I-patch – An implicit representation for multi-sided surfaces

Ágoston Sipos, Tamás Várady, **Péter Salvi**, Márton Vaitkus

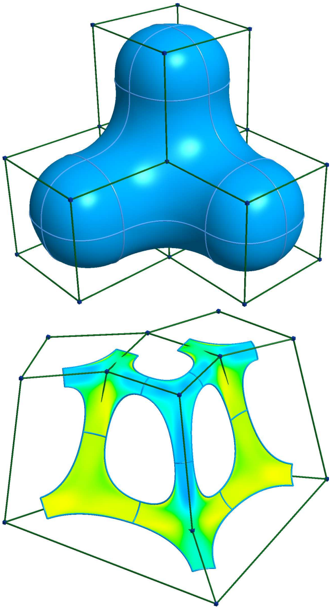
Budapest University of Technology and Economics

Dagstuhl Seminar on Geometric Modeling

November 21–26, 2021

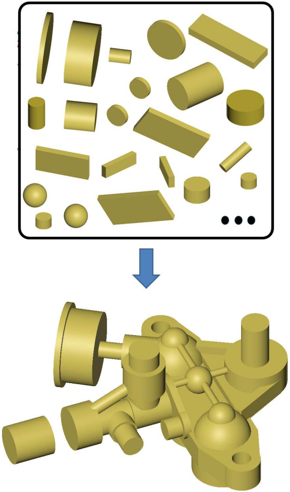
Outline

- Motivation
- I-patch
  - Patch equation
  - Reformulation
  - Ribbons & bounding surfaces
- Discussion
  - Shape parameters
  - Normalization
- Applications
  - Hole filling & vertex blends
  - Polyhedral design
  - Mesh approximation
- Conclusion



Multi-sided *free-form* patches with implicit surfaces

- Why implicit surfaces?
  - Arbitrary number of sides
  - No parameterization
  - Connection to regular surfaces
  - Intersection is easy
  - Efficient ray casting
  - Efficient point membership
- For free-form shapes, as well!
- Recently used for...
  - Fitting with neural networks
  - Topology optimization
  - Sketch-based modeling etc.
- Possible drawbacks
  - Hard to tessellate
  - Rigid
  - Shape problems
  - Geometrically meaningful control?



Based on Wu et al. (2018)

Outline

Liming curves and functional splines

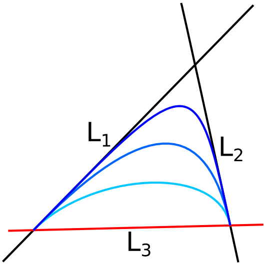
- Liming-formula (1947)
  - Conic sections
  - Based on implicit lines
  - Geometric meaning
  - Fullness parameter (λ)

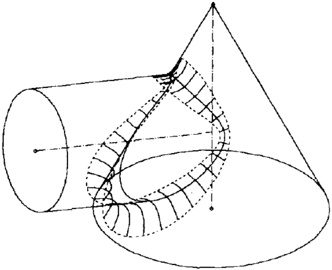
    $$C : (1-\lambda)L_1L_2 - \lambda L_3^2 = 0$$

- Generalization by Li et al. (1990)
  - "Functional splines"
  - Base surface (f)
  - Transversal surface (g)

    $$S : (1-\lambda)f - \lambda g^{k+1} = 0$$

- Good for G^k blends
- Not really multi-sided!



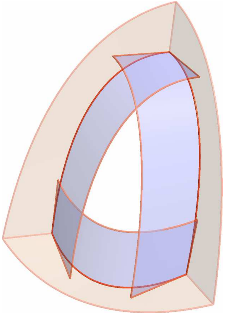


Li, Hoschek & Hartmann (1990)

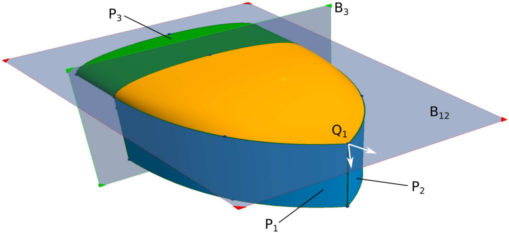
I-patch

- Multi-sided generalization
  - Várady et al. (2001)
  - Primary ribbons (P_i)
  - Bounding surfaces (B_i)
  - Weights (w_i, w)

$$I : \sum_{i=1}^{n} w_i P_i \prod_{\substack{j=1 \\ j \neq i}}^{n} B_j^{k+1} - w \prod_{j=1}^{n} B_j^{k+1} = 0$$



- Can handle singular vertices



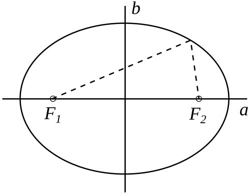
Distance-based formula

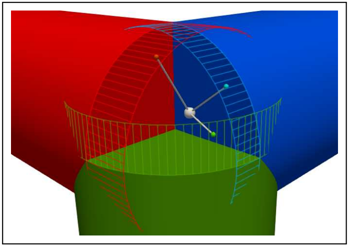
$$I : \sum_{i=1}^{n} w_i P_i \prod_{\substack{j=1 \\ j \neq i}}^{n} B_j^{k+1} - w \prod_{j=1}^{n} B_j^{k+1} = 0$$

- Dividing by $\prod_{j=1}^{n} B_j^{k+1}$:

$$\hat{I} : \sum_{i=1}^{n} w_i \frac{P_i}{B_i^{k+1}} = \sum_{i=1}^{n} d_i = w$$

- Similar to the ellipse in logic ($E : d_1 + d_2 = 2a$)

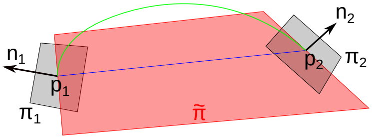




- Reproduces the ellipsoid (P_i elliptic cylinders, B_i planes)

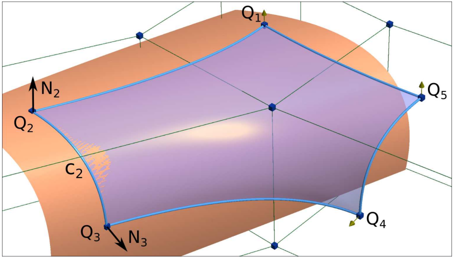
Constructing ribbons & bounding surfaces

- Hierarchy based on endpoints & end-normals
  - Plane + Plane
    ⇒ Straight
  - Liming-surface + Plane
    ⇒ Conic
  - I-loft + Plane
    ⇒ I-segment (2D I-patch)
  - General

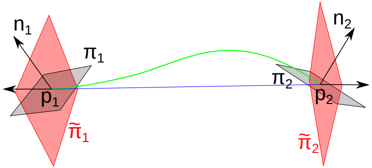


Liming-surface

(with planes instead of lines)



Curved ribbon & planar bounding

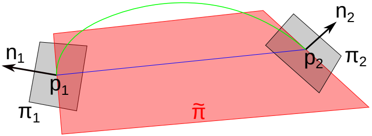


I-loft

(2-sided I-patch using planes)

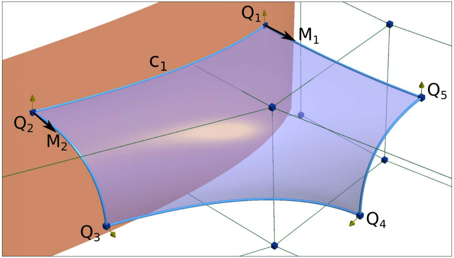
Constructing ribbons & bounding surfaces

- Hierarchy based on endpoints & end-normals
  - Plane + Plane
    $\Rightarrow$ Straight
  - Liming-surface + Plane
    $\Rightarrow$ Conic
  - I-loft + Plane
    $\Rightarrow$ I-segment (2D I-patch)
  - General

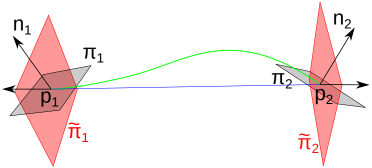


Liming-surface

(with planes instead of lines)



Planar ribbon & curved bounding

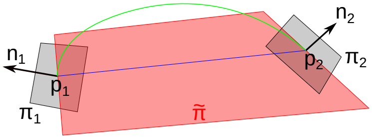


I-loft

(2-sided I-patch using planes)

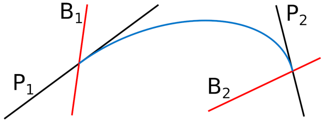
Constructing ribbons & bounding surfaces

- Hierarchy based on endpoints & end-normals
  - Plane + Plane
    ⇒ Straight
  - Liming-surface + Plane
    ⇒ Conic
  - I-loft + Plane
    ⇒ I-segment (2D I-patch)
  - General

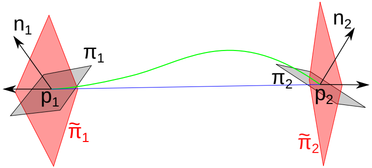


Liming-surface

(with planes instead of lines)



I-segment



I-loft

(2-sided I-patch using planes)

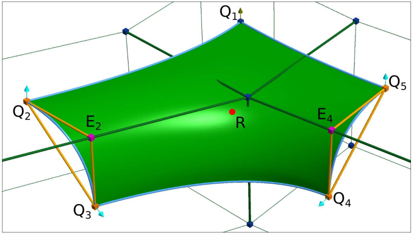
Outline

Shape parameters

- Various weights:
  - Liming-surfaces: λ_i
  - I-lofts: $w_1^{(i)}$, $w_2^{(i)}$, $w^{(i)}$
  - I-patch itself: w_i, w
- Application-dependent
- Ribbon/bounding parameters often fixed
- Default weights normalize the distance fields at a "central" reference point $\mathbf{R}$:

$$d_i(\mathbf{R}) = w_i \frac{P_i(\mathbf{R})}{B_i(\mathbf{R})^{k+1}} = \pm 1$$

- Use w to interpolate $\mathbf{R}$



- Alternative:
  - Minimize fairness energy
  - Mesh-based

Distance field

- Good SDF needed for offset, intersection, approximation etc.
- Not Euclidean (even with a good scalar multiplier)
- Simple approximation:
  Division by gradient norm ($\hat{f} = f/\|\nabla f\|$)
- Reinterpretation of I-patches as weighted ribbons:

$$I : \sum_{i=1}^{n} P_i \cdot \alpha_i = w, \qquad \alpha_i = \frac{w_i}{B_i^{k+1}}$$

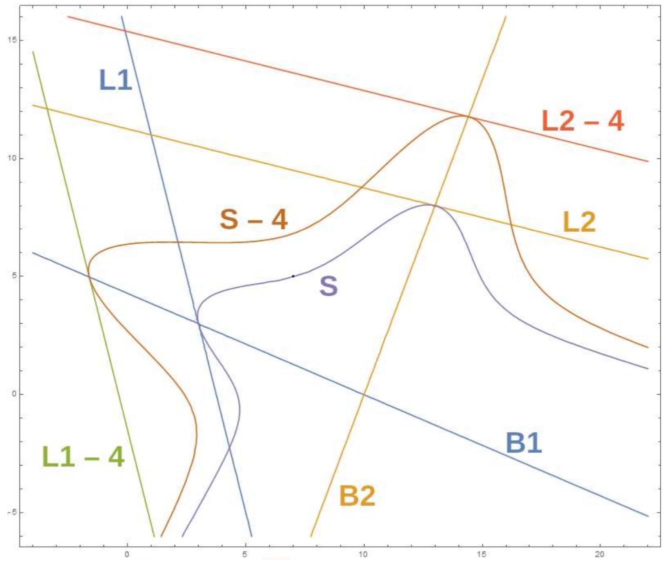
- Normalize by dividing with the sum of blends:

$$\hat{I} : \frac{\sum_{i=1}^{n} P_i \cdot \alpha_i - w}{\sum_{i=1}^{n} \alpha_i} = 0$$

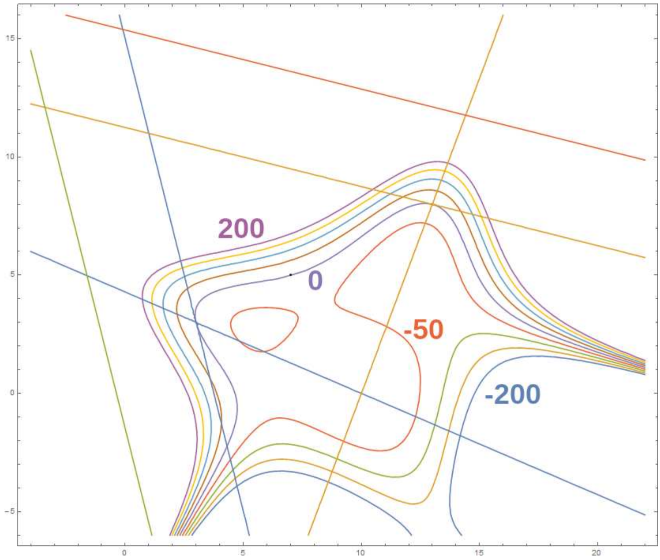
- Algebraic offset is also a normalized I-patch:

$$\hat{I} - d = \frac{\sum_{i=1}^{n} (P_i - d) \cdot \alpha_i - w}{\sum_{i=1}^{n} \alpha_i}$$

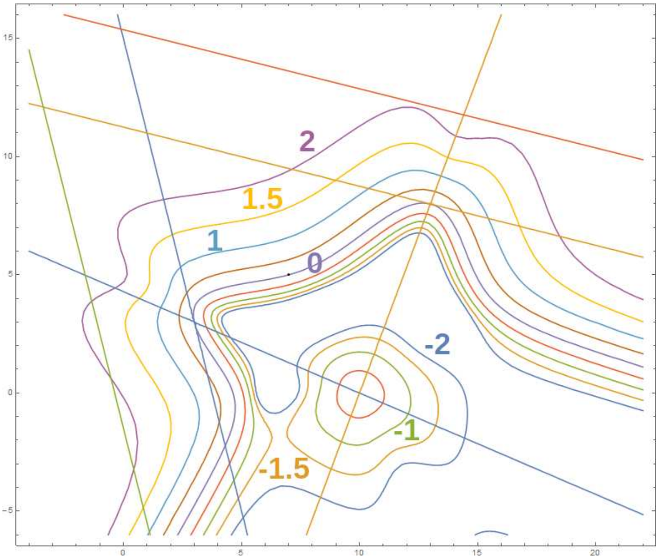
2D Example – Offset with "faithful" normalization



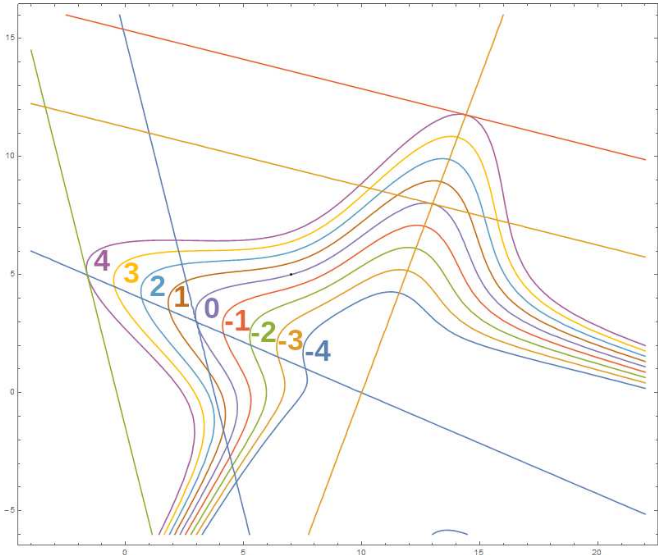
2D Example – No normalization



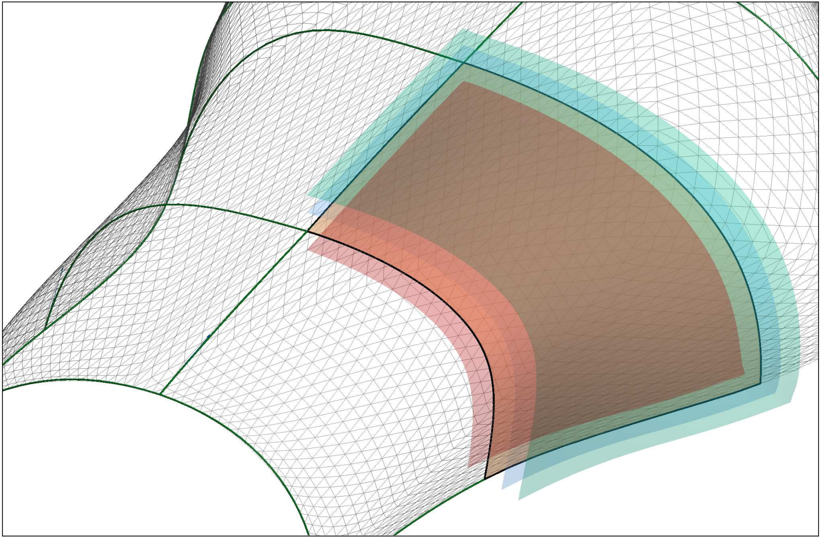
2D Example – Gradient normalization



2D Example – "Faithful" normalization

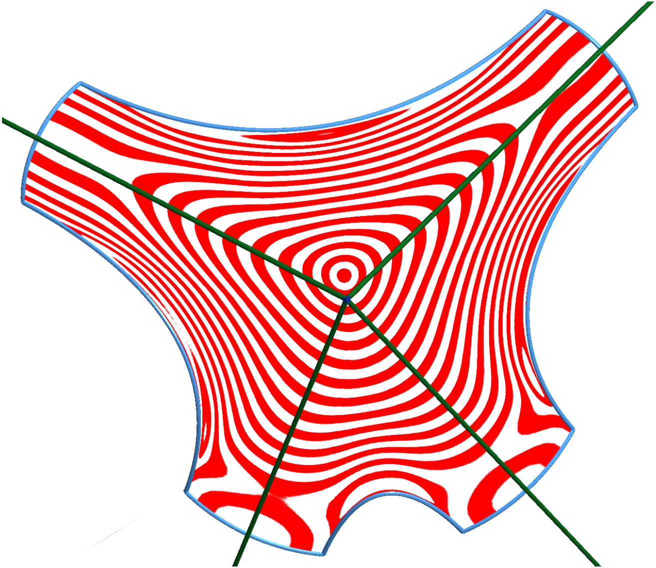


3D Example – Offset surfaces

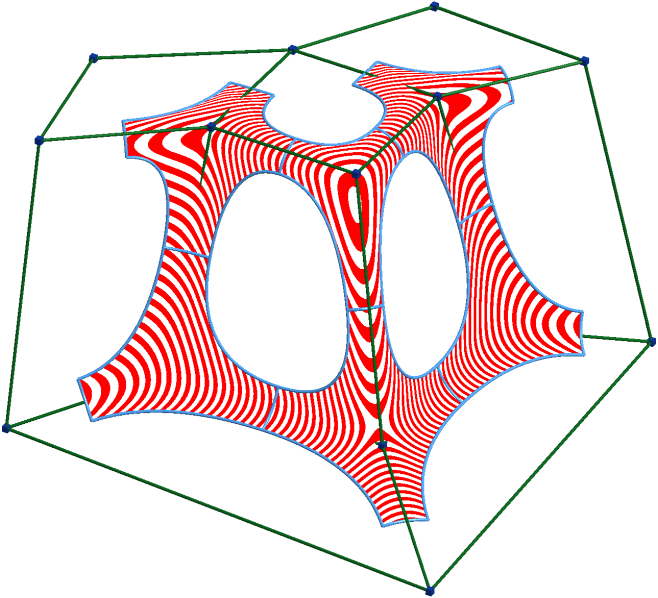


Outline

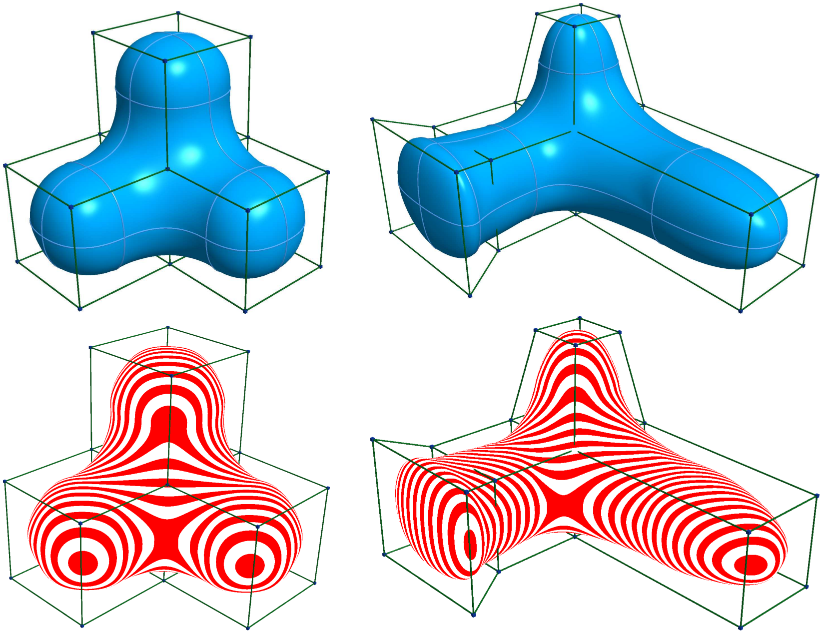
Setback vertex blends – 8-sided with isophotes



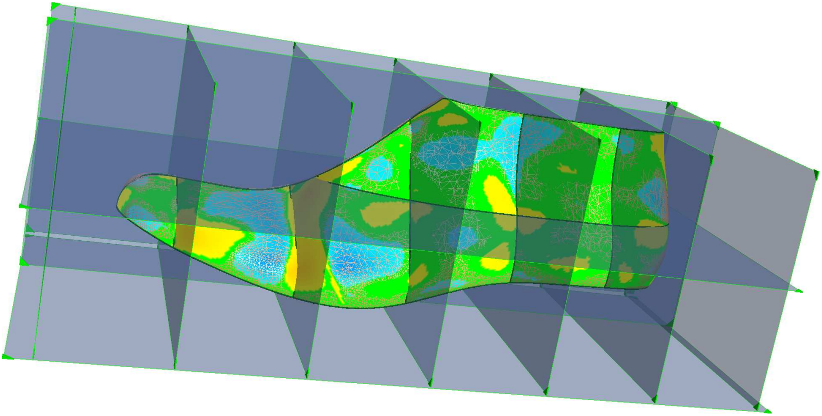
Setback vertex blends – Six patches with contours



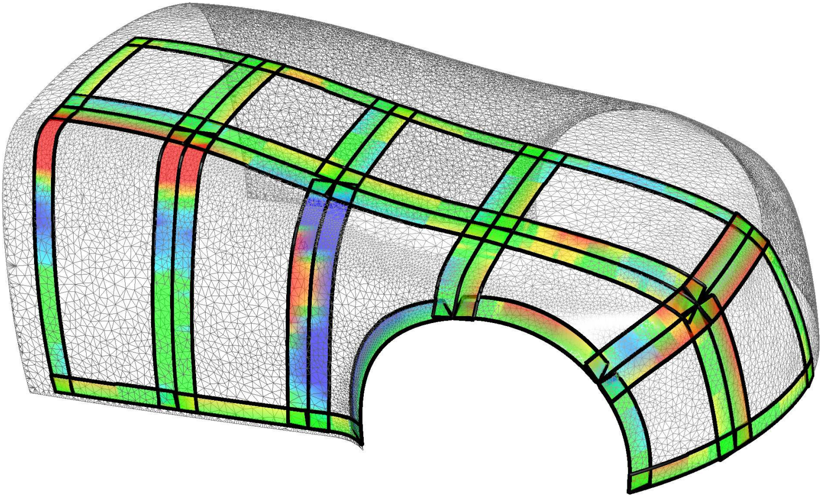
Polyhedral design



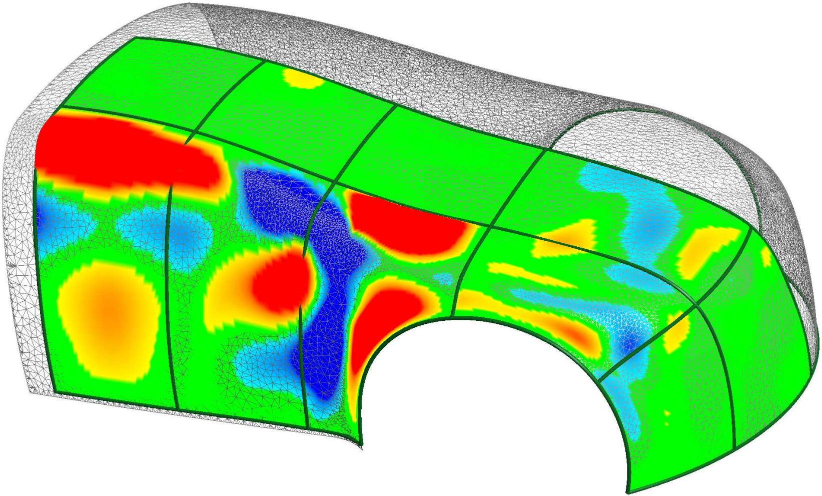
Mesh approximation – Cell-based subdivision



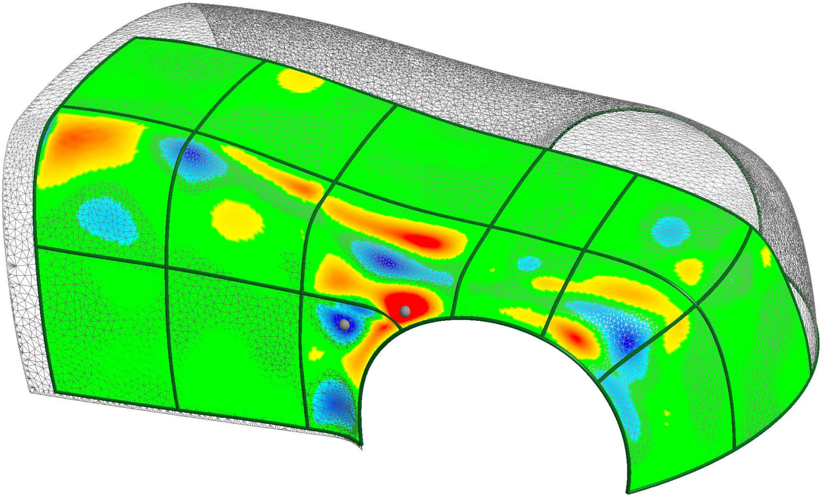
Mesh approximation – Adaptive refinement (with T-nodes)



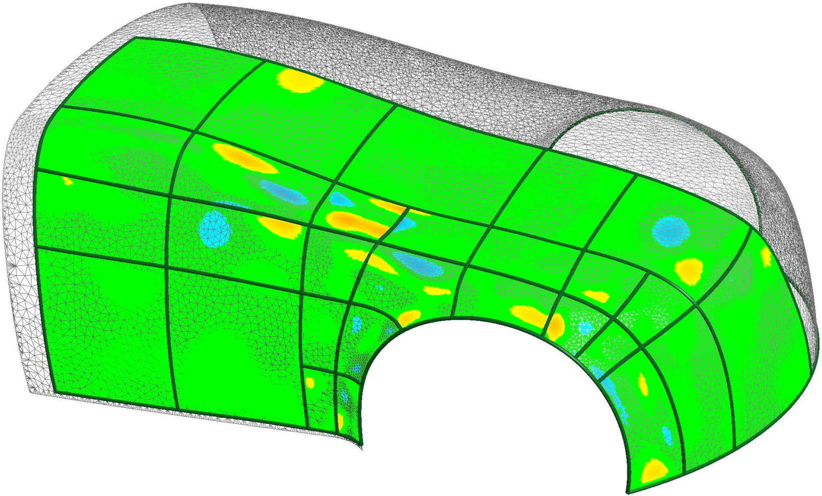
Mesh approximation – Adaptive refinement (with T-nodes)



Mesh approximation – Adaptive refinement (with T-nodes)



Mesh approximation – Adaptive refinement (with T-nodes)



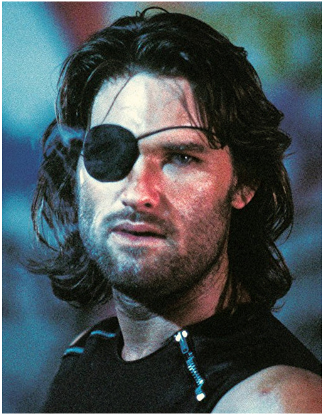
Outline

Conclusion

- A few steps towards “usable” implicit, multi-sided patches
- Geometric reinterpretation
- Ribbon & bounding surface constructions
- Faithful normalization
- Applications
  - Vertex blends
  - Polyhedral modeling
  - Mesh approximation
- Tessellation & handling common shape problems
- Future work
  - More geometric meaning to the shape parameters
  - Increase robustness
  - Implementation on the GPU

Any questions?

Thank you
for your attention.



https://github.com/agostonsipos/I-patch/

Budapest University of Technology and Economics

https://3dgeo.iit.bme.hu/