Multi-sided Bézier surfaces over concave polygonal domains

Péter Salvi, Tamás Várady

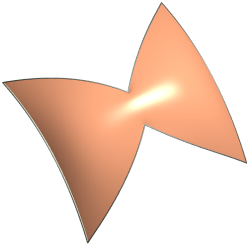
Budapest University of Technology and Economics

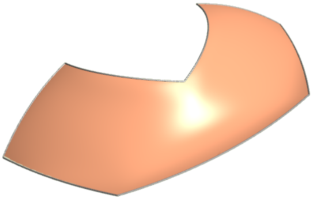
SMI 2018

Lisbon, June 6th–8th

Outline

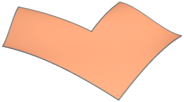
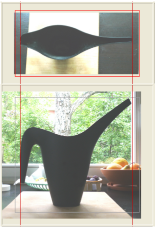
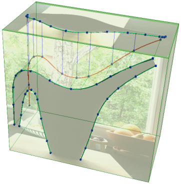
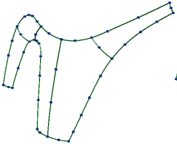
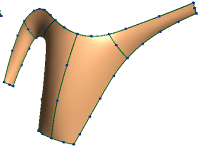
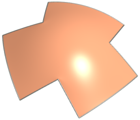
1. Introduction
   - Motivation
   - Previous work
2. Generalized Bézier (GB) patch
   - Control structure
   - Domain & parameterization
   - Blending functions
3. Concave GB patch
   - Reinterpretation
   - Building blocks
   - Additional control
4. Examples
5. Conclusion and future work





Multi-sided patches

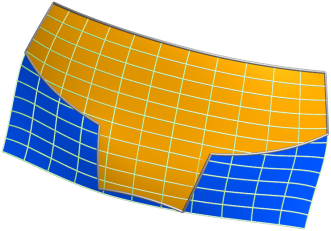
- Curve network based design
  - Feature curves
  - Automatic surface generation
- Hole filling
  - E.g. vertex blends
  - Cross-derivative constraints
- "Concave" configurations

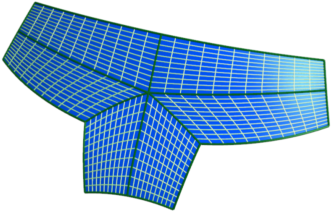







Representation?

Conventional representations used in CAD systems

- Trimmed tensor product surfaces
  - Detailed interior control
  - Continuity problems
  - Different edge types
    $\Rightarrow$ inherently asymmetric
- Division into smaller quadrilaterals
  - (Semi-)automatic splitting curves
  - Underdetermined entities
  - Reduced continuity
- Our goals:
  - C^∞ 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
- Kato (1991, 2000)
  - 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)

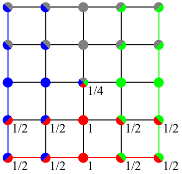
Generalized Bézier (GB) patch

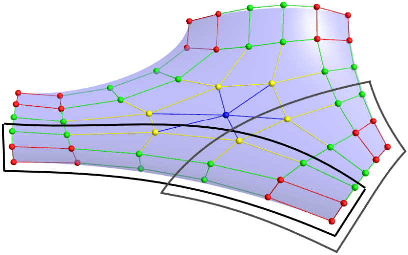
T. Várady, P. Salvi, Gy. Karikó,
A Multi-sided Bézier Patch with a Simple Control Structure.
Computer Graphics Forum, Vol. 35(2), pp. 307-317, 2016.

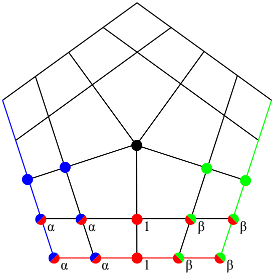
T. Várady, P. Salvi, I. Kovács,
Enhancement of a multi-sided Bézier surface representation.
Computer Aided Geometric Design, Vol. 55, pp. 69-83, 2017.

Control net derivation from the quadrilateral case

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

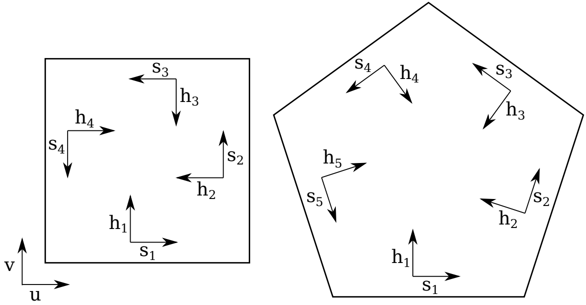






Domain

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

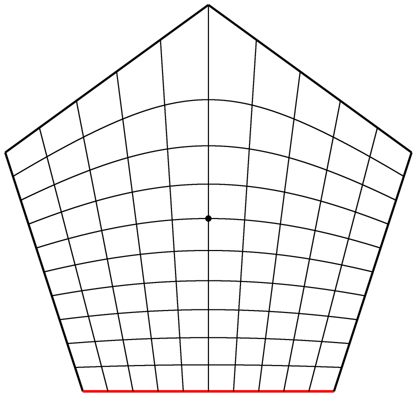


Local parameters

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

Barycentric coordinates λ_i

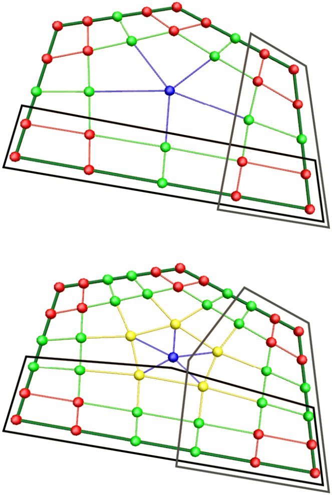
- $\lambda_i \geq 0$
  [positivity]
- $\sum_{i=1}^{n} \lambda_i = 1$
  [partition of unity]
- $\sum_{i=1}^{n} \lambda_i(u, v) \cdot P_i = (u, v)$
  [reproduction]
- $\lambda_i(P_j) = \delta_{ij}$
  [Lagrange property]



Half-Bézier ribbons

$$R_i(s_i, h_i) = \sum_{j=0}^{d} \sum_{k=0}^{l-1} C^i_{j,k} \cdot \mu^i_{j,k} B^i_{j,k}(s_i, h_i)$$

- $C^i_{j,k}$: j-th CP on side i, layer k
- $B^i_{j,k}(s_i, h_i) = B^d_j(s_i) \cdot B^d_k(h_i)$ bivariate Bernstein polynomials
- $\mu^i_{j,k}$ rational function of h_i, $h_{i\pm1}$
  - 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 ⇒ weight of the central control point:

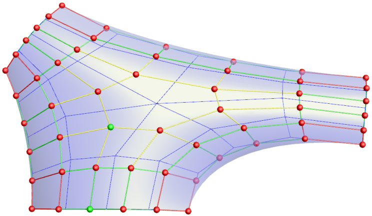
$$B_0(u,v) = 1 - \sum_{i=1}^{n} \sum_{j=0}^{d} \sum_{k=0}^{l-1} \mu_{j,k}^{i} B_{j,k}^{i}(s_i, h_i)$$

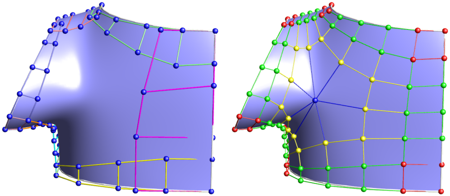
- Patch equation:

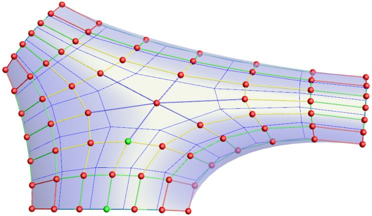
$$S(u,v) = \sum_{i=1}^{n} R_i(s_i, h_i) + C_0 B_0(u,v)$$

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





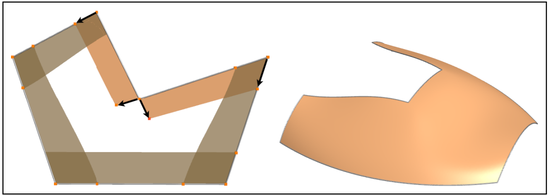


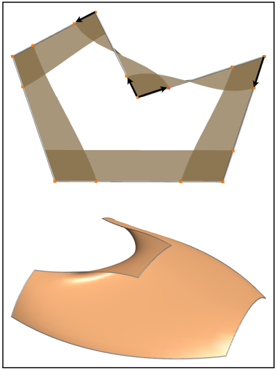
Concave GB patch

Problem: ribbon orientation

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

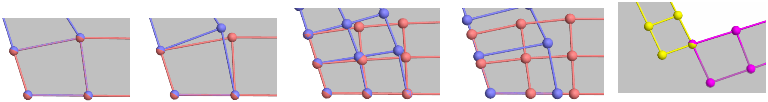
- Does not work for the concave case
- Interpolants should point towards the interior of the surface
- Control point placement?





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
- $\mu^i_{j,k}$ weight function *still* ensures the interpolating property
- Local d_i and l_i values for each ribbon
- Various possible configurations:



$4+4 \to 4$ $\quad$ $4+4 \to 5$ $\quad$ $9+9 \to 13$ $\quad$ $9+4 \to 12$ $\quad$ $4+4 \to 7$

Ribbons

$$R_i(s_i, h_i) = \sum_{j=0}^{d_i} \sum_{k=0}^{l_i-1} C^i_{j,k} \cdot \mu^i_{j,k} B^i_{j,k}(s_i, h_i)$$

- $(d_i + 1) \times l_i$ 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

Blending functions

- Blend of $C^i_{j,k}$ is $\mu^i_{j,k} B^i_{j,k}(s_i, h_i)$, where

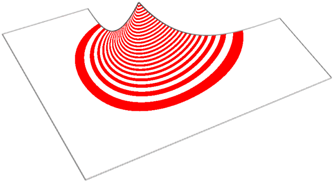
$$B^i_{j,k}(s_i, h_i) = B^{d_i}_j(s_i) \cdot B^{2l_i-1}_k(h_i)$$

$$\mu^i_{j,k} = \begin{cases} \alpha_i = h^2_{i-1} / \left(h^2_{i-1} + h^2_i\right), & \text{when } 2j < d \\ 1, & \text{when } 2j = d \\ \beta_i = h^2_{i+1} / \left(h^2_{i+1} + h^2_i\right), & \text{when } 2j > d \end{cases}$$

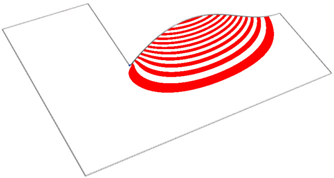
- No central control point
  $\Rightarrow$ weight deficiency solved by normalization:

$$S(u,v) = \frac{1}{B_{\mathrm{sum}}(u,v)} \cdot \sum_{i=1}^{n} R_i(s_i, h_i)$$

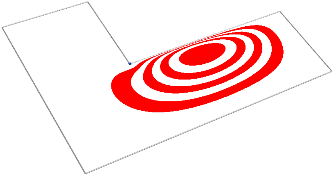
Blending function examples



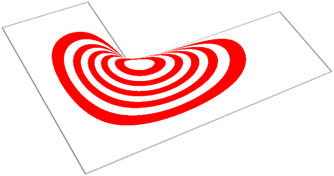
$\mu^1_{0,0} B^3_0(s_1) B^3_0(h_1)$



$\mu^1_{1,0} B^3_1(s_1) B^3_0(h_1)$



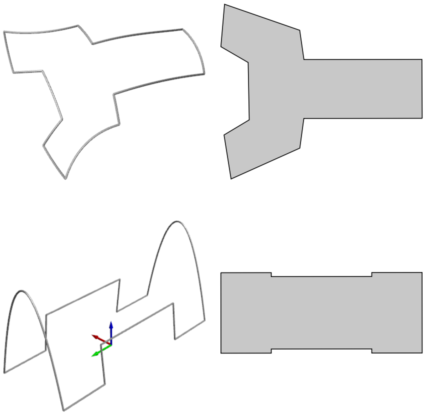
$\mu^1_{1,1} B^3_1(s_1) B^3_1(h_1)$



$\mu^1_{0,1} B^3_0(s_1) B^3_1(h_1)$

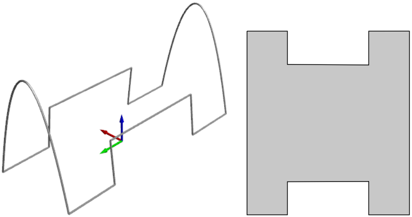
Domain generation – projection

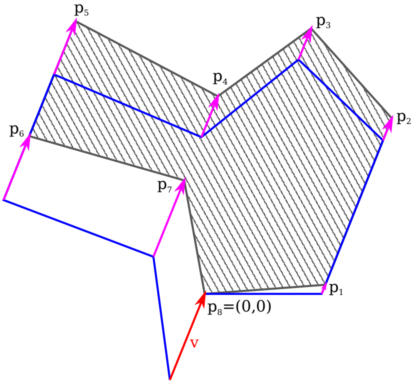
- Project vertices on a best fit plane
- Simple
- Works well on:
  - Relatively flat objects
- Fails for:
  - Highly curved models
- Goals:
  - Preserve angles
  - Preserve arc lengths



Domain generation – heuristic algorithm

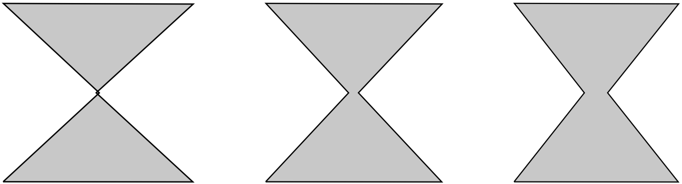
- Normalize the angles
- Draw an open polygon
- Distribute the deviation
  - Proportionally to edges
- Better results:





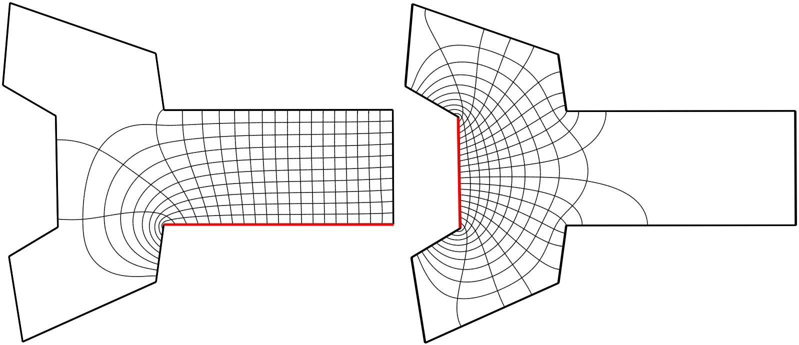
Domain generation – validation

- 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

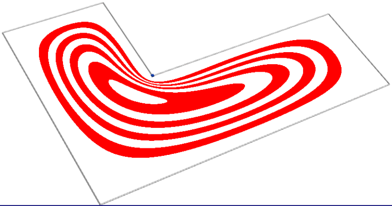
- Based on harmonic coordinates (computed discretely)

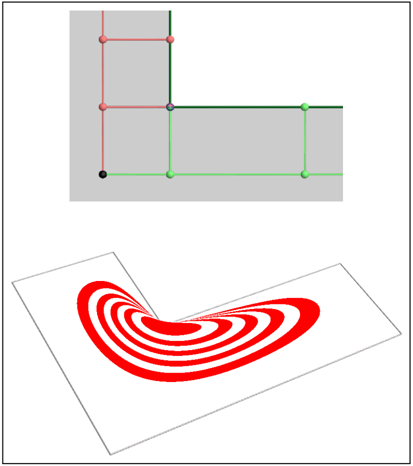


Editing with additional control points

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

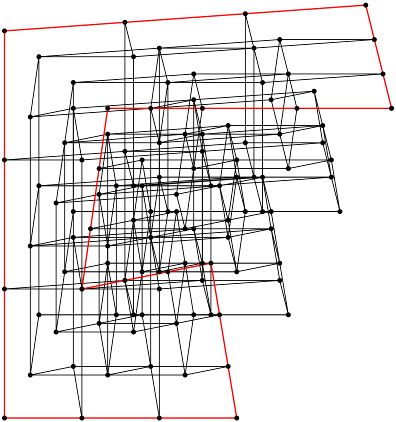
1. Concave corner blend:
   - using $B_1^{2l_i-1}(h_i)$ weights
2. Central blend:
   - $\prod_i h_i^2$ (scaled)

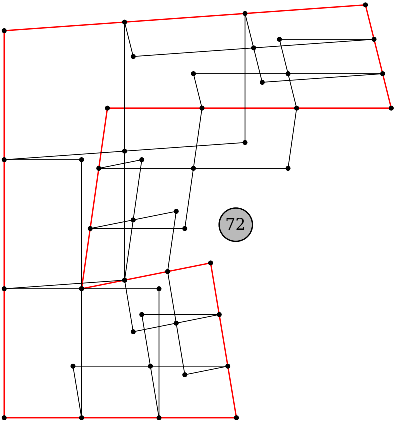




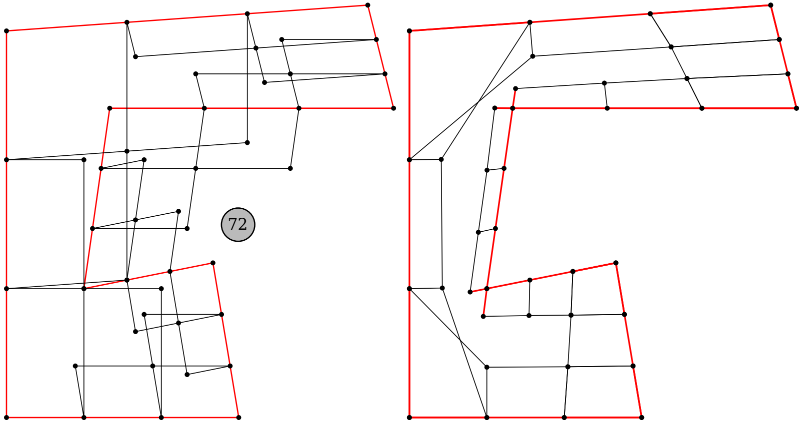
Examples

S-patch control net (full, “ribbons”)

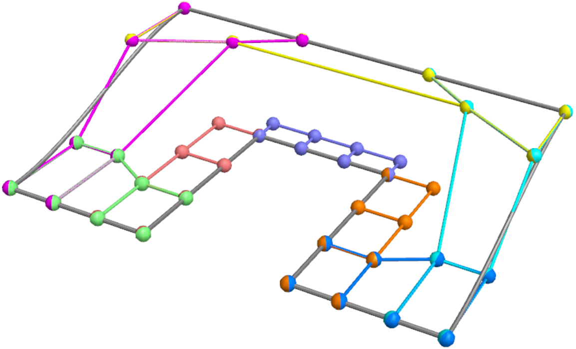




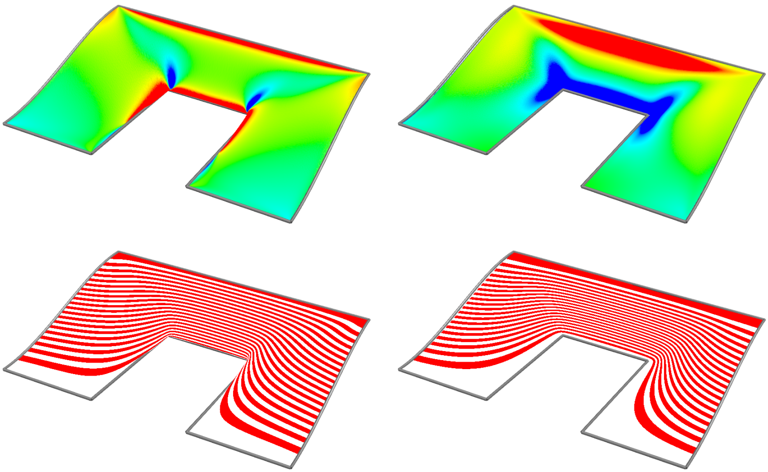
S-patch vs. GB patch ribbons



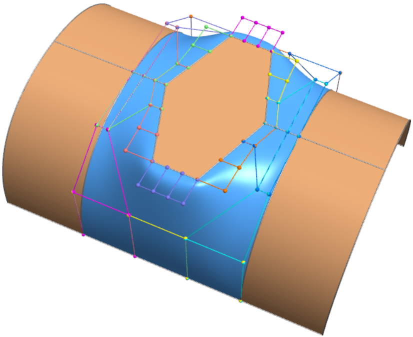
Test object #2

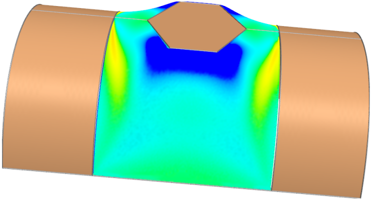


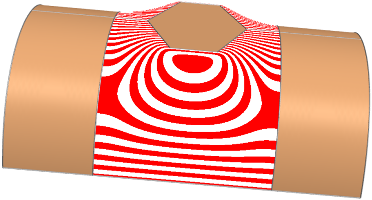
Kato's transfinite patch vs. GB (mean curvature, contours)



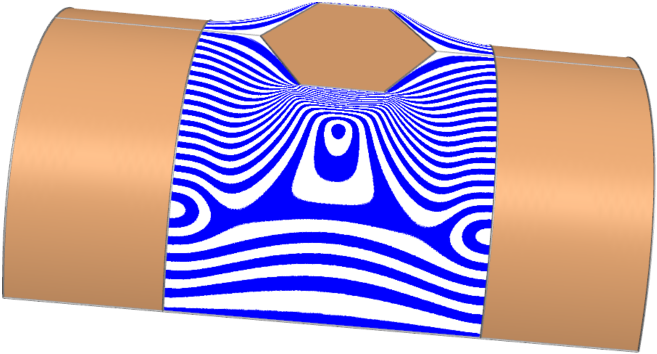
Test object #3 (mean curvature, contours)



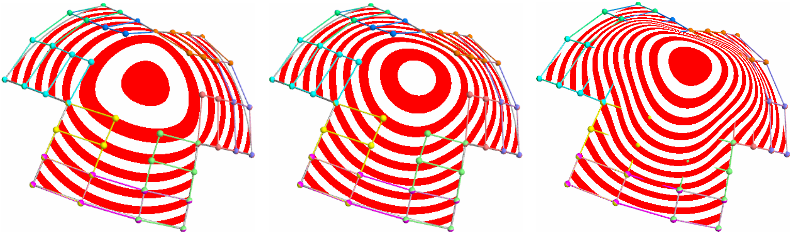




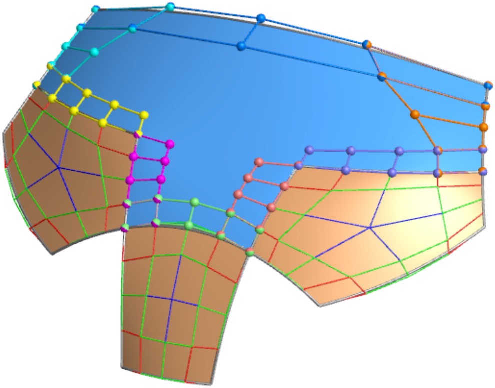
Test object #3 (isophotes)



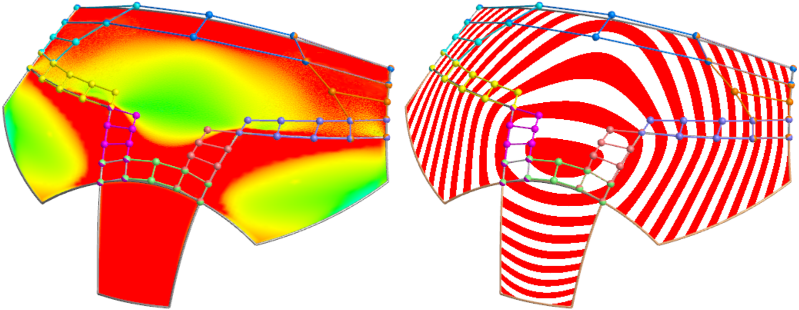
Editing with the central control point (contours)



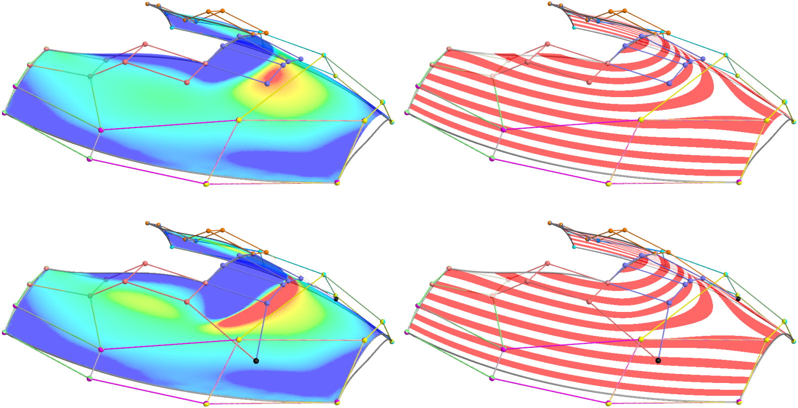
Network of patches (control networks)



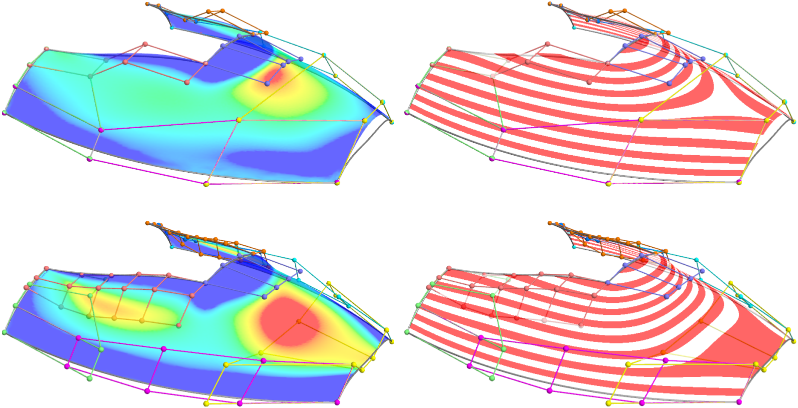
Network of patches (mean curvature, contours)



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.