Discrete Curvature Computation

Work Report

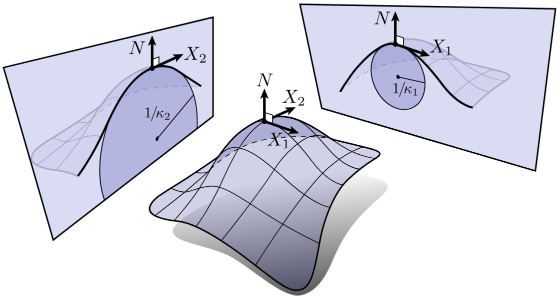
MinliangLIN

November 9, 2018

Outline

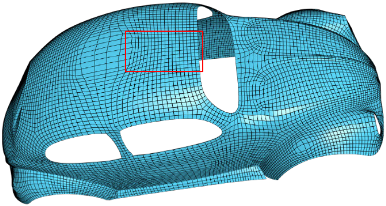
What is curvature?

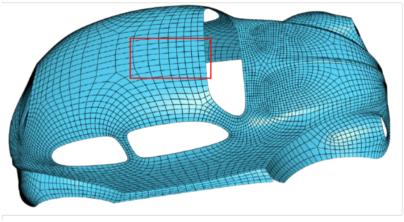
- A number indicating how blend a curve is, which lays on surface
- How fast does a curve leave the tagent plane?



Motivation

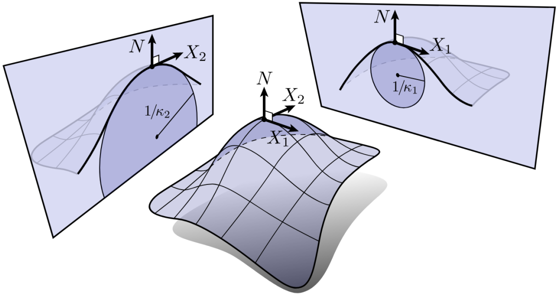
controlling the direction and size of quadrangulation





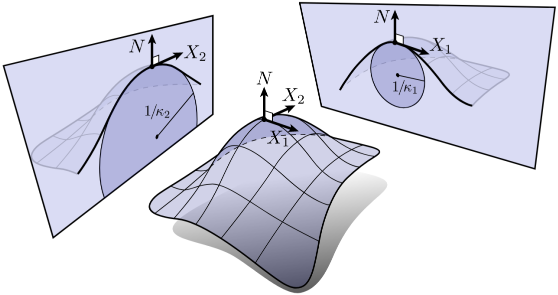
What is curvature

- A quadratic form of tagent vector $X^T B X$



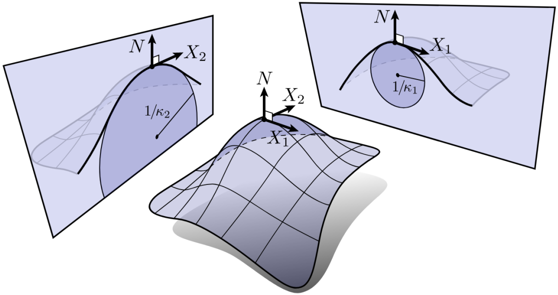
What is curvature

- A quadratic form of tagent vector $X^T B X$
- B is a **linear map / tensor** from dX to dN



What is curvature

- A quadratic form of tagent vector $X^T B X$
- B is a **linear map / tensor** from dX to dN
- mean curvature $\kappa_H = tr(B)/2$

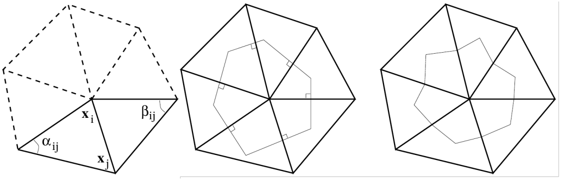


How to compute **mean curvature** on vertex?

- multiply by normal vector $\mathbf{K}(\mathbf{x}) = \kappa_H \mathbf{n}$
- $\int \int_{\mathcal{A}_M} \mathbf{K}(\mathbf{x}) dA = \frac{1}{2} \sum_{j \in N_1(i)} (cot\alpha_{ij} + cot\beta_{ij})(\mathbf{x}_i - \mathbf{x}_j)$

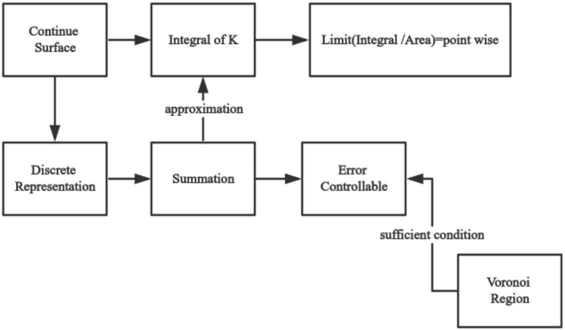
How to compute **mean curvature** on vertex?

- multiply by normal vector $\mathbf{K}(\mathbf{x}) = \kappa_H \mathbf{n}$
- $\int \int_{\mathcal{A}_M} \mathbf{K}(\mathbf{x}) dA = \frac{1}{2} \sum_{j \in N_1(i)} (cot\alpha_{ij} + cot\beta_{ij})(\mathbf{x}_i - \mathbf{x}_j)$



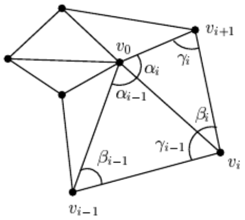
How to compute **mean curvature** on vertex?

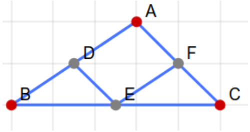
- multiply by normal vector $\mathbf{K}(\mathbf{x}) = \kappa_H \mathbf{n}$
- $\int\int_{\mathcal{A}_M} \mathbf{K}(\mathbf{x}) dA = \frac{1}{2} \sum_{j \in N_1(i)} (cot\alpha_{ij} + cot\beta_{ij})(\mathbf{x}_i - \mathbf{x}_j)$



How to compute **mean curvature** on vertex?

- multiply by normal vector $\mathbf{K}(\mathbf{x}) = \kappa_H \mathbf{n}$
- $\int\int_{\mathcal{A}_M} \mathbf{K}(\mathbf{x}) dA = \frac{1}{2}\sum_{j \in N_1(i)} (cot\alpha_{ij} + cot\beta_{ij})(\mathbf{x}_i - \mathbf{x}_j)$





How to get the tensor/matrix of vertex?

- $B = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$
- sampling along different direction
- least square fitting the data constrain to mean curvature

How to get the tensor/matrix of vertex?

- $B = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$
- sampling along different direction
- least square fitting the data constrain to mean curvature
- $E(a,b,c) = \sum_j w_j(\mathbf{d}_{i,j}^T B \mathbf{d}_{i,j} - \kappa_{i,j}^N)^2 \ s.t. \ a + c = 2\kappa_H, \cancel{ac - b^2 = \kappa_G}$

Problems

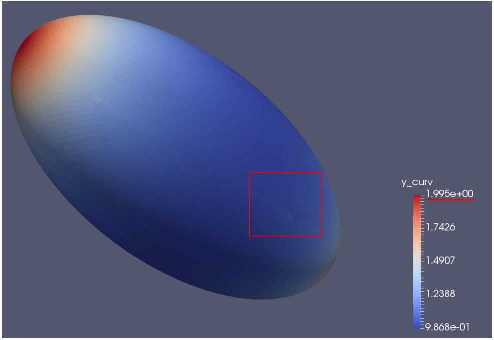
- Some pontential trouble makers
  1. Hard constrain of κ_H is replaced by soft constrain.
  2. w_js are not used
  3. Area of computating κ_H is just barycenter area, and cotangents of obtuse angles are not handled, which may cause some negative weight of summation of κ_H.
  4. [MDSB03] states that "the curvature values computed from the lesast squares are often less accurate in practice".
  5. We compute curvature of facet as the average of vertex with uniform weight

Conclusion

result is bad when tessellation is irregular

Conclusion

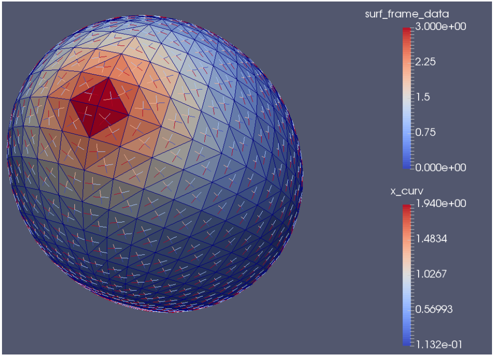
result is bad when tessellation is irregular



Conclusion

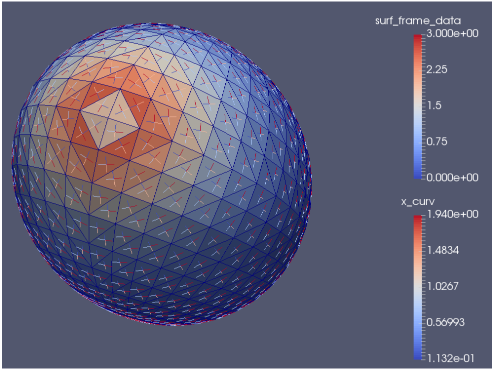
result is bad when tessellation is irregular

- reflection



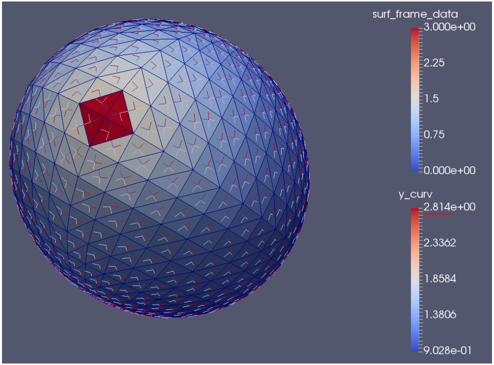
Conclusion

result is bad when tessellation is irregular



Conclusion

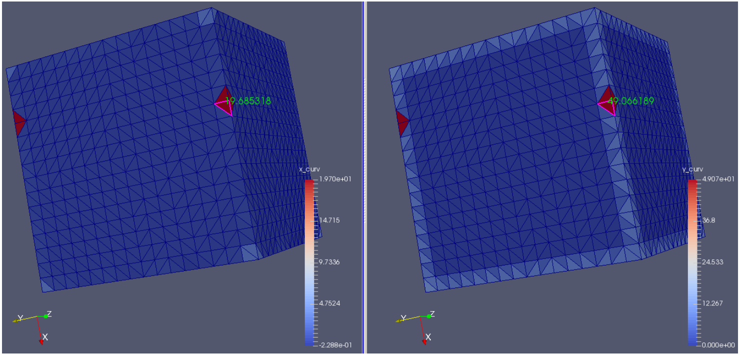
result is bad when tessellation is irregular



Conclusion

result is bad when tessellation is irregular

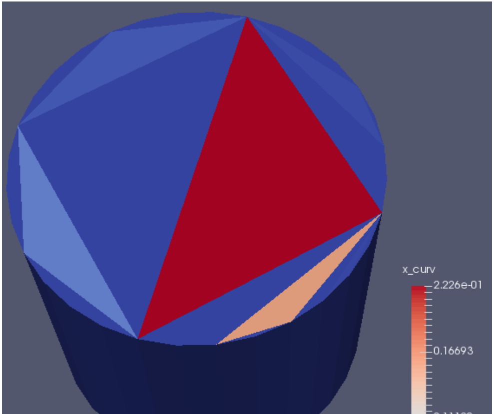
- burst



Conclusion

result is bad when tessellation is irregular

- spreading



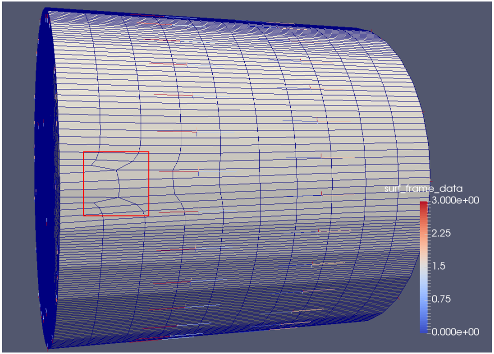
Ask more questions

- Pipeline:
  $weight1 * principal\_curvature + weight2 * feature\_line \rightarrow constrain \rightarrow metric \rightarrow frame\_field \rightarrow quad$
- How to reduce weight of bad *principal_curvature*?
  1. conflict with feature
  2. "flat/sphere like" region
     current solution: $weight = min(|e^{|pc_1 - pc_2|} - 1|, 100)$, still fail on the previous big triangle.
  3. other solution: subdivide or other way to change the bad tessellation.

Ask more questions

- Pipeline:
  $weight1 * principal\_curvature + weight2 * feature\_line \rightarrow constrain \rightarrow metric \rightarrow frame\_field \rightarrow quad$
- How to reduce weight of bad *principal_curvature*?
  1. conflict with feature
  2. "flat/sphere like" region
     current solution: $weight = min(|e^{|pc_1 - pc_2|} - 1|, 100)$, still fail on the previous big triangle.
  3. other solution: subdivide or other way to change the bad tessellation.
- what is the difference between *metric* and *frame_field*?

Other burst



Reference

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr, *Discrete differential-geometry operators for triangulated 2-manifolds*, Visualization and mathematics III, Springer, 2003, pp. 35–57.