Differential Geometry:

Conformal Metrics

[*Conformal Flattening by Curvature Prescription and Metric Scaling*. Ben-Chen *et al*., 2008]
[*Conformal Equivalence of Triangle Meshes*. Springborn *et al*., 2009]

Conformal Maps

Recall:

A conformal map is an angle-preserving transformation.

Liouville's Theorem:

In dimensions greater than 2, the *Möbius transformations* (translations, rotations, scales, and inversions) are the only conformal maps.

Möbius Transformations

Given points x,y$\in \boldsymbol{R}^n$ and given a Möbius transformation ϕ, we have:

1. If ϕ is a translation:

$$\left\|\phi(x)-\phi(y)\right\|=\left\|x+c-y-c\right\|=\left\|x-y\right\|$$

Möbius Transformations

Given points x,y$\in \boldsymbol{R}^n$ and given a Möbius transformation ϕ, we have:

1. If ϕ is a translation:
$$\|\phi(x)-\phi(y)\| = \|x+c-y-c\| = \|x-y\|$$
2. If ϕ is a rotation:
$$\|\phi(x)-\phi(y)\| = \|\phi(x-y)\| = \|x-y\|$$

Möbius Transformations

Given points x,y$\in \boldsymbol{R}^n$ and given a Möbius transformation ϕ, we have:

1. If ϕ is a translation:
$$\|\phi(x)-\phi(y)\| = \|x+c-y-c\| = \|x-y\|$$
2. If ϕ is a rotation:
$$\|\phi(x)-\phi(y)\| = \|\phi(x-y)\| = \|x-y\|$$
3. If ϕ is a scale:
$$\|\phi(x)-\phi(y)\| = \|sx-sy\| = s\|x-y\|$$

Möbius Transformations

Given points x,y$\in \boldsymbol{R}^n$ and given a Möbius transformation ϕ, we have:

1. If ϕ is a translation:
$$\|\phi(x)-\phi(y)\| = \|x+c-y-c\| = \|x-y\|$$
2. If ϕ is a rotation:
$$\|\phi(x)-\phi(y)\| = \|\phi(x-y)\| = \|x-y\|$$
3. If ϕ is a scale:
$$\|\phi(x)-\phi(y)\| = \|sx-sy\| = s\|x-y\|$$
4. If ϕ is an inversion:
$$\|\phi(x)-\phi(y)\| = \left\|\frac{x}{\|x\|^2}-\frac{y}{\|y\|^2}\right\| = \frac{1}{\|x\|\|y\|}\|x-y\|$$

Möbius Transformations

Thus, given points $x,y \in \boldsymbol{R}^n$ and given a Möbius transformation ϕ, we can express the distance between the transformed points as the original distance, modulated by the product of functions that only depends on the individual points' positions.

$$\left\|\phi(x)-\phi(y)\right\| = \rho(x)\rho(y)\left\|x-y\right\|$$

Gauss-Bonnet

Recall:

Given a triangle mesh M with boundary ∂M, the total curvature is:

$$\sum_{v\in M,v\notin\partial M} K_v + \sum_{v\in\partial M} \kappa_v = 2\pi\chi(M)$$

with K_v the Gaussian curvature at interior v:

$$K_v = 2\pi - \sum_{(v_1,v,v_2)\in T} \angle v_1 v v_2$$

with κ_v the Gaussian curvature at boundary v:

$$\kappa_v = \pi - \sum_{(v_1,v,v_2)\in T} \angle v_1 v v_2$$

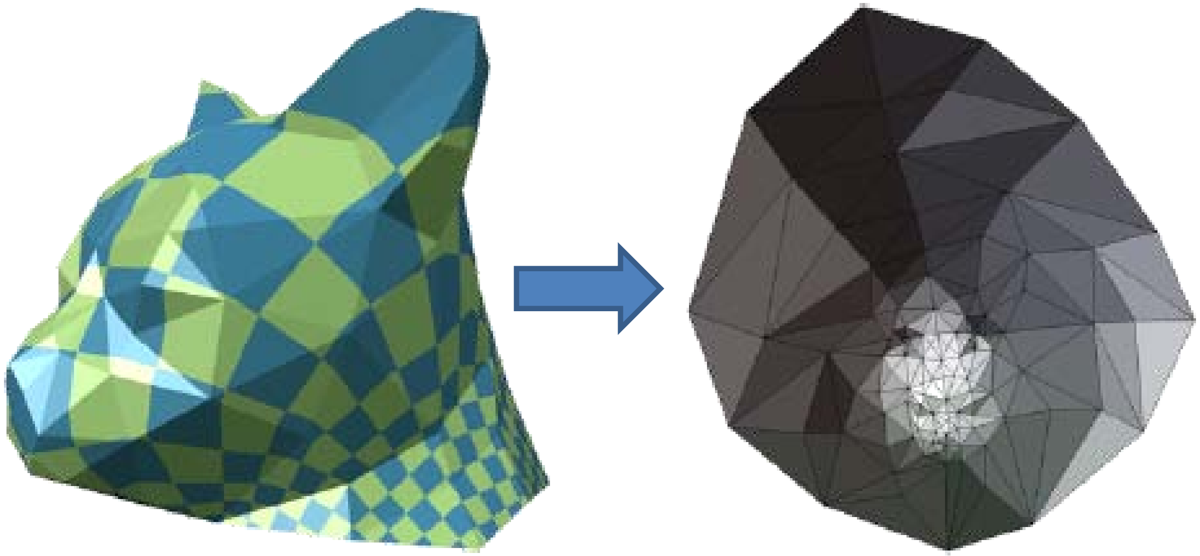
and $\chi(M)$ the Euler characteristic:

$$\chi(M) = |T| - |E| + |V|$$

Conformal Flattening

Goal:

Given a triangle mesh *M*, we would like to find a conformal mapping of *M* into the plane.



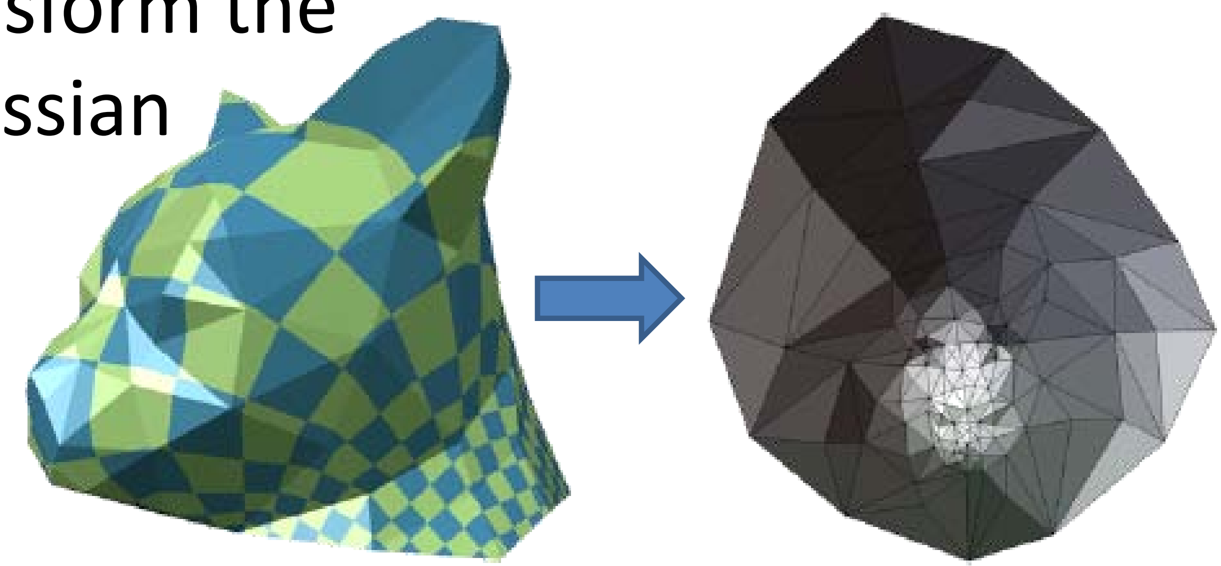
Conformal Flattening

Goal:

Given a triangle mesh *M*, we would like to find a conformal mapping of *M* into the plane.

Approach:

Conformally transform the mesh so the Gaussian curvature at all interior vertices is equal to 0.



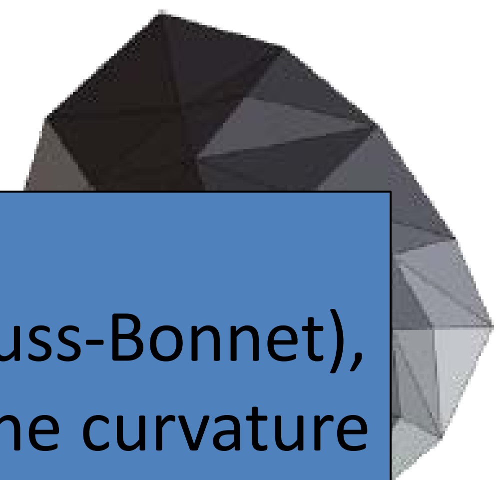
Conformal Flattening

Goal:

Given a triangle mesh *M*, we would like to find a conformal mapping of *M* into the plane.

Approach:

Conformally transform the mesh so the Gaussian
cu
in
is



Note:

Since the total curvature is constant (Gauss-Bonnet), this means that we will have to "push" the curvature to the boundaries.

Conformal Flattening

Question:

How do we “conformally transform” a mesh?

Conformal Flattening

<u>Question</u>:

How do we “conformally transform” a mesh?

<u>Answer</u>:

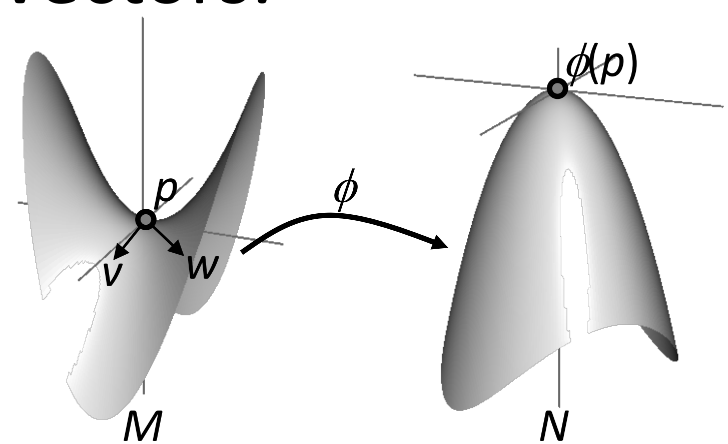
Recall that a conformal map ϕ is a map that locally looks like an isotropic scaling.

Conformal Flattening

Continuous Setting:

Given a map ϕ from a surface *M* to a surface *N*, we can define a metric on *M* in two ways:

1. We can set the length of vectors in the tangent space of *p* to be the embedded length of the embedded vectors.

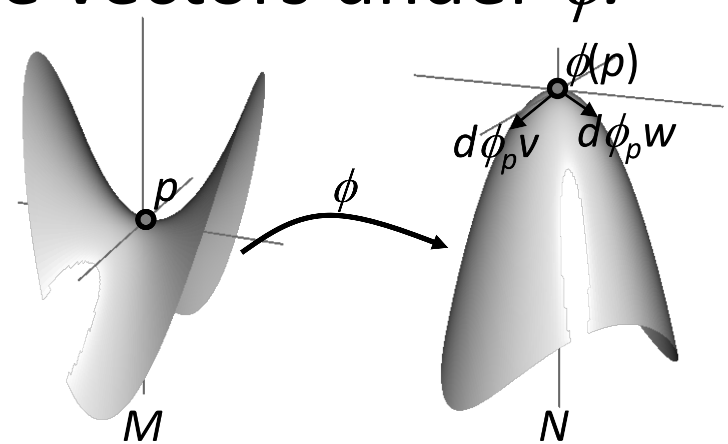


Conformal Flattening

<u>Continuous Setting</u>:

Given a map ϕ from a surface M to a surface N, we can define a metric on M in two ways:

2. We can set the length of vectors in the tangent space of p to be the embedded length of the image of the vectors under ϕ.



Conformal Flattening

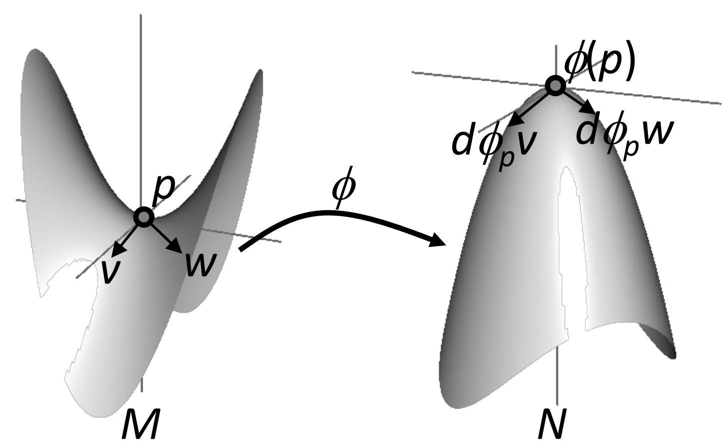
Continuous Setting:

Given a map ϕ from a surface *M* to a surface *N*, we can define a metric on *M* in two ways.

The two surfaces are *conformally equivalent* if at every point *p*, the two metrics differ by a (positive) scale factor:

$$g_N = e^{2u} g_M$$

for some smooth function $u: M \to \mathbf{R}$.



Conformal Flattening

<u>Discrete Setting</u>:

Given a triangle mesh *M*, a *discrete metric* on *M* is a function $l\!:\!E\rightarrow\boldsymbol{R}^{>0}$ (an association of lengths to edges) satisfying the triangle inequality:

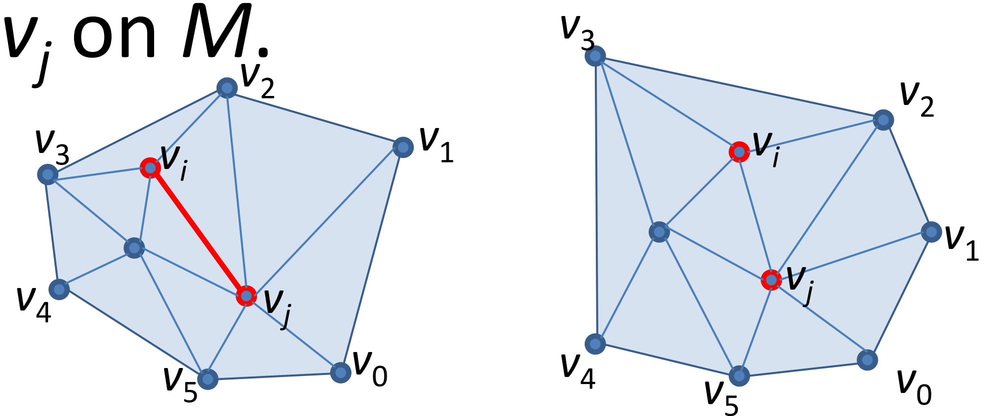
$$l_{v_1v_2}+l_{v_2v_3}\geq l_{v_3v_1}\qquad\forall(v_1,v_2,v_3)\in T$$

Conformal Flattening

Discrete Setting:

Given two triangle meshes M and N with the same topology, we can define a discrete metric on M in two ways:

1. For vertices v_i and v_j lying on an edge, we can set l_{ij} to be the distance between the positions of v_i and v_j on M.

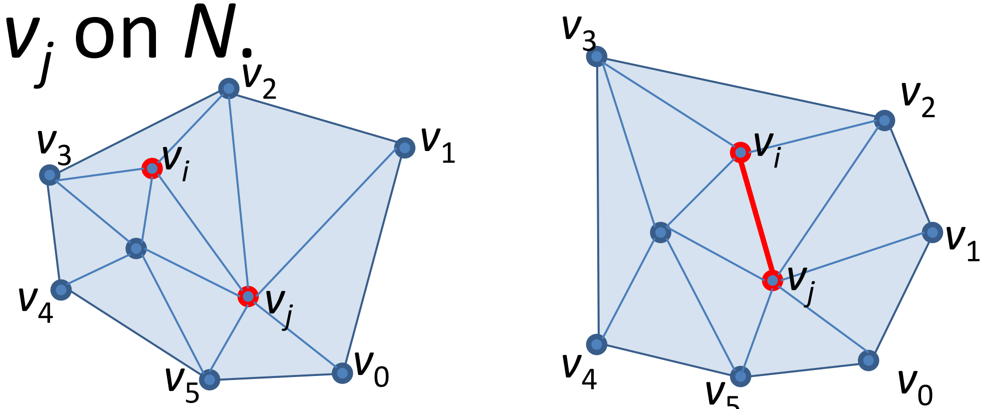


Conformal Flattening

<u>Discrete Setting</u>:

Given two triangle meshes *M* and *N* with the same topology, we can define a discrete metric on *M* in two ways:

2. For vertices v_i and v_j lying on an edge, we can set l_{ij} to be the distance between the positions of v_i and v_j on *N*.

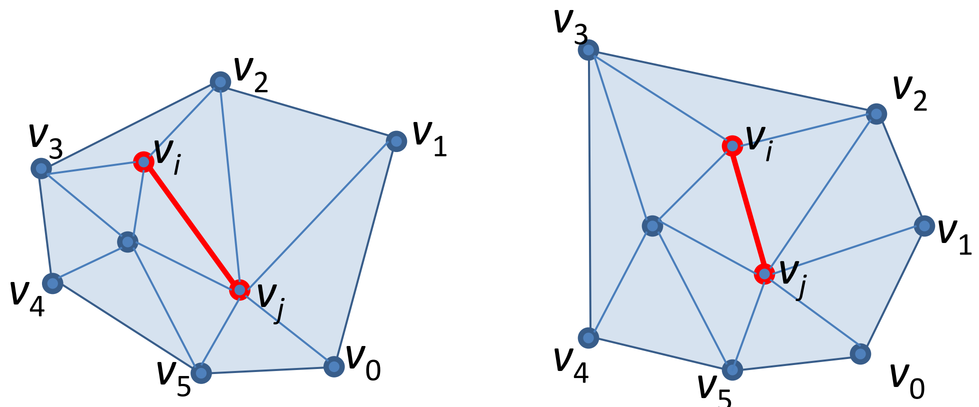


Conformal Flattening

<u>Discrete Setting</u>:

Given two triangle meshes *M* and *N* with the same topology, we can define a discrete metric on *M* in two ways.

The two meshes are *conformally equivalent* if the two discrete metrics differ by a positive scale factor.



Conformal Flattening

<u>Discrete Setting</u>:

Given two triangle meshes *M* and *N* with the same topology, we can define a discrete metric on *M* in two ways.

The two meshes are *conformally equivalent* if the two discrete metrics differ by a positive scale factor.

That is, if there is a function $u{:}V\rightarrow\boldsymbol{R}$ with:

$$l_{ij}^{N}=l_{ij}^{M}e^{(u_i+u_j)/2}$$

Conformal Flattening

Note 1:

If ϕ is a Möbius transformation, we know that we can define a function ρ:$V\rightarrow\boldsymbol{R}$ such that:

$$\left\|\phi(v_i)-\phi(v_j)\right\|=\rho(v_i)\rho(v_j)\left\|v_i-v_j\right\|$$

Conformal Flattening

Note 1:

If ϕ is a Möbius transformation, we know that we can define a function ρ:$V\rightarrow\boldsymbol{R}$ such that:

$$\left\|\phi(v_i)-\phi(v_j)\right\|=\rho(v_i)\rho(v_j)\left\|v_i-v_j\right\|$$

Thus, if we set u_i=2log(ρ(v_i)), we have:

$$l_{ij}^{\phi}=l_{ij}e^{(u_i+u_j)/2}$$

so that the image of *M* under a Möbius transformation is conformally equivalent to *M*.

Conformal Flattening

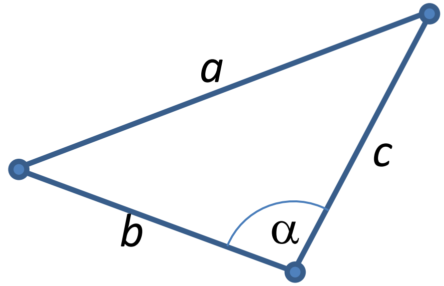
Note 2:

Using a discrete metric, we can talk about conformal maps between surfaces without explicitly establishing an embedding of the mesh into 3D.

Conformal Flattening

<u>Note 3</u>:

If we have a discrete metric, we know the lengths of the edges of all the triangles, which means we can compute the angles.



$$\tan\frac{\alpha}{2} = \sqrt{\frac{(a-b+c)(a+b-c)}{(a+b+c)(-a+b+c)}}$$

Conformal Flattening

Approach:

To find a conformal map sending the mesh *M* into the plane, we want to solve for the scale values u_i that result in the appropriate angles:

1. At vertices *v* interior to the mesh, we want the mesh to be flat:
$$\sum_{(v_1,v,v_2)\in T} \angle v_1 v v_2 = 2\pi$$
2. At boundary vertices *v*, we can prescribe the curvature:
$$\kappa_v = \pi - \sum_{(v_1,v,v_2)\in T} \angle v_1 v v_2$$

Conformal Flattening

Approach:

We do this by defining an energy $E(u)$ which is minimized precisely when the angle-sum constraints are met.

Conformal Flattening

Approach:

We do this by defining an energy $E(u)$ which is minimized precisely when the angle-sum constraints are met.

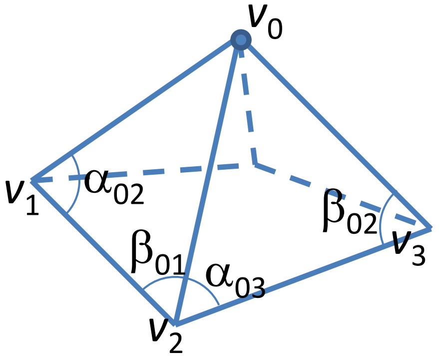
Specifically, if we set Θ_i to be the desired angle-sum at vertex *i*, we find an energy $E(u)$ that has the property:

$$\frac{\partial E}{\partial u_i} = \Theta_i - \sum_{(v_1, v_i, v_2) \in T} \angle v_1 v_i v_2$$

Conformal Flattening

Differentiating the gradient of the energy with respect to the scaling values, we get a Hessian that is the cotangent-weight Laplacian:

$$\frac{\partial^2 E}{\partial u_i u_j} = \begin{cases} \frac{1}{2}\left(\cot\left(\alpha_{ij}\right)p + \cot\left(\beta_{ij}\right)\right) & \text{if } i \neq j \text{ and } v_j \in \mathrm{Nbr}(v_j) \\ -\sum_{v_k \in \mathrm{Nbr}(v_i)} \frac{\partial^2 E}{\partial u_i u_j} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$



Conformal Flattening

Differentiating the gradient of the energy with respect to the scaling values, we get a Hessian that is the cotangent-weight Laplacian:

$$\frac{\partial^2 E}{\partial u_i u_j} = \begin{cases} \frac{1}{2}\left(\cot\left(\alpha_{ij}\right)p + \cot\left(\beta_{ij}\right)\right) & \text{if } i \neq j \text{ and } v_j \in \mathrm{Nbr}(v_j) \\ -\sum_{v_k \in \mathrm{Nbr}(v_i)} \frac{\partial^2 E}{\partial u_i u_j} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Since we know that the cotangent-weight Laplacian is positive (semi) definite, this implies that the energy must have a unique minimum.

Conformal Flattening

Differentiating the gradient of the energy with respect to the scaling values, we get a Hessian that is the cotangent-weight Laplacian:

$$\frac{\partial^2 E}{\partial \cdots} = \begin{cases} \frac{1}{2}\left(\cot(\alpha_{ij})p + \cot(\beta_{ij})\right) & \text{if } i \neq j \text{ and } v_j \in \mathrm{Nbr}(v_j) \\ -\sum \frac{\partial^2 E}{\partial \cdots} & \text{if } i = j \end{cases}$$

Si

La

th

<u>Note</u>:
The cotangent-weight Laplacian has a kernel consisting of constant functions.
This implies that applying a uniform scale to all edge-lengths (i.e. scaling the mesh) does not change the energy.

Conformal Flattening

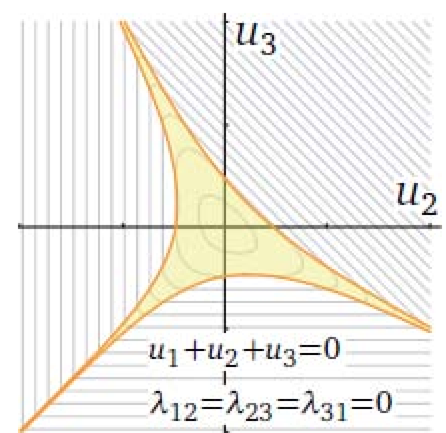
Approach:

1. Using the energy *E* we can evolve the discrete metric so that the angle sums of interior vertices go to zero.
2. Once we have obtained the solution, we can lay-out the surface in the plane.

Conformal Flattening

<u>Challenge</u>:

Not all values u_i give rise to a valid discrete metric. In particular, we could end up with values of u_i such that the edge-lengths of the new mesh do not satisfy the triangle inequality.

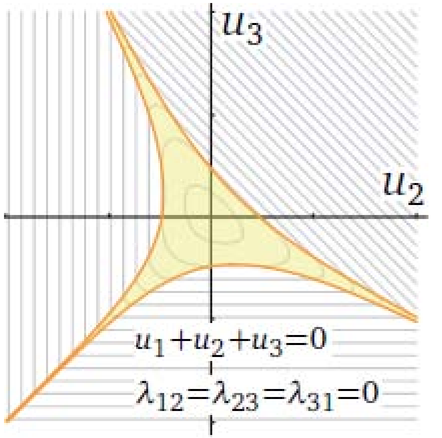


Conformal Flattening

<u>Challenge</u>:

Not all values u_i give rise to a valid discrete metric. In particular, we could end up with values of u_i such that the edge-lengths of the new mesh do not satisfy the triangle inequality.

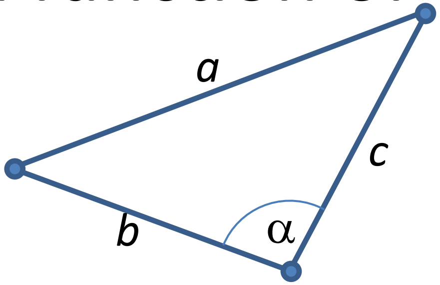
Thus, even though our energy is convex, the domain over which we seek a solution is not.



Conformal Flattening

Challenge:

Specifically, if we consider the equation for the angle as a function of edge-length:



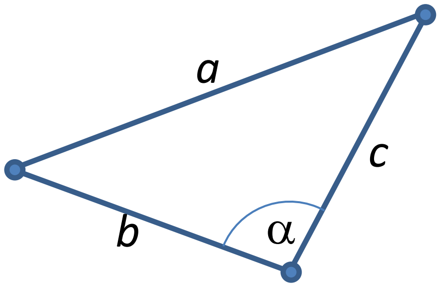
$$\tan\frac{\alpha}{2}=\sqrt{\frac{(a-b+c)(a+b-c)}{(a+b+c)(-a+b+c)}}$$

we see that if $a>b+c$, then we end up taking the squre-root of a negative number and the angle is no longer real-valued.

Conformal Flattening

<u>Challenge</u>:

In the work of Springborn *et al.*, this problem is addressed by modifying the equations, setting α to 2π (and the other two angles to zero) whenever $a>b+c$.

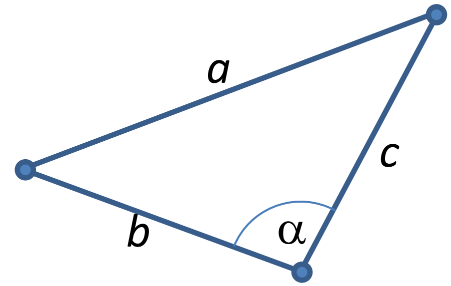


Conformal Flattening

Challenge:

In the work of Springborn *et al.*, this problem is addressed by modifying the equations, setting α to 2π (and the other two angles to zero) whenever $a>b+c$.

They find that unless the triangulation is nearly degenerate, the solution to the optimization will be a valid (planar) triangulation.



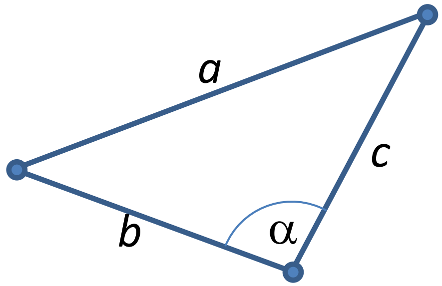
Conformal Flattening

<u>Challenge</u>:

In the work of Springborn *et al.*, this problem is addressed by modifying the equations, setting α to 2π (and the other two angles to zero) whenever $a>b+c$.

They find that unless the triangulation is nearly degenerate, the solution to the optimization will be a valid (planar) triangulation.

When the triangulation is bad, a simple edge-flip fixes the problem.



Specifying Boundary Constraints

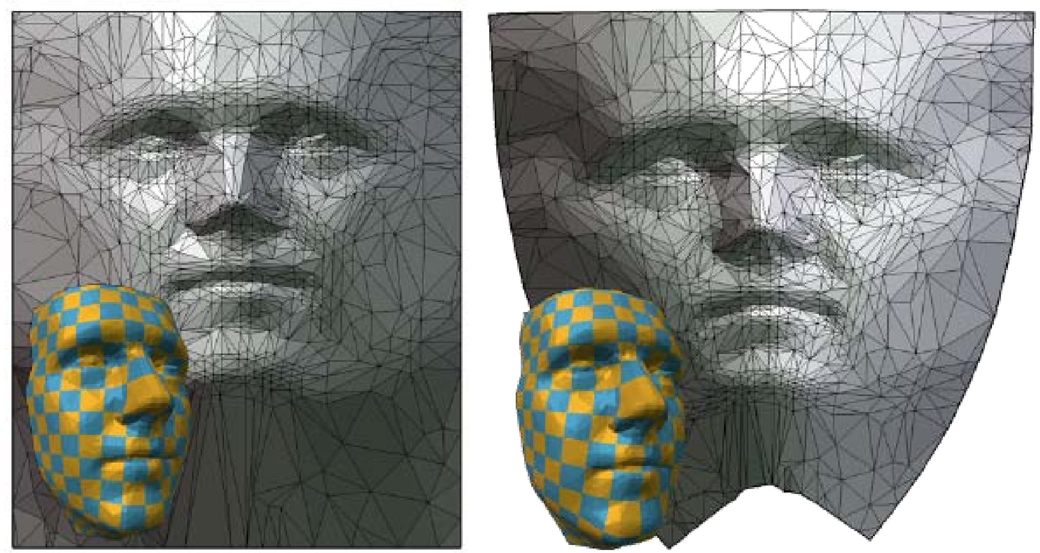
There are two ways to specify the boundary constraints for the conformal map:

1. We can specify the desired angle sum at all boundaries by fixing the values of κ_i.

Specifying Boundary Constraints

There are two ways to specify the boundary constraints for the conformal map:

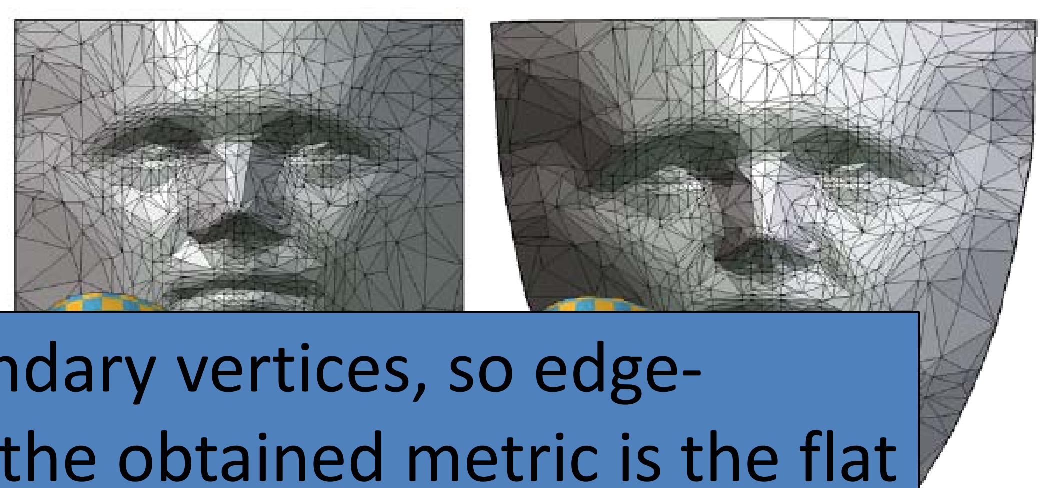
1. We can specify the desired angle sum at all boundaries by fixing the values of κ_i.
2. We can specify the values of the u_i at the boundary and let the κ_i be free.



Specifying Boundary Constraints

There are two ways to specify the boundary constraints for the conformal map:

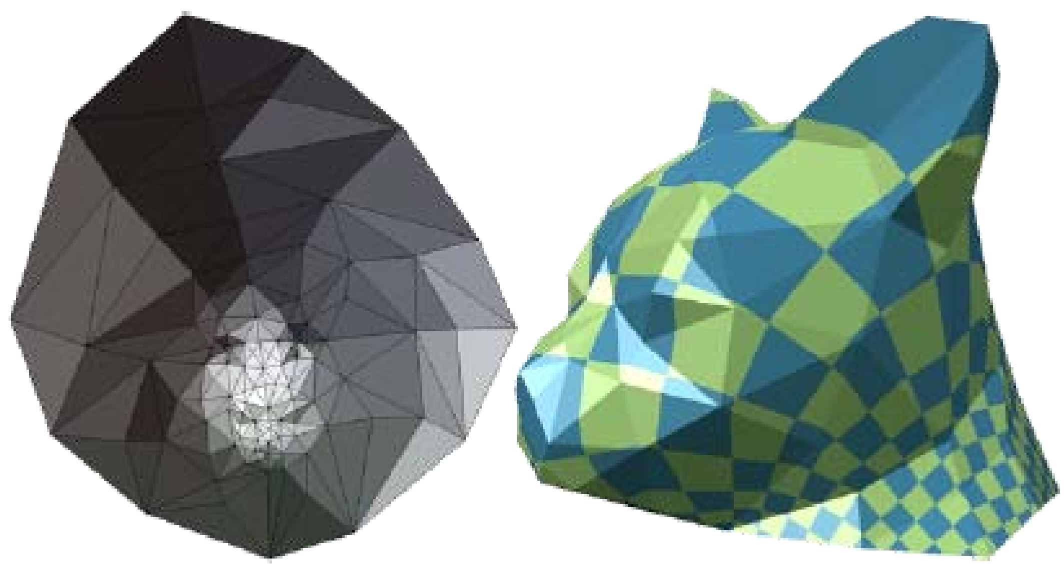
1. We can specify the desired angle sum at all boundaries by fixing the values of κ_i.
2. We can specify the values of the u_i at the boundary and let the κ_i be free.



Setting u_i=0 for all boundary vertices, so edge-lengths are preserved, the obtained metric is the flat conformal metric minimizing (Dirichlet) distortion.

Improving the Mapping

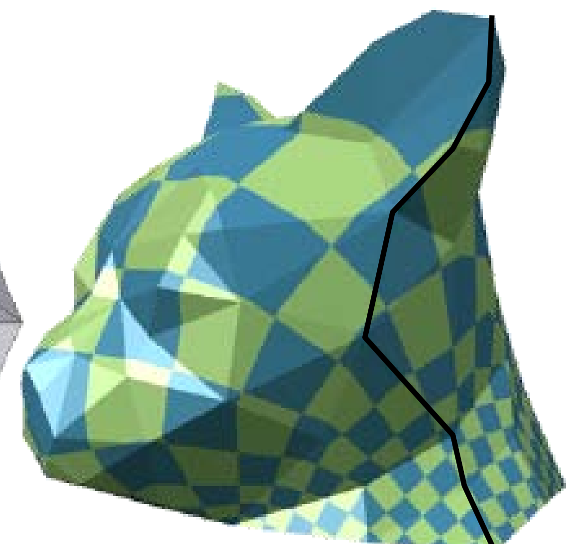
Although we obtain a conformal mapping of the mesh into the plane, there may be a lot of distortion in the interior of the mesh.



Improving the Mapping

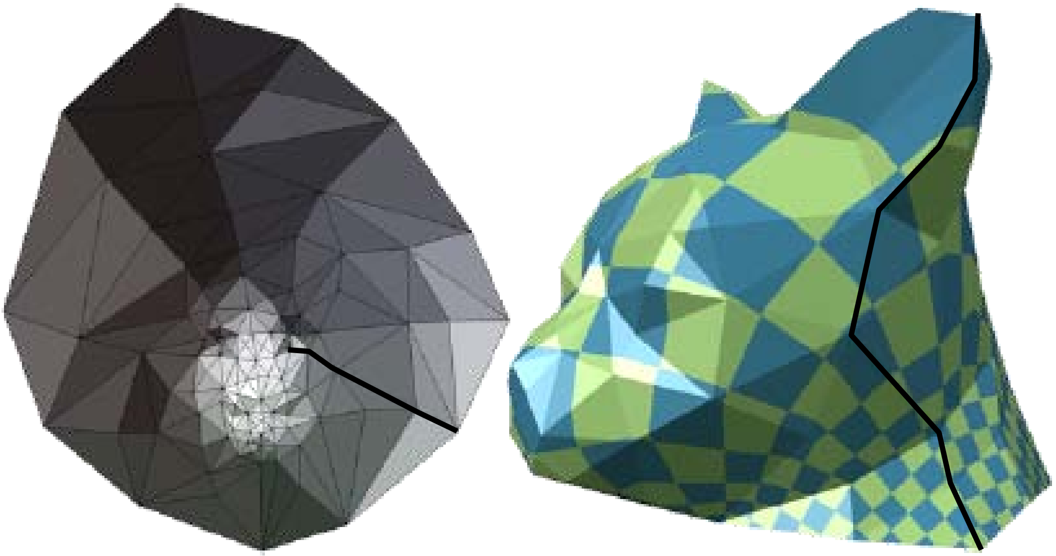
Although we obtain a conformal mapping of the mesh into the plane, there may be a lot of distortion in the interior of the mesh.

We can address this problem by extending the boundary into the mesh so that points with large distortion become part of the boundary.



Improving the Mapping

This allows us to map the surface to the plane in such a way that the angles around the point of distortion sum to a value less than 2π.



Improving the Mapping

The problem is that if we separate the triangles sharing an edge, the same edge will usually be stretched by different amounts, and it will be hard to continuously map texture across the edge.

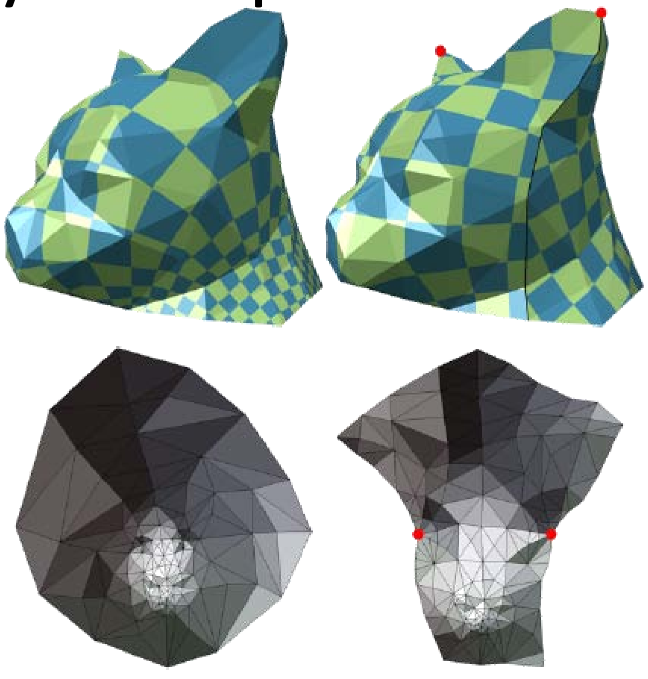
Cone Singularities

Using the metric evolving approach, we can address this problem by specifying that the angle sum around a particular point should be less than 2π.

Cone Singularities

In solving the system, we don't split the edge, so the angle scales remain consistent.

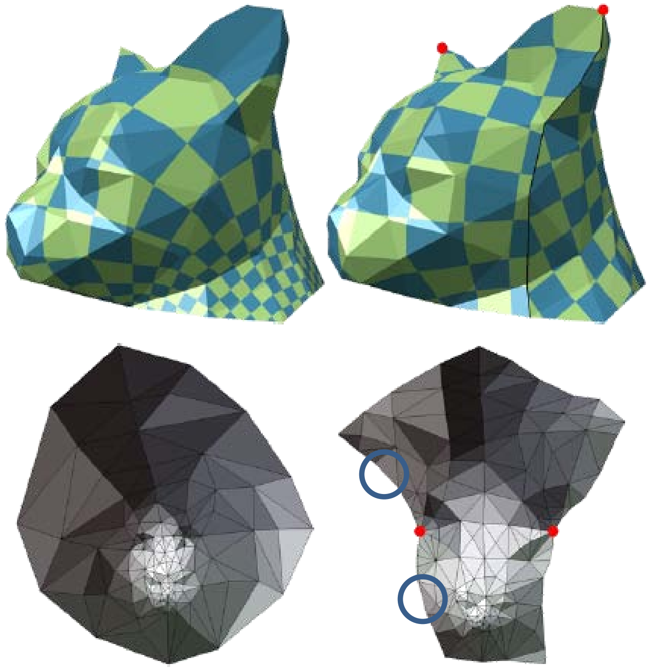
However, in laying-out the mesh, triangles that were originally edge-adjacent may end up not being edge-adjacent.



Cone Singularities

Using this approach, we are guaranteed that:

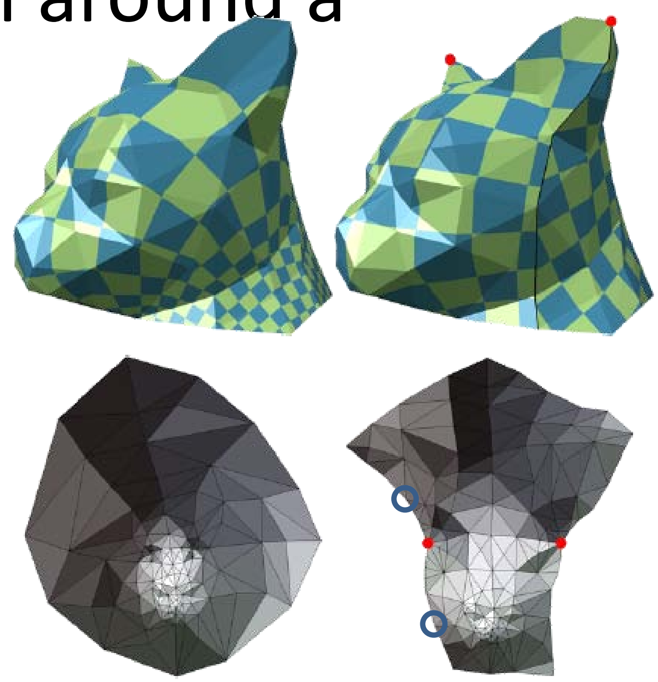
1. Two different instances of the same edge will have the same length.



Cone Singularities

Using this approach, we are guaranteed that:

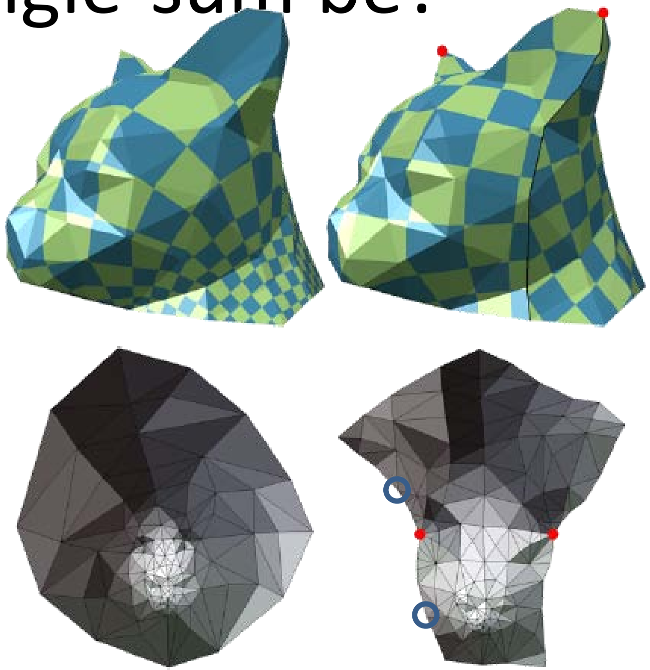
1. Two different instances of the same edge will have the same length.
2. And the cumulative angle-sum around a duplicated vertex is 2π.



Cone Singularities

In order to use the cone-singularities, there are two questions that need to be addressed:

1. Where should the cone-singularities go?
2. What should the prescribed angle-sum be?



Cone Singularities

1. Where should the cone-singularities go?

Using the scaling function u, the extrema of u are the points with the most scale distortion, so these are good candidates for singularities.

Cone Singularities

1. Where should the cone-singularities go?

Using the scaling function u, the extrema of u are the points with the most scale distortion, so these are good candidates for singularities.

We can iterate this process to add new cone vertices until the difference between the min. and max. distortion is within reasonable bounds.

Cone Singularities

2. What should the prescribed angle-sum be?

We can use the "natural" embedding, setting u_i=0 at the cone vertex instead of forcing the cone angle [Springborn *et al*.].

Cone Singularities

2. What should the prescribed angle-sum be?

We can use the “natural” embedding, setting u_i=0 at the cone vertex instead of forcing the cone angle [Springborn *et al*.].

Alternatively, we can simulate a stochastic process in which Gaussian curvature “walks-around” the mesh and accumulates in the cone vertices [Ben-Chen *et al*.].

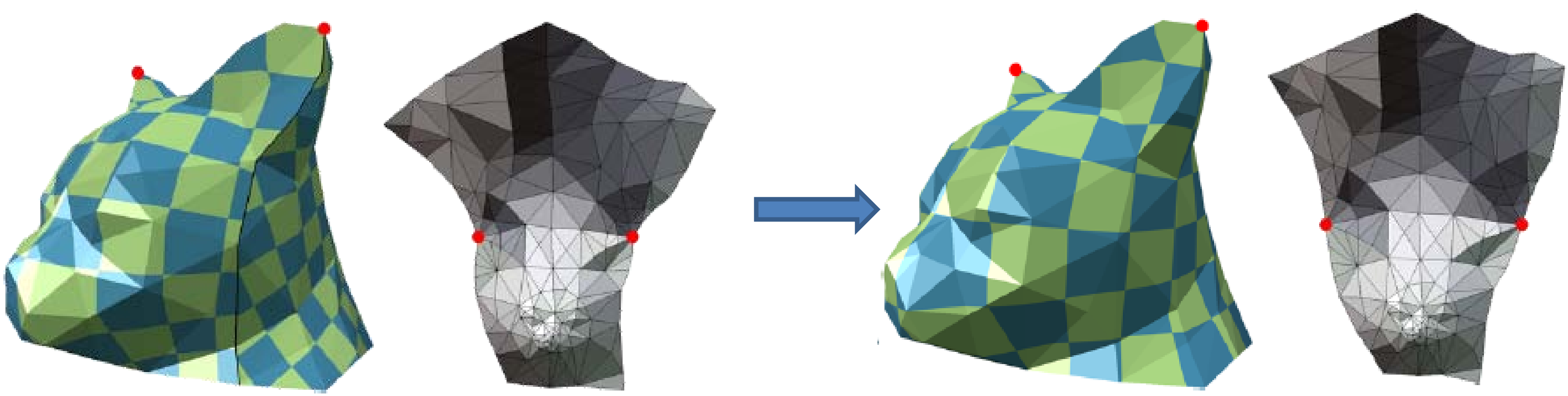
Texture Rectification

2. What should the prescribed angle-sum be?

In some applications, texture continuity may be more important than minimal distortion. In this case, we may want to force the cone-angles.

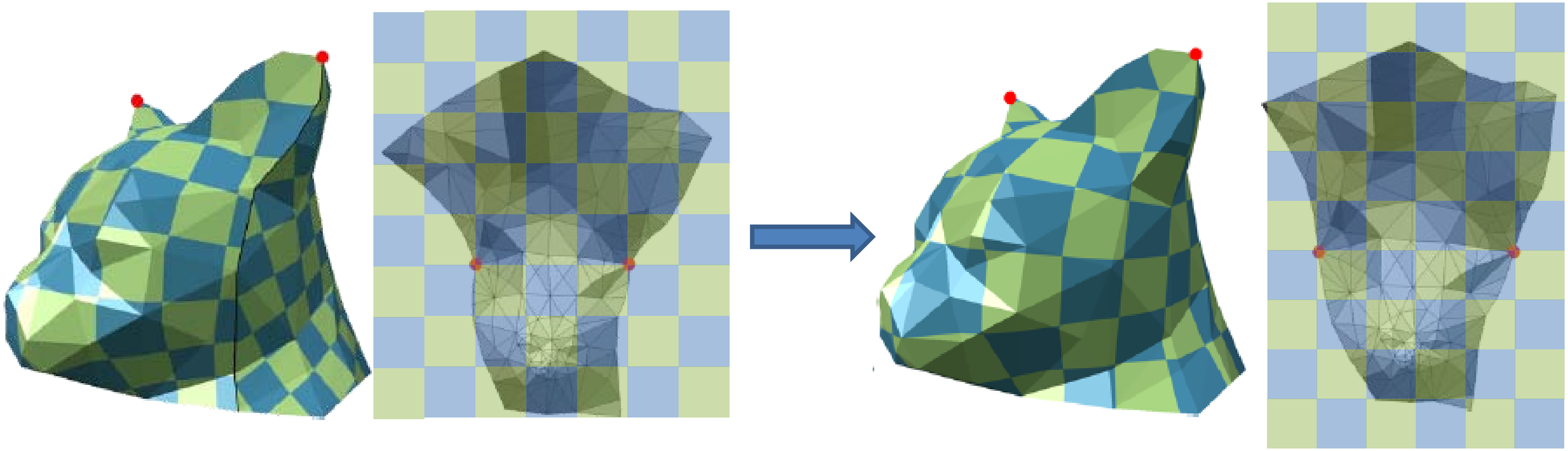
Texture Rectification

For example, if we have two cones, we can force the angles at the cones to be 2π. Then, if we have a texture with four-fold symmetry, and we scale/translate so that the cones are at (0,0) and (0,1), we get a seamless texturing.



Texture Rectification

For example, if we have two cones, we can force the angles at the cones to be 2π. Then, if we have a texture with four-fold symmetry, and we scale/translate so that the cones are at (0,0) and (0,1), we get a seamless texturing.



Texture Rectification

The situation becomes more complicated if we hare more cones but we can still get continuous texturing.