Differential Geometry:

Circle Packings

[*A Circle Packing Algorithm*, Collins and Stephenson]

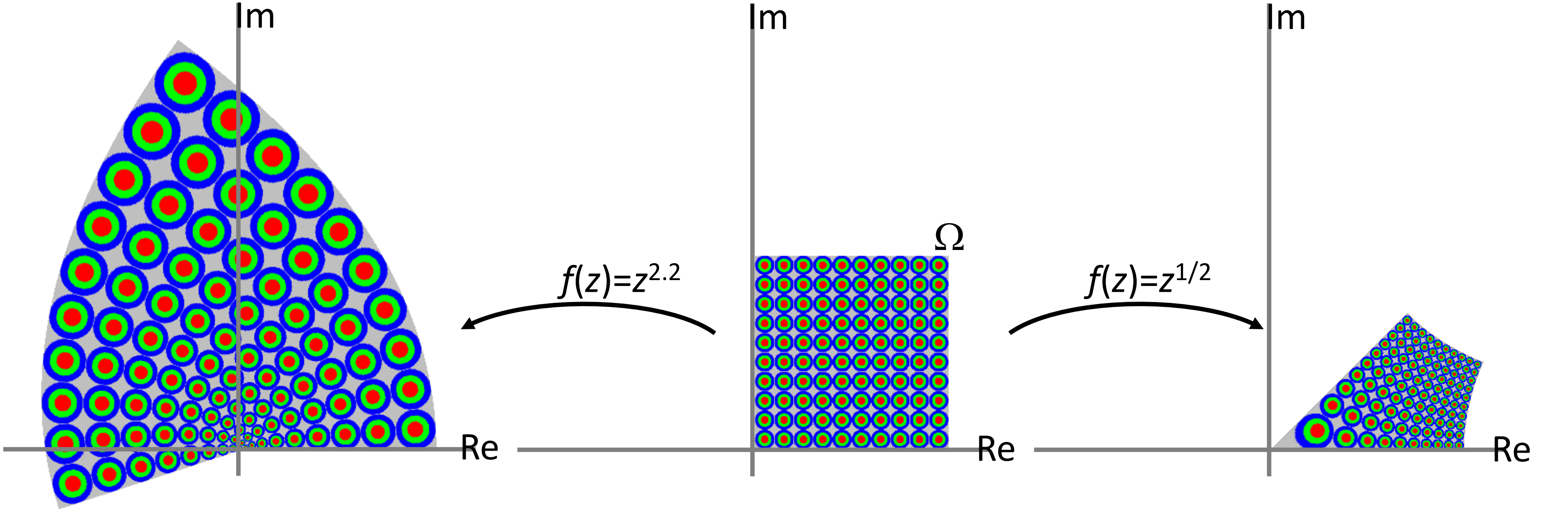
[*CirclePack*, Ken Stephenson]

Conformal Maps

Recall:

Given a domain $\Omega \subset \mathbb{R}^2$, the map $F:\Omega \to \mathbb{R}^2$ is *conformal* if it preserves oriented angles.

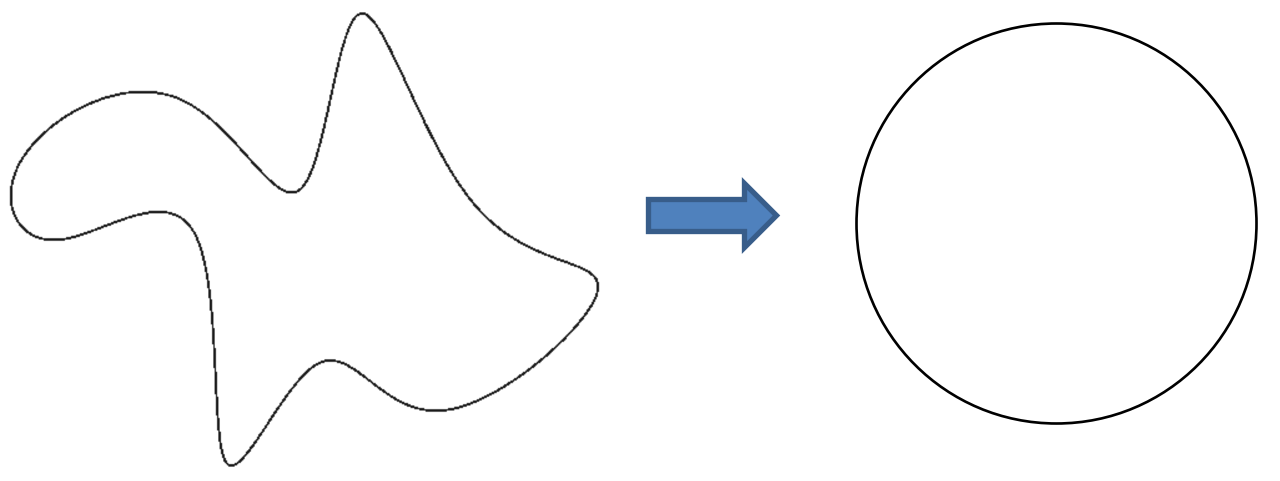
That is, the map sends "tiny" circles to circles.



Conformal Maps

Challenge:

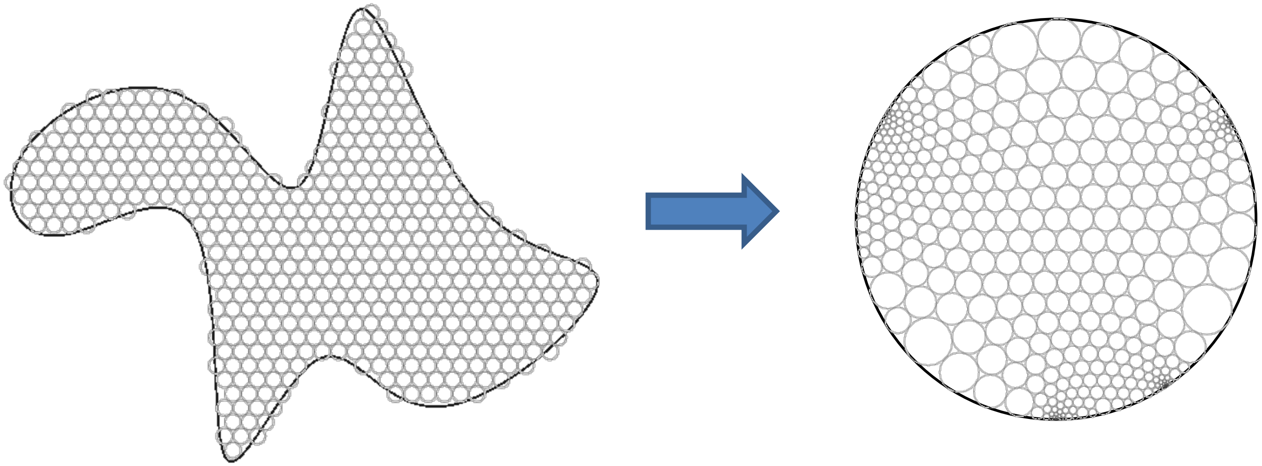
Given a curve in the plane, find a conformal map that sends the curve to the unit disk.



Conformal Maps

Intuition:

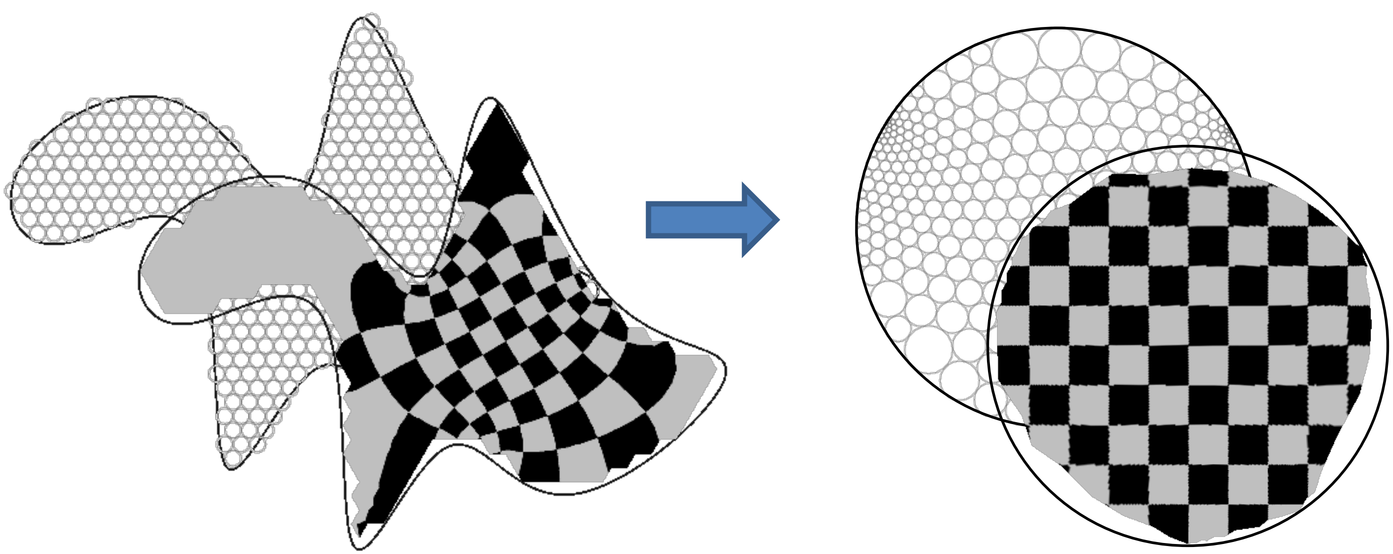
Since conformal maps send "tiny" circles to circles, we can get at a conformal map by packing the inside of curve with circles and mapping the packing into the disk.



Conformal Maps

<u>Thurston's Conjecture [1985]</u>:
<u>Proved by Rodin and Sullivan [1987]</u>:

Although the mapping of packings at a finite circle radius is not necessarily conformal…



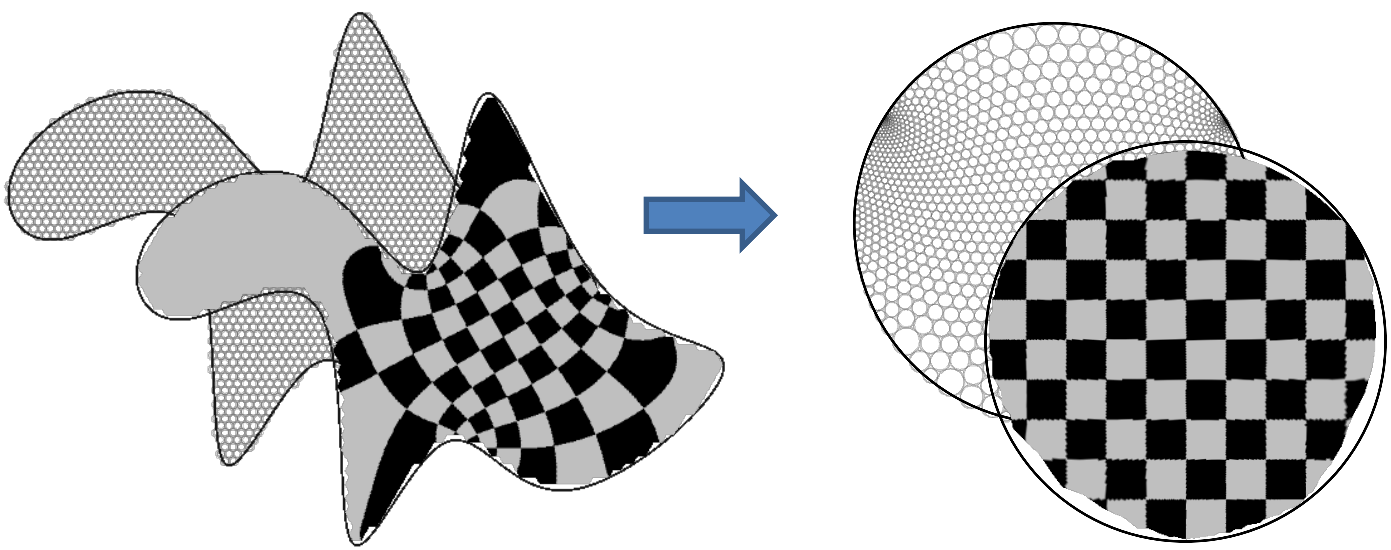
Conformal Maps

Thurston's Conjecture [1985]:

Proved by Rodin and Sullivan [1987]:

Although the mapping of packings at a finite circle radius is not necessarily conformal…

In the limit it will be.

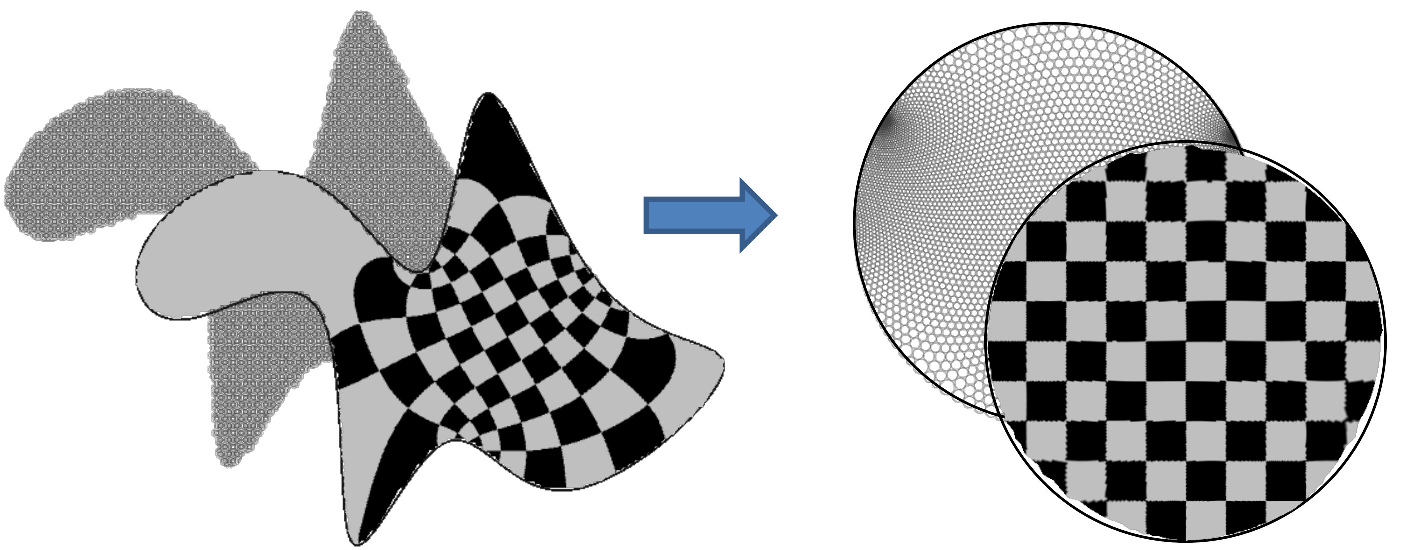


Conformal Maps

<u>Thurston's Conjecture [1985]</u>:
<u>Proved by Rodin and Sullivan [1987]</u>:

Although the packing at a finite circle radius the mapping is not necessarily conformal…

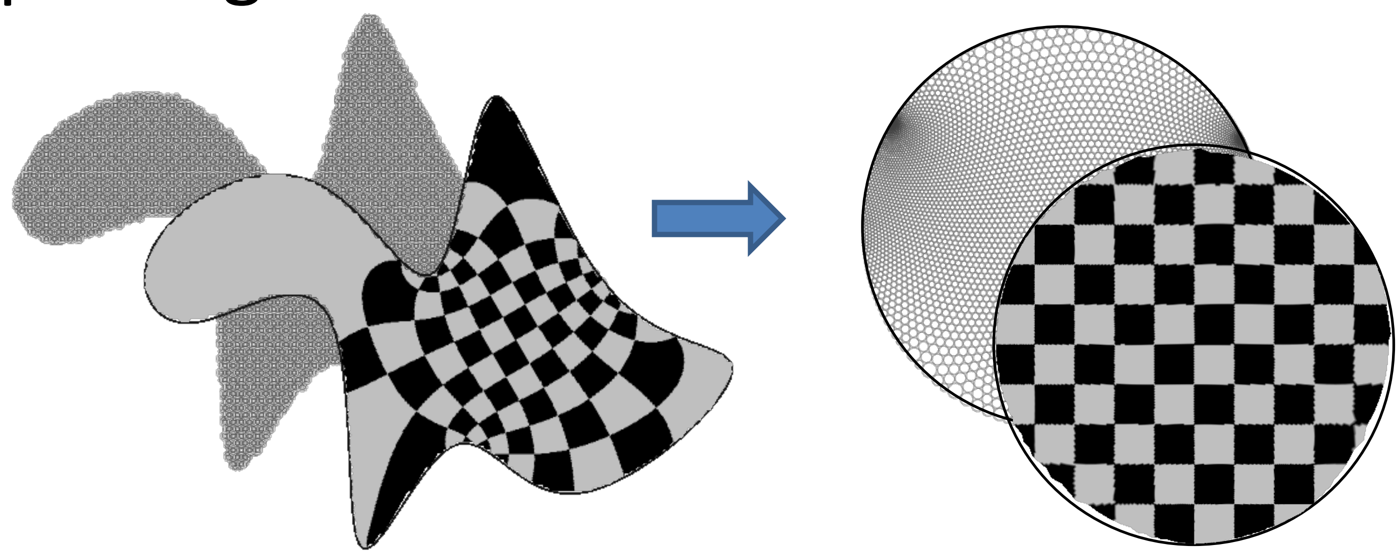
In the limit it will be.



Circle Packing

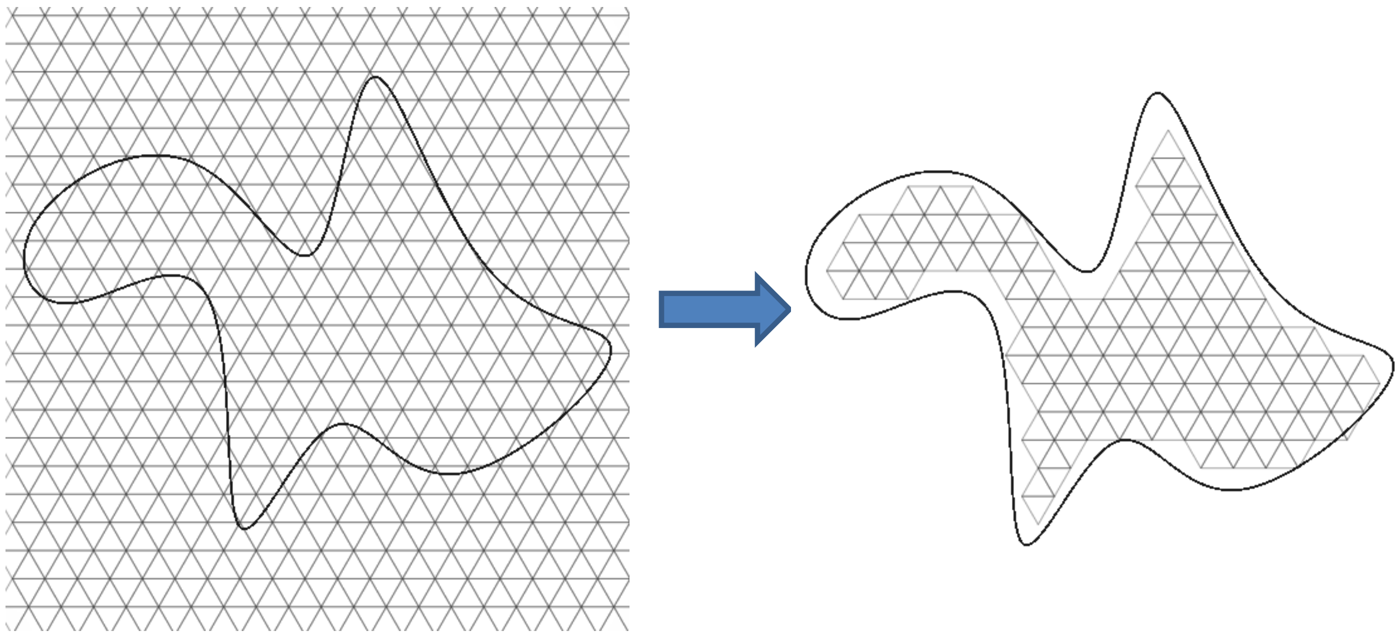
Key Steps:

1. We need to define a circular packing within the interior of the curve.
2. We need to transform the packing into a packing of a disk.



Circle Packing

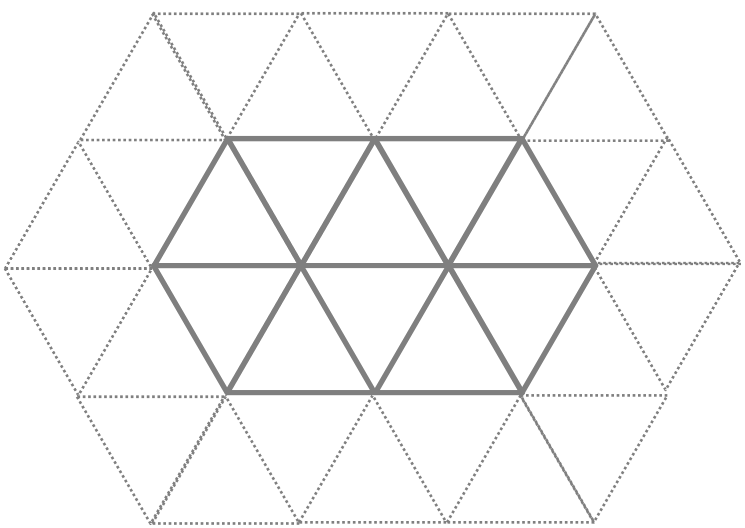
To define a packing within the curve, we can consider the restriction of a hexagonal lattice to the interior of the curve.

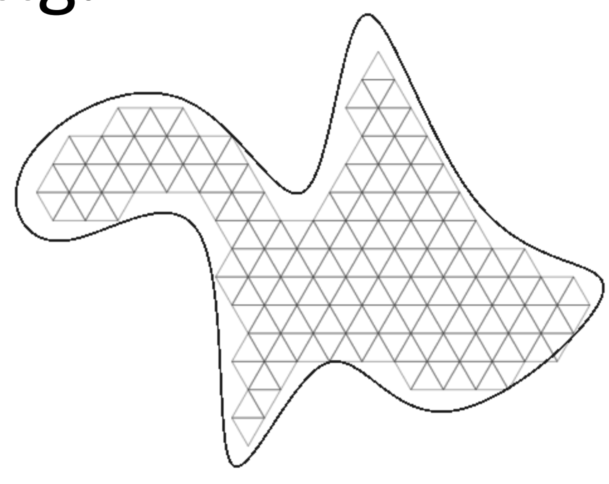


Circle Packing

To define a packing within the curve, we can consider the restriction of a hexagonal lattice to the interior of the curve.

On a hexagonal lattice, we can center circles on the vertices to get a packing.

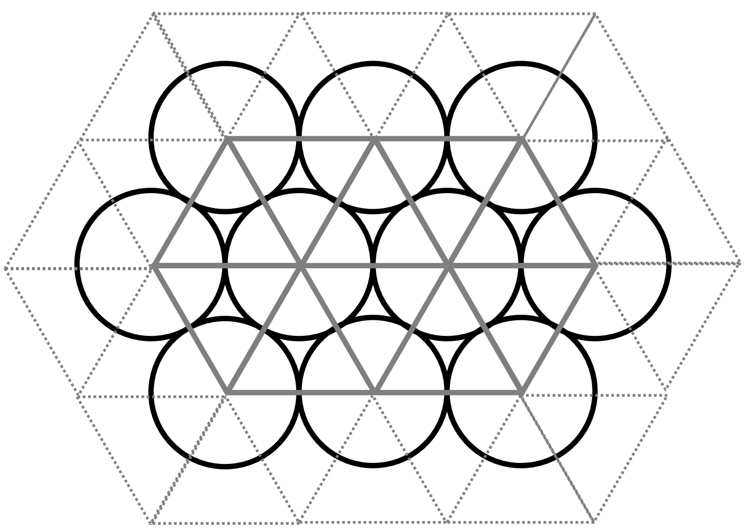


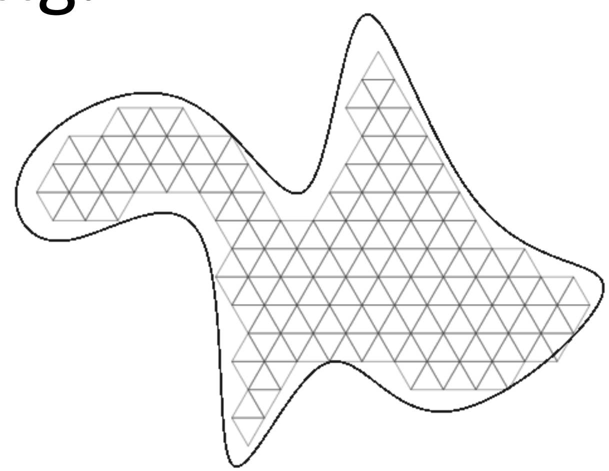


Circle Packing

To define a packing within the curve, we can consider the restriction of a hexagonal lattice to the interior of the curve.

On a hexagonal lattice, we can center circles on the vertices to get a packing.

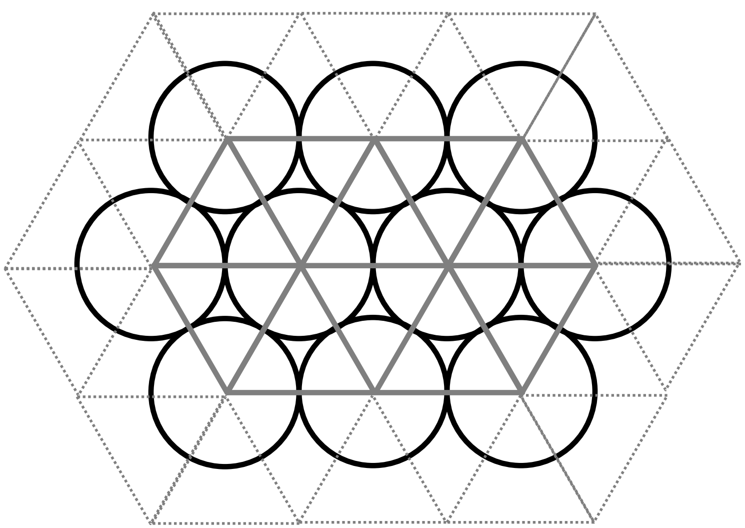


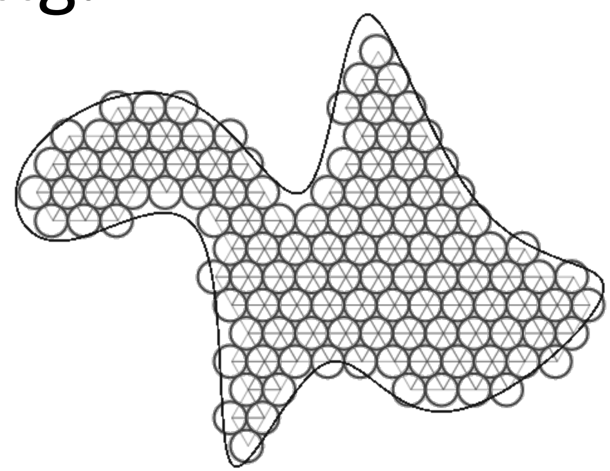


Circle Packing

To define a packing within the curve, we can consider the restriction of a hexagonal lattice to the interior of the curve.

On a hexagonal lattice, we can center circles on the vertices to get a packing.





Circle Packing

Definition:

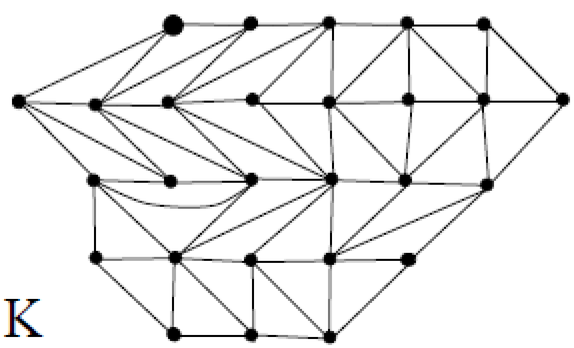
Given a triangulation *K*, a collection $P=\{c_v\}$ of circles in Ω is a *circle packing for K* if:

1. The packing *P* has a circle c_v associated with each vertex *v* of *K*.
2. Two circles $c_u, c_v \in P$ are tangent whenever (u,v) is an edge in *K*.
3. Three circles $c_u, c_v, c_w \in P$ form a positively oriented triple in Ω whenever (u,v,w) is a positively oriented triangle in *K*.

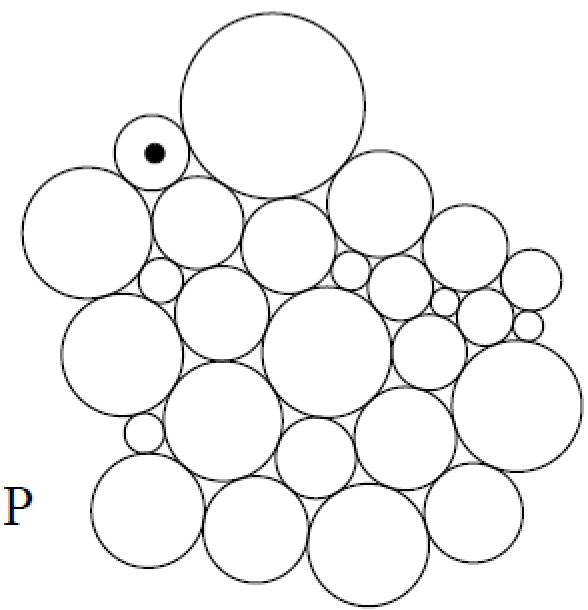
Circle Packing

<u>Definition</u>:

Given a triangulation *K*, a collection $P=\{c_v\}$ of circles in Ω is a *circle packing for K* if:



K

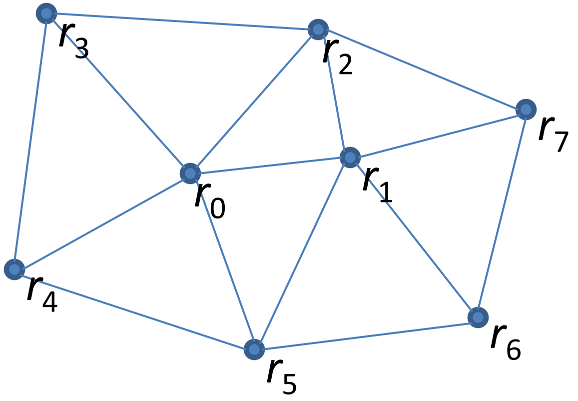


P

Circle Layout

Note:

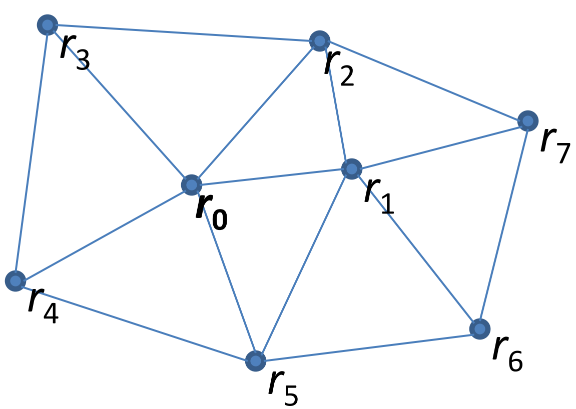
Given a triangulation K and an assignment of radii to the vertices of K that correspond to a valid circle-packing, we can construct a (essentially) unique circle packing.

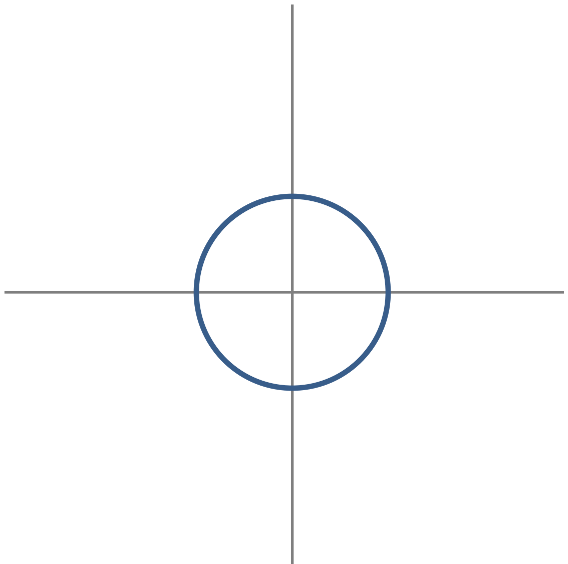


Circle Layout

<u>Algorithm</u>:

1. Pick an interior vertex v and place down a circle at the origin with radius r_v.

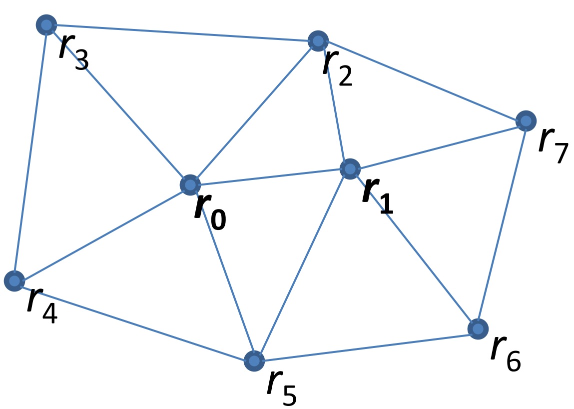


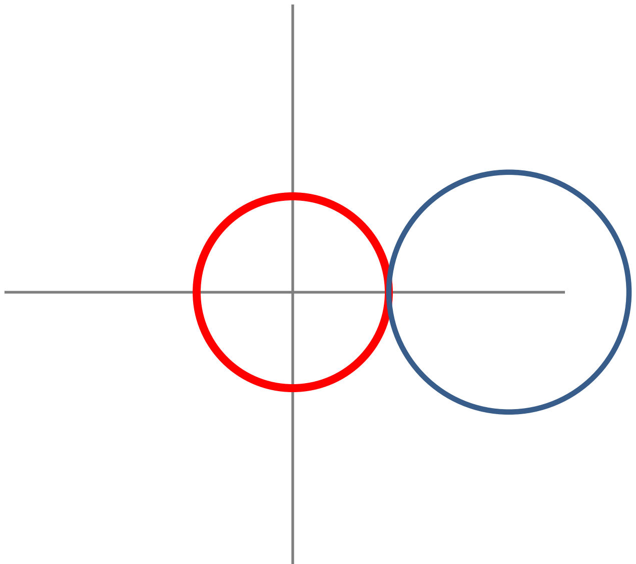


Circle Layout

Algorithm:

1. Pick an interior vertex *v* and place down a circle at the origin with radius r_v.
2. For each neighbor, add the associated circles, with correct radii, ensuring tangency.

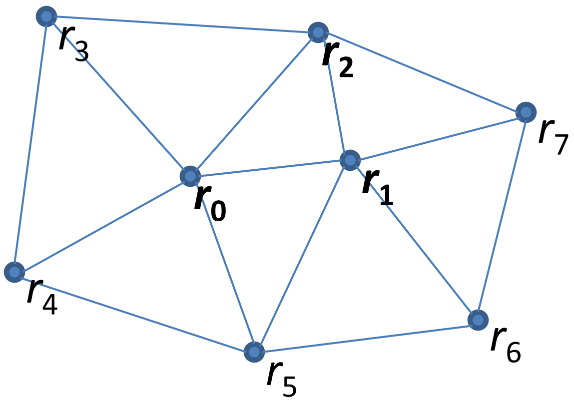


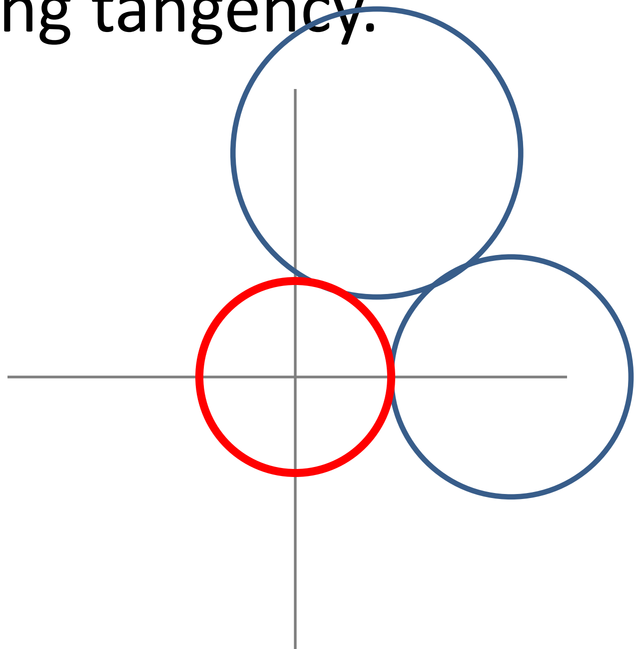


Circle Layout

Algorithm:

1. Pick an interior vertex *v* and place down a circle at the origin with radius r_v.
2. For each neighbor, add the associated circles, with correct radii, ensuring tangency.

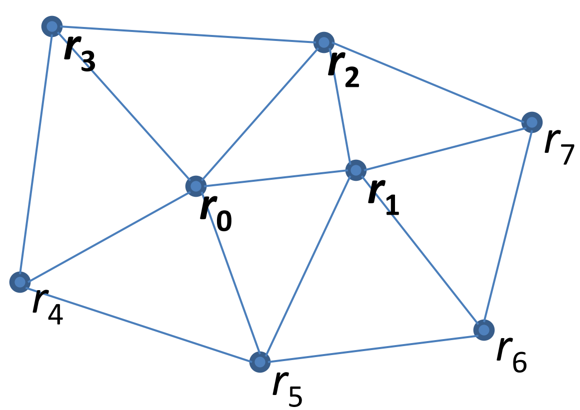


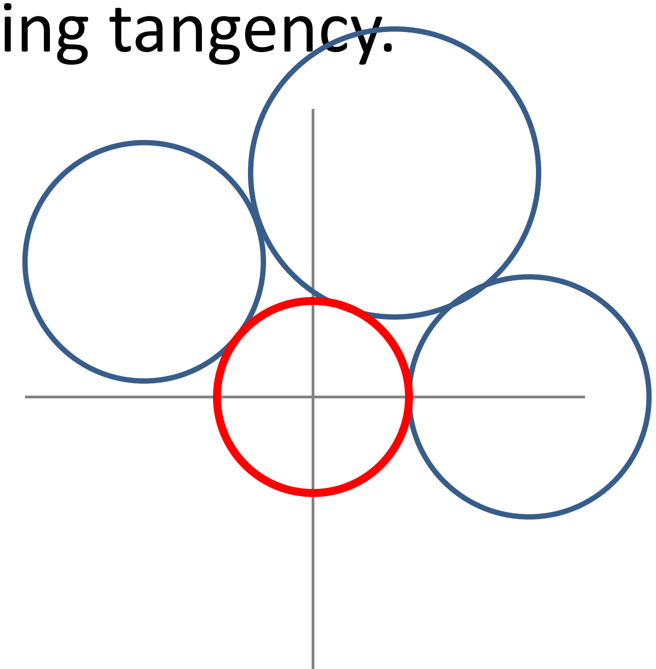


Circle Layout

<u>Algorithm</u>:

1. Pick an interior vertex *v* and place down a circle at the origin with radius r_v.
2. For each neighbor, add the associated circles, with correct radii, ensuring tangency.

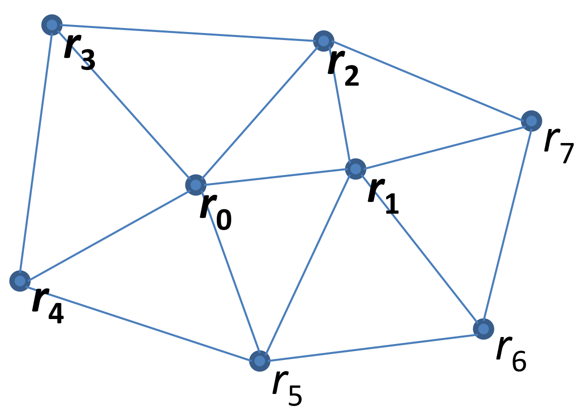


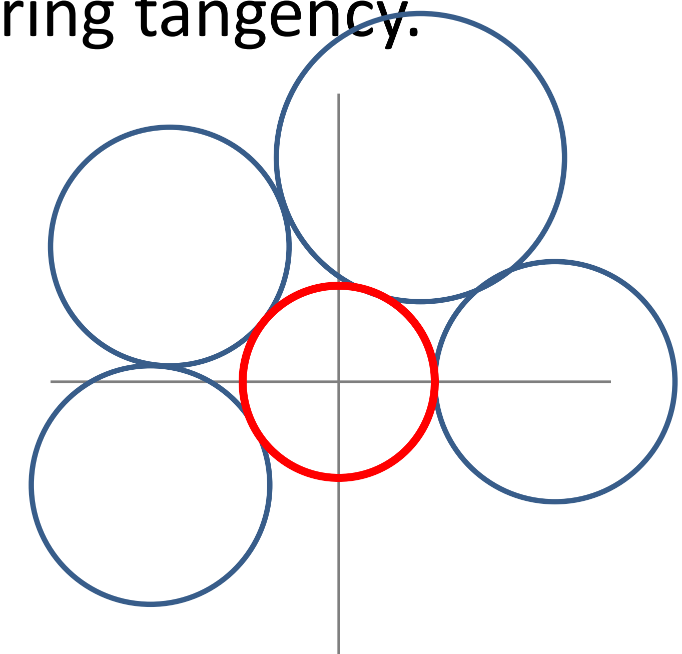


Circle Layout

<u>Algorithm</u>:

1. Pick an interior vertex v and place down a circle at the origin with radius r_v.
2. For each neighbor, add the associated circles, with correct radii, ensuring tangency.

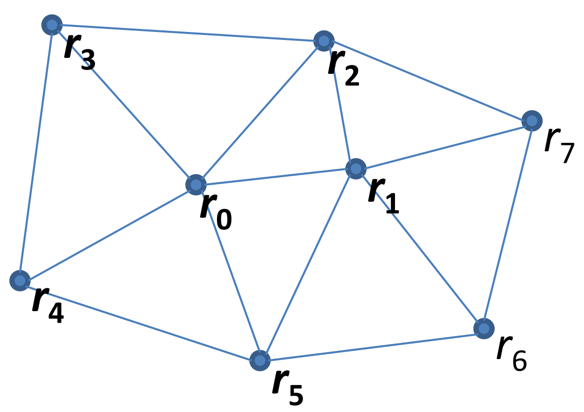


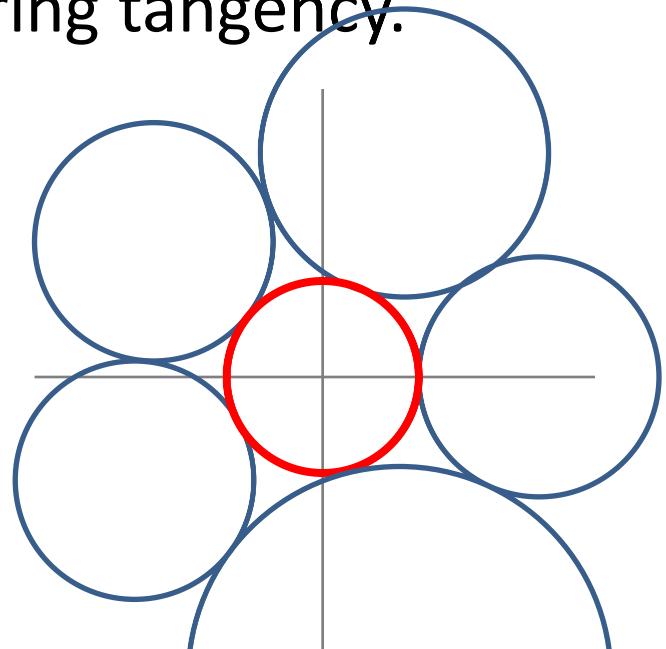


Circle Layout

<u>Algorithm</u>:

1. Pick an interior vertex *v* and place down a circle at the origin with radius r_v.
2. For each neighbor, add the associated circles, with correct radii, ensuring tangency.

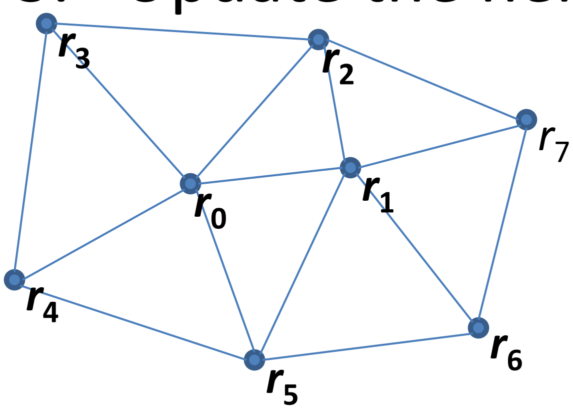


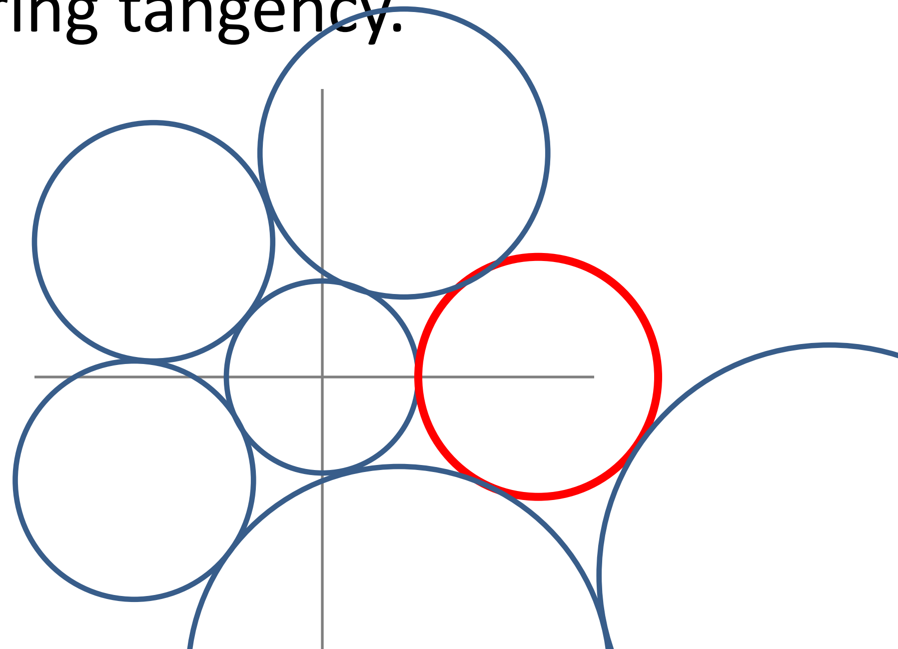


Circle Layout

<u>Algorithm</u>:

1. Pick an interior vertex *v* and place down a circle at the origin with radius r_v.
2. For each neighbor, add the associated circles, with correct radii, ensuring tangency.
3. Update the neighbors.

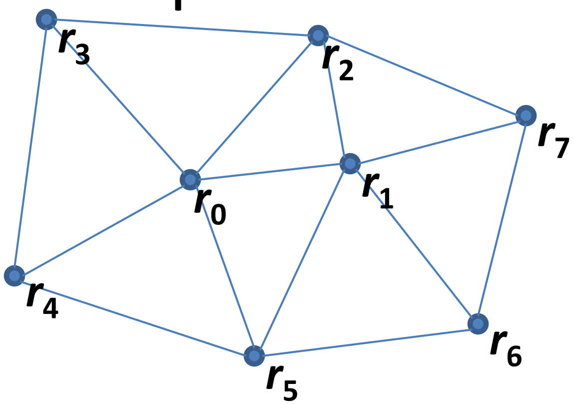


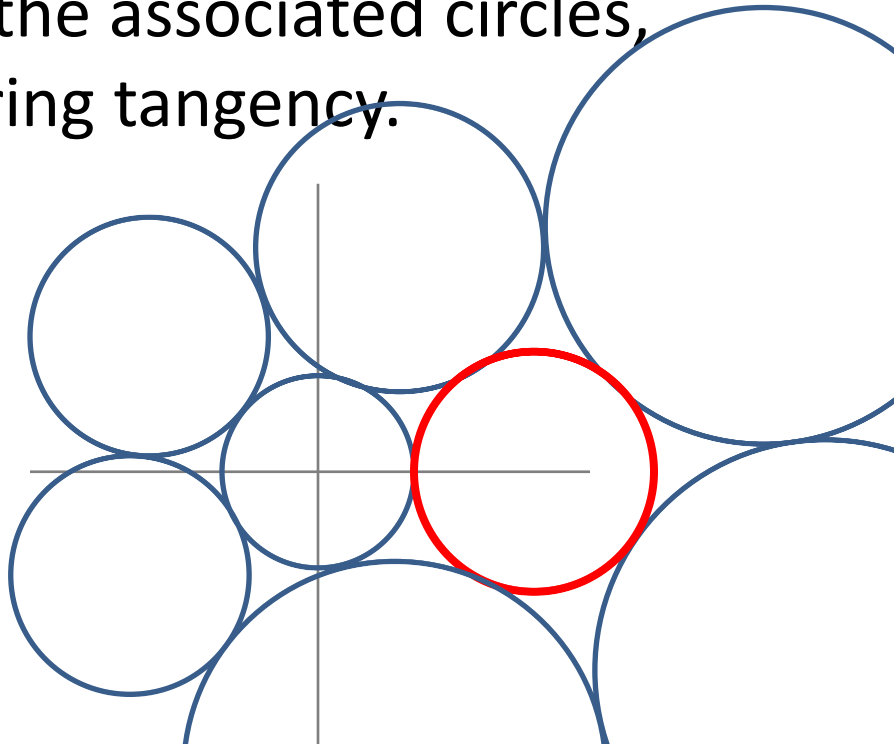


Circle Layout

<u>Algorithm</u>:

1. Pick an interior vertex *v* and place down a circle at the origin with radius r_v.
2. For each neighbor, add the associated circles, with correct radii, ensuring tangency.
3. Update the neighbors.

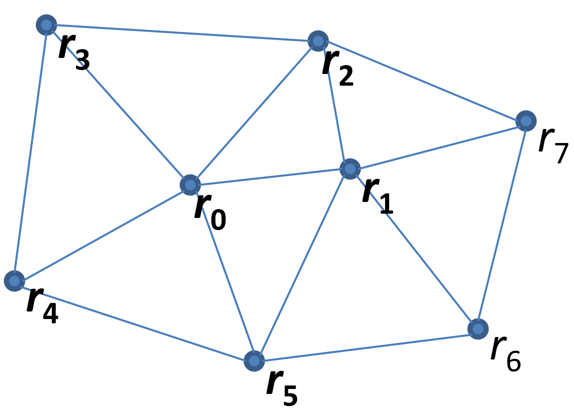


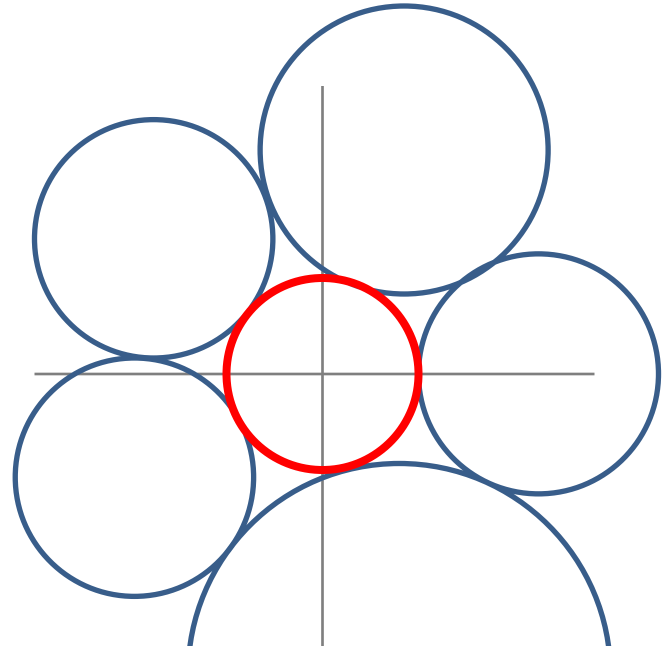


Circle Layout

<u>Necessary Condition</u>:

When we finish laying out an interior vertex, the last and first circles should be tangent.



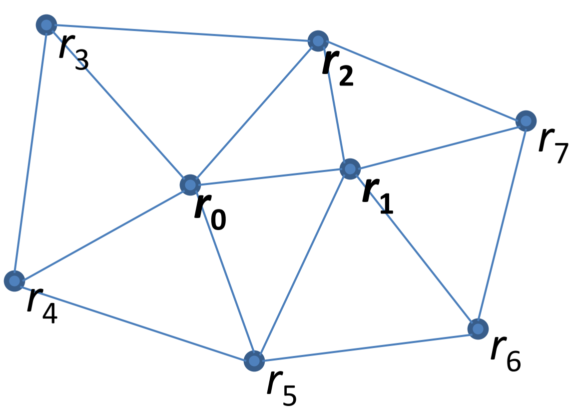


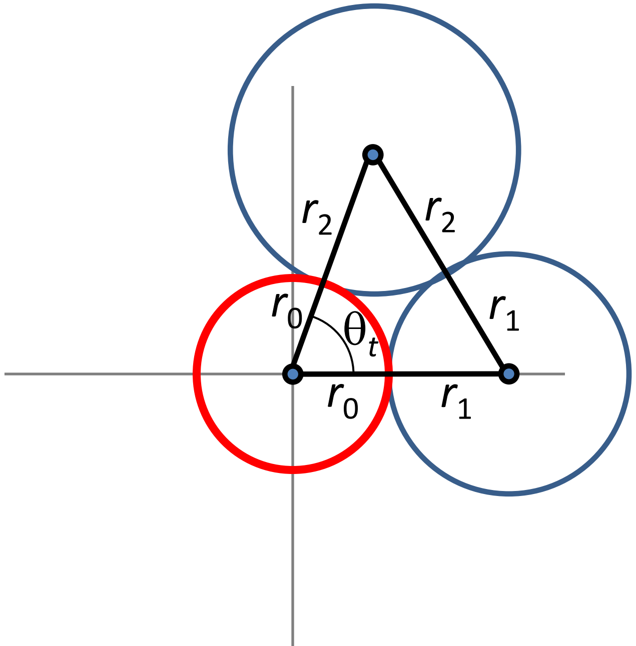
Circle Layout

<u>Necessary Condition</u>:

Given a triangle t={v_0,v_1,v_2}, we can compute the angle θ_t at v_0 using the law of cosines:

$$(r_1+r_2)^2=(r_0+r_1)^2+(r_0+r_2)^2-2(r_0+r_1)(r_0+r_2)\cos\theta_t$$

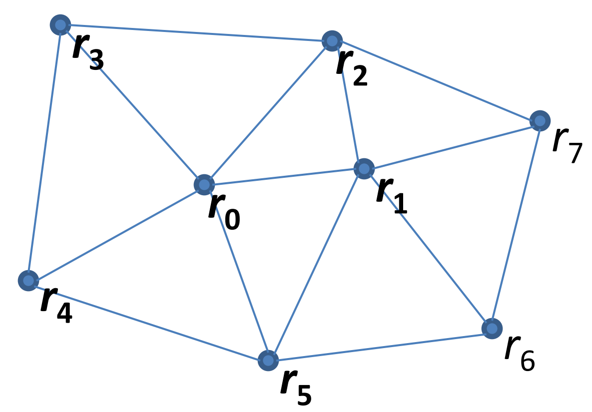


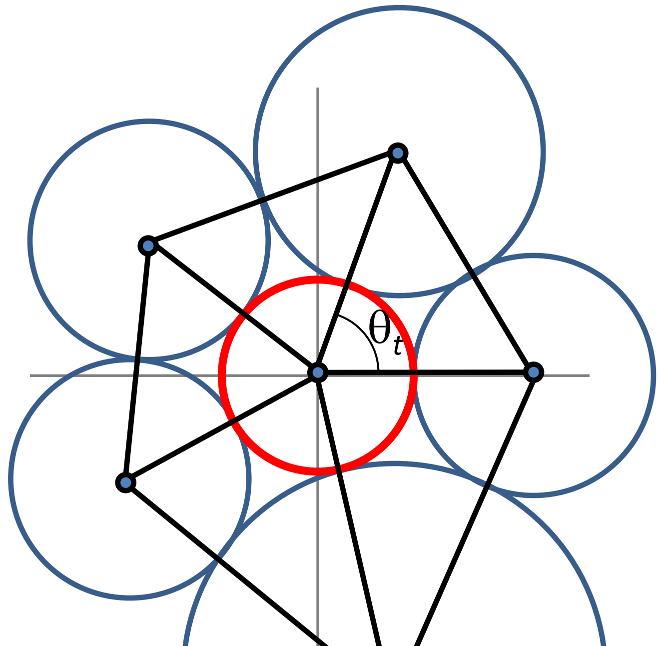


Circle Layout

<u>Necessary Condition</u>:

The last and first circles are tangent iff. the sum of angles about every interior angle *v* is $2\pi n$.





Circle Packing

<u>Necessary and Sufficient Condition</u>:

Given a triangulation *K* of a topological disk and a constraint radius at each boundary vertex, there is an (essentially) unique circle packing realizing the boundary constraints, with interior angles summing to 2π.

Circle Packing

General Approach:

1. Start with an initial assignment of radii to vertices that agrees with the prescribed radii on the boundary.

Circle Packing

<u>General Approach</u>:

1. Start with an initial assignment of radii to vertices that agrees with the prescribed radii on the boundary.
2. For each interior vertex, adjust the radius so that the angle-sum gets closer to 2π.

Circle Packing

General Approach:

1. Start with an initial assignment of radii to vertices that agrees with the prescribed radii on the boundary.
2. For each interior vertex, adjust the radius so that the angle-sum gets closer to 2π.
3. Repeat step 2 until convergence.

Circle Packing

In order for this to work:

- The algorithm needs to converge to an assignment of radii.
- The solution to which the algorithm converges has to be correct.

Circle Packing

Proof of Convergence:

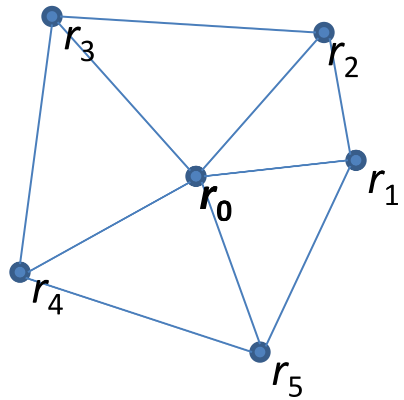
To show convergence, we show that the error over all the interior vertices:

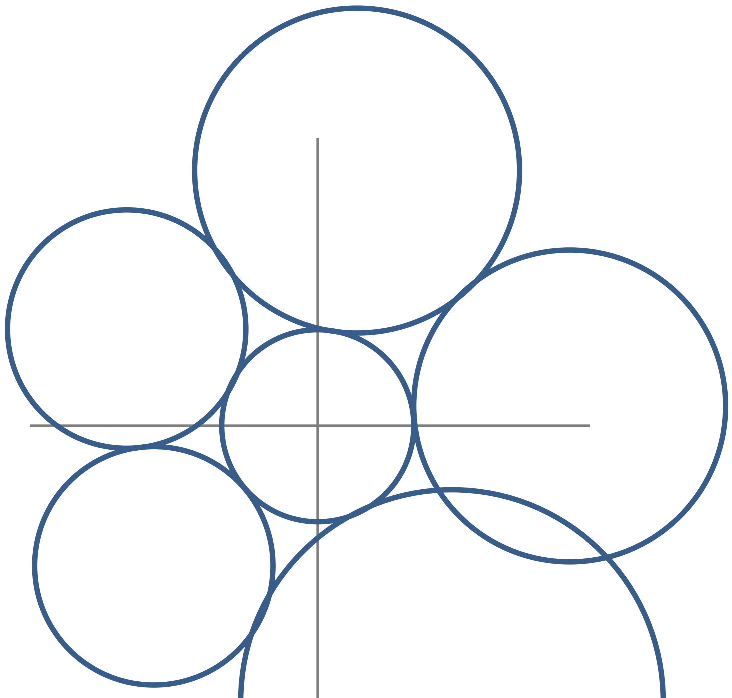
$$E = \sum_{v \in K^\circ} \left| \text{AngleSum}(v) - 2\pi \right|$$

is always decreasing.

Circle Packing

To do this, we need to consider what happens to the angle sum at v_0 when we change the radii.

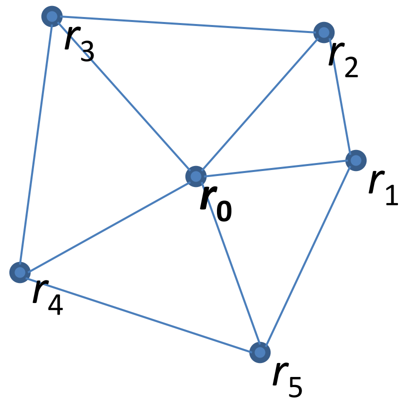


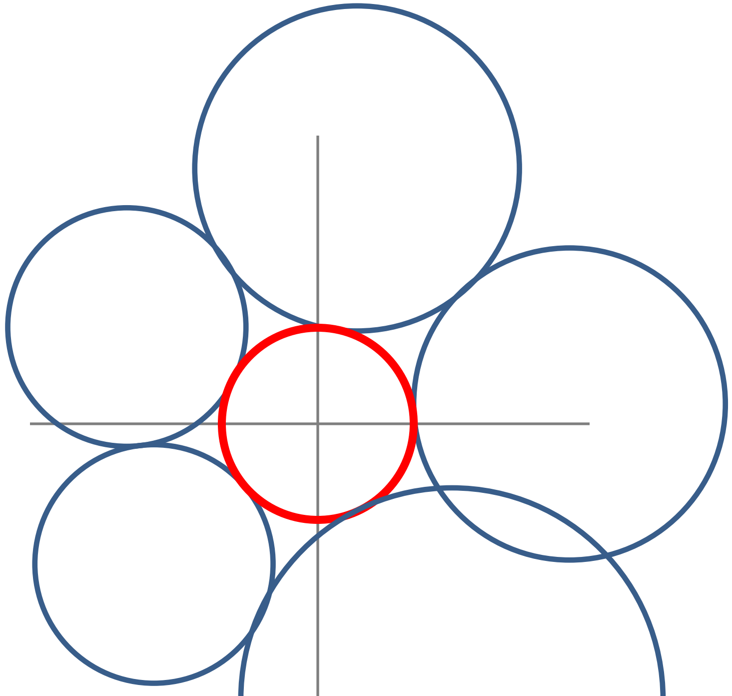


Circle Packing

To do this, we need to consider what happens to the angle sum at v_0 when we change the radii.

<u>Modifying the radius of v_0:</u>



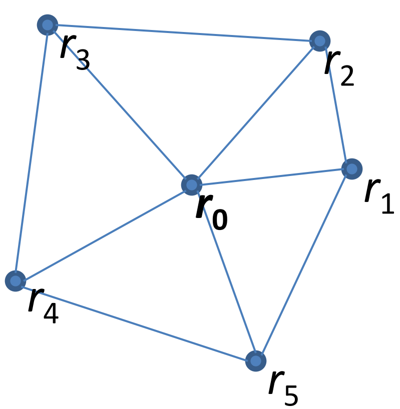


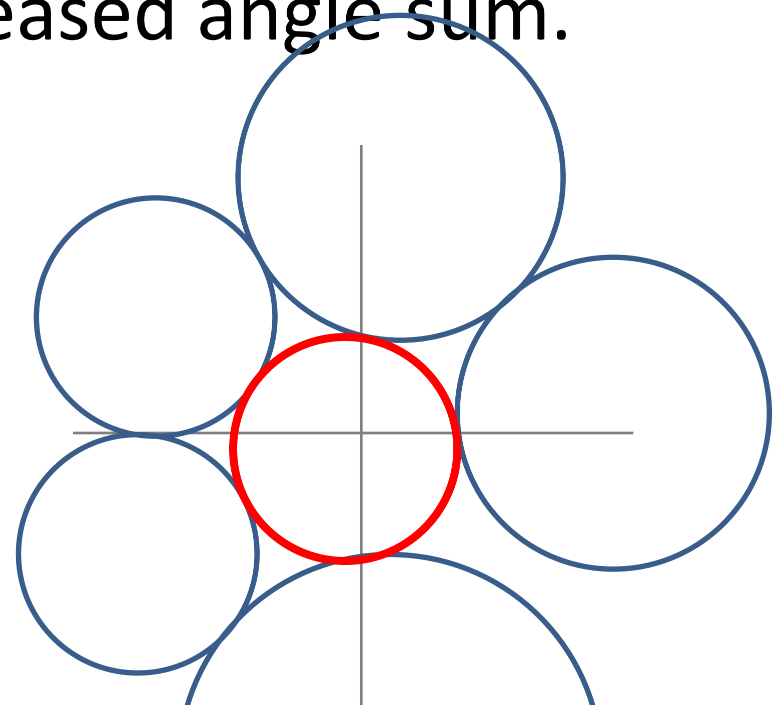
Circle Packing

To do this, we need to consider what happens to the angle sum at v_0 when we change the radii.

<u>Modifying the radius of v_0</u>:

Increased radius $\Rightarrow$ decreased angle sum.



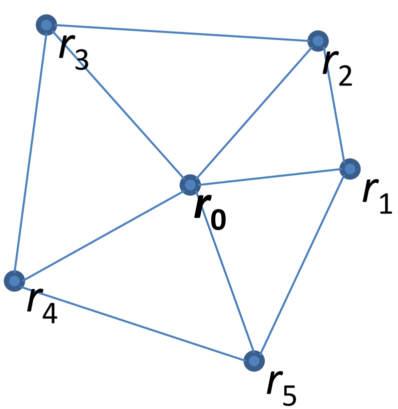


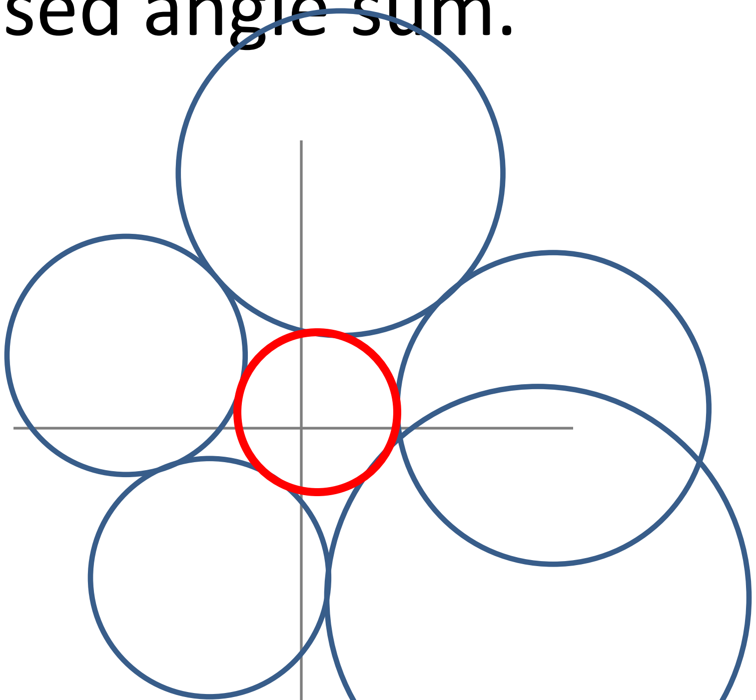
Circle Packing

To do this, we need to consider what happens to the angle sum at v_0 when we change the radii.

<u>Modifying the radius of v_0</u>:

Decreased radius $\Rightarrow$ increased angle sum.

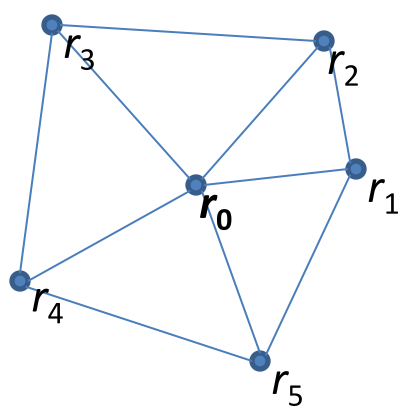


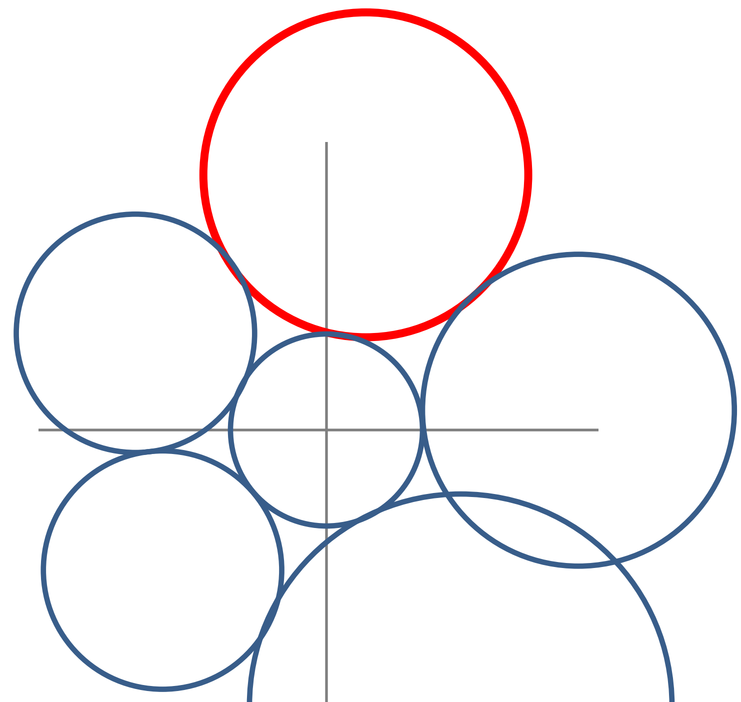


Circle Packing

To do this, we need to consider what happens to the angle sum at v_0 when we change the radii.

<u>Modifying the radius of a neighbor of v_1:</u>



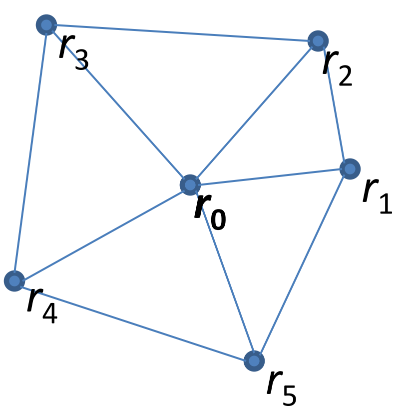


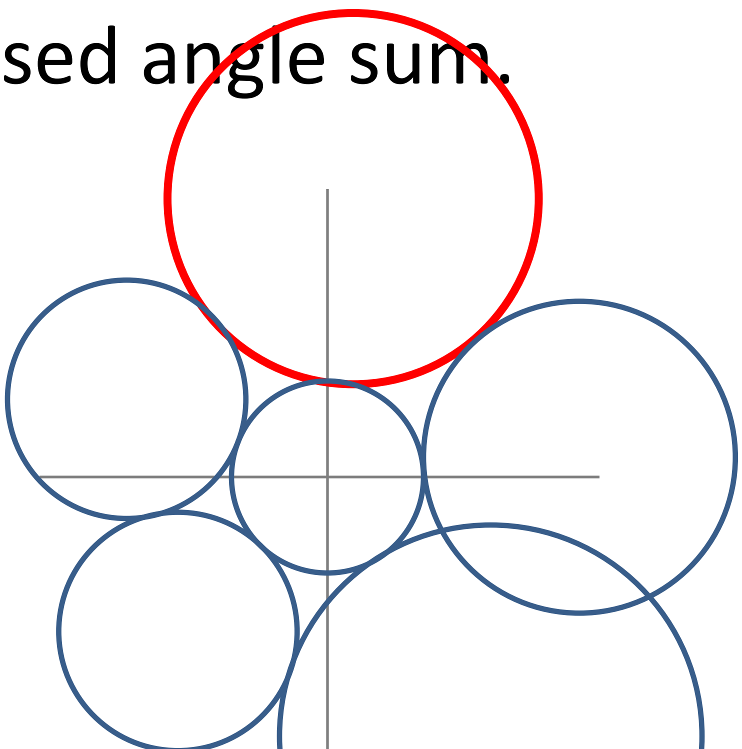
Circle Packing

To do this, we need to consider what happens to the angle sum at v_0 when we change the radii.

<u>Modifying the radius of a neighbor of v_0:</u>

Increased radius $\Rightarrow$ increased angle sum.



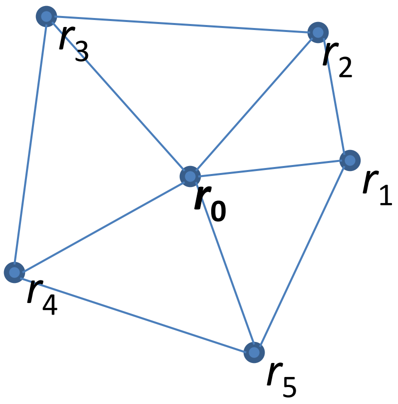


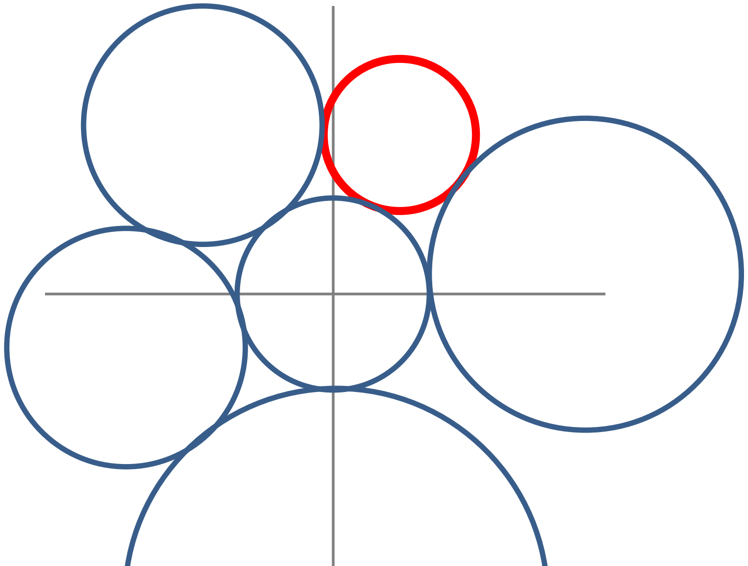
Circle Packing

To do this, we need to consider what happens to the angle sum at v_0 when we change the radii.

<u>Modifying the radius of a neighbor of v_0</u>:

Decreased radius $\Rightarrow$ decreased angle sum.





Circle Packing

<u>Proof of Convergence</u>:

To show convergence, we show that the error over all the interior vertices:

$$E = \sum_{v \in K^\circ} \left| \mathrm{AngleSum}(v) - 2\pi \right|$$

is always decreasing.

Circle Packing

<u>Proof of Convergence</u>:

To show convergence, we show that the error over all the interior vertices:

$$E = \sum_{v \in K^\circ} \left| \text{AngleSum}(v) - 2\pi \right|$$

is always decreasing.

Note that for any triangle $t \in K$, the sum of the angles defined by any assignment of radii is always π.

Circle Packing

Proof of Convergence:

To show convergence, we show that the error over all the interior vertices:

$$E = \sum_{v \in K^\circ} \left| \mathrm{AngleSum}(v) - 2\pi \right|$$

is always decreasing.

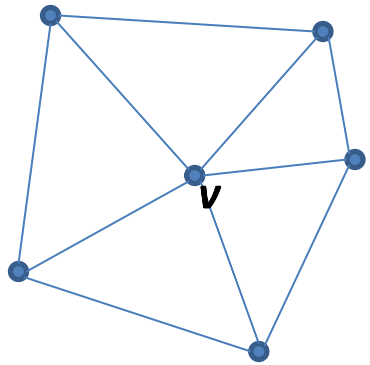
Note that for any triangle $t \in K$, the sum of the angles defined by any assignment of radii is always π.

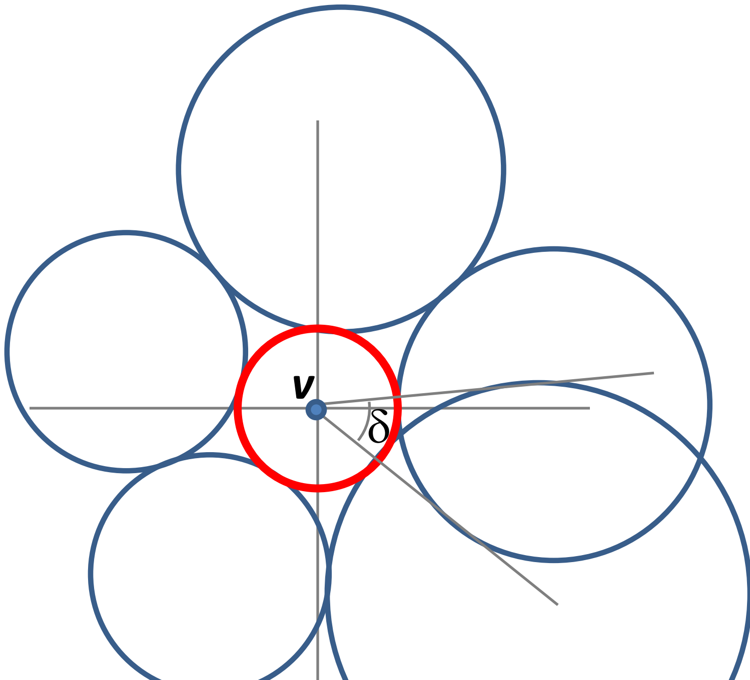
So regardless of the assignment the sum of angles will be πx(# of tris in K).

Circle Packing

<u>Proof of Convergence</u>:

Suppose that *v* is an interior vertex at which the angle sum is $2\pi+\delta$.

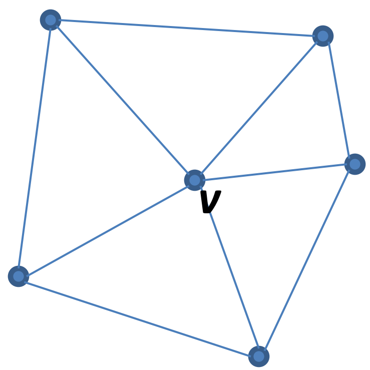


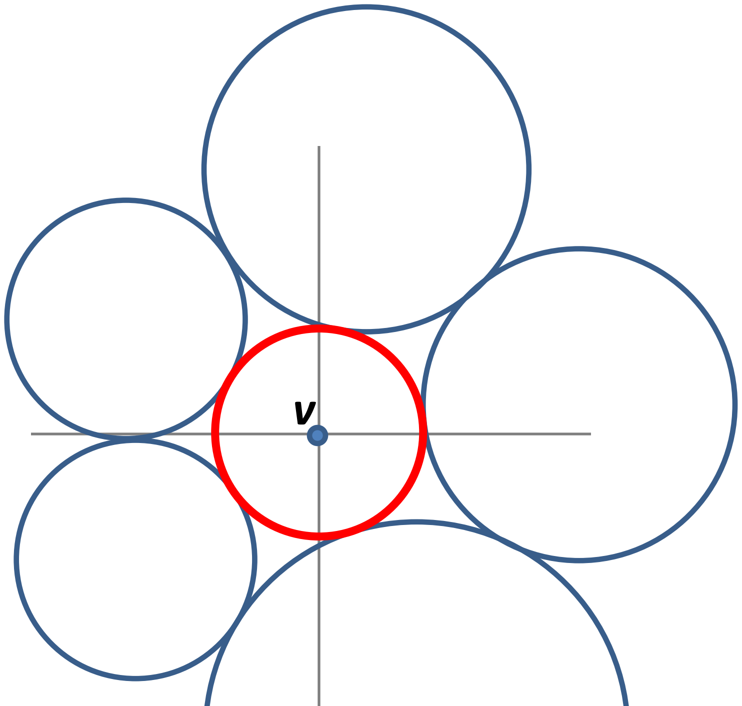


Circle Packing

<u>Proof of Convergence</u>:

If we increase the radius at *v* so the angle-sum becomes 2π:



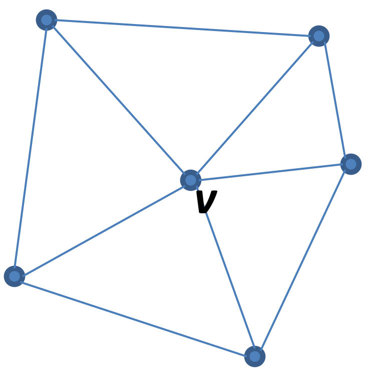


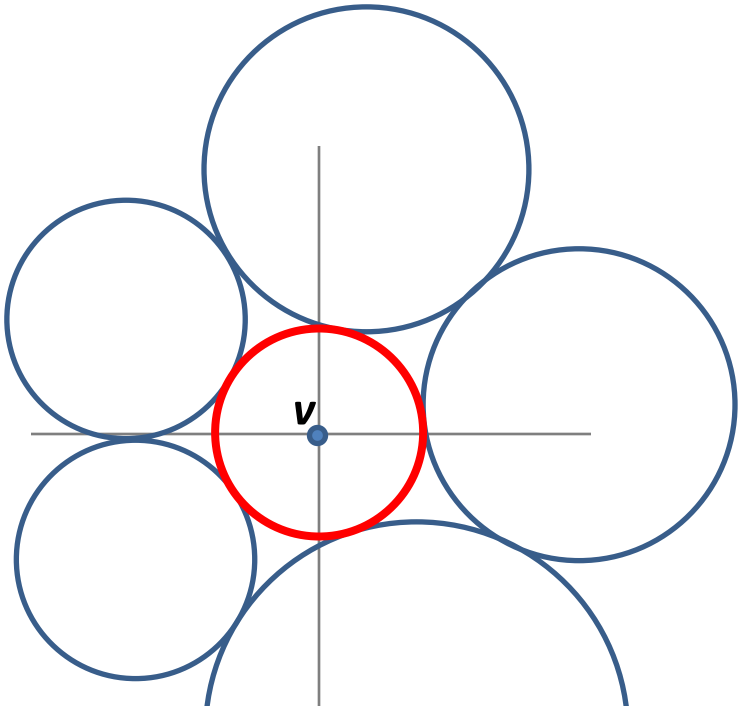
Circle Packing

<u>Proof of Convergence</u>:

If we increase the radius at v so the angle-sum becomes 2π:

- At v, the error reduces by δ.



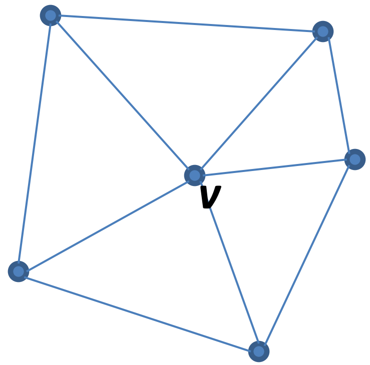


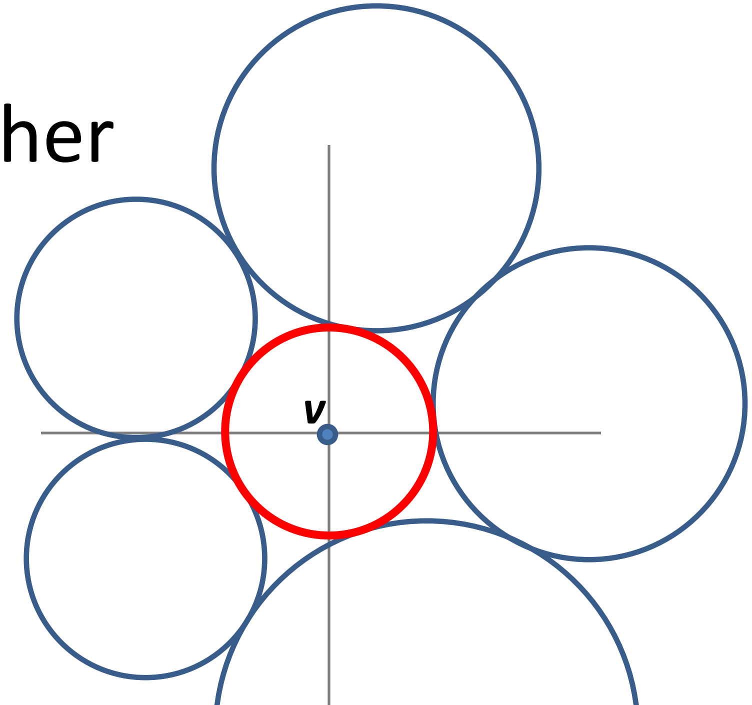
Circle Packing

Proof of Convergence:

If we increase the radius at *v* so the angle-sum becomes 2π:

- At *v*, the error reduces by δ.
- The total change at all the other vertices is δ.





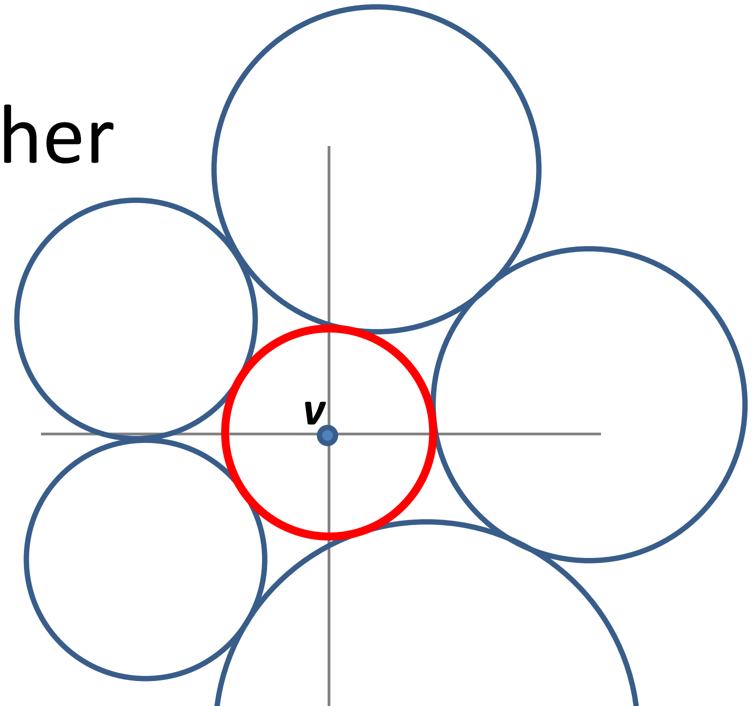
Circle Packing

Proof of Convergence:

If we increase the radius at *v* so the angle-sum becomes 2π:

- At *v*, the error reduces by δ.
- The total change at all the other vertices is δ.

So in the worst case, the total error at all other vertices increases by δ.



Circle Packing

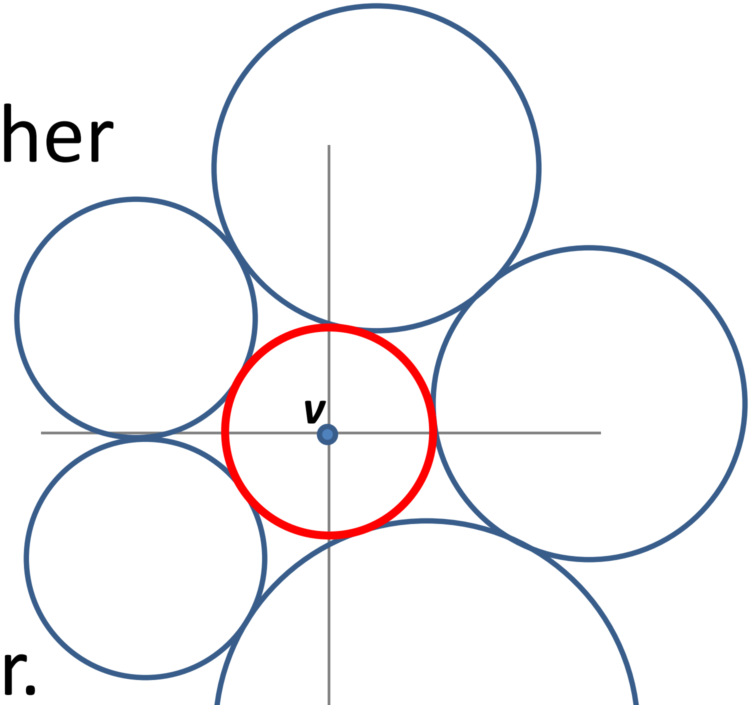
<u>Proof of Convergence</u>:

If we increase the radius at *v* so the angle-sum becomes 2π:

- At *v*, the error reduces by δ.
- The total change at all the other vertices is δ.

So in the worst case, the total error at all other vertices increases by δ.

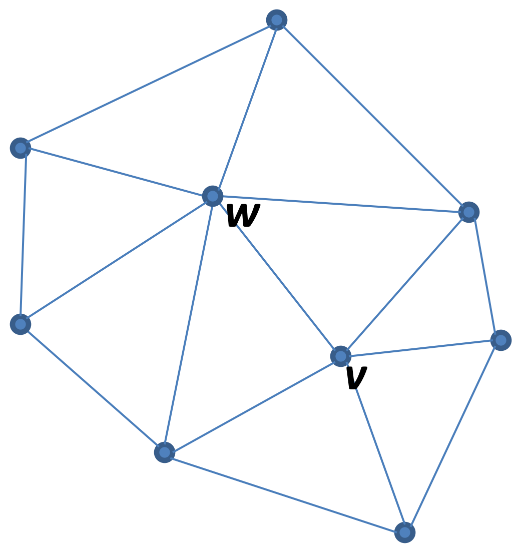
$\Rightarrow$ The error doesn't get bigger.

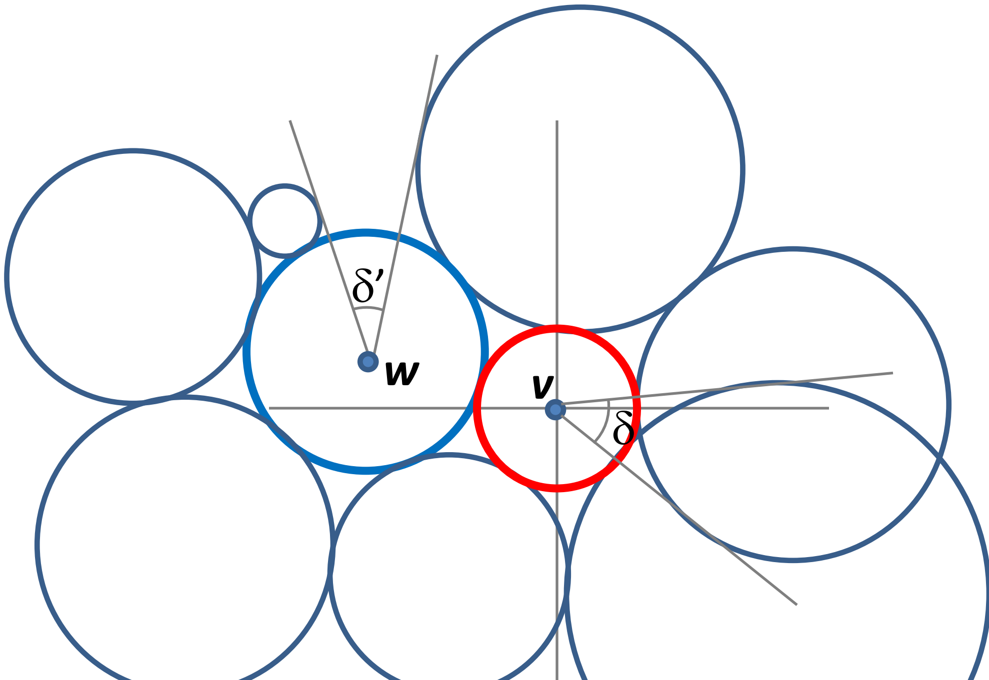


Circle Packing

<u>Proof of Convergence</u>:

However, if there is even one interior neighbor *w* of *v* whose angle-sum is too small…

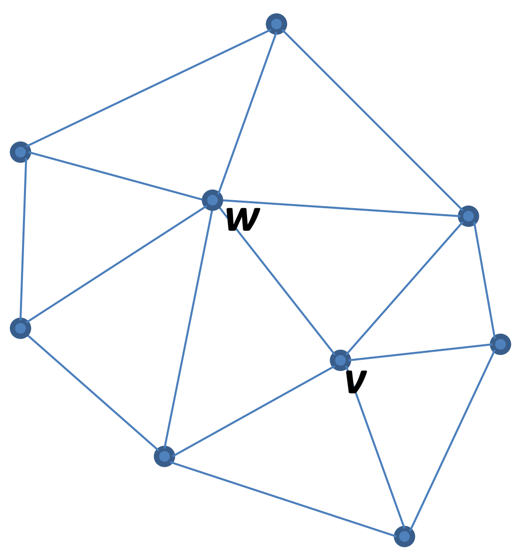


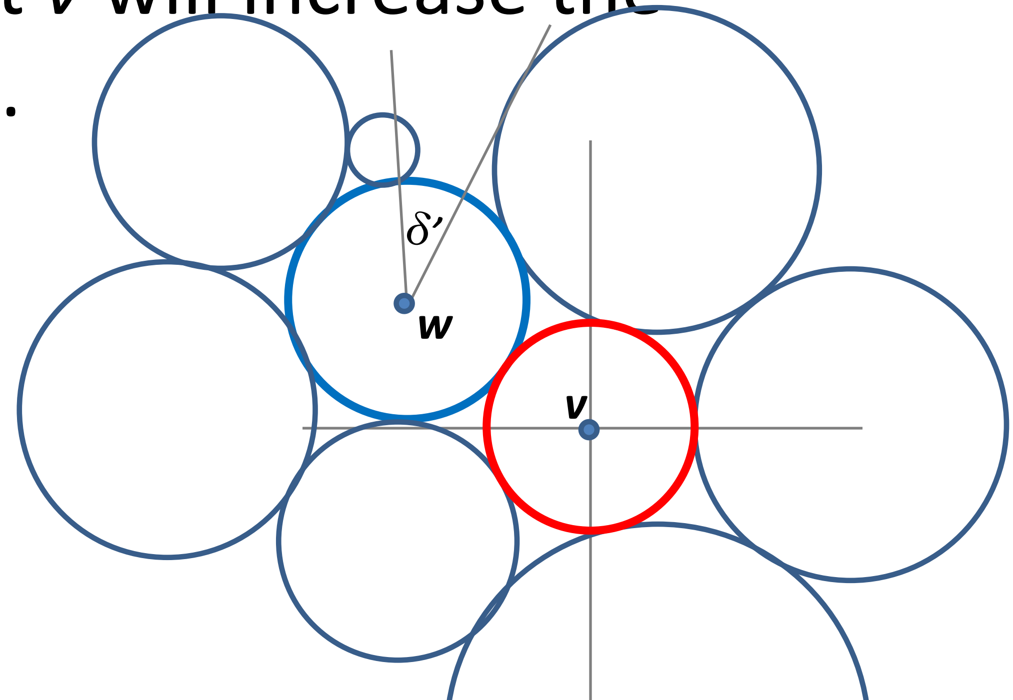


Circle Packing

<u>Proof of Convergence</u>:

However, if there is even one neighbor of *v* whose angle-sum is too small…
Increasing the radius at *v* will increase the neighbor's angle-sum…





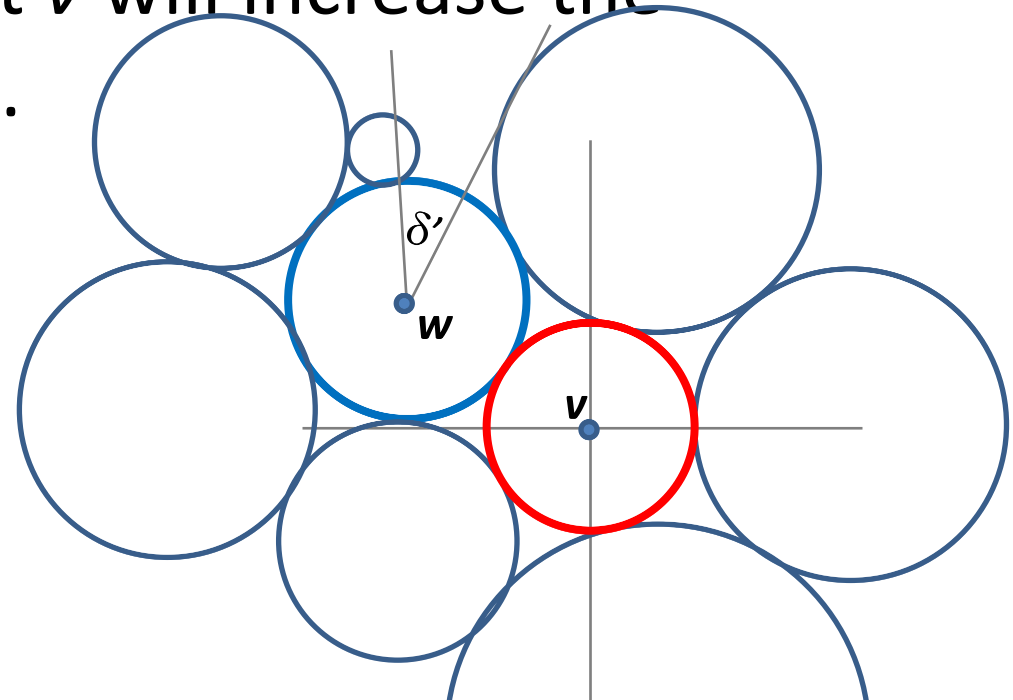
Circle Packing

Proof of Convergence:

However, if there is even one neighbor of *v* whose angle-sum is too small…

Increasing the radius at *v* will increase the neighbor's angle-sum…

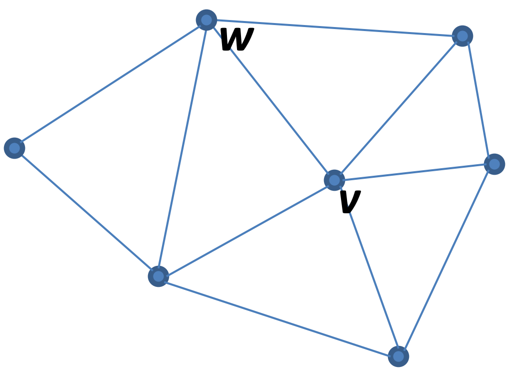
And the total error will decrease.

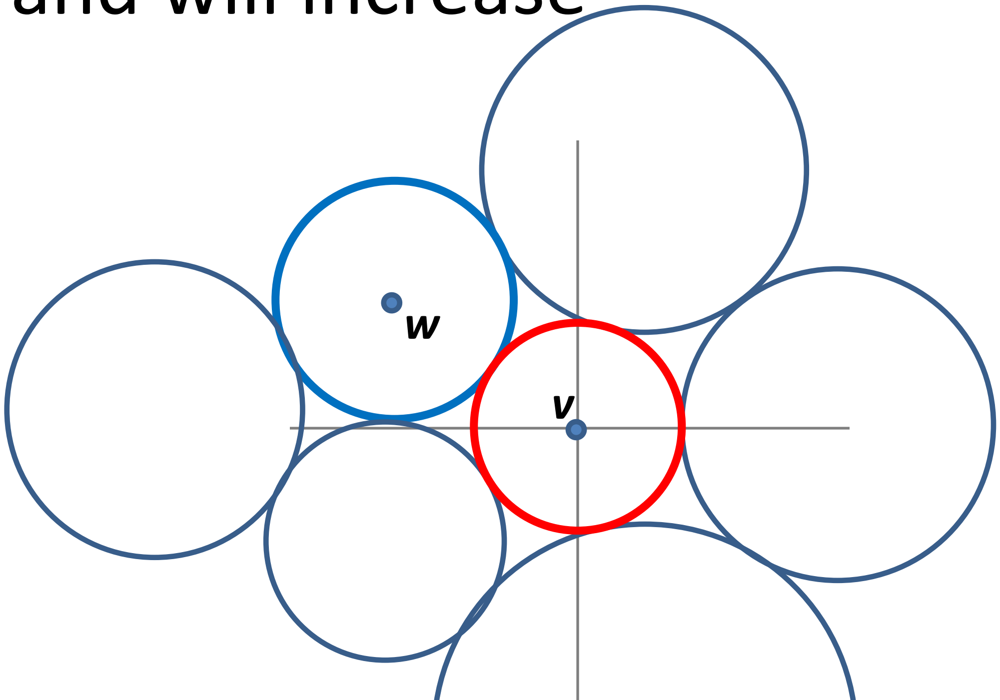


Circle Packing

<u>Proof of Convergence</u>:

Similarly, if *v* has a neighbor *w* which is on the boundary, increasing the radius at *v* will decrease the error at *v* and will increase the angle-sum at *w*...



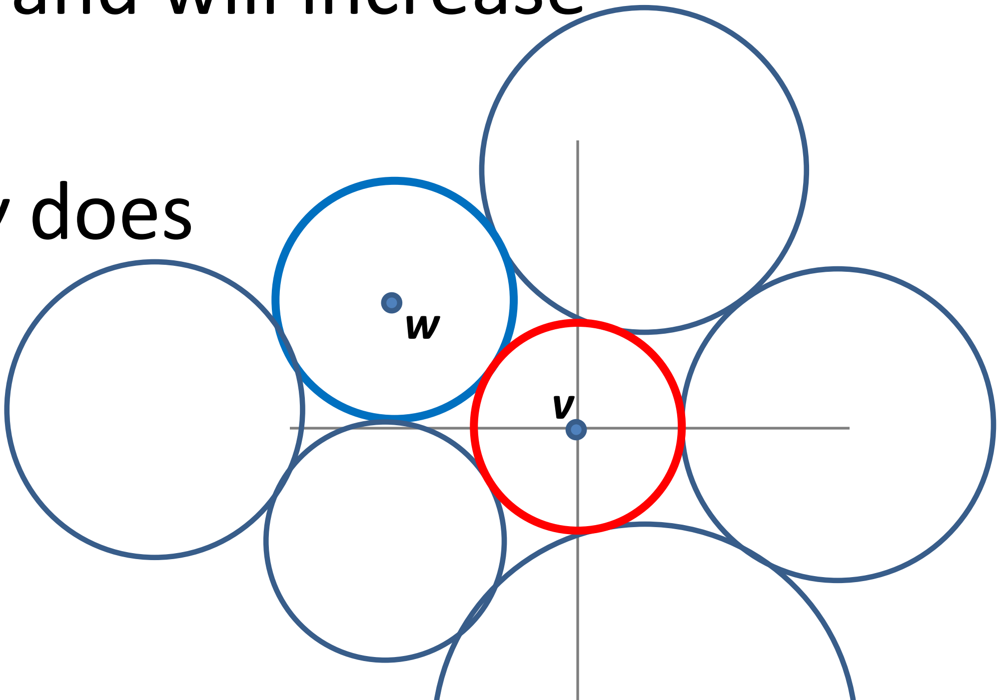


Circle Packing

<u>Proof of Convergence</u>:

Similarly, if *v* has a neighbor *w* which is on the boundary, increasing the radius at *v* will decrease the error at *v* and will increase the angle-sum at *w*...

But the angle-sum at *w* does not contribute to the error...



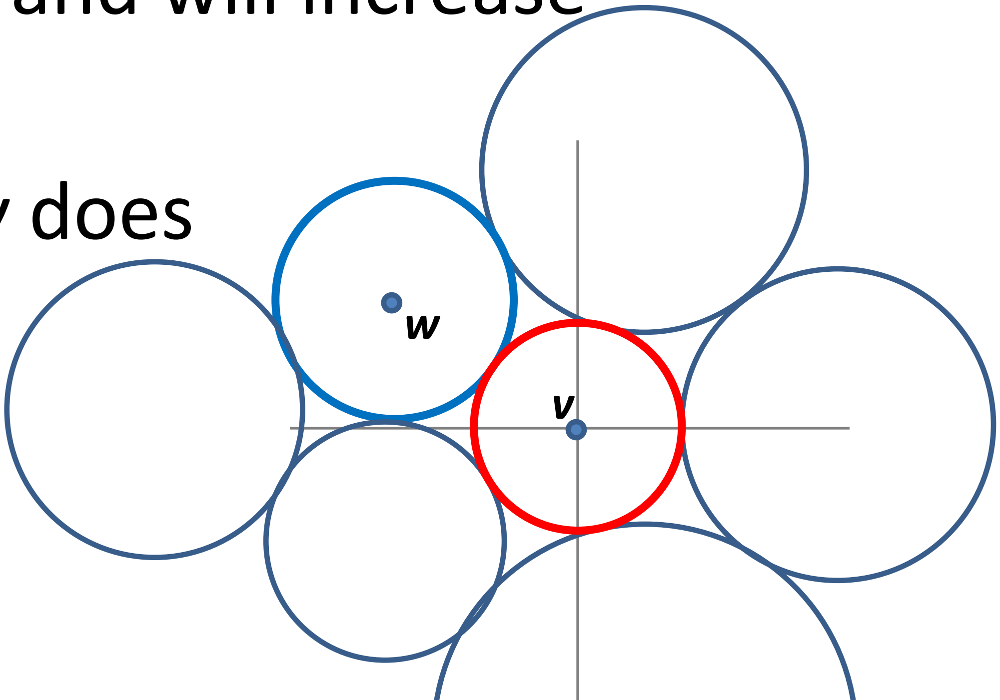
Circle Packing

<u>Proof of Convergence</u>:

Similarly, if *v* has a neighbor *w* which is on the boundary, increasing the radius at *v* will decrease the error at *v* and will increase the angle-sum at *w*…

But the angle-sum at *w* does not contribute to the error…

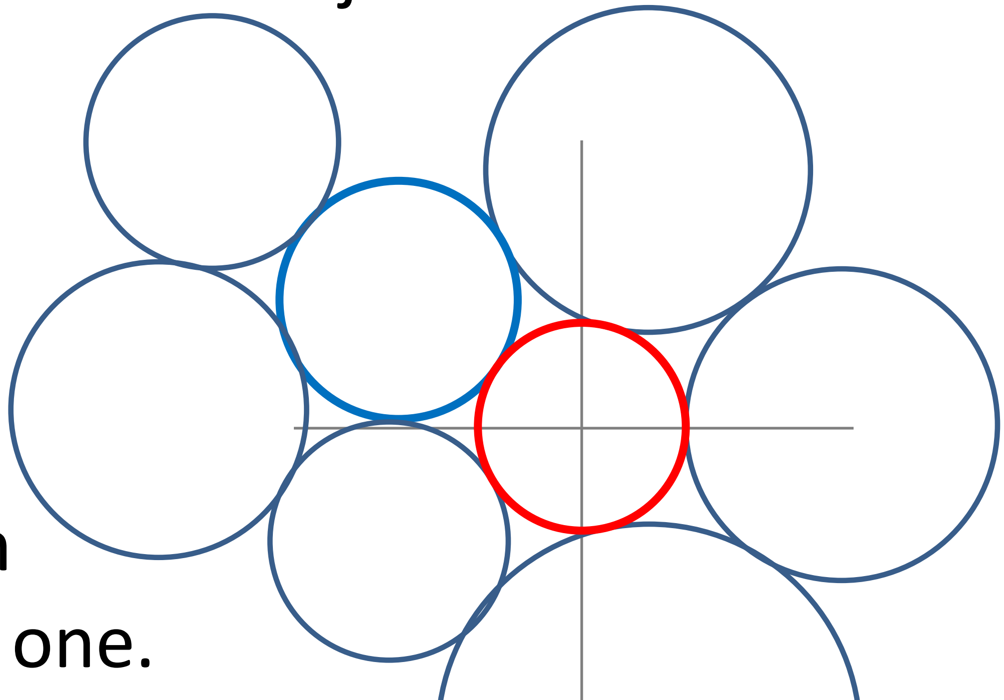
So the total error will decrease.



Circle Packing

Proof of Convergence:

If the angle-sum is never smaller than 2π and equals 2π at vertices adjacent to the boundary, increasing the radius of a vertex adjacent to a boundary-neighbor will not increase the error in the current iteration, but will ensure that there is a boundary-neighbor with angle sum not equal to 2π the next one.



Circle Packing

<u>Proof of Convergence</u>:

If the angle-sum is never smaller than 2π and equals 2π at vertices adjacent to the boundary, increasing the radius of a vertex adjacent to a boundary-neighbor will not increase the error in

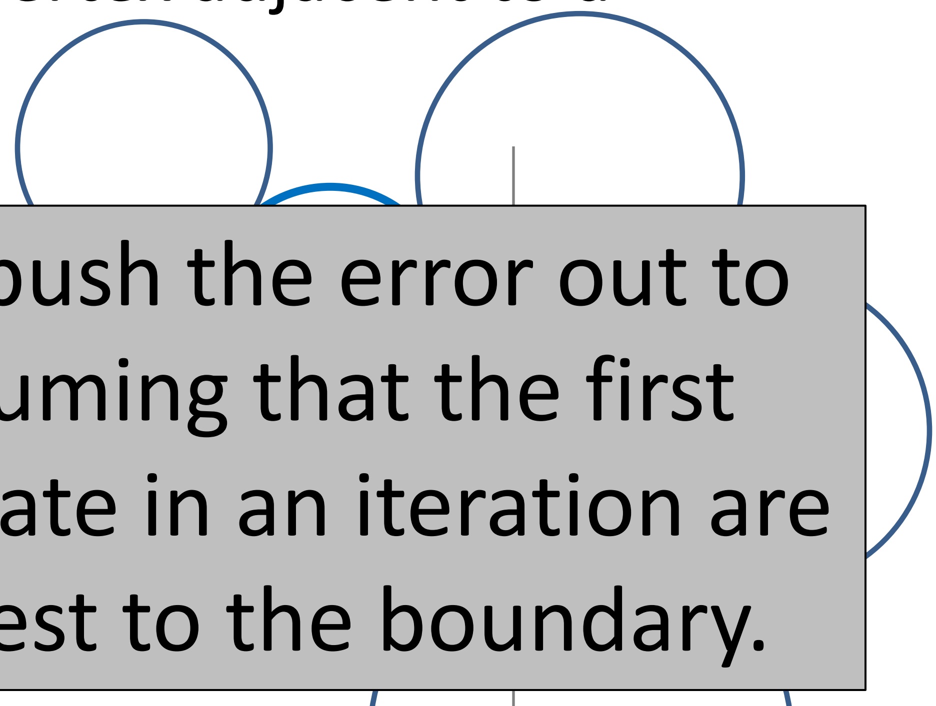
t

b

t

n

n



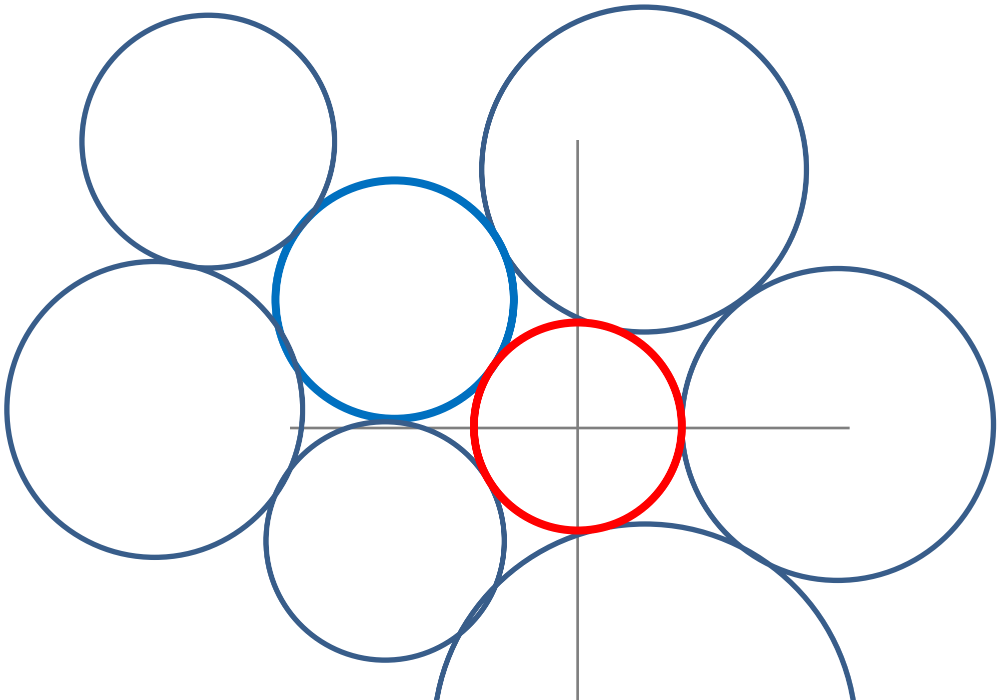
Repeating, we can push the error out to the boundary, assuming that the first vertices that we update in an iteration are those that are closest to the boundary.

Circle Packing

<u>Proof of Convergence</u>:

$$E = \sum_{v \in K^{\circ}} \left| \mathrm{AngleSum}(v) - 2\pi \right|$$

Since the error is never negative, and since it will decrease at every multi-iteration (if it's not already zero) then it has to converge.



Circle Packing

Note:

- This does not guarantee that the error will converge at zero (i.e. that we get a valid assignment of radii).

Circle Packing

Note:

- This does not guarantee that the error will converge at zero (i.e. that we get a valid assignment of radii).
- This doesn't tell us how to map into the unit disk, only how to satisfy the condition that boundary vertices have prescribed radii.

Applications to Surfaces

Observation:

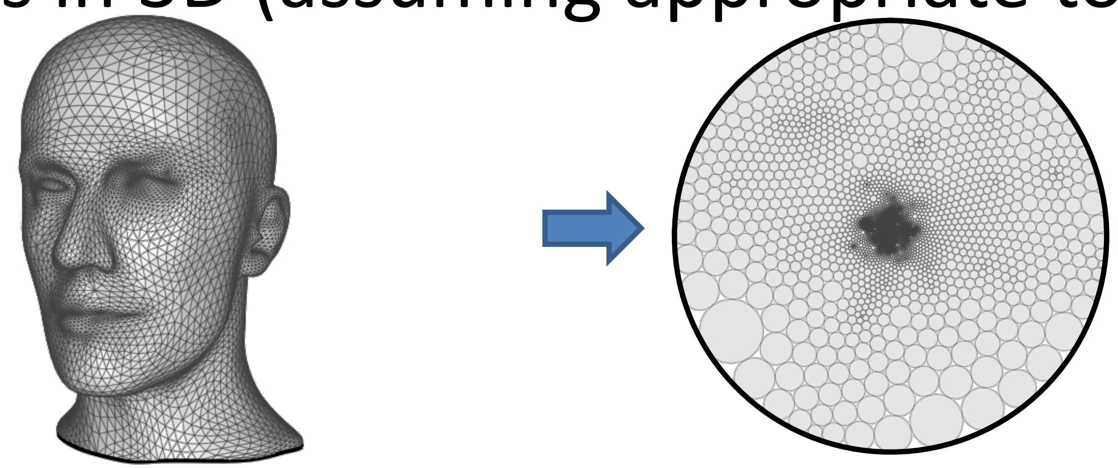
None of the above discussion required that the initial triangulation lived in the plane, only that it was a triangulation of a topological disk.

Applications to Surfaces

Note:

None of the above discussion required that the initial triangulation lived in the plane, only that it was a triangulation of a topological disk.

So we can apply the packing to triangulations of surfaces in 3D (assuming appropriate topology).

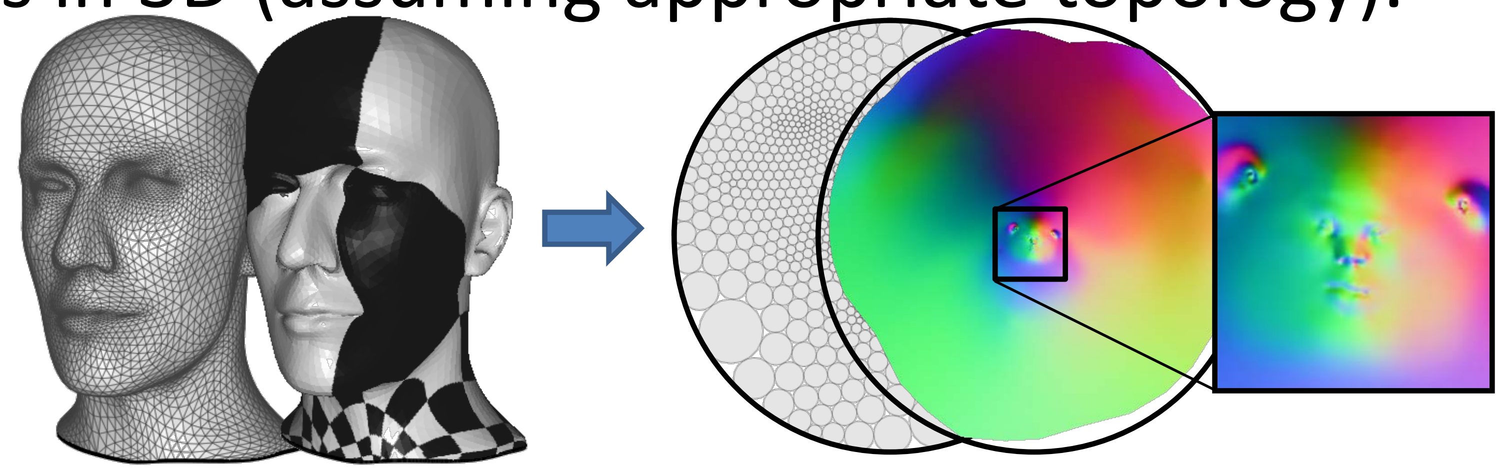


Applications to Surfaces

Note:

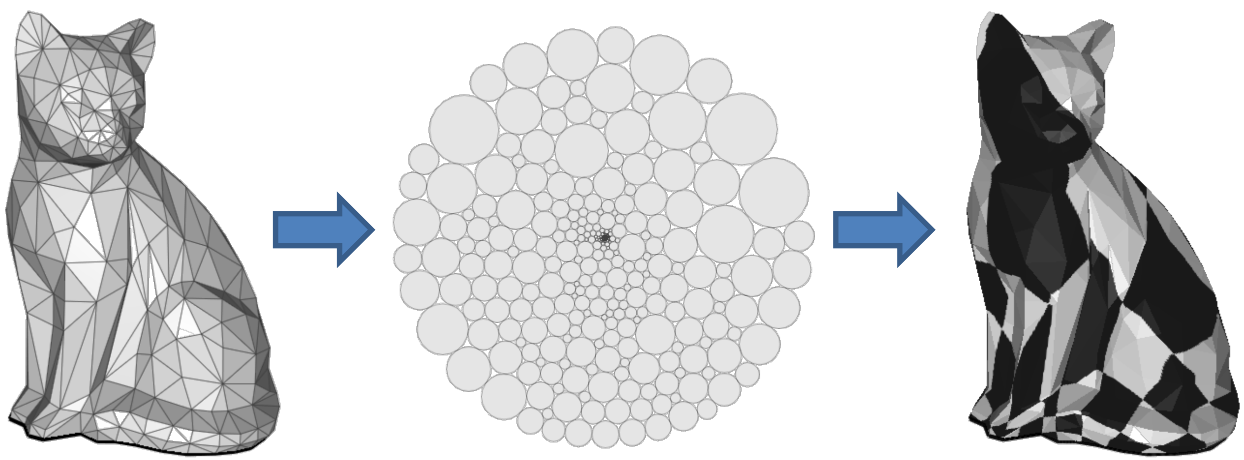
None of the above discussion required that the initial triangulation lived in the plane, only that it was a triangulation of a topological disk.

So we can apply the packing to triangulations of surfaces in 3D (assuming appropriate topology).



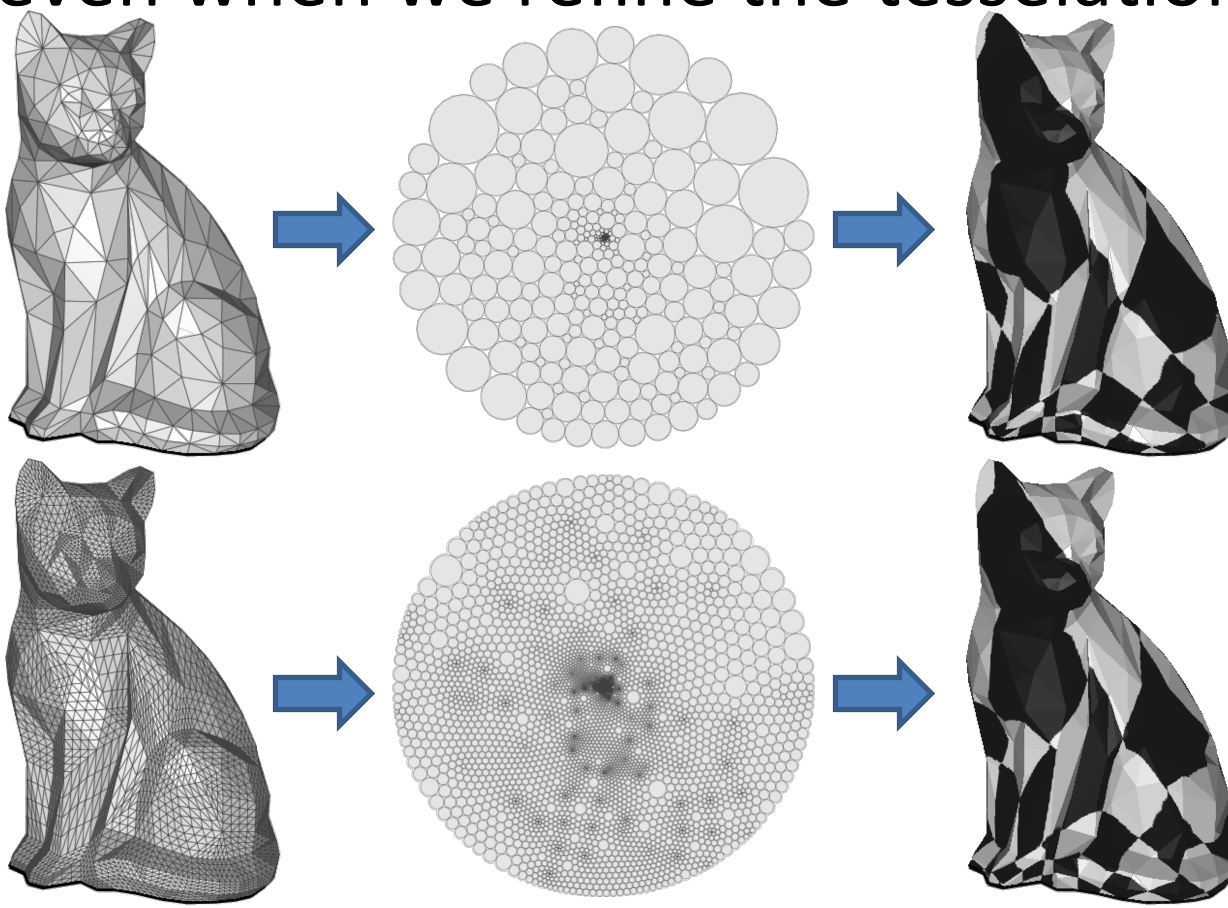
Applications to Surfaces

Q: So why is the map not conformal?



Applications to Surfaces

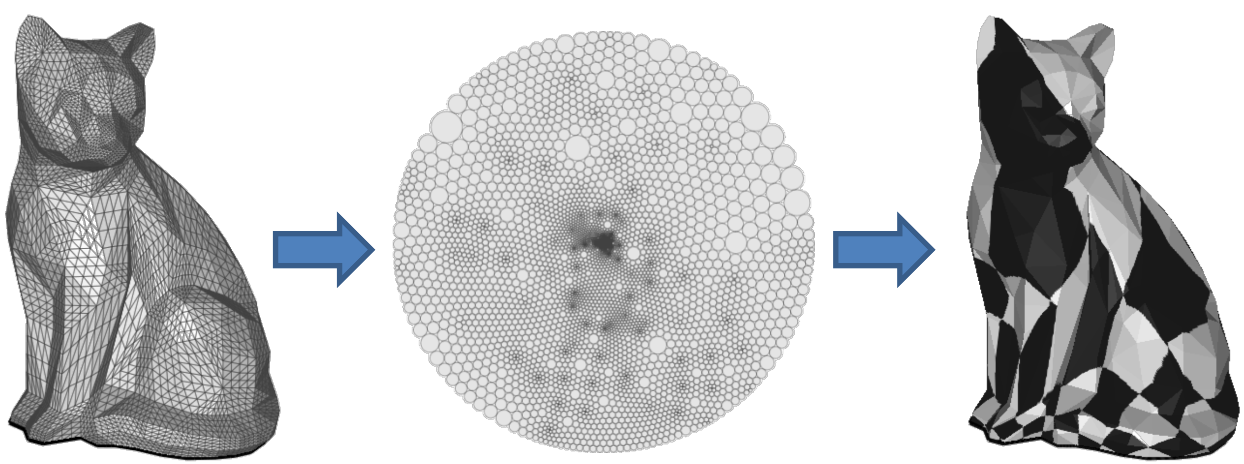
Q: So why is the map not conformal?
Not even when we refine the tesselation?



Applications to Surfaces

Q: So why is the map not conformal?
Not even when we refine the tesselation?

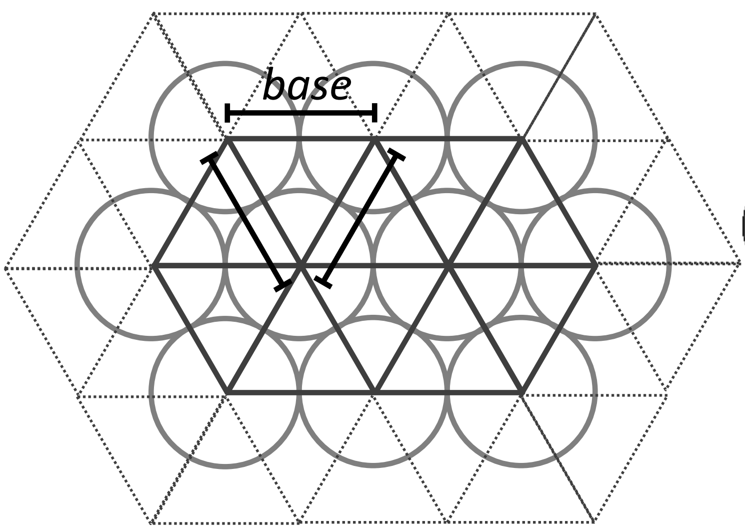
A: Because in computing the mapping, we never used information about the geometry of the mesh, only the topology of the triangulation.

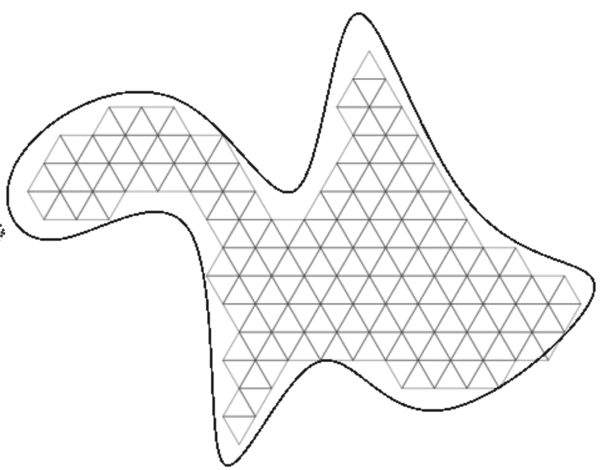


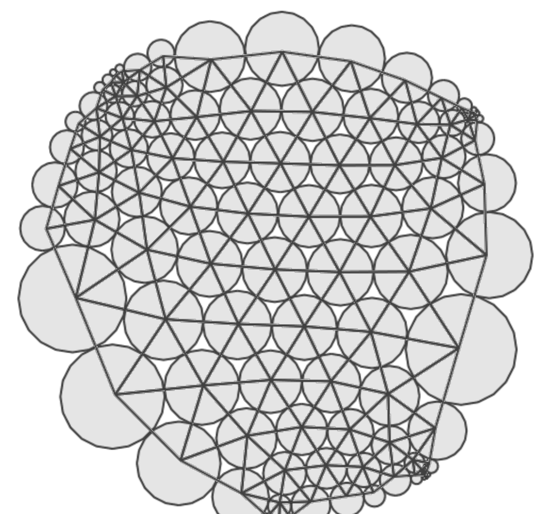
Applications to Surfaces

In the planar case things are OK since we were using a regular hexagonal lattice.

If we think of the vertices as circles with radius *base*/2, the length of the edge between two vertices equals the sum of the radii.

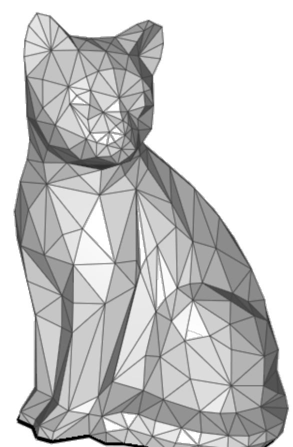


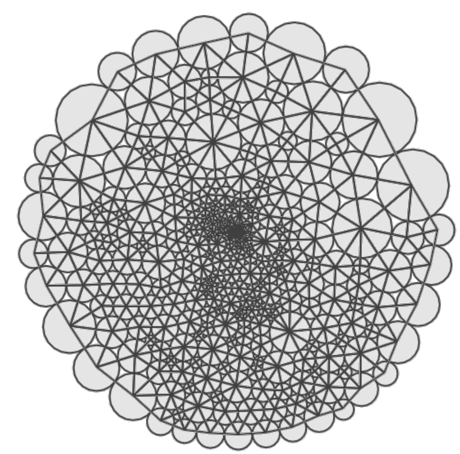




Applications to Surfaces

On a triangle mesh, we cannot assign a radius to each vertex so that length of the edge between two vertices equals the sum of the radii.





Applications to Surfaces

On a triangle mesh, we cannot assign a radius to each vertex so that length of the edge between two vertices equals the sum of the radii.

The number of edges is roughly three times the number of vertices, so we have more constraints than degrees of freedom, and the system cannot be solved (in general).