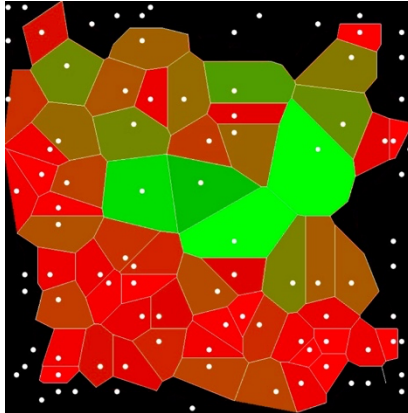


Histological Analysis Using Voronoi Diagrams

Henry Webb



1. Abstraction

In this study, I present a cell analysis program that utilizes Voronoi diagrams for the visualization and statistical analysis of cellular structures. Leveraging the computational geometry library CGAL for Delaunay triangulation and SFML for graphical representation, the program effectively processes and analyzes cell data. This approach is particularly applicable in histological analysis, offering enhanced precision in cell segmentation and measurement.

My results demonstrate the program's potential to calculate statistics quickly and automatically and use those to create informative visualizations. This project not only connects to the field of computational biology but also be applied to cell analysis in various other scientific and medical applications.

2. Introduction

2.1 Histological Image Analysis

Histological analysis is a vital aspect of medical pathology and has been significantly enhanced by the incorporation of digital image processing techniques. This advancement has revolutionized the way tissues and cells are examined on a microscopic level. The analysis starts with image segmentation, a complex process that involves differentiating individual cells from their

surroundings. This step is crucial for the precise identification of cellular structures within much larger complex arrangements as seen below in figure 2.1.

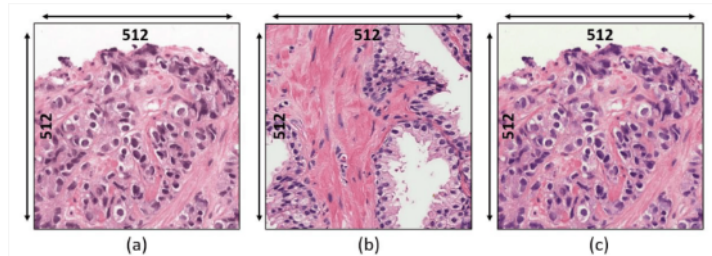


Figure 2.1 Cell Slide Example

Next, feature extraction provides a quantitative analysis of cellular characteristics. It checks aspects like size, shape, and textural nuances, supplying critical data that can help detect pathological changes within cells. These details are essential for creating a morphological profile of each cell, which is used for recognizing irregular cellular behavior indicative of diseases.

The final stage of the process involves the application of classification algorithms. These programs interpret the extracted features to categorize cells, often leveraging advanced machine learning techniques to help with diagnostics. The ability to distinguish between benign and malignant cells, among other classifications, is the backbone of modern pathology, and is essential for both research and medical applications alike.

When put together, these processes represent a shift towards automation in pathology, which will ensure increased precision and reproducibility in histological cell analysis.

3. Geometric Structures

3.1 Delaunay Triangulation

Delaunay triangulation is a fundamental part of computational geometry, which is used for creating a mesh of triangles from a given set of points in a plane. The defining property of a delaunay triangulation is that no point is inside the circumcircle of any triangle. This leads to triangles that are as close to equilateral as possible, minimizing “skinny” triangles. The outside of this triangulation is known as the convex hull.

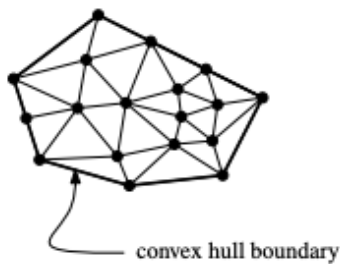


Figure 3.1 Delaunay Triangulation

The algorithm and computational structure have several practical applications, ranging from geographical information systems to numerical simulations and computer graphics. In two dimensions, the Delaunay triangulation maximizes the minimum angle of all the angles of the triangles in the triangulation, avoiding narrow triangles by considering overall geometry rather than local features.

3.2 Voronoi Diagram

Voronoi diagrams partition a plane based on a set of points into distinct regions. For each seed point, the corresponding Voronoi cell consists of all points in the plane closer to that seed than to any other. This creates a tessellation of the plane into convex polygons, with each cell reflecting the area around a given seed point. These diagrams are used in various fields of science and

technology, including materials science, spatial analysis, and the modeling of natural processes.

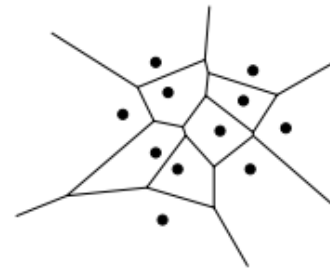


Figure 3.2 Voronoi Diagram

In a Voronoi diagram, the boundaries between cells are equidistant from the nearest two seed points, forming a network of vertices and edges that section out regions of proximity. The mathematical properties of Voronoi diagrams allow for the analysis of spatial relationships and distributions, making them perfect for use cases such as geospatial studies, biology for modeling cellular patterns, and in computer science for solving nearest-neighbor search problems and network modeling.

3.3 Dual Property

The dual relationship between Voronoi diagrams and Delaunay triangulations is an essential part of computational geometry. Voronoi diagrams segment a plane into regions based on proximity to a set of points, while Delaunay triangulations connect these points to form triangles. This duality means that the vertices of a Voronoi diagram correspond to the circumcenters of the Delaunay triangles, and vice versa. The edges of the Voronoi diagram are perpendicular bisectors of the edges in the Delaunay triangulation, reflecting a deep geometric connection between these two constructs.

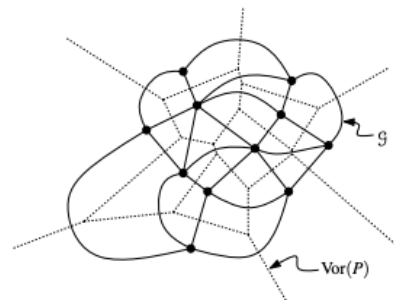


Figure 3.3 Dual Graph of Vor(P)

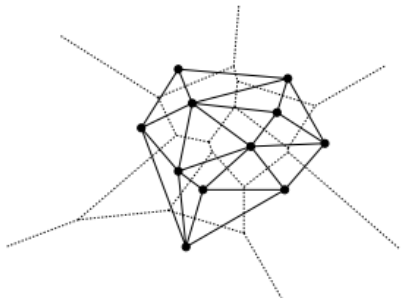


Figure 3.4 Delaunay Graph $DG(P)$

This dual nature allows for the seamless transformation of data from one representation to the other, making them helpful in various applications. In spatial modeling, for instance, Delaunay triangulations offer a way to model terrain or spatial data, while the corresponding Voronoi diagrams can be used for nearest-neighbor analysis or regional partitioning for visualizations.

4. Implementation

4.1 Input Creation

The scope of the project had to be altered slightly to ensure feasibility within the given timeframe and to follow content from the class. Therefore, I made the decision to pivot towards a simplified version of the project that focused more on the analysis of the cells once the centers were known. To do this, I manually created the input files for the program instead of the image analysis as mentioned before.

Each test file contained points represented by x,y coordinates, simulating various spatial data scenarios. Three non-uniform test files were designed to reflect small (10 points), medium (100 points), and large (1000 points) datasets, each of which will be used at different times during the implementation. A fourth file, with a uniform distribution and 100 points, provided a way for assessing the accuracy of the system's computational geometry algorithms, specifically, the precision of area and perimeter calculations. This method ensured there would be small input files for initial testing, and large files for later stress testing of the program.

4.2 Delaunay Triangulation

Utilizing the robust capabilities of the CGAL (Computational Geometry Algorithms Library), the program was able to efficiently implement Delaunay triangulations from our input files. This geometric structure is crucial for the program as it establishes the interconnectivity of points, laying the groundwork for further computational analysis and other geometric structures to be used in later steps.

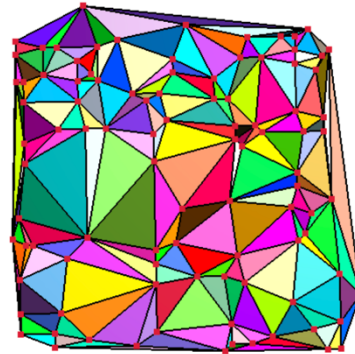


Figure 4.1 Delaunay Triangulation of medium input

4.3 Voronoi Diagram

The Voronoi diagram's creation in this project was created by leveraging the dual relationship with Delaunay triangulations. Through the code, Delaunay triangulations are first computed using CGAL, which forms the foundational structure. The Voronoi diagram is then derived from these triangulations, utilizing their geometric properties. Each Voronoi cell corresponds to a node in the Delaunay graph, and its edges are perpendicular to those of the Delaunay triangles (figure 3.4).

In the second phase, the graphical representation of the Voronoi diagram was achieved using SFML. This library steps in to visually depict what CGAL computes, turning numerical data into a clear, graphical layout. SFML's ability to render shapes and lines is used to draw the Voronoi cells and their boundaries, providing a visual understanding of the spatial relationships between points as seen below in figure 4.2. This visualization is crucial for analyzing and interpreting the results, as well as for the future steps in this process.

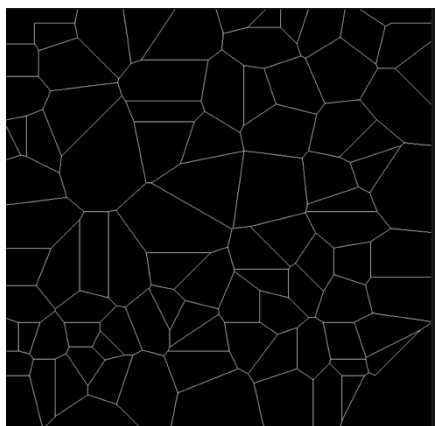


Figure 4.2 Voronoi Diagram of medium input using SFML

4.4 Area & Perimeter Calculations

The area calculation of each Voronoi cell is a crucial part of this program and directly used for the visualization in the final step of this project. In the code, the area of each Voronoi cell is calculated by first gathering the vertices of the cell using CGAL's Delaunay triangulation methods. These vertices are then organized using a convex hull algorithm, which constructs the smallest convex polygon enclosing all the points. After obtaining the sorted vertices of the convex hull, the area is calculated using a mathematical formula that sums the cross-products of the coordinates of the vertices. This method ensures accurate computation of the areas, essential for the precise analysis of spatial properties in the Voronoi diagram.

Perimeter calculation is another essential when trying to make insights into the geometric properties of each cell. For example, finding a cell with a very high perimeter compared to its area can suggest a non-uniform shape for that cell. The perimeter calculation is implemented using CGAL's iterators, which allow for efficient traversal of each Voronoi cell's edges, through their Delaunay counterparts. The iterators enable the program to move around each cell, accessing vertices to measure edge lengths. The lengths are computed using CGAL's `squared_distance` function, which calculates the Euclidean distance between two points, providing an accurate measurement for each edge.

4.5 Visualizations

The last and arguably most important part of the project was to create the final visualizations based on the previously calculated area and perimeter values. As previously mentioned CGAL does not provide direct support for creating visualizations of Voronoi diagrams, so I continued to use SFML for this section as well. One additional step in the visualization was removing the border cells as they were not counted in our calculations. The approach I chose to do this was by filtering out edges connected to infinite size faces, ensuring that only finite Voronoi cells are visualized. This step is essential otherwise the cells with infinite faces would skew later calculations.

Testing visualizations played a key role in ensuring correctness, especially in verifying the points used for Voronoi cells. By visualizing various scenarios, including highlighting vertices of specific edges, and focusing on boundary cells, the testing phase ensured that the program was using the correct input points for each step as well as computing the Voronoi diagram correctly.

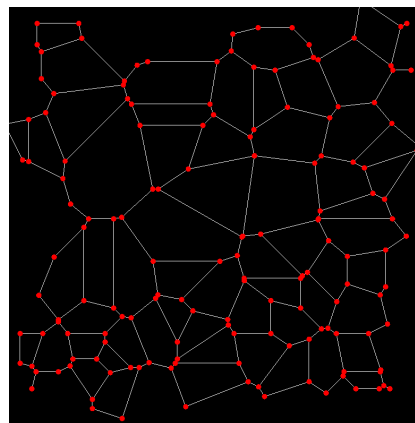


Figure 4.3 Test Visualization of Voronoi Vertices on top of their Edges

In the final output which can be seen in Figure 4.4 below, the color generation for cells was tied to their areas. By scaling colors from red (representing smaller areas) to green (for larger areas), the code uses the minimum and maximum area values as reference points. This color-coding provides an intuitive understanding of the cells' relative sizes at a glance, adding an additional layer of analysis to the visualization. This

visualization is just a basic example of some of the potential outputs for this program.

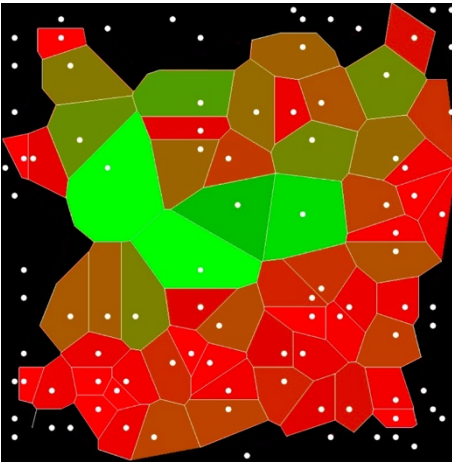


Figure 4.4 Final Output Heatmap Based on Area

5. Future Work

5.1 Bug Fixes

During the project, there were various issues with the output, such as the appearance of extra edges and the filling of unbounded faces near the edges of the diagram. Despite my best efforts, these problems remained unresolved within the project's timeframe. However, I have ideas for future improvements, one of which implements a surface mesh using CGAL. This approach aims to refine the diagram's accuracy by recalculating and redefining the edges and faces more precisely, potentially resolving the existing inaccuracies and enhancing the overall quality of the output visualization.

5.2 New Functionality

The next addition to the program would be the implementation of more advanced visualizations, such as cell density heatmaps and markers for irregularly shaped cells. Heatmaps would visually represent the concentration of cells in different areas, providing insights into cell distribution patterns. Marking irregular cells can highlight anomalies in cell shapes, aiding in more detailed analyses and easy detection.

Along with enhancing the visualizations, another next step would be creating calculations for cell irregularity. Functions to measure uniformity using internal angles and the ratio of area to perimeter would offer a quantitative assessment of cell shape and size regularity. This feature would be particularly useful in studies where cell morphology is critical such as in histology as mentioned.

Another feature to be added is the ability for users to set specific conditions for visualization, such as displaying cells with an area above a certain threshold or not showing groups of cells below a certain density. This functionality would allow for more targeted and customized analysis, catering to specific research needs or highlighting aspects of the cellular structure. This extra level of interactivity would significantly increase the program's usability in various research and medical scenarios.

References

- Belsare, A. D., and M. M. Mushrif. "Histopathological image analysis using image processing techniques: An overview." *Signal & Image Processing* 3.4 (2012): 23.
- De Berg, Mark. *Computational geometry: algorithms and applications*. Springer Science & Business Media, 2000.
- Bhattacharjee, S., Ikromjanov, K., Carole, K.S., Madusanka, N., Cho, N-H., Hwang, Y-B., Sumon, R.I., Kim, H-C., Choi, H-K. "Cluster Analysis of Cell Nuclei in H&E-Stained Histological Sections of Prostate Cancer and Classification Based on Traditional and Modern Artificial Intelligence Techniques." *Diagnostics* 12.1 (2022): 15. Web.
- Gurcan, Metin N et al. "Histopathological image analysis: a review." *IEEE reviews in biomedical engineering* vol. 2 (2009): 147-71. doi:10.1109/RBME.2009.2034865
- Edla, D. R., and P. K. Jana. "Clustering Biological Data Using Voronoi Diagram." *Advanced Computing, Networking and Security*. Edited by P. Santhi Thilagam et al., Springer, 2012, pp. 213-221, *Lecture Notes in Computer Science*, vol. 7135, doi:10.1007/978-3-642-29280-4_21.