Euclidean Voronoi Diagrams for Circles in a Circle

D.-S. Kim, D. Kim and K. Sugihara

By Ian Livingston
Student #: 10386570
NSID: ijl708

1. INTRODUCTION

The purpose of this paper is to provide a detailed description of the paper *Euclidean Voronoi Diagrams for Circles in a Circle* by Kim et al[1]. The goal is to present this information in a manner which is as clear and straight forward as possible. Firstly, this paper will describe the problem and the solution outlined by Kim et al. In the solution, the various lemmas will be described in detail, as well as justifications as to why they work. Secondly, descriptions of some of the mathematical approaches used in the solution will be provided, as well as justification as to why these approaches work. Detailed proofs will not be presented for these lemmas and mathematical formulas as this information can be found in the original paper. Finally, an example will be presented to demonstrate how the algorithm can be used to find a solution.

2. PROBLEM

In class we studied in great detail what point Voronoi diagrams are, and various efficient algorithms to generate them. In a sense the concepts learned there apply directly to this problem. The problem of generating a circle Voronoi diagram in a circle is conceptually simple [Fig 1]. Just like in point Voronoi diagrams, bisecting lines are generated such that the distance between two circles is equal at the bisecting line. Also any Voronoi vertex between three bisecting lines will represent a point that is an equal distance from three circles. These circles are known as generators for the vertex.

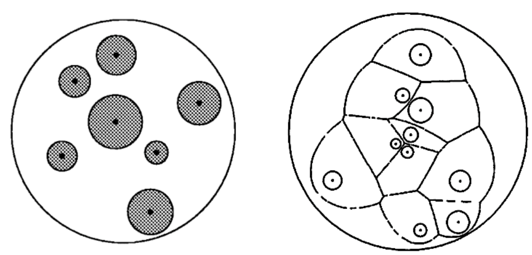


Fig 1: The goal is to turn the plot on the left into something similar to the image on the right. Note that the image on the right was not generated from the one on the left.

This problem is limited, since it is assumed that a vertex can only exist between three bisecting lines. While it is possible to provide a circle arrangement such that a vertex of four bisecting lines is generated; this algorithm assumes that this is not the case. Also, the bounding circle (the encloser) is considered to be a generator in this problem. As such bisecting lines are generated between the inner circles and the encloser that surrounds them. Finally, the inner circles do not intersect or touch each other, or the encloser.

In their paper Kim et al. state that there exists real practical applications for this problem. They suggest that the problem maybe key in cable designs for motor vehicles. By using this algorithm, it would be possible to minimize the material required for, and weight of the cables. Since the design of these cables can occur beforehand, it will help during vehicle's design phase since the radii of these cables could be precisely estimated.

[1] D.-S. Kim, D. Kim and K. Sugihara, Euclidean Voronoi Diagram for Circles in a Circle, *International Journal of Computational Geometry & Applications.* Vol. 15, No 2 (2005) Pg. 209-228

3. SOLUTION

Generating a normal first order Voronoi diagram can be done in O(n log n) time using Fortune's algorithm. Unfortunately, due to the complexity of this problem it is not possible to use Fortune's algorithm to generate a solution. The main problem is that this problem no longer uses single points in a plane to generate a Voronoi topology. In essence we can view the boundary of a circle as a series of points, or a curve. This means that our previous algorithms are not efficient enough to produce a solution.

In order to generate a solution quickly we must do a measure of preprocessing. The goal of the preprocessing is to develop a seed topology [Fig 2]. The seed topology

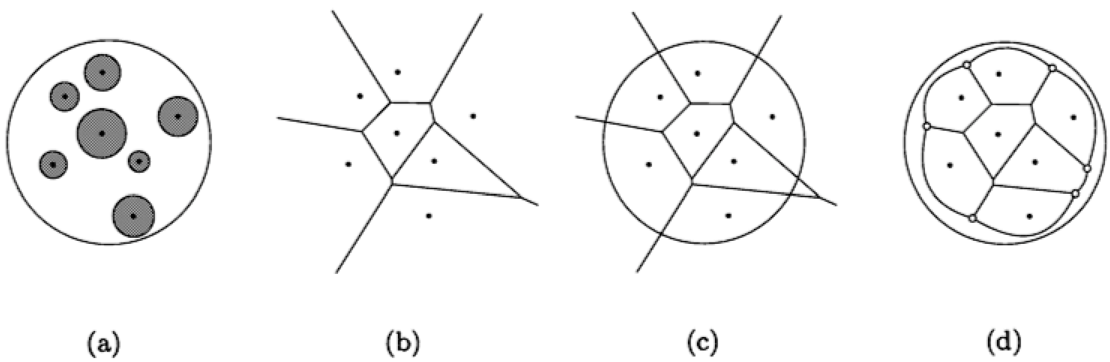


Fig 2: (a) The original circle diagram (b) The point Voronoi diagram generated using the circle centres (c) The point Voronoi diagram with encloser circle (d) Completed seed topology with new-born edges and vertices

will have a similar shape to that of the desired final topology. As was mentioned above, each vertex is associated with three generators in the same way that a point Voronoi diagram vertex, is associated with three points. It can also be said that if a vertex exists then there exists three edges. Each edge is associated with two generators (the edge bisects the generators), and must have a vertex at either end. This allows us to say that for each edge and its two vertices, there exists four generators, with three being used to generate each of the vertices with the remaining generator being the mate for that vertex [Fig 3].

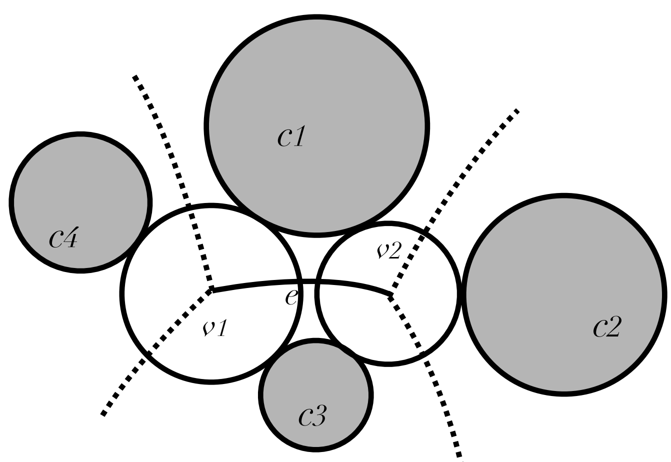


Fig 3: *c1, c2, c3* are the generators for *v2* with *c4* as a mate. *c1, c4, c3* generate *v2* with *c2* as a mate. This relationship is very important for the edge-flip operation in the algorithm.

The edges and vertices in the seed topology are points of interest to be examined during the creation of the final topology. The preprocessing, in essence, creates a collection of circle triples that can be examined during the edge-flip step of the algorithm.

3.1. The Algorithm

The process for generating a Voronoi diagram for circles in a circle is divided up into three steps. The first step is the generation of the seed topology. In this step the centre points of the circles are treated as points, in a 1st order point Voronoi diagram. The encloser is added to the point Voronoi diagram and new-born vertices are generated at intersection points. New-born edges are added between consecutive new-born vertices [Fig 3, Fig 4].

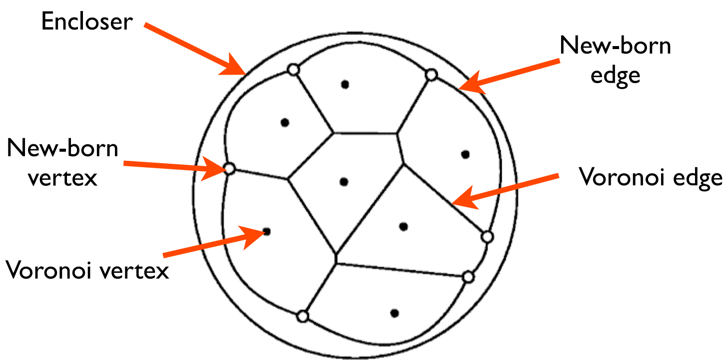


Fig 4: Labeled seed topology

The second step involves applying a series of edge-flip operations based on five lemmas. The final step is generating the line equations for the edges. These edges are stored in rational quadratic Bézier form. These equations can correctly represent the curved edges present in this problem. This step can be combined with the second step as correct vertices and edges are generated. Once these three steps have been completed, a correct solution will have been generated. In the next section the lemmas and tangent circles used in the edge-flip step will be described. Also, the process for generating and storing edges shall be discussed.

4. DESCRIPTION

4.1. Apollonius' Circles

In the problem of finding the position of Voronoi vertices for circle Voronoi diagrams, the tangent circles turn out to be Apollonius circles. The process of finding the tangent circle is equivalent to Apollonius' tenth problem. There are three combinations of tangent circles created from three generators [Fig 5].

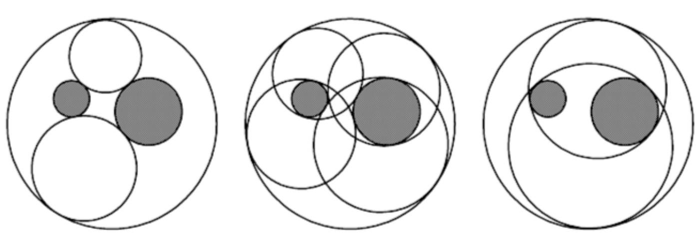


Fig 5: Case 1, circles are tangent but empty. Case 2 and 3 are tangent but not empty.

For circle Voronoi diagrams, like point Voronoi diagrams, we are only interested in the case where the tangent circles are empty. Kim et al. have shown in another paper that it is possible to generate these tangents via Möbius transformation[2]. To find the tangent circles, the common tangent lines for two appropriately mapped circles is computed and then the Möbius transformation is used to compute the tangents.

4.2. Möbius transformation

Computing the Apollonius circles can be efficiently done using a Möbius transformation in the complex plane. This paper will not describe, in-depth, the mathematics required to compute these circles since they are discussed in detail in the parent paper. The material is not easily paraphrased. The process involves applying four properties of the Möbius transformation, such that a Apollonius circle is generated with the desired tangent characteristics. This process includes a step where the smallest of the three generators is shrunk in size until it is a single point. The other generators are also shrunk by the radius of the smallest generator. If the encloser is a generator, then it's radius is increased by the shrunken circles radius. When the tangent circle is computed, it's centre point will be in the correct location, but it's radius will be to large to the order of a single radius of the shrunken circle. The correct tangent is found by shrinking the tangent circle by the correct amount. This description is illustrated in Figure 6.

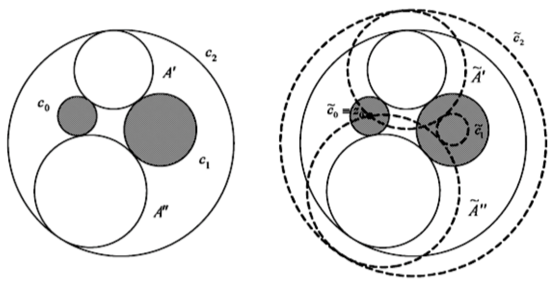


Fig 6: c_0 is shrunk as is c_1 the tangent circles are expanded as is the encloser. Positional information remains the same.

4.3. Lemmas

4.3.1. *If the tangent circles of both end vertices are valid, then the edge should not be flipped [Fig 7].*

In Figure 7, we can see that the tangent circles, A1 and A2, do not intersect with their mate circles. Since they do not intersect we can infer that the edge in the solution will have the same shape as the one generated in the point Voronoi diagram, from the seed topology (dotted line). It can also be said that since the tangent circle does not intersect any other circles, then it is known that its three generators are the closest neighbors.

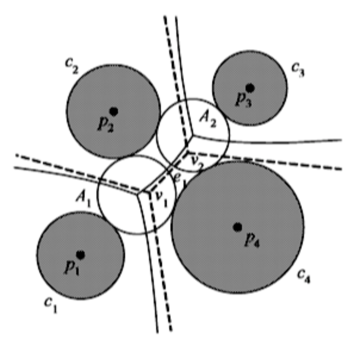


Fig 7: Tangent circles do not intersect their mates

4.3.2. *If both tangent circles exist and each circle intersects its mate, then the edge should be flipped [Fig 8].*

[2] D.-S. Kim, D. Kim and K. Sugihara, Voronoi diagram of circle set from Voronoi diagram of a point set: II. Geometry, *Comput. Aided Geom. Des.* 18 (2001)

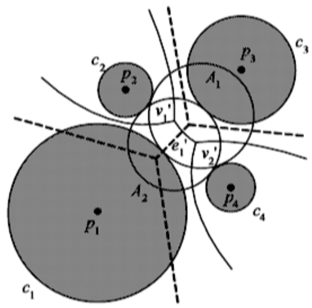


Fig 8: Tangent circles intersect their mates

In the case shown in Figure 8, we see that both tangent circles intersect with their mates. As such it can be said that the three generators for the tangent circles A1 and A2, are not the closest neighbors to their tangent circles. In this case the edge *e,* must be flipped perpendicular to the original point Voronoi edge. The flipped edge, e_1', can be seen perpendicular to the dotted line in Figure 8. Also note that the new vertices v_1' and v_2' are the centre points of new tangent circles created by circles c_1, c_2, c_3 and c_3, c_4, c_1. These new tangent circles do not intersect their mates and therefore conform to the first lemma.

4.3.3. If both tangent circles exist and only one of the two tangent circles intersects its mate, then the edge should not be flipped.

The difficulty in this case is that each tangent circle suggests a different course of action. The tangent circle that intersects it mate would suggest that the edge should be flipped, while the tangent circle that is valid suggests that the edge should remain. In this case it is not possible to determine the correct action. This issue is resolved not by flipping the current edge, but rather one of the incident edges connected to one of the vertices. Since this new flip will be resolved in a later step we can safely leave the current edge in it current form.

4.3.4. If one tangent circle exists and the circle intersects its mate, then the edge should be flipped.

Since the seed topology is generated from a point Voronoi diagram, it is possible that there does not exist a tangent circle in the circle voronoi diagram [Fig 9, Fig 10]. In this case we have an edge with only a single tangent circle. The vertex v_1 that does not have a tangent circle, would suggest that v_1 should be removed from the final solution. However, this might not be the case as some of the incident edges to v_1 might still be relevant. It turns out that we can solve this problem with edge-flip operations based on the remaining tangent circle for v_2. Similar to the Second lemma, if the remaining tangent circle for an edge intersects its mate, the edge should be flipped. In Figure 9, we can see that the tangent circles presented for p_1, p_3, c_2 are invalid because they inclose c_2. Therefore, no tangent circle exists for vertex v_1. If we draw a tangent circle for the points p_1, p_3, p_4, we can see that the circle intersects its mate c_2. The result is that e_1 should be flipped. We can see that this must be the case by inspecting the diagram.

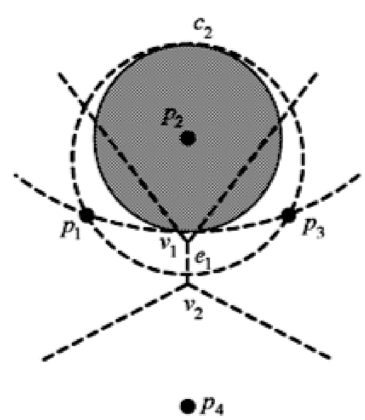


Fig 9: Only the tangent circle for v_2 exists and it intersects its mate c_2

4.3.5. If only one tangent circle circle exists and the circle is valid, then the edge should not be flipped.

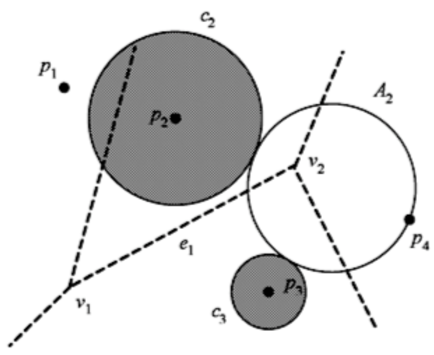


Fig 10: Only the tangent circle for v_2 exists but it does not intersects its mate p_1

This case is identical to the 4th except that the tangent circle does not intersect it mate. In this case there is no edge-flip. This is not to suggest that the edge does not require flipping, rather we must wait for an incident edge to be flipped. By inspecting Figure 10, we can see that no tangent circle exists for v_1 but it is also known that edge e_1 must exist, again by inspection. It can be seen that there must exist an edge between p_1 and c_2. Thus, it's then inferred that there must be an fourth generator along with p_1, c_2, and c_3. Since we know no tangent circle exists between p_1, c_2, and c_3, we can say that there must be a c_i such that p_1, c_2, c_i form a tangent circle, with mate c_3 and another tangent circle for c_3, c_2, c_i. Therefore, an edge flip must occur in this case as expected.

4.4. Edge Representation

The edges in the circle Voronoi diagram are conic. This means that the edges can be either linear, elliptic or hyperbolic. A edge that exists between two inner circles can only be linear or hyperbolic. If the circles are the same size then the Voronoi edge between them will be linear. Otherwise the edge will curve hyperbolically towards the smaller of the two inner circles. An edge that is generated by an inner circle and the encloser will always be elliptic is shape.

It is known that a conic edge can be represented in rational quadratic Bézier form. In order to be able to compute the Bézier curve five parameters are required.

1. The positions of both end points of the curve

2. The tangent lines at both end points, tangent to the curve

3. An arbitrary point that the curve passes through

The end points of the curves are the centre points of the tangent circles that were generated by the Möbius transformation. The tangent lines can be cleverly computed by bisecting the angle created by lines that run perpendicular to the contact points between the generators. This can be clearly seen in Figure 11. The final parameter is a passing point which can be found by finding an equidistance point between the two generators.

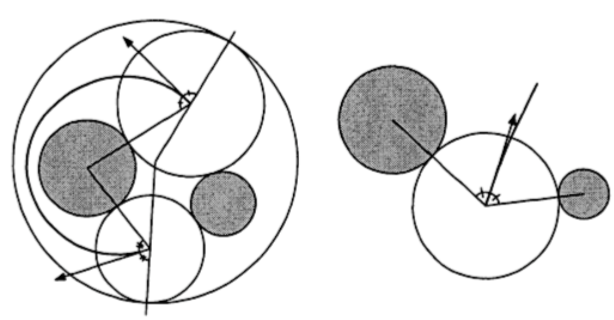
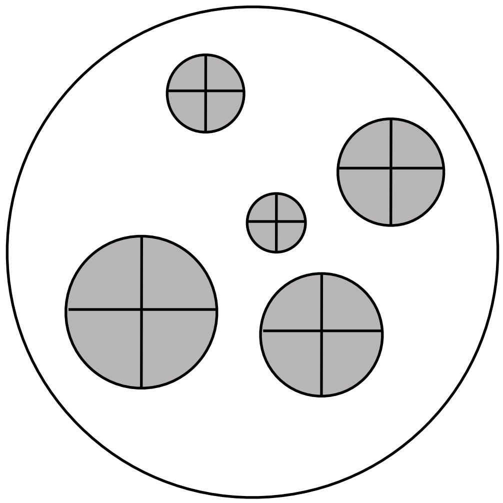


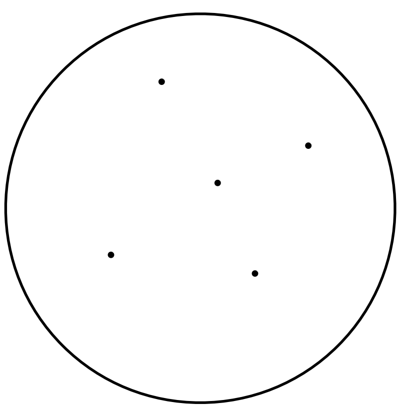
Fig 11: Two examples where tangents have been generated by the angle computed from the lines that run perpendicular to the contact points between the tangent circles and the generators.

4. EXAMPLE

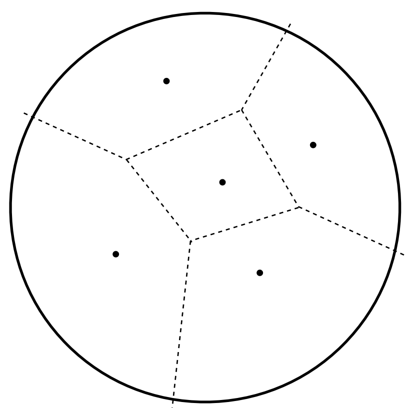
The problem does not lend itself well to manual generation due to the level of math that is required. Instead, this example will focus on demonstrating the lemmas and the generation of a correct topology in a visual way. In the first diagram there are five inner circles of different sizes and arrangement and they are contained within a circle encloser.



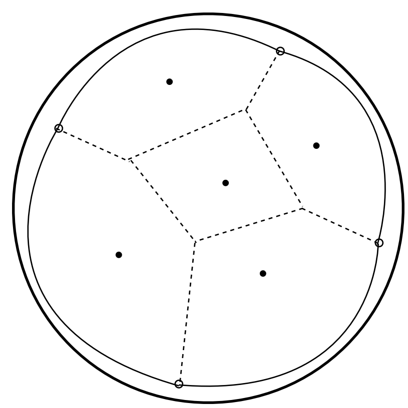
From this we must generate a seed topology. To do this a point Voronoi diagram must be generated based on the centre points of the circle. The next set of images demonstrates this. Any standard point Voronoi algorithm will work, for example, Fortune's plane sweep algorithm.



Centre points of inner circles



Point Voronoi diagram

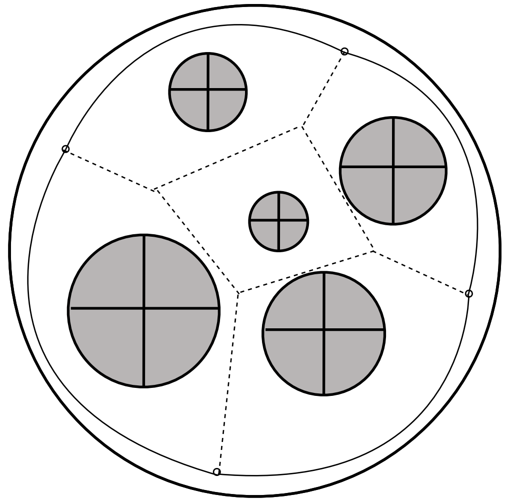


Point Voronoi with new-born edges and vertices for encloser

The final image is the final seed topology that will be used for the edge flip operations.

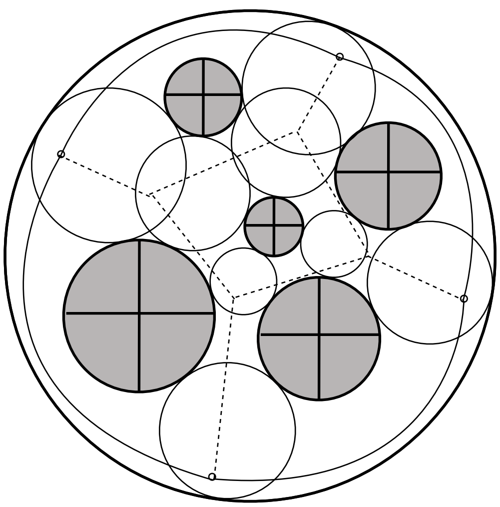
Next, the inner circles are added back to the seed topology and the edge flip operations are applied.

Now the tangent circles are added (below). These tangents are generated based on the point Voronoi vertices from the seed topology. In this example the tangents will be added based on observation. Normally, the tangent circles would be generated via Möbius transformation as described by Kim et al, but that method requires *(x,y)* point information, which is not present for this example.



Seed topology with inner circles added

Below is an image with the appropriate tangent circles added. With the tangent circles in place we can begin applying the edge-flip lemmas to generate the correct topology. In the description above it was stated that the edge-flip step and the final step could be done at the same time as the topology was corrected. In this example the steps are separated so that it is clear how each step works.

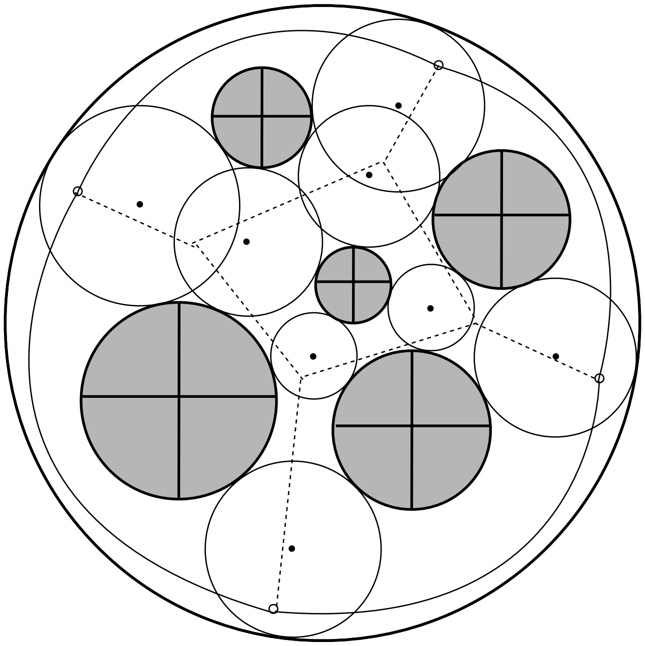


Seed topology tangent circles. tangents generated based on point Voronoi diagram vertices

In the same way that tangent circle centre points represent Voronoi vertices, the centres of these tangent circles represent Voronoi vertices for the circle Voronoi diagram. It is now possible to apply the edge flip lemmas. For this example, due to bad judgement when designing the original diagram, the only lemma that applies is the first one.

If the tangent circles of both end vertices are valid, then the edge should not be flipped.

By inspection it can be seen that none of the tangent circles intersect their mates. Therefore, all the edges in this example turn out to be in the correct orientation. Intuitively this makes sense and will be confirmed when the circle Voronoi edges are generated.

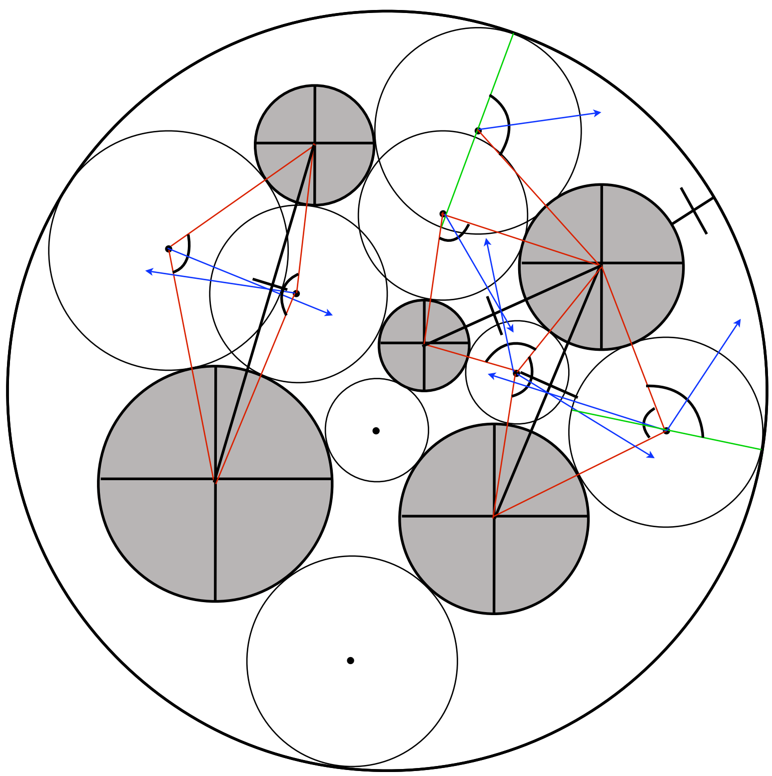


Centre points of the tangent circles represent the circle Voronoi vertices

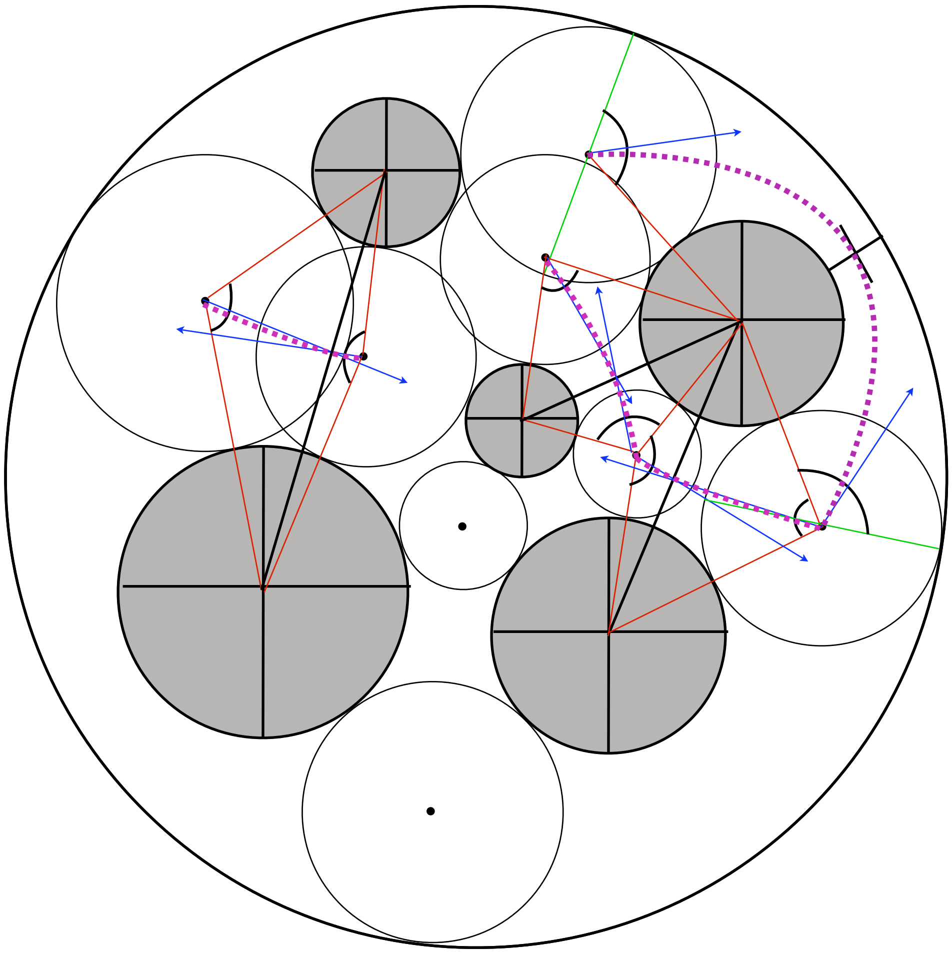
Once the centre points for the tangent circles are found, it is possible to start finding the Bézier curve form edges. In the description it was stated that five parameters are required to calculate the Bézier curve line equation. This example does not use specific coordinates. Therefore, some geometric rules must be applied to show that the parameters can, indeed be found. To illustrate this clearly, the geometric rules will only be applied to a subsection of the example.

The first two parameters are already present. The start and end points of the line correspond to the centres of the tangent circles for that edge. To find the tangents we add line segments connecting the generator and tangent centres (red). The bisectors of these angles created by these line segments are the tangents (blue) that are required for the Bézier curve.

In the following diagram the geometric relationship has been added. The red lines connect the centres of inner circles and tangent circles. The blue arrows bisect the angle created by the connected red lines. These arrows are the tangents to the Bézier curve. The green lines bisect the tangent circles based on their contact point with the encloser. Finally, there are a black lines that connect the centres of the inner circles together. There are also small black lines to represent the equidistance points. Due to the nature of circles, we know that the black line connecting the centres of



Generating the five parameters for the Bézier curve.

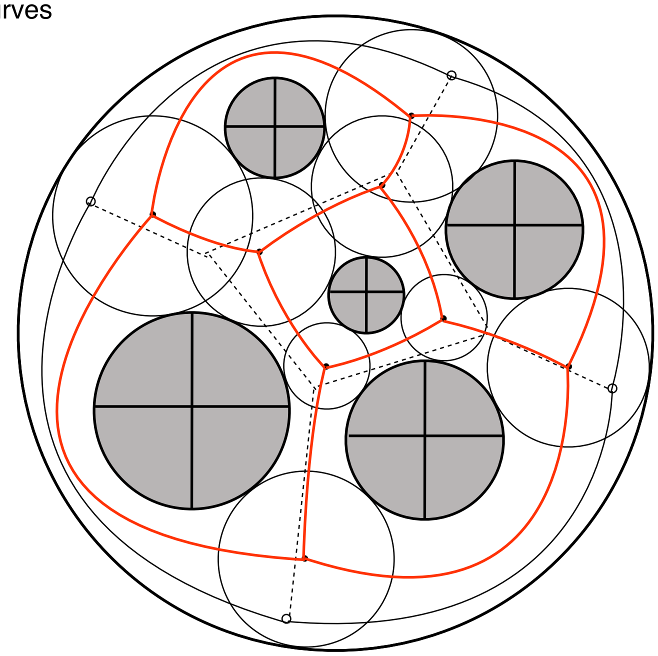


inner circles will pass through the shortest point between the circles. We know that the point represented by the smallest equidistance must exist on the Bézier curve. This point can be used as the passing point, which is the final parameter required to generate the edges.

Adding the Bézier curves

In the above diagram the Bézier curves are shown in purple. This demonstrates how the five parameters are used to produce a correct edge.

Finally, we can create the correct and complete circle Voronoi diagram. In this final diagram the original seed and final topologies are combined. In this example it can be clearly seen that the final topology matches the edge orientation of the seed topology, but generally this may not be the case.



Combined topologies. Red is the final solution.