# Representing Ordered Order-$k$ Voronoi Diagrams

Ye Wang
School of Business (IT)
James Cook University
Cairns, QLD 4870, Australia
Email: Ye.Wang@my.jcu.edu.au

Kyungmi Lee
School of Business (IT)
James Cook University
Cairns, QLD 4870, Australia
Email: Joanne.Lee@jcu.edu.au

Ickjai Lee
School of Business (IT)
James Cook University
Cairns, QLD 4870, Australia
Email: Ickjai.Lee@jcu.edu.au

***Abstract*—The ordered order-$k$ Voronoi diagram is a popular generalization of the ordinary Voronoi diagram modeling ordered $k$ nearest sites. Despite of the usefulness of the ordered order-$k$ Voronoi diagram, its structural complexity and wealth of information captured result in poor readability. Most research has focused on computational complexity, and visualization/representation of the ordered order-$k$ Voronoi diagram has received little attention. This paper attempts to represent the ordered order-$k$ Voronoi diagram through its relationship with other higher order Voronoi diagrams. The proposed method highlights topological relationships between/among higher order Voronoi diagram families, and enhances the readability of the ordered order-$k$ diagram.**

## I. Introduction

The Ordinary Voronoi Diagram (OVD) has gained a lot of attention due to its mathematical robustness and soundness, and has been applied to many domain-specific applications [1]. The OVD segments the space in such a way that every location in the space is allocated to the closest generator. It tessellates the space into mutually exclusive and collectively exhaustive regions. A set of locations closest to more than one generator formulates the OVD (Voronoi edges and vertices). There are many generalized Voronoi diagrams to extend its capability and applicability to model more complex real world situations. One such generalization is Order-$k$ Voronoi Diagrams (O$k$VD) modeling unordered $k$ closest generators that will become Ordered Order-$k$ Voronoi Diagrams (OO$k$VD) when $k$ generators are ordered. The wealth of information modeled by O$k$VD and OO$k$VD makes them useful for many applications and decision makings [1]. However, this richness makes visualization difficult, results in poor readability, and creates a poverty of attention [2].

O$k$VD and OO$k$VD Voronoi regions do not necessarily contain their corresponding generators for $k \geq 2$. Thus, it is not straightforward to represent the relationship between a Voronoi region and its generators. There have been some algorithms for computing Higher Order Voronoi Diagrams (HOVDs) (O$k$VD and OO$k$VD) [3], [4], [5], [6], but representation/visualization of HOVD has attracted relatively less attention [7], [8]. Traditional visualization approaches focus on O$k$VD, in particular, the relationship between the O$k$VD region and its generators through raster Voronoi tessellations.

This paper revisits these traditional visualization approaches, interprets OO$k$VD through a complete set of O$k$VDs, and highlights the relationship between O$k$VD and OO$k$VD to visualize OO$k$VD. The main aim of this paper is not to focus on visualizing a single OO$k$VD, but interactively representing OO$k$VD through a complete set of O$k$VDs by highliting the relationship between O$k$VD and OO$k$VD. The approach is based on vector Voronoi diagrams [9], [10], [11]. Section II overviews background information including definitions on HOVDs, computational algorithms, and existing visualization approaches. Section III reviews the Delaunay triangle-based data structure [10], [12], analyzes the relationship between O$k$VDs and OO$k$VDs, and provides details on OO$k$VD visualization. Section IV briefly reviews and concludes.

## II. Preliminaries

### A. Voronoi Definitions

The Voronoi diagram naturally divides the study region $S$ into mutually exclusive and collective exhaustive sub-regions [1]. Thus, it is regarded as a point-to-region transformation. For a given set of $P = \{p_1, p_2, \ldots, p_n\}$, the OVD $\mathcal{V}$ is a set of $\{V(p_1), \ldots, V(p_n)\}$ Voronoi regions where each $V(p_i)$ (for $p_i \in P$) is the set of locations that are closer to $p_i$ than other locations in $S$. The O$k$VD $\mathcal{V}^{(k)}$ is a set of all order-$k$ Voronoi regions $\{V(P_1^{(k)}), \ldots, V(P_l^{(k)})\}$, where the order-$k$ Voronoi region $V(P_i^{(k)})$ ($V(P_i^{(k)}) \neq \emptyset$) for a certain subset $P_i^{(k)}$

consisting of $k$ points out of $P$ defined as:

$$V(P_i^{(k)}) = \{p | \max_{p_r \in P_i^{(k)}} d(p, p_r) \leq \min_{p_s \in P \setminus P_i^{(k)}} d(p, p_s)\}.$$

Note that the number of order-$k$ Voronoi regions is $O(k(n-k))$ and the actual size of $l$ is $O(n^2)$ in $\mathbb{R}^2$ [5].

In the O$k$VD, $k$ generators are not ordered, however in some situations an ordered set (sequence) would be of interest. This is modeled by the OO$k$VD $\mathcal{V}^{<k>}$. It is a set of ordered order-$k$ Voronoi regions $\{V(P_1^{<k>}), \ldots, V(P_m^{<k>})\}$ $(V(P_i^{<k>}) \neq \emptyset)$, where $m = n(n-1)\ldots(n-k+1)$ and the ordered order-$k$ Voronoi region of $P_i^{<k>} = \{p_{i1}, \ldots, p_{ik}\}$ defined as:

$$V(P_i^{<k>}) = \{p | d(p, p_{i1}) \leq \cdots \leq d(p, p_{ik}) \leq d(p, p_j),$$

$$p_j \in P \setminus \{p_{i1}, \ldots, p_{ik}\}\}.$$

$\mathcal{V}(P_i^{<k>})$ is a refinement of $\mathcal{V}(P_i^{(k)})$, namely $V(P_i^{(k)}) = \bigcup_{P_j^{<k>} \in A^{<k>}(P_i^{(k)})} V(P_j^{<k>})$, where $A^{<k>}(P_i^{(k)})$ is the sequence of all possible $k$-tuples made of $p_{i1}, \ldots, p_{ik}$ [1].

One popular variant of the OVD similar to $\mathcal{V}^{(k)}$ is the $k$-th Nearest Voronoi Diagram ($k$NVD). This is particularly useful when users are interested in only $k$-th nearest region. The $k$NVD $\mathcal{V}^{[k]}$ is a set of all $k$-th nearest Voronoi regions $\mathcal{V}^{[k]} = \{V^{[k]}(p_1), \ldots, V^{[k]}(p_n)\}$, where the $k$-th nearest Voronoi region $V^{[k]}(p_i)$ is defined as:

$$V^{[k]}(p_i) = \{p | d(p, p_i) \leq d(p, p_j),$$

$$p_j \in P \setminus \{k \text{ nearest generators of } p_i\}\}.$$

$\mathcal{V}(P_i^{<k>})$ is a refinement of $\mathcal{V}(P_i^{[k]})$. That is, $V(P_i^{[k]}) = \bigcup_{(p_{j1}, \ldots, p_{jk-1}) \in A^{<k-1>}(P \setminus \{p_i\})} V((p_{j1}, \ldots, p_{jk-1}, p_i))$, where $A^{<k-1>}(P \setminus \{p_i\})$ is the set of all possible $(k-1)$-tuples consisting of $k-1$ elements out of $P \setminus \{p_i\}$ [1], [13].

### B. Voronoi Algorithms

A list of algorithms to compute HOVDs has been proposed in the literature [3], [4], [5]. Some algorithms use Delaunay triangle-based data structures to compute HOVDs and others directly compute HOVDs without any data structure. The former better supports topological queries at the expense of memory. In this paper, we focus on algorithms based on Delaunay triangle-based data structures since this paper focuses on visualizing topological relationships. Dehne [4] proposed an algorithm based on a Delaunay triangle-based data structure that constructs the complete order-$k$ Voronoi diagrams in $O(n^4)$ time. This Delaunay triangle-based data structure is a robust candidate for geospatial applications since it supports both geometrical and topological neighborhoods. However, it only supports the complete order-$k$ Voronoi diagrams. Lee and Lee [10] further studied this structure, and extended it to the complete set of higher order Voronoi diagram families ($\mathcal{V}^{(k)}$, $\mathcal{V}^{<k>}$ and $\mathcal{V}^{[k]}$). In the literature, interrelationships of higher order Voronoi diagram families and applications to the real-world problem have been studied [10], [14].

### C. Voronoi Visualization

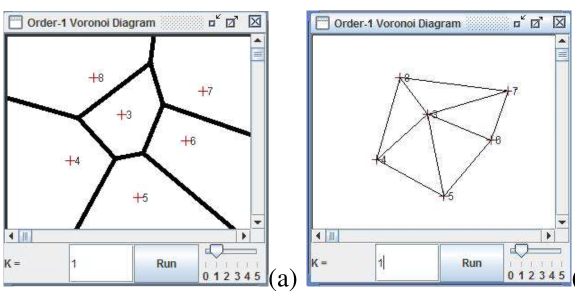
Voronoi diagrams have been widely used for visualizing domain-specific datasets [15], [16], [17], [18], but little research has been conducted to visualize Voronoi diagrams themselves [7], [8]. The OVD has been typically visualized through Voronoi lines, vertices and regions [1]. Interactive visualization could be used to visualize the OVD. Interactive visualization is a visualization approach that takes user input to focus on, and displays detailed information on focused area in real-time [19]. It it good to highlight areas of focus, but not good for producing an overall view simultaneously. Due to its filtering nature, interactive visualization is particularly important in complex datasets or data-rich environments where scalability is of issue [20]. Voronoi growth models (similarly to contour models) could be used to visualize how each generator starts generating Voronoi regions. When two growing regions meet each other, then they stop growing and formulating a Voronoi edge. This growth model based visualization dynamically shows how the OVD is formed, but it it not straightforward to use this approach to visualize HOVDs. This is because the OVD has one generator within its corresponding Voronoi region whilst HOVDS have multiple generators not necessarily within a corresponding higher order Voronoi region.

Palmer [7] explores visual aspects of HOVD through contour lines and textures based on raster HOVD. He introduces order-$k$ plots (overlaying nearest neighbor plots for 1,...,$k$-1. That is order-3 plot is the order-3 Voronoi diagram overlaid with contour lines of order-1 and order-2 (1st nearest plot and second nearest plot). However, it becomes too complex when $k$ or $n$ grows. Other researchers [8] attempted to visualize HOVD regions using color and shading (with cushions (different intensity levels to give shading to represent the distance to the closest site. Brighter as closer to the generators whilst darker as close to the edges.) and bevels (colored for influencing sites and scaled to reflect the $k$-th order distance and hierarchy)), and to superimpose to implicitly show the $k$ sites.Basically these approaches focus on visually answering which are the $k$ sites that influence a given partition. However, these traditional approaches share some common drawbacks: 1) based on raster Voronoi diagrams, thus incapable of visualizing rich topological relationships; 2) only for visualizing one O$k$VD; 3) not scalable with $k$ or $n$.

## III. HOVD VISUALIZATION

### A. *HOVD Relationships*

The Delaunay triangulation is a dual graph of the OVD. It is obtained by connecting two generators when they share a Voronoi edge in common. The circumcircle of each Delaunay triangle does not contain any other generator in it. These Delaunay triangles are Order-0 Delaunay triangles. This set of Order-0 Delaunay triangles forms the Delaunay triangulation from which the OVD can be constructed. The relationship between the two diagrams is shown in Fig 1. Accordingly, Order-1 Delaunay triangles are those triangles whose corresponding circumcircles contain only one generator in it. Similarly, Order-$k$ Delaunay triangles are those triangles whose corresponding circumcircles contain $k$ generators in it.



(a) (b)

Fig. 1: The OVD and its dual Delaunay Triangulation; (a) The OVD; (b) Its dual Delaunay triangulation.

A complete set of HOVDs can be computed from this set of Order-$k$ Delaunay triangles [4], [10]. Table I summarizes the relationship between the complete Order-$k$ Delaunay triangle data structure and various HOVD families. For order 1, O1VD, OO1VD and 1NVD are the same as the OVD that can be computed from the Order-0 Delaunay triangles (Delaunay triangulation). For order 2, O2VD is computed from both Order-0 Delaunay triangles and Order-1 Delaunay triangles (Readers may refer to [4], [10] for detailed derivation). OO2VD is an overlay of O1VD and O2VD. 2NVD is the same as OO2VD. For order 3, O3VD is derived from both Order-1 Delaunay triangles and Order-2 Delaunay triangles. OO3VD is obtained by superimposing from O1VD to O3VD whilst 3NVD is obtained by overlaying O2VD and O3VD. Similarly, O$k$VD is calculated from both Order-($k$-2) Delaunay triangles and Order-($k$-1) Delaunay triangles. OO$k$VD is an overlay of all O$k$VDs (from O1VD to O$k$VD) whilst $k$NVD is an overlay of O$k$-1VD and O$k$VD.

TABLE I: The relationship between the complete Order-$k$ Delaunay triangle data structure and HOVDs.

| Order $k$ | O$k$VD | OO$k$VD | $k$NVD |
|---|---|---|---|
| 1 | Order-0 triangle | O1VD | O1VD |
| 2 | Order-0 triangle<br>&<br>Order-1 triangle | O1VD<br>$\sim$<br>O2VD | O1VD<br>&<br>O2VD |
| 3 | Order-1 triangle<br>&<br>Order-2 triangle | O1VD<br>$\sim$<br>O3VD | O2VD<br>&<br>O3VD |
| 4 | Order-2 triangle<br>&<br>Order-3 triangle | O1VD<br>$\sim$<br>O4VD | O3VD<br>&<br>O4VD |
| ... | ... | ... | ... |
| $k$ | Order-($k$-2) triangle<br>&<br>Order-($k$-1) triangle | O1VD<br>$\sim$<br>O$k$VD | O$k$-1VD<br>&<br>O$k$VD |

### B. *OO$k$VD Visualization*

OO$k$VD visualization highlights the relationship between O$k$VD and OO$k$VD discussed in the previous subsection. Voronoi edges are utilized to highlight the relationship, and interactive visualization is used to display interested area.

The width of a Voronoi edge is inversely proportional to the order of $k$. It becomes thinner as $k$ grows. This is to emphasize lower order O$k$VD. Different colors are used for different $k$. The user is able to choose these colors, but black for Order-1, red for Order-2, blue for Order-3, and green for Order-4 are used in the program by default. Figure 2 shows from O1VD to O4VD for a set $P = \{p_3, p_5, \ldots, p_{10}\}$ of 8 points. Using different widths and colors, it becomes easier to differentiate O$k$VDs.

Figure 3 depicts $k$NVDs. Figure 3(a) shows 2NVD of $P$ which is an overlay of O1VD (Fig. 2(a)) and O2VD (Fig. 2(b)). O1VD and O2VD do not share any Voronoi edge, but they do share some Voronoi vertices. This is because O1VD is derived from Order-0 Delaunay triangles, and O2VD is derived from both Order-0 Delaunay triangles and Order-1 Delaunay triangles. Thus, O1VD and O2VD share Voronoi vertices generated from Order-0 Delaunay triangles. Figure 3(b) illustrates 3NVD of $P$ which is an overlay of O2VD

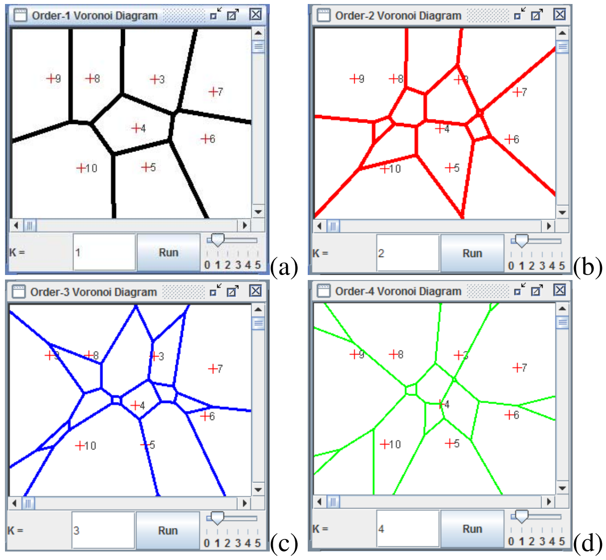



Fig. 2: Order-$k$ Voronoi diagrams ($|P| = 8$) where $p_i \in P$ is depicted as a cross (+) and an associated identification number $i$: (a) $\mathcal{V}^{(1)}(P)$; (b) $\mathcal{V}^{(2)}(P)$; (c) $\mathcal{V}^{(3)}(P)$; (d) $\mathcal{V}^{(4)}(P)$.

(Fig. 2(b))and O3VD (Fig. 2(c)). Similarly, O2VD and O3VD do not share any Voronoi edge, but they do some Voronoi vertices. These shared vertices are from Order-1 Delaunay triangles since O2VD is generated from both Order-0 and Order-1 Delaunay triangles whilst O3VD is from both Order-1 and Order-2 Delaunay triangles. That is, O2VD Voronoi vertices are shared with either O1VD (those generated from Order-0 Delaunay triangles) or O3VD (those generated from Order-1 Delaunay triangles). These relationships are clearly illustrated with visualization of $k$NVDs with different colors and edge widths. Similarly, Figure 3(c) depicts 4NVD of $P$ which is an overlay of O3VD (Fig. 2(c)) and O4VD (Fig. 2(d)).

Figure 4 illustrates OO3VD and OO4VD of $P$. Figure 4(a) shows $\mathcal{V}^{<3>}(P)$ which is obtained by superimposing O1VD, O2VD and O3VD of $P$. As can be seen from the figure, the entire spectrum of Voronoi vertex sharing (relationship among O$k$VDs) is explained with OO$k$VDs differently from $k$NVD which only shows the sharing between O$k$VD and O$k$-1VD. Figure 4(b) displays $\mathcal{V}^{<4>}(P)$ which is obtained from the combination of O1VD, O2VD, O3VD and O4VD of $P$. Even though an OO$k$VD becomes complex as $k$ grows, the vertex sharing relationship can be easily identified. One thing to note is that a Voronoi vertex of O$k$VD is shared by at maximum two O$k$VDs. Voronoi edges of O$k$VDs are not shared whilst Voronoi regions of O$k$VDs are decomposed into smaller Voronoi regions.

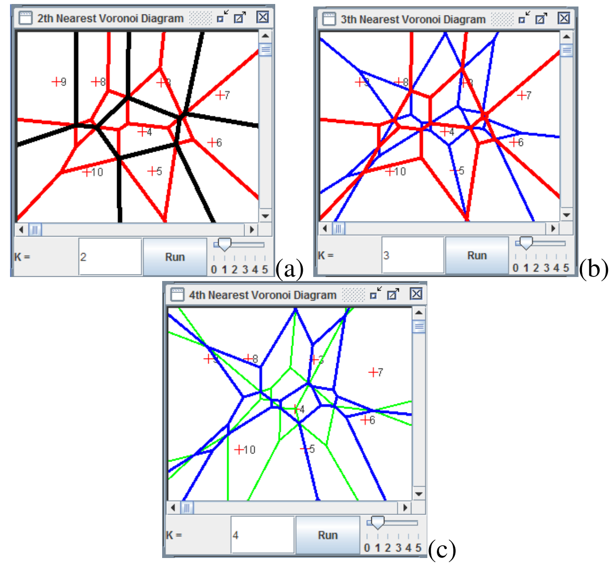



Fig. 3: $k$NVDs ($|P|$ is the same as in Fig. 2): (a) $\mathcal{V}^{[2]}(P)$; (b) $\mathcal{V}^{[3]}(P)$; (c) $\mathcal{V}^{[4]}(P)$.

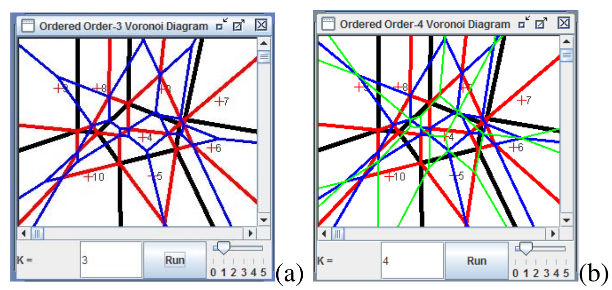



Fig. 4: OO$k$VDs ($|P|$ is the same as in Fig. 4): (a) $\mathcal{V}^{<3>}(P)$; (b) $\mathcal{V}^{<4>}(P)$.

Figure 5 shows a screen capture of the program. The program extends [10] to visualize the relationship between/among HOVD families. The program builds a unified Delaunay triangle data structure from which a complete set of HOVD families can be drawn. It is a vector-based Voronoi diagram, thus a rich set of relationship could be extracted from the data structure. The user is able to draw as many HOVD families as he/she wants for the same dataset, and is able to visualize them simultaneously. The snapshot shown in Figure 5 shows 8 different HOVDs: O1VD, O2VD, O3VD, O4VD, OO4VD, 2NVD, 3NVD, and 4NVD. When the user mouse clicks at the highlighted (red) region in OO4VD, then corresponding higher order Voronoi regions in all opened windows (HOVDs) are highlighted. Figure 5 clearly shows that the mouse clicked region in OO4VD corresponds to Voronoi region of $p_{10}$ in $\mathcal{V}^{(1)}(P)$, $p_{10}$ and $p_6$ in $\mathcal{V}^{(2)}(P)$, $p_{10}$, $p_6$ and $p_4$ in $\mathcal{V}^{(3)}(P)$, and $p_{10}$, $p_6$, $p_4$ and $p_9$ in $\mathcal{V}^{(4)}(P)$. This implicitly states that $p_{10}$, $p_6$, $p_4$ and $p_9$ are 4 ordered order nearest points from the mouse clicked area in OO4VD. The program not only illustrates this ordered generator information, but also relationships among HOVD families.

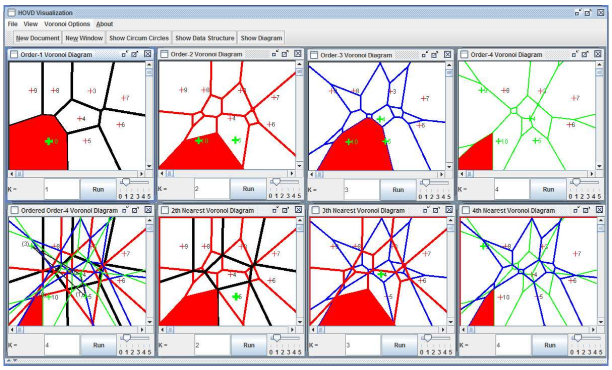



Fig. 5: A screen capture of the program ($|P|$ is the same as in Fig. 2).

## IV. Final Remarks

As algorithms for HOVD families become mature, it is becoming a research problem how to represent/visualize complex structural inter-relationships among HOVD families. This paper employs the Delaunay triangle-based data structure to visualize topological relationships of OO$k$VDs with other HOVD families. It uses different colors and widths for Voronoi edges to highlight the inter-relationships.

## References

[1] A. Okabe, B. N. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed. West Sussex: John Wiley & Sons, 2000.

[2] H. A. Simon, "Designing Organizations for an Information-Rich World," *Computers, Communications, and the Public Interest*, pp. 37–72, 1971.

[3] F. Aurenhammer and O. Schwarzkopf, "A Simple On-Line Randomized Incremental Algorithm for Computing Higher Order Voronoi Diagrams," in *Proceedings of the Seventh Annual Symposium on Computational Geometry*. North Conway, NH: ACM Press, 1991, pp. 142–151.

[4] F. K. H. A. Dehne, "An $O(N^4)$ Algorithm to Construct all Voronoi Diagrams for K-Nearest Neighbor Searching in the Euclidean plane," in *Proceedings of the International Colloquium on Automata, Languages and Programming*, ser. Lecture Notes in Computer Science 154, J. Díaz, Ed. Barcelona, Spain: Springer, 1983, pp. 160–172.

[5] D.-T. Lee, "On $k$-Nearest Neighbor Voronoi Diagrams in the Plane," *IEEE Transactions on Computers*, vol. 31, no. 6, pp. 478–487, 1982.

[6] B. Chazelle and H. Edelsbrunner, "An Improved Algorithm for Constructing $k$th-Order Voronoi Diagrams," *IEEE Transactions on Computers*, vol. 36, no. 11, pp. 1349–1354, 1987.

[7] J. D. Palmer, "Using Line and Texture to Visualize Higher-Order Voronoi Diagrams," in *Proceedings of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering*, ser. ISVD '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 166–172. [Online]. Available: http://dx.doi.org/10.1109/ISVD.2006.36

[8] A. Telea and J. J. van Wijk, "Visualization of Generalized Voronoi Diagrams," in *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization, year = 2001,*.

[9] I. Lee, R. Pershouse, and K. Lee, "Geospatial Cluster Tessellation through the Complete Order-$k$ Voronoi Diagrams," in *Proceedings of the Eight International Conference on Spatial Information Theory*, ser. Lecture Notes in Computer Science 4736. Springer, 2007, pp. 321–336.

[10] I. Lee and K. Lee, "A Generic Triangle-Based Data Structure of the Complete Set of Higher Order Voronoi Diagrams for Emergency Management," *Computers, Environments and Urban Systems*, vol. 33, no. 2, pp. 90–99, 2009.

[11] I. Lee, R. Pershouse, P. Phillips, and C. Christensen, "What-if Emergency Management System: A Generalized Voronoi Diagram Approach," in *Proceedings of the First Pacific-Asia Workshop on Intelligence and Security Informatics*, ser. Lecture Notes in Computer Science 4430, C. C. Yang, D. Zeng, M. Chau, K. Chang, Q. Yang, X. Cheng, J. Wang, F.-Y. Wang, and H. Chen, Eds. Chengdu, China: Springer, 2007, pp. 58–69.

[12] I. Lee and M. Gahegan, "Interactive Analysis using Voronoi Diagrams: Algorithms to Support Dynamic Update from a Generic Triangle-Based Data Structure," *Transactions in GIS*, vol. 5, no. 4, 2001, in press.

[13] M. Gahegan and I. Lee, "Data Structures and Algorithms to Support Interactive Spatial Analysis Using Dynamic Voronoi Diagrams," *Computers, Environments and Urban Systems*, vol. 24, no. 6, pp. 509–537, 2000.

[14] A. Okabe, B. N. Boots, and K. Sugihara, "Nearest Neighbourhood Operations with Generalized Voronoi Diagrams: A Review," *International Journal of Geographical Information Systems*, vol. 8, no. 1, pp. 43–71, 1994.

[15] M. Abellanas and B. Palop, "Urban Data Visualization with Voronoi Diagrams," in *Proceedings of the International Conference on ICCSA*, ser. Lecture Notes in Computer Science 5072. Springer, 2008, pp. 126–136.

[16] M. Balzer, O. Deussen, and C. Lewerentz, "Voronoi Treemaps for the Visualization of Software Metrics," in *Proceedings of the 2005 ACM Symposium on Software Visualization*. ACM Press, 2005, pp. 165–172.

[17] F. Dupuis, J.-F. Sadoc, R. Jullien, B. Angelov, and J.-P. Mornon, "Voro3D: 3D Voronoi Tessellations Applied to Protein Structures," *Bioinformatics*, vol. 21, no. 8, pp. 1715–1716, 2005.

[18] M. Horn, M. Tobiasz, and C. Shen, "Visualizing Biodiversity with Voronoi Treemaps," in *Proceedings of the Sixth Annual International Symposium on Voronoi Diagrams in Science and Engineering*, 2009.

[19] C. Ware, *Information Visualization: Perception for Design*, 2nd ed. San Francisco, CA: Morgan Kaufmann, 2004.

[20] B. Shneiderman, *Designing the User Interface*, 3rd ed. Reading, MA: Addison-Wesley, 1998.