FINAL PROJCET
A CIRCLE PACKING ALGORITHM IN 2D

MORAN KIM

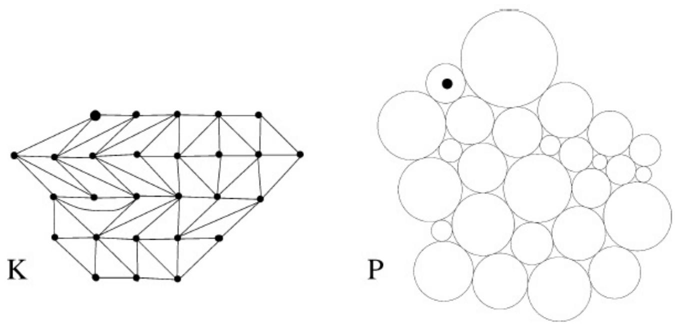
DEPARTMENT OF MATHEMATICS, EWHA WOMANS UNIVERSITY, SOUTH KOREA
E-mail address: moran.kim.9@gmail.com

1. CONTRACT

In my final project, I want to do implementing a circle packing in 2 dimension. I studied about it when I undergraduate student. But I did not implemented it myself. I want to try it as my final project. And want to deepen my understanding about the proof of convergence of a method, used to construct a circle packing. I will follow the paper [A circle packing algorithm] written by Charles R. Collins and Kenneth Stephenson. This paper published at "Computational Geometry" in 2003. In this paper they introduced the "Uniform Neighbor Method" to get a circle packing in 2D. The proof of the convergence using UNM is not fully described. Therefore, I want to set my goal of this projcet as, 1) Summarizing the algorithm(20 %), 2) Implementation for circle packing algorithm using UNM method(50%), 3) Prove the convergence(20%). and if necessary(if possible)(10%), I will add a contribution to this UNM method.

2. SUMMARY OF A CIRCLE PACKING ALGORITHM

A circle packing algorithm is a configuration of circles realizing a specified pattern of tangencies. Packing combinatorics are encoded in abstract simplicial 2-complexes K which triangulate oriented topological surfaces. This algorithm is restricted to the case in which K is a finite triangulation of a closed topological disc.

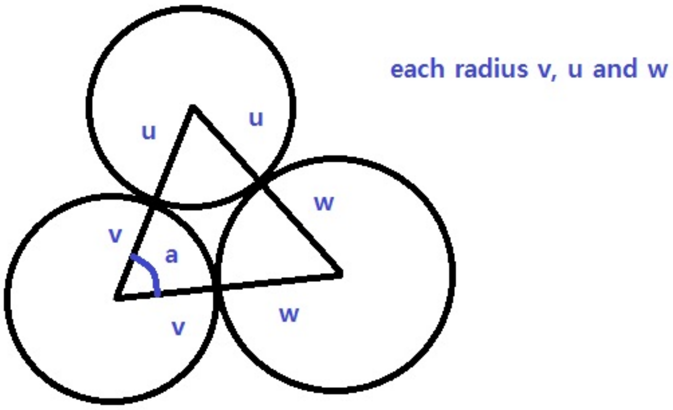


$[Complexes]$
The vertices of K are of two types, interior and boundary. If we call a neighboring circle as a petal, when v is interior, the list of petals is closed. i.e. the circle with center v is surrounded by the petals if v is an interior vertex.

$[Packings]$
A configuration P of circles in the euclidean plane is a circle packing for K if it has a circle C_v associated with each vertex v of K so that the following conditions hold: (1) if there is a neighboring edge between u and v, then C_u and C_v are mutually tangent, and if (2) $< u, v, w >$ is a positively oriented face of K, then $< C_u, C_v, C_w >$ is a positively oriented triple of mutually tangent circles.

$[Labels]$
A labe(putative radius)l for K can be thought of as a function $R : K^{(0)} \to (0, \infty)$ assigning a positive value to each vertex of K.

$[AngleSums]$
Given labels$x, y, z \in (0, \infty)$, lay out a mutually tangent triple $< c_x, c_y, c_z >$ of circles in the plane with radii x, y, z and connect the circle centers to form a triangle T. The angle of T at the center of C_x, denoted by $\alpha(x; y, z)$, can be computed from the labels using the law of sines.



$a(v; u, w) = 2\sin^{-1}(\sqrt{\frac{u}{v+u}\frac{w}{v+w}})$
A set of circles $C_v, C_{v1}, ..., C_{vk}$ with labels from R, which is circle packing label, will fit together coherently in the plane if and only if $\theta(v; R) = 2\pi n$ for some integer $n \geq 1$.

Given a complex K, a label R is said to satisfy the packing condition at an interior vertex $v \in R$ if $\theta(v; R) = 2\pi n$ for some integer $n \geq 1$. The label R is said to be a packing label if the packing condition is satisfied at every interior vertex.

In this paper it assumes with boundary conditions(appropriate labels for boundary vertices) a circle packing exists uniquely.

$[The Uniform Neighbor Model (UNM)]$

Focusing on the flower for v, we treat the label r as a variable, and the petal labels $r_1, ..., r_k$ as fixed parameters. For a given value $r = r_0$, the associated "reference" label is the number $\hat{r}$ for which the following equality holds:

$$\theta(r_0; r_1, ..., r_k) = \theta(r_0; \underbrace{\hat{r}, ..., \hat{r}}_{\text{k}}) =: \hat{\theta}(r_0; \hat{r}) \quad .$$

In other words, laying out a flower with petal circles of the uniform radius $\hat{r}$ would yield the same angle sum as with the original petal radii $r_1, ..., r_k$ when the center circle has radius r_0. Using the UNM requires two steps. First, given a value for v, determine $\hat{v}$ so that $\hat{\theta}(v; \hat{v}) = \theta(v; \{v_j\})$. Second, solve for a new value for v(call it u) so that $\hat{\theta}(u; \hat{v}) = A(v)$. where $A(v)$ represent the goal angle sum at the vertex v. In our case the $A(v) = 2\pi$ for every interior vertex v. The advantage of UNM methods is that these equations can be solved explicitly as follows.

using the equation above,

$$a(v; u, w) = 2\sin^{-1}\left(\sqrt{\frac{u}{v+u}\frac{w}{v+w}}\right)$$

(1) Given a value for v -> determine $\hat{v}$ so $\hat{\theta}(v; \hat{v}) = A(v)$.

let $\beta = \sin(\frac{\theta}{2k})$

$\rightarrow$ **Here** $\hat{v}$ **is computed from the equation** $\theta = \hat{\theta}$.

Then $\theta = k\ \alpha$ **so that by substituting** $\alpha = \frac{\theta}{k}$.

$$\sin(\frac{\theta}{2k}) = \frac{\hat{v}}{v+\hat{v}} \quad \text{->} \quad \boxed{\hat{v} = \frac{\beta}{1-\beta}v}$$

(2) Find u such that $\hat{\theta}(u; \hat{v}) = A(v)$

Let $\delta = \sin\frac{A}{2k}$

$\rightarrow$ **u should satisfy the equation** $\hat{\theta}(u; \hat{v}) = A(v)$

i.e. $\alpha = \frac{A}{k}$

$$\sin(\frac{\alpha}{2}) = \sin(\frac{A}{2k}) = \sqrt{\frac{\hat{v}}{v+\hat{v}}\frac{\hat{v}}{v+\hat{v}}} = \frac{\hat{v}}{v+\hat{v}}$$

$$\text{->} \quad \boxed{u = \frac{(1-\delta)\hat{v}}{\delta}}$$

3. Proof of the Convergence of Uniform Neighbor Model method

•Notation

For a label R, define "excess" e at an interior vertex v and the "total error" E by

$$e(v) = \theta(v; R) - A(v), \quad E = E(R) = \sum_{v:interior} |e(v)|$$

Lemma 1. *Let* $\theta(r) = \theta(r; r_1, ..., r_k)$ *and* $\hat{\theta}(r) = \theta(r; \hat{r}) = \theta(r; \underbrace{\hat{r}, ..., \hat{r_k}}_{k})$ *with* $\hat{r}$ *chosen so that* $\theta(r_0) = \hat{\theta}(r_0)$ *for some* $r_0 > 0$. *Assuming labels* $r_1, ..., r_k$ *are not all equal, then*

$$\frac{d\hat{\theta}}{dr}(r_0) < \frac{d\theta}{dr}(r_0),$$

Moreover, $\theta(r) < \hat{\theta}(r)$ *for* $0 < r < r_0$ *and* $\theta(r) > \hat{\theta}(r)$ *for* $r > r_0$.

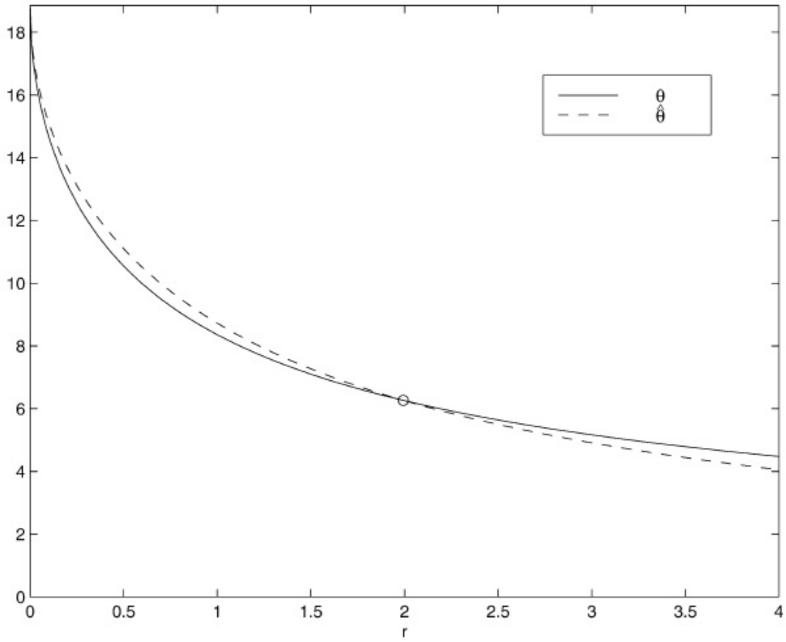
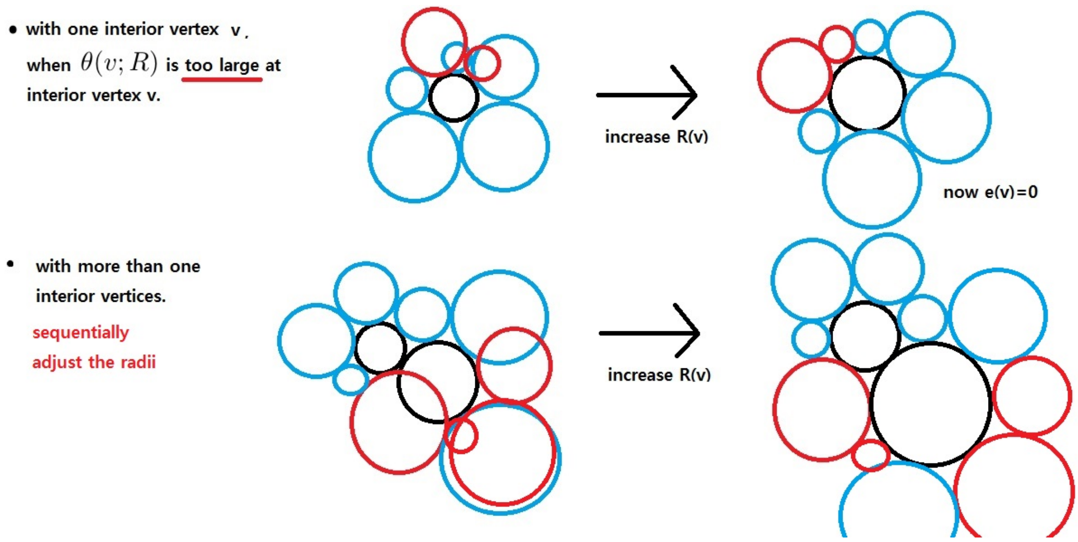


Fig. 4. Angle sums of the original and reference flowers.

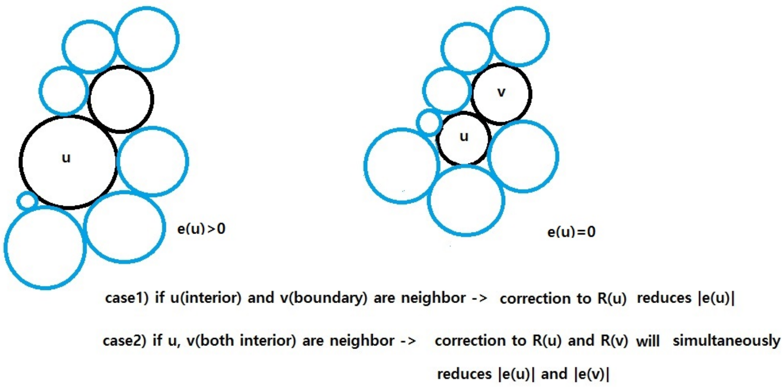
Lemma 2. *E is monotone decreasing with UNM label correction.*

Proof. Let F denote the number of faces of K. Each has three angles which sum to π. The total angle is $\sum_{v \in K} \theta(v; R) = F\pi$, independent of R. Thus the total angle is a conserved quantity and so any adjustment of a label simply causes a redistribution of that angle among the vertices.

(1) $e(v) > 0$; Suppose that $\theta(v; R)$ is too large at some interior v, so $e(v) > 0$. Let us increase the label $R(v)$ until $e(v) = 0$. Since we can sequentially correct the labels. At worst, E remains unchanged.



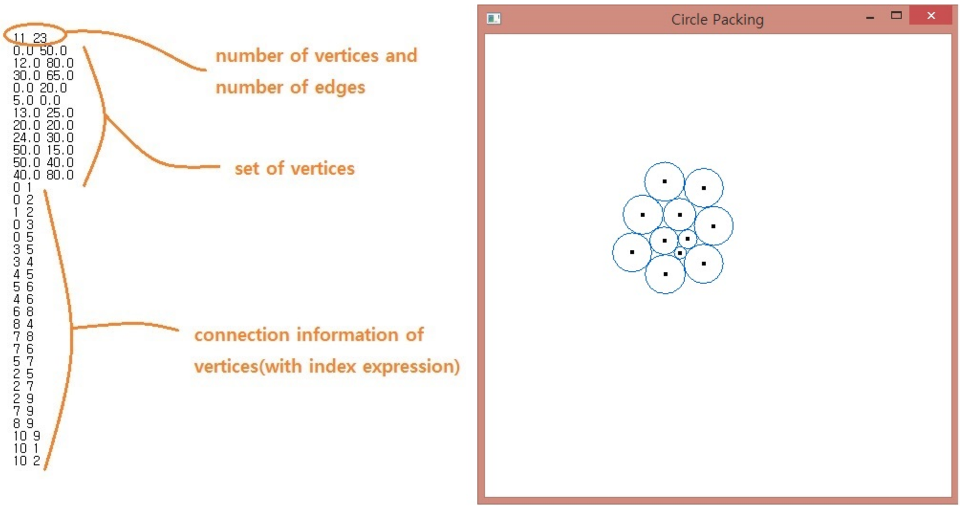
If u and v are neighboring vertices and u is an interior vertex or u is a boundary vertex, then the correction to $R(v)$ simultaneously reduces $|e(u)|$ and $|e(v)|$, and so E decreases.



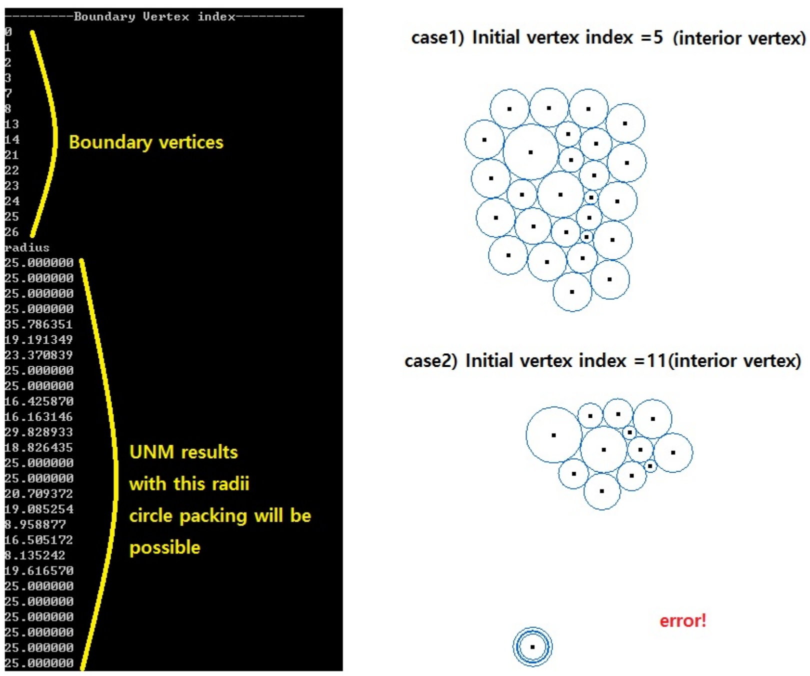
For $e(v) < 0$, we can simply apply same logic. Since we defined the error as sums of the differece of the current angle sum and the target angle sum(in our case, target angle sum=2π), If we do correction to $R(v)$ with UNM method then e(v) can only be decreased. Therefore E cannot increase. $\square$

4. Limitations in the Implementation

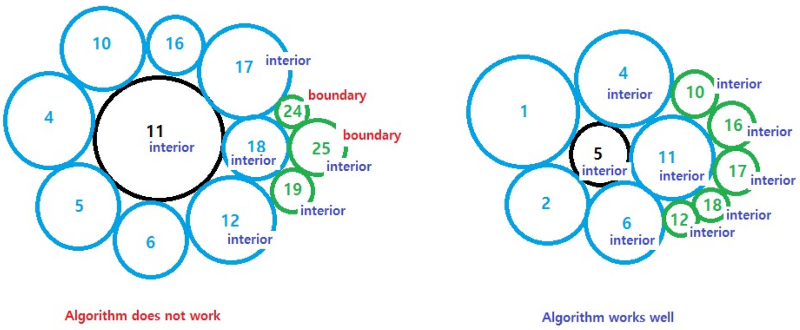
I constructed a Circle Packing algorithm with Uniform Neighbor Model[1]. In my program the inputs are set of vertices and the connection information between two vertices. Then the program outputs a configuration of circle packing P.



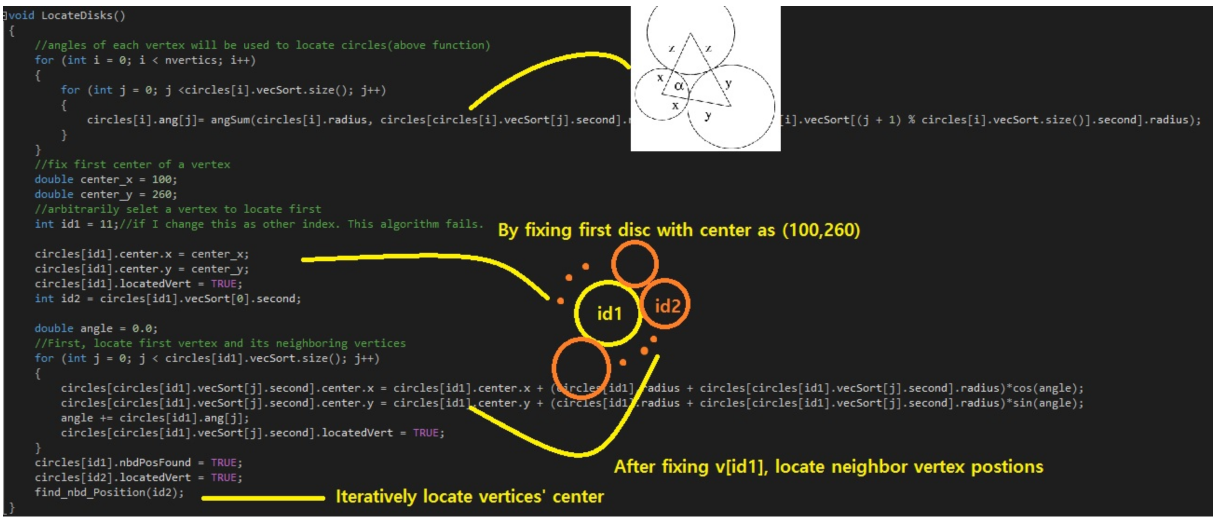
When I presented in 12/17 Thursday, there was an error with the locating part with disc. I tried to fix it. But there is still errors. In my code, it works well only with the vertices which have special character. I want to show one example. The input file is ”complex1.txt” and the boundary vertices and the UNM results of radii is like this.

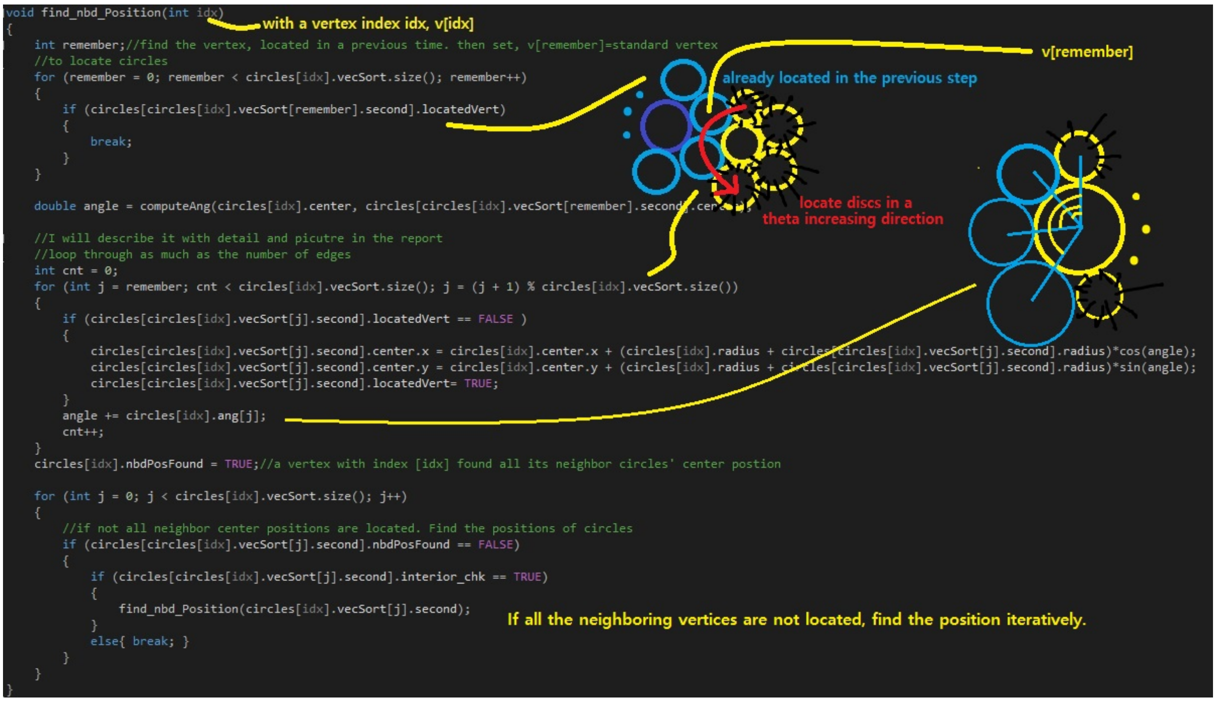


The output difference between the two case is that, the vertex $v[5]$ has a special property. With $v[5]$, since its neighbor vertices are all interior, the positions of circles tightly located for the begin. However, the with $v[11]$, even though it is an interior point since some of the neighboring vertices are not interior they causes a problem(only in my code). I need to see the locating part further.



The following two pictures describe the locating discs part. As we can see in the first picture,we need to give a location of the firs disc with user defined (x,y) positions. And then we iteratively find appropriate discs' positions.





```
void find_nbd_Position(int idx)
{
    int remember;//find the vertex, located in a previous time. then set, v[remember]=standard vertex
    //to locate circles
    for (remember = 0; remember < circles[idx].vecSort.size(); remember++)
    {
        if (circles[circles[idx].vecSort[remember].second].locatedVert)
        {
            break;
        }
    }

    double angle = computeAng(circles[idx].center, circles[circles[idx].vecSort[remember].second].center);

    //I will describe it with detail and picutre in the report
    //loop through as much as the number of edges
    int cnt = 0;
    for (int j = remember; cnt < circles[idx].vecSort.size(); j = (j + 1) % circles[idx].vecSort.size())
    {
        if (circles[circles[idx].vecSort[j].second].locatedVert == FALSE )
        {
            circles[circles[idx].vecSort[j].second].center.x = circles[idx].center.x + (circles[idx].radius + circles[circles[idx].vecSort[j].second].radius)*cos(angle);
            circles[circles[idx].vecSort[j].second].center.y = circles[idx].center.y + (circles[idx].radius + circles[circles[idx].vecSort[j].second].radius)*sin(angle);
            circles[circles[idx].vecSort[j].second].locatedVert= TRUE;
        }
        angle += circles[idx].ang[j];
        cnt++;
    }
    circles[idx].nbdPosFound = TRUE;//a vertex with index [idx] found all its neighbor circles' center postion

    for (int j = 0; j < circles[idx].vecSort.size(); j++)
    {
        //if not all neighbor center positions are located. Find the positions of circles
        if (circles[circles[idx].vecSort[j].second].nbdPosFound == FALSE)
        {
            if (circles[circles[idx].vecSort[j].second].interior_chk == TRUE)
            {
                find_nbd_Position(circles[idx].vecSort[j].second);
            }
            else{ break; }
        }
    }
}
```


5. CONCLUSION

In this final project, I had to understand the paper clearly and then summarize. and based on the understanding I need to implement the circle packing algorithm using UNM method. I did those part. I did everything from scratch. But there are insufficients in the "proving convergence in Error" and a "contribution" parts. I tried to prove that the error E monotone decreasing. To show this we need the lemmas in the paper. Using these lemmas I can prove this 'convergence' part by only drawing some examples and considering them.

$$e(v) = \theta(v;R) - A(v), \quad E = E(R) = \sum_{v:interior} |e(v)|$$

i.e. in the proving part I did not prove it mathematically rather I used some pictures to understand the claim(Error is monotone decreasing) in the paper. and I did not attatch contributions here. and there are still errors with disc locating part.