1 Delaunay Triangulation, Voronoi Diagram, and Gabriel Graph

Relate the Delaunay triangulation and Voronoi diagram to edges that appear in the Gabriel Graph. (Hint: The Gabriel Graph is a subgraph of the Delaunay triangulation). Don't all edges of a Delaunay triangle cross a Voronoi edge? Prove it in both directions.

($\Rightarrow$) For two points $p_i, p_j \in S$, $\overline{p_i p_j}$ is the edge that connects them, and C_{ij} is the circle with $\overline{p_i p_j}$ as the diameter. We say that C_{ij} is an empty circle with no points of S in its interior and therefore $\overline{p_i p_j} \in \mathcal{GG}(S)$; we then consider a third point $p_k \in S$, where $p_k \neq p_i, p_j$ and p_k is not interior to C_{ij}. Either

1. p_r is on the circumference of C_{ij}. Therefore, by definition[1], $\Delta p_i p_j p_j$ is a face in the Delaunay triangulation, and so $\overline{p_i p_j} \in \mathcal{DT}(S)$ (as do the other edges formed by the three points).

2. p_r is exterior to C_{ij}. Therefore, by definition[2], $\overline{p_i p_j} \in \mathcal{DT}(S)$.

The $\mathcal{DT}(S)$ is the dual of the $\mathcal{VD}(S)$; by definition, the vertices of $\mathcal{DT}(S)$ are Voronoi sites and the edges of $\mathcal{VD}(S)$ are those that connect those circumcenters in $\mathcal{DT}(S)$ that share a triangle edge (and so, cross that edge). We can conclude that for $\overline{p_i p_j} \in \mathcal{GG}(S)$ if and only if $\overline{p_i p_j} \in \mathcal{DT}(S)$, because $mathcalGG(S) \subseteq \mathcal{DT}(S)$.

Since $\mathcal{GG}(S) \subseteq \mathcal{DT}(S)$, we can find $\mathcal{GG}(S)$ very easily. First, compute the $\mathcal{DT}(S)$, which can be done in $O(n \log n)$ time and $O(n)$ space[3]. Then, We can walk the edges of $\mathcal{DT}(S)$ in $O(n)$ time, and check each if the circle with just that edge, $\overline{p_i p_j}$, as the diameter contains an interior point or not. Since this was an edge in the $\mathcal{DT}$, there is only one potential point that could be interior, the third point in the triangle; so we only need to check if the neighbors of p_i, p_j are interior to C_{ij}. If it has interior points, then discard that edge; otherwise, it is an edge in $\mathcal{GG}(S)$. The total runtime is $O(n \log n)$ due to computing the $\mathcal{DT}(S)$, which is optimal and faster than checking every pair of points with every other point for being interior, which would come out to be $O(n^3)$ time.

a.

[1] Theorem 9.6.i [2] pg. 198
[2] Theorem 9.6.ii [2] pg. 198
[3] Theorem 9.12 [2] pg. 206

2 Voronoi Diagrams in L_1 and L_∞ Metrics

Reference: [3], [4]
In an arbitrary d-dimensional plane R^d, for a real number $1 \leq p \leq \infty$, the L_p-distance between two points $q_i(x_{i1}, x_{i2}, ...x_{id})$ and $q_j(x_{j1}, x_{j2}, ...x_{jd})$ is given by the norm:

$$d_p(q_i, q_j) = \Big[\sum_{k=1}^{d} \Big(|x_k(q_i) - x_k(q_j)| \Big) \Big]^{\frac{1}{p}}$$

The plane in which the $L - p$ distance is measured is denoted R^d_p.

In a 2D plane R^2, given two points $q_i(x_i, y_i)$ and $q_j(x_j, y_j)$, the distance between them in the L_1 metric (R^2_1 plane) is given by:

$$d_1(q_i, q_j) = |x_i - x_j| + |y_i - y_j|$$

The distance between them in the L_∞ metric (R^2_∞ plane) is given by:

$$d_\infty(q_i, q_j) = \max\Big(|x_i - x_j|, |y_i - y_j| \Big)$$

There exists a linear mapping f from the R^2_∞ plane to the R^2_1 plane. For a point $(x, y) \in R^2_\infty$, there exists a point $(x', y') \in R^2_1$ where $x' = \frac{y+x}{2}$ and $y' = \frac{y-x}{2}$. The distance between two points $q_i, q_j \in R^2_\infty$ is equal to the distance between their images $q'_i, q'_j \in R^2_1$.

For example, let $q_i(x_i, y_i), q_j(x_j, y_j) \in R^2_\infty$ and let $x_j > x_i$, $y_i > y_j$, and $y_j - y_i > x_j - x_i$. So the between them is $d_\infty(q_i, q_j) = y_j - y_i$.

The distance between $q'_i(x'_i, y'_i), q'_j(x'_j, y'_j) \in R^2_1$ is equal to $d_\infty(q_i, q_j)$:

$$\begin{aligned}
d_1(q'_i, q'_j) &= |x'_j - x'_i| + |y'_j - y'_i| \\
&= \Big(\frac{y_j + x_j}{2} - \frac{y_i + x_i}{2} \Big) + \Big(\frac{y_j - x_j}{2} - \frac{y_i - x_i}{2} \Big) \\
&= \frac{1}{2}\Big((y_j - y_i) + (x_j - x_i) \Big) + \frac{1}{2}\Big((y_j - y_i) - (x_j - x_i) \Big) \\
&= \frac{1}{2}\Big(2(y_i - y_j) + (x_j - x_i) - (x_j - x_i) \Big) \\
&= y_i - y_j \\
&= d_\infty(q_i, q_j)
\end{aligned}$$

a.

3 Three

a.

4 Project

References

[1] Jake and Diane's office hours, classmates: Stephanie, Alex, Anju with homework problem discussions.

[2] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. 2008. Computational Geometry: Algorithms and Applications (3rd ed. ed.). Springer-Verlag TELOS, Santa Clara, CA, USA.

[3] D.T Lee and C.K. Wong, "Voronoi Diagrams in L_1 and (L_∞) Metrics with 2-Dimentional Storage Applications". SIAM J Comput. Vol. 9, No. 1, 200-212. 1980.

[4] Yunzhi Shi, "Sarah's Interactive Voronoi Diagram". 2017.