Multi-sided Bézier surfaces over concave polygonal domains

Péter Salvi, Tamás Várady

Budapest University of Technology and Economics

SMI 2018

Lisbon, June 6th–8th
Outline

1. Introduction
   - Motivation
   - Previous work
2. Generalized Bézier (GB) patch
   - Control structure
   - Domain & parameterization
   - Blending functions
3. Concave GB patch
   - Reinterpretation
   - Building blocks
   - Additional control
4. Examples
5. Conclusion and future work

P. Salvi, T. Várady
BME
Multi-sided Bézier surfaces over concave polygonal domains
Multi-sided patches

- Curve network based design
  - Feature curves
  - Automatic surface generation
- Hole filling
  - E.g. vertex blends
  - Cross-derivative constraints
- “Concave” configurations

Representation?
Conventional representations used in CAD systems

- Trimmed tensor product surfaces
  - Detailed interior control
  - Continuity problems
  - Different edge types
    ⇒ inherently asymmetric
- Division into smaller quadrilaterals
  - (Semi-)automatic splitting curves
  - Underdetermined entities
  - Reduced continuity
- Our goals:
  - $C^\infty$ continuity
  - Editing with control points
    (with interior control)
  - No additional artificial curves
Concave surface representations

- Loop and DeRose (1989), Smith and Schaefer (2015)
  - S-patches – multivariate Bézier patches
  - Beautiful theory
  - Difficult to use

  - Transfinite surface interpolation
  - Supports holes
  - No internal control
  - Singular blends cause high curvature variation

- Pan et al. (2015), Stanko et al. (2016)
  - Discrete methods minimizing fairness energies

(See comparisons later)
Generalized Bézier (GB) patch

T. Várady, P. Salvi, Gy. Karikó,
A Multi-sided Bézier Patch with a Simple Control Structure.

T. Várady, P. Salvi, I. Kovács,
Enhancement of a multi-sided Bézier surface representation.
Control net derivation from the quadrilateral case

- Control grid $\rightarrow n$ ribbons
- Degree: $d$, Layers: $l = \lceil \frac{d^2}{2} \rceil$
- Control points: $C_{j,k}^i$
  - $i \in [1..n]$, $j \in [0..d]$, $k \in [0..l-1]$
- Weighting functions: $\mu_{j,k}^i$
Domain

- Regular domain in the \((u, v)\) plane
- Side-based local parameterization functions \(s_i\) and \(h_i\)
  - Based on Wachspress barycentric coordinates \(\lambda_i(u, v)\)
Local parameters

- \( s_i = \frac{\lambda_i}{\lambda_{i-1} + \lambda_i} \)
- \( h_i = 1 - \lambda_{i-1} - \lambda_i \)

Barycentric coordinates \( \lambda_i \)

- \( \lambda_i \geq 0 \) [positivity]
- \( \sum_{i=1}^{n} \lambda_i = 1 \) [partition of unity]
- \( \sum_{i=1}^{n} \lambda_i(u, v) \cdot P_i = (u, v) \) [reproduction]
- \( \lambda_i(P_j) = \delta_{ij} \) [Lagrange property]
Half-Bézier ribbons

\[ R_i(s_i, h_i) = \sum_{j=0}^{d} \sum_{k=0}^{l-1} C_{j,k}^i \cdot \mu_{j,k}^i B_{j,k}^i(s_i, h_i) \]

- \( C_{j,k}^i \): j-th CP on side i, layer k
- \( B_{j,k}^i(s_i, h_i) = B_j^d(s_i) \cdot B_k^d(h_i) \)
  - bivariate Bernstein polynomials
- \( \mu_{j,k}^i \): rational function of \( h_i, h_{i \pm 1} \)
  - 1 on side i, 0 on the others
- The surface interpolates the ribbons at the boundary (\( G^1/G^2 \))
Central weight & patch equation

- Weights do not add up to 1
- Deficiency $\Rightarrow$ weight of the central control point:

$$B_0(u, v) = 1 - \sum_{i=1}^{n} \sum_{j=0}^{d} \sum_{k=0}^{l-1} \mu_{j,k}^i B_{j,k}^i(s_i, h_i)$$

- Patch equation:

$$S(u, v) = \sum_{i=1}^{n} R_i(s_i, h_i) + C_0 B_0(u, v)$$
Degree elevation

- Linear and bilinear combinations
- Modifies the surface interior
- Control net generated by reductions and elevations
  - Default positions
  - Merging Bézier ribbons of different degrees
Concave GB patch
Problem: ribbon orientation

Convex case: prev. tangent $\rightarrow$ next tangent

- Does not work for the concave case
- Interpolants should point towards the interior of the surface
- Control point placement?
Independent ribbons

- Original constraints of the GB patch:
  - Common $d$ degree, same $l = \lceil d/2 \rceil$ number of layers
  - Corresponding control points of adjacent ribbons are identical

- These can be lifted! $\Rightarrow$ Ribbons become independent entities

- $\mu_{j,k}^i$ weight function still ensures the interpolating property

- Local $d_i$ and $l_i$ values for each ribbon

- Various possible configurations:

  4 + 4 $\rightarrow$ 4
  4 + 4 $\rightarrow$ 5
  9 + 9 $\rightarrow$ 13
  9 + 4 $\rightarrow$ 12
  4 + 4 $\rightarrow$ 7
Building blocks

Ribbons

\[ R_i(s_i, h_i) = \sum_{j=0}^{d_i} \sum_{k=0}^{l_i-1} C_{j,k} \cdot \mu_{j,k} B_{j,k}^i(s_i, h_i) \]

- \((d_i + 1) \times l_i\) control points
- Degrees:
  - \(d_i\) (edgewise)
  - \(2l_i - 1\) (cross-boundary)
- Degree elevation:
  - Independently in the two parametric directions
  - Adding a layer increases the degree by 2

P. Salvi, T. Várady
Multi-sided Bézier surfaces over concave polygonal domains
Blending functions

- Blend of \( C_{j,k}^i \) is \( \mu_{j,k}^i B_{j,k}^i(s_i, h_i) \), where

\[
B_{j,k}^i(s_i, h_i) = B_{j,d_i}^i(s_i) \cdot B_{k}^{2l_i-1}(h_i)
\]

\[
\mu_{j,k}^i = \begin{cases} 
\alpha_i = h_{i-1}^2 / \left( h_{i-1}^2 + h_i^2 \right), & \text{when } 2j < d \\
1, & \text{when } 2j = d \\
\beta_i = h_{i+1}^2 / \left( h_{i+1}^2 + h_i^2 \right), & \text{when } 2j > d 
\end{cases}
\]

- No central control point
  \[ \Rightarrow \text{weight deficiency solved by normalization:} \]

\[
S(u, v) = \frac{1}{B_{\text{sum}}(u, v)} \cdot \sum_{i=1}^{n} R_i(s_i, h_i)
\]
Introduction

Generalized Bézier patch

Concave GB patch

Examples

Conclusion

Building blocks

Blending function examples

\[ \mu_{0,0} B_0^3(s_1) B_0^3(h_1) \]

\[ \mu_{1,0} B_1^3(s_1) B_0^3(h_1) \]

\[ \mu_{1,1} B_1^3(s_1) B_1^3(h_1) \]

\[ \mu_{0,1} B_0^3(s_1) B_1^3(h_1) \]
Domain generation – projection

- Project vertices on a best fit plane
- Simple
- Works well on:
  - Relatively flat objects
- Fails for:
  - Highly curved models
- Goals:
  - Preserve angles
  - Preserve arc lengths
Domain generation – heuristic algorithm

- Normalize the angles
- Draw an open polygon
- Distribute the deviation
  - Proportionally to edges
- Better results:
Domain generation – validation

- The domain may have self-intersections / bottlenecks
- Minimum segment–segment distance parameter
  - E.g. 10% of the MBR axis
- Enlarge all convex angles
  - Enlargement factor (e.g. 1.1)
  - Distribute the surplus among the concave angles
- Iterate until the domain becomes valid
Parameterization

- Based on harmonic coordinates (computed discretely)
Editing with additional control points

- “Hollow” areas of low weight
  1. Concave corners
  2. Areas far from boundaries

Concave corner blend:
- using $B_{1}^{2l_{i}−1}(h_{i})$ weights

Central blend:
- $\prod_{i} h_{i}^{2}$ (scaled)
Examples
S-patch control net (full, “ribbons”)
S-patch vs. GB patch ribbons
Test object #2
Kato’s transfinite patch vs. GB (mean curvature, contours)
<table>
<thead>
<tr>
<th>Introduction</th>
<th>Generalized Bézier patch</th>
<th>Concave GB patch</th>
<th>Examples</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test object #3 (mean curvature, contours)
Test object #3 (isophotes)
Editing with the central control point (contours)
Network of patches (control networks)
Network of patches (mean curvature, contours)
Editing – corner CPs (mean curvature, contours)
Editing – degree elevation (mean curvature, contours)
Conclusion
Summary

Extension of Generalized Bézier patches
- Implicit assumptions lifted
  - Independent degrees / control points
- New ribbons & blending weights
- Concave domain generation & parameterization
- Additional control points

Future work
- Interior control
  - How to add more control points?
  - How to define good blending functions?
- Alternative parameterization?
Any questions?

Thank you for your attention.