

IFT 6112

INTRODUCTION

<http://www-labs.iro.umontreal.ca/~bmpix/teaching/6112/2018/>



Mikhail Bessmeltsev

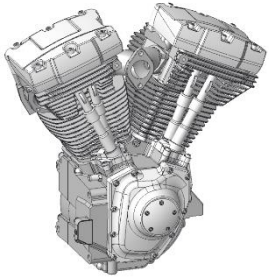
Today

1. Geometric modeling?
2. Shape analysis?
3. Administrative stuff
4. Studying shapes: what we need
5. Applications

Today

- 1. Geometric modeling?**
- 2. Shape analysis?**
3. Administrative stuff
4. Studying shapes: what we need
5. Applications

WHAT IS GEOMETRIC MODELING?



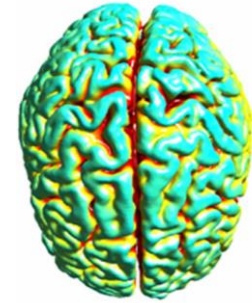
Mechanical Engineering



Fabrication



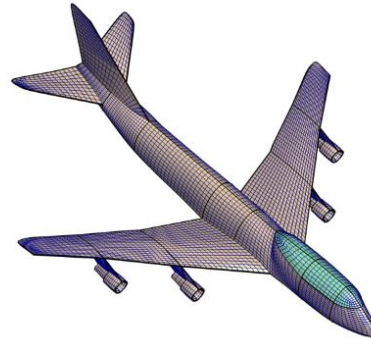
Architecture and civil engineering



Medical



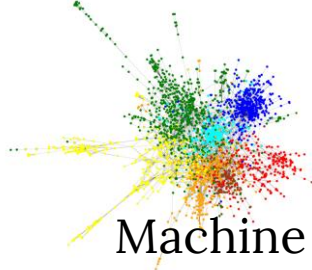
Computer Vision



Geometric Modeling



Computer Graphics



Machine Learning

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } g(x) \leq 0 \\ &\quad h(x) = 0 \end{aligned}$$

Numerical Optimization



Augmented/Virtual Reality

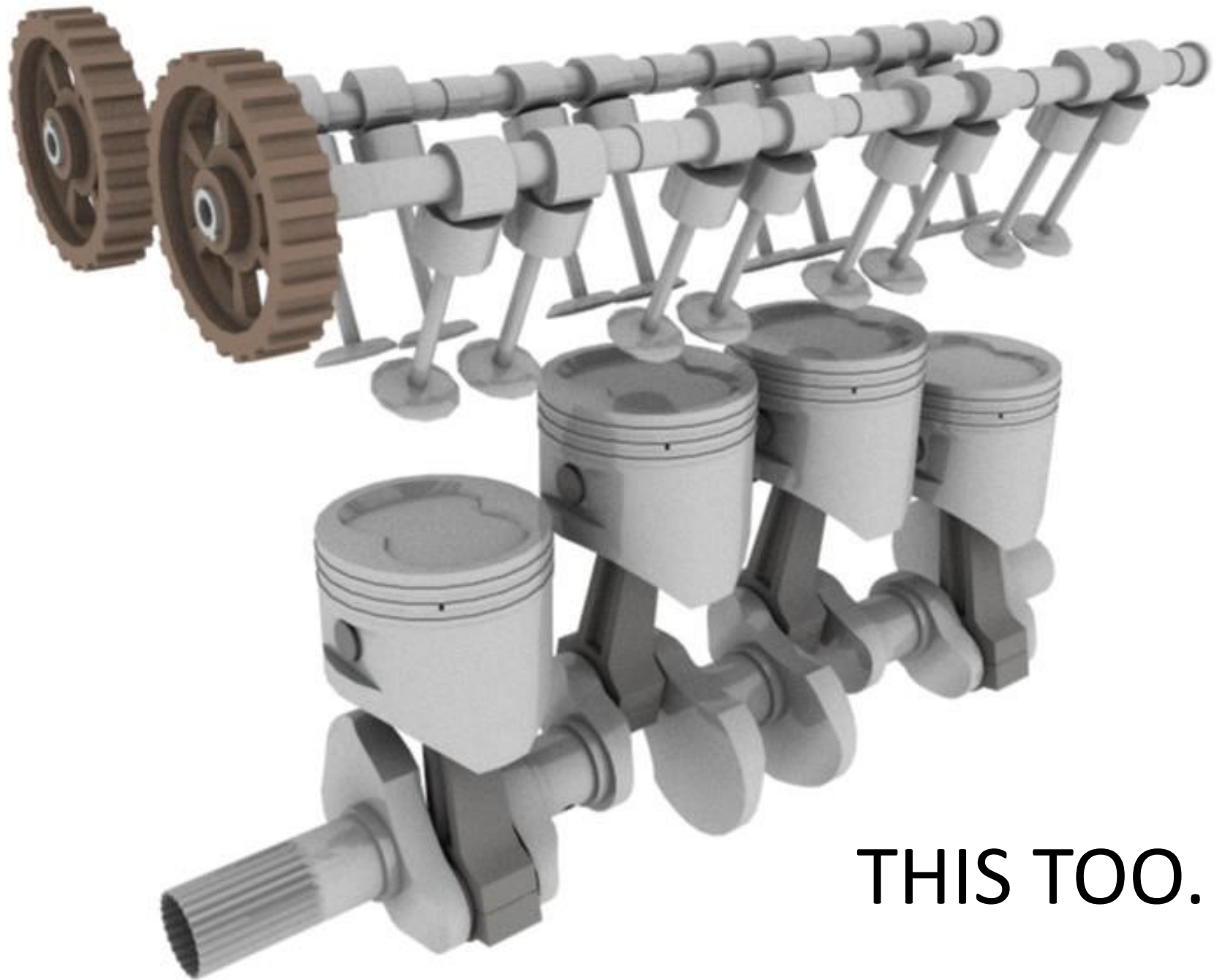


Human-Computer Interaction⁵

THIS.

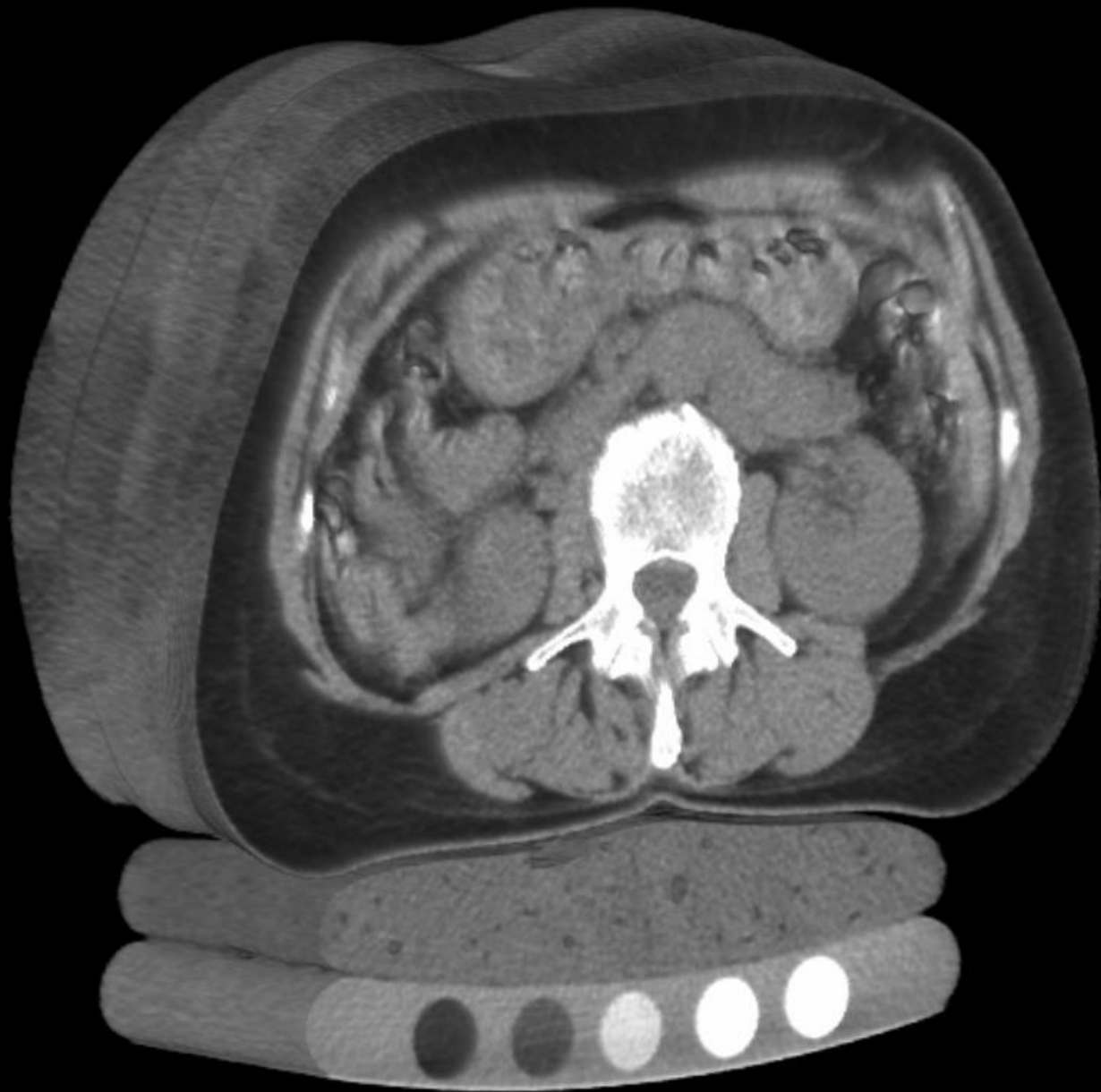
AND THIS.





THIS TOO.

AND EVEN THIS.

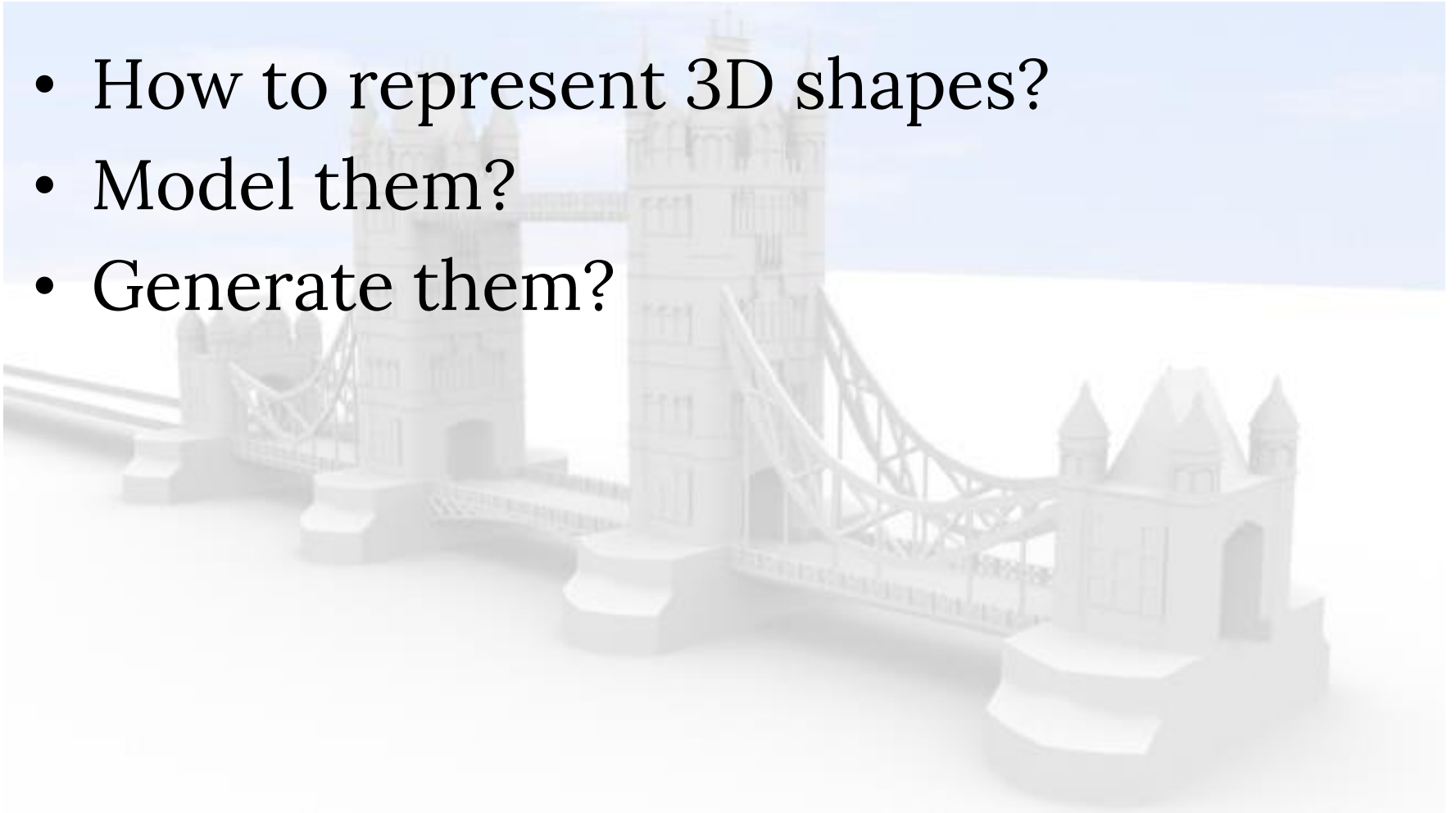


MODELING



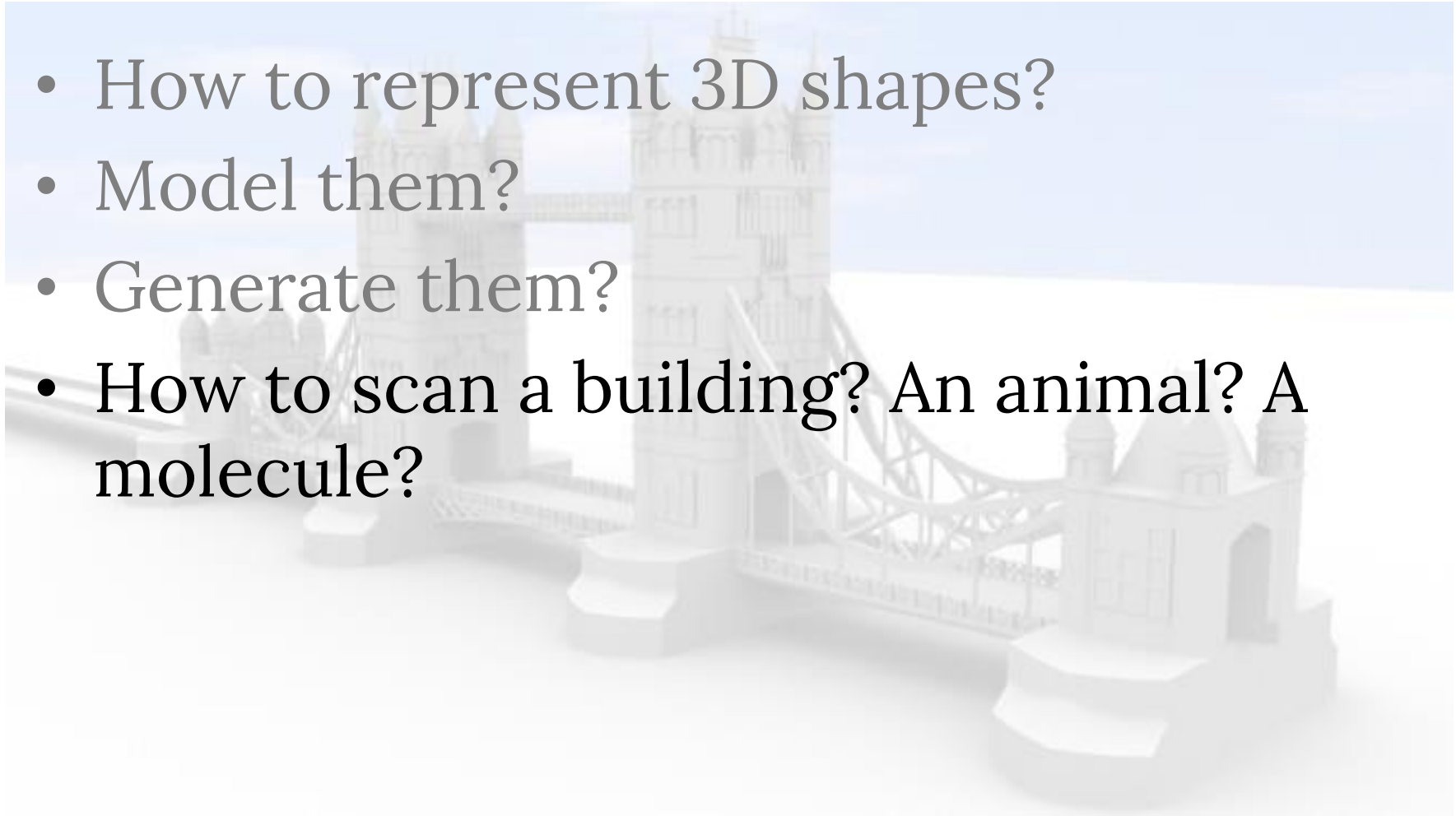
MODELING

- How to represent 3D shapes?
- Model them?
- Generate them?



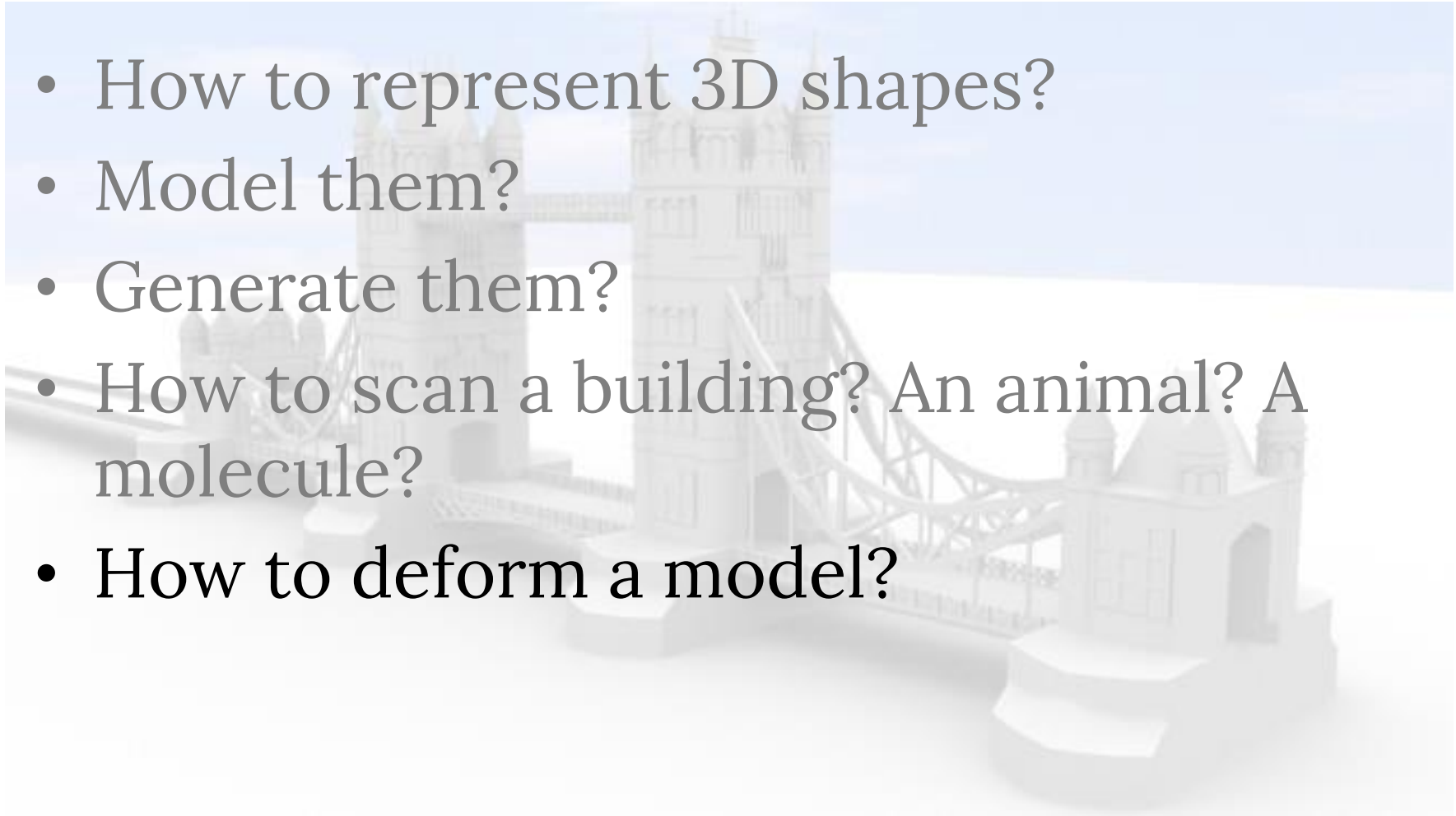
MODELING

- How to represent 3D shapes?
- Model them?
- Generate them?
- How to scan a building? An animal? A molecule?



MODELING

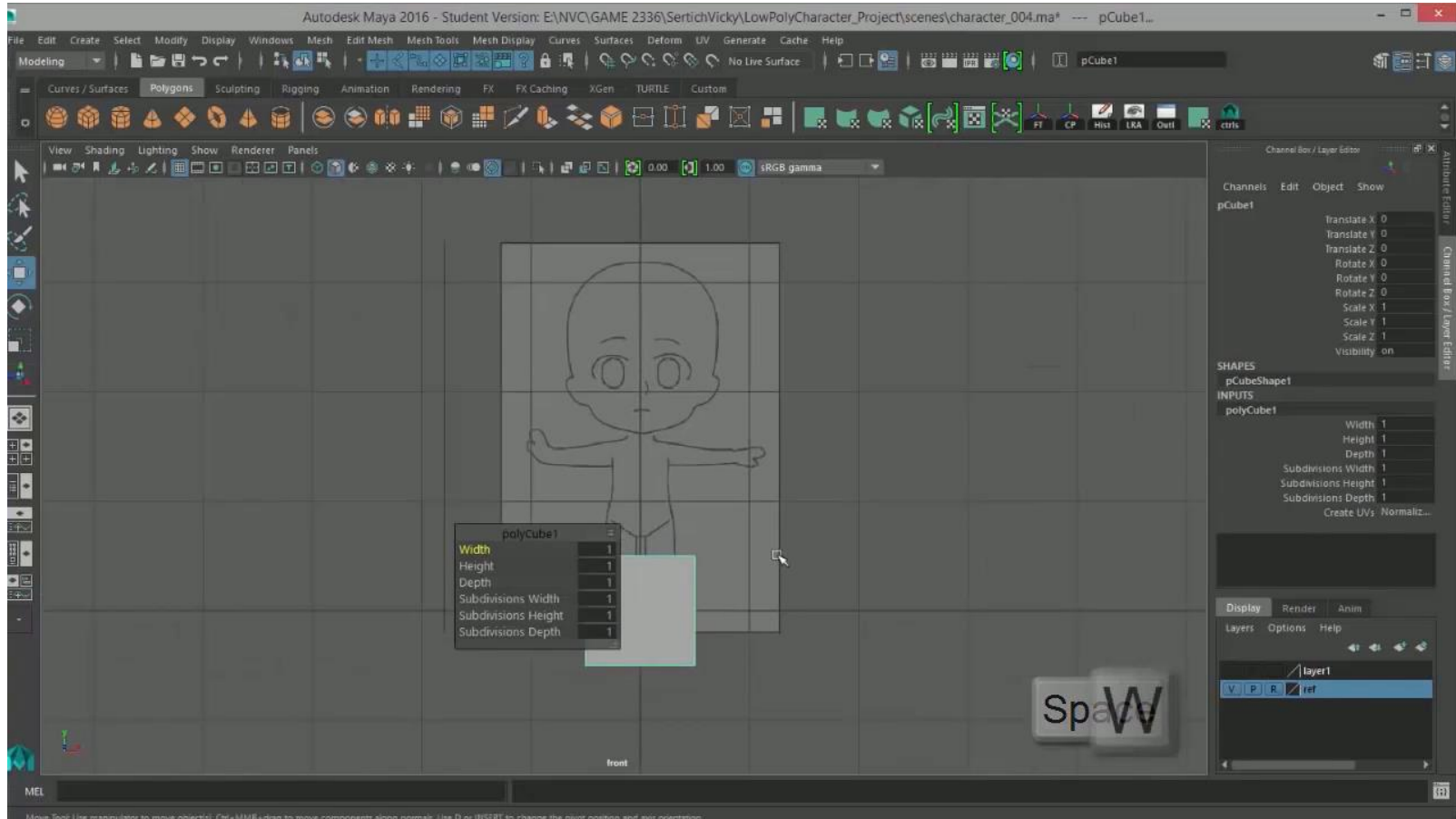
- How to represent 3D shapes?
- Model them?
- Generate them?
- How to scan a building? An animal? A molecule?
- How to deform a model?



MODELING

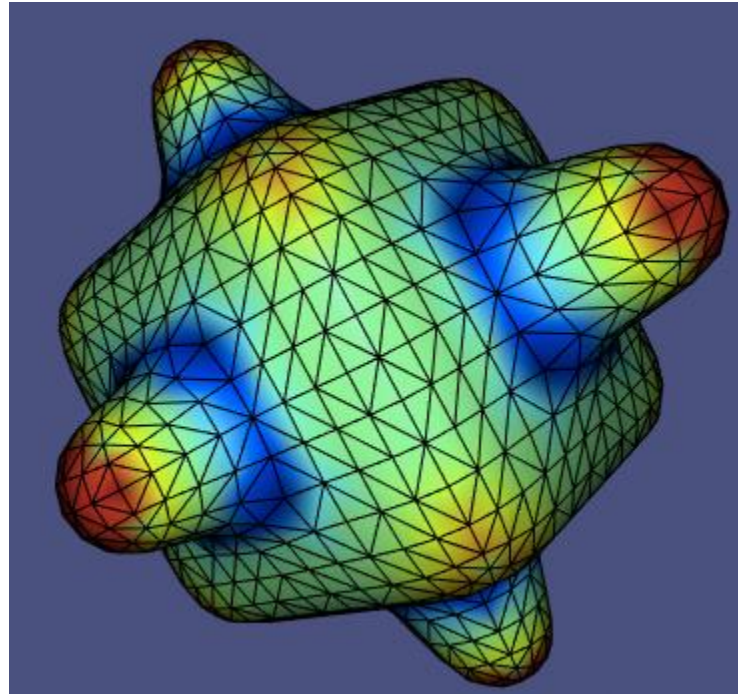
- How to represent 3D shapes?
- Model them?
- Generate them?
- How to scan a building? An animal? A molecule?
- How to deform a model?
- ...

MODELING SESSION



SHAPE ANALYSIS

- Which parts of 3D shape are *interesting*?



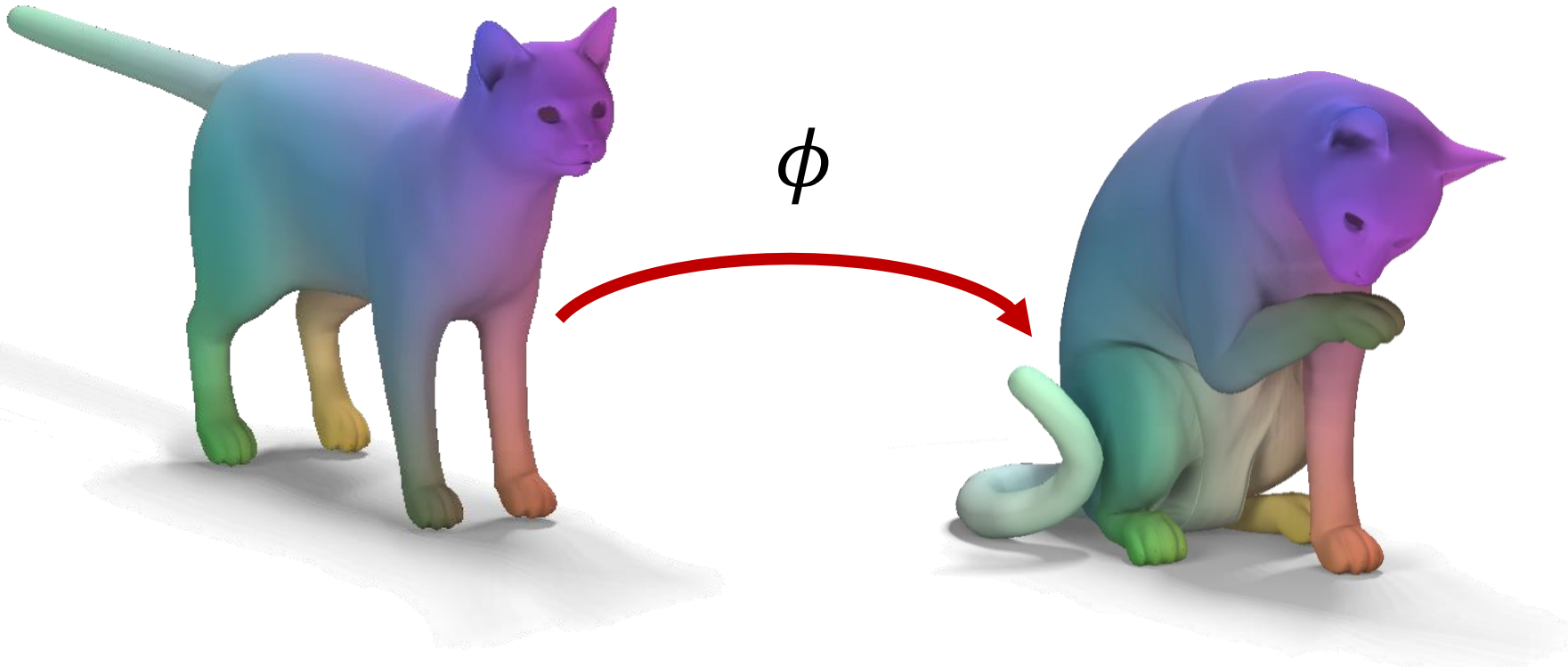
SHAPE ANALYSIS

- How can we *compare* shapes?



SHAPE ANALYSIS

- How can we *compare* shapes?
- How to map them onto each other?



Today

1. Geometric modeling?
2. Shape analysis?
- 3. Administrative stuff**
4. Studying shapes: what we need
5. Applications

FORMALITIES

Lectures: Tue 9.30-11.30 & Thu 14.30-16.30

Office hour: Fri, 1-2 pm AA 6191

and

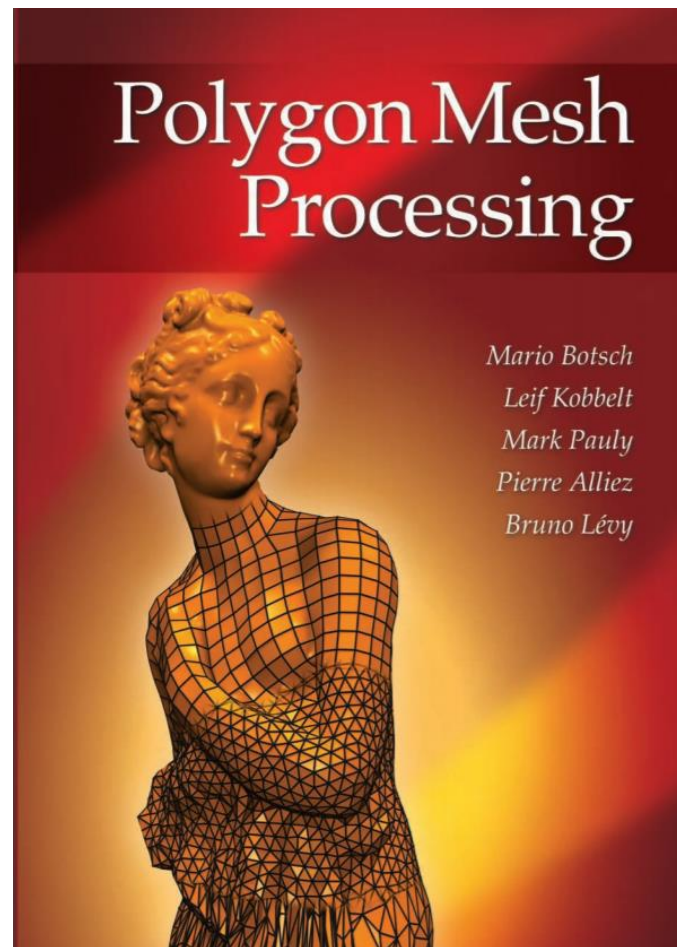
Use Piazza!

<http://piazza.com/umontreal.ca/fall2018/ift6112>

TEXTBOOK (optional)

Polygon Mesh Processing by Botsch et al.

Free online at UdeM library



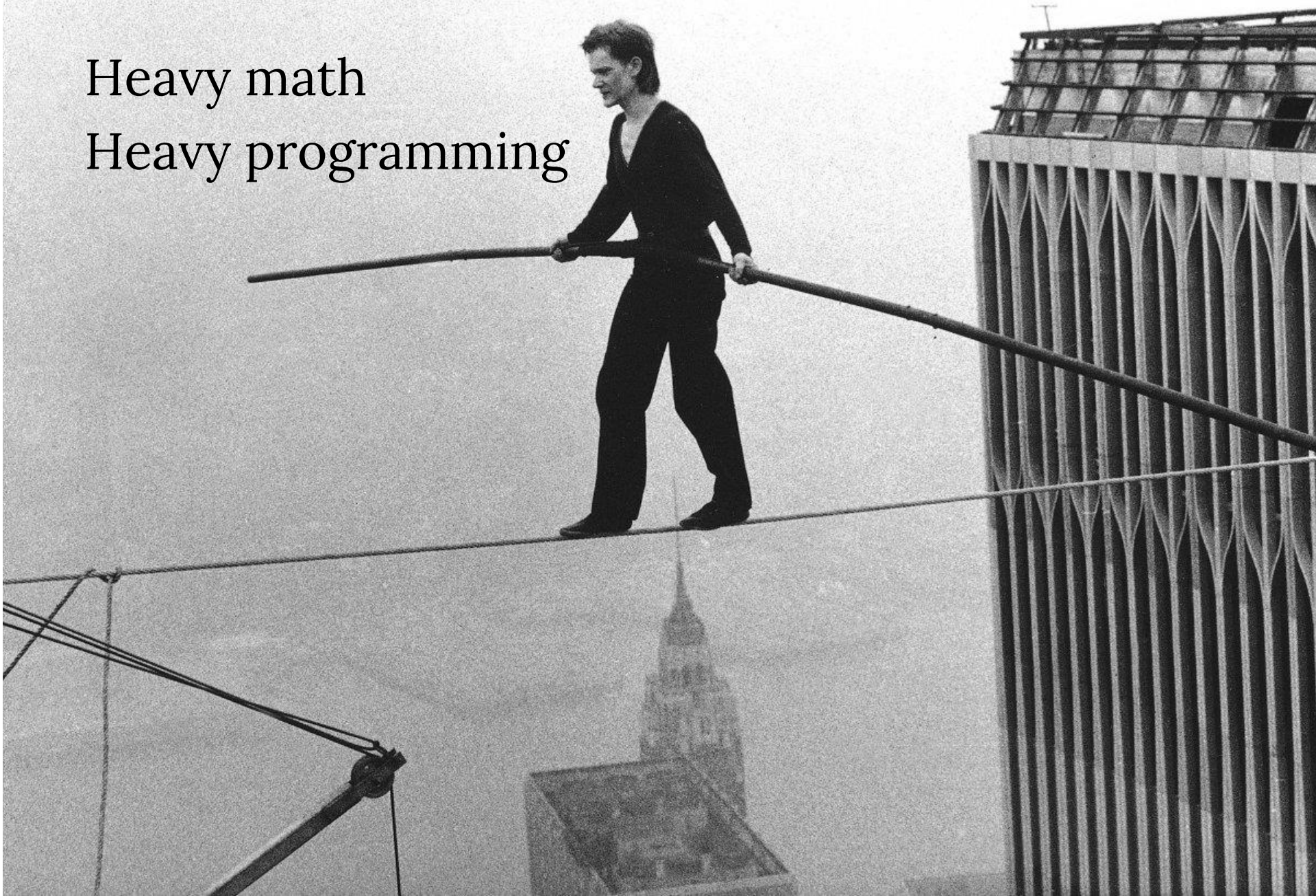
PREREQS

- For sure:
 - Calculus
 - Linear algebra
- More or less:
 - Differential Geometry
 - Partial Differential Equations

THIS IS NOT AN EASY COURSE.

Heavy math

Heavy programming

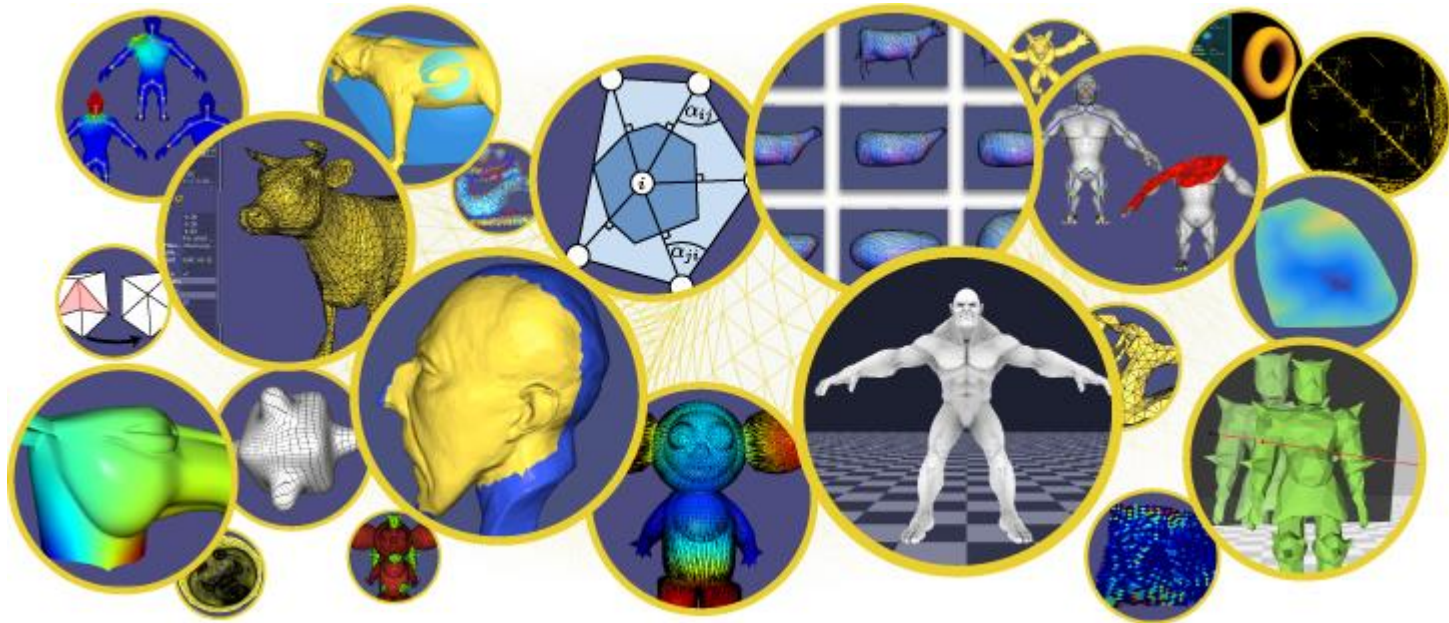


GRADING

Assignments (33%)		100%
Paper Presentation (25%)		
Participation (2%)		
Final Project (40%)		

ASSIGNMENTS (33%)

- Mostly coding, but math also
- ~ 2 weeks long
- Recommended framework: libigl (C++)



<http://libigl.github.io/libigl/>

FACE TO FACE GRADING

- Understand each line of your code
- Plagiarism policy

PAPER PRESENTATION (25%)

- Choose a SIGGRAPH paper on a related topic
 - Give a talk!
 - Others: ask questions
-
- Choose a paper (and post about it in Piazza) by **Oct 21st**

PARTICIPATION (2%)

- Classroom
- Questions/discussion at paper and project presentations

FINAL PROJECT (40%)

1. Choose a recent SIGGRAPH paper
2. Propose an improvement/modification
3. Implement
4. Present!

Important dates:

Project proposal: Oct 16th, by midnight

Face-to-face grading: Nov, 29th

Final Presentation: Dec 4th, in class

WHAT YOU WILL LEARN

Fundamentals:

Mesh data structures
Discrete Differential Geometry
Discrete Laplacians
Vector Fields

Applications:

Shape Deformations
Surface Reconstruction
Mesh Parameterization
Remeshing
Animation/Sketch-based modeling/...

WHAT YOU **WON'T** LEARN

- How to use Maya/3D studio
MAX/Zbrush etc.
- Artistic skills

ROADMAP

- Data Structures $\int dx$
- Recap of differential geometry $\iiint dx dy dz$
- Mesh Simplification & Saliency $\iint dx dy$
- Vector Fields $\iiint dx dy dz$
- Mesh Parameterization $\iint dx dy$
- Shape Deformation $\iint dx dy$
- 2D modeling $\int dx$
- Remeshing $\iint dx dy$



Today

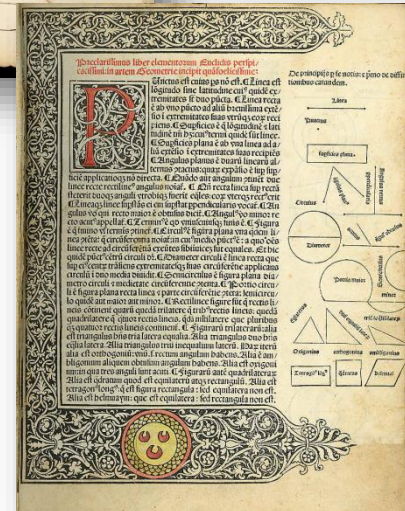
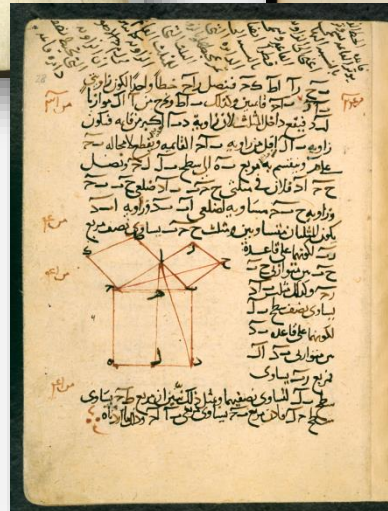
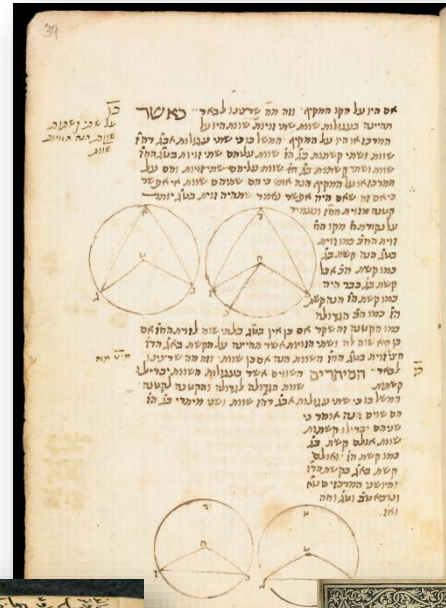
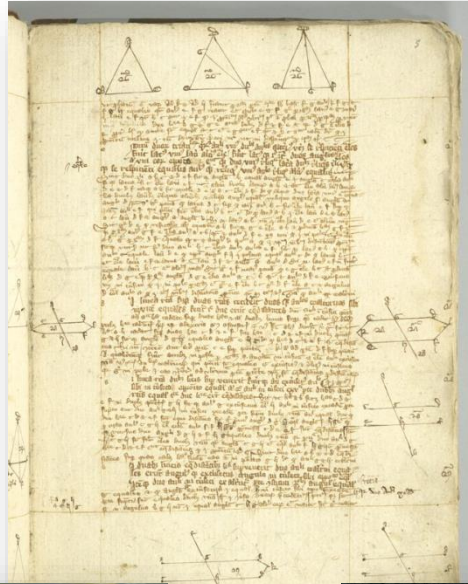
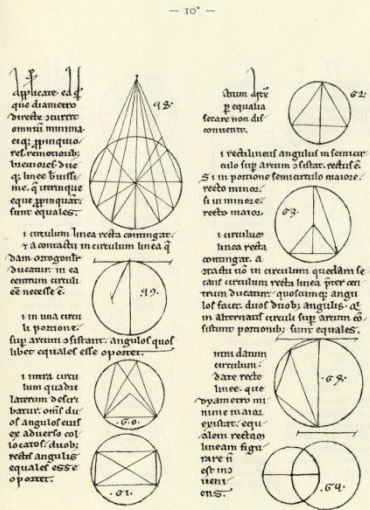
1. Geometric modeling?
2. Shape analysis?
3. Administrative stuff
- 4. Studying shapes: what we need**
5. Applications

We want to study shapes

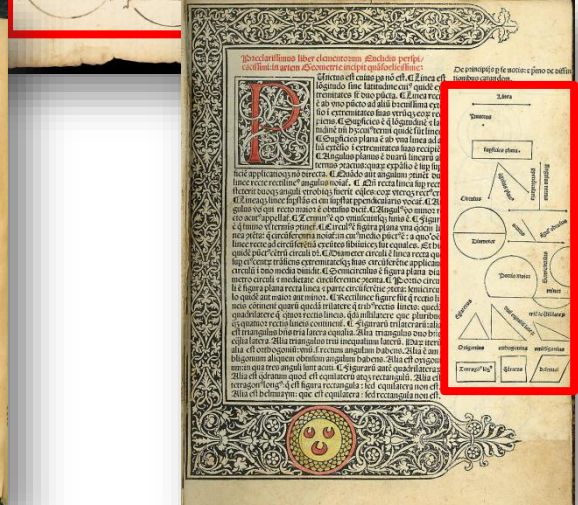
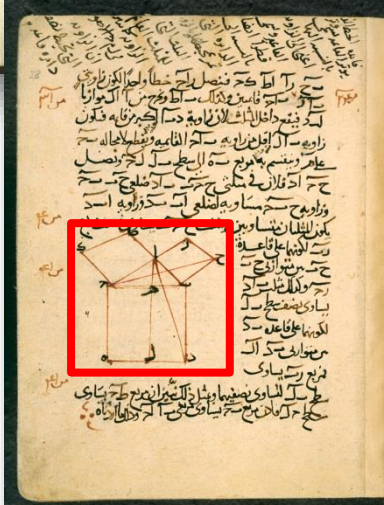
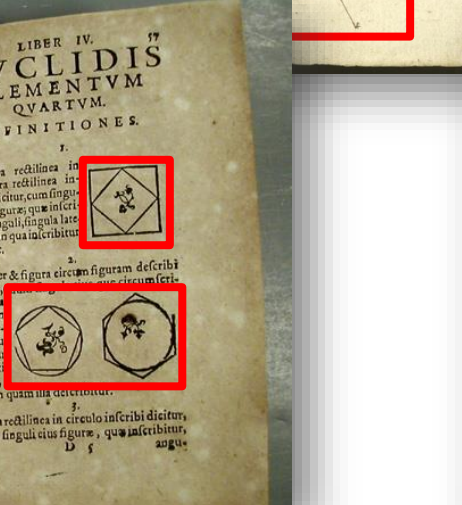
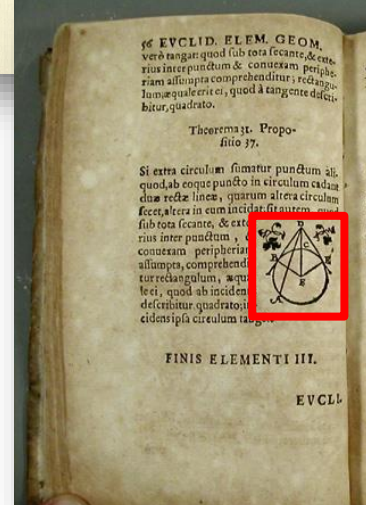
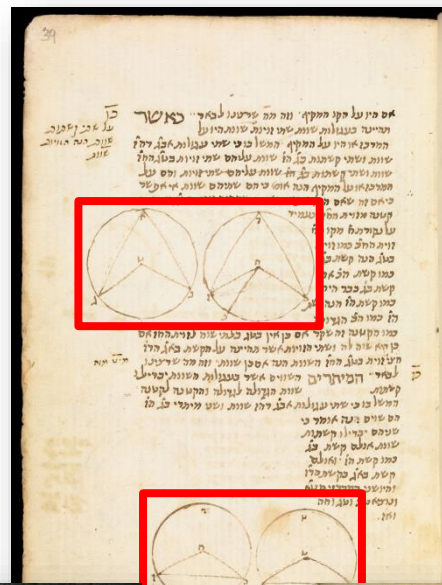
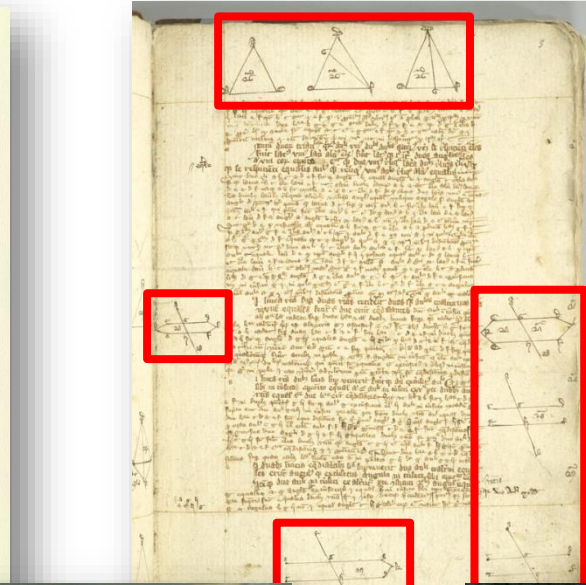
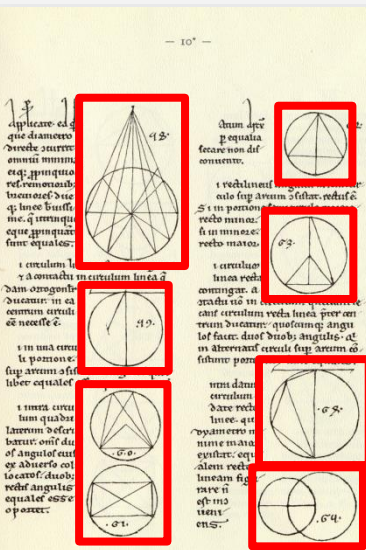
- What (pure) mathematicians think:



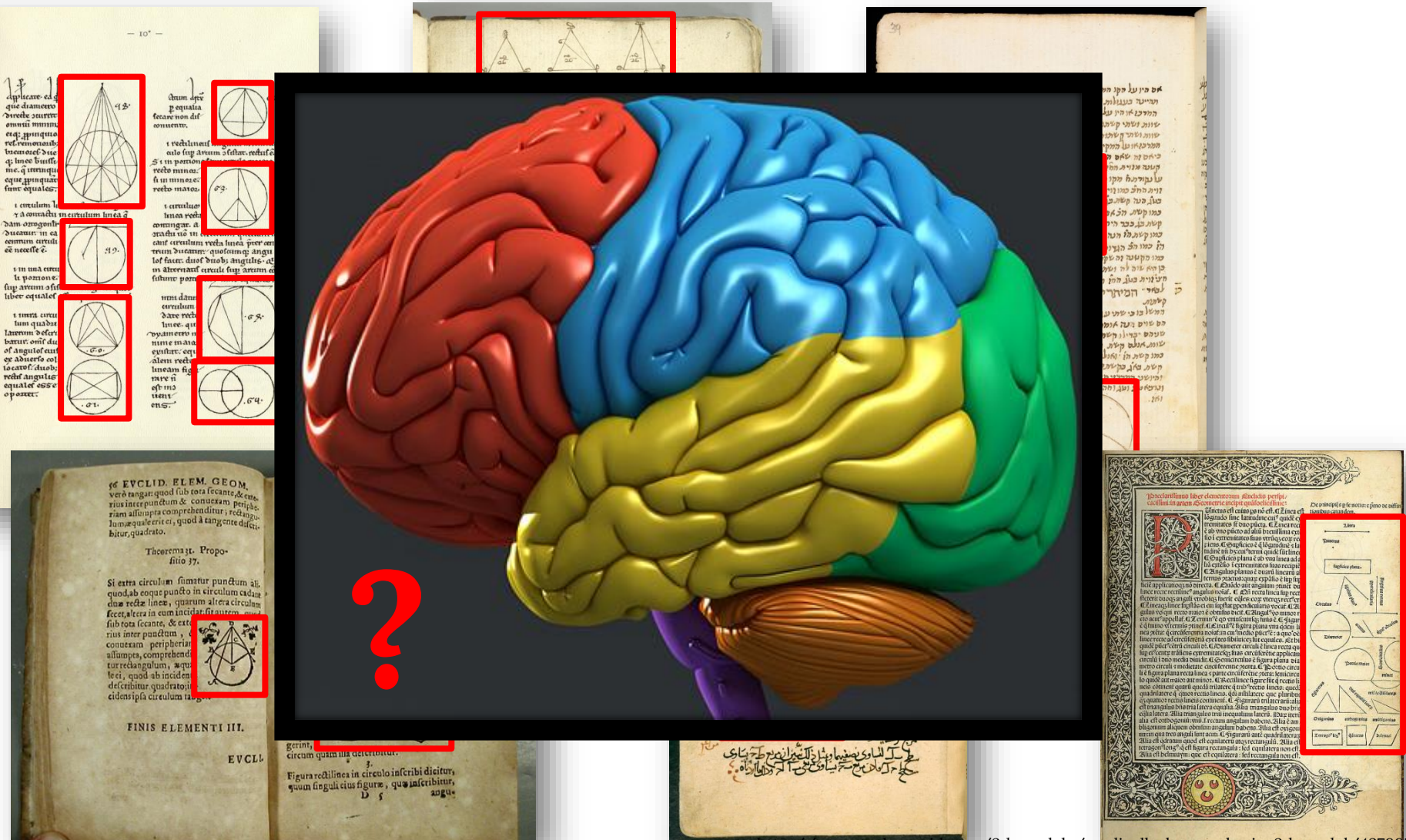
Euclidean Geometry



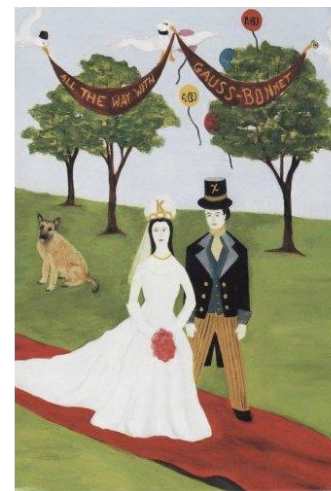
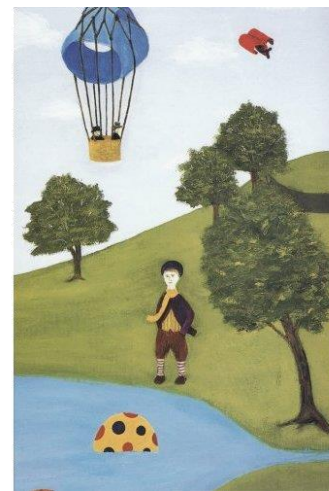
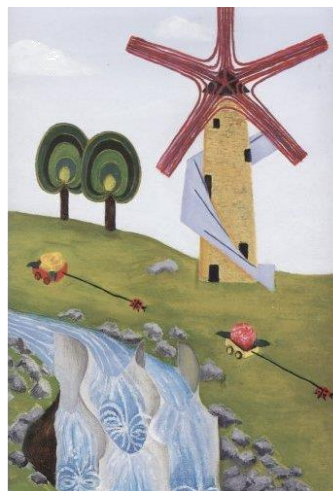
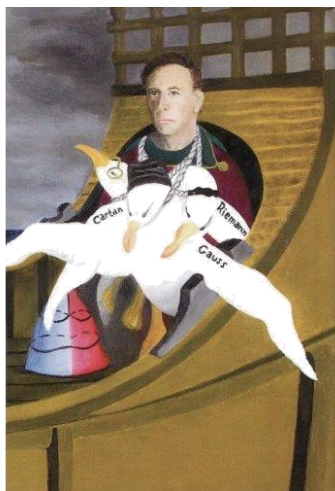
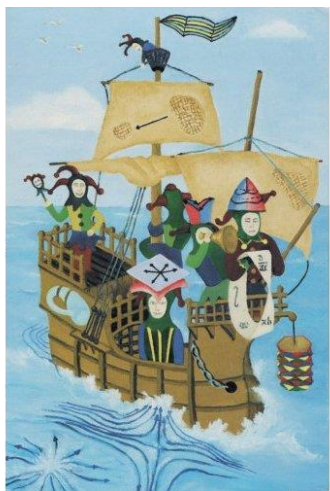
Euclidean Geometry



Euclidean Geometry

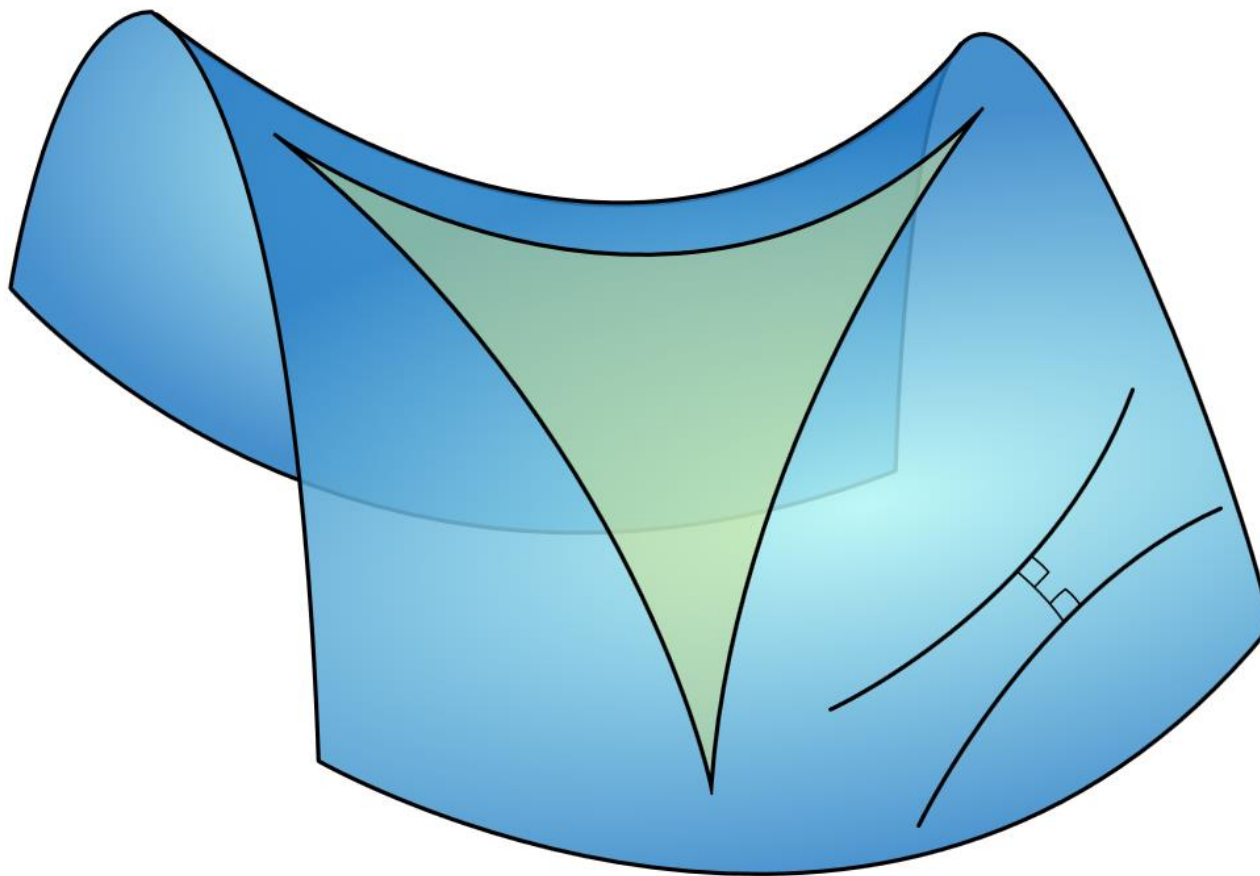


Differential Geometry



Spivak: *A Comprehensive Introduction to Differential Geometry*

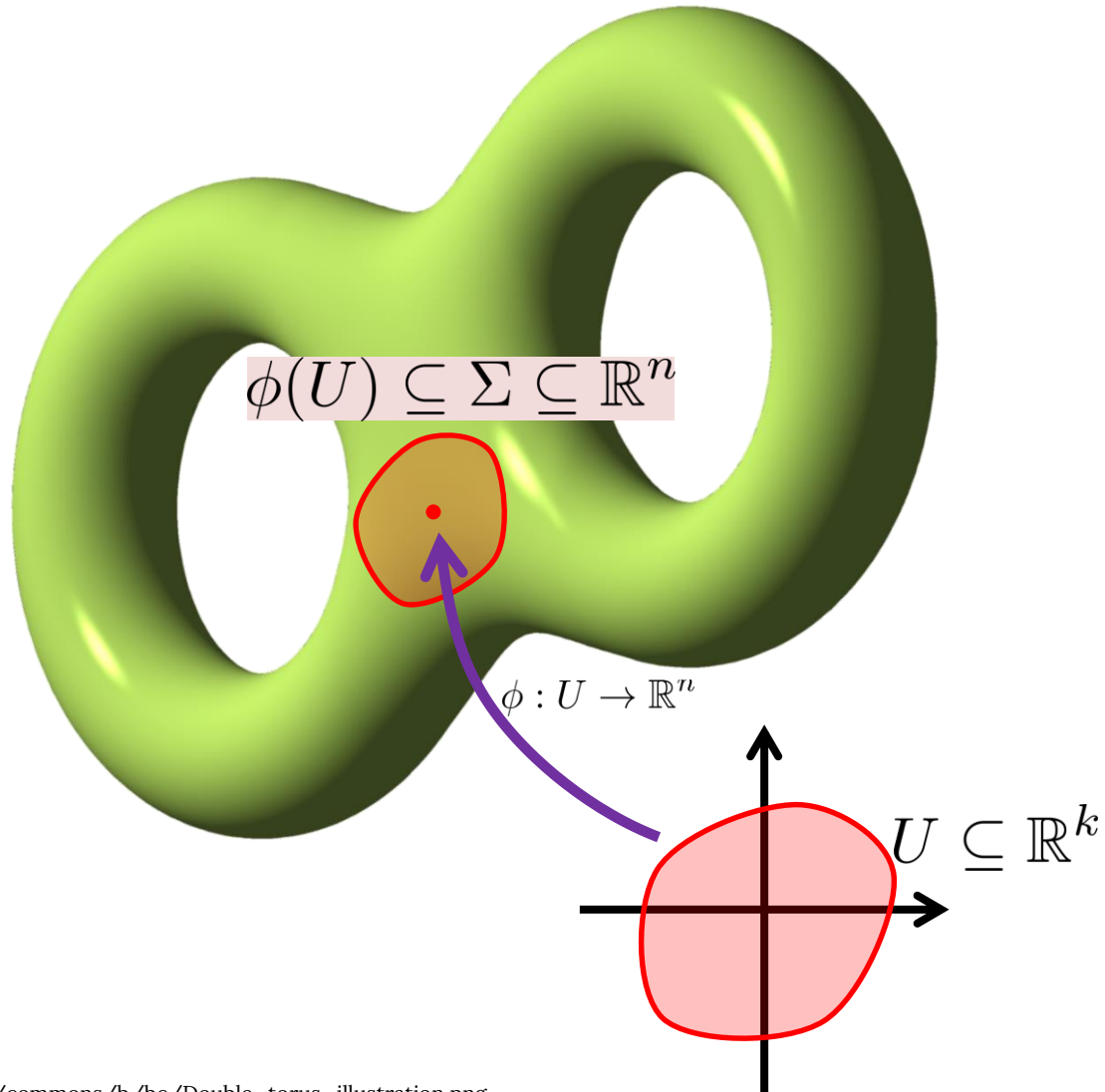
Differential Geometry



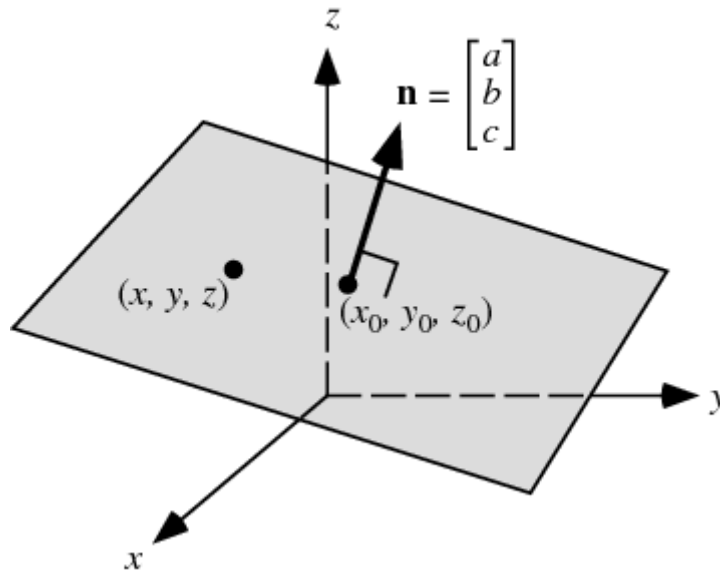
http://en.wikipedia.org/wiki/Differential_geometry

Study of smooth manifolds

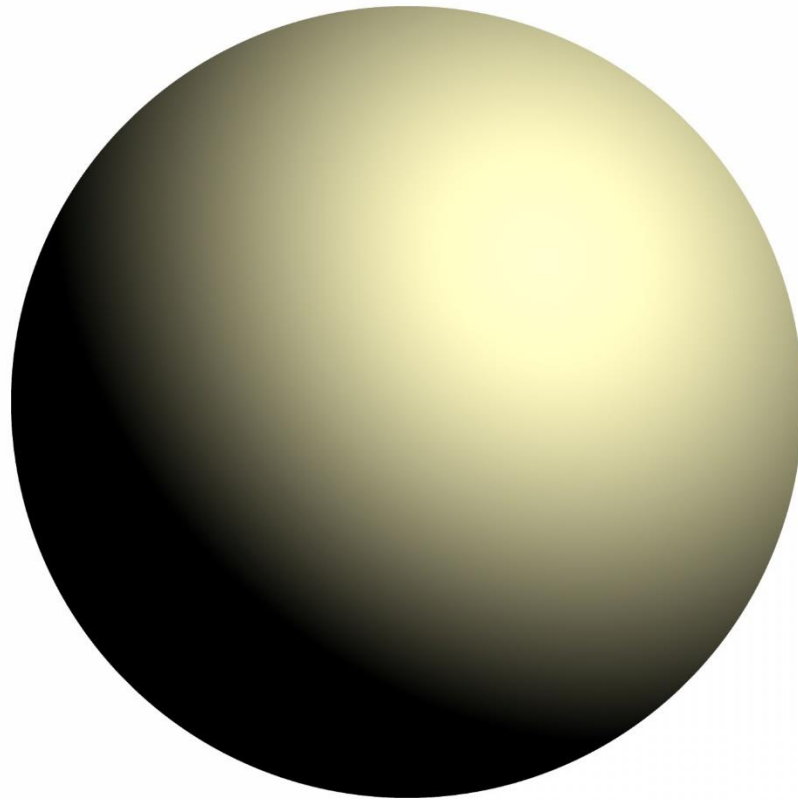
Manifold



Manifold?



Manifold?



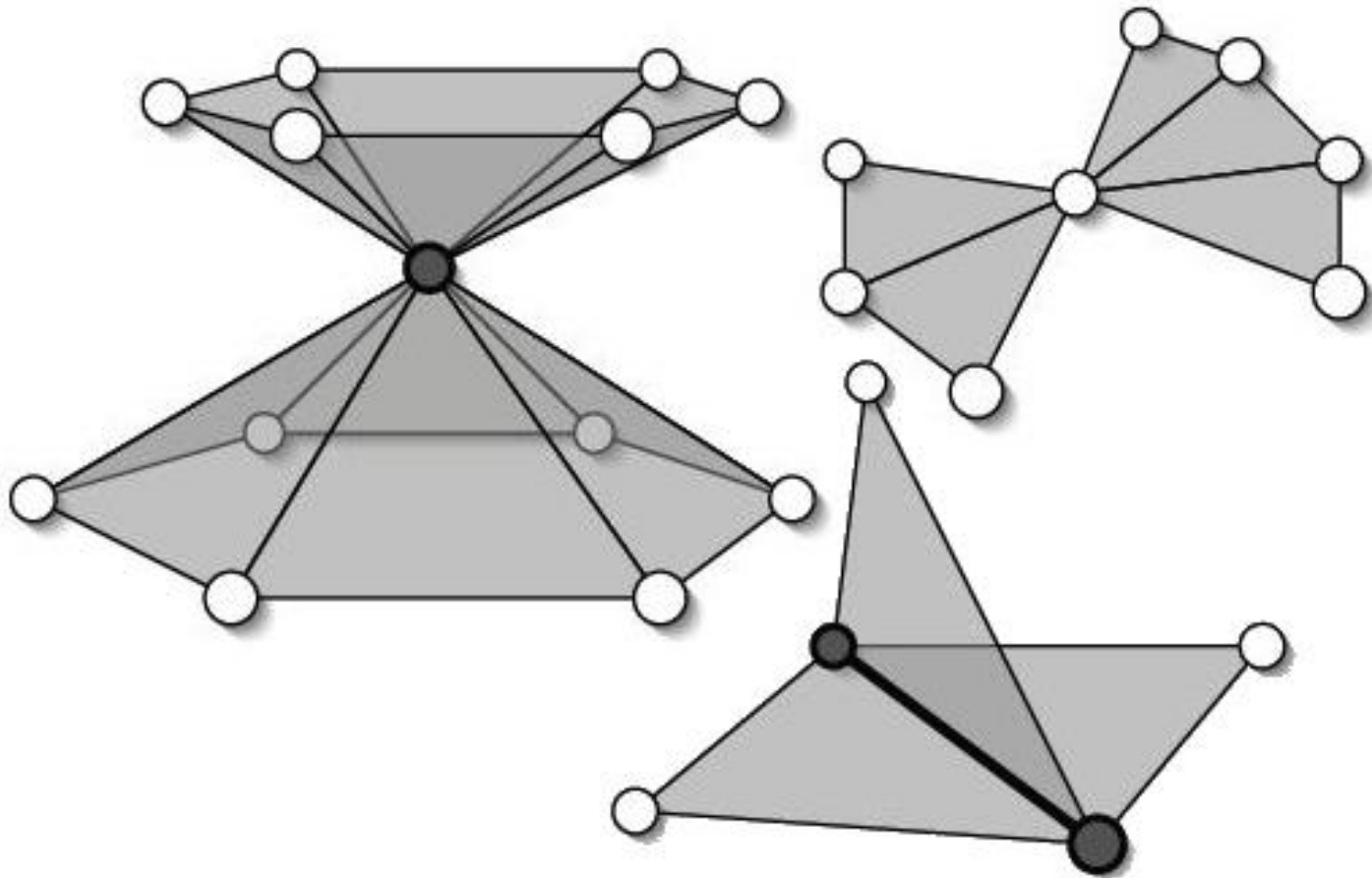
Manifold?



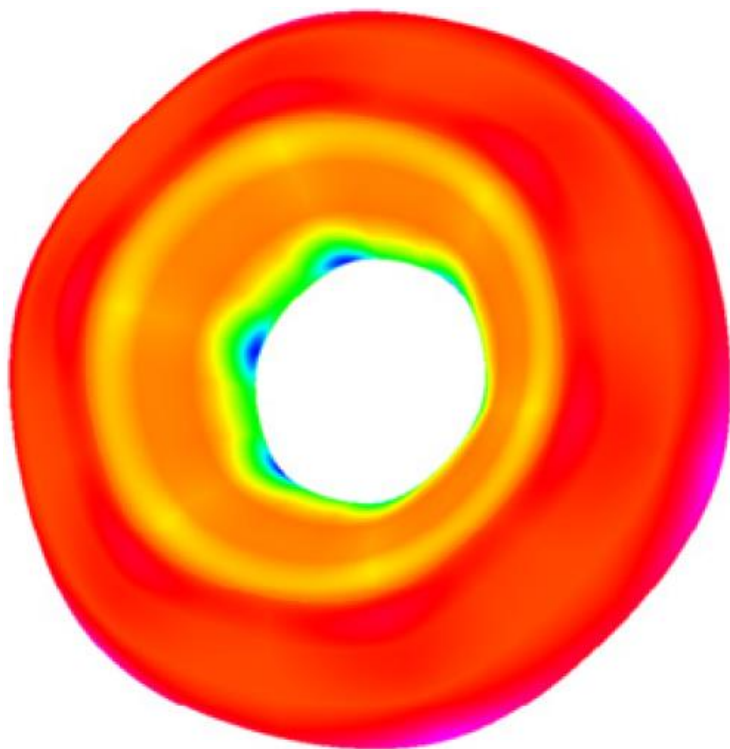
Manifold?



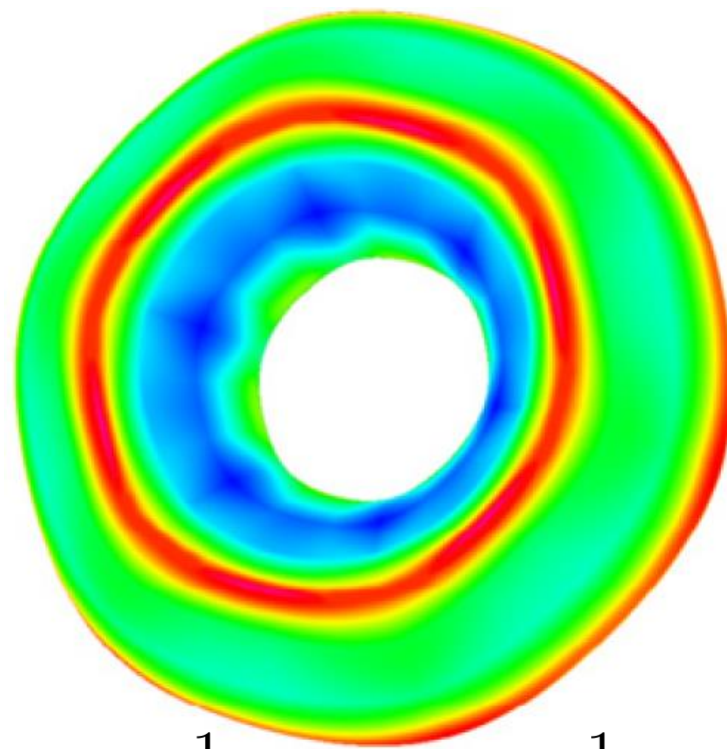
Manifold?



Differential Geometry Toolbox



$$K := \kappa_1 \kappa_2 = \det \mathbb{II}$$



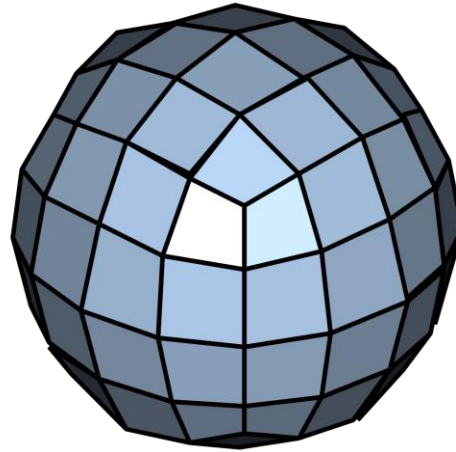
$$H := \frac{1}{2}(\kappa_1 + \kappa_2) = \frac{1}{2} \operatorname{tr} \mathbb{II}$$

<http://www.sciencedirect.com/science/article/pii/S0010448510001983>

Curvature and shape properties

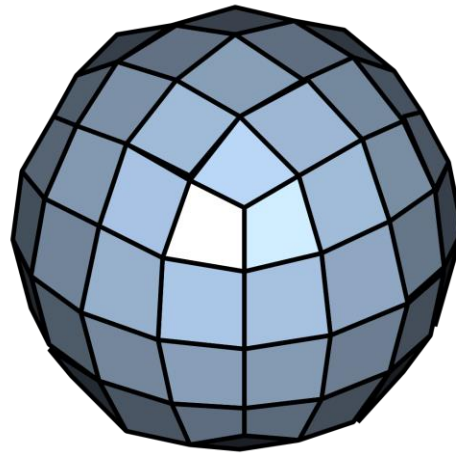
We want to study shapes

- What graphics people think:



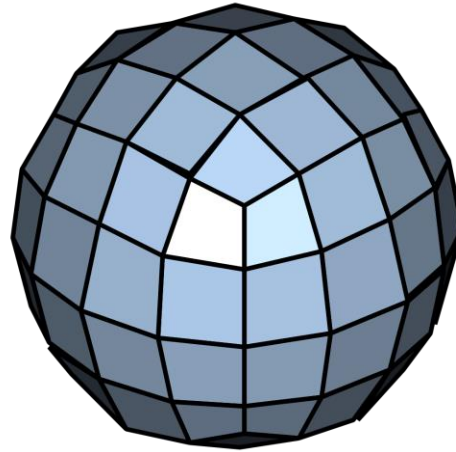
We want to study shapes

- Collection of **flat triangles/quads**
- Approximates a **smooth surface**



We want to study shapes

- Can such a mesh have curvature?



Discrete differential geometry

Structure preservation:

- Keeping properties from the continuous abstraction exactly true in a discretization.

LES GRANDES CONFÉRENCES PUBLIQUES DU CRM
September 12 septembre 2018

Alexander Bobenko (Technische Universität Berlin)

The Discrete Charm of Geometry

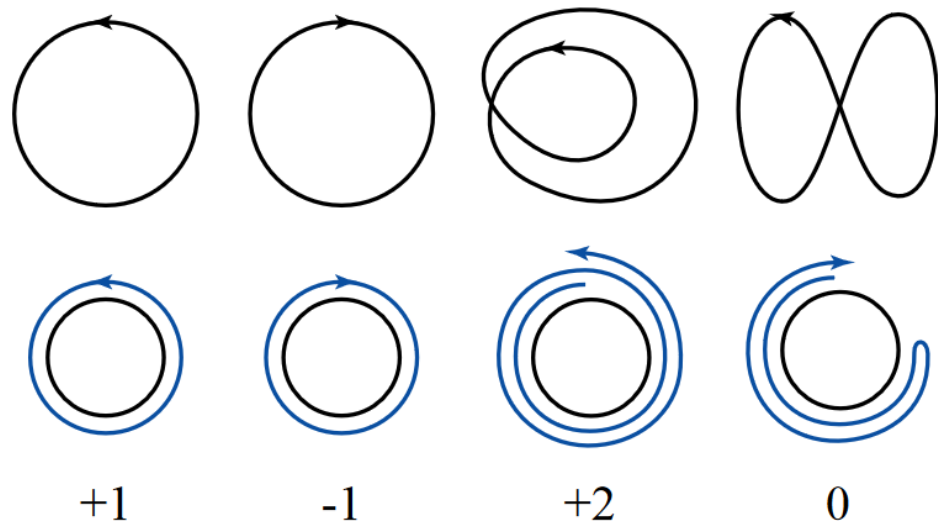


RÉSUMÉ/ABSTRACT:

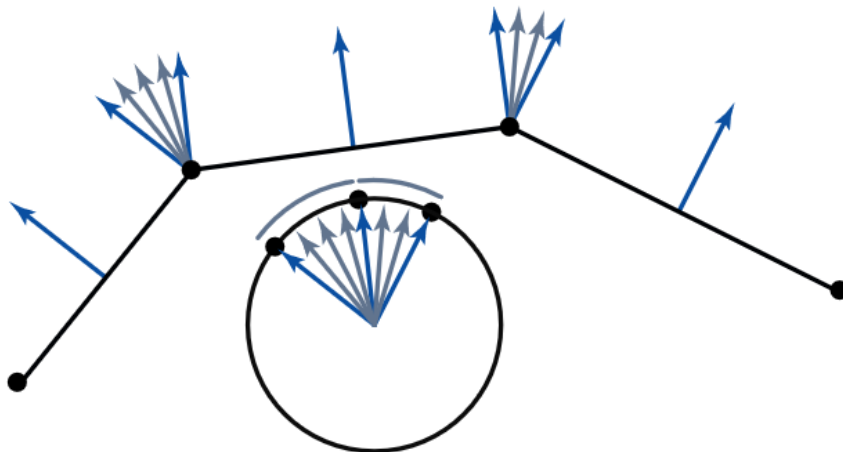
Discrete geometric structures (points, lines, triangles, rectangles, polygons, etc.) are ubiquitous in everyday life, ranging from visible sophisticated freeform structures in contemporary architecture to hidden geometric algorithms in computer generated imagery. On the other hand, discrete geometric structures have also proven to be very useful in modeling and approximating continuous shapes (e.g. curves and surfaces) and real processes. This lecture provides a non-technical and pictorial introduction to the foundations of a new branch of mathematics which underpins these real world situations.

By way of simple and concrete examples, we will illustrate the paradigm of so-called structure-preserving discretizations. These include toy spinning tops, elastic rods, smoke rings and vortex lines in fluids, conformal texture mappings in computer graphics, free form glass and steel structures, and animations from Hollywood movies. We will also show excerpts from our new computer-animated movie entitled "conform!". This film has won Best Experimental Short Film at the Berlin Short Film Festival.

Example: Turning numbers



$$\int_{\Omega} \kappa ds = 2\pi k$$



$$\sum_i \alpha_i = 2\pi k$$

Discrete differential geometry

Eurographics Symposium on Geometry Processing (2007)
Alexander Belyaev, Michael Garland (Editors)

Discrete Laplace operators: No free lunch

Max Wardetzky¹

Saurabh Mathur²

Felix Kälberer¹

Eitan Grinspun² †

¹Freie Universität Berlin, Germany

²Columbia University, USA

the continuous setting. Building on the smooth setting, we present a set of natural properties for discrete Laplace operators for triangular surface meshes. We prove an important theoretical limitation: discrete Laplacians cannot satisfy all natural properties; retroactively, this explains the diversity of existing discrete Laplace operators.

operators for triangular surface meshes. We prove an important theoretical limitation: discrete Laplacians cannot satisfy all natural properties; retroactively, this explains the diversity of existing discrete Laplace operators. Finally, we present a family of operators that includes and extends well-known and widely-used operators.

1. Introduction

Discrete Laplace operators on triangular surface meshes span the entire spectrum of geometry processing applications, including mesh filtering, parameterization, pose transfer, segmentation, reconstruction, re-meshing, compression, simulation, and interpolation via barycentric coordinates [Tau00, Zha04, FH05, Sor05].

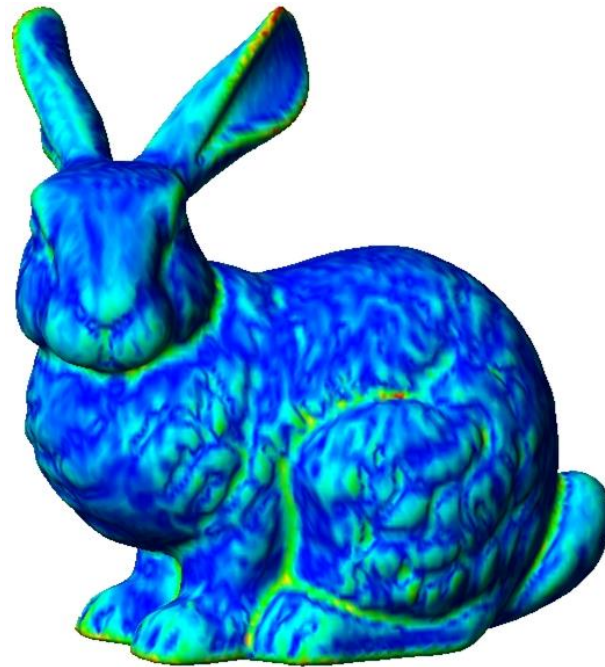
In applications one often requires certain structural prop-

1.1. Properties of smooth Laplacians

Consider a smooth surface S , possibly with boundary, equipped with a Riemannian metric, *i.e.*, an intrinsic notion of distance. Let the intrinsic L^2 inner product of functions u and v on S be denoted by $(u, v)_{L^2} = \int_S uv \, dA$, and let $\Delta = -\operatorname{div} \operatorname{grad}$ denote the intrinsic smooth Laplace-Beltrami operator [Ros97]. We list salient properties of this operator:

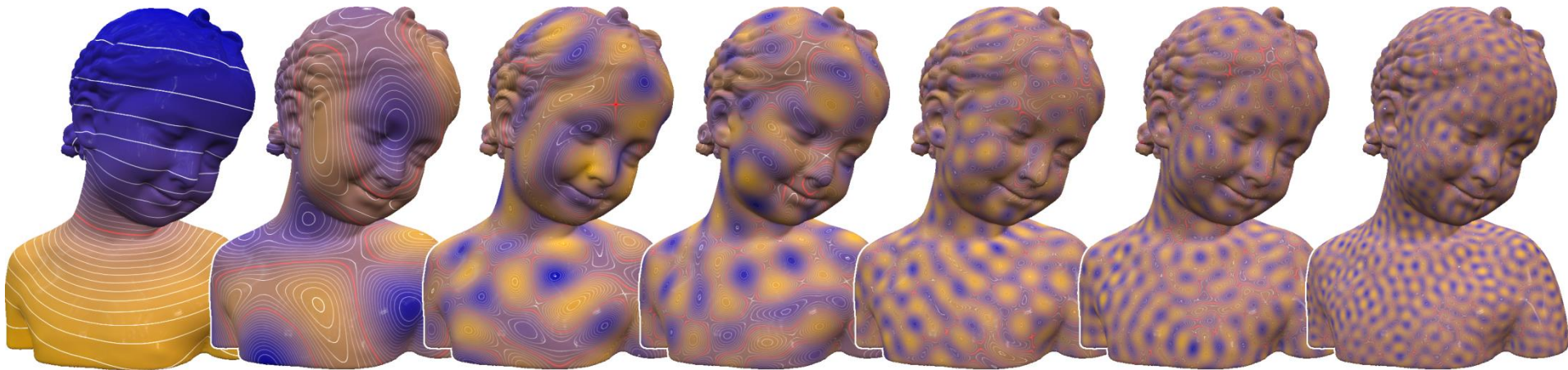
(NULL) $\Delta u = 0$ whenever u is constant.

Theory: grad



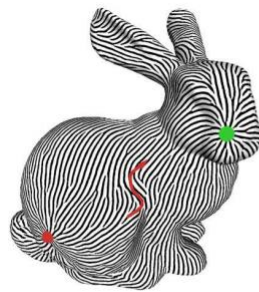
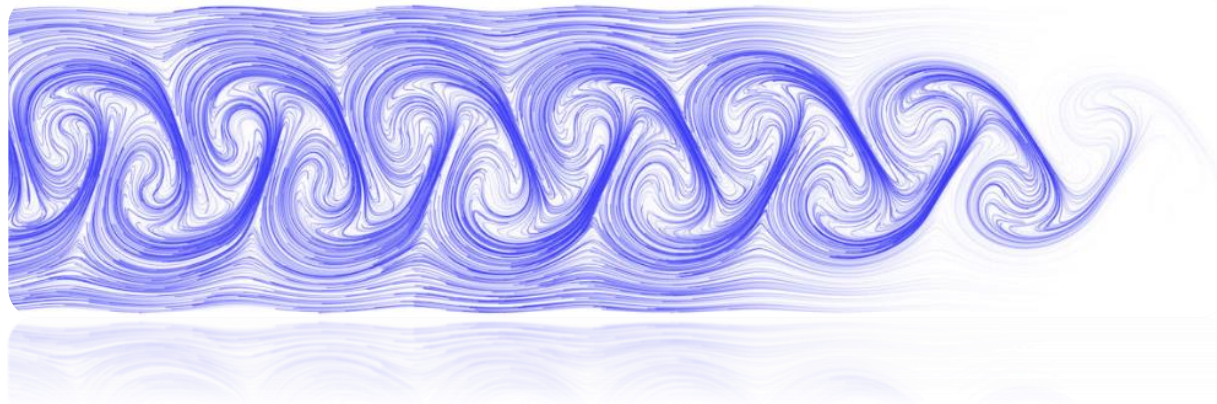
Theory: Laplacian

$$\Delta$$



Vallet and Lévy.
Spectral Geometry Processing with Manifold Harmonics.
EG 2008

Theory: Vector Fields and Flows



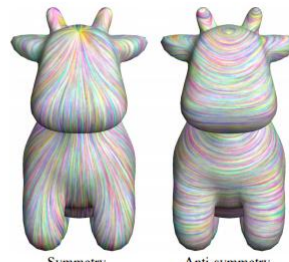
(a) [FSDH07]



(b) [BCBSG10]



(c) [KCPS13]



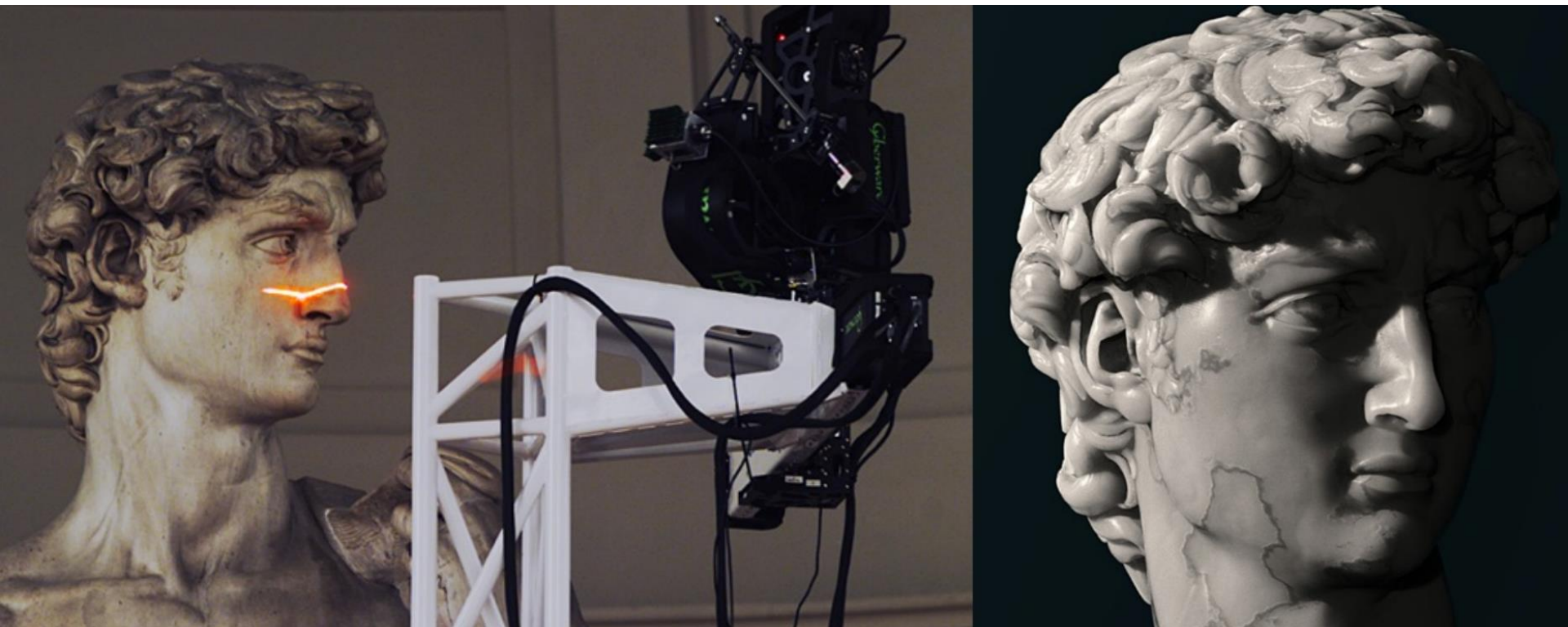
(d) [ABCCO13]

Vaxman et al.
Directional field synthesis, design, and processing. EG STAR 2016.

Today

1. Geometric modeling?
2. Shape analysis?
3. Administrative stuff
4. Studying shapes: what we need
- 5. Applications**

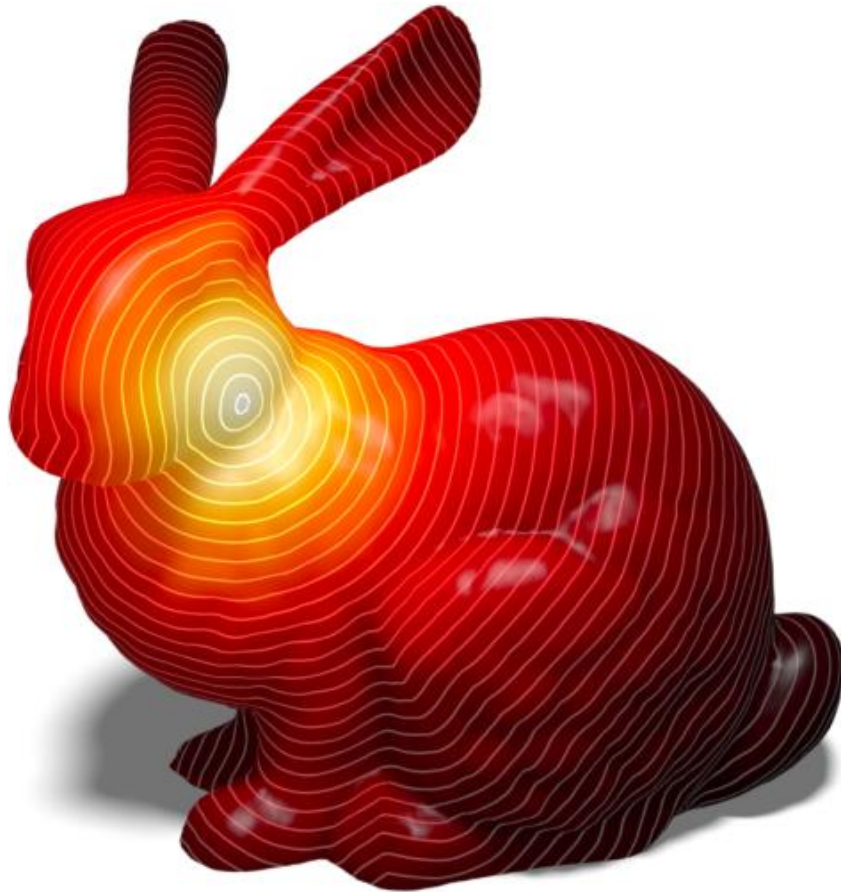
Application: 3D Reconstruction



Applications: Mesh Deformation

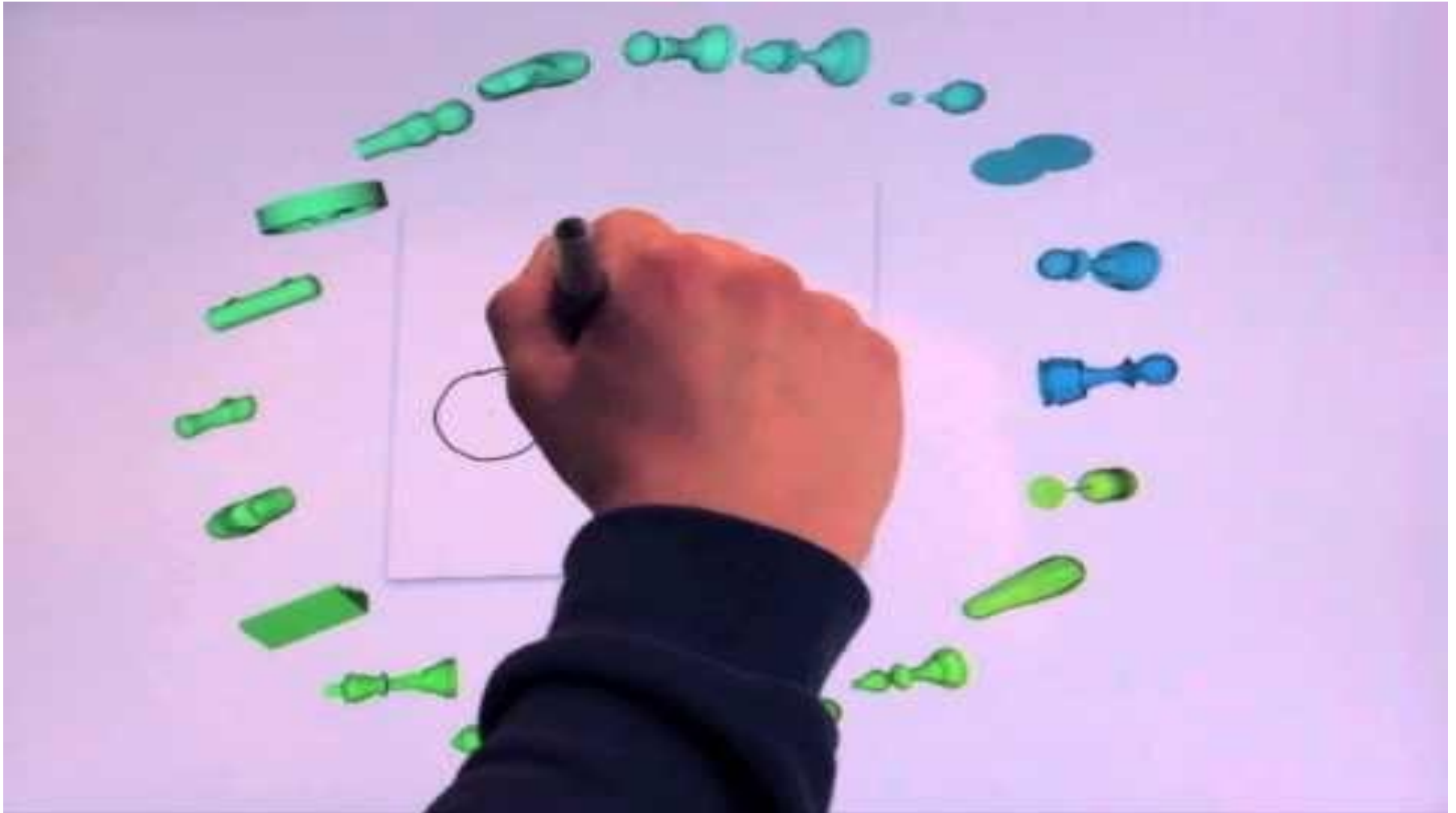


Applications: Heat and Geodesics

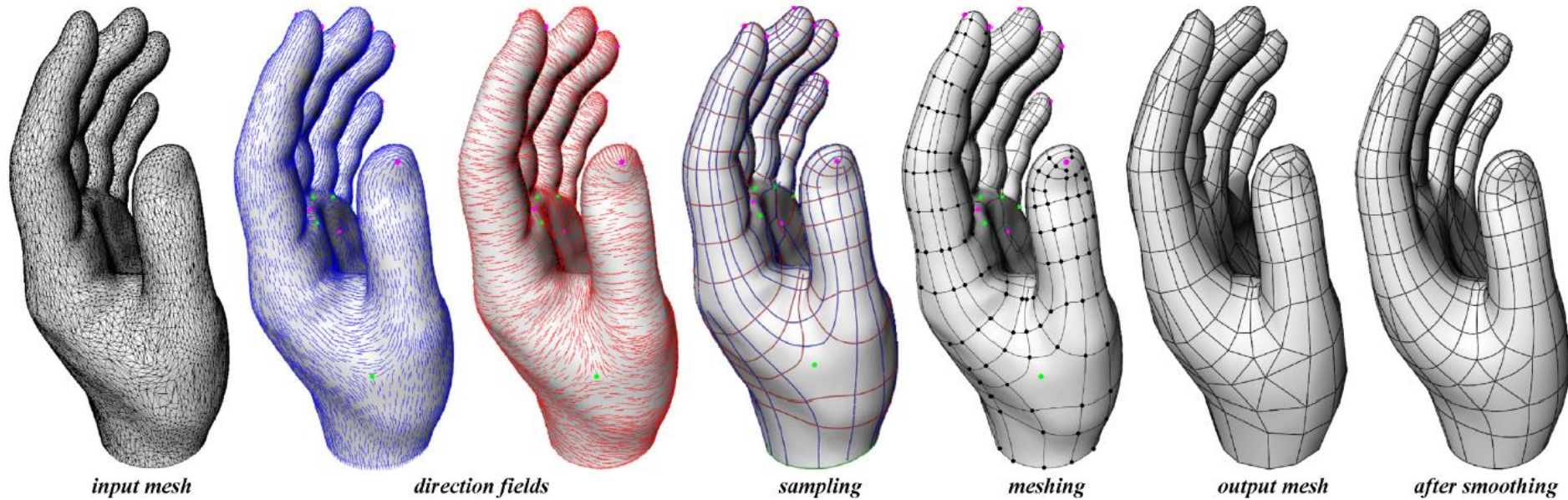


Crane, Weischedel, Wardetzky.
Geodesics in heat. TOG 2013.

Applications: Retrieval



Applications: Remeshing

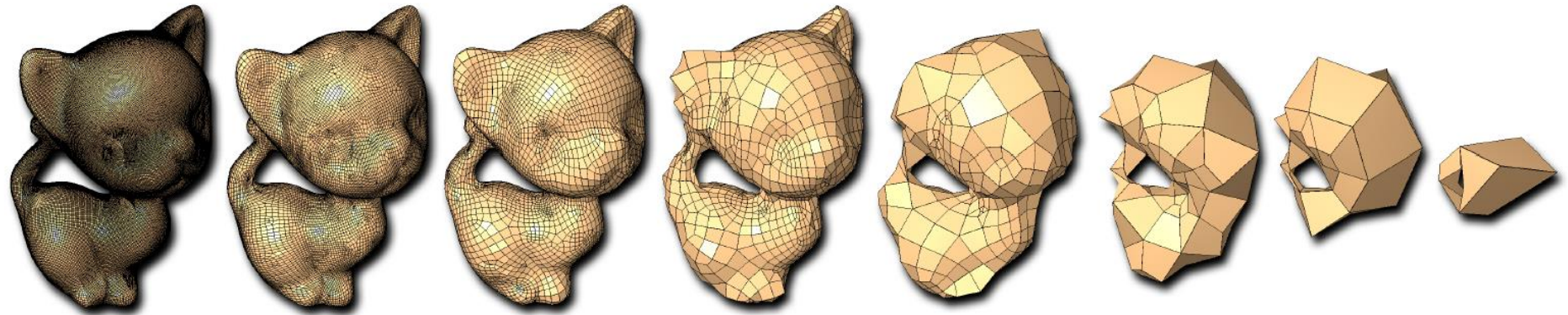


Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. 2003. Anisotropic polygonal remeshing. ACM Trans. Graph. 22, 3 (July 2003)

Applications: QuadMeshing



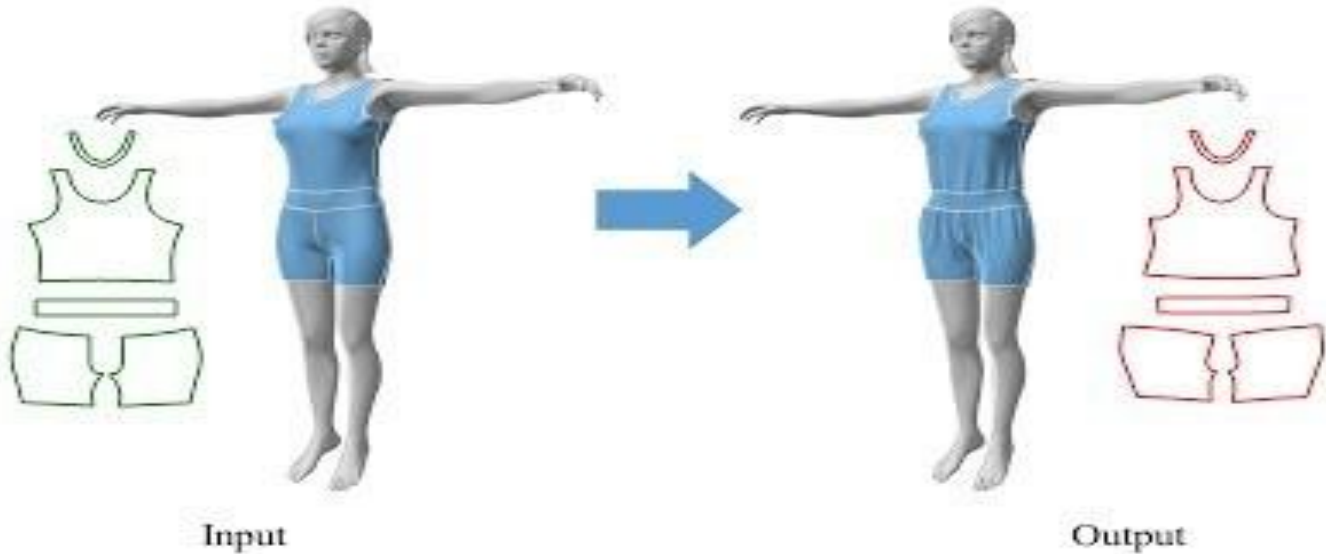
Applications: Mesh Simplification



Joel Daniels, Cláudio T. Silva, Jason Shepherd, and Elaine Cohen. 2008. Quadrilateral mesh simplification. *ACM Trans. Graph.* 27, 5, Article 148 (December 2008)

Applications: Cloth Modeling

FoldSketch



Applications: Sketch-Based Modeling

...perceptual validation

#1 Humans are consistent in perceiving regularity cues in a design sketch.

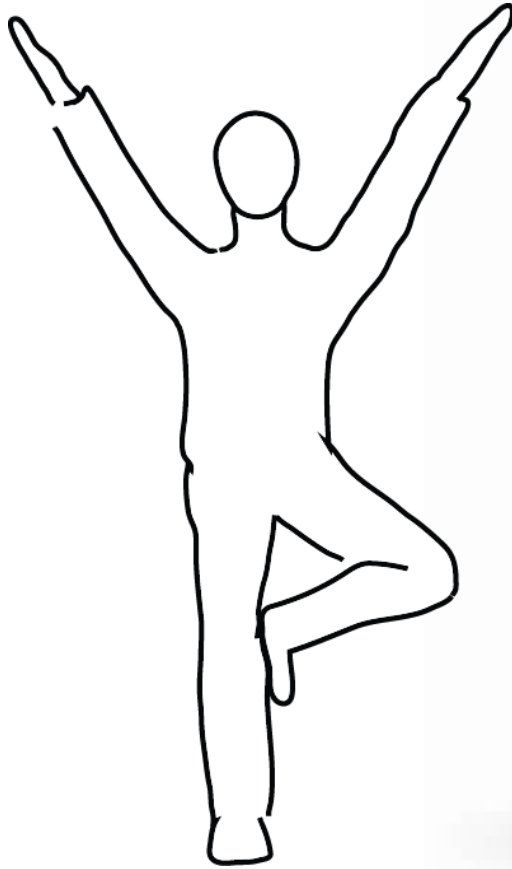
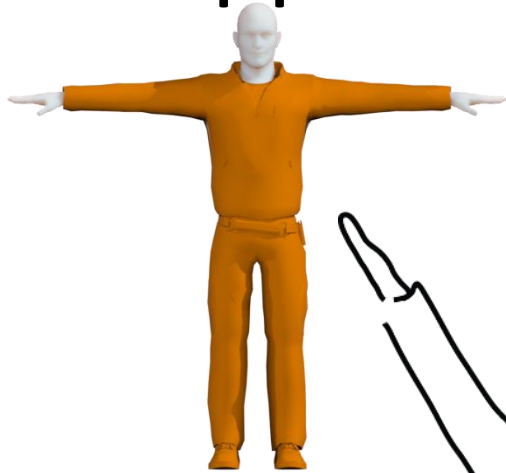


Applications: Sketch-Based Modeling

Input:

rigged character model
+ gesture Drawing

Result



</INTRO>

<http://www-labs.iro.umontreal.ca/~bmpix/teaching/6112/2018/>

Piazza! Piazza! Piazza!

Office hour: Fri, 1-2 pm, AA 6191



libigl & Eigen

- Eigen: a linear algebra library
- libigl: geometry processing library

Eigen

```
#include <iostream>
#include <Eigen/Dense>
using Eigen::MatrixXd;
int main()
{
MatrixXd m(2,2);
m(0,0) = 3;
m(1,0) = 2.5;
m(0,1) = -1;
m(1,1) = m(1,0) + m(0,1);
std::cout << m << std::endl;
}
```

Eigen

```
#include <iostream>
#include <Eigen/Dense>
using namespace Eigen;
using namespace std;
int main()
{
    Matrix3d m = Matrix3d::Random();
    m = (m + Matrix3d::Constant(1.2)) * 50;
    cout << "m =" << endl << m << endl;
    Vector3d v(1,2,3);
    cout << "m * v =" << endl << m * v << endl;
}
```

Eigen: Sparse Matrices

```
#include <Eigen/Sparse>
#include <vector>
#include <iostream>
typedef Eigen::SparseMatrix<double> SpMat; // declares a column-major sparse matrix type of
double
typedef Eigen::Triplet<double> T;

int main(int argc, char** argv)
{
    if(argc!=2) {
        std::cerr << "Error: expected one and only one argument.\n";
        return -1;
    }
    int n = 300; // size of the image
    int m = n*n; // number of unknowns (=number of pixels)
    // Assembly:
    std::vector<T> coefficients; // list of non-zeros coefficients
    Eigen::VectorXd b(m); // the right hand side-vector resulting from the constraints
    buildProblem(coefficients, b, n);
    SpMat A(m,m);
    A.setFromTriplets(coefficients.begin(), coefficients.end());
    // Solving:
    Eigen::SimplicialCholesky<SpMat> chol(A); // performs a Cholesky factorization of A
    Eigen::VectorXd x = chol.solve(b); // use the factorization to solve for the given right hand
    side
    // Export the result to a file:
    saveAsBitmap(x, n, argv[1]);
    return 0;
}
```