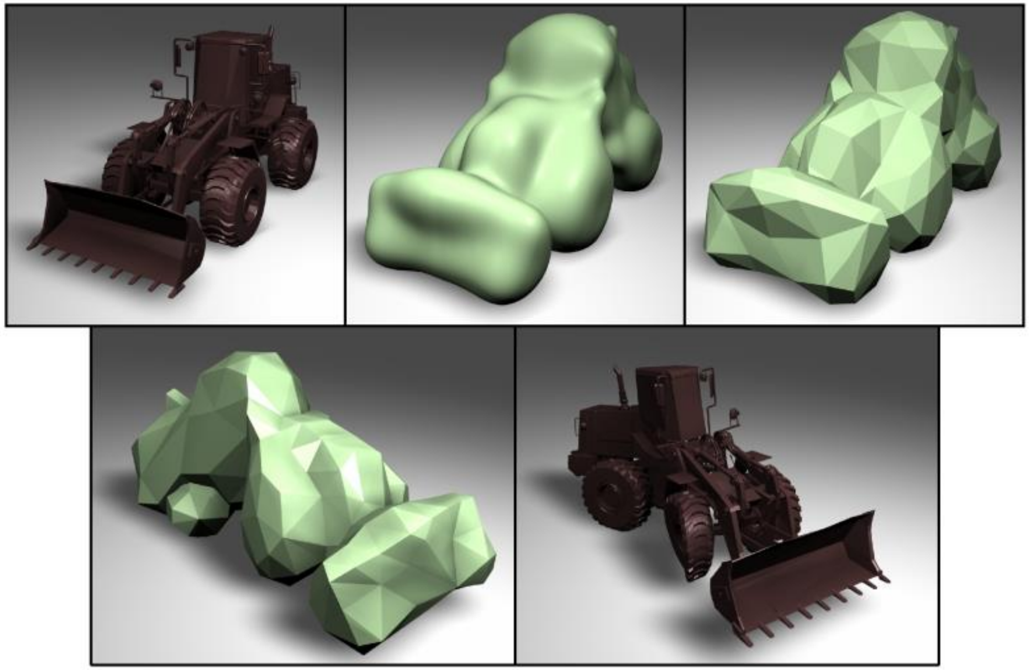


Interpolating and Approximating Implicit Surfaces from Polygon Soup

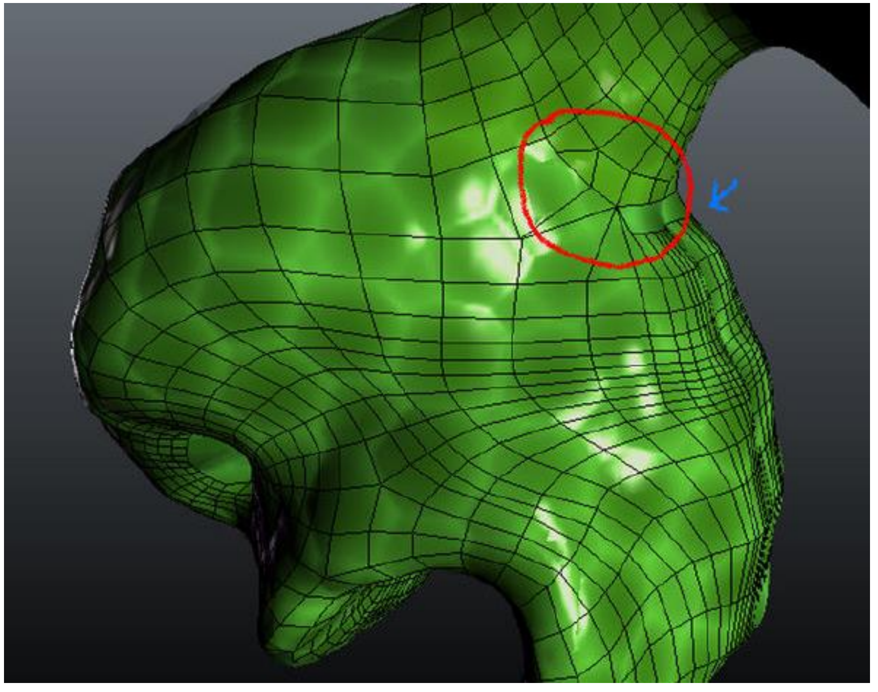
James Doverspike

Motivation

- Polygonal models are everywhere
- Often contain holes, gaps, T-junctions, self-intersections, and non-manifold structure
  - Precludes their use in anything but rendering
- Excessive detail or triangles with bad aspect ratios

Topology

- 3D-CG community shuns the use of triangles because they cause pinches
  - Subdivision and smoothing fail



Polygon density

- Algorithms might assume density corresponds to detail
  - Often not the case → "Triangle soup"
- Artists may add or remove polygons independent of detail

Purpose

- Transform arbitrary polygon data to an implicit surface that is approximative or interpolative
  - Approximation controlled by minimum feature size parameter
    - Zero corresponds to exactly interpolative
- Surface is now manifold and water-tight

Properties

- Surface is the convex hull of input
  - Family of increasingly smooth approximations that converge to a circumscribing ellipsoid
- Degree of approximation controlled by MLS weighting function
- A “clean” polygonal model can be extracted

Moving Least-squares

- Two constraints
  - Implicit function forced to value over surface region of each polygon
  - Implicit function upward gradient forced to polygon’s outward gradient
- Tightness of surface and convex hull containment controlled by constraints
- Unrelated to Alexa et al. 2001, IMLS vs. MLS
- Related to Ohtake et al., 2003a

Least-squares

Over-determined linear system

$$\begin{bmatrix} \boldsymbol{b}^\mathsf{T}(\boldsymbol{p}_1) \\ \vdots \\ \boldsymbol{b}^\mathsf{T}(\boldsymbol{p}_N) \end{bmatrix} \boldsymbol{c} = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}$$

Adding weights for moving least-squares

$$\begin{bmatrix} w(\boldsymbol{x}, \boldsymbol{p}_1) & & \\ & \ddots & \\ & & w(\boldsymbol{x}, \boldsymbol{p}_N) \end{bmatrix} \begin{bmatrix} \boldsymbol{b}^\mathsf{T}(\boldsymbol{p}_1) \\ \vdots \\ \boldsymbol{b}^\mathsf{T}(\boldsymbol{p}_N) \end{bmatrix} \boldsymbol{c} = \begin{bmatrix} w(\boldsymbol{x}, \boldsymbol{p}_1) & & \\ & \ddots & \\ & & w(\boldsymbol{x}, \boldsymbol{p}_N) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}$$

With weights

$$w(r) = \frac{1}{(r^2 + \epsilon^2)}$$

Moving least-squares

Matrix form:

$$\boldsymbol{W}(\boldsymbol{x})\ \boldsymbol{B}\,\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{W}(\boldsymbol{x})\ \boldsymbol{\phi}$$

Normal equations:

$$\boldsymbol{B}^{\top}\ (\boldsymbol{W}(\boldsymbol{x}))^2\ \boldsymbol{B}\ \boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{B}^{\top}\ (\boldsymbol{W}(\boldsymbol{x}))^2\ \boldsymbol{\phi}$$

$$f(\boldsymbol{x}) = \boldsymbol{b}^{\top}(\boldsymbol{x})\ \boldsymbol{H}^{-1}\ \boldsymbol{B}^{\top}\ (\boldsymbol{W}(\boldsymbol{x}))^2\ \boldsymbol{\phi}$$ where $$\boldsymbol{H} = \boldsymbol{B}^{\top}\ (\boldsymbol{W}(\boldsymbol{x}))^2\ \boldsymbol{B}$$

Derivative:

$$f'(\boldsymbol{x}) = (\boldsymbol{b}^{\top})'(\boldsymbol{x})\ \boldsymbol{H}^{-1}\ \boldsymbol{B}^{\top}\ (\boldsymbol{W}(\boldsymbol{x}))^2\ \boldsymbol{\phi}\ -\ \boldsymbol{b}^{\top}(\boldsymbol{x})\ \boldsymbol{H}^{-1}\boldsymbol{H}'\boldsymbol{H}^{-1}\ \boldsymbol{B}^{\top}\ (\boldsymbol{W}(\boldsymbol{x}))^2\ \boldsymbol{\phi}\ +\ \boldsymbol{b}^{\top}(\boldsymbol{x})\ \boldsymbol{H}^{-1}\ \boldsymbol{B}^{\top}\ ((\boldsymbol{W}(\boldsymbol{x}))^2)'\ \boldsymbol{\phi}$$

$$\boldsymbol{H}' = \boldsymbol{B}^{\top}\ ((\boldsymbol{W}(\boldsymbol{x}))^2)'\ \boldsymbol{B}$$

Polygon constraints

- Key insight: integrate least squares from entire polygon and not just the vertices

$$\left(\sum_{k=1}^{K} \boldsymbol{A}_k\right) \boldsymbol{c}(\boldsymbol{x}) = \sum_{k=1}^{K} \boldsymbol{a}_k$$

$$\boldsymbol{A}_k = \int_{\Omega_k} w^2(\boldsymbol{x}, \boldsymbol{p})\, \boldsymbol{b}(\boldsymbol{p})\, \boldsymbol{b}^{\mathsf{T}}(\boldsymbol{p}) \;\, \mathrm{d}\boldsymbol{p} \qquad \boldsymbol{a}_k = \int_{\Omega_k} w^2(\boldsymbol{x}, \boldsymbol{p})\, \boldsymbol{b}(\boldsymbol{p})\, \phi_k \;\, \mathrm{d}\boldsymbol{p}$$

- Large polygons are no longer under-samplings

Results

- Self-intersecting polygons cause saddles
- Implicit surface fills holes

Fast evaluation

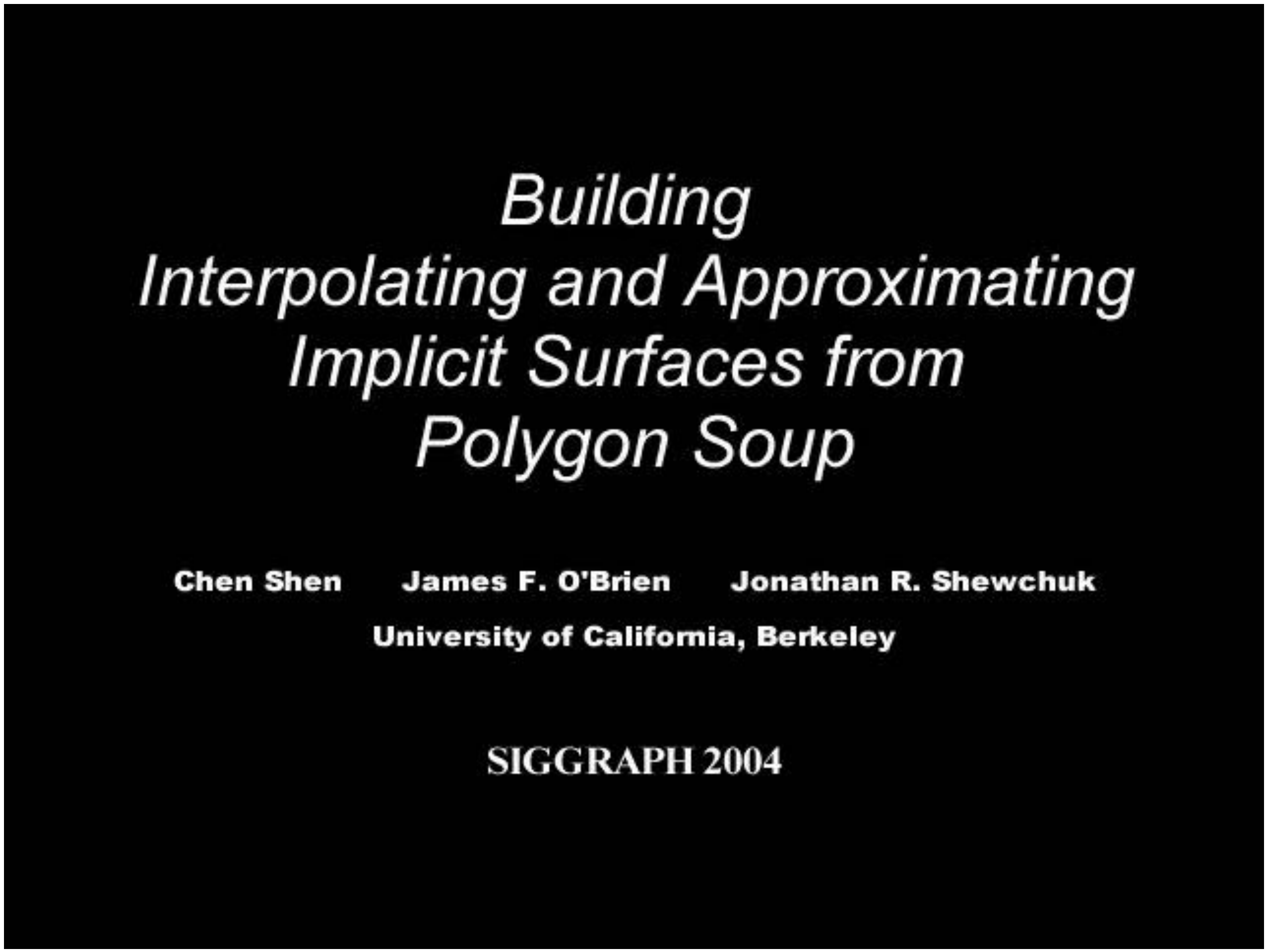
- Triangles stored in a K-D tree with their unweighted integrals

$$\boldsymbol{A}_k = \int_{\Omega_k} w^2(\boldsymbol{x}, \boldsymbol{p})\, \boldsymbol{b}(\boldsymbol{p})\, \boldsymbol{b}^\mathsf{T}(\boldsymbol{p}) \ \mathrm{d}\boldsymbol{p}$$

$$\boldsymbol{a}_k = \int_{\Omega_k} w^2(\boldsymbol{x}, \boldsymbol{p})\, \boldsymbol{b}(\boldsymbol{p})\, \phi_k \ \mathrm{d}\boldsymbol{p}$$

- Far triangles are then grouped and weighted by their distance to *x*

Results


Building
Interpolating and Approximating
Implicit Surfaces from
Polygon Soup
Chen Shen
James F. O'Brien
Jonathan R. Shewchuk
University of California, Berkeley
SIGGRAPH 2004