MESH MATH AND BEYOND
On creating, storing and using geometry
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

A shape is born

Stuff happens to the shape

A shape dies

SGI coffee

“Who cares if it’s any good, Justin’s paying*!”

*terms and conditions apply
SHAPES HAVE A LIFE CYCLE TOO

A shape is born

Stuff happens to the shape

A shape dies
SHAPES HAVE A LIFE CYCLE TOO

Overwatch - Tracer 3D print model - time-lapse sculpt, by “Printed Obsession”, Youtube
SHAPES HAVE A LIFE CYCLE TOO

A shape is born

Stuff happens to the shape

A shape dies
SHAPES HAVE A LIFE CYCLE TOO

How devs break bones to make animation feel right, by “Jenna Stoeber (Polygon)”, Youtube
SHAPES HAVE A LIFE CYCLE TOO

A shape is born

Stuff happens to the shape

A shape dies
SHAPES HAVE A LIFE CYCLE TOO

How devs break bones to make animation feel right, by “Jenna Stoeber (Polygon)”, Youtube
WHAT YOU’VE SEEN SO FAR

NORMAL VECTOR

- The normal vector $\hat{n}$ is the unit-length perpendicular vector to a triangle and positively oriented.
- $\hat{n} = e_1 \times e_2$, $n = \hat{n}/||\hat{n}||$
- Normals point outside

A shape is born → Normal Vector → 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

NORMAL VECTOR
- The normal vector is the unit-length perpendicular vector to a triangle and positively oriented:
  \[ \mathbf{n} = \mathbf{e}_1 \times \mathbf{e}_2 = \frac{\mathbf{a}(\mathbf{b})(\mathbf{c})}{\|\mathbf{a}\|\|\mathbf{b}\|\|\mathbf{c}\|} \]
- Normals point outside
WHAT’S WRONG WITH MESHES?
REASON 1: NOT EVERY SHAPE IS BORN AS A MESH

A shape is born

Stuff happens to the shape

A shape dies
REASON II: “STUFF” CAN BE HARD TO DO ON MESHES

A shape is born

Stuff happens to the shape

A shape dies
SHAPE REPRESENTATIONS IN 2D
SHAPE REPRESENTATIONS IN 2D
2D’s version of “Surface” is a curve $\gamma$.
...HOW DO I STORE A CURVE ON A COMPUTER?
AN OPTION: FINITE SET OF POINTS

$p_1 = [0,0]$

$p_2 = [0.322, 0.28]$

$p_3 = [0.456, 0.420]$

$p_4 = [0.662, 0.4435]$

$p_5 = [0.722, 0.852]$
AN OPTION: FINITE SET OF POINTS

\( p_1 = [0,0] \)
\( p_2 = [0.322,0.28] \)
\( p_3 = [0.456,0.420] \)
\( p_4 = [0.662,0.4435] \)
\( p_5 = [0.722,0.852] \)
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 ↔ 2
2 ↔ 3
3 ↔ 4
4 ↔ 5
AN OPTION: POINTS + CONNECTIVITY

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

?
AN OPTION: POINTS + CONNECTIVITY

1[0.0] + 1 ↔ 2
2[0.322,0.28] + 2 ↔ 3
3[0.456,0.420] + 3 ↔ 4
4[0.662,0.4435] + 4 ↔ 5
5[0.722,0.852]

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 ↔ 2 + 2 ↔ 3 + 3 ↔ 4 + 4 ↔ 5

Linear interpolation
AN OPTION: POLYLINE

1[0,0] + 1 ↔ 2 + Linear interpolation
2[0.322,0.28] 2 ↔ 3
3[0.456,0.420] 3 ↔ 4
4[0.662,0.4435] 4 ↔ 5
5[0.722,0.852]
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? (not easy)
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? (x)
- Looks bad! (x)
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

5 constraints
SPLINES

Cubic spline
SPLINES

Cubic spline

Catmull-Rom Spline
SPLINES

Font as a B-spline curve

Data: G. Farin, Curves and Surfaces for Computer Aided Geometric Design
SPLINES
A Class of $C^2$ Interpolating Splines

Cem Yuksel
University of Utah

A Class of $C^2$ 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

- **Point cloud**
- **Points + Connectivity + Piecewise flat interpolation**
- **Points + Connectivity + Polynomial interpolation**
CLIPasso: Semantically-Aware Object Sketching

Yael Vinker1,2, Ehuan Pajouheshgar1, Jessica Y. Bo1, Roman Christian Bachmann1, Amit Haim Berman2, Daniel Cohen-Or1, Amir Zamir1, Ariel Shamir2

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

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

Abstraction 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 materizer 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.

The main objective is to produce visualizable objects that preserve the essential visual features of the subject, which requires learning a mapping between shape and visual abstraction levels. We begin with the mid-level abstraction of a Picasso painting (Figure 3). 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 composed 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, 23]. However, creating abstractions is more difficult for machines to achieve. The abstraction process suggests that the artist selects visual features that capture
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**

\[ f(x, y) = \max(|x|, |y|) - 1 \]

\[ f(x, y) = 0 \]
**IMPLICIT SHAPE REPRESENTATIONS**

\[ f(x, y) = \max( |x|, |y| ) - 1 \]

- Positive values outside shape
- Negative values inside shape
**Implicit Shape Representations**

\[ f(x, y) = \max(x, y) - 1 \]

- Positive values outside shape
- Negative values inside shape
- \( f(x, y) = 0 \)
**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

\[ f_{\text{intersection}} = \max(f_1, f_2) \]
**IMPLICIT SHAPE REPRESENTATIONS**

\[ f_{\text{subtraction}} = \max(f_1, -f_2) \]
**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

- Point cloud
- Points + Connectivity + Piecewise flat interpolation
- Implicit
- Points + Connectivity + Polynomial interpolation
Implicit function
Signed distance function

\[ f(x, y) = 0 \]
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_meshBOOLEANS
- 207_implicit_surfaces

Same repo as yesterday
WHAT'S THE PLAN NOW?

Now

- 201_polylines
- 202_wrappers

Later in the day

- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_meshBOOLEANS
- 207_implicit_surfaces
WHAT'S THE PLAN NOW?

The boring one

201_polylines
202_wrappers
203_basic_mesh_modeling
204_timing
205_mex
206_mesh_booleans
207_implicit_surfaces
WHAT'S THE PLAN NOW?

The boring one
Requires installing stuff
WHAT'S THE PLAN NOW?

The boring one
Requires installing stuff
For reference, no need to do it today
WHAT'S THE PLAN NOW?

The interesting one

- 201_polylines
- 202_wrappers
- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_mesh_booleans
- 207_implicit_surfaces
WHAT'S THE PLAN NOW?

The interesting one
It's meant to be hard!
Error using `error`
WHAT TO DO WHEN A TASK IS HARD?

**silvi**

**help**

**oded**

In general, or you need help now?

**silvi**

now, if you're not busy

**oded**

Want to proofread?

**silvi**

Your Asia submission?

**oded**

So

Direct Message | Jan 12th, 2020 | View conversation

**silvi**

Much

Direct Message | Jan 12th, 2020 | View conversation

**silvi**

Stress

Direct Message | Jan 12th, 2020 | View conversation

**oded**

Do you want me to get you something

**silvi**

**WHY IS MY BLENDER NOT WORKING**

**silvi**

**17:04**

Why are my shadows so ugly

they should be so sharp with environment lighting

**oded**

**17:21**

Is it too bright?

Is your material overtly shiny?

**silvi**

**11:04**

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?

**oded**

**11:07**

There is a wireframe node in the cycles shader

Like when you make a material

play around with that

**silvi**

**11:07**

There you go

thank you oded

**oded**

Have you saved and quit?
GO!

- 201_polylines
- 202_wrappers
- 203_basic_mesh_modeling
- 204_timing
- 205_mex
- 206_meshBOOLEANS
- 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

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”

Everyone else

Taylor

Circle (2D sphere) of life, professional progression, etc.

“Oh no, I am back where I started! The curve became a sphere!”
Option 3: “The graph”

Taylor, very ahead of the curve

The curve, becoming a sphere
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

20 cm

3D scanning
Point cloud

3D scanning

20 cm
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" Solán et al. 2019
...always ask yourselves why

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.

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.'"
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
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

Most common shape representation
Mesh

Most common shape representation
Mesh

Most common shape representation

Easiest to work with for most applications
Mesh

Most common shape representation

Easiest to work with for most applications

Very hard/impossible to capture directly

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

Most common shape representation

Easiest to work with for most applications

Very hard/impossible to capture directly

“The first real object ever 3D scanned and rendered was a VW Beetle” by Jason Torchinsky
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
Surface spline
CAD surface
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
The Shape Matching Element Method: Direct Animation of Curved Surface Models

TY TRUSTY, University of Toronto, Canada
HONGLIN CHEN, University of Toronto, Canada
DAVID E. 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.

Parametric surface

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

- 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\textsuperscript{1,3\dag} Peter Florence\textsuperscript{2,3\dag} Julian Straub\textsuperscript{3} Richard Newcombe\textsuperscript{3} Steven Lovegrove\textsuperscript{3}

\textsuperscript{1}University of Washington \textsuperscript{2}Massachusetts Institute of Technology \textsuperscript{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\textsuperscript{1}\textsuperscript{E}  Alec Jacobson\textsuperscript{2}\textsuperscript{E}  Derek Nowrouzezahrai\textsuperscript{3}\textsuperscript{E}  Wojciech Jarosz\textsuperscript{1}

\textsuperscript{1}Dartmouth College  \textsuperscript{2}University of Toronto  \textsuperscript{3}McGill University

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

Point cloud

Meshes

Implicit

Parametric surface
MOST ARE SIMPLE…
MOST ARE SIMPLE…

- Point cloud
- Meshes
- Implicit
- Parametric surface
MOST ARE SIMPLE…

- **Point cloud**
- **Implicit**
- **Meshes**
- **Parametric surface**
...SOME REALLY AREN’T!

- Point cloud
- Meshes
- Implicit
- Parametric surface
...SOME REALLY AREN’T!

- Point cloud
- Meshes
- Implicit
- Parametric surface
...SOME REALLY AREN’T!

Poisson Surface Reconstruction

Michael Kazhdan1, Matthew Bolitho1 and Hugues Hoppe2

1Adobe Research, University of North Carolina, USA
2Microsoft 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 induces a well-conditioned sparse linear system. We describe a quality adaptive multiscale algorithm whose time and space complexities are proportional to the size of the reconstructed model. Experimenting with publicly available raw 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, fitting 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 [Kazhdan06] we compute a 3D indicator function \( g \) defined as \( g(p) = 1 \) at points inside the model, and \( g(p) = 0 \) at points outside, and then obtain the reconstructed surface by extracting an appropriate value.

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 (the indicator function is constant almost everywhere), except at points near the surface, where it is equal to the normal of the 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 solving the problem system, i.e. finding the scalar function \( g \) whose gradient best approximates a vector field \( F \) defined by the samples, i.e. solving \( \nabla g = F \). If we apply the divergence operator, this variational problem transforms into a standard Poisson problem: compute the scalar function

\[ \nabla \cdot g = \text{div}(F) \]

Figure 1: Animate Visualization of Poisson reconstruction in 3D.

\( g \) whose Laplacian (divergence of gradient) equals the divergence of the vector field \( F \).

\[ \nabla^2 g = \nabla \cdot \nabla g = \text{div}(F) \]

We will make three definitions precise in Sections 2 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 functions (RBF) approaches, Poisson reconstruction creates very smooth surfaces that robustly approximate noisy data. But, whereas ideal RBFs are globally supported and not-decaying, the Poisson problem admits a hierarchy of locally supported functions, and therefore the solution induces a well-conditioned sparse linear system.
...SOME REALLY AREN’T!

- Point cloud
- Meshes
- Implicit
- Parametric surface
...SOME REALLY AREN'T!

Abstract

We present a novel algorithm, called marching cubes, that extracts triangle meshes of continuous scalar surface from 3D density data. Unlike a direct-mapping approach to parametrization in the traditional marching cubes, we create a new table that defines triangle mapping. The algorithm preserves the 3D medical data in a single order and integrates triangle vertices using linear interpolation. We first generate the marching cubes that represent the surface of the volume. The direct image produced from the generated surface models is the result of matching the information (intensity, labeled data, and gradient information) present in the original 3D data. Other methods preserve the intensity (ITK), magnetic resonance (MRI), and single photon emission computed tomography (SPECT). However, the quality and functionality of marching cubes. We also discuss improvements from image processing and more suitable modeling algorithms.

1. INTRODUCTION.

Three-dimensional surfaces of the anatomy offer valuable medical data, images of these surfaces, constructed from multiple 2D slices of computed tomography (CT), magnetic resonance (MRI), and single photon emission computed tomography (SPECT), help physicians to understand the complex anatomy present in the brain. Incorporation of 3D medical imaging data into clinical care has been real progress, but more often shrinks data volume in the relatively platforms, who sometimes have difficulty visualizing the 3D anatomy.

Researchers have reported the applications of 3D medical images in a variety of fields. The visualisation of complex medical objects and images cannot be separated. Medical images are the key part of medical research. They are often used in the visualization of complex medical objects and images. The visualisation of complex medical objects and images cannot be separated. Medical images are the key part of medical research. They are often used in the visualization of complex medical objects and images.
A SHAPE’S LIFE CYCLE
A SHAPE’S LIFE CYCLE

- Point cloud
- Fine triangle mesh
- Coarse triangle mesh
- Fabricable triangle mesh
- Fabricated object
A SHAPE’S LIFE CYCLE

- Artist quad mesh
- Mesh dilation
- Implicit dilation
- Implicit surface
- Simplified dilation
- Cage for collision detection
A SHAPE’S LIFE CYCLE

CT scan (implicit)

Stress analysis

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.
WHAT’S THE PLAN NOW?

Earlier (and also now)

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

Now
WHAT'S THE PLAN NOW?

The interesting ones

Read only