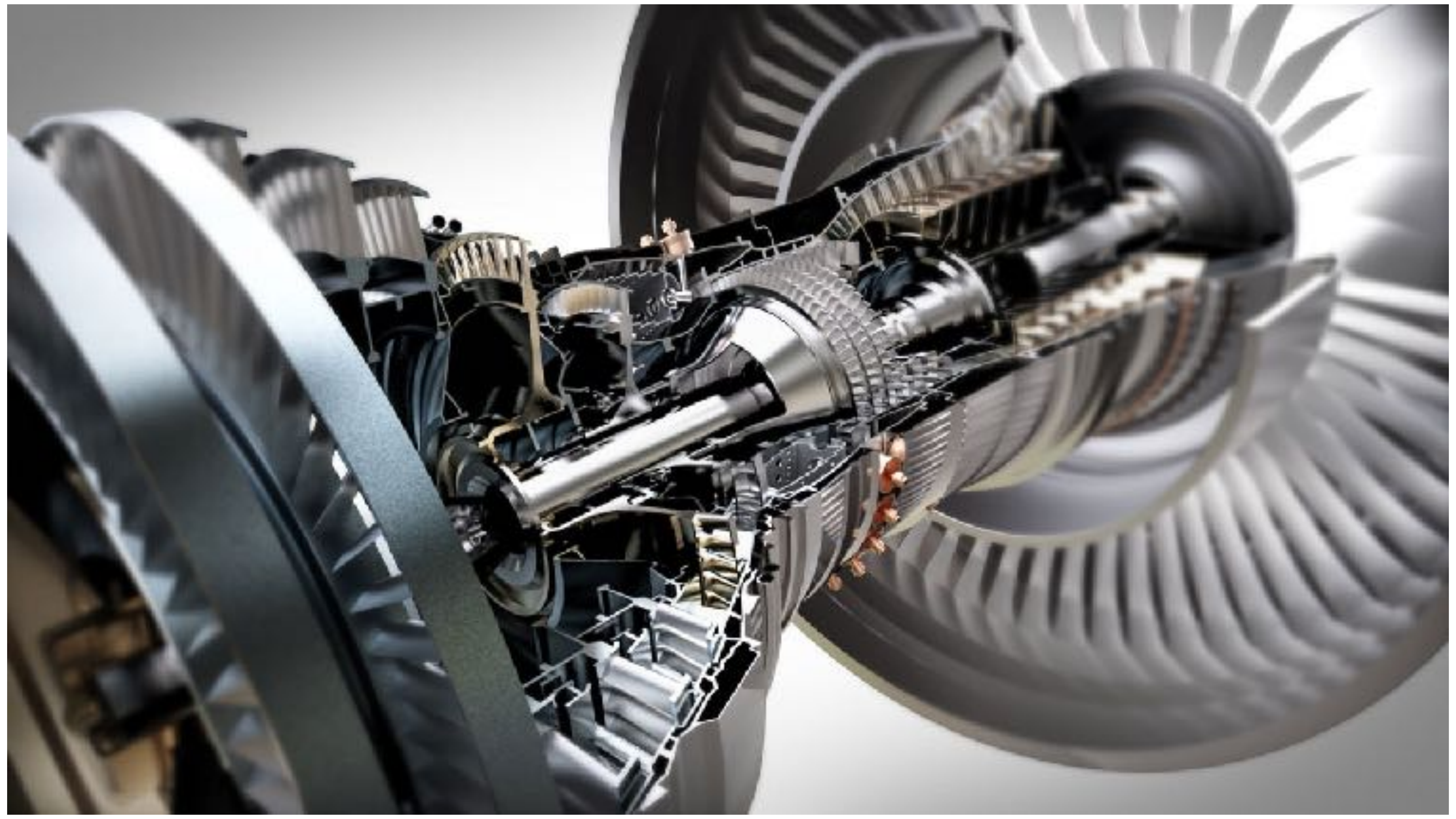


Parametric surface

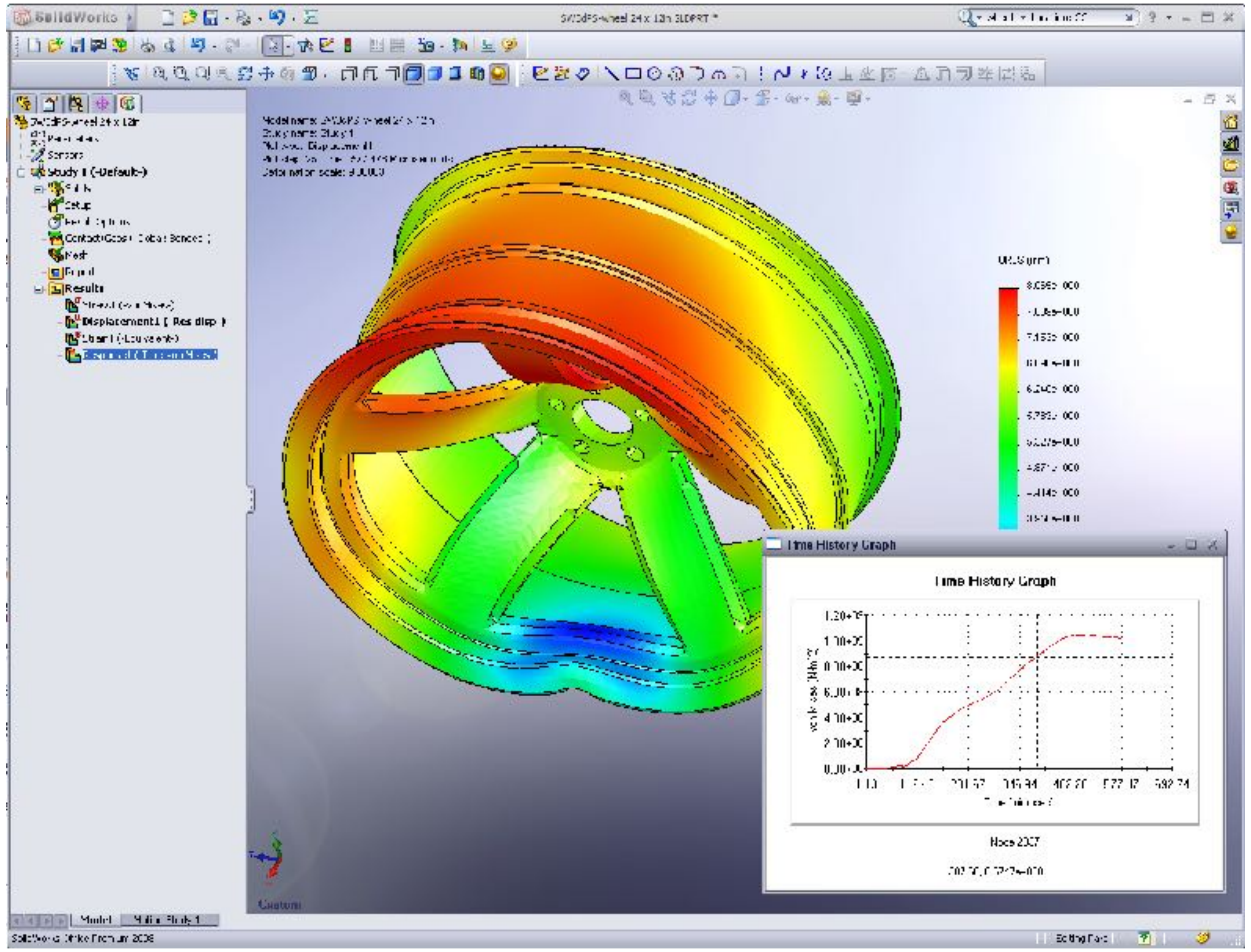


Used in discipline that value *extreme* precision

Industrial machines can fabricate them



Parametric surface



Used in discipline that value *extreme* precision

Industrial machines can fabricate them

Everything else: really complicated

Parametric surface

The Shape Matching Element Method: Direct Animation of Curved Surface Models

TY TRUSTY, University of Toronto, Canada
HONGLIN CHEN, University of Toronto, Canada
DAVID I.W. LEVIN, University of Toronto, Canada

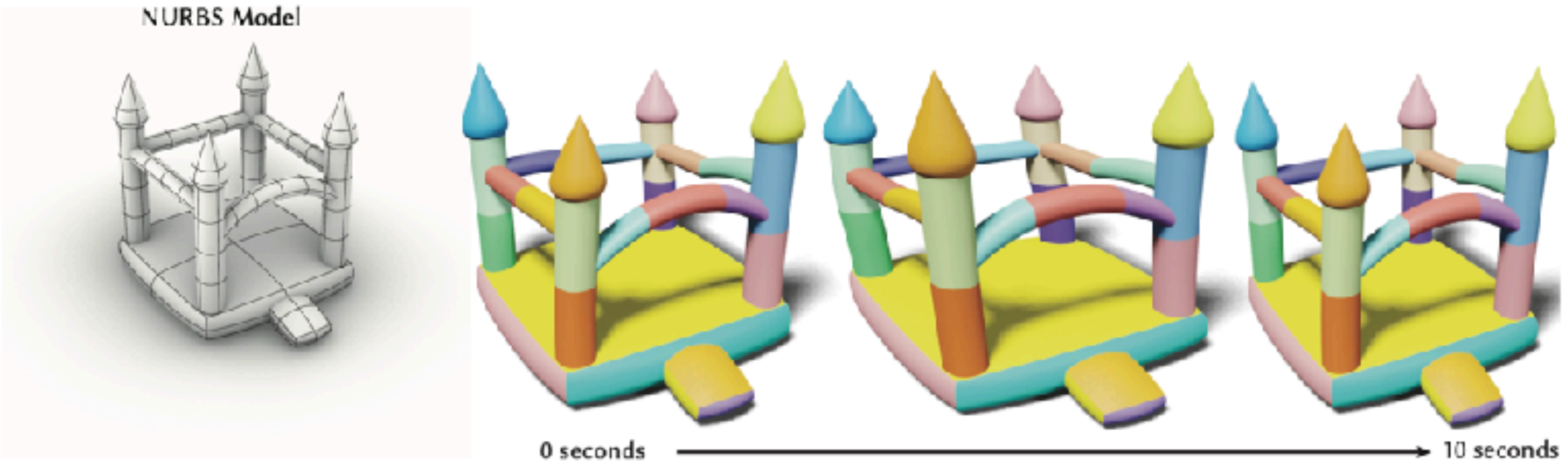


Fig. 1. Using the shape matching element method we can directly simulate this NURBS surface model of a bouncy castle as a volumetric elastic object without the need for volumetric meshing of any kind.

Used in discipline that value *extreme* precision

Industrial machines can fabricate them

Everything else: really complicated

Main research questions:

How to do *[thing we know how to do on meshes]* with parametric surfaces?

The Shape Matching Element Method: Direct Animation of Curved Surface Models

TY TRUSTY, University of Toronto, Canada
HONGLIN CHEN, University of Toronto, Canada
DAVID I.W. LEVIN, University of Toronto, Canada

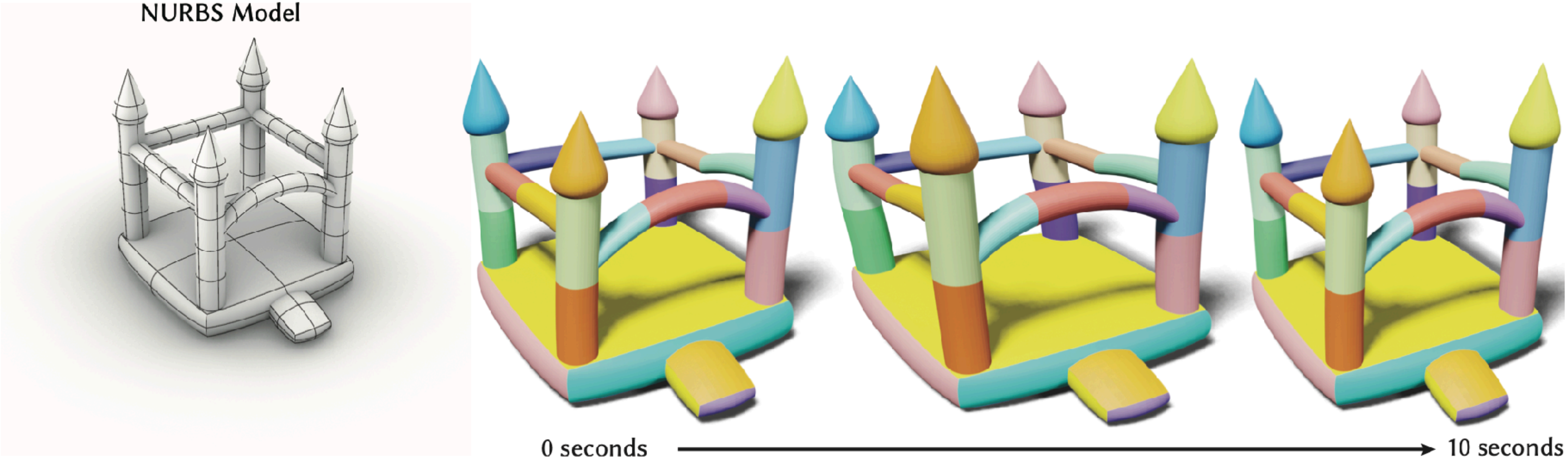
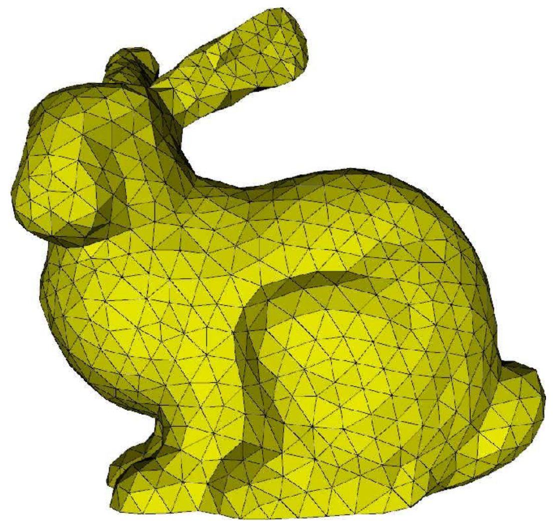


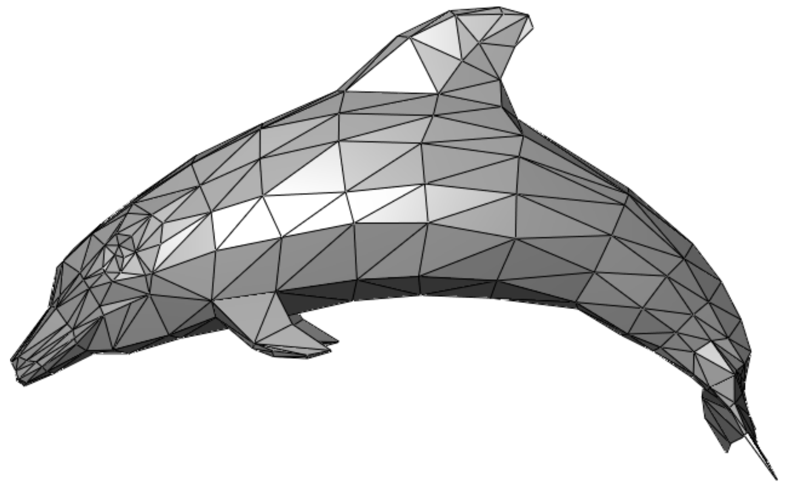
Fig. 1. Using the shape matching element method we can directly simulate this NURBS surface model of a bouncy castle as a volumetric elastic object without the need for volumetric meshing of any kind.

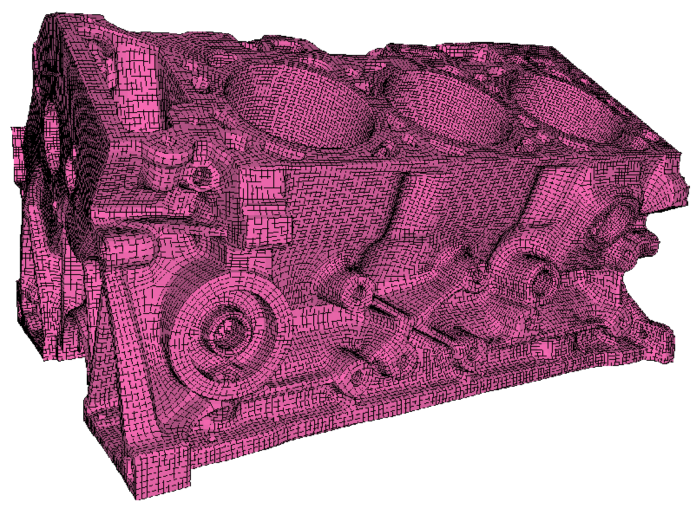
SHAPE REPRESENTATIONS IN 3D

Point cloud

Meshes

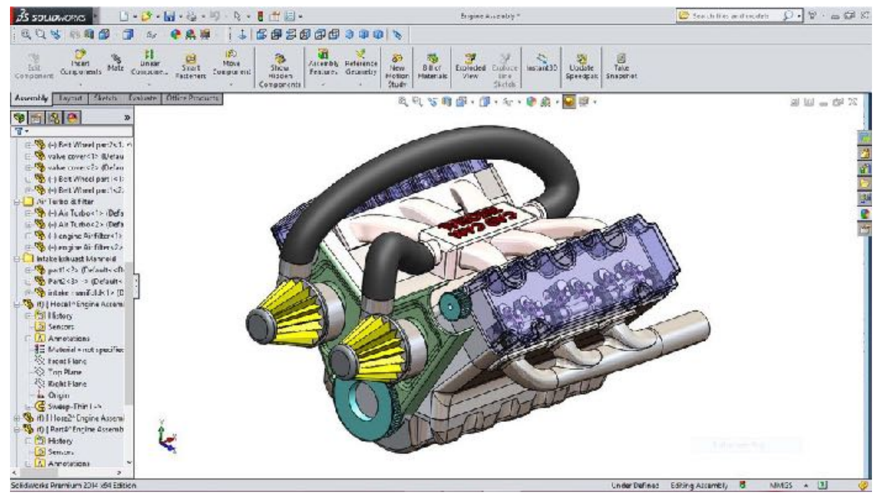


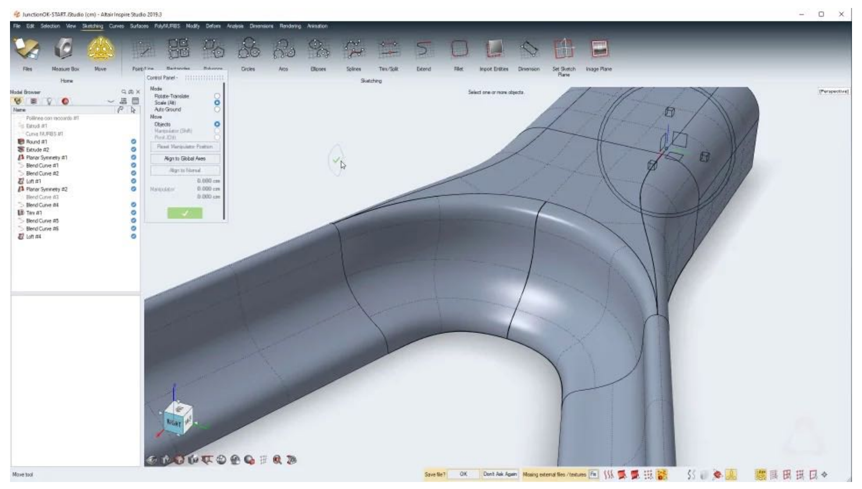


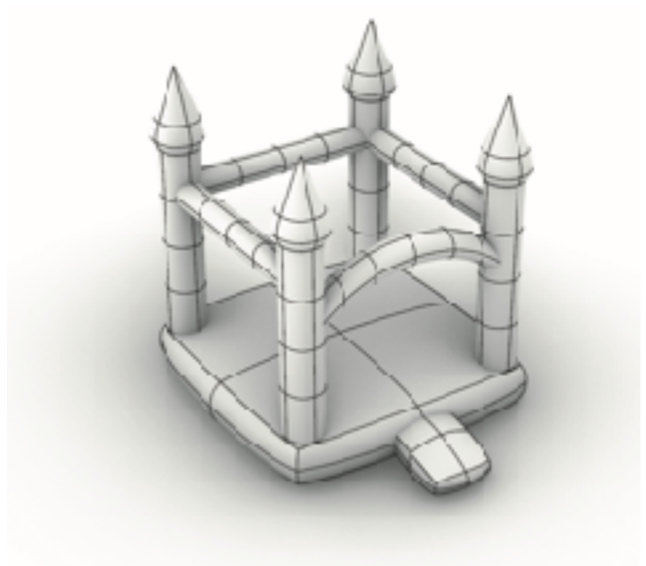


Implicit

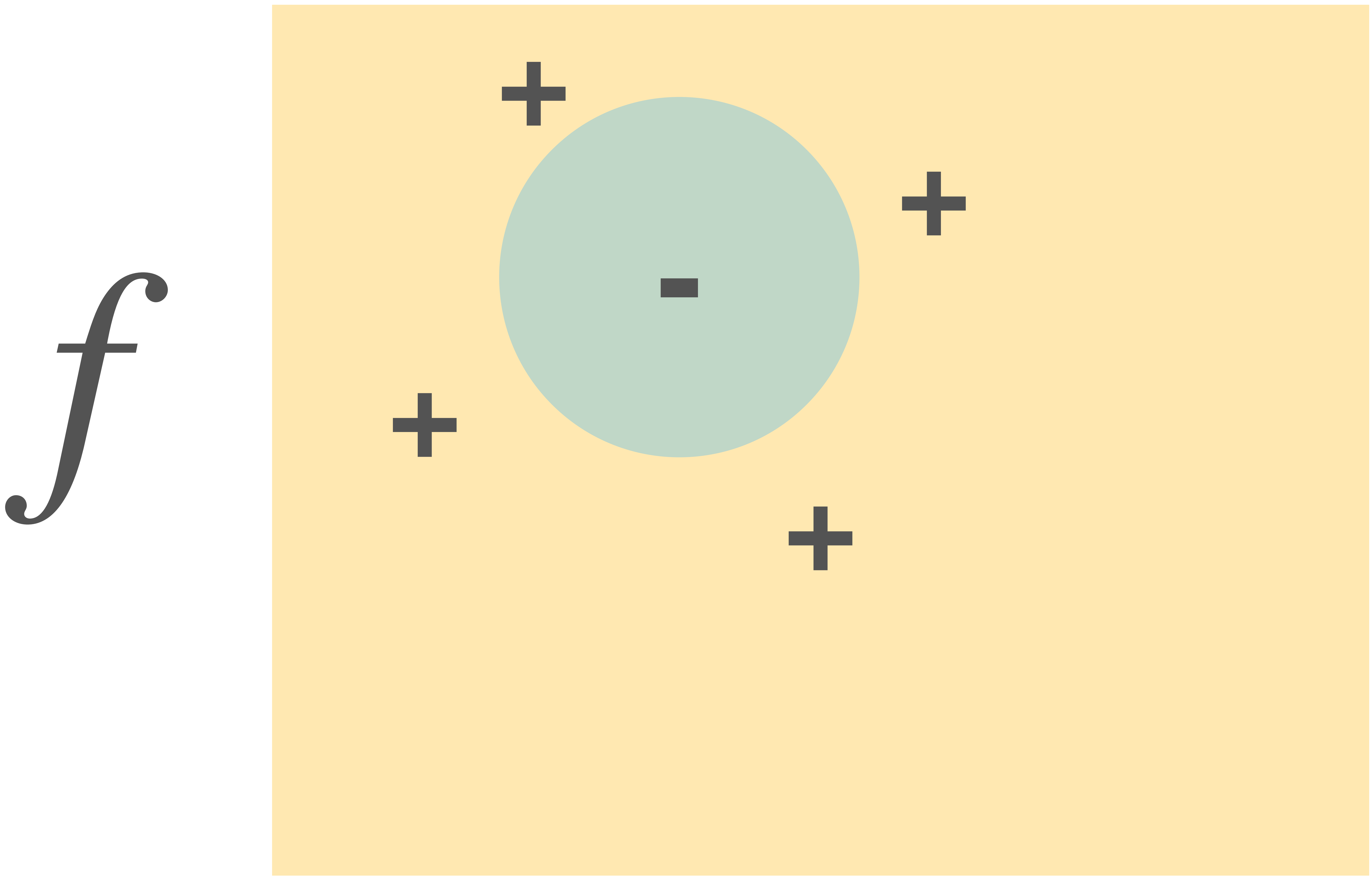
Parametric surface







Implicit surface



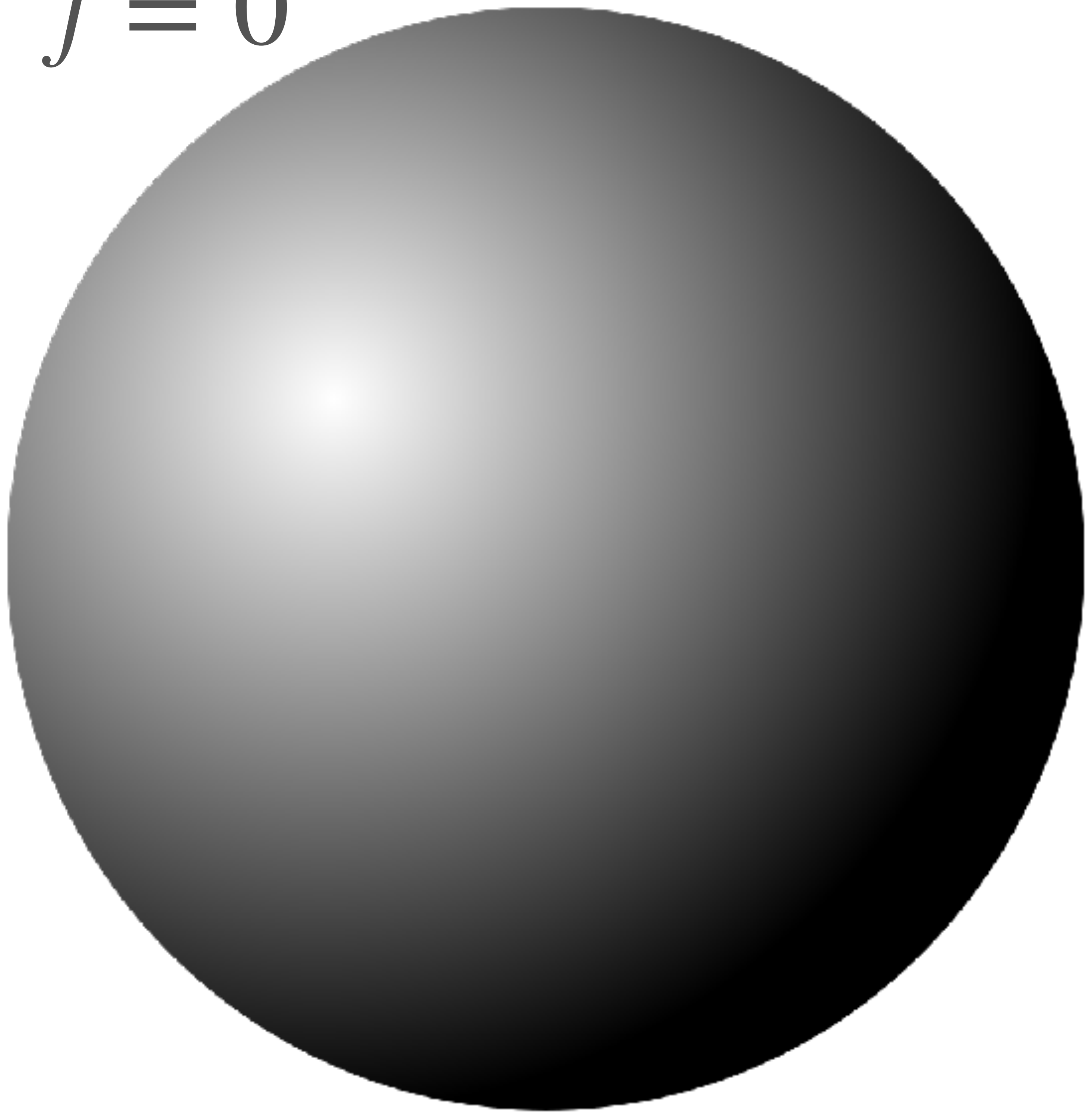
Implicit surface

$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

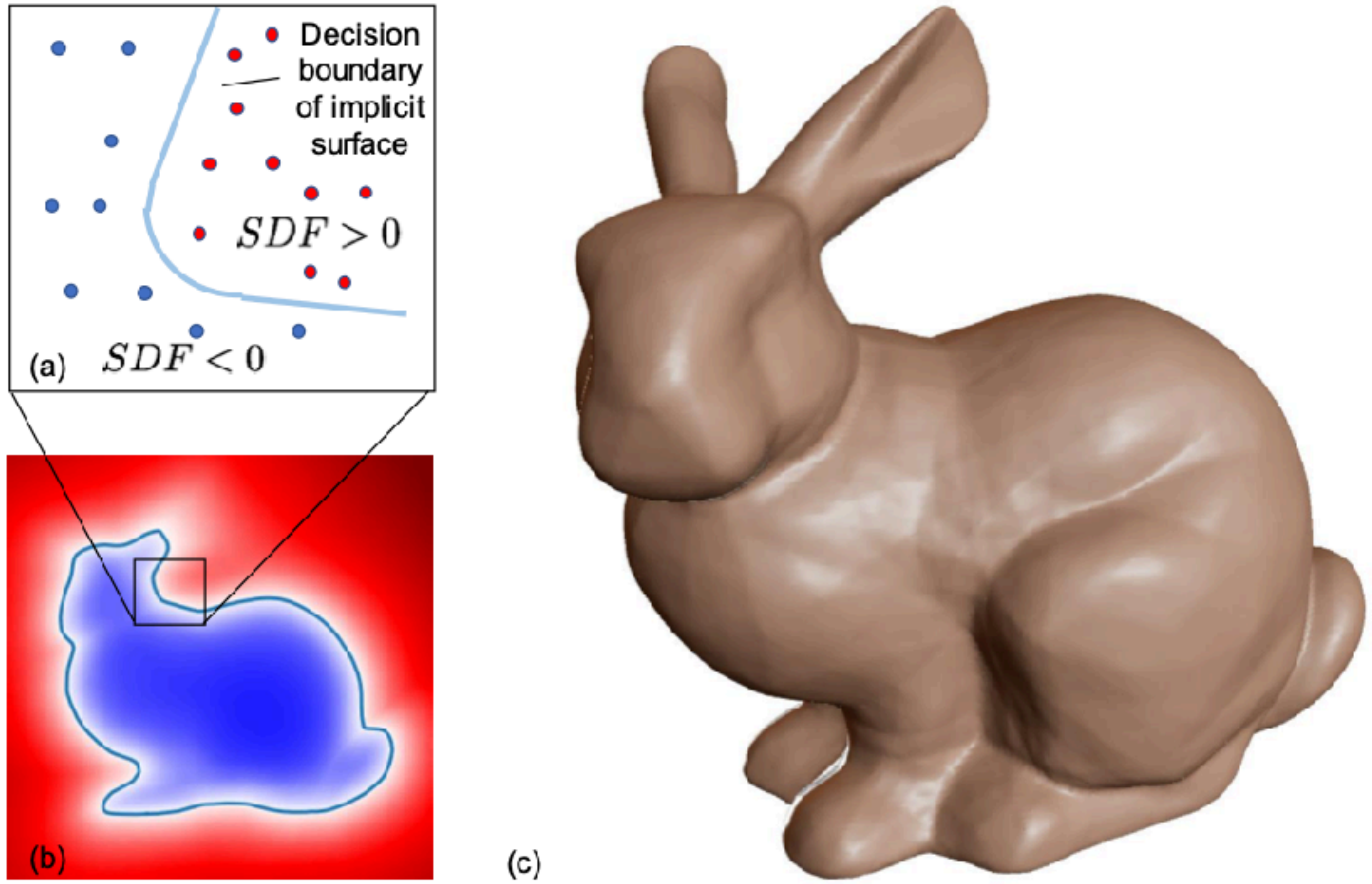
Implicit surface

$$f(x, y, z) = x^2 + y^2 + z^2 - 1$$

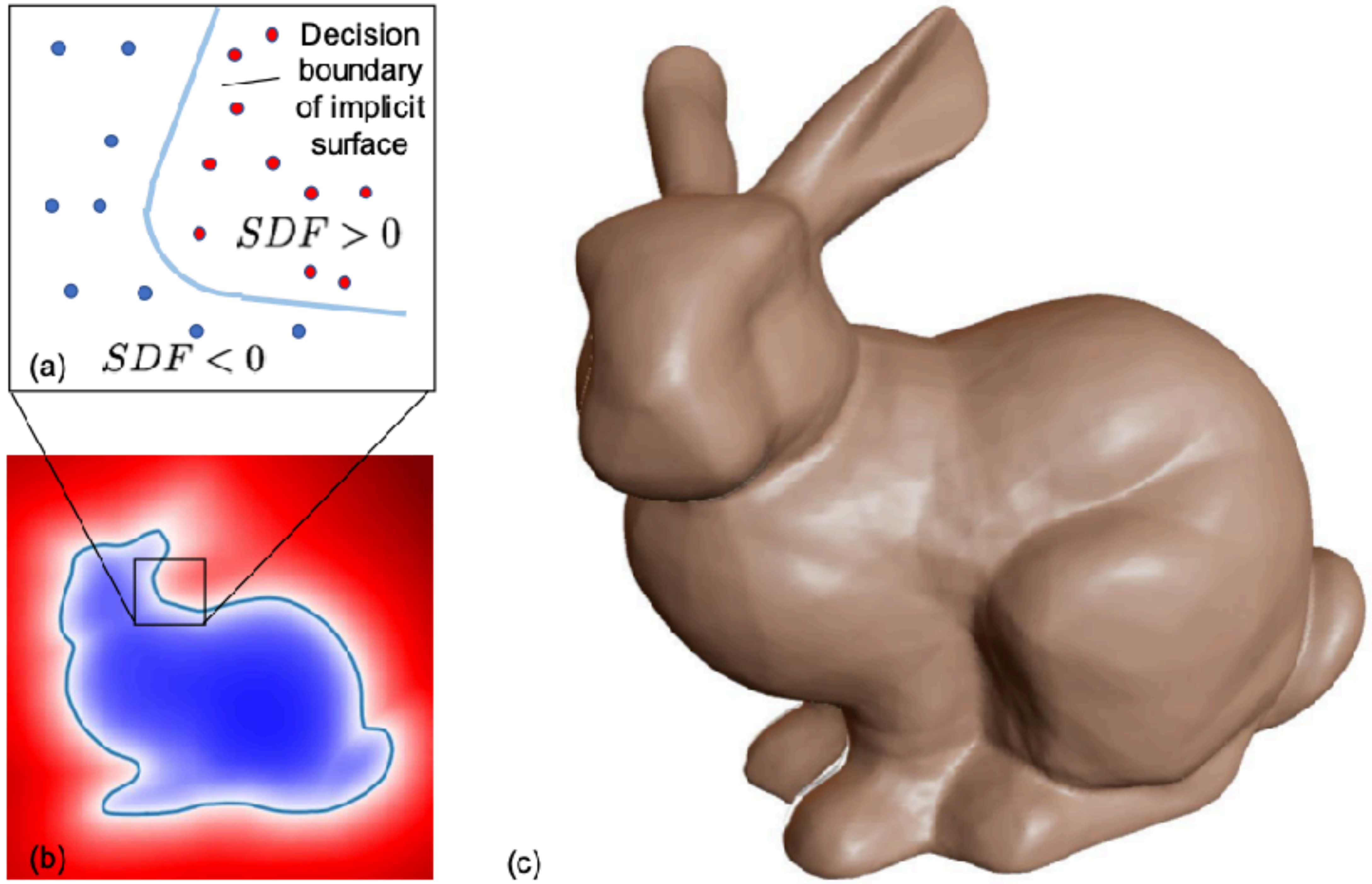
$f = 0$



Implicit surface



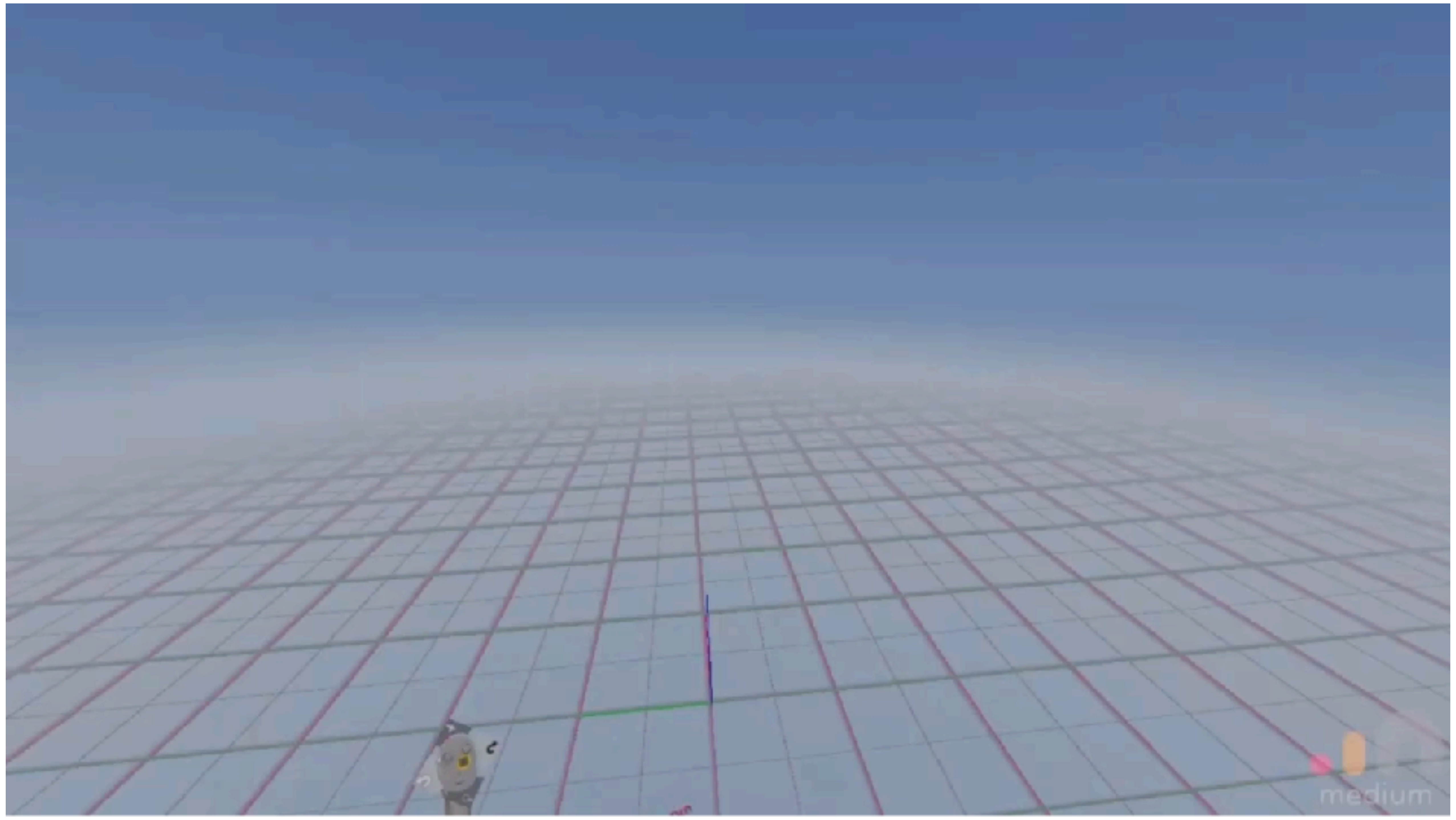
Implicit surface



- ✔ Use *machine learning*
- ✔ Easy boolean operations
- ✘ …almost everything else

Implicit surface

Used when fast boolean operations are necessary



Implicit surface

Used when fast boolean operations are necessary

Implicit surface

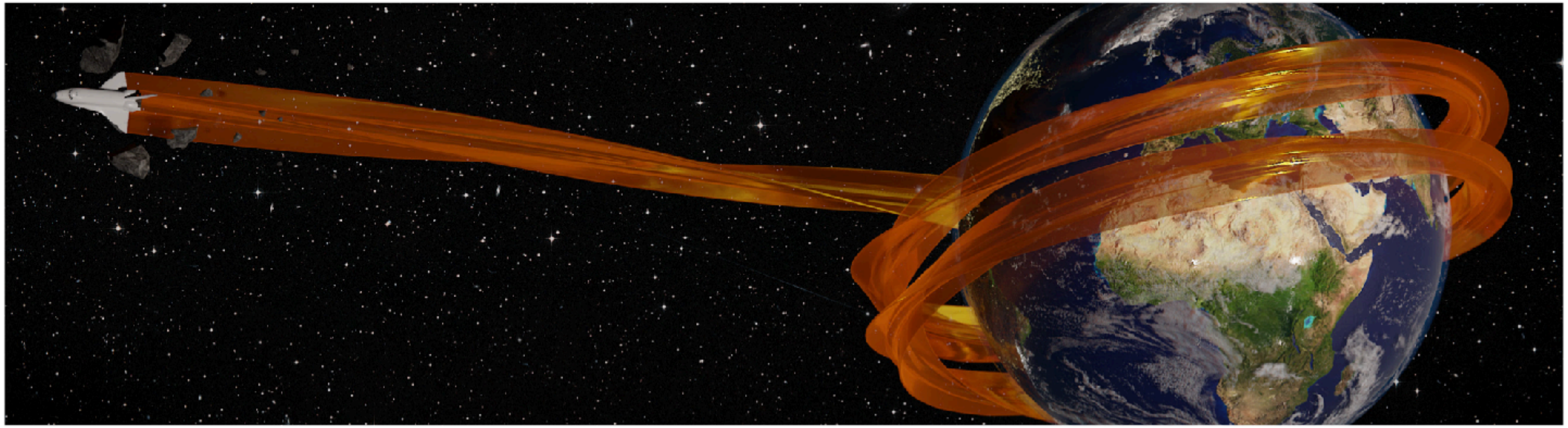
Used when fast boolean operations are necessary

Implicit surface

Used when fast boolean operations are necessary

Swept Volumes via Spacetime Numerical Continuation

SILVIA SELLÁN, University of Toronto
NOAM AIGERMAN, Adobe Research
ALEC JACOBSON, University of Toronto and Adobe Research



Implicit surface

Used when fast boolean operations are necessary

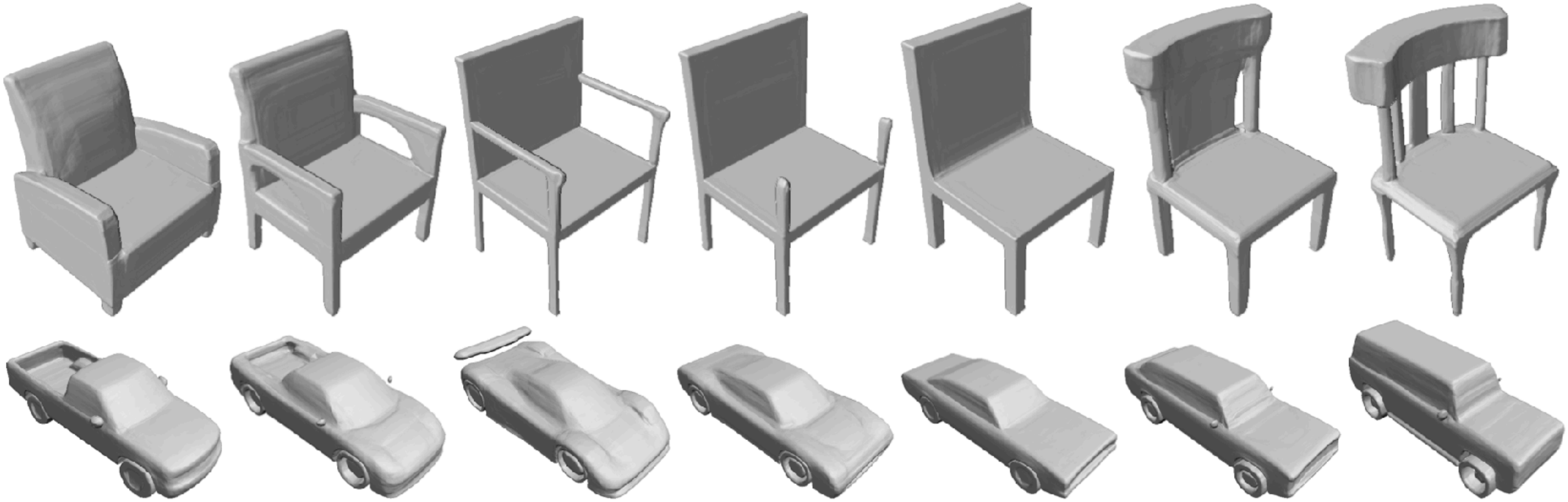
Used in machine learning applications

Implicit surface

DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation

Jeong Joon Park[1,3†] Peter Florence [2,3†] Julian Straub[3] Richard Newcombe[3] Steven Lovegrove[3]

[1]University of Washington [2]Massachusetts Institute of Technology [3]Facebook Reality Labs



Implicit surface

Used when fast boolean operations are necessary

Used in machine learning applications

Research questions:

Implicit surface

Used when fast boolean operations are necessary

Used in machine learning applications

Research questions:

How to render an implicit?

Massively Parallel Rendering of Complex Closed-Form Implicit Surfaces

Matthew J. Keeter, independent researcher

ACM Transactions on Graphics (Proceedings of SIGGRAPH), 2020

Figure 1: An assortment of implicit surfaces rendered using our technique. Left: an extruded text string, rotated and rendered as a heightmap. Center: a bear head sculpted using smooth blending operations, with normals found by automatic differentiation. Right: a complex architectural model rendered with screen-space ambient occlusion and perspective. All models are rendered directly from their mathematical representations, without triangulation or raytracing.

Implicit surface

Used when fast boolean operations are necessary

Used in machine learning applications

Research questions:

- How to render an implicit?
- How to deform an implicit?

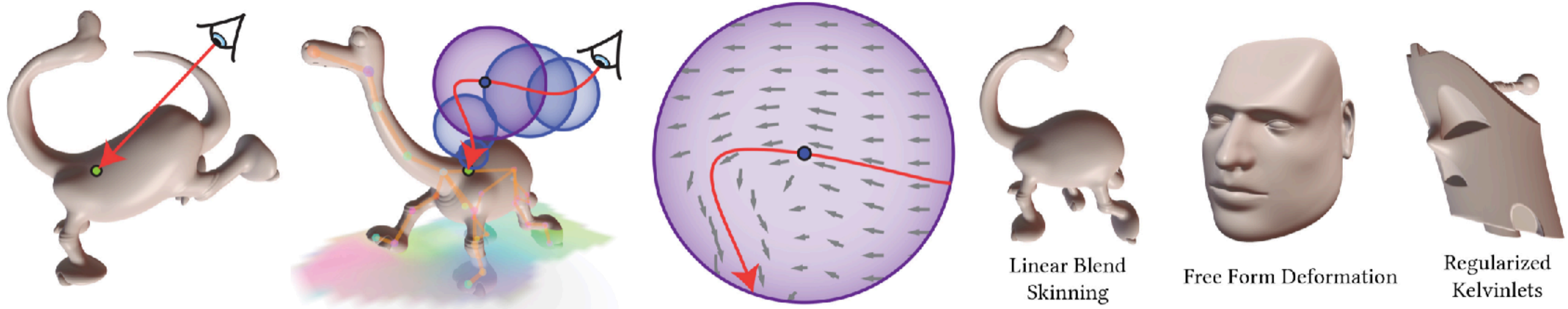
Implicit surface

Non-linear sphere tracing for rendering deformed signed distance fields

Dario Seyb[1] Alec Jacobson[2] Derek Nowrouzezahrai[3] Wojciech Jarosz[1]

[1]Dartmouth College [2]University of Toronto [3]McGill University

In *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 2019



(a) Deformed Space (b) Undeformed Space (c) Initial Value Problem (d) Example Deformations

?

Implicit surface

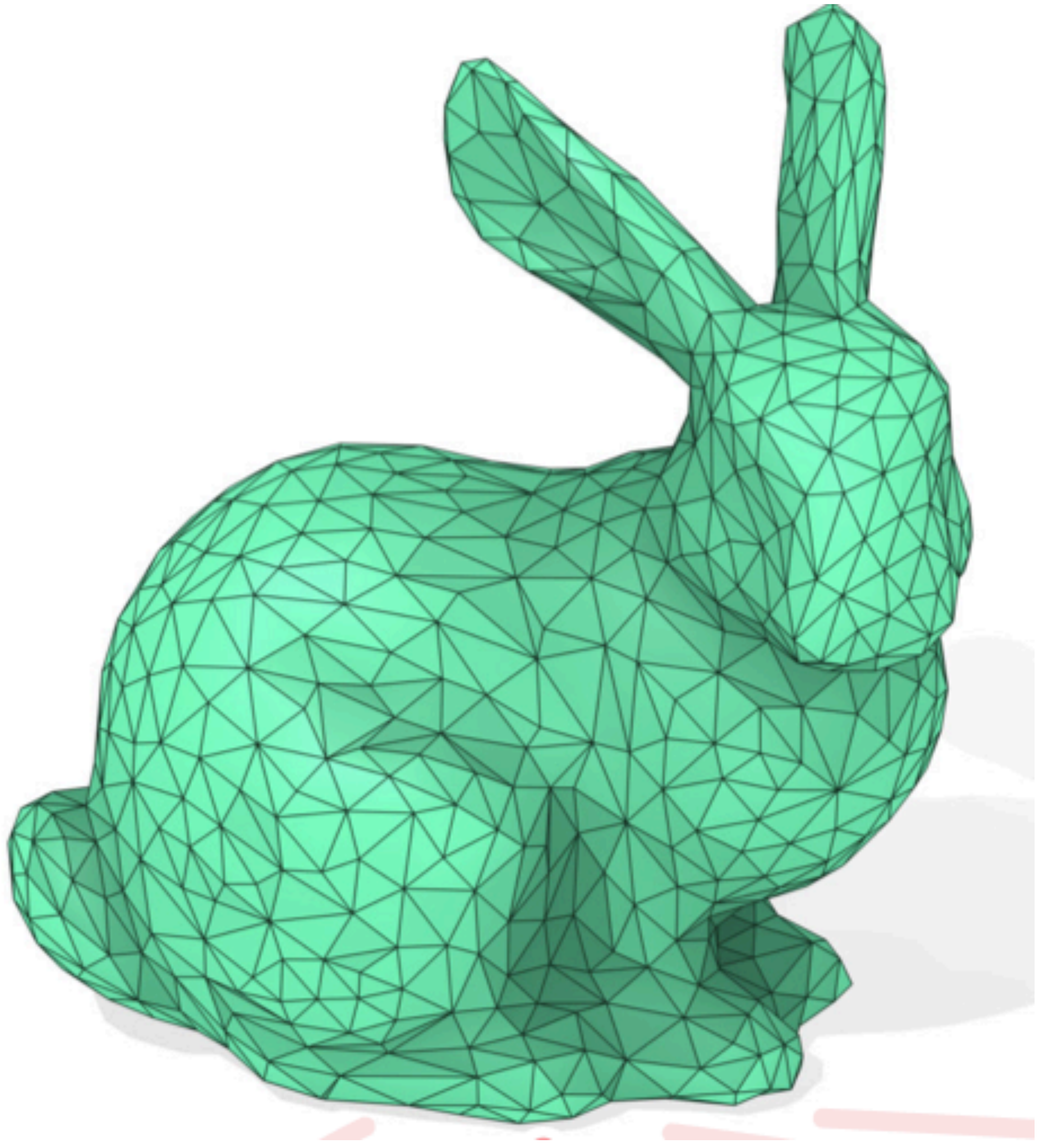
Used when fast boolean operations are necessary

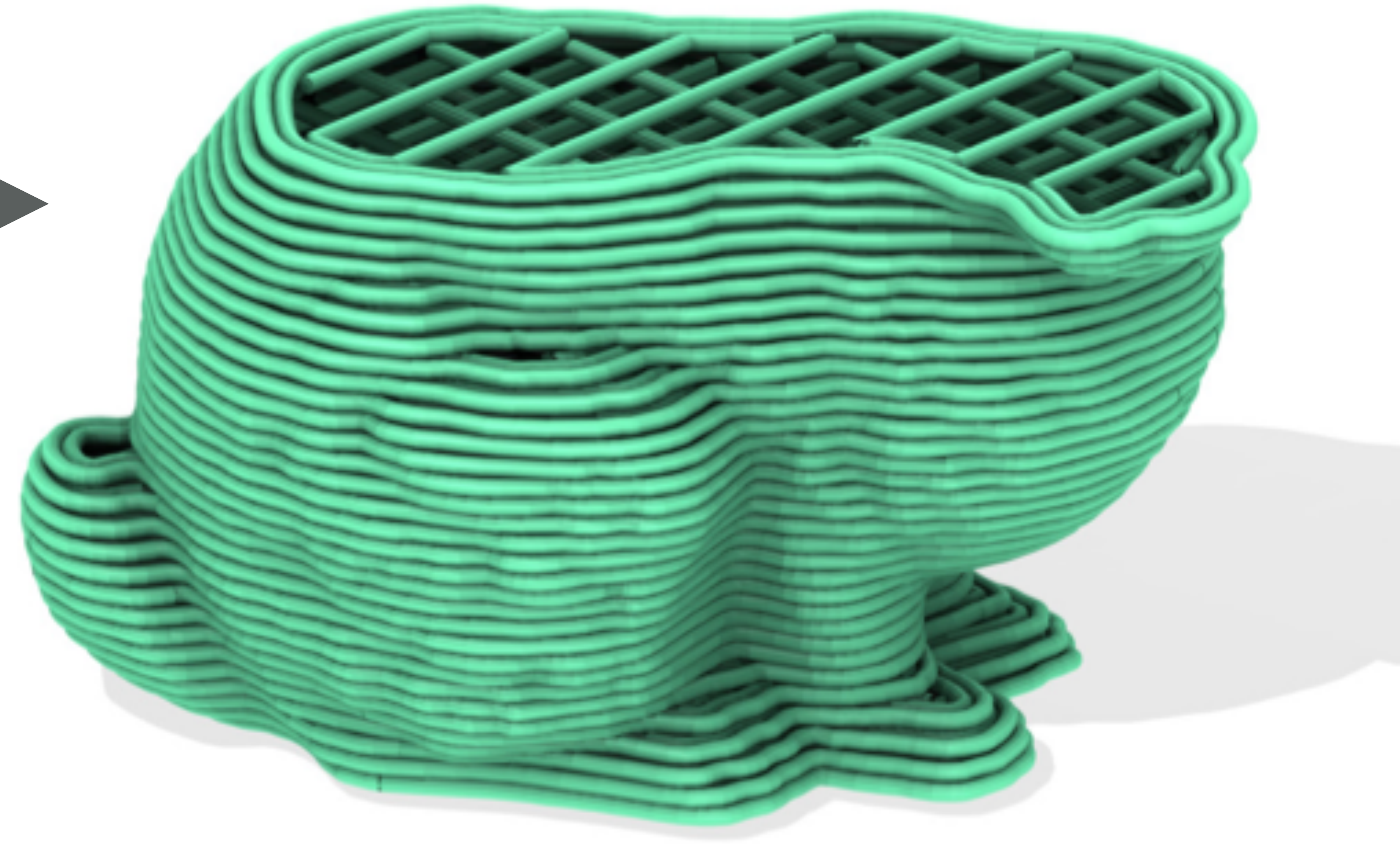
Used in machine learning applications

Research questions:

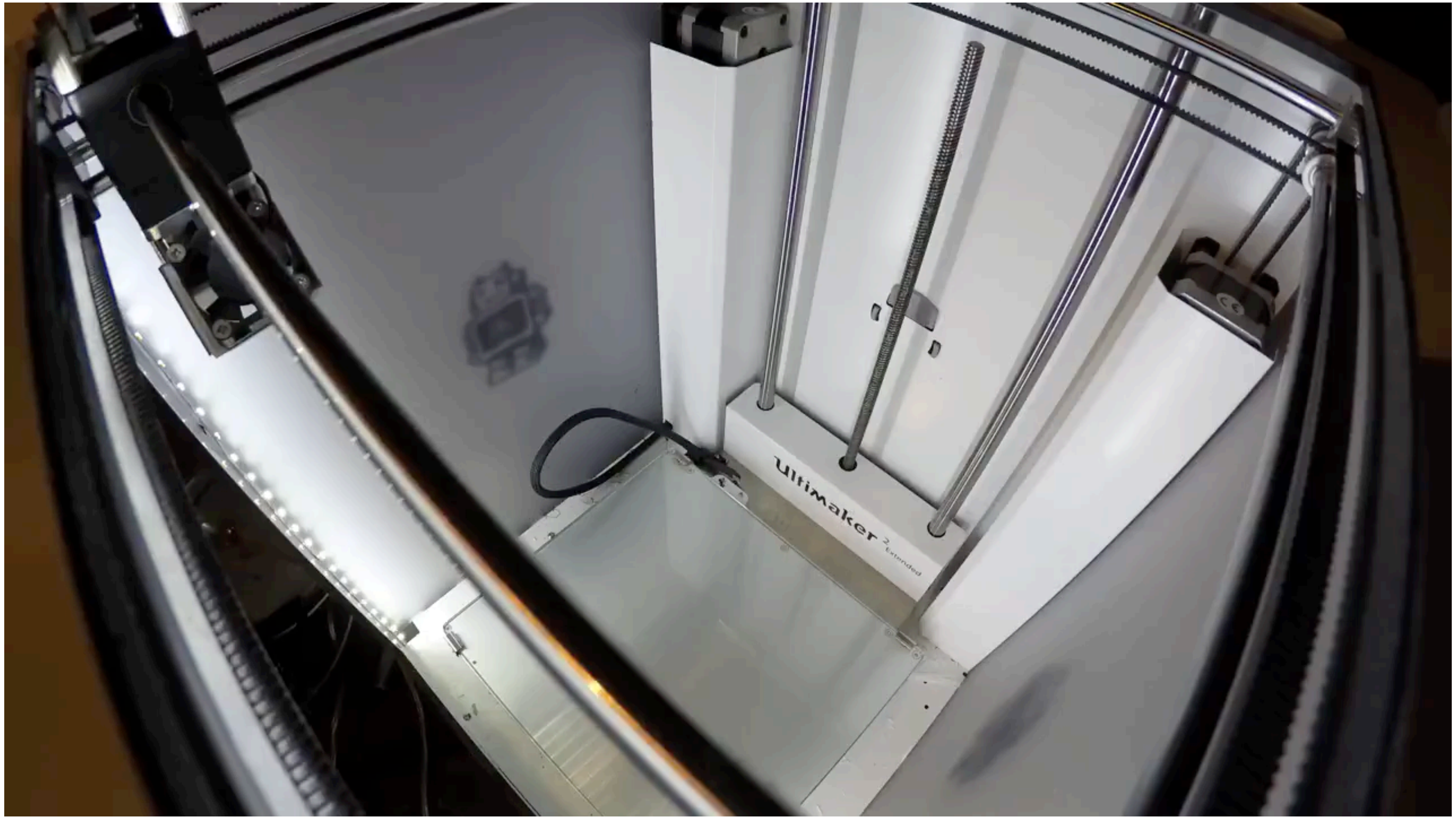
- How to render an implicit?
- How to deform an implicit?
- How to repair an implicit?

3D printing uses implicit shapes (even if they don't want you to know)





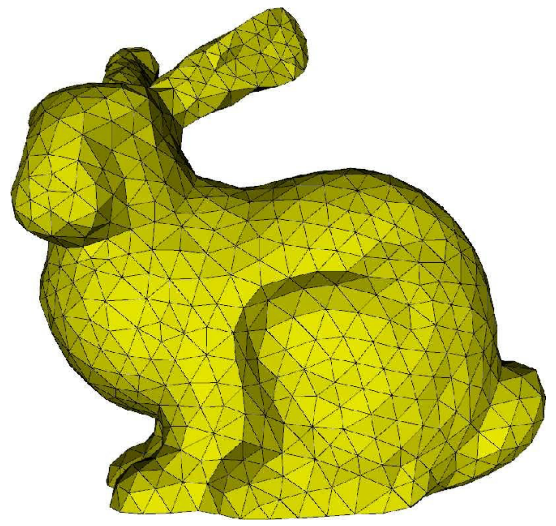
3D printing uses implicit shapes (even if they don’t want you to know)

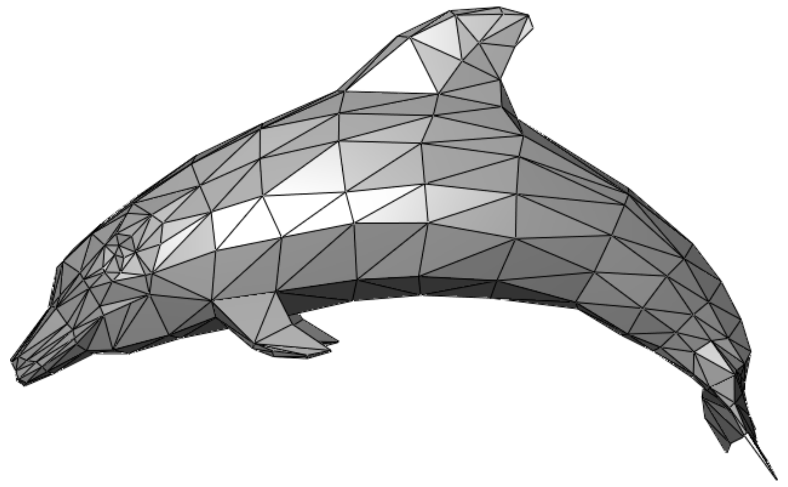


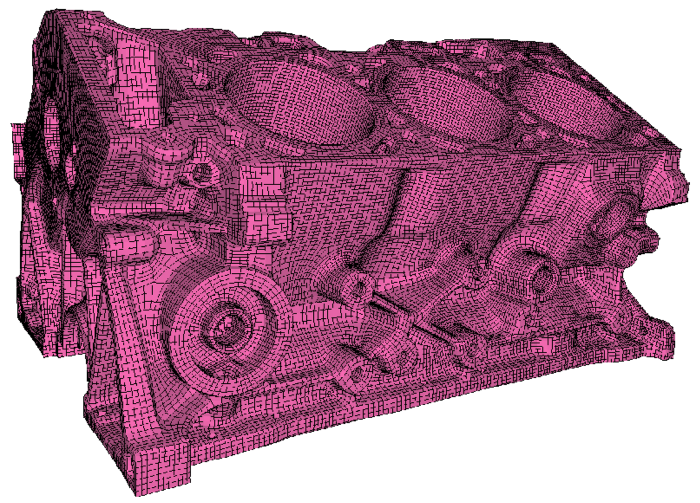
SURFACE REPRESENTATIONS IN 3D

Point cloud

Meshes

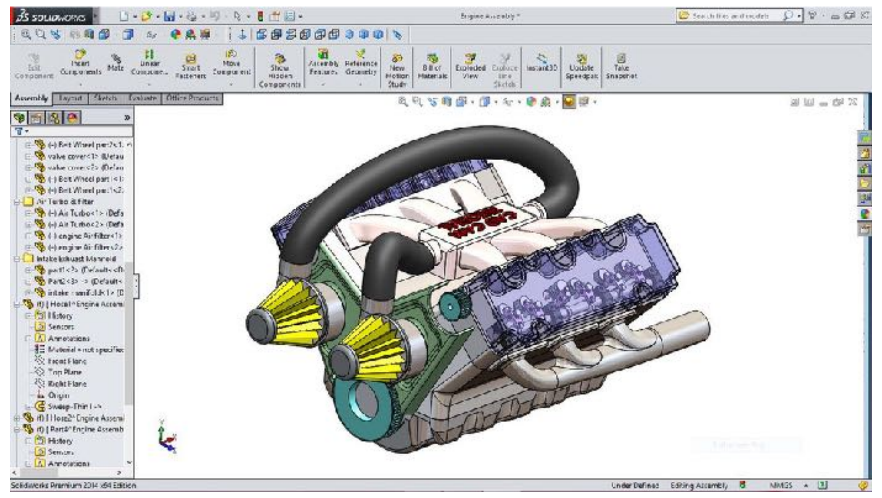


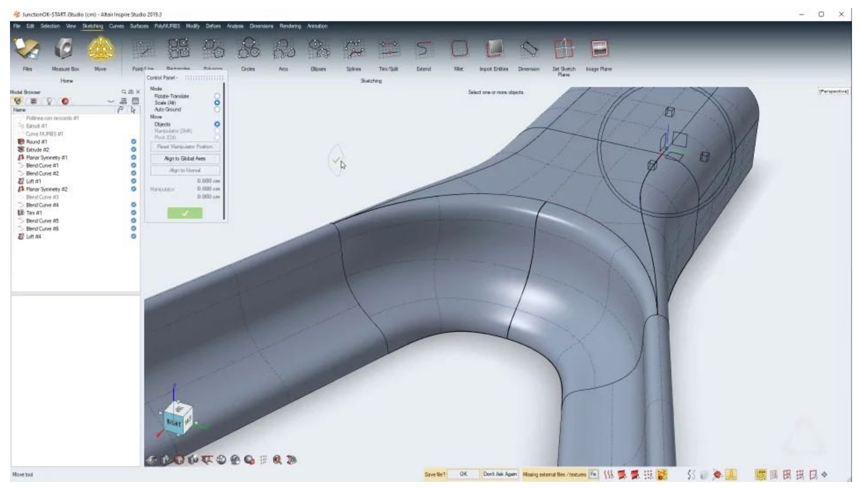


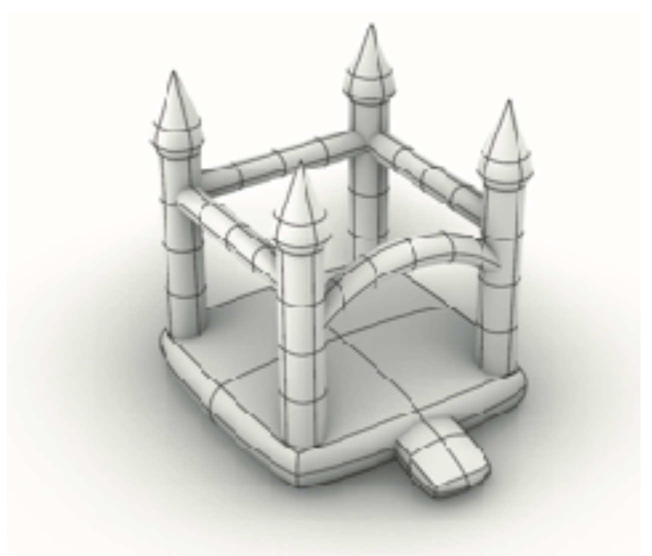


Implicit

Parametric surface

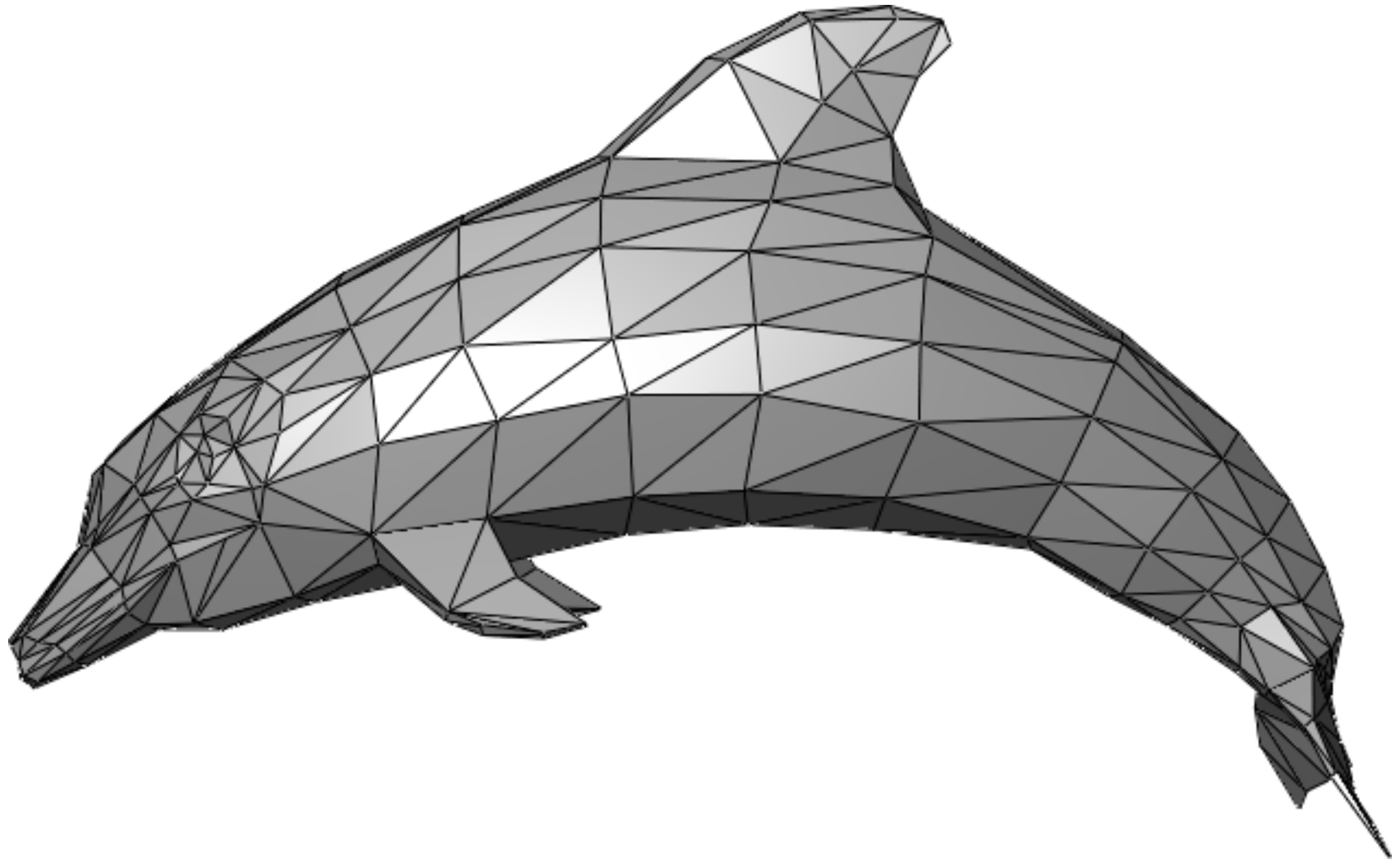




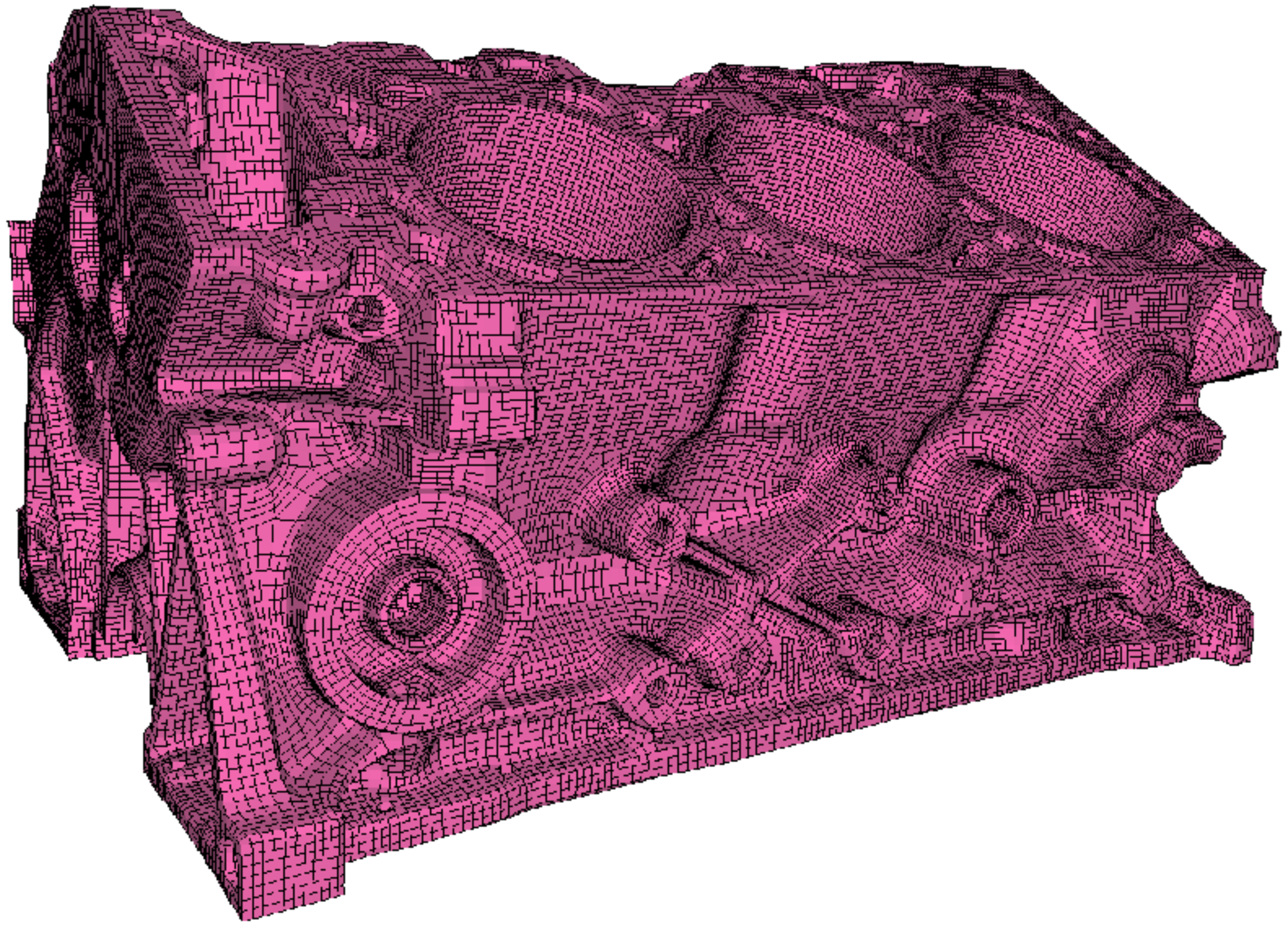


VOLUMETRIC REPRESENTATIONS IN 3D

Surface mesh

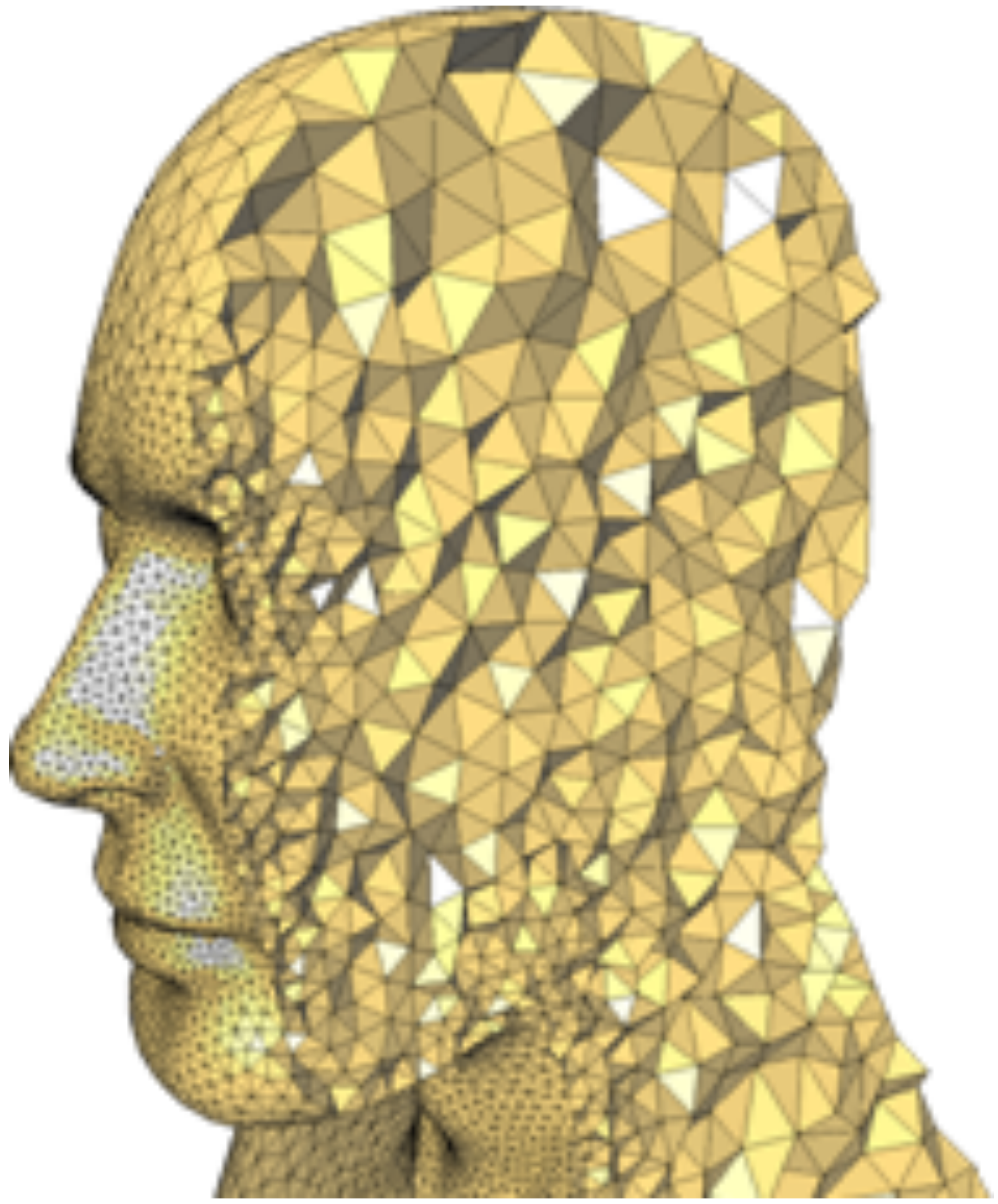


Triangle mesh

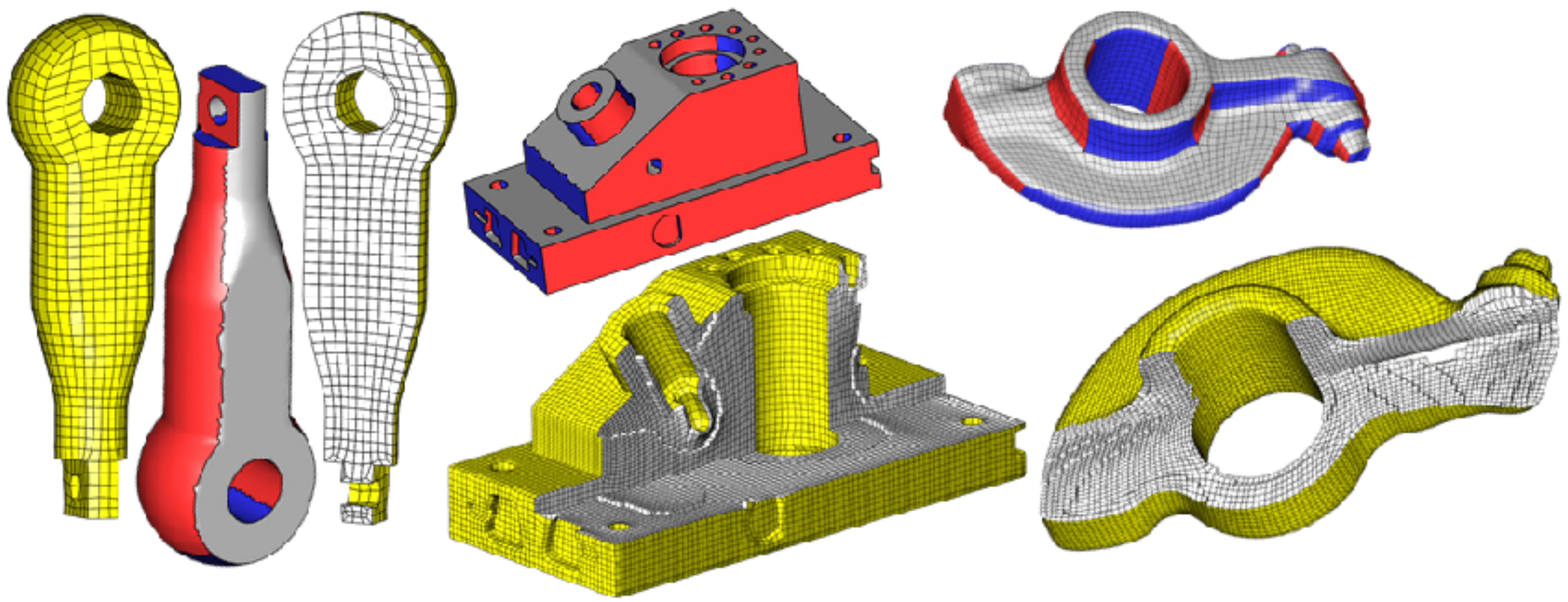


Quad mesh

Volume mesh



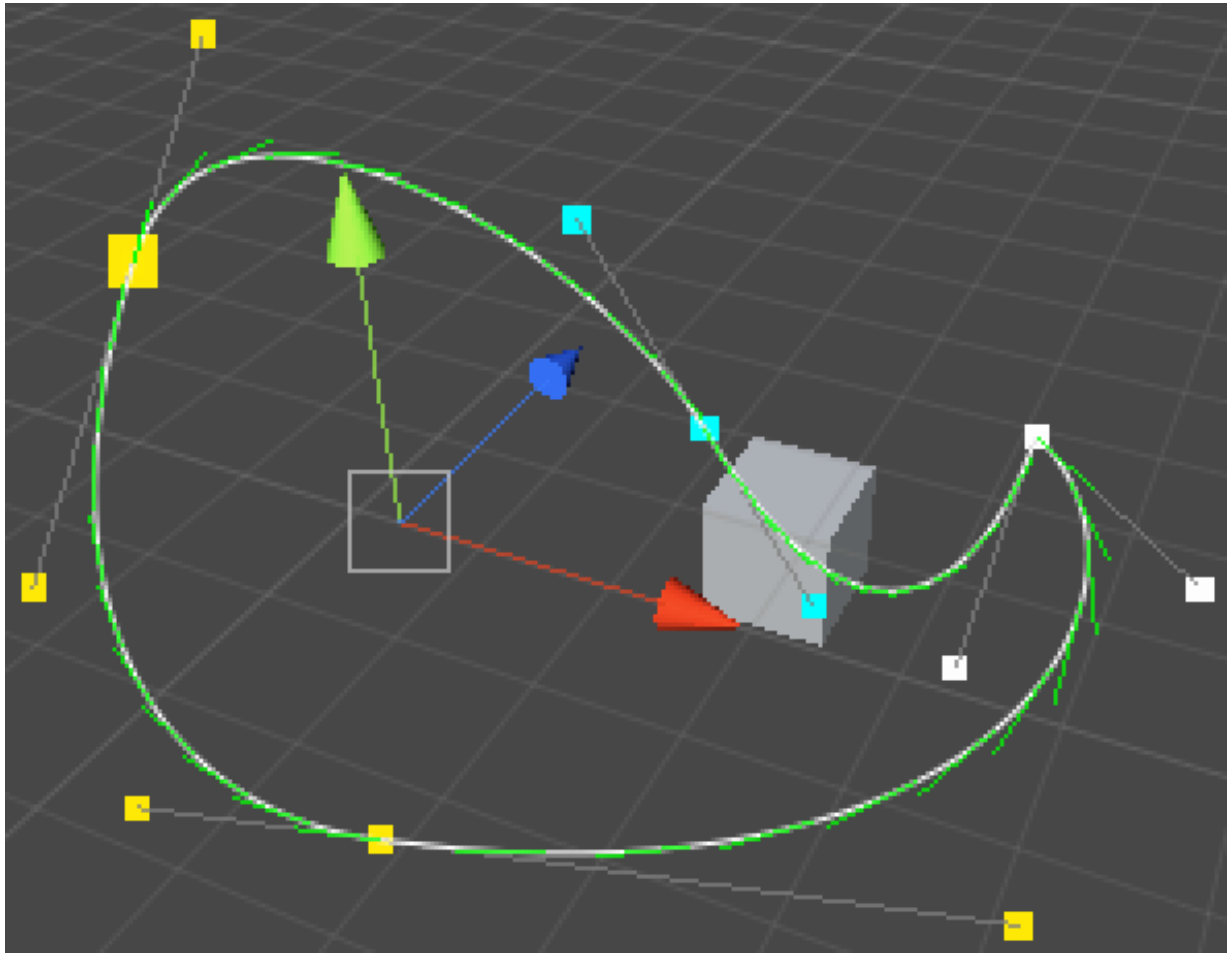
Tetrahedral mesh



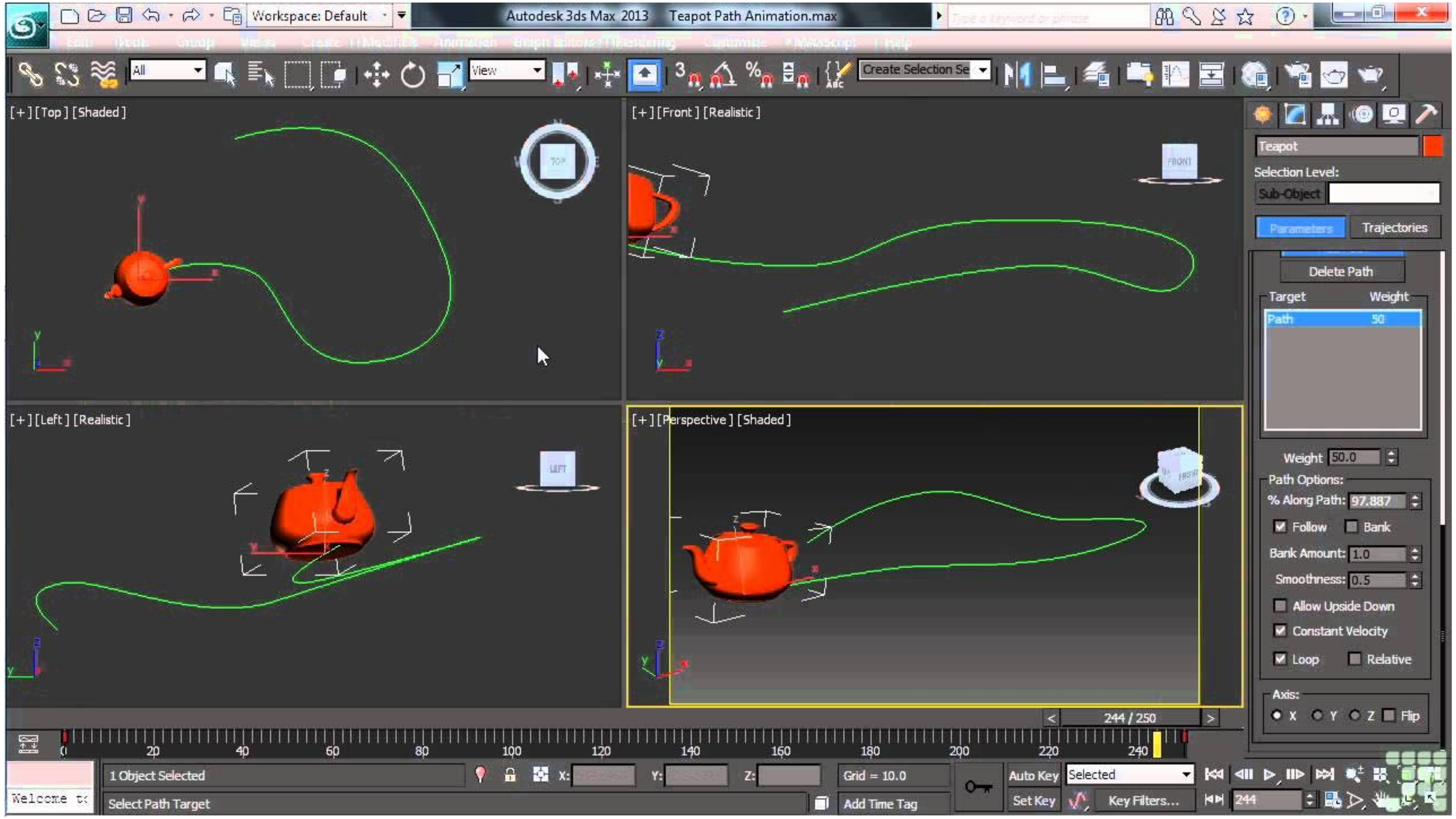
Hexahedral mesh

CURVES IN 3D

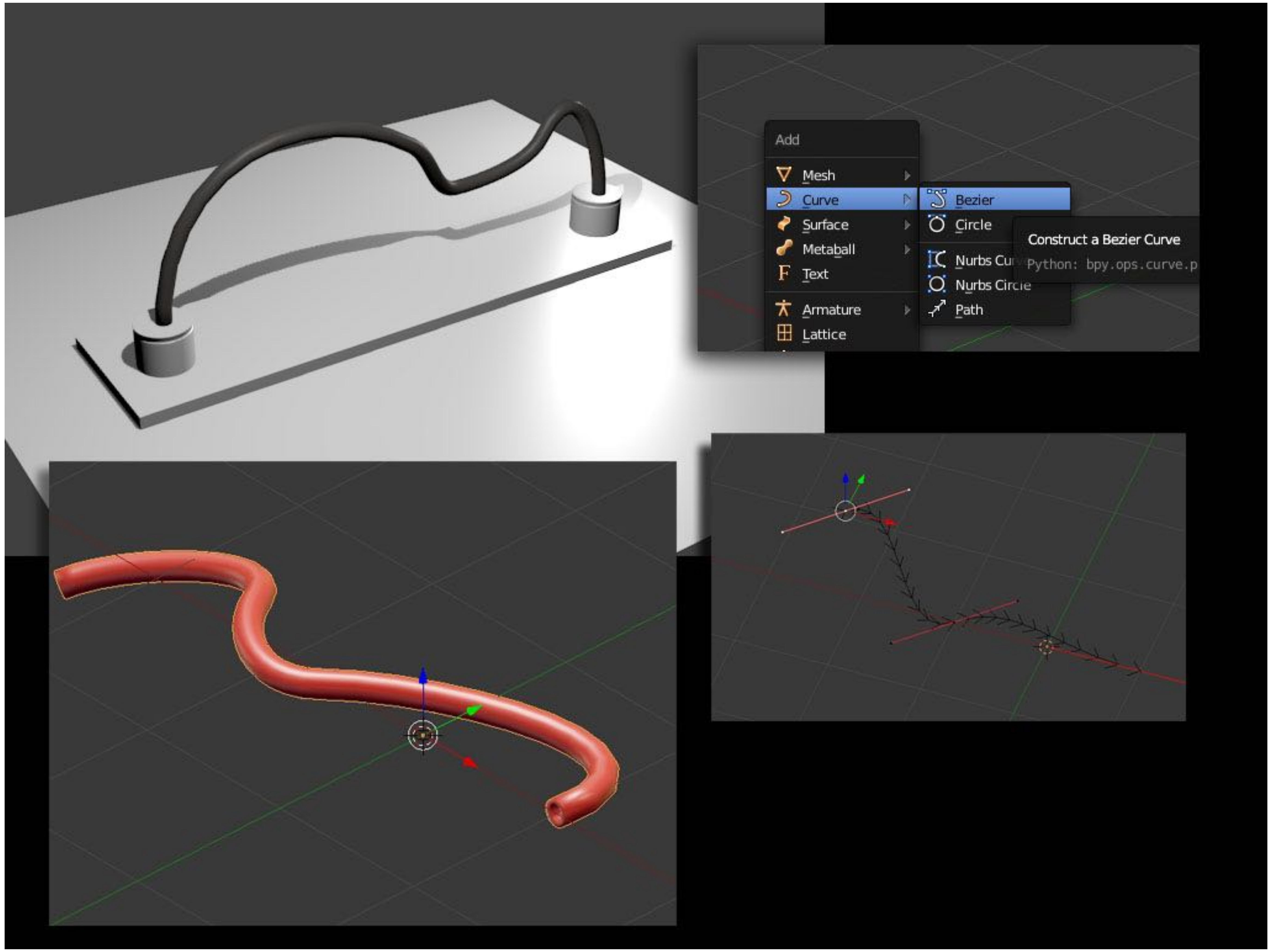
SPLINES IN 3D



SPLINES IN 3D



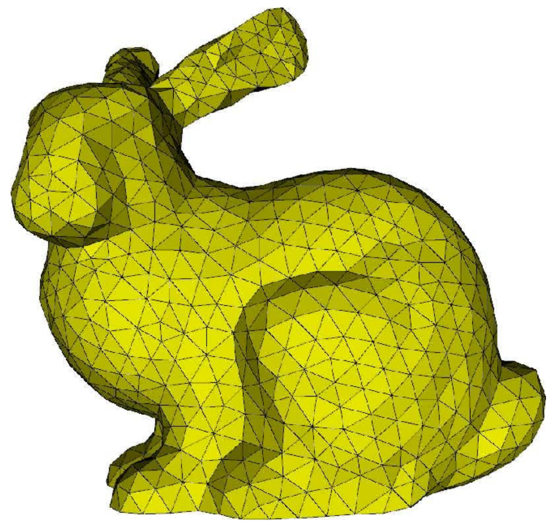
SPLINES IN 3D

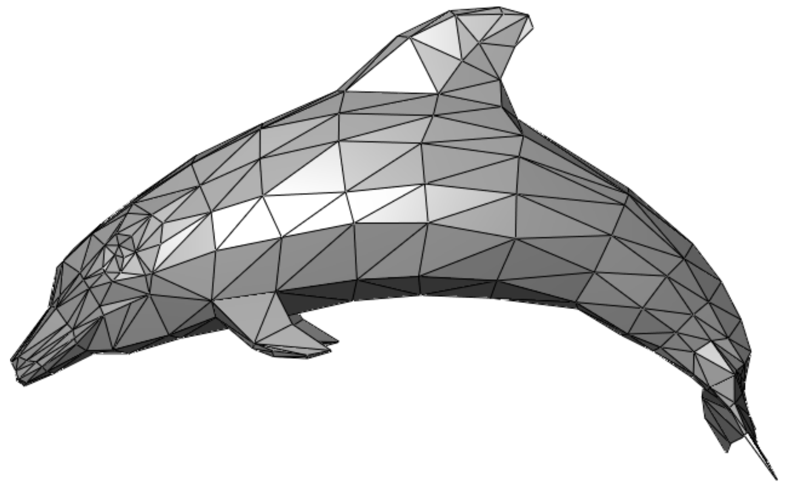


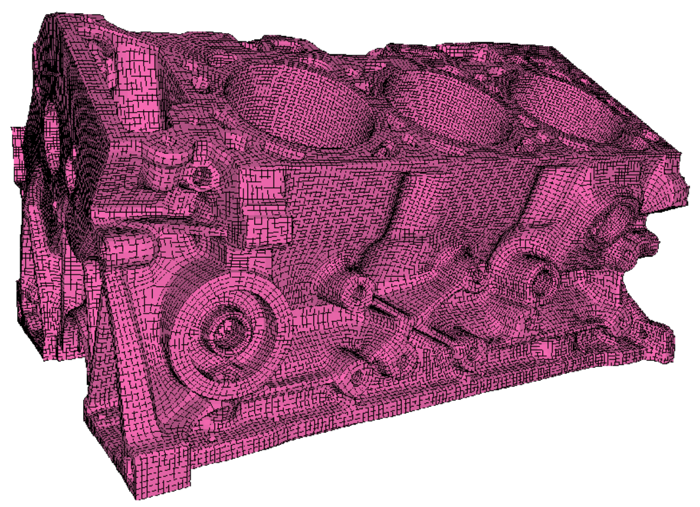
SURFACE REPRESENTATIONS IN 3D

Point cloud

Meshes

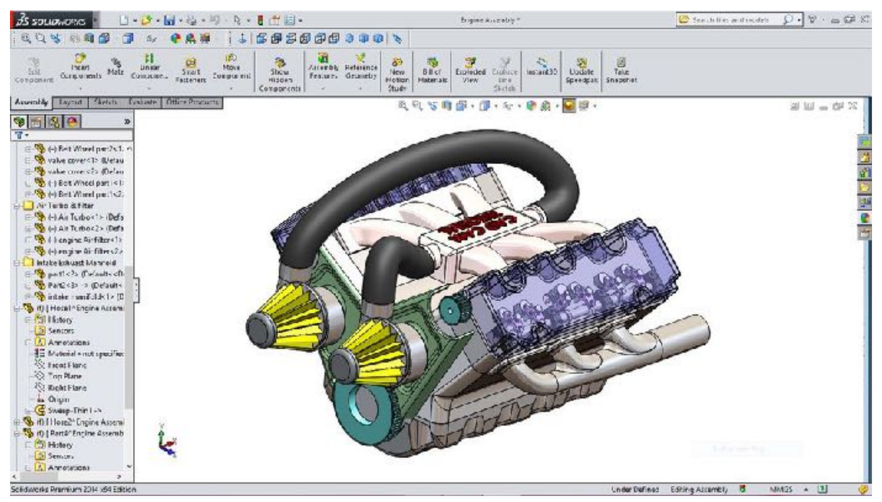


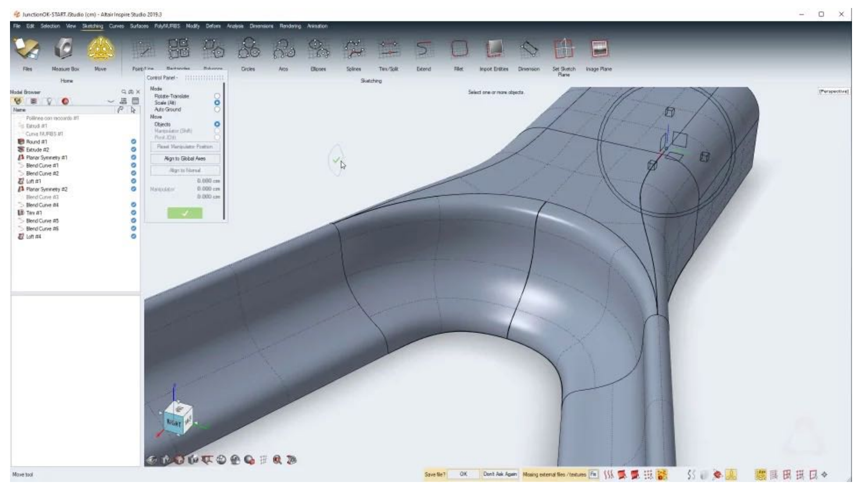


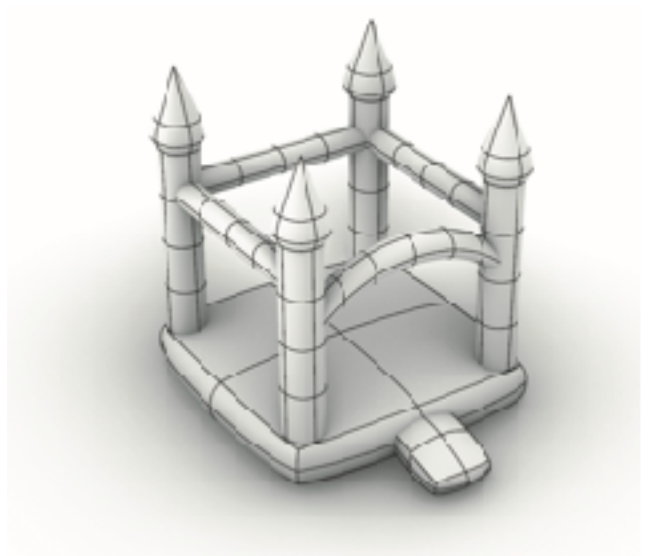


Implicit

Parametric surface



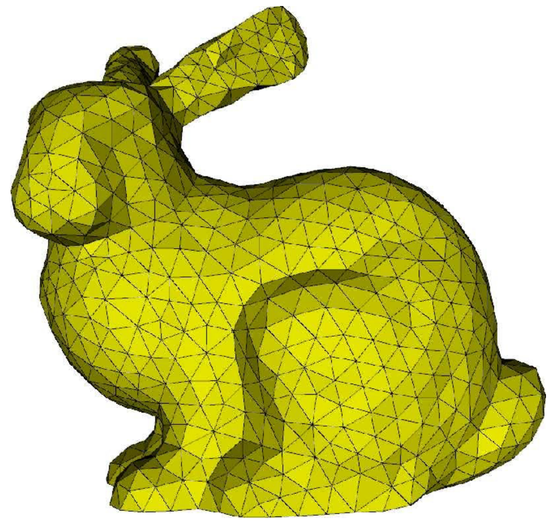


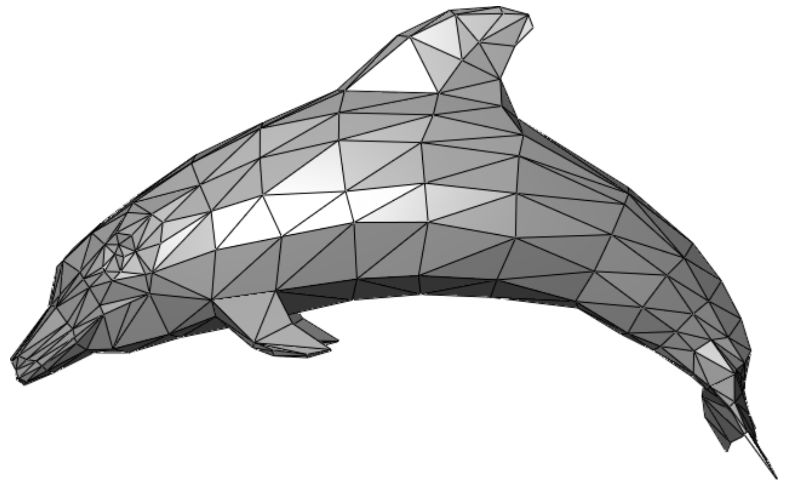


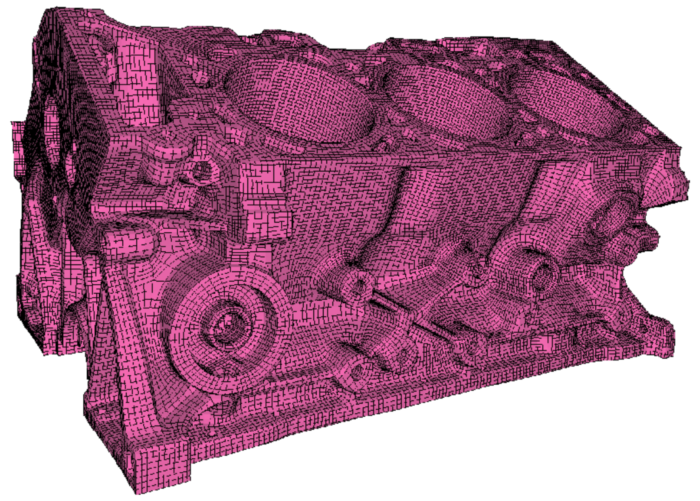
CONVERTING BETWEEN REPRESENTATIONS

Point cloud

Meshes

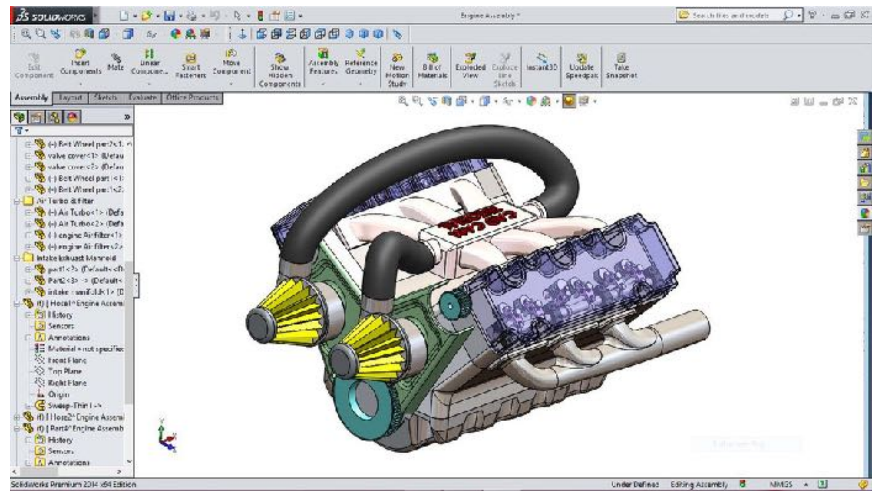


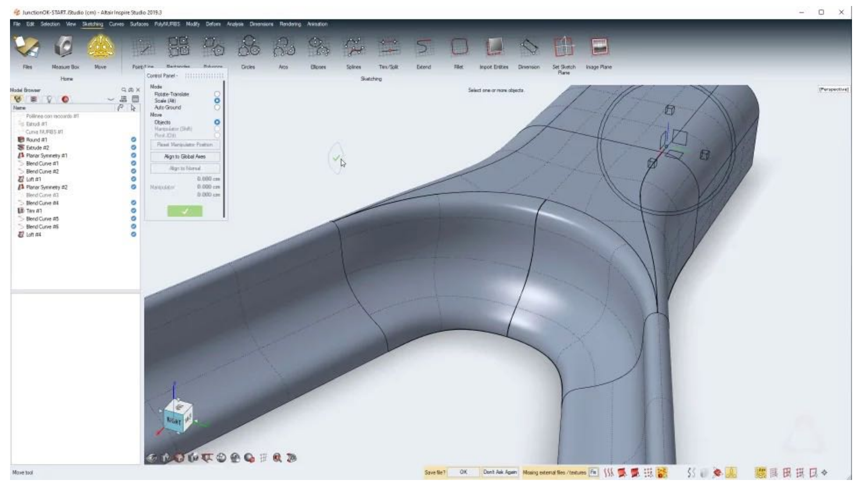


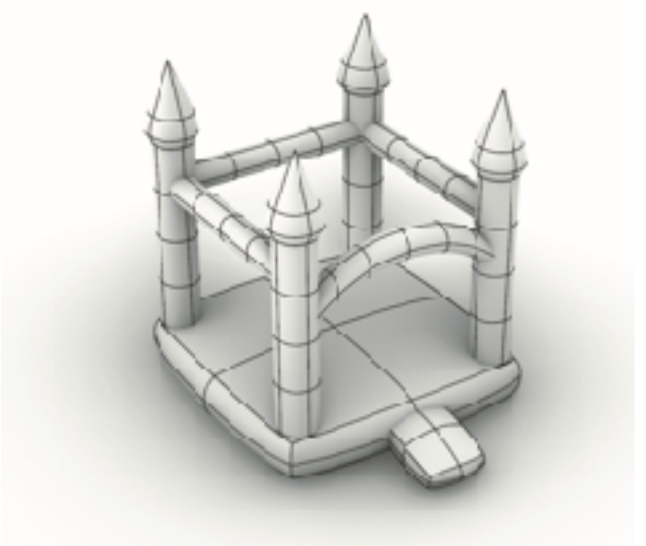


Implicit

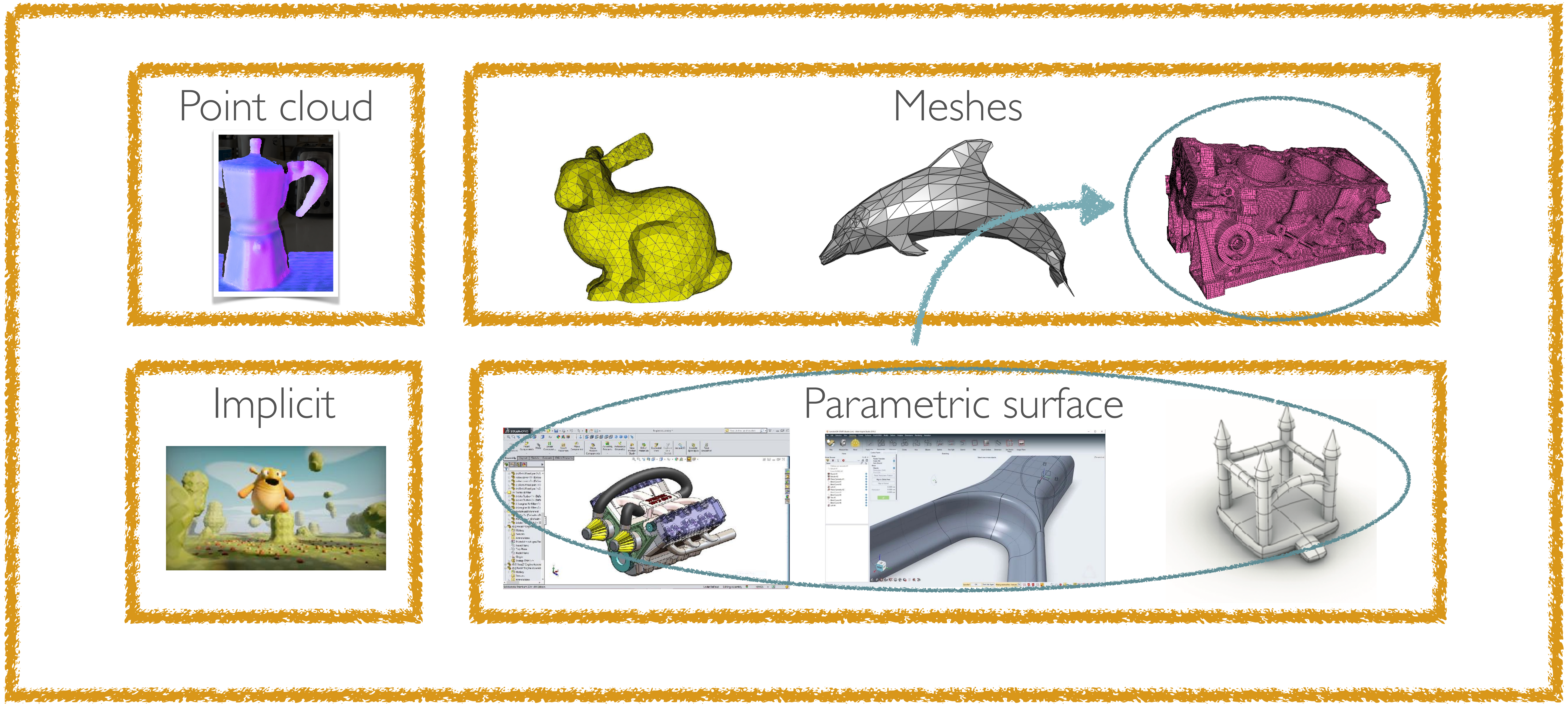
Parametric surface







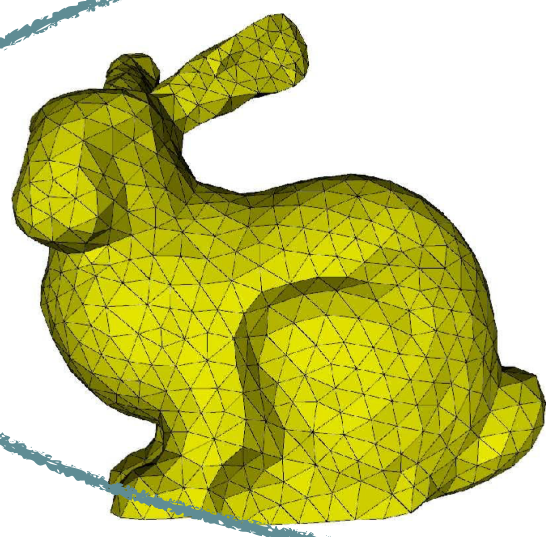
MOST ARE SIMPLE…

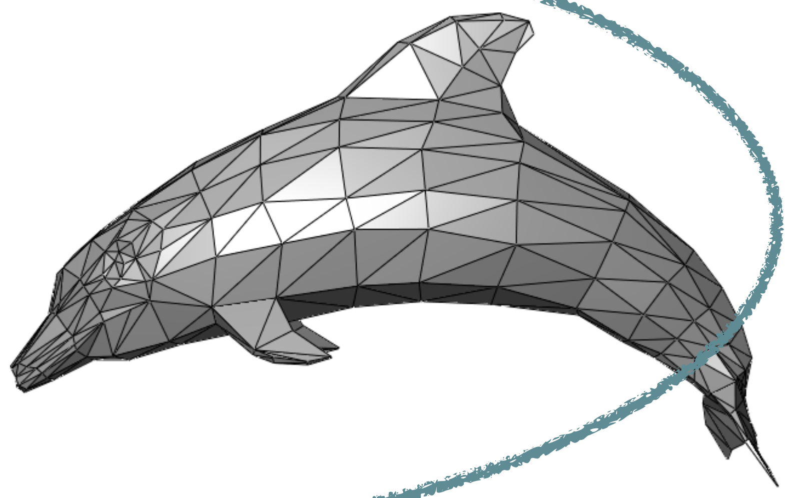


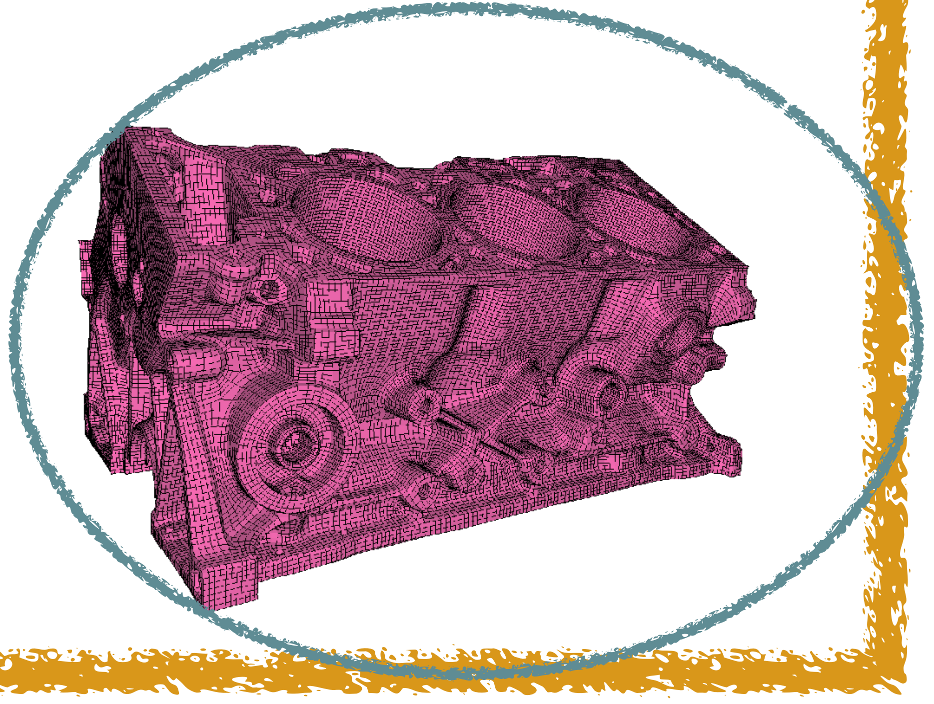
MOST ARE SIMPLE…

Point cloud

Meshes

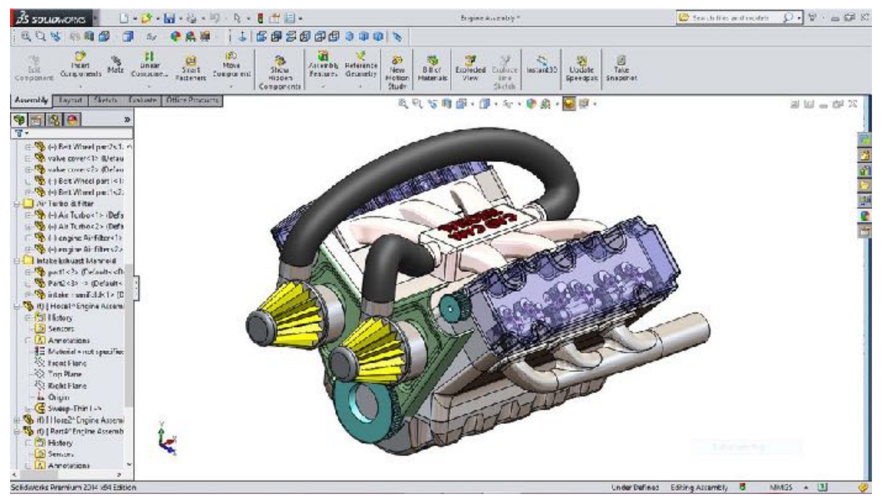


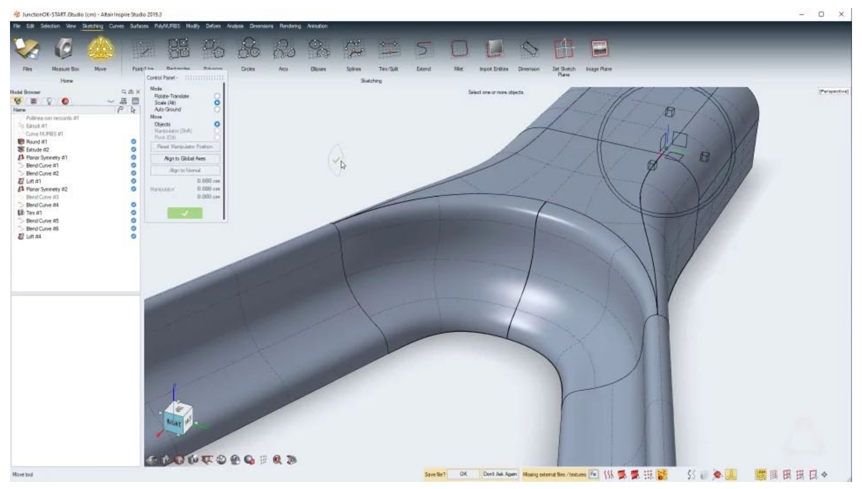


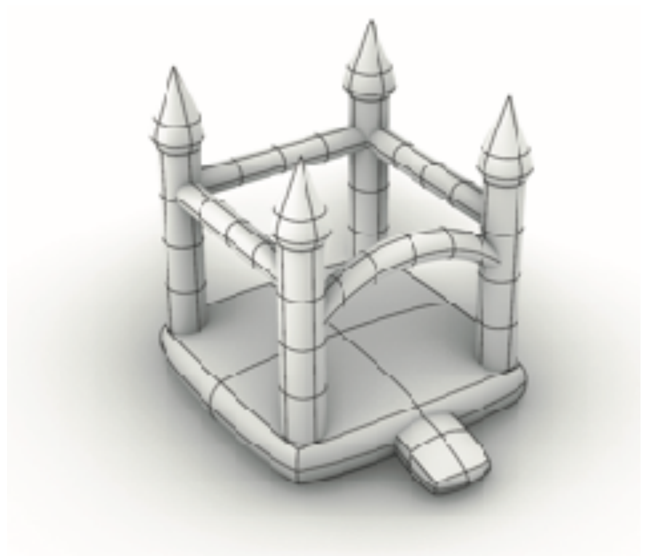


Implicit

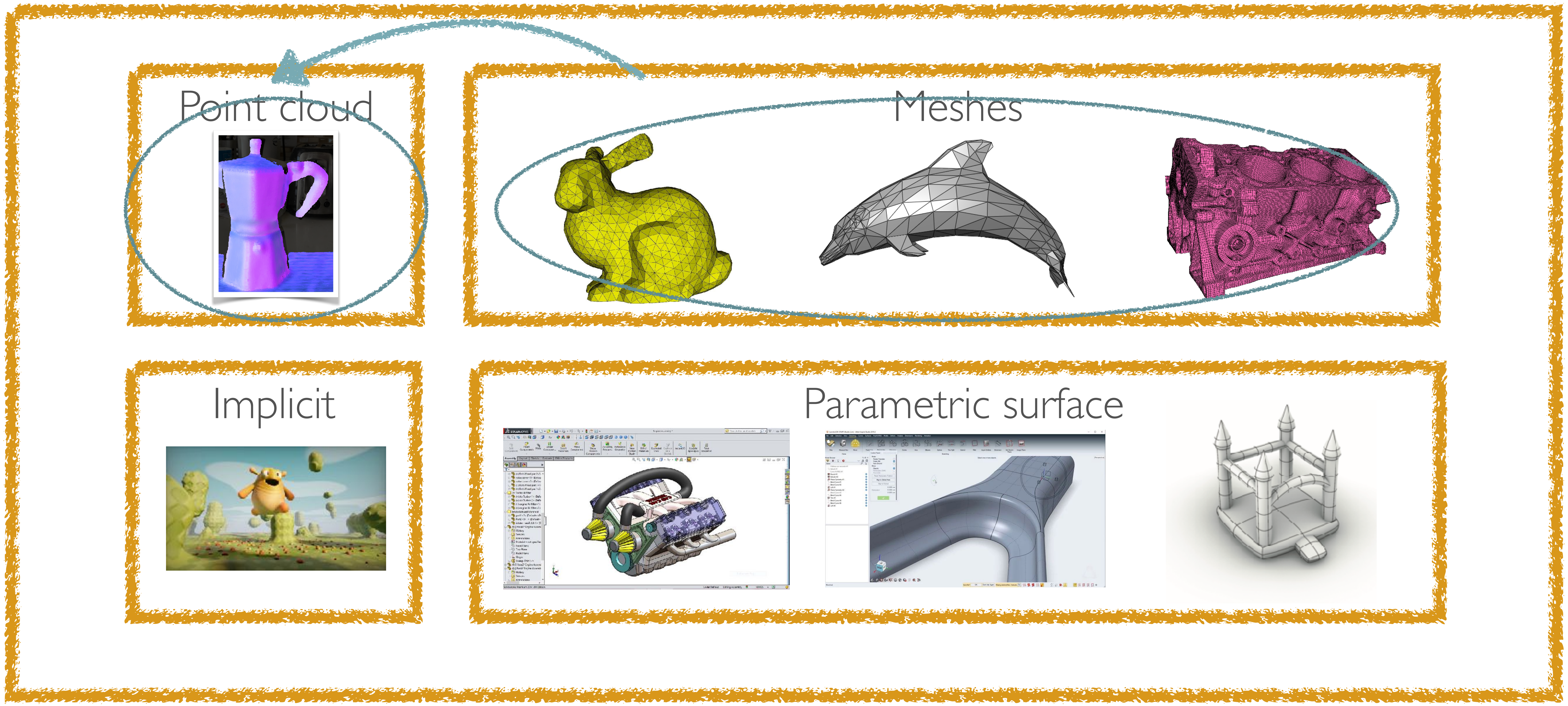
Parametric surface



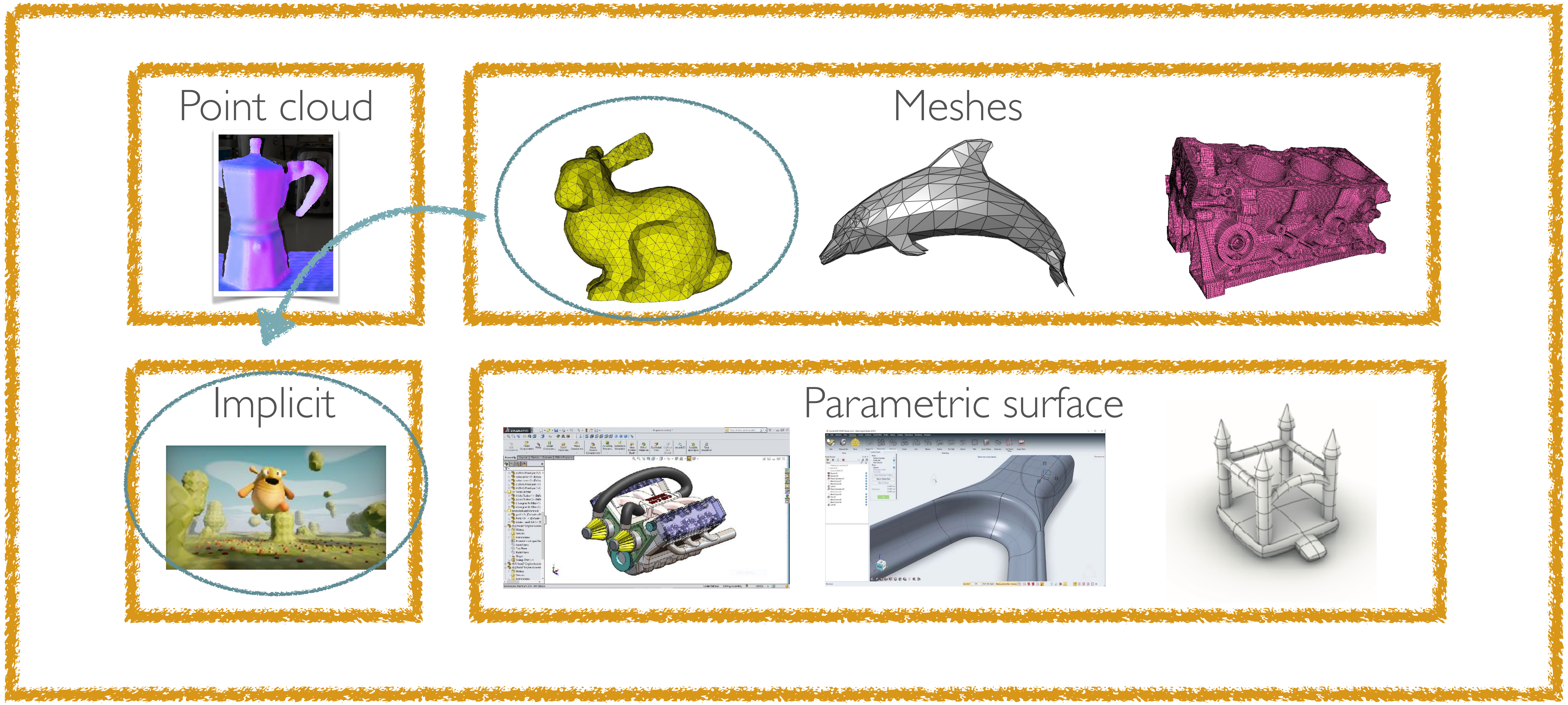




MOST ARE SIMPLE…



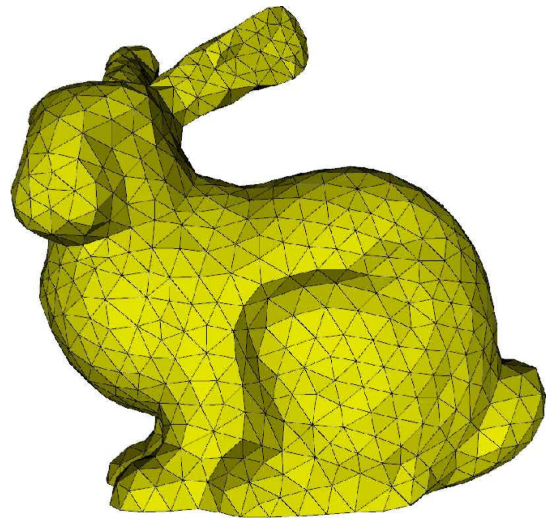
MOST ARE SIMPLE…

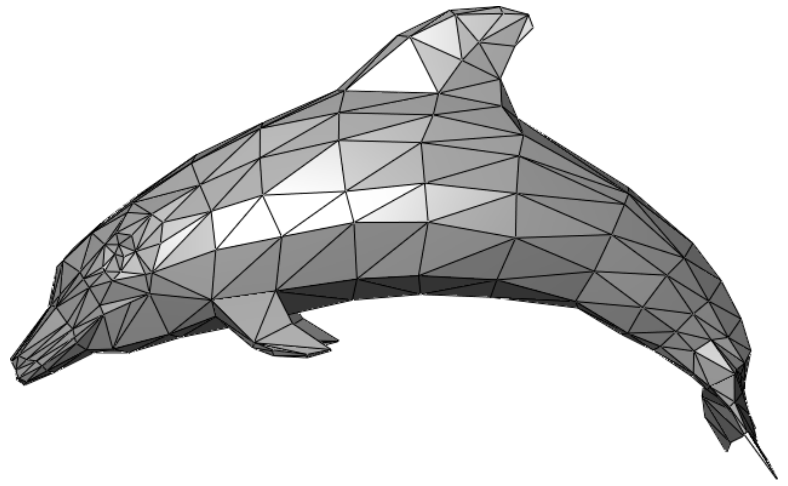


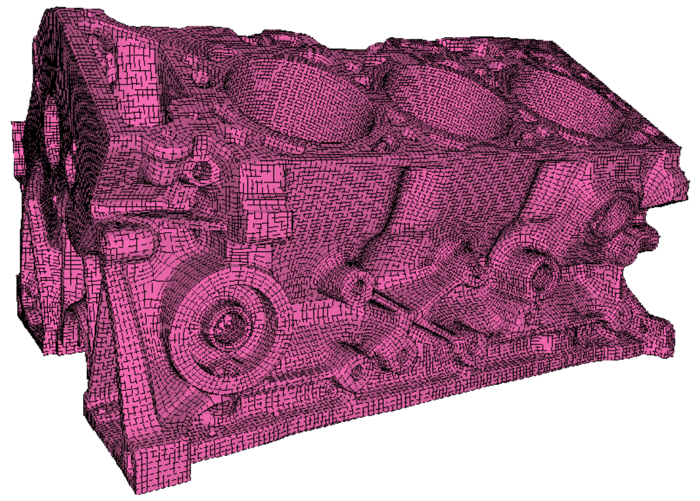
…SOME REALLY AREN'T!

Point cloud

Meshes

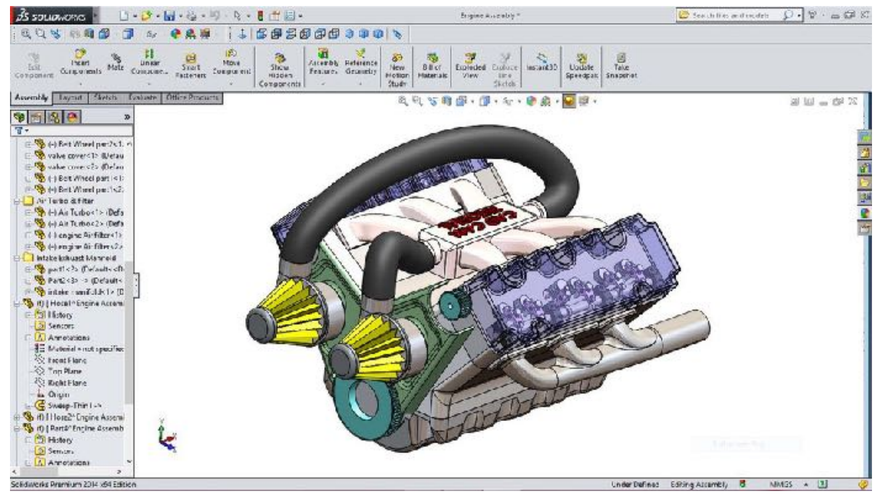


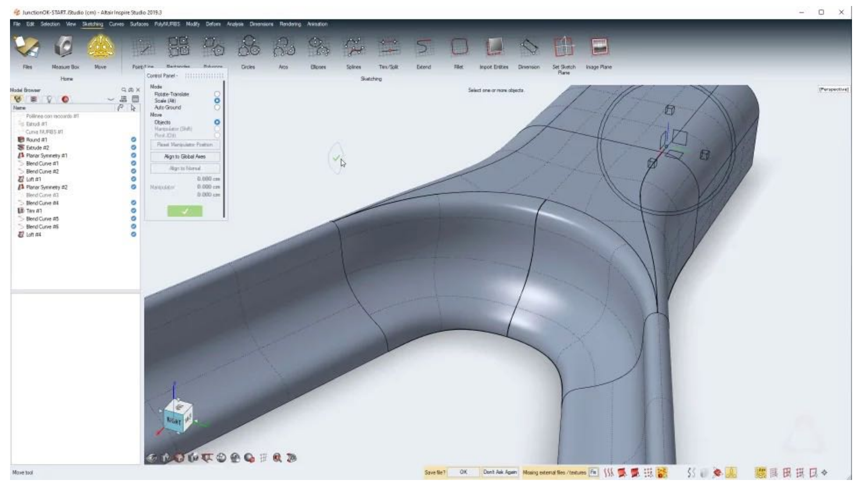


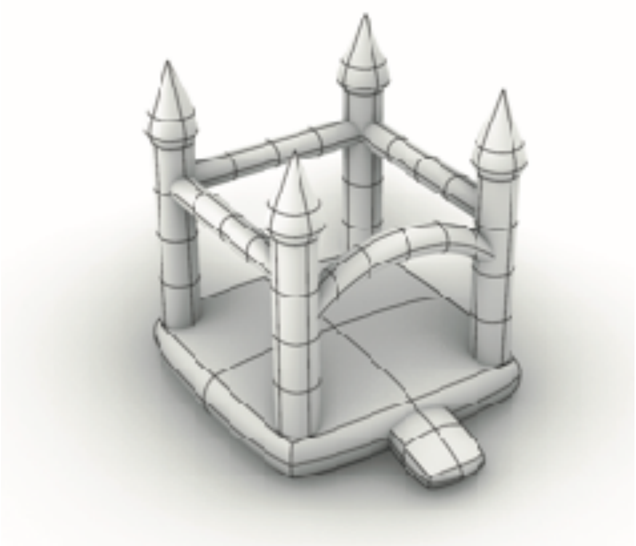


Implicit

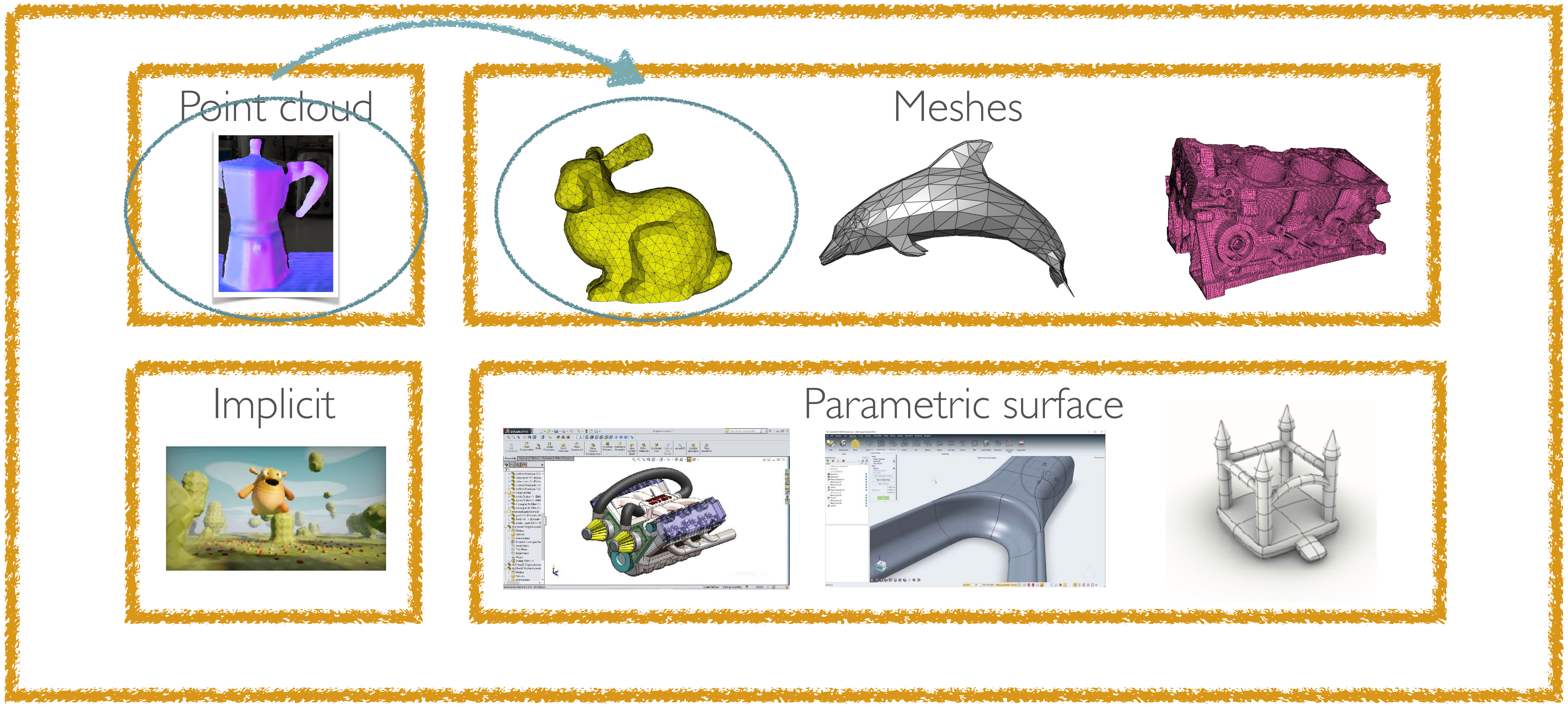
Parametric surface



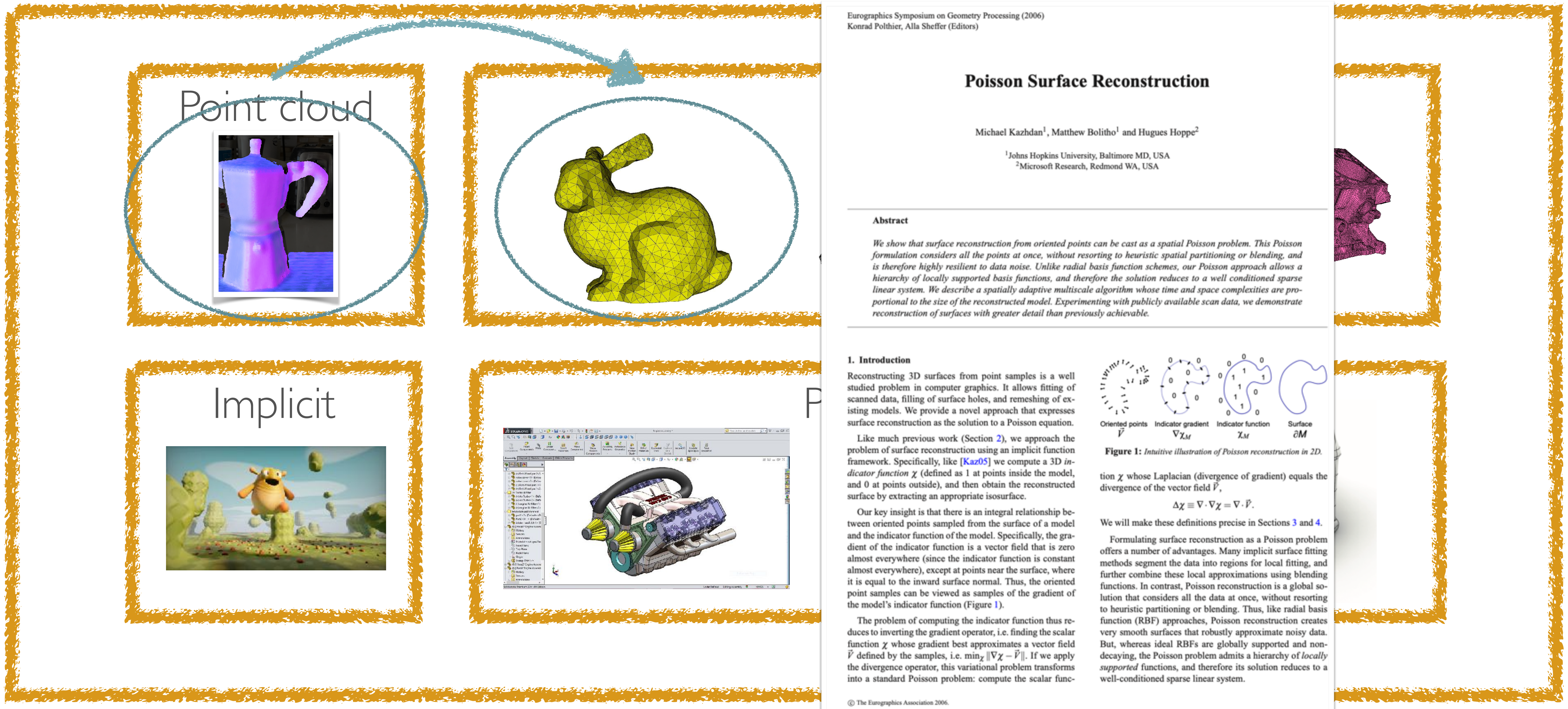




…SOME REALLY AREN'T!



…SOME REALLY AREN'T!



Eurographics Symposium on Geometry Processing (2006)
Konrad Polthier, Alla Sheffer (Editors)

Poisson Surface Reconstruction

Michael Kazhdan[1], Matthew Bolitho[1] and Hugues Hoppe[2]

[1]Johns Hopkins University, Baltimore MD, USA
[2]Microsoft Research, Redmond WA, USA

Abstract

We show that surface reconstruction from oriented points can be cast as a spatial Poisson problem. This Poisson formulation considers all the points at once, without resorting to heuristic spatial partitioning or blending, and is therefore highly resilient to data noise. Unlike radial basis function schemes, our Poisson approach allows a hierarchy of locally supported basis functions, and therefore the solution reduces to a well conditioned sparse linear system. We describe a spatially adaptive multiscale algorithm whose time and space complexities are proportional to the size of the reconstructed model. Experimenting with publicly available scan data, we demonstrate reconstruction of surfaces with greater detail than previously achievable.

1. Introduction

Reconstructing 3D surfaces from point samples is a well studied problem in computer graphics. It allows fitting of scanned data, filling of surface holes, and remeshing of existing models. We provide a novel approach that expresses surface reconstruction as the solution to a Poisson equation.

Like much previous work (Section 2), we approach the problem of surface reconstruction using an implicit function framework. Specifically, like [Kaz05] we compute a 3D *indicator function* χ (defined as 1 at points inside the model, and 0 at points outside), and then obtain the reconstructed surface by extracting an appropriate isosurface.

Our key insight is that there is an integral relationship between oriented points sampled from the surface of a model and the indicator function of the model. Specifically, the gradient of the indicator function is a vector field that is zero almost everywhere (since the indicator function is constant almost everywhere), except at points near the surface, where it is equal to the inward surface normal. Thus, the oriented point samples can be viewed as samples of the gradient of the model's indicator function (Figure 1).

The problem of computing the indicator function thus reduces to inverting the gradient operator, i.e. finding the scalar function χ whose gradient best approximates a vector field $\vec{V}$ defined by the samples, i.e. $\min_\chi \|\nabla\chi - \vec{V}\|$. If we apply the divergence operator, this variational problem transforms into a standard Poisson problem: compute the scalar function χ whose Laplacian (divergence of gradient) equals the divergence of the vector field $\vec{V}$,

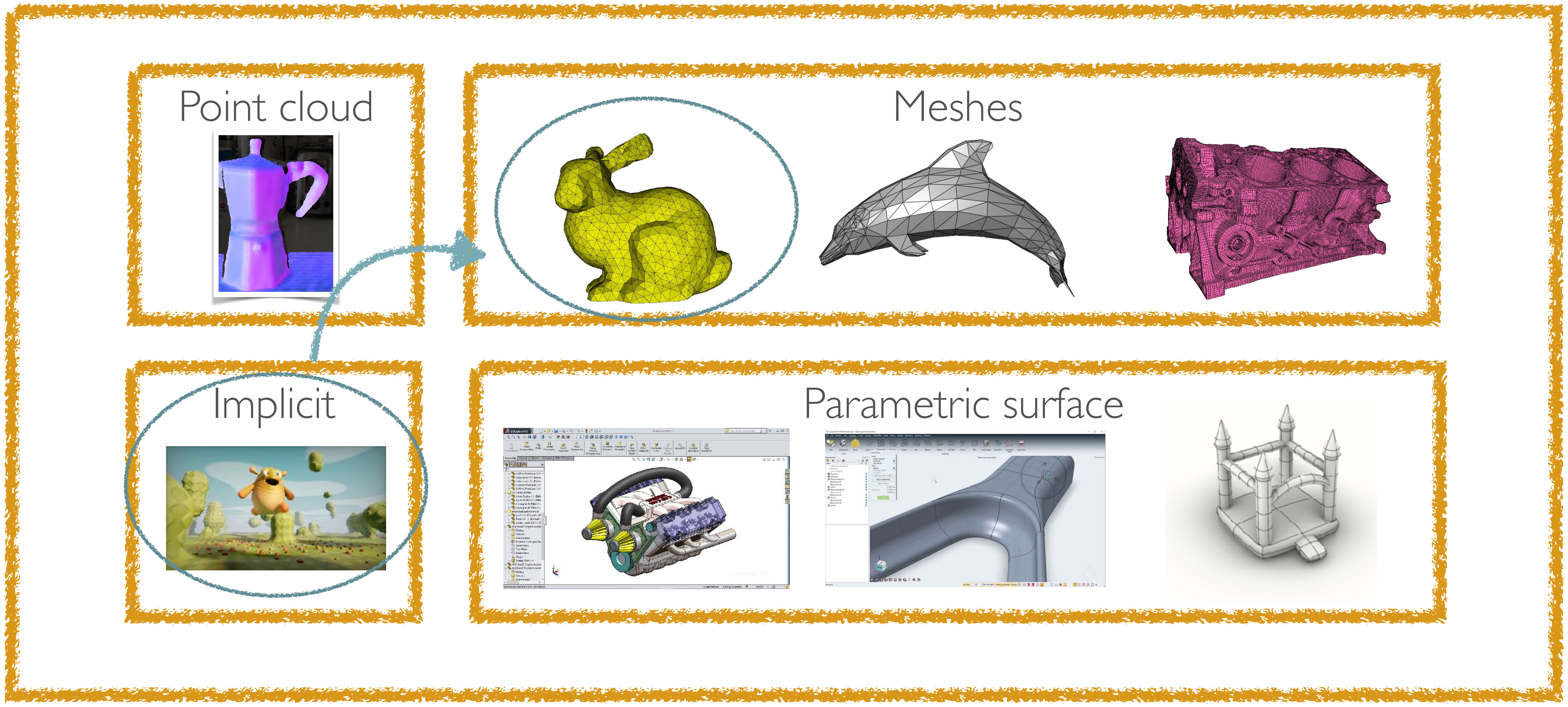
Figure 1: *Intuitive illustration of Poisson reconstruction in 2D.*

$$\Delta\chi \equiv \nabla\cdot\nabla\chi = \nabla\cdot\vec{V}.$$

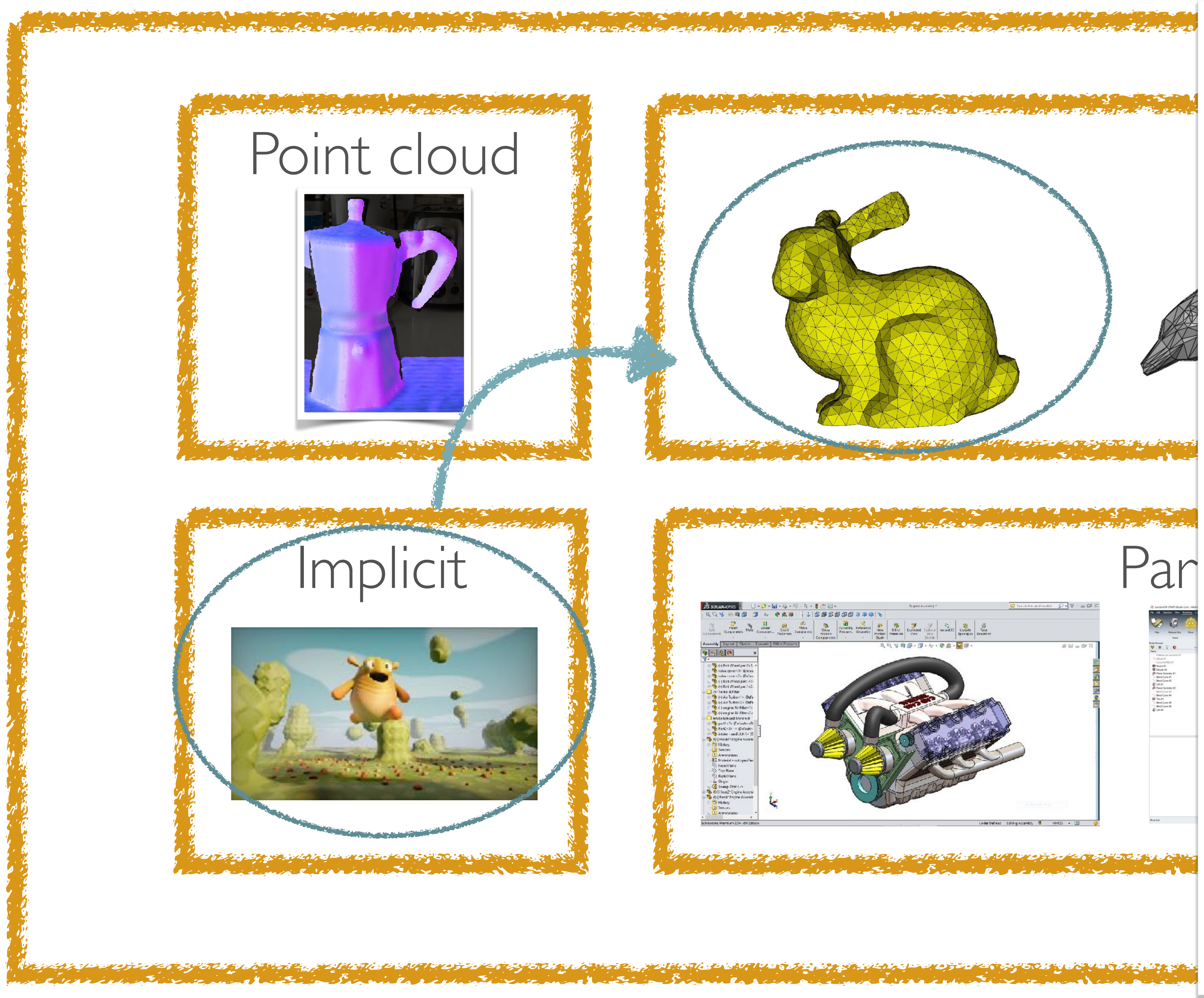
We will make these definitions precise in Sections 3 and 4.

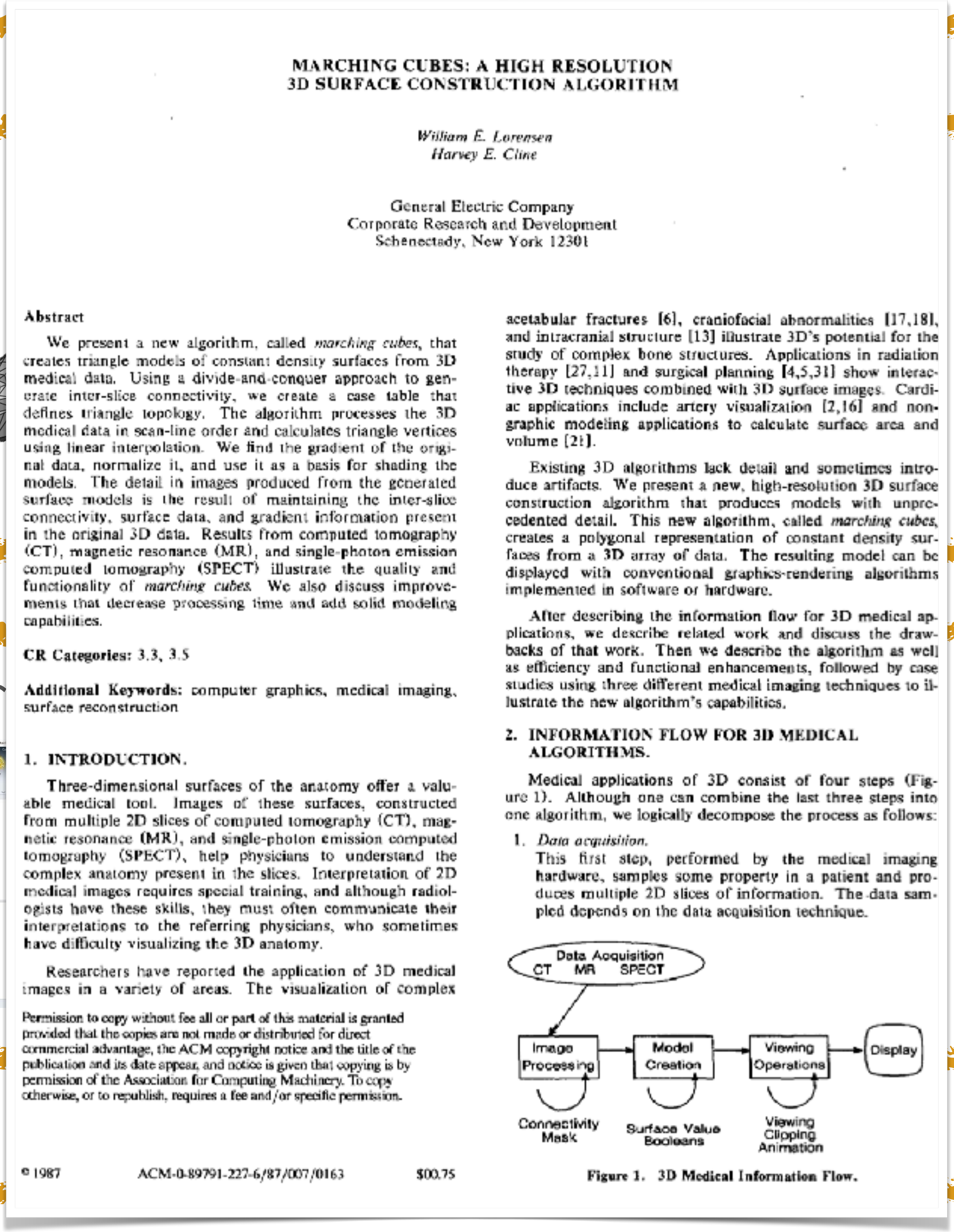
Formulating surface reconstruction as a Poisson problem offers a number of advantages. Many implicit surface fitting methods segment the data into regions for local fitting, and further combine these local approximations using blending functions. In contrast, Poisson reconstruction is a global solution that considers all the data at once, without resorting to heuristic partitioning or blending. Thus, like radial basis function (RBF) approaches, Poisson reconstruction creates very smooth surfaces that robustly approximate noisy data. But, whereas ideal RBFs are globally supported and non-decaying, the Poisson problem admits a hierarchy of *locally supported* functions, and therefore its solution reduces to a well-conditioned sparse linear system.

…SOME REALLY AREN'T!



…SOME REALLY AREN'T!





MARCHING CUBES: A HIGH RESOLUTION
3D SURFACE CONSTRUCTION ALGORITHM

William E. Lorensen
Harvey E. Cline

General Electric Company
Corporate Research and Development
Schenectady, New York 12301

Abstract

We present a new algorithm, called *marching cubes*, that creates triangle models of constant density surfaces from 3D medical data. Using a divide-and-conquer approach to generate inter-slice connectivity, we create a case table that defines triangle topology. The algorithm processes the 3D medical data in scan-line order and calculates triangle vertices using linear interpolation. We find the gradient of the original data, normalize it, and use it as a basis for shading the models. The detail in images produced from the generated surface models is the result of maintaining the inter-slice connectivity, surface data, and gradient information present in the original 3D data. Results from computed tomography (CT), magnetic resonance (MR), and single-photon emission computed tomography (SPECT) illustrate the quality and functionality of *marching cubes*. We also discuss improvements that decrease processing time and add solid modeling capabilities.

CR Categories: 3.3, 3.5

Additional Keywords: computer graphics, medical imaging, surface reconstruction

1. INTRODUCTION.

Three-dimensional surfaces of the anatomy offer a valuable medical tool. Images of these surfaces, constructed from multiple 2D slices of computed tomography (CT), magnetic resonance (MR), and single-photon emission computed tomography (SPECT), help physicians to understand the complex anatomy present in the slices. Interpretation of 2D medical images requires special training, and although radiologists have these skills, they must often communicate their interpretations to the referring physicians, who sometimes have difficulty visualizing the 3D anatomy.

Researchers have reported the application of 3D medical images in a variety of areas. The visualization of complex acetabular fractures [6], craniofacial abnormalities [17,18], and intracranial structure [13] illustrate 3D's potential for the study of complex bone structures. Applications in radiation therapy [27,11] and surgical planning [4,5,31] show interactive 3D techniques combined with 3D surface images. Cardiac applications include artery visualization [2,16] and nongraphic modeling applications to calculate surface area and volume [21].

Existing 3D algorithms lack detail and sometimes introduce artifacts. We present a new, high-resolution 3D surface construction algorithm that produces models with unprecedented detail. This new algorithm, called *marching cubes*, creates a polygonal representation of constant density surfaces from a 3D array of data. The resulting model can be displayed with conventional graphics-rendering algorithms implemented in software or hardware.

After describing the information flow for 3D medical applications, we describe related work and discuss the drawbacks of that work. Then we describe the algorithm as well as efficiency and functional enhancements, followed by case studies using three different medical imaging techniques to illustrate the new algorithm's capabilities.

2. INFORMATION FLOW FOR 3D MEDICAL ALGORITHMS.

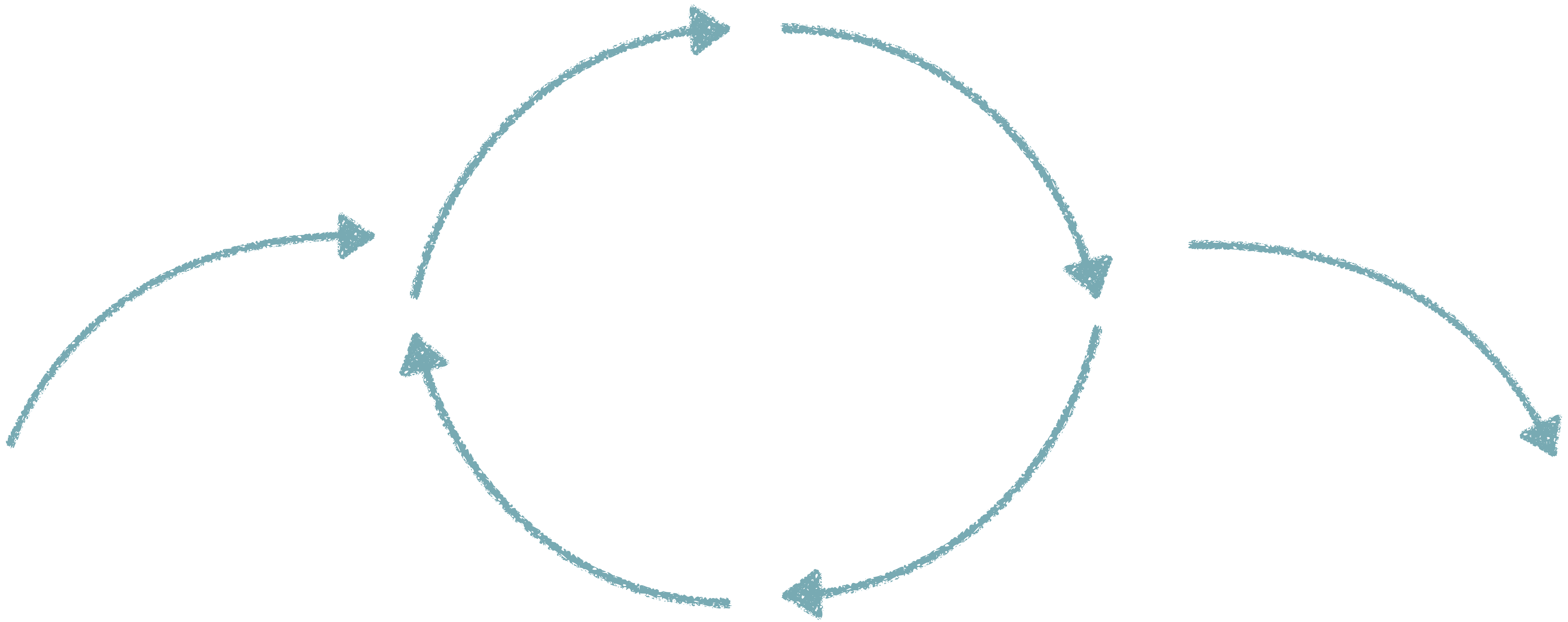
Medical applications of 3D consist of four steps (Figure 1). Although one can combine the last three steps into one algorithm, we logically decompose the process as follows:

1. *Data acquisition.*
   This first step, performed by the medical imaging hardware, samples some property in a patient and produces multiple 2D slices of information. The data sampled depends on the data acquisition technique.

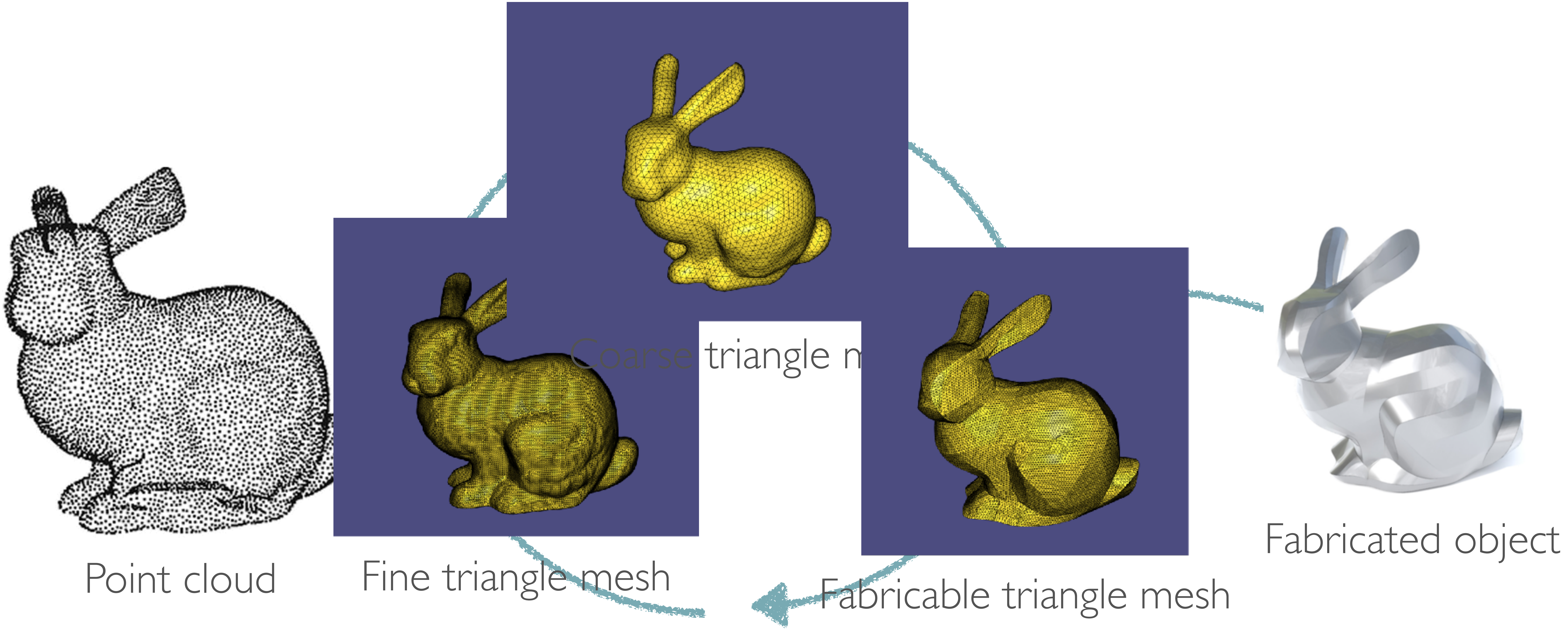
Data Acquisition
CT MR SPECT
Image Processing
Model Creation
Viewing Operations
Display
Connectivity Mask
Surface Value Booleans
Viewing Clipping Animation

Figure 1. 3D Medical Information Flow.

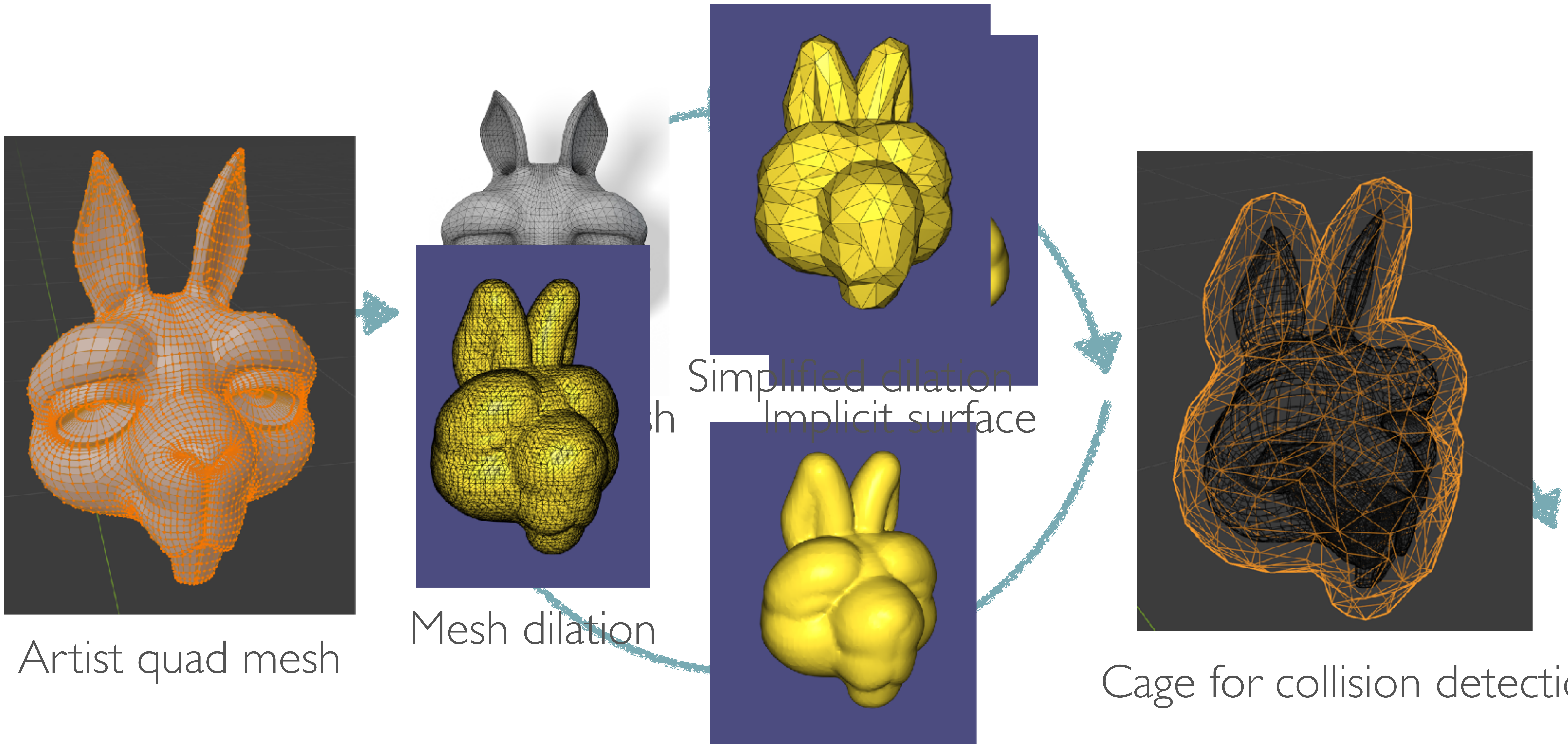
A SHAPE'S LIFE CYCLE



A SHAPE'S LIFE CYCLE



A SHAPE'S LIFE CYCLE



Artist quad mesh

Mesh dilation

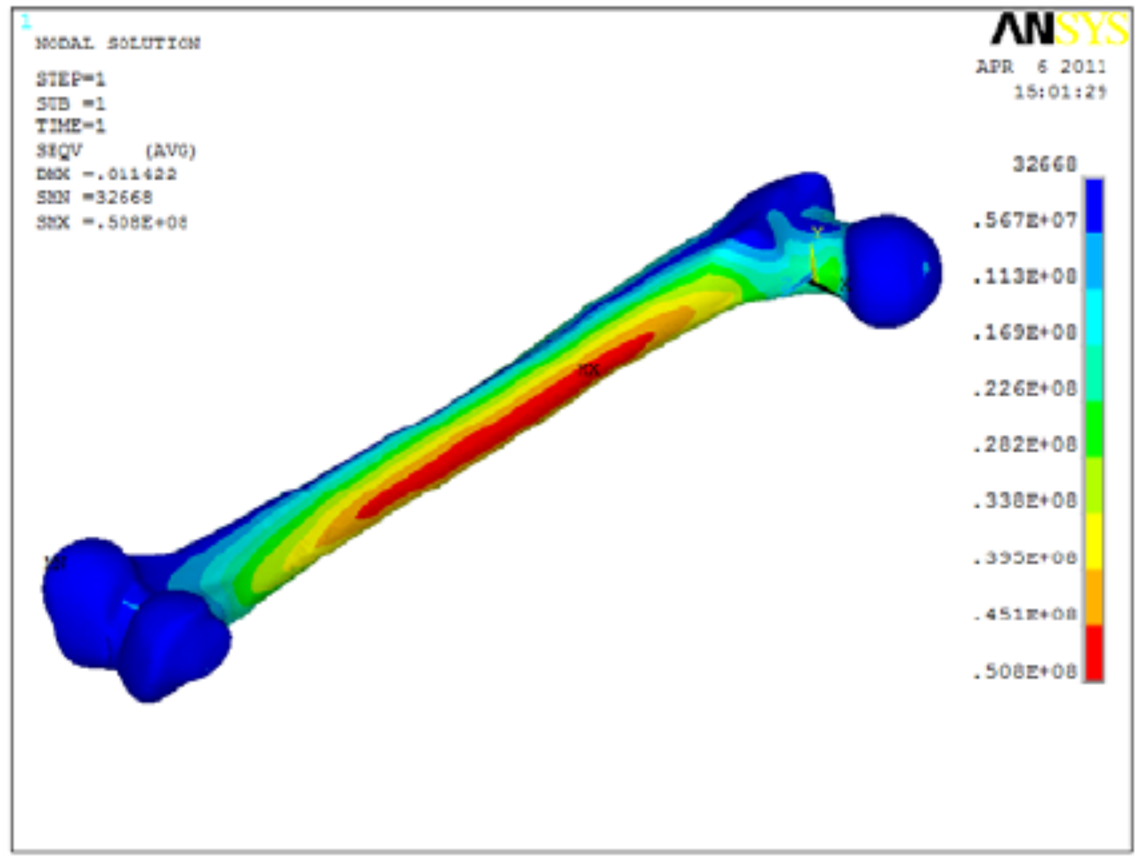
Simplified dilation

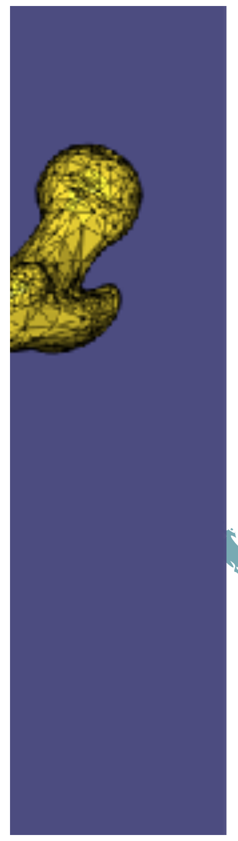
Implicit surface

Implicit dilation

Cage for collision detection

A SHAPE'S LIFE CYCLE



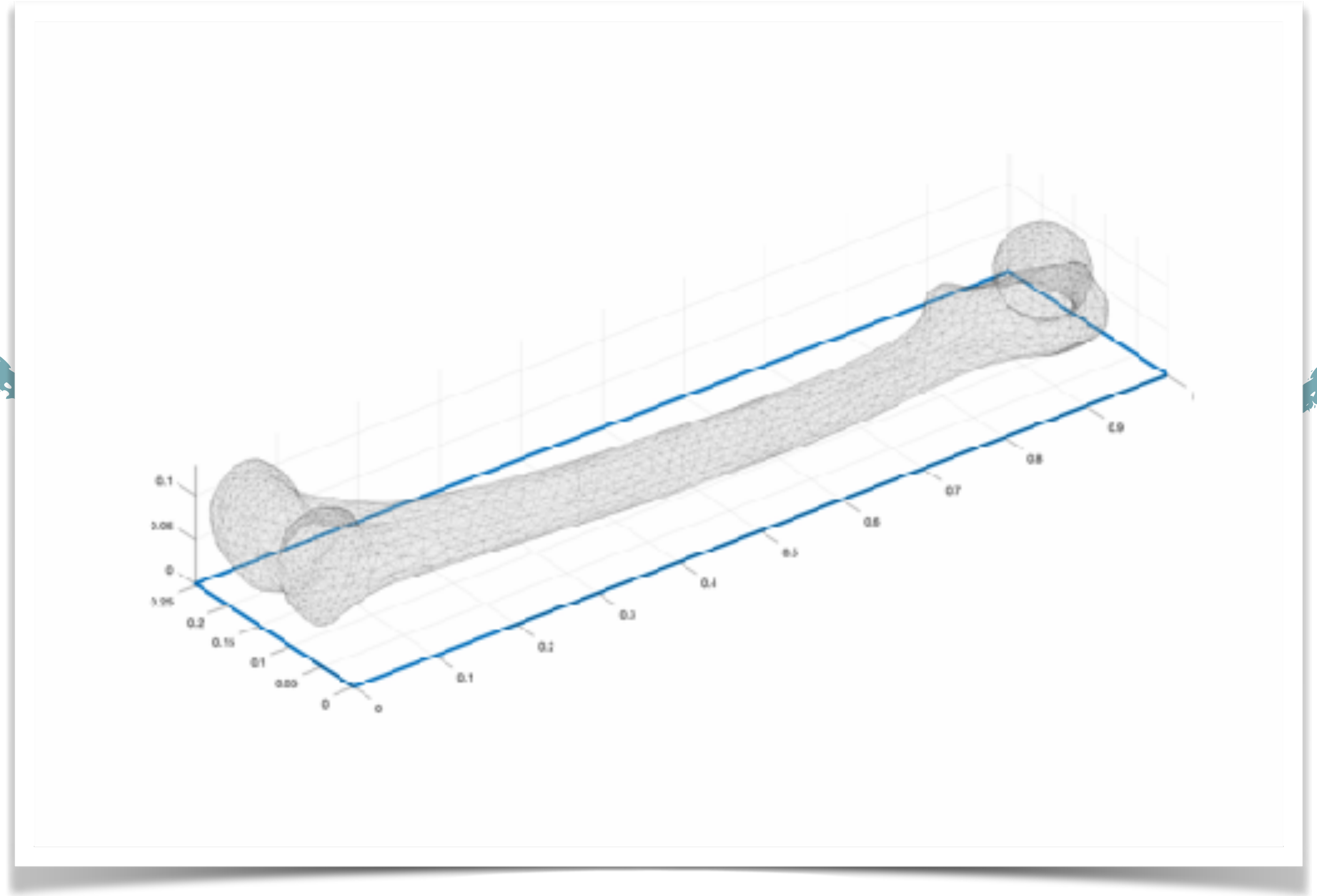


Stress analysis

Triangle mesh

CT scan (implicit)

Diagnostic



Tetrahedral mesh

TODAY'S LESSON

There are many shape representations, with different advantages and disadvantages

We can't control which representation a shape is in so we need to study all

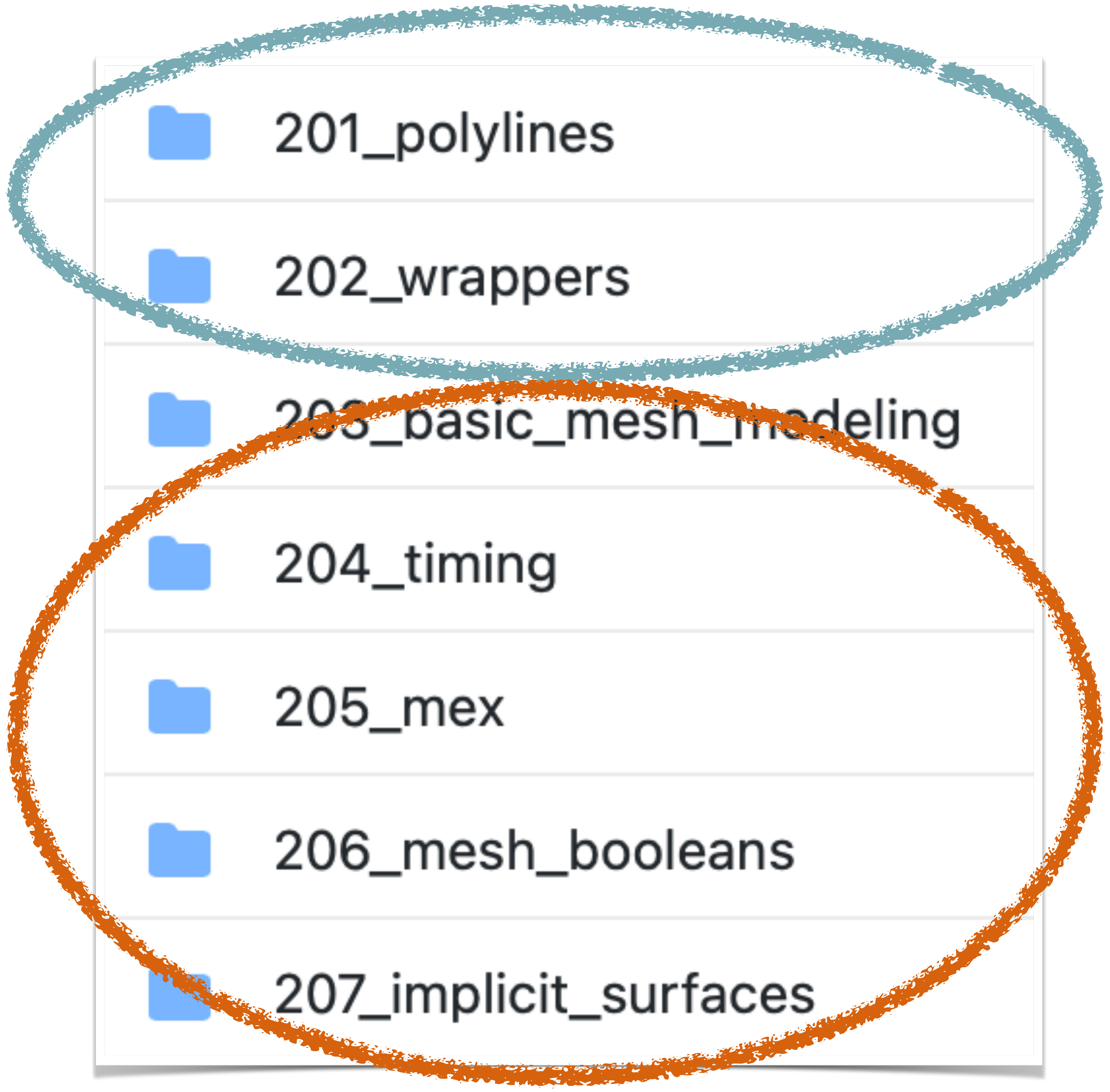
Each representation leads itself to different research questions

Developing an intuition in 2D helps to understand 3D

MESH MATH AND BEYOND

On creating, storing and using geometry

WHAT'S THE PLAN NOW?



Earlier (and also now)

Now

WHAT'S THE PLAN NOW?

- 201_polylines
- 202_wrappers
- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_mesh_booleans
- 207_implicit_surfaces

The interesting ones

Read only