Interpolating and Approximating Implicit Surfaces from Polygon Soup

Chen Shen James F. O'Brien Jonathan R. Shewchuk

Presented by Yuan Chen

Outline

- Overview
- Algorithm
- Value Constraints
- Normal Constraints
- Interpolation and Approximation
- Preprocessing
- Results & Discussion

Overview

- Input: Arbitrary set of polygons
  -- holes, gaps, self-intersections, non-manifold
- Output: implicit surfaces
  -- watertight, with some other nice properties.

Algorithm

- Use MLS to generate the implicit surface
- Enforce true normal constraints
- Adjust the implicit surface to fit tightly around the input polygons and enclose all input vertices.
- A hierarchical fast evaluation scheme
- Optional preprocessing

Value Constraints at Points

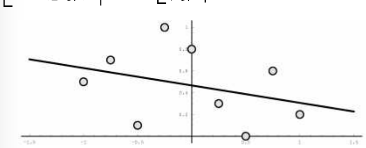
- N points located at positions p_i, $i \in [1 \ldots N]$
- Build a function $f(x)$ which approximates the values ϕ_i at those points
- *b(x)* is the vector of basis functions
- *c* is the unknown vector of coefficients

Value Constraints at Points

- Standard least-squares fit:

$$\begin{bmatrix} b^T(p_1) \\ \vdots \\ b^T(p_N) \end{bmatrix} c = \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}$$

, the resulting function $f(x) = b^T(x)c$



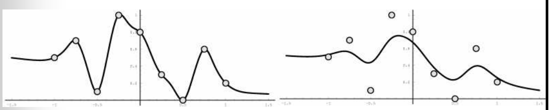
Images from B-CAM website.

Value Constraints at Points

- Moving least-squares fit:

$$\begin{bmatrix} w(x,p_1) & & \\ & \ddots & \\ & & w(x,p_N) \end{bmatrix} \begin{bmatrix} b^T(p_1) \\ \vdots \\ b^T(p_N) \end{bmatrix} c = \begin{bmatrix} w(x,p_1) & & \\ & \ddots & \\ & & w(x,p_N) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}$$

-weighting function $w(x, p_i)$ is used to vary c depending on x.



Images from B-CAM website.

Value Constraints at Points

$$\begin{bmatrix} w(x,p_1) & & \\ & \ddots & \\ & & w(x,p_N) \end{bmatrix} \begin{bmatrix} b^T(p_1) \\ \vdots \\ b^T(p_N) \end{bmatrix} c = \begin{bmatrix} w(x,p_1) & & \\ & \ddots & \\ & & w(x,p_N) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \vdots \\ \phi_N \end{bmatrix}$$

$$\Downarrow$$

$$W(x)Bc(x) = W(x)\phi$$

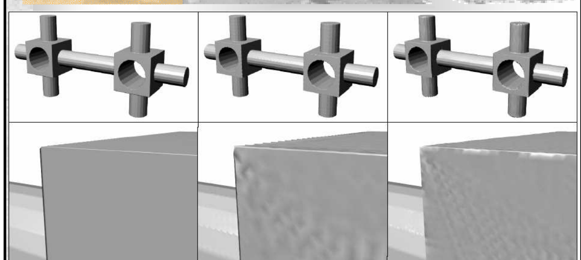
$$\Downarrow$$

The normal equations: $B^T(W(x))^2 Bc(x) = B^T(W(x))^2\phi$

The fit function: $f(x) = b^T(x)H^{-1}B^T(W(x))^2\phi$

Value Constraints at Points

- Using point constraints scattered over input polygons
  -- potentially a large number of points
  -- undesirable bumps and dimples
- So, scatter an infinite number of points continuously across the surface of each polygon.



Generated using integrated polygonal constraints

Generated using different densities of scattered point constraints

Images from Shen04 paper.

Value Constraints over Polygons

- MLS formulation for point constraints:

$$B^T(W(x))^2 Bc(x) = B^T(W(x))^2\phi$$

$$\Downarrow$$

$$\left(\sum_{i=1}^{N} w^2(x,p_i)b(p_i)b^T(p_i)\right)c(x) = \sum_{i=1}^{N} w^2(x,p_i)b(p_i)\phi_i$$

Value Constraints over Polygons

- Integrate value constraints over K polygons, $\Omega_k, k \in [1...K]$:

$$\left(\sum_{i=1}^{N} w^2(x,p_i)b(p_i)b^T(p_i)\right)c(x) = \sum_{i=1}^{N} w^2(x,p_i)b(p_i)\phi_i$$

$$\Downarrow$$

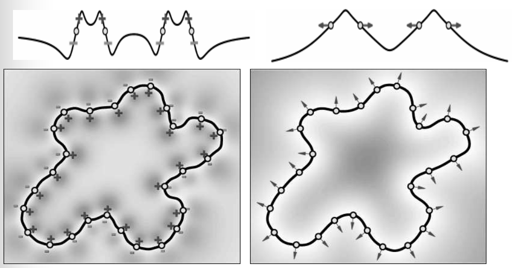
$$\left(\sum_{k=1}^{K} A_k\right)c(x) = \sum_{k=1}^{K} a_k, \quad A_k = \int_{\Omega_k} w^2(x,p)b(p)b^T(p)dp$$

$$a_k = \int_{\Omega_k} w^2(x,p)b(p)\phi_k dp$$

Normal Constraints

- One of popular normal constraints: pseudo-normal constraints: [Turk and O'Brien, 1999]
  - -- A zero constraint at a point on the surface;
    A positive constraint offset slightly outside the surface
    A negative one slightly inside
  - -- This approach does not work well.
    - . an essential singularity at that point.
    - . undesirable oscillatory

Normal Constraints



Images from B-CAM website.

Normal Constraints

- A function *S(x)* associated with polygon Ω_k is defined , with normal constraints:

$$S_k(x) = \phi_k + (x - q_k)^T \tilde{n}_k$$
$$= \psi_{ok} + \psi_{xk} x + \psi_{yk} y + \psi_{zk} z$$

$\tilde{n}$ is the normal of polygon, p is an arbitrary point on the polygon.

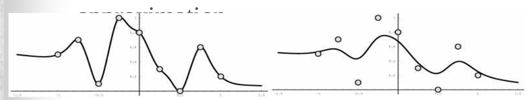
- These functions are blended using MLS instead of constant values.

Interpolation and Approximation

- The weighting function in the paper is set to be: $w(x, p_i) = \dfrac{1}{\|x - p_i\|^2 + \varepsilon^2}$

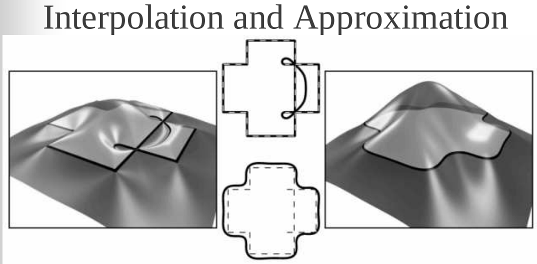
The parameter ε allows a degree of control over the function' s behavior.

- ε =0, interpolation ε !=0, approximation



Images from B-CAM website.

Interpolation and Approximation



The center images show input constraints as dotted lines and the contour as a solid line. The outer images show the resulting function as a height-field.

Images from B-CAM website.

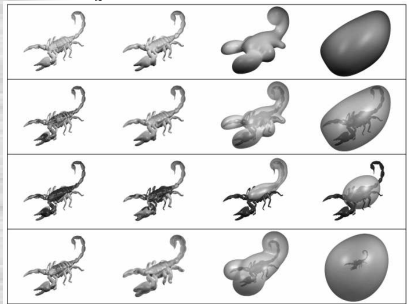
Two problems in approximation

- Surfaces may fall away
  -- ε is large
- The original vertices may fall outside of the implicit surface.

Adjustment

- -- Sample the average value of MLS function with the desired ε over the input polygons.
  -- Extract a surface using that as iso-value.
- --Iteratively adjust the ϕ values assigned to the vertices

Adjustment



Iterative adjustment with different values of ε

The original and constructed surfaces together.

Adjust the surface to average value but no iterative adjustment.

No adjustment at all

Images from B-CAM website.

Preprocessing

- Remove interior polygons
- Force the normal to point outward and be consistent.

Performance

Model	Fig.	P. In	ϵ	V. Out	Time	ϵ	V. Out	Time	ϵ	V. Out	Time	ϵ	V. Out	Time
Heavy Loader	1	37	0	2000	11:42	0.05	800	64:06	5	62	72:48	30	30	92:34
Teapot	10	6.3	0	1000	5:50	0.8	300	10:28	10	53	22:02	60	26	42:31
Cow	10	5.8	0	1000	5:37	0.8	300	8:04	7	61	23:35	120	23	58:19
Bunny	10	69	0	1500	8:13	0.4	400	19:34	10	50	41:36	60	28	72:23
Dragon	10	870	0	2000	12:23	0.6	400	82:21	7	46	89:54	30	21	97:04
Scorpion	10	78	0	1500	9:54	0.6	400	67:50	5	65	61:06	30	26	80:55
Intersecting Star	11	0.05	0	1000	4:44	1	300	5:18	10	48	7:03	110	28	8:20
Machine Part	11	0.8	0	1000	4:45	0.8	300	5:54	7	60	8:50	120	29	20:21
Deck Chair	11	3.9	0	1000	5:01	0.8	300	8:29	10	55	27:06	120	30	52:36
Armchair	11	3.4	0	1000	4:56	0.4	300	7:53	7	62	28:28	120	28	41:09
Cube Shape	11	0.01	0	1000	4:44	0.8	300	5:01	10	45	3:06	80	25	1:52

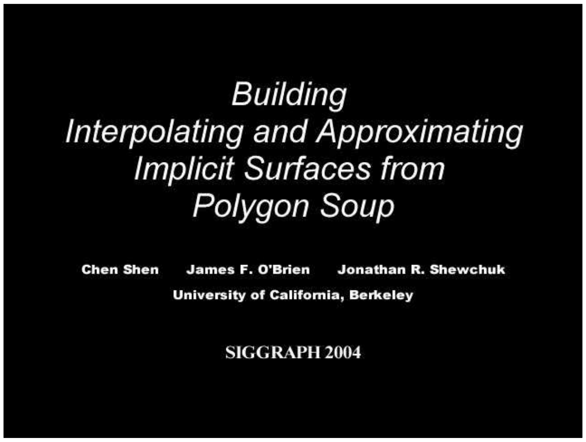
Figure from Shen04 paper.

Discussion

- Watertight & Have to be watertight
- Adjustment procedure….
- Time-consuming…

Movie

from B-CAM website


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