Modelling with piecewise implicit patches

Sipos Ágoston

Budapest University of Technology, Dept. of Control Eng. and Information Technology, Computer Graphics & Geometry Group

2019. 11. 04.

Contents

Contents

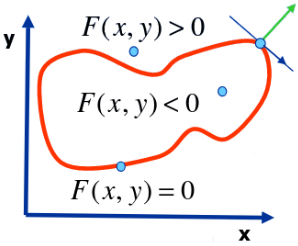
Implicit surfaces

$$f : \mathbb{R}^3 \to \mathbb{R}$$

scalar function

$$\{\mathbf{p} \in \mathbb{R}^3 : f(p) = 0\}$$

implicit surface



Means a space partitioning to „inside” and „outside” which define a separating surface.

Implicit surfaces

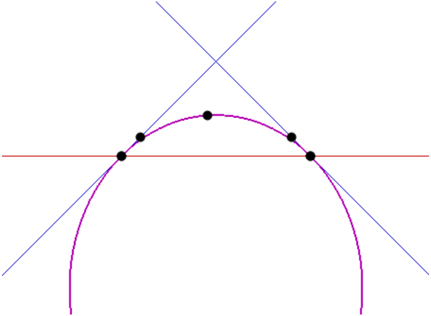
Known implicit functions for standard objects:
Plane: $f(x, y, z) = ax + by + cz + d$
Sphere: $f(x, y, z) = x^2 + y^2 + z^2 - r^2$
Cylinder: $f(x, y, z) = x^2 + y^2 - r^2$
Cone: $f(x, y, z) = x^2 + y^2 - z^2$
Torus: $f(x, y, z) = (\sqrt{x^2 + y^2} - R)^2 + z^2 - r^2$

Lacks easy algorithm for evaluation
However, many things (approximation, offsetting) can be easier
Hardware-accelerated rendering might be available

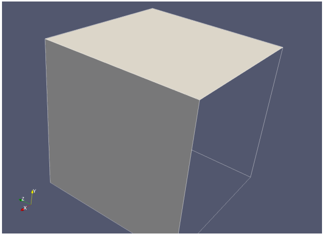
Contents

Liming's method

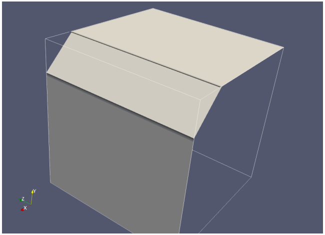
- We have two straight lines ($L_1 = 0, L_2 = 0$) and want their corner to be blended
- We take a cutting line ($C_1 = 0$)
- $F = (1-\lambda)L_1 L_2 + \lambda C_1^2$ is a correct blend curve in implicit form
- F touches L_1 and L_2 with G^1 continuity



Liming's method in 3D

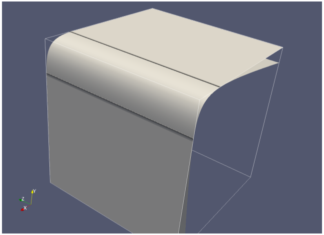


Original edge

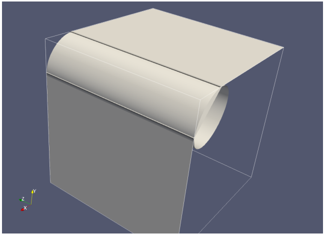


Cut edge

Liming's method in 3D



Liming-surface, $\lambda = 0.2$

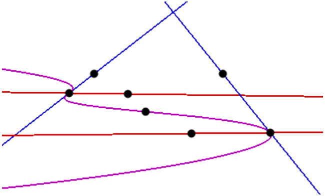


Liming-surface, $\lambda = 0.6$

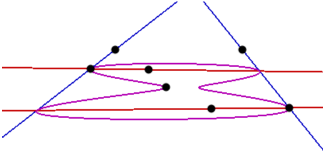
Functional spline

- Direkt generalization of Liming's method using the fact that the union of n implicit surfaces is their product
- $F = (1-\lambda)\prod_{i=1}^{n} P_i + \lambda \prod_{i=1}^{m} B_i^2$, where P_i, $i = 1..n$ primary surfaces (requiring G^1 continuous connection), B_i, $i = 1..m$ cutting surfaces
- Problem: by unioning things we lose logical grouping

Functional spline



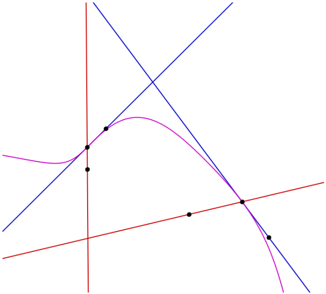
Aesthetic curve



Functional spline

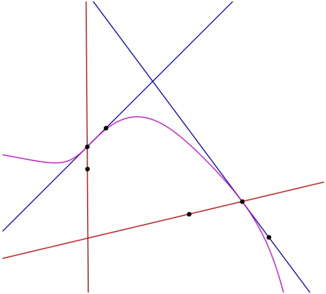
I-Patch

- P_i, $i = 1..n$ primary surfaces, B_i, $i = 1..n$ boundary surfaces
- Surfaces with the same index together represent a side of the patch
- $I = \sum_{i=1}^{n} (w_i P_i \prod_{j \neq i} B_j^2) + w \prod_{i=1}^{n} B_i^2$, $w_i, w \in \mathbb{R}$



- E.g. $n = 2$:
  $I = w_1 P_1 B_2^2 + w_2 P_2 B_1^2 + w B_1^2 B_2^2$
- n real coefficients (the equation can be divided with a scalar)

I-Patch – Properties



$I = w_1 P_1 B_2^2 + w_2 P_2 B_1^2 + w B_1^2 B_2^2$

Boundary interpolation:

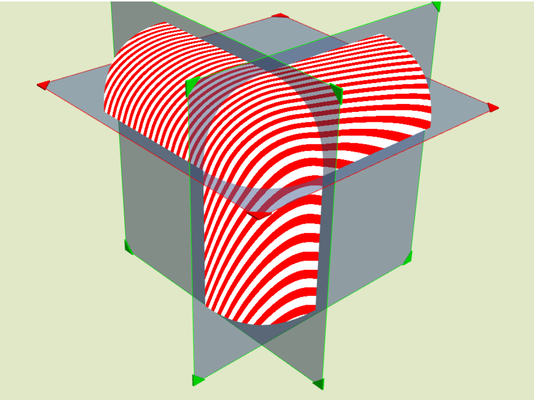
- i^{th} side: $\{P_i = 0\} \cap \{B_i = 0\}$
- The equation is trivially zero

G^1 continuity:

- on the i^{th}. side to P_i
- $I = QP_i + WB_i^2, \qquad Q, W : \mathbb{R}^n \to \mathbb{R}$
- $(\nabla I) = (\nabla Q) P_i + Q (\nabla P_i) + (\nabla W) B_i^2 + W 2B_i (\nabla B_i) =$
  $= 0 + Q (\nabla P_i) + 0 + 0$
- If exponents in formula are $n+1$: G^n continuity

3D equations - with ribbons

$$I = \sum_{i=1}^{n} (w_i P_i \prod_{j \neq i} B_j^2) + w \prod_{i=1}^{n} B_i^2, \; w_i, w \in \mathbb{R}$$



3D equation - with corners

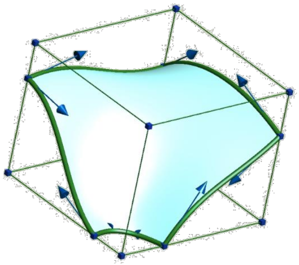
$$\sum_{k=1}^{n} C_{k,k+1}\left(\prod_{i\neq k, i\neq k+1} B_i^2\right) + \sum_{k=1}^{n} w_k\left(\prod_{i\neq k} B_i^2\right) + w\prod_{i=1}^{n} B_i^2$$

Contents

Cell-based modelling

- For a cell-partitioning, have an implicit function for each cell
- Continuity constraints on cell-boundary

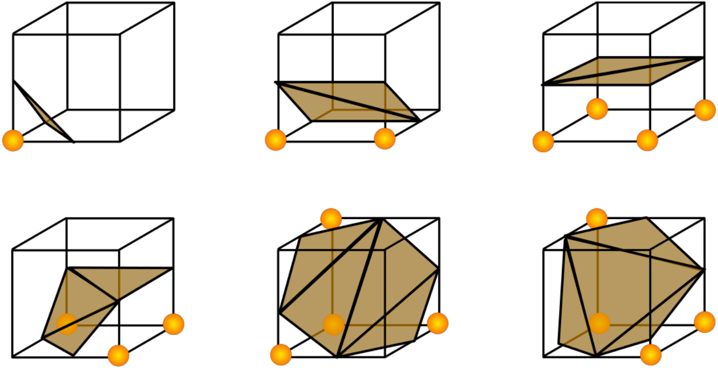
Input: tangential planes in corner points



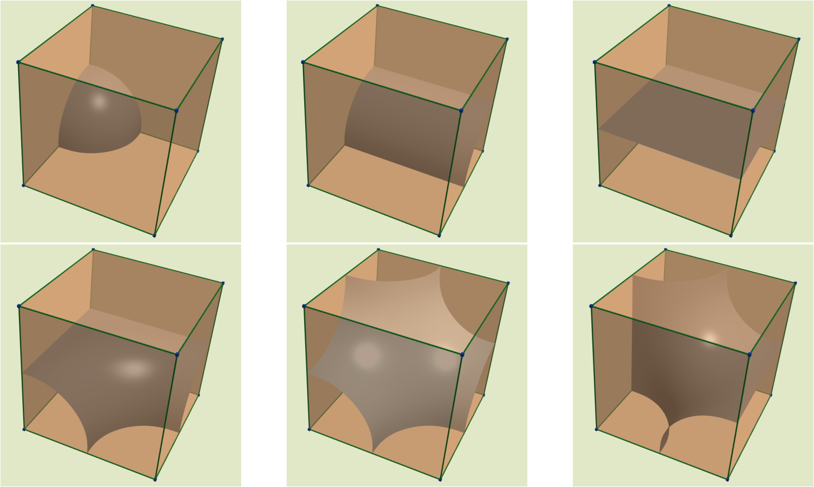
N-sided patch

Topology

Classifying vertices

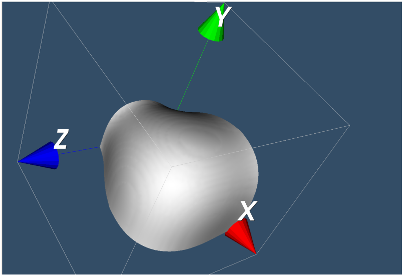


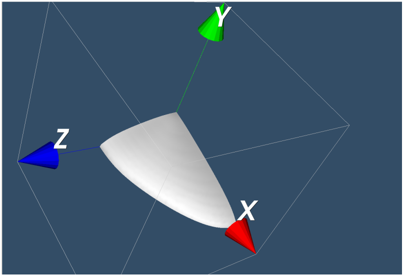
„Ideal" patches



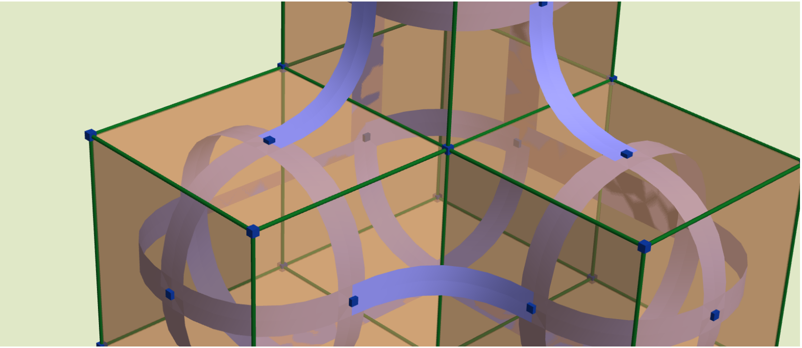
Problems

Automatic calculation of coefficients is needed.

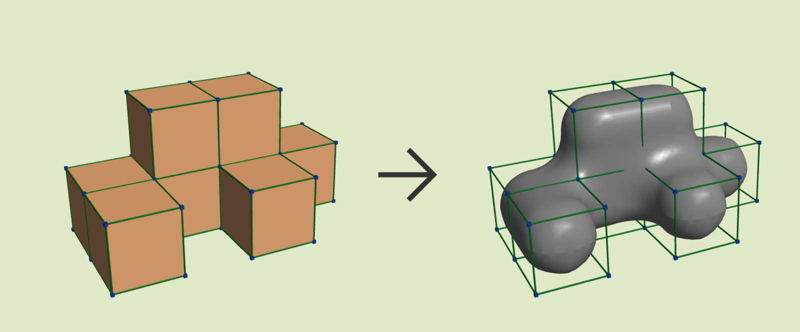




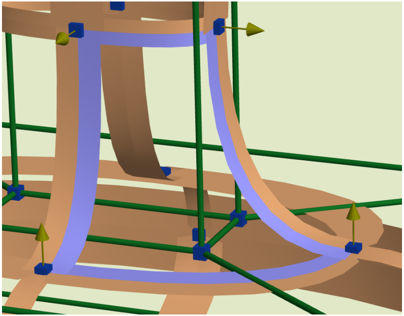
Polyhedral blending

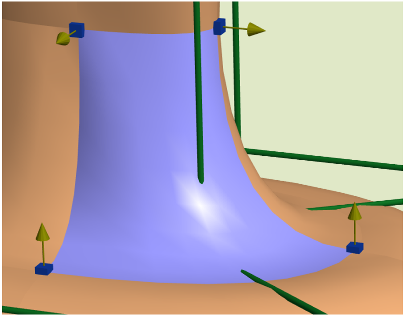


Polyhedral blending



More difficult configurations





Summary

Advantages:

- Free-form geometry in implicit form.
- More practical for fitting, offsetting and raycasting
- Automatic fulfillment of continuity constraints

Difficulties:

- Not easy to handle all geometric configurations correctly
- Automatic setting of coefficients is not trivial

Thank you for your attention!