

IFT 6113

SKETCH-BASED MODELING

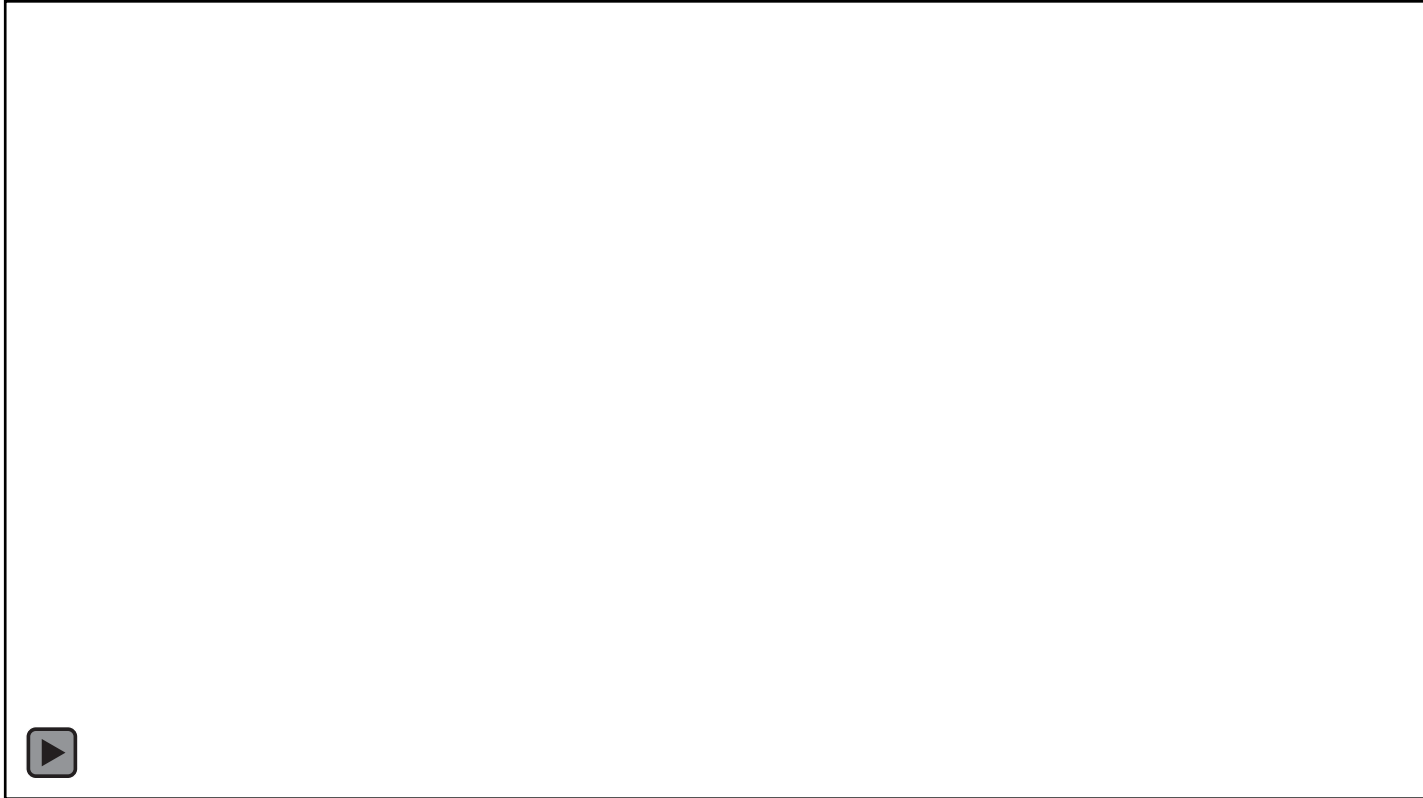
tiny.cc/ift6113



Robust Flow-Guided Neural Prediction for Sketch-Based Freeform Surface Modeling by Li et al., SIGGRAPH ASIA 2018

Mikhail Bessmeltsev

Motivation



30x speed
© ARTV Tutorials

Applications

- Modeling
 - Natural shapes
 - CAD-models
 - Buildings
 - Trees
- Animating
 - Characters
 - Liquids
 - Etc.

Teddy

Takeo Igarashi

Hidehiko Tanaka

University of Tokyo

Satoshi Matsuoka

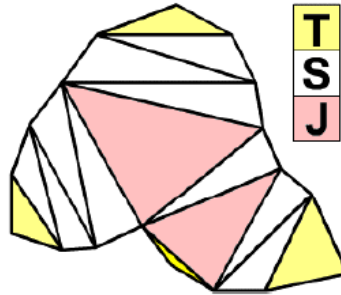
Tokyo Institute of Technology

**Teddy:
A Sketching
Interface for 3D
Freeform Design**

Teddy



a) initial 2D polygon



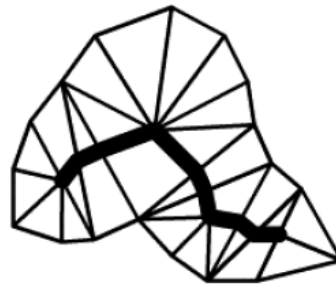
b) result of CDT



c) chordal axis



d) fan triangles



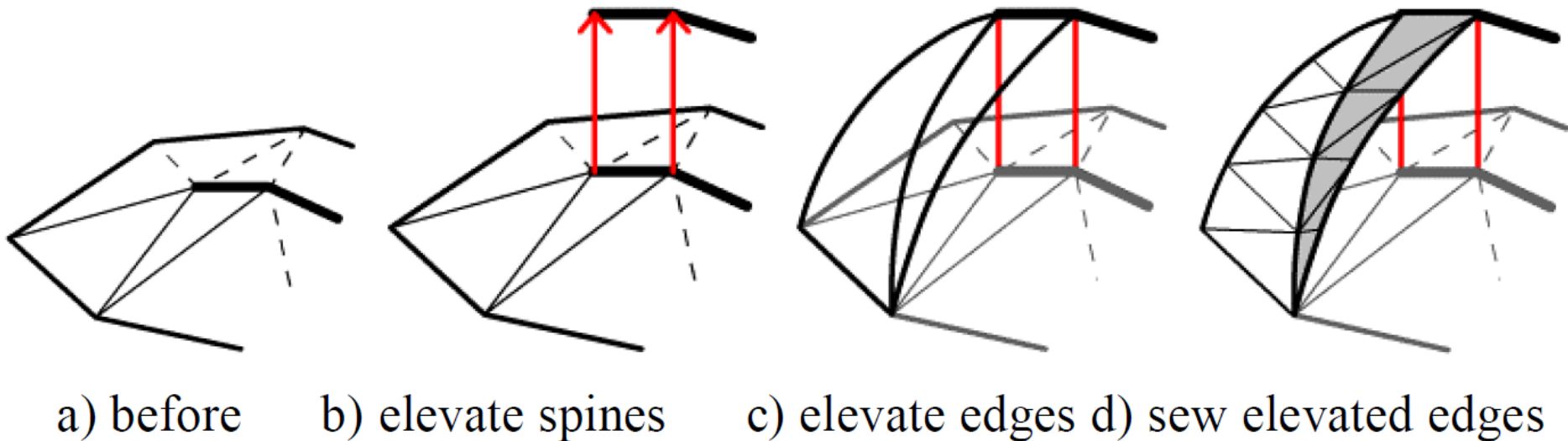
e) resulting spine



f) final triangulation

Teddy

- Skeleton \rightarrow 3D:
 - Depth proportional to distance to the outer contour
- Assume every 'fan' edge is a part of an ellipse:



Can we interpret
natural drawings?

What lines do we draw?

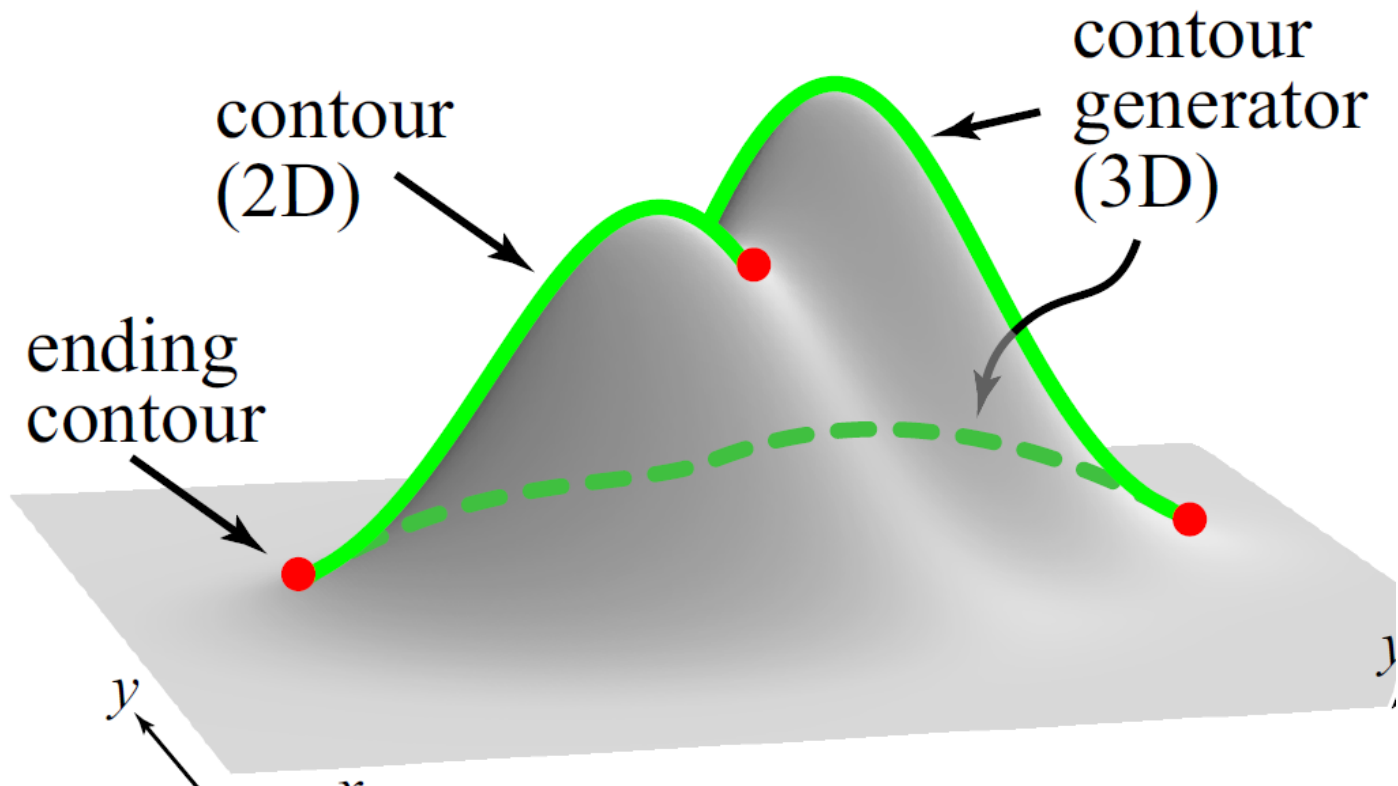


© Ivan Huska, easy-drawings-and-sketches.com

What lines should we render?

Occlusion contours

$$n \cdot (p - O_{camera}) = 0$$



What lines should we render?

Occlusion contours

$$n \cdot (p - O_{camera}) = 0$$



Demo

What lines should we render?

Occlusion contours

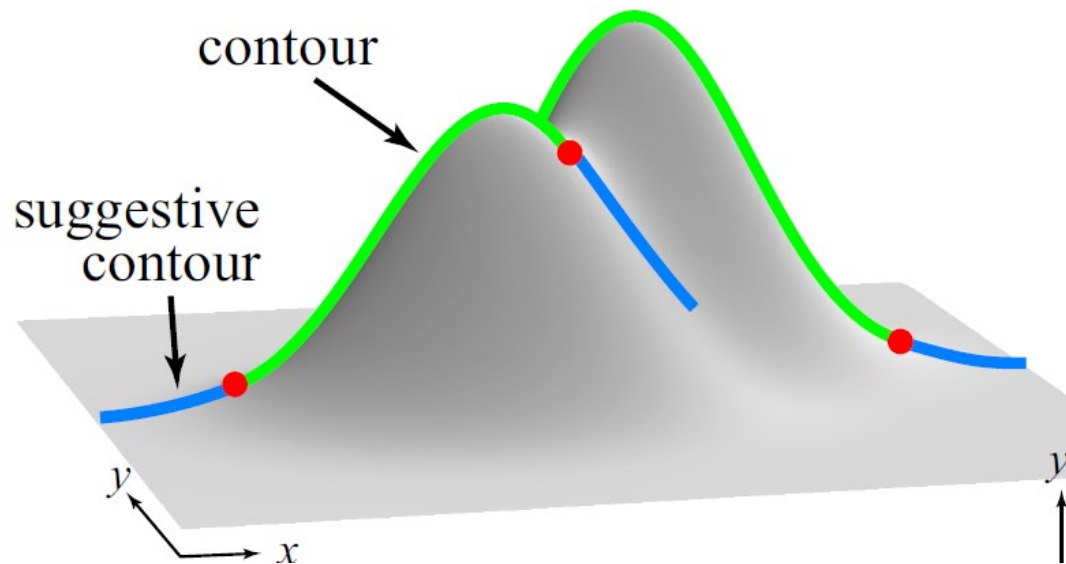
$$n \cdot (p - O_{camera}) = 0$$

How to find
contours in the
nearby views?



Suggestive contours

- Extend the occlusion contours



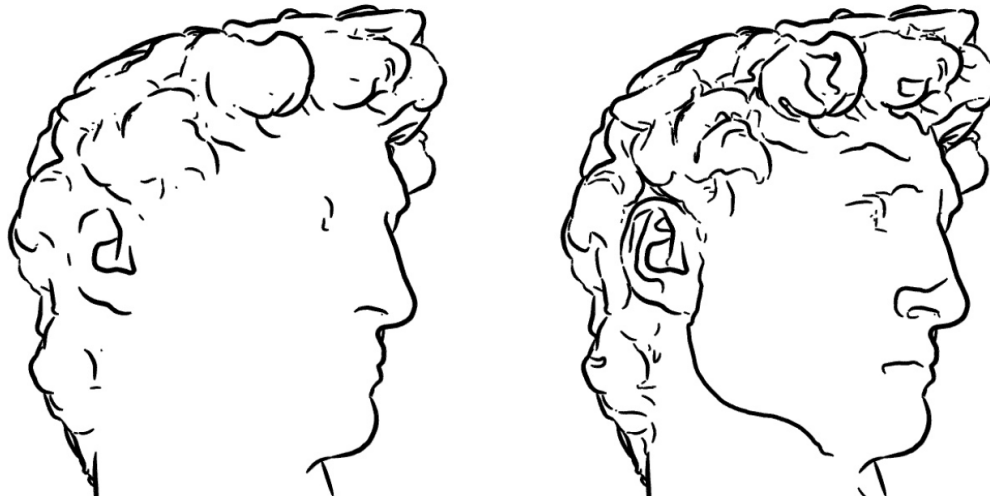
What lines should we render?

- Occlusion contours

$$n \cdot (p - O_{camera}) = 0$$

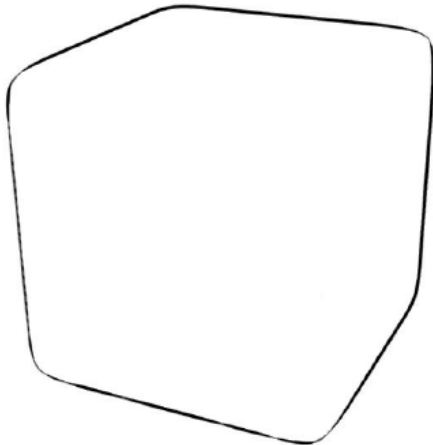
- Suggestive contours

$$\min_p n \cdot (p - O_{camera})$$

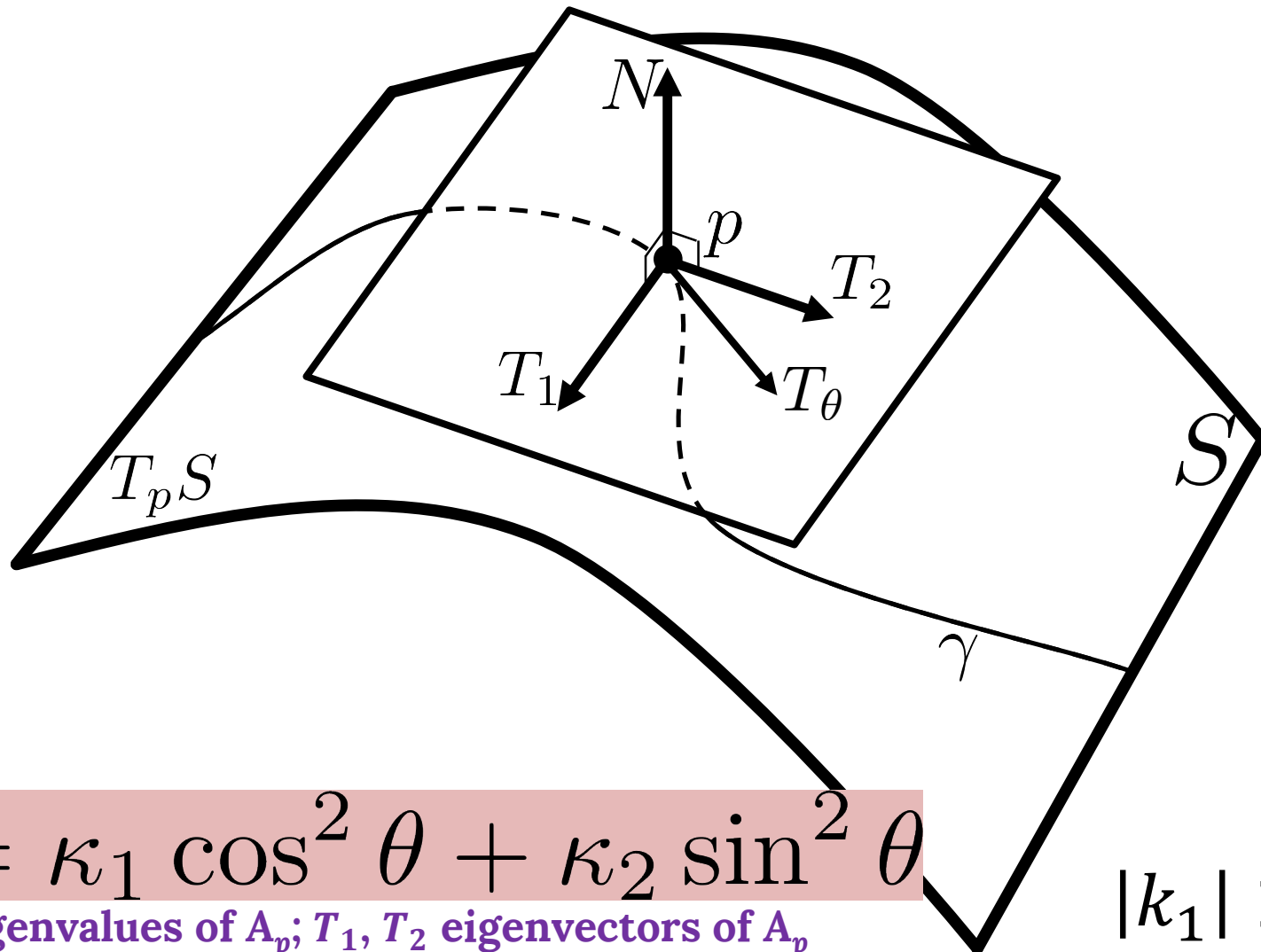


Not a universal tool

Objects without concavities don't have suggestive contours



Recall: Principal Directions and Curvatures



$$\kappa_\theta = \kappa_1 \cos^2 \theta + \kappa_2 \sin^2 \theta$$

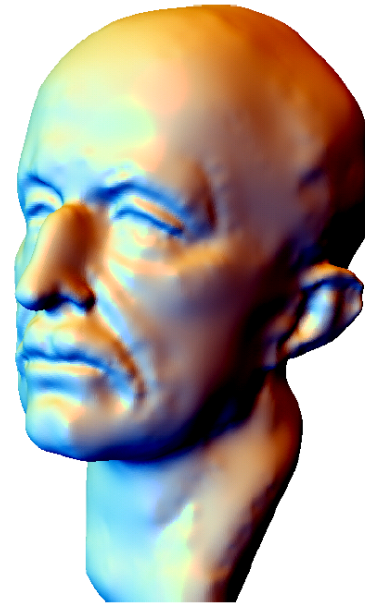
κ_1, κ_2 eigenvalues of A_p ; T_1, T_2 eigenvectors of A_p

$$|k_1| > |k_2|$$

Ridges and valleys

Extrema of principal curvature

$$\frac{\partial k_1}{\partial T_1} = 0$$



Ridges and valleys

Local max of $k_1 > 0$ **ridge**

Local min of $k_1 < 0$ **valley**



Recall:

Second Fundamental Form

$$DN_p : T_p S \rightarrow T_p S$$



$$A_p(V, W) := -\langle DN_p(V), W \rangle$$

“Shape operator”

Apparent Ridges

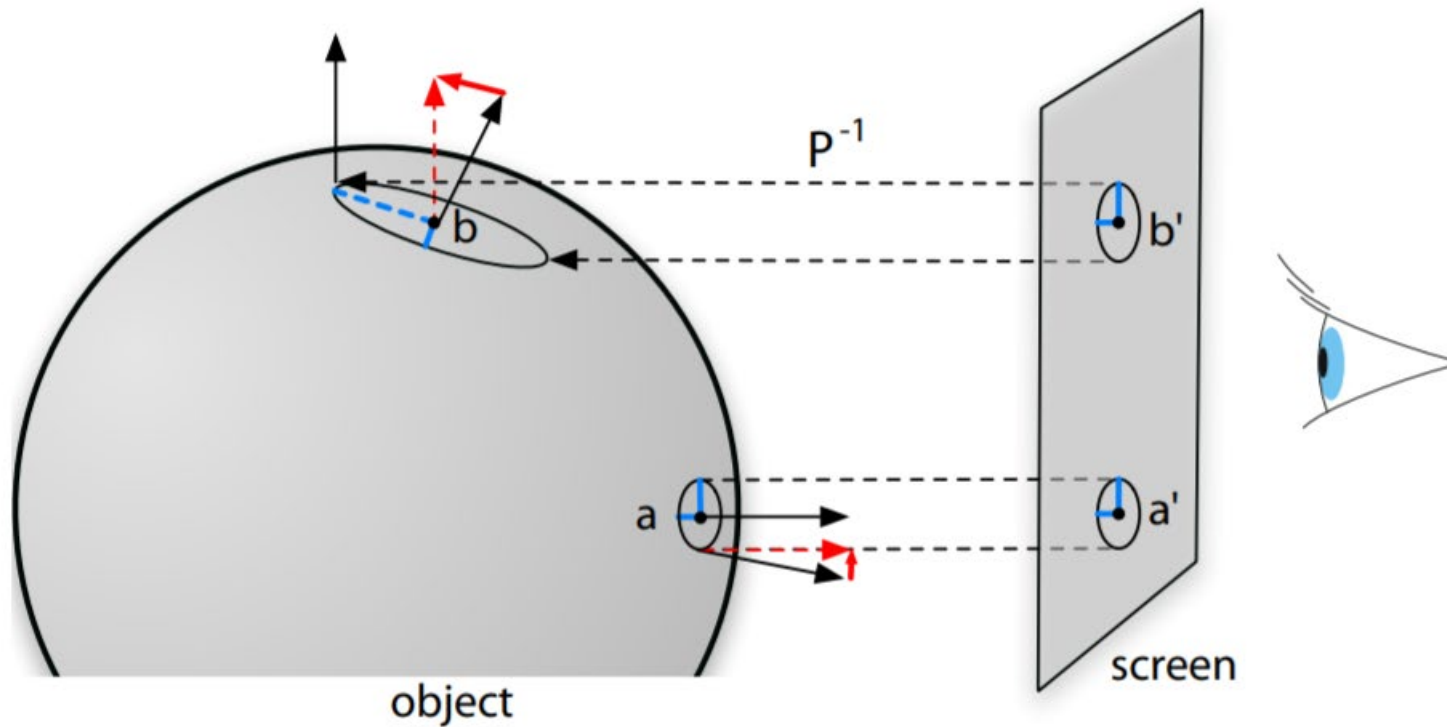


Figure 4: *The maximum view-dependent curvature at b' is much larger than at a' uniquely because of projection.*

Apparent Ridges

$$A'_{p(V,W)} = -\langle \text{proj}_{screen} DN_p(V), W \rangle$$

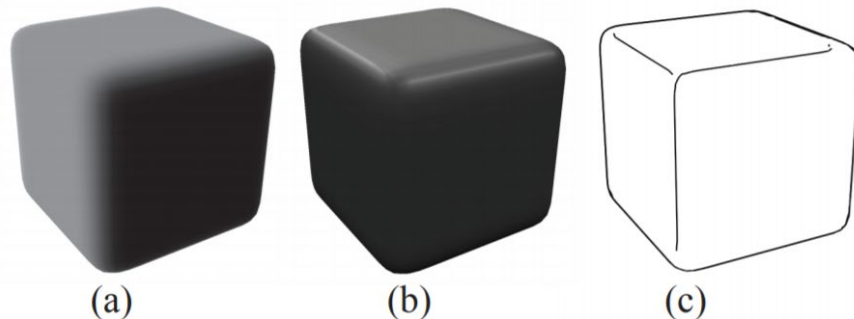


Figure 2: Depiction of a cube with traditional computer graphics shading and with line drawing (using our apparent ridges).

What lines do we draw?

Where Do People Draw Lines?

Forrester Cole, Aleksey Golovinskiy, Alex Limpacher, Heather Stoddart Barros,
Adam Finkelstein, Thomas Funkhouser, and Szymon Rusinkiewicz

Princeton University

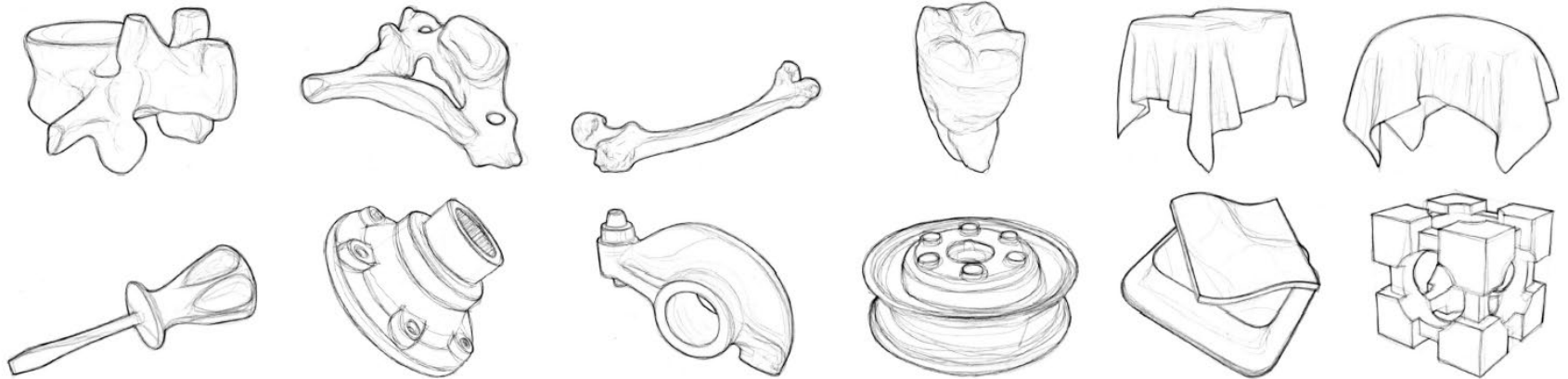
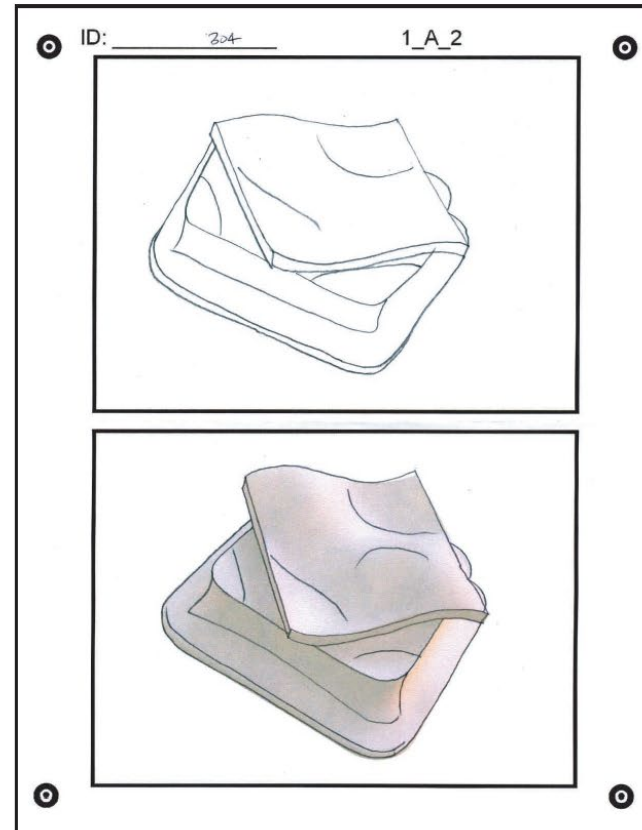
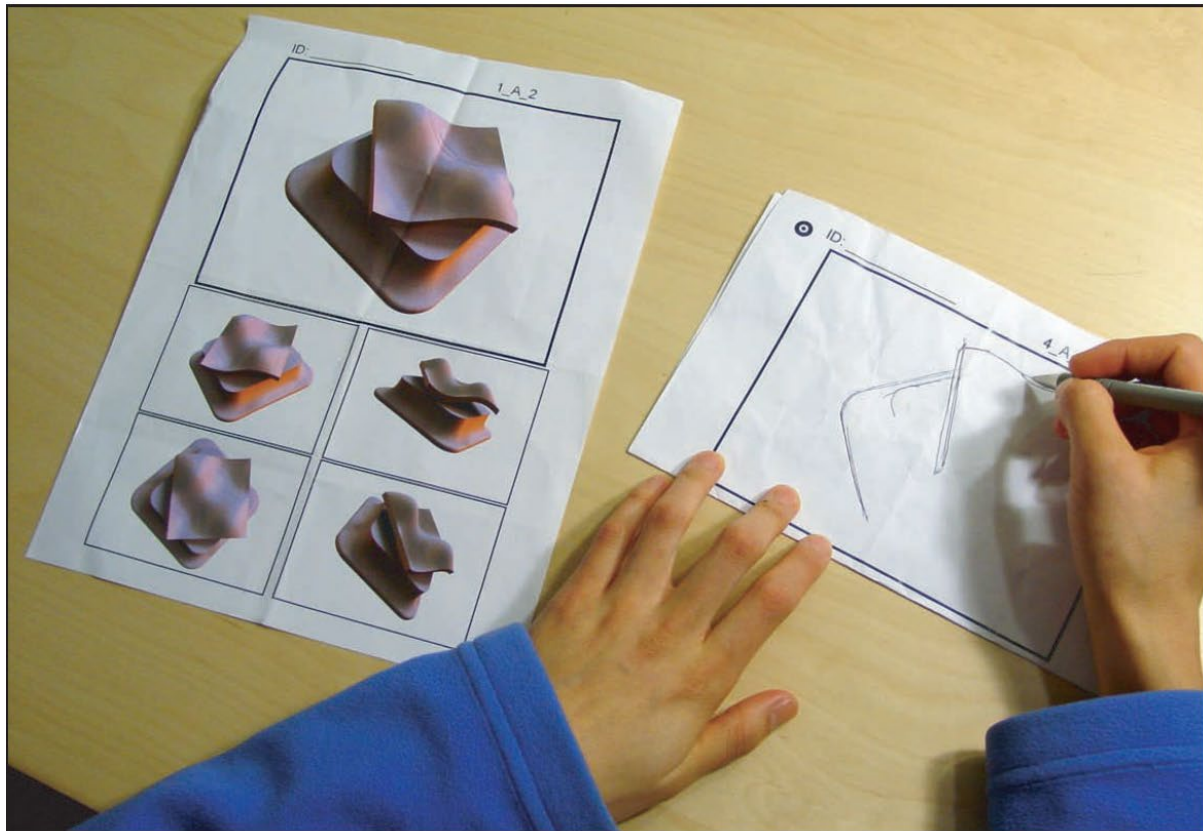


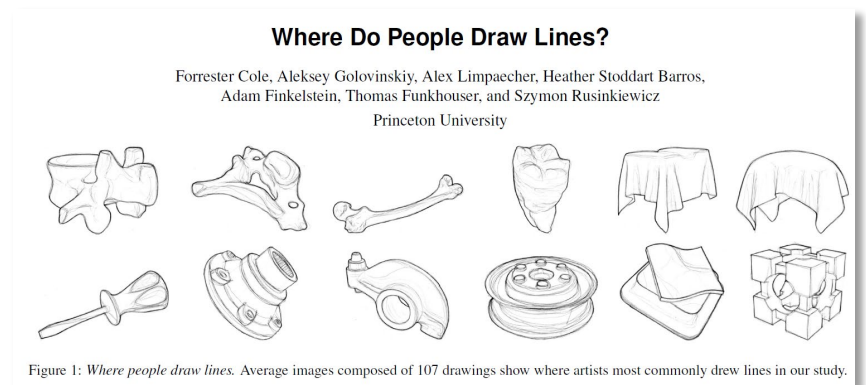
Figure 1: *Where people draw lines.* Average images composed of 107 drawings show where artists most commonly drew lines in our study.

What lines do we draw?



What lines do we draw?

- Occlusion contours
- Suggestive contours
- Ridges and valleys
- Apparent ridges
- Feature lines



Can you classify these lines?



© Ivan Huska, easy-drawings-and-sketches.com

Stats

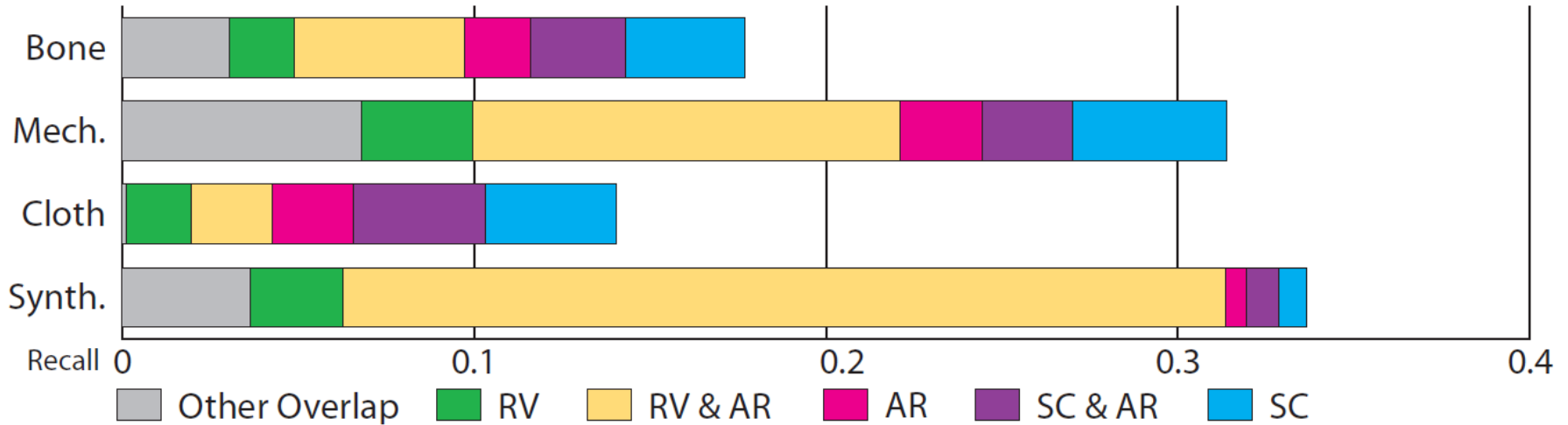


Figure 9: *Non-contour lines*. Categorization of artists' lines that are not exterior or interior occluding contours: geometric ridges and valleys (RV), apparent ridges (AR), suggestive contours (SC), and combinations.

If we can label the lines, can we
reconstruct 3D?

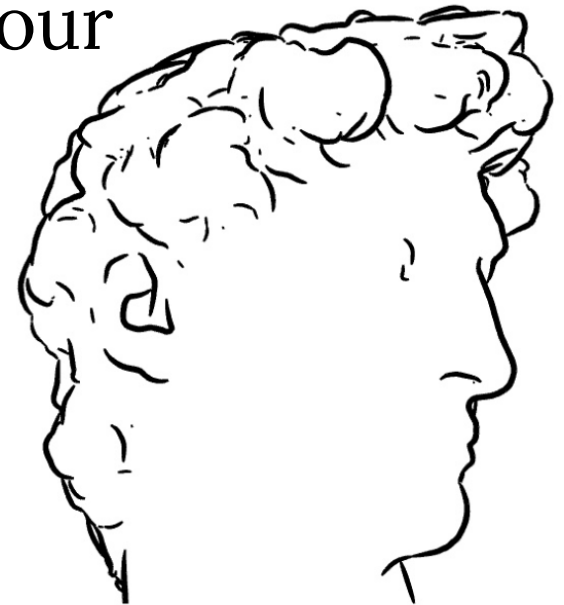
Issues

- Infinite # of 3D surfaces have the same 2D
- Drawings are approximate

Occlusion contours

$$n \cdot (p - O_{camera}) = 0$$

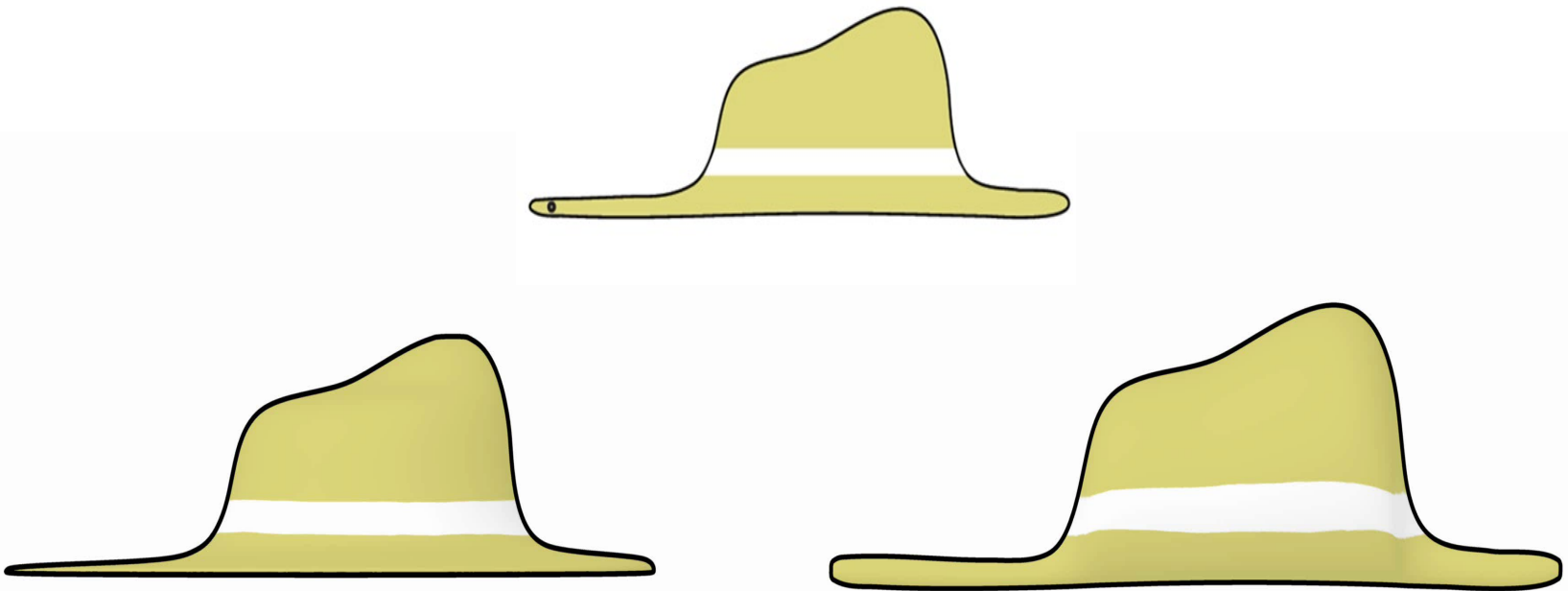
- *Contour generators* not parallel to the screen
- Sign of contour curvature = sign of Gaussian curvature
- Gaussian curvature around contour generators' endpoints < 0
- Depth discontinuities!



Occlusion contours

$$n \cdot (p - O_{camera}) = 0$$

Necessary, but **not sufficient**
for shape recognition

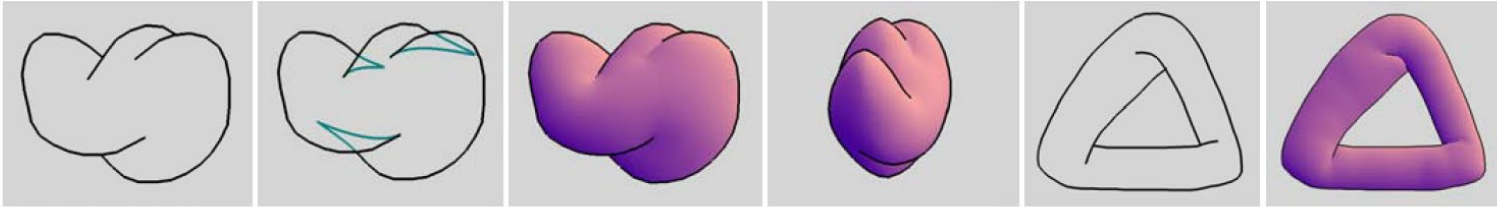


Completing Contours

SmoothSketch: 3D free-form shapes from complex sketches

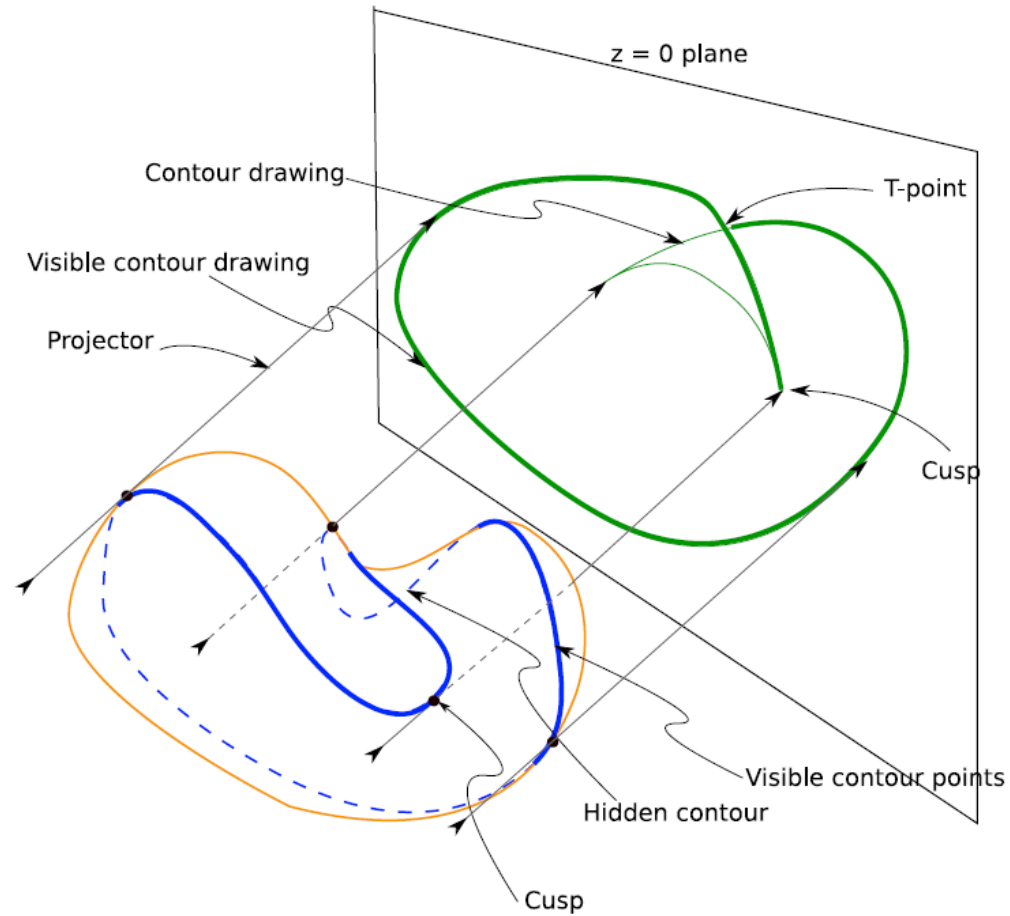
Olga A. Karpenko*
Brown University

John F. Hughes†
Brown University



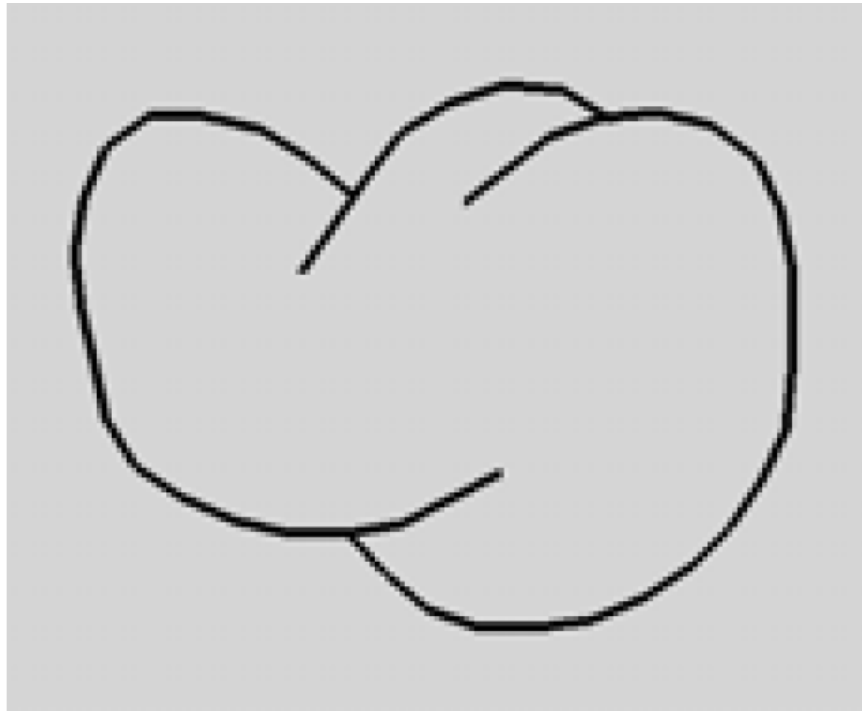
We only see *visible* contours
Complete into loops and inflate!

Completing Contours



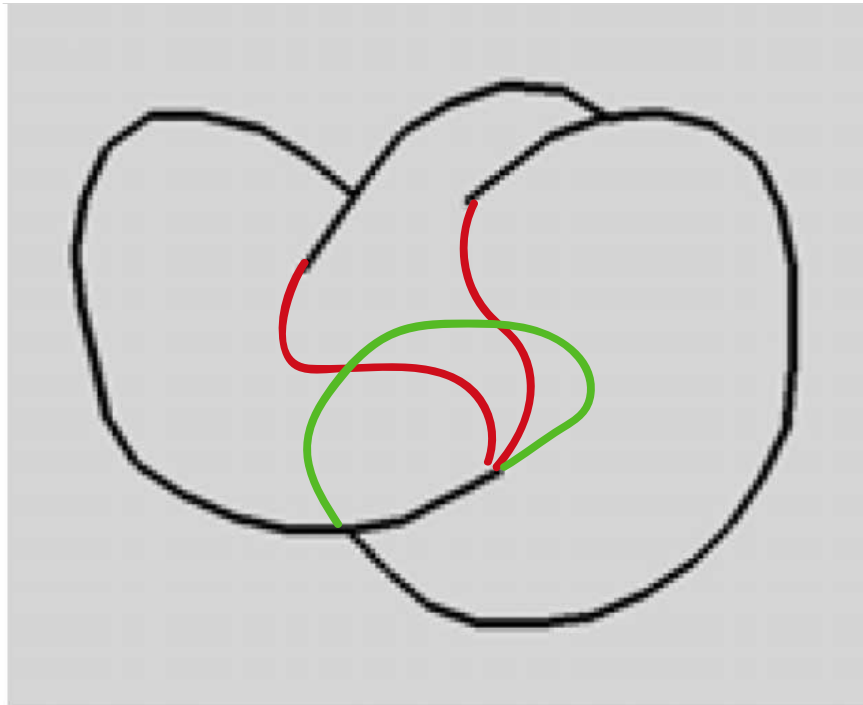
Completing Contours

- Heuristics:
 - Measure some energy of Bézier curve

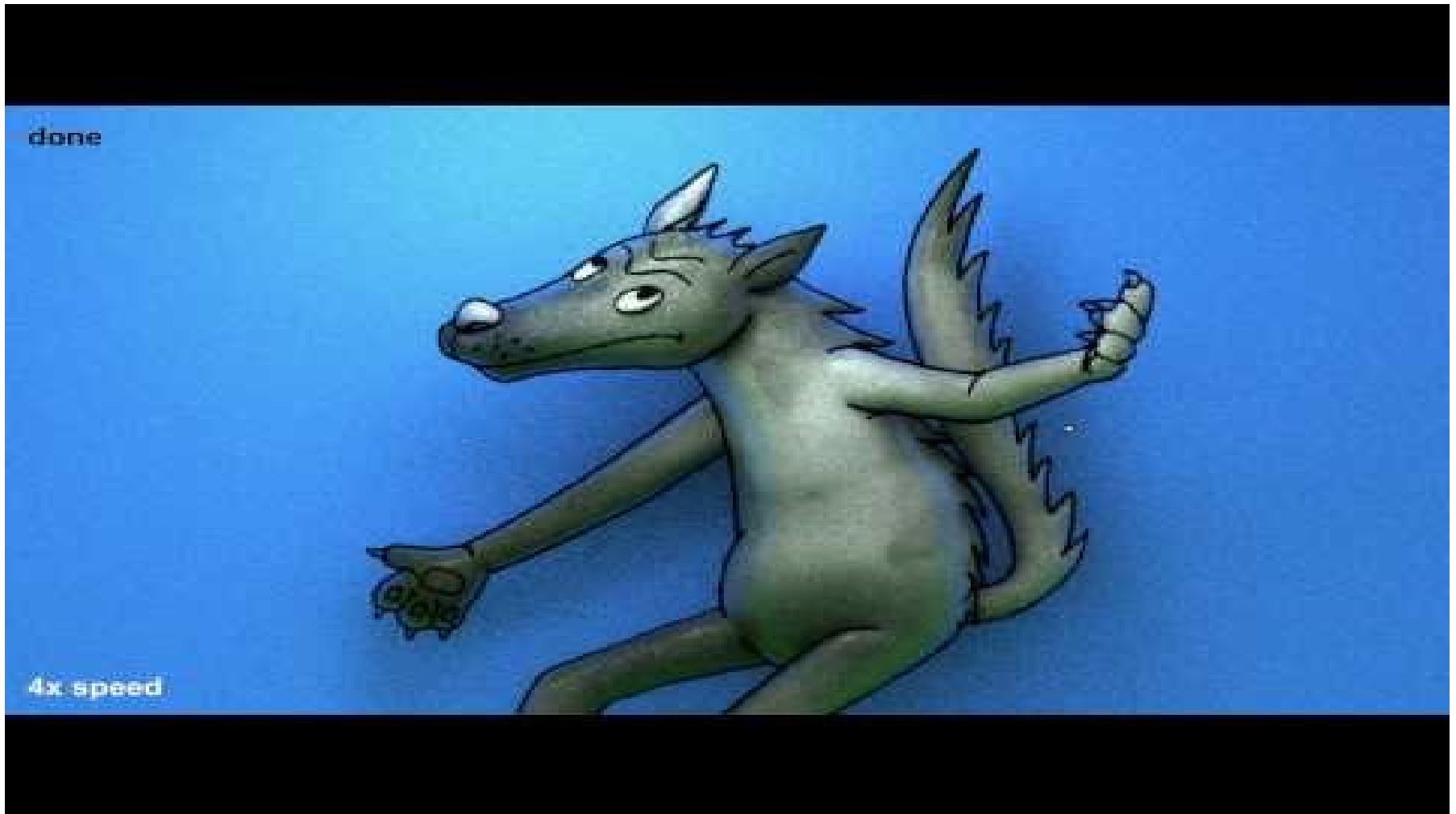


Completing Contours

- Heuristics:
 - Measure some energy of Bézier curve



A simpler problem: illumination effects

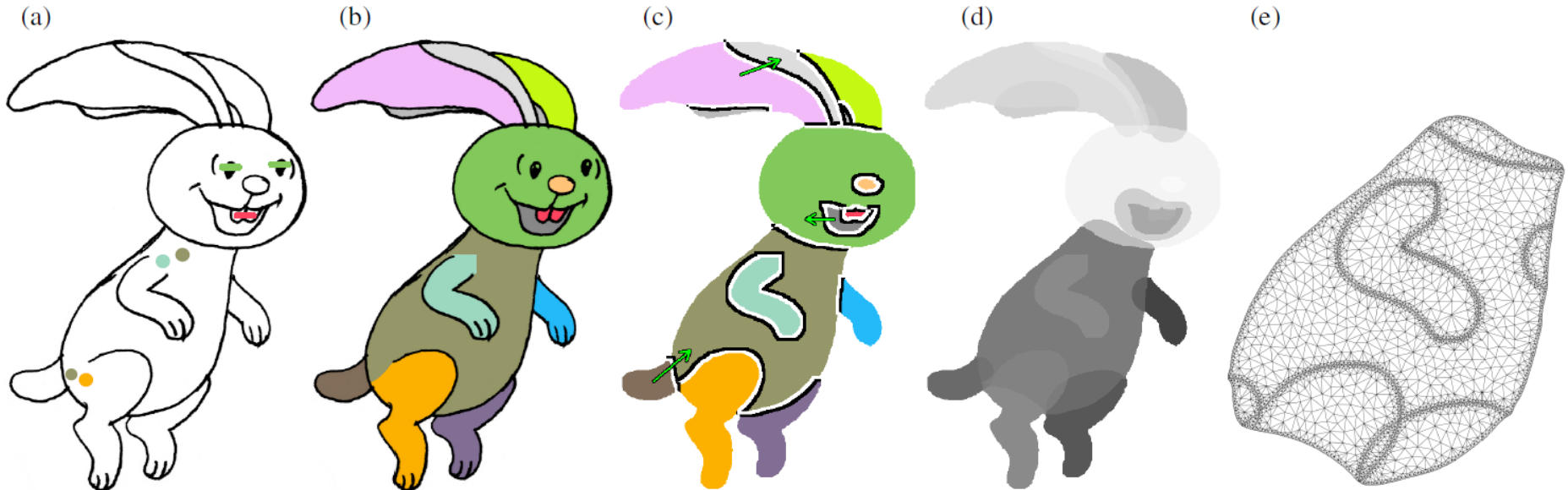


A simpler problem: illumination effects



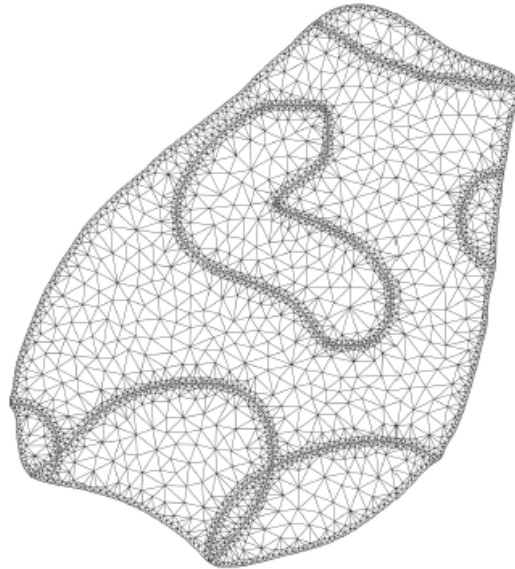
Preprocessing

- Complete all regions
- Find/ask user to specify depth order
- Constrained Delaunay Triangulation (CDT)



How to lift to 3D?

(e)

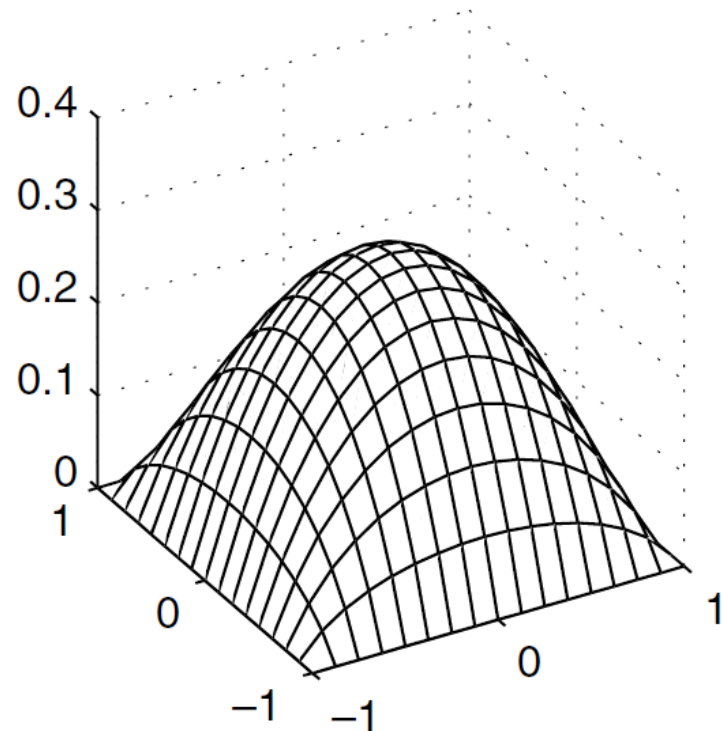
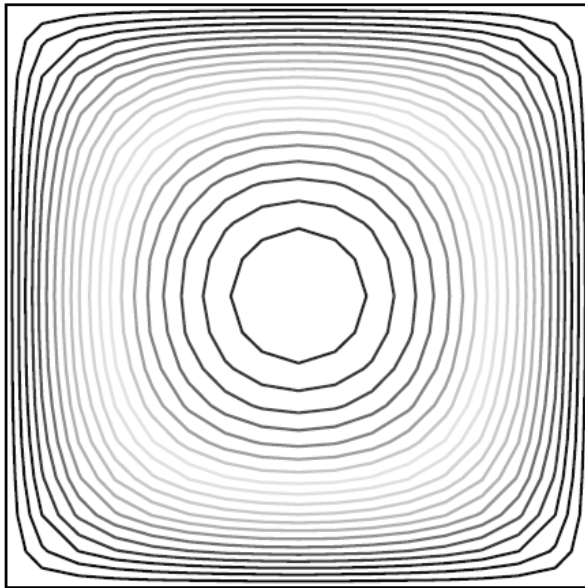


Idea

Steady-state heat
equation
(with a heat source)

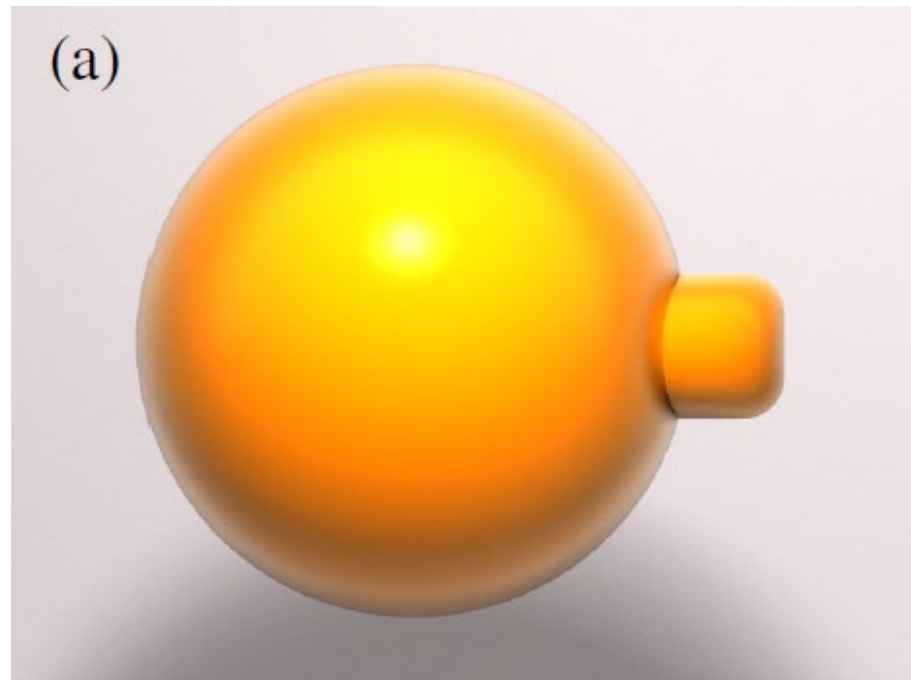
$$\Delta z_0 = a$$
$$z_0 \Big|_{\partial\Gamma} = 0$$

$$a \in \mathbb{R}$$



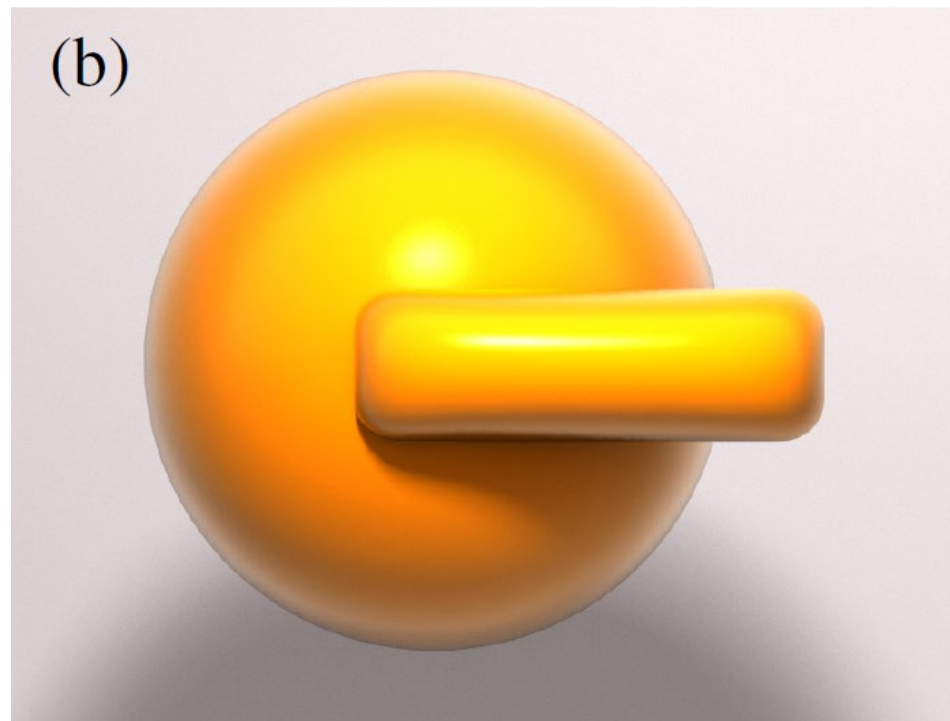
After inflation

- Each piece is centered around $z = 0$
- Need to use z-order! $z_1 < z_2$
- Need a smooth model



After inflation

- Each piece is centered around $z = 0$
- **Need to use z-order!** $z_1 < z_2$
- Need a smooth model

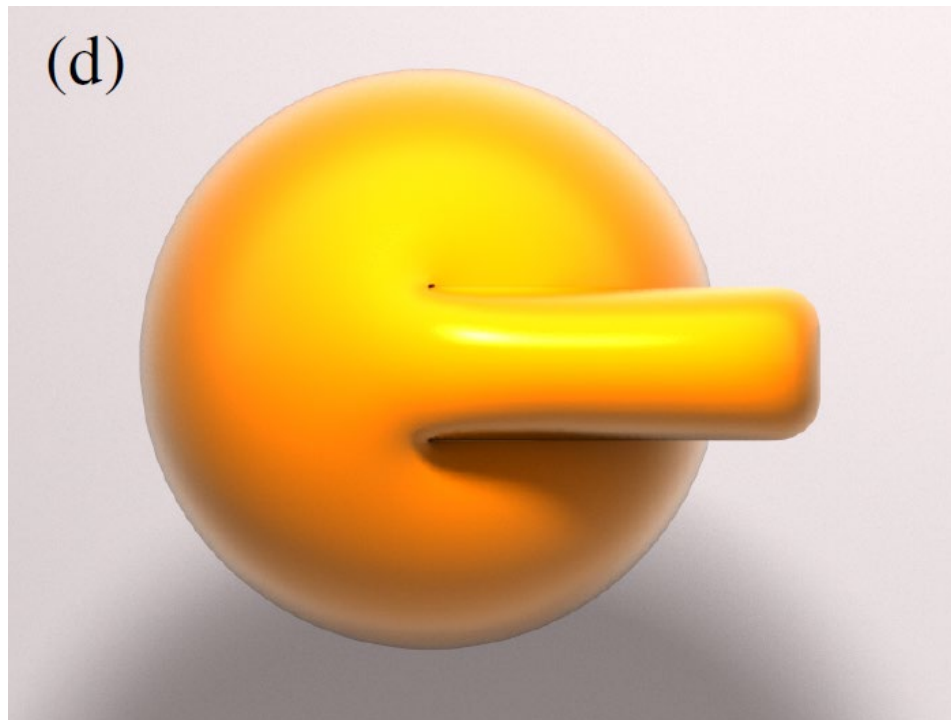


Smooth function?

$$z^i = z_0^i + g^i(x)$$

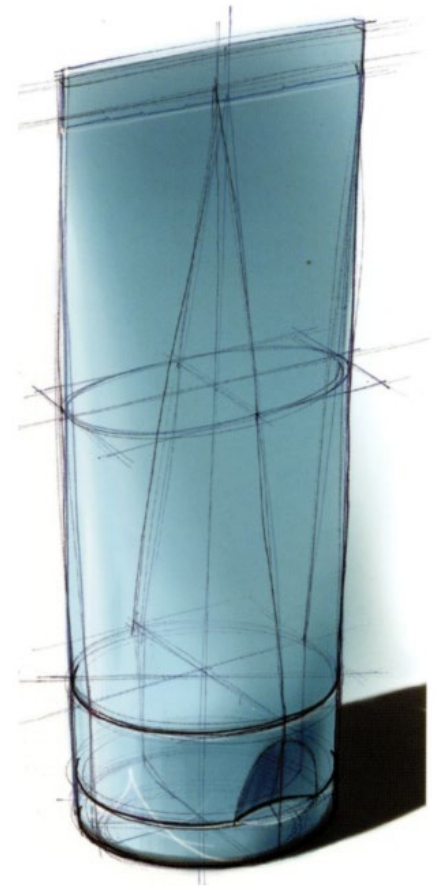
$$\min \int_{\Omega} \|\nabla g_i\|^2 dx$$

$$\text{s.t. } z_i < z_j$$



Cross-sections

- Lines of curvature
- At intersections:
 - Orthogonal
 - Define a tangent plane
 - (often) belong to perpendicular planes



A simpler problem: shade?

CrossShade: Shading Concept Sketches Using Cross-Section Curves

Cloud Shao^{1*}

¹ University of Toronto

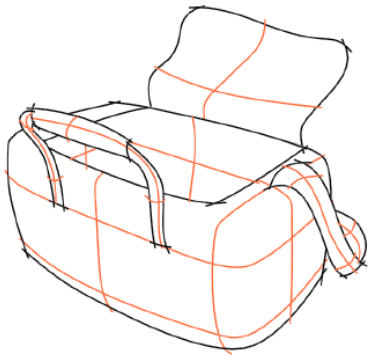
Adrien Bousseau^{2*}

² REVES - INRIA Sophia Antipolis

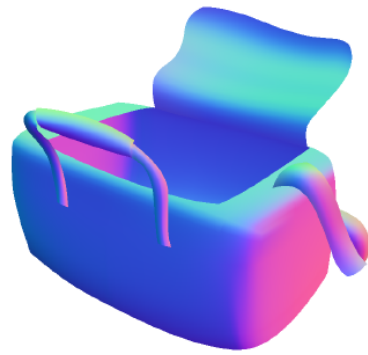
Alla Sheffer³

³ University of British Columbia

Karan Singh¹



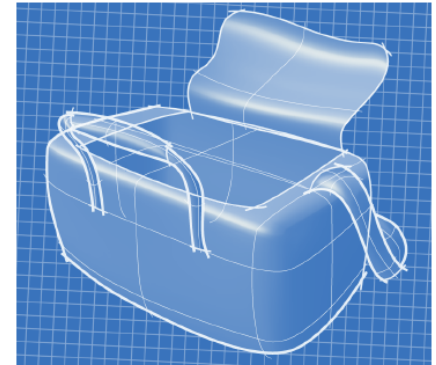
(a) Input curves



(b) Estimated normals



(c) Shading



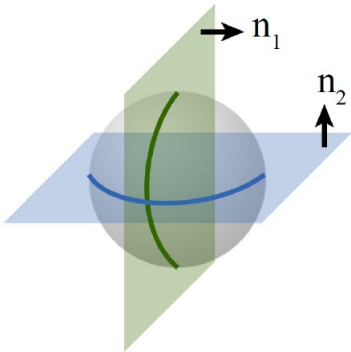
CrossShade

Find a normal field over the drawing

n_i : tangent plane normals

t_{ij} : tangents

$$\mathbf{n}_1 \cdot \mathbf{n}_2 = 0$$



CrossShade

Find a normal field over the drawing

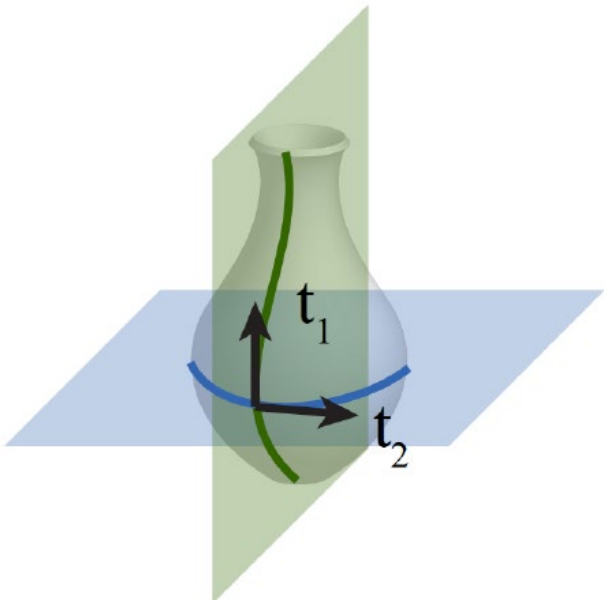
n_i : tangent plane normals

t_{ij} : tangents

$$\mathbf{n}_1 \cdot \mathbf{n}_2 = 0$$

$$\mathbf{t}_1 \cdot \mathbf{t}_2 = 0$$

Cross-sections as
curvature lines

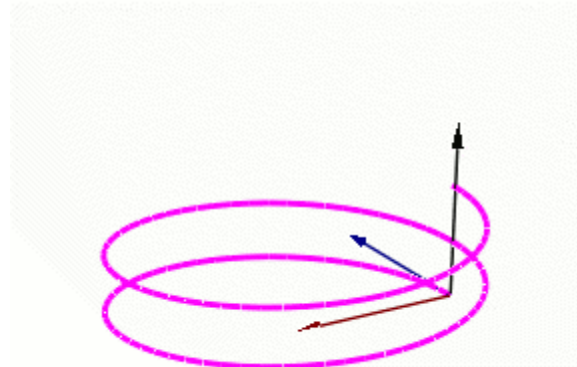
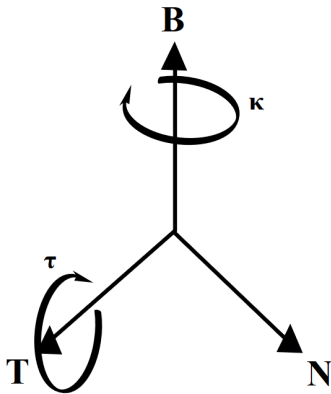


Recall:

Frenet Frame: Curves in \mathbb{R}^3

$$\frac{d}{ds} \begin{pmatrix} T \\ N \\ B \end{pmatrix} = \begin{pmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{pmatrix} \begin{pmatrix} T \\ N \\ B \end{pmatrix}$$

- **Binormal:** $T \times N$
- **Curvature:** In-plane motion
- **Torsion:** Out-of-plane motion



Recall:

Intuition

$$\text{proj}_{T_{\gamma(s)}S} [\gamma''(s)] = 0$$

- **The only acceleration is out of the surface**
 - **No steering wheel!**



CrossShade

Find a normal field over the drawing

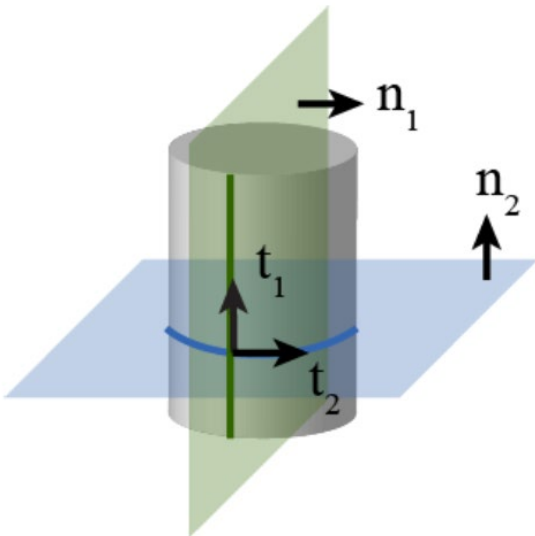
n_i : tangent plane normals

t_{ij} : tangents

$$t_1 \times n_2 = 0$$

$$t_2 \times n_1 = 0$$

Geodesics?



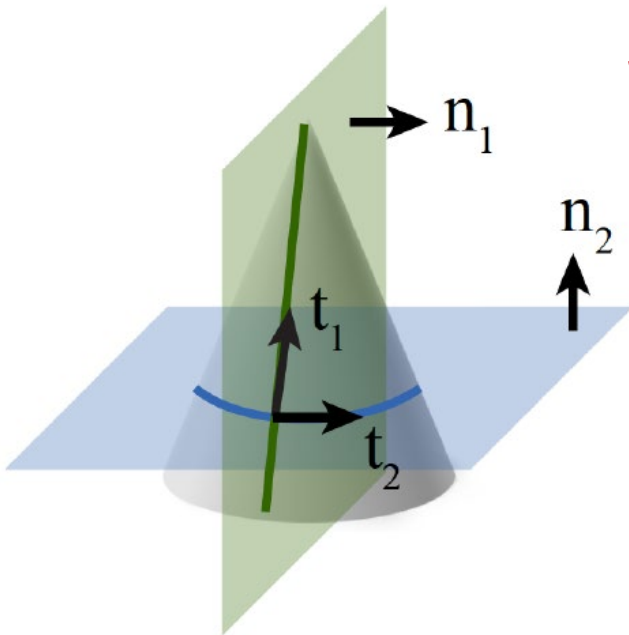
“humans perceive intersecting cross-section curves as geodesics”

CrossShade

Find a normal field over the drawing

n_i : tangent plane normals

t_{ij} : tangents



~~$$t_1 \times n_2 = 0$$~~

$$t_2 \times n_1 = 0$$

At least one is
a geodesic

CrossShade: overview

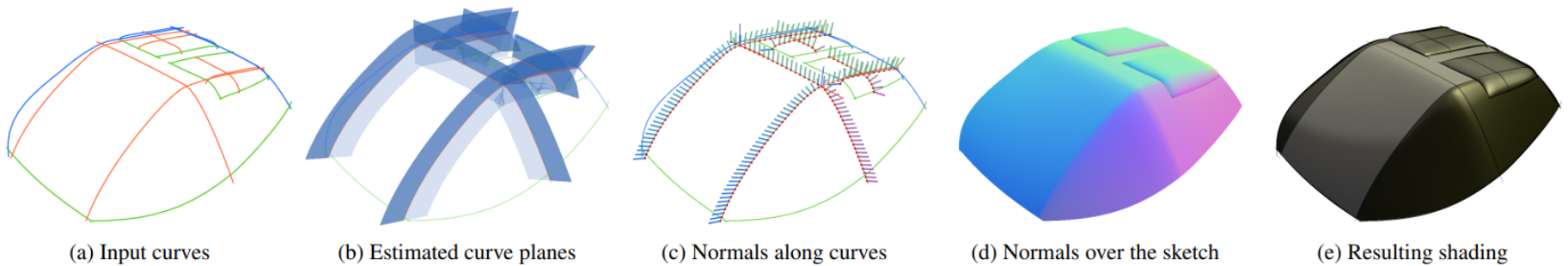
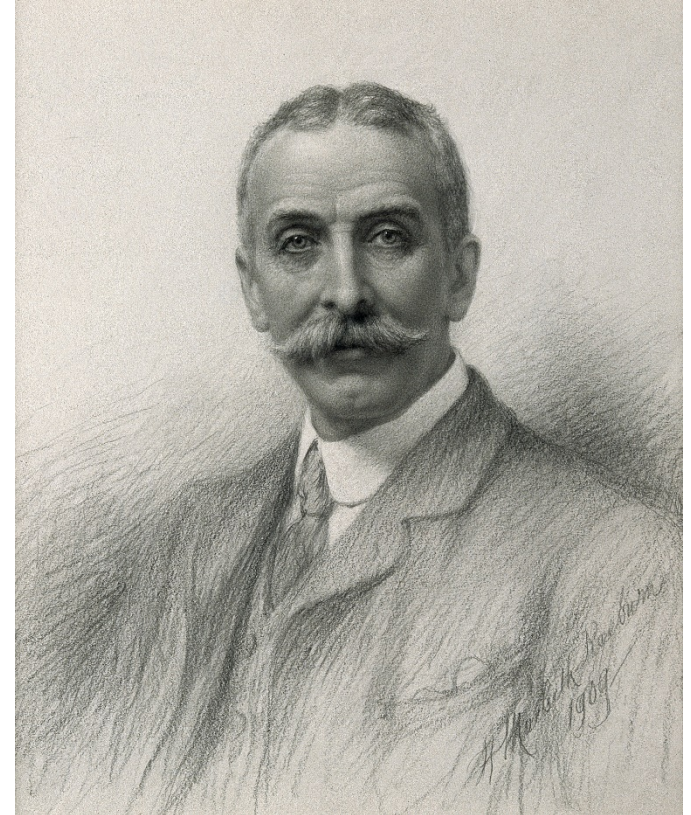


Figure 4: *Our algorithm takes as input an annotated sketch (a). Orange curves denote cross-sections, blue curves represent smooth silhouettes, and green curves correspond to other object boundaries. We first optimize for the supporting plane of each cross-section and compute the 3C cross-sections based on those (b). We use the resulting 3D curves to compute 3D normals at each intersection and interpolate normals along the curves (c). We finally generate a normal field in between the curves using Coons' interpolation (d).*

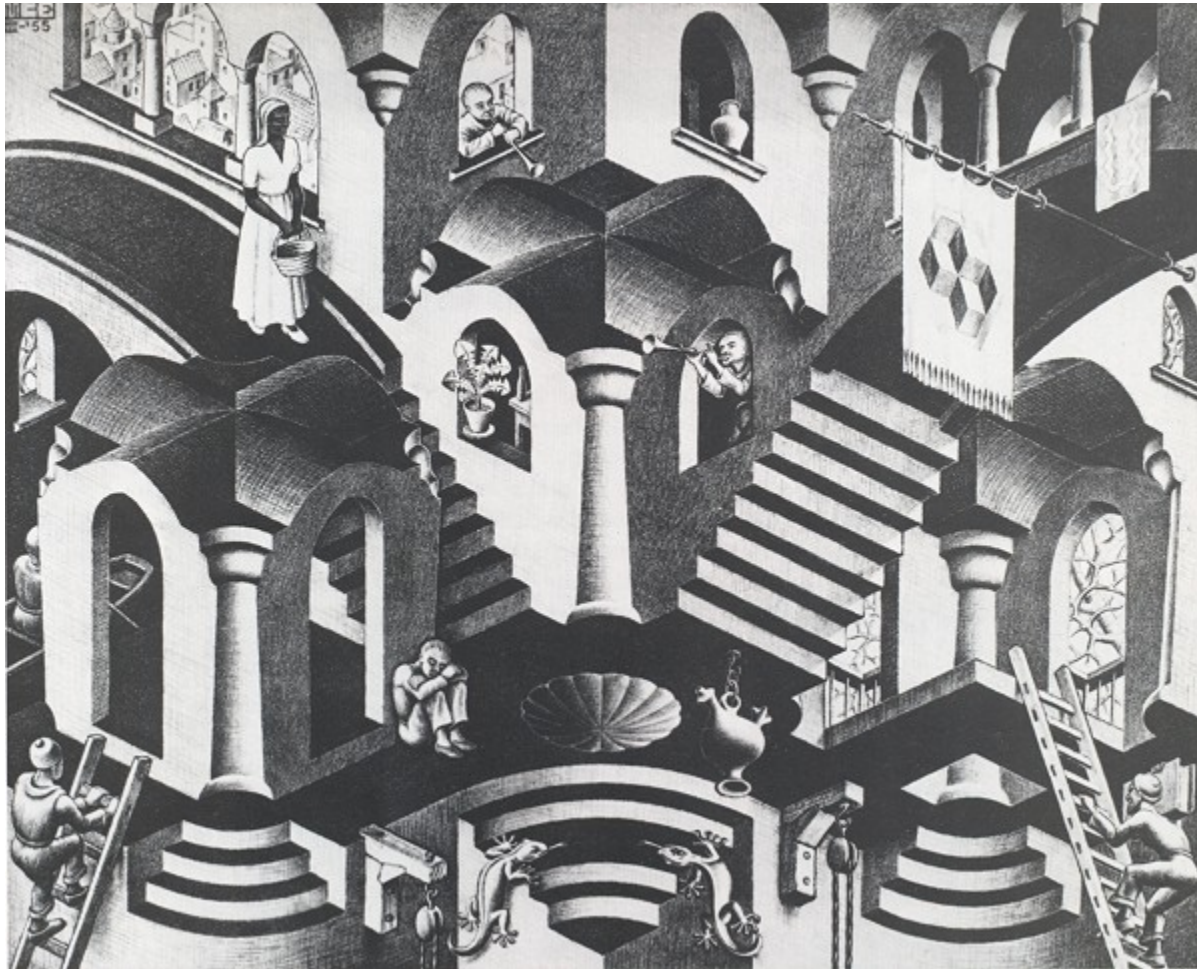
Shading/Hatching

- Approximates actual surface shading?
- Assume a shading model
 - E.g. Lambertian
- How to get illuminance per point?
- Shape from shading



Edward Law. Pencil drawing by H. M. Raeburn, 1909

Concave or convex?



M.C. Escher, "Concave and convex"

Shape priors

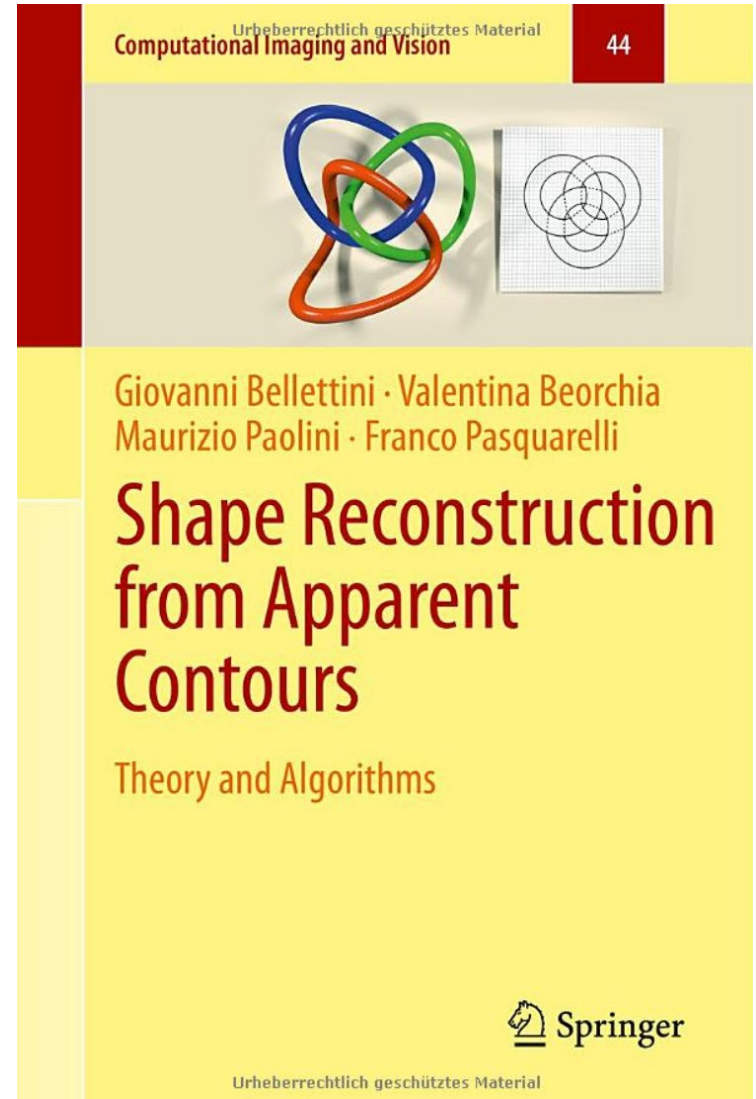
- Developability?
 - Garments
- Geometric primitives
 - Spheres, cylinders,...

Shape representations

- Meshes
- Generalized cylinders
- Bézier
- NURBS
- ...

A rigorous approach

Only for smooth shapes



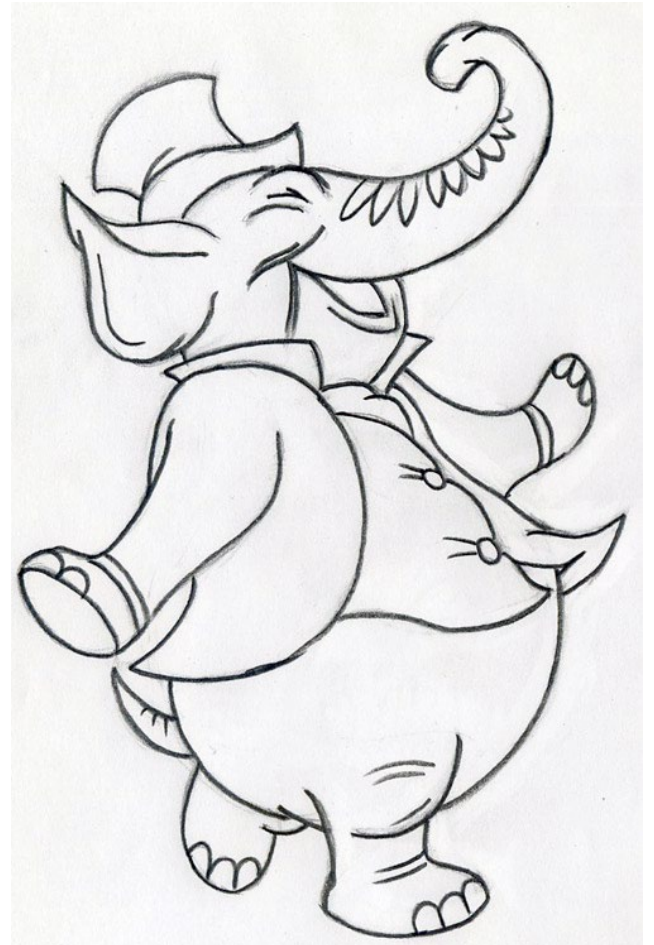
Junctions

- Local depth depth order!



What to do with hidden parts?

- Use continuity
- Use symmetry
- Use anatomical priors?



Symmetry

Computers & Graphics 46 (2015) 221–230



Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag



SMI 2014

Modeling 3D animals from a side-view sketch



Even Entem^{a,b,*}, Loic Barthe^a, Marie-Paule Cani^b, Frederic Cordier^c, Michiel van de Panne^d

^a IRIT - University of Toulouse, France

^b University of Grenoble-Alpes, CNRS (Laboratoire Jean Kuntzmann) and Inria, France

^c University of Haute Alsace, France

^d University of British Columbia, Canada

ARTICLE INFO

Article history:

Received 8 July 2014

Received in revised form

29 September 2014

Accepted 29 September 2014

Available online 8 October 2014

Keywords:

Implicit modelling

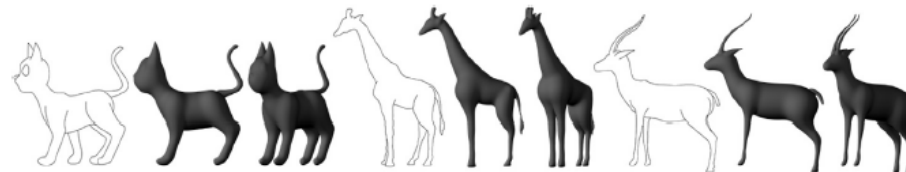
Sketch-based modelling

Organic shapes

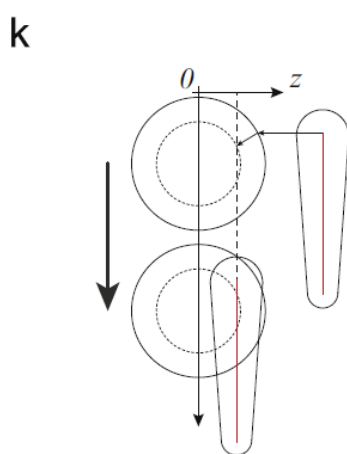
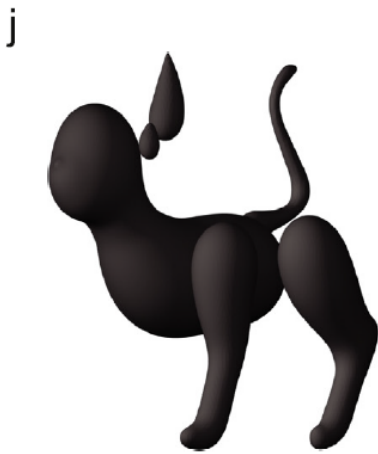
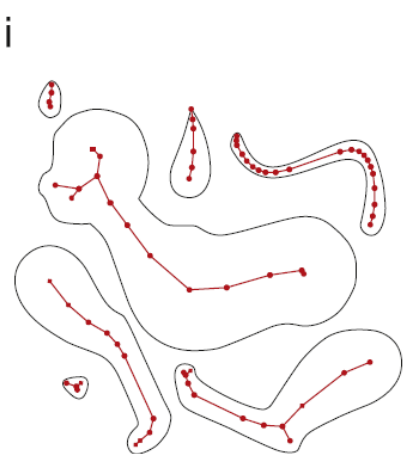
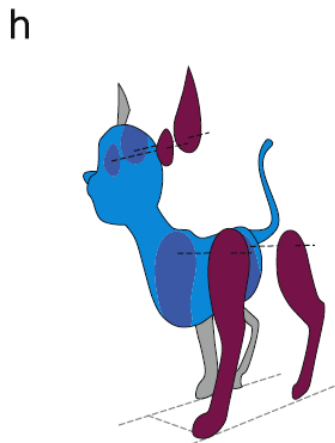
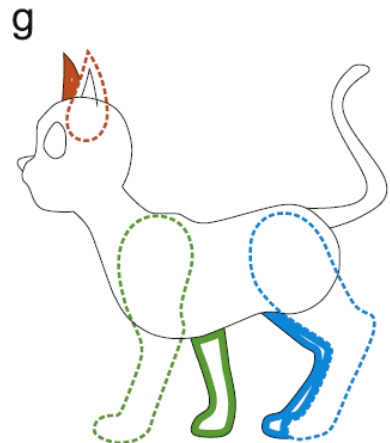
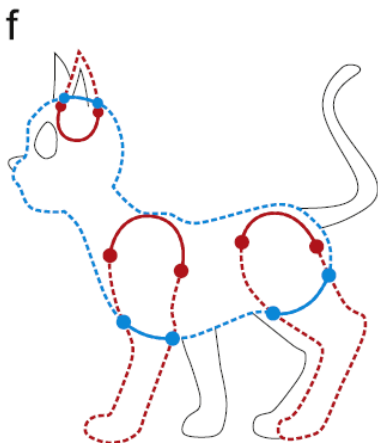
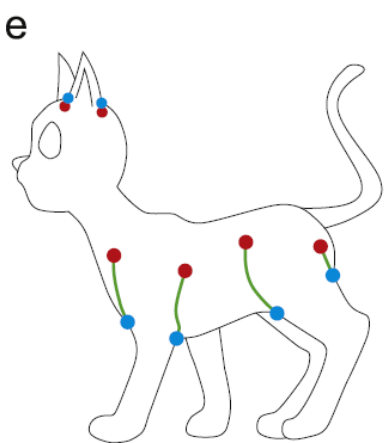
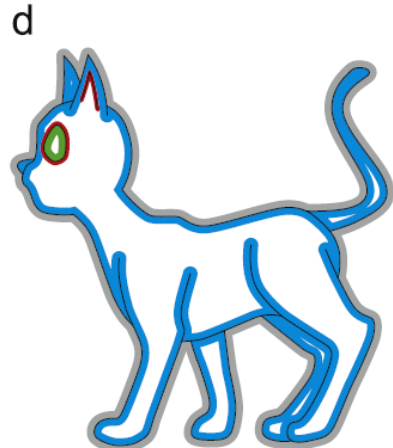
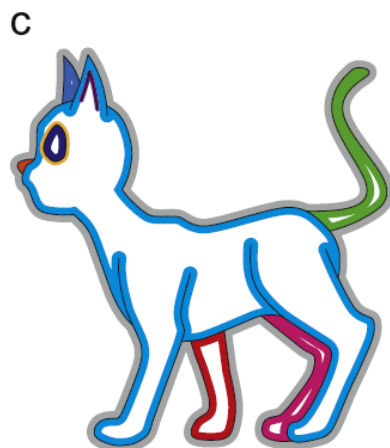
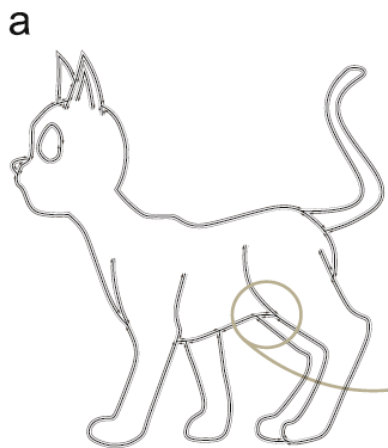
Direct reconstruction

ABSTRACT

Using 2D contour sketches as input is an attractive solution for easing the creation of 3D models. This paper tackles the problem of creating 3D models of animals from a single, side-view sketch. We use the *a priori* assumptions of smoothness and structural symmetry of the animal about the sagittal plane to inform the 3D reconstruction. Our contributions include methods for identifying and inferring the contours of shape parts from the input sketch, a method for identifying the hierarchy of these structural parts including the detection of approximate symmetric pairs, and a hierarchical algorithm for positioning and blending these parts into a consistent 3D implicit-surface-based model. We validate this pipeline by showing that a number of plausible animal shapes can be automatically constructed from a single sketch.



Symmetry



Animation: Line of Action

The Line of Action: an Intuitive Interface for Expressive Character Posing

Martin Guay*

Marie-Paule Cani

Rémi Ronfard

Laboratoire Jean Kuntzmann - Université de Grenoble - Inria

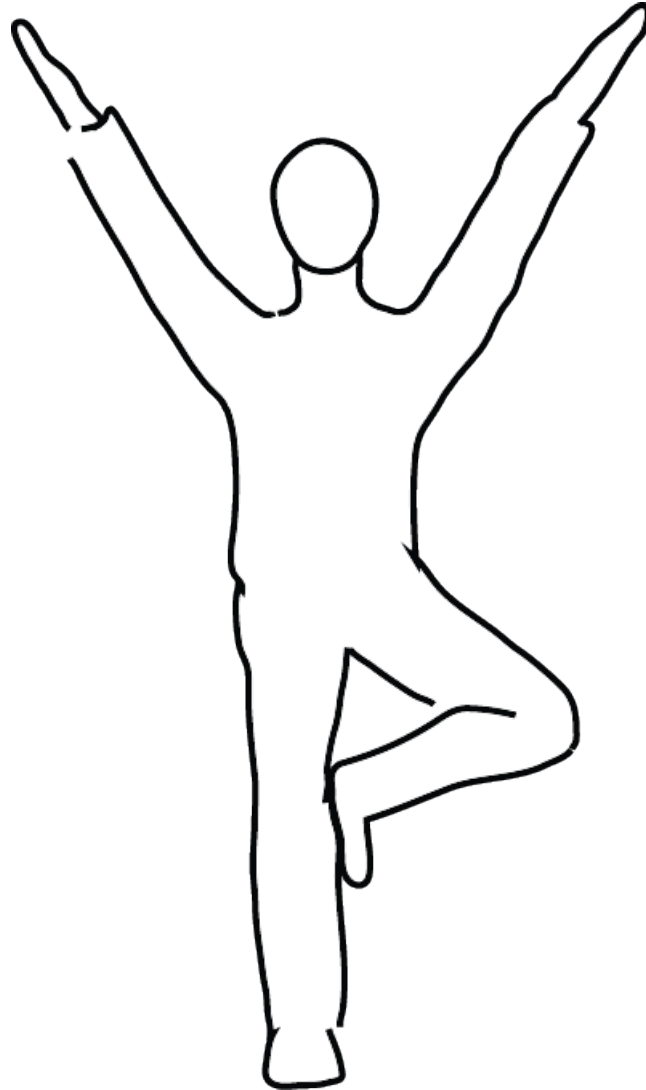


Figure 1: *Expressive character poses created in a few seconds each, by sketching intuitive lines of action.*

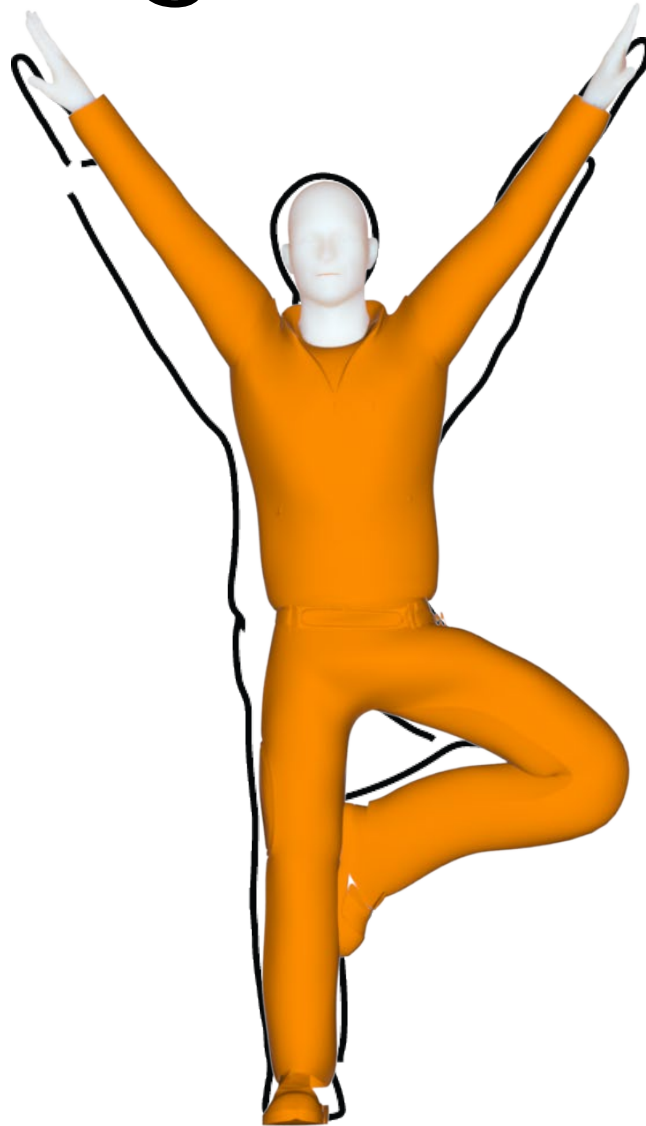
Animation: Line of Action

- Assume we know which bones
- Joint x, y positions should be along the LOA
 - Where along LOA?
- More importantly, bones should be parallel to LOA
- Need curve – bone chain correspondence!

Drawings are inexact



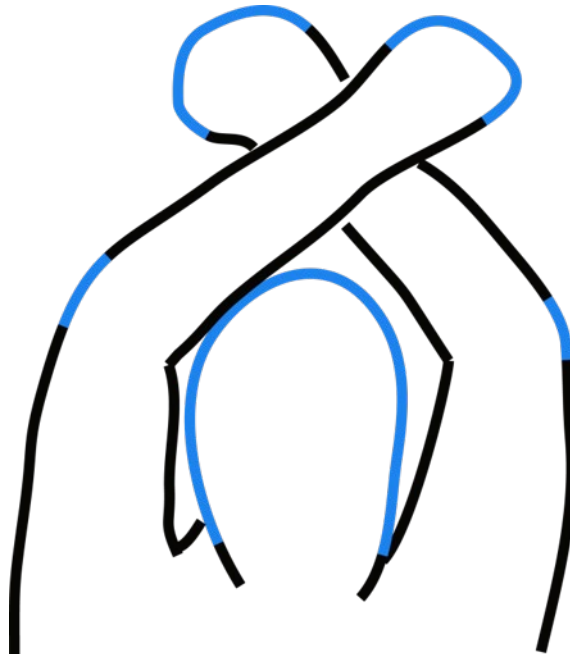
Drawings are inexact



2D embedding

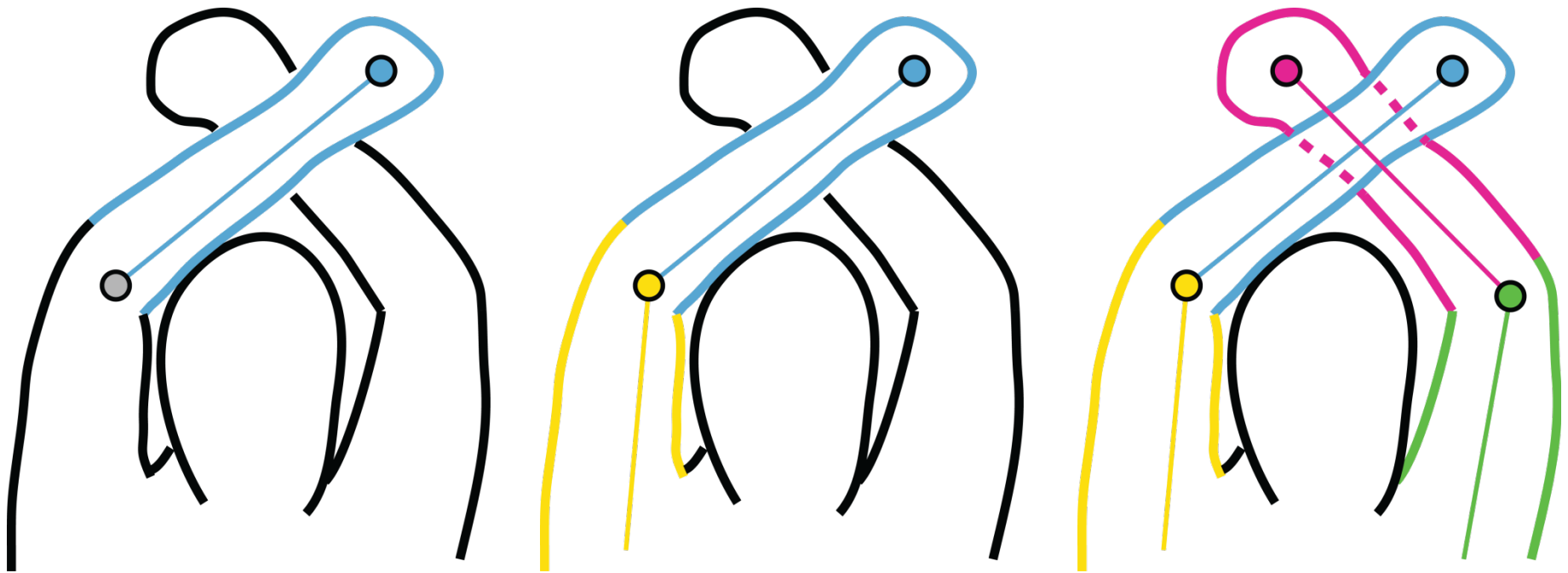
Low-curvature lines ~ body parts

Circular contours ~ joints



2D embedding

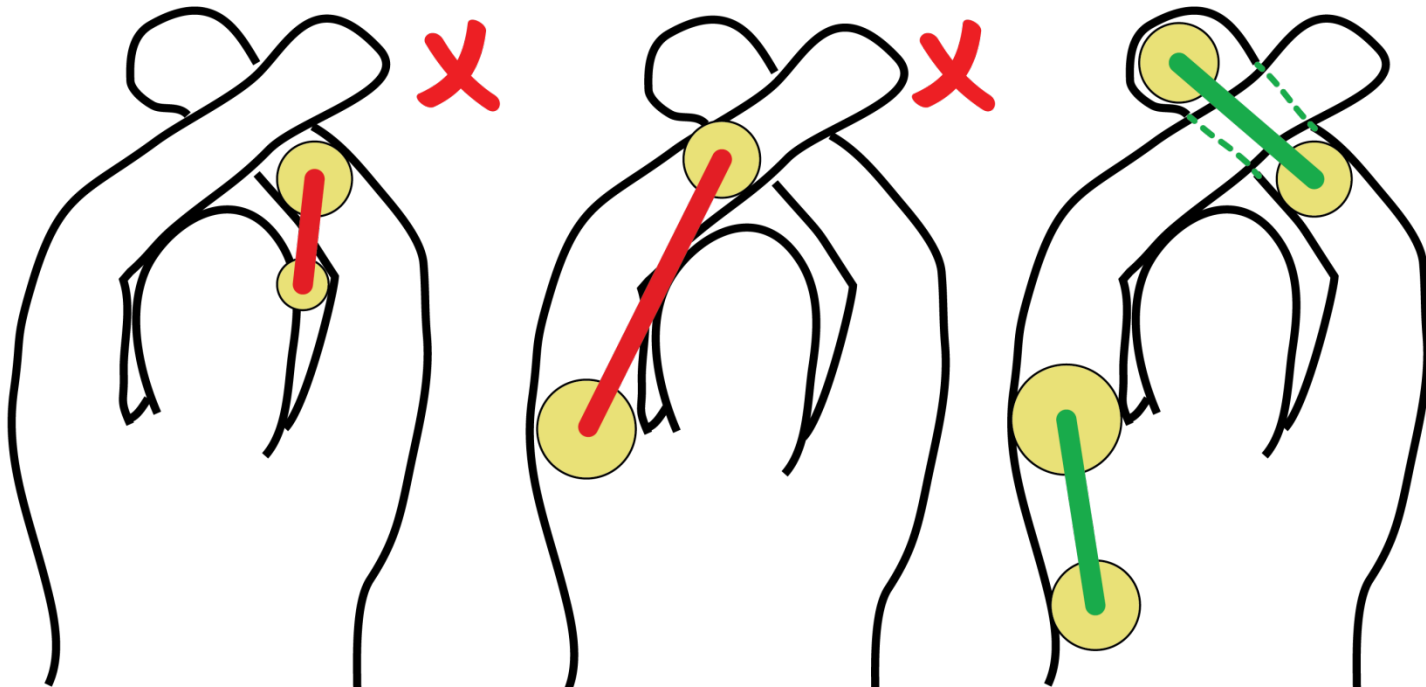
Adjacencies



2D embedding

Orientation

Gestalt continuation

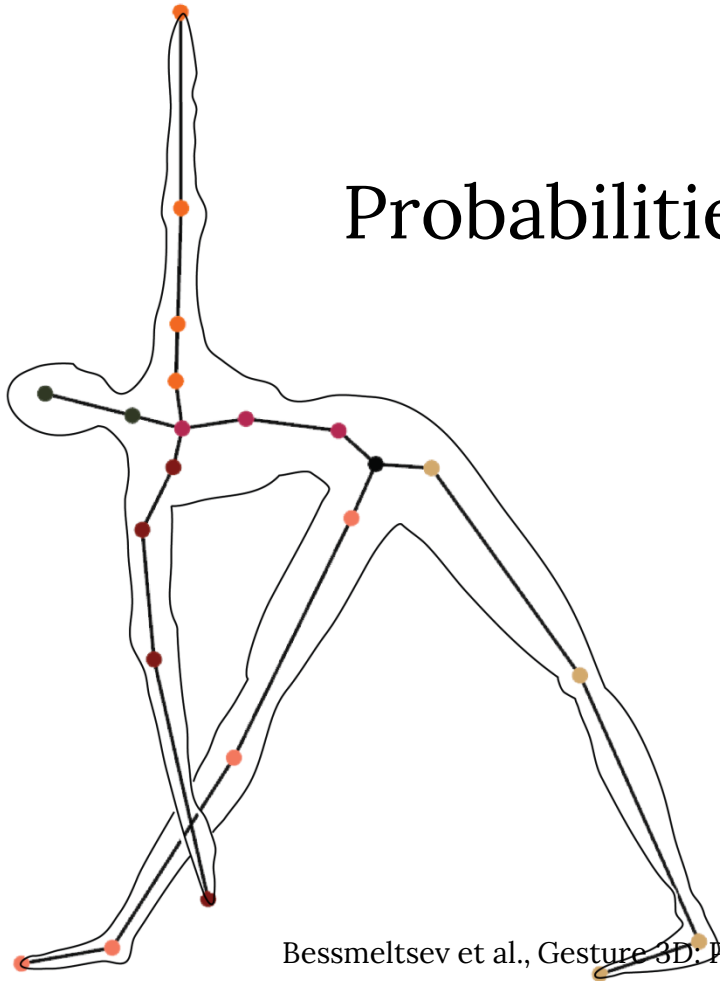


2D embedding

Input: rigged character + gesture drawing

Probabilities of joint and bone locations

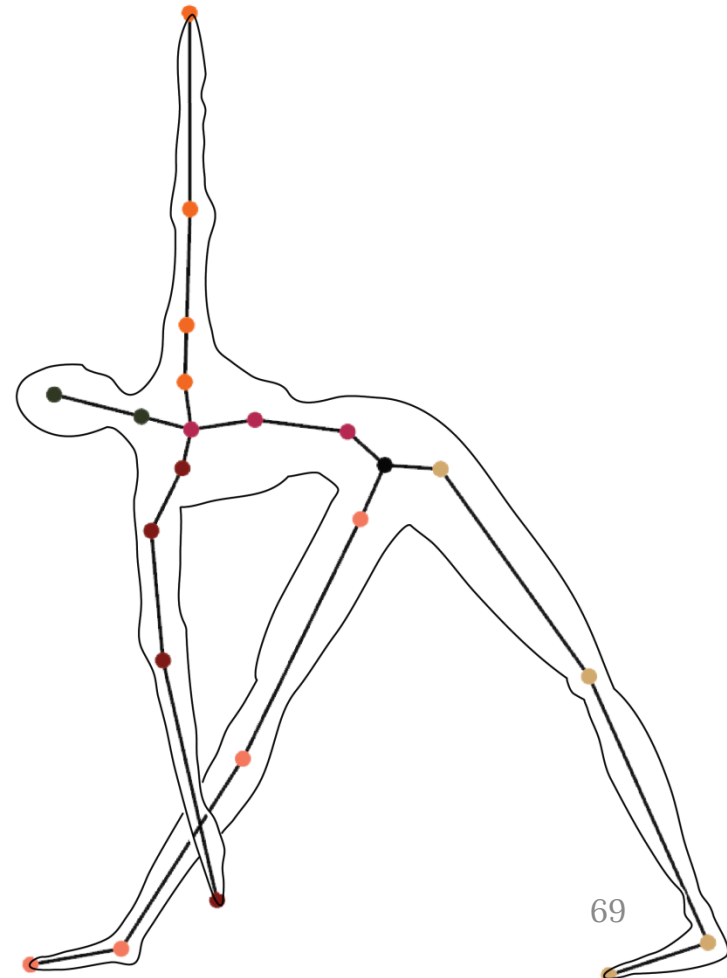
Discrete optimization



3D optimization

Balance 2D embedding with

- Simplicity
- Foreshortening cues



3D optimization

Balance 2D embedding with

- Simplicity
- Foreshortening cues
- Depth order from T-Junctions



Answering your question: yes

3D Sketching using Multi-View Deep Volumetric Prediction

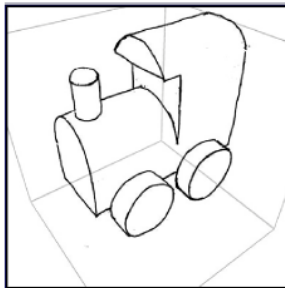
Johanna Delanoy
Inria Université Côte d'Azur

Mathieu Aubry
LIGM (UMR 8049), Ecole des Ponts

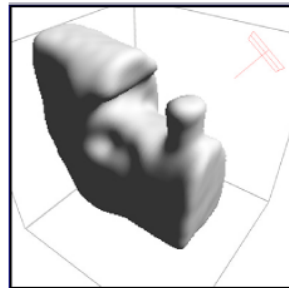
Phillip Isola
OpenAI

Alexei A. Efros
UC Berkeley

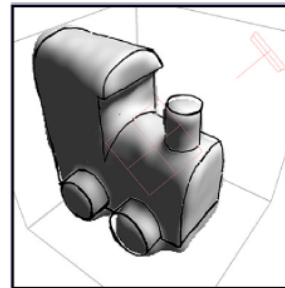
Adrien Bousseau
Inria Université Côte d'Azur



a) Initial drawing



b) 3D prediction
seen from another viewpoint



c) New drawing
and updated prediction



d) 3D printed objects

Figure 1: Our sketch-based modeling system can process as little as a single perspective drawing (a) to predict a volumetric object (b). Users can refine this prediction and complete it with novel parts by providing additional drawings from other viewpoints (c). This iterative sketching workflow allows quick 3D concept exploration and rapid prototyping (d).

Sketches are messy

and often in bitmap format

- Reconstruct directly
 - Probably ML/DL?
- Vectorize
- Simplify

Sketches are messy

and often in bitmap format

