Interactive Visualization of Higher Order Nearest Neighbors and Regions

Higher Order Information Visualization

Ye Wang, Kyungmi Lee & Ickjai Lee
School of Business (IT)
James Cook University
Cairns, QLD 4870, Australia
ye.wang@my.jcu.edu.au, joanne.lee@jcu.edu.au, ickjai.lee@jcu.edu.au

***Abstract*—This paper proposes various visual analytics approaches for *k*-nearest neighbor queries and k^{th} order region queries based on higher order Voronoi families. Proposed approaches not only reveal required geometrical information, but various topological intra-, inter-, and cross-relations.**

***Index Terms*—*k*-nearest neighbors, higher order Voronoi diagram, interactive visualization.**

I. Introduction

The *k*-Nearest Neighbor Query (*k*NNQ) is a fundamental problem that has been extensively studied in the literature [1,2]. For a given set *P* of points and a set *Q* of queries, it is to find *k* Nearest Neighbors (*k*NN) from points in *P* for each query *q* in *Q*. It has been one of the most popular queries in many areas such as spatial databases [3], classification [4], pattern recognition [1], disaster management [5], and data mining [6]. One example would be to find *k* nearest evacuation centers for a given location in emergency management in order to reduce casualties and damages. It is especially useful when the first (*k*-1) neighbors are mal-functioning, fully booked, closed, or inaccessible.

*k*NNQ is complemented by the k^{th} Order Region Query (*k*ORQ) that returns a region *R* for a certain subset $P_i^{(k)}$ in *P*, where the set of *k*NN for any location *l* in *R* is the same as $P_i^{(k)}$. It is especially useful for finding a region with similar nearest neighbor characteristics. It has been widely used in city planning, location optimization, market analysis, and network zoning [1,7].

Both *k*NNQ and *k*ORQ are modeled by Higher Order Voronoi Diagram (HOVD) families (order-*k* Voronoi diagram, ordered order-*k* Voronoi diagram and k^{th}-nearest Voronoi diagram), that are generalizations of the Ordinary Voronoi Diagram (OVD) [1,8]. Previous work focused on algorithmic computation approaches [9,10,11] and data structural approaches [12,13], and visual analytical aspects of HOVD families have attracted less attention. Recently, there have been some approaches proposed in the literature to visualize HOVD [14,15,16]. However, they are image-based raster approaches that highlight geometrical information (such as locations of *k*NN) but failed to stress topological relations (relationship between *k*NNQ and *k*ORQ, or relationship among *k*ORQ for various *k* values). In visual analytics, it is much more important to enable a user to explore, understand, reveal and analyze inherent generic topological relations. A rich set of topological relations could not be modeled by raster Voronoi diagrams, but rather modeled by vector Voronoi diagrams. Recently, Wang *et al.* [15], attempted to visualize ordered order-*k* Voronoi diagrams using a vector Voronoi approach with different colors and line widths. However, the approach is limited to a certain set of topological relations, and unable to interactively support a rich set of topological information including *k*NNQ and *k*ORQ.

This paper investigates effective and efficient interactive visualization approaches for *k*NNQ and *k*ORQ through HOVD families. In addition, proposed analytical visualization approaches help users explore various geometrical and topological relations among nearest neighbors, between nearest neighbors and k^{th} order regions, and among diverse k^{th} order regions. This paper proposes an interactive visualization framework for *k*NNQ and *k*ORQ based on a unified HOVD data structure from which a complete set of HOVD families could be drawn [5]. The framework enables users to effectively explore hidden nearest neighbor structures and relations from various HOVDs for a given dataset. Contributions of the proposed framework are: 1) to explicitly support unordered *k*NNQ and *k*ORQ through the complete set of order-*k* Voronoi diagrams; 2) to explicitly support ordered *k*NNQ and *k*ORQ through the complete set of ordered order-*k* Voronoi diagrams; 3) to explicitly support k^{th} order *k*NNQ and *k*ORQ through the complete set of k^{th}-nearest Voronoi diagrams; 4) to effectively support (ordered, unordered, and/or k^{th} order) *k*NNQ and *k*ORQ for multiple *k* values. In particular, the last contribution is of great significance since it allows users to explore what-if *k*NN and *k*OR analysis for different *k* values. For instance, users are able to explore relationships between ordered 3NN and unordered 2OR.

This paper is organized as follows. Section II reviews background preliminaries and previous work to draw problem statements. Section III introduces a proposed interactive visualization framework for *k*NNQ and *k*ORQ through HOVD families. Section IV discusses various proposed visual analytical approaches. Section V concludes with final remarks.

II. Preliminaries and Related Work

A set of various Voronoi diagrams provides a robust framework for both *k*NNQ and *k*ORQ. This section briefly overviews the ordinary Voronoi diagram, and its higher order generalizations. It then reviews algorithmic aspects and visual analytical approaches of higher order Voronoi generalizations.

A. Higher Order Voronoi Diagrams

The Ordinary Voronoi Diagram (OVD) is a space tessellation that naturally divides the study region S into mutually exclusive and collectively exhaustive subregions. For a given set of, the OVD V is a set of $\{V(p_1),\ldots,V(p_n)\}$ Voronoi regions where each $V(p_1)$ (for $p_1 \in P$) is the set of locations that are closer to p_1 than any other location in S. OVD is structured to effectively support 1NNQ and 1ORQ.

The Order-*k* Voronoi Diagram (O*k*VD) is a popular generalization of OVD that captures unordered *k*NN and structured to support unordered *k*NNQ and unordered *k*ORQ. O*k*VD (denoted by $V^{(k)}$ is a set of all order-*k* Voronoi regions $\{ V(P_1^{(k)}),\ldots, V(P_l^{(k)})\}$, where the order-*k* Voronoi region $V(P_i^{(k)})$ ($V(P_i^{(k)}) \neq \phi$) 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 *k*NN are unordered in O*k*VD. Therefore it is useful when order is of no importance in *k*NNQ and *k*ORQ, or when order-generated complexities need to be eliminated. The Ordered Order-k Voronoi Diagram (OO*k*VD, $V^{\langle k\rangle}$) models order and it is set of ordered order-*k* Voronoi regions $\{ V(P_1^{<k>}),\ldots, V(P_m^{<k>})\}$ ($V(P_i^{<k>}) \neq \phi$), where $m = n(n\text{-}1),\ldots,(n\text{-}k\text{+}1)$ and the ordered order-*k* Voronoi region of $P_m^{<k>} = \{p_{i1,\ldots,} p_{ik}\}$ defined as:

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

$V(P_1^{<k>})$ is a refinement of $V(P_1^{(k)})$, namely $V(P_1^{<k>}) = \cup_{Pj<k>\in A<k>(Pi<k>)} V(P_j^{<k>})$, where $A^{<k>}(P_i^{(k)})$ is the sequence of all possible *k*-tuples made of $p_{i1,\ldots,}$ p_{ik} [7]. Even though OO*k*VD is more complex and harder to visualize than O*k*VD, it effectively supports ordered *k*NNQ and unordered *k*ORQ.

B. Algorithmic Approaches

Several algorithmic approaches have been proposed to efficiently calculate O*k*VD [9,10,11]. They are grouped into two clusters: with data structure and without data structure. As an example of the former (with data structure), [5] proposed an $O(n^4)$ time algorithm based on a Delaunay triangle-based data structure from which the complete set of order-*k* Voronoi diagrams is derived. Recently, Lee and Lee [10] proposed a unified Delaunay-triangle data structure that supports complete sets of all three higher order Voronoi families, O*k*VD, OO*k*VD and *k*NVD for all *k*. Building the data structure requires $O(n^4)$ time, and once it is determined computing the complete set of higher order Voronoi families require $O(n^3 \log n)$ [10].

Several other attempts [9,11] have been made without data structure in order to improve efficiency. These approaches (without data structure) directly build O*k*VD from *P*. These without data structure based approaches require less memory than the data structure based approaches, but do not interactively support data analysis (suitable for visual analytics). Currently, the best known algorithm for an order-*k* Voronoi diagram is $O(k(n\text{-}k)\log n + n\log^3 n)$ [17].

The unified Delaunay triangle-based data structure is a robust candidate for visual analytics since it dynamically supports what-if analysis and interactive visual analysis. The structure supports complete sets of HOVD families that provide answers for various *k*NNQ and *k*ORQ.

C. Visualization Approaches

Even though *k*NNQ and *k*ORQ are useful for many applications, most research has focused on either algorithm aspects, or application areas of *k*NNQ and *k*ORO. Visual analytical aspects of *k*NNQ and *k*ORQ have received less attention. Similarly, visual analytical aspects of HOVDs have received relatively poor attention due to their complex nature [14,15,16]. Interactive visualization combines human interaction with computers to enhance focused visual illustrations of information. It is good to prune away unnecessary information, and to focus on interested areas to spotlight. Its pruning capability is well suited for data-rich environments and complex datasets, and a solid candidate for interactive constrained data mining [18]. A combination of interactive visualization and HOVD is a good marriage since it utilizes the pruning capability of interactive visualization to focus on interested information of HOVD.

Generally, there are four types of information in HOVD: generators (points), Voronoi vertices, Voronoi edges, and Voronoi regions. Two types, generators and Voronoi regions, are directly used for *k*NNQ and *k*ORQ respectively. *k*NNQ returns a subset of generators whilst *k*ORQ reports one requested Voronoi region from HOVD families. A Voronoi vertex is formed when a location l in S is equidistant from more than 2 generators whilst a Voronoi edge is created when l is equidistant from exactly 2 generators. These Voronoi vertices and edges are parts of corresponding Voronoi regions and they have a tight relationship with Voronoi regions and thus with *k*NNQ and *k*ORQ.

There has been some research on HOVD visualization, but not in relation to *k*NNQ and *k*ORQ. Palmer [14] investigates visual aspects of O*k*VD (highlighting unordered *k*ORQ) through contour lines and textures based on raster O*k*VD. He introduces order-*k* plots (overlaying nearest neighbor plots for 1,…,*k*-1). That is order-3 plots 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. Since it is based on raster Voronoi diagrams (without data structure model) and also does not support interactive visualization, it is neither well suited for visual analytics, nor dynamic what-if analysis. Another approach [15] attempted to visualize O*k*VD regions using color and shading (with cushions (different intensity levels to give shading to represent the distance to the closest site. Brighter as close 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) non-interactivity; 2) based on raster Voronoi diagrams, thus incapable of visualizing rich topological relations; 3) based on without data structure model that is not well suited for dynamic what-if analysis; 4) only for visualizing one O*k*VD for a given *k*; 5) unable to visualize various topological relations (for instance the relationship between 4NNQ and 5NNQ) of *k*NNQ for various *k*; 6) lack of visual analytics support.

III. Interactive Visualization Framework

A. Visualization Framework

This paper is based on the unified Delaunay triangle based data structure which consists of a complete set of Order-*k* Delaunay triangles (from Order-0 to Order-(*k*-1)) [5]. A complete set of HOVD families could be drawn from this data structure, and the relationship between the data structure and HOVD families is shown in Table I. For example, O4VD can be obtained by a combination of Order-2 and Order-3 triangles, OO4VD is an overlay of four O*k*VDs (from O1VD to O4VD), whilst 4NVD is a join of O3VD and O4VD.

TABLE I. The relationship between HOVD families and the unified data structure

Order *k*	O*k*VD	Oo*k*VD	*k*NVD
1	Order-0 triangle	O1VD	O1VD
2	Order-0 triangle & Order-1 triangle	O1VD ~ O2VD	O1VD & O2VD
3	Order-1 triangle & Order-2 triangle	O1VD ~ O3VD	O2VD & O3VD
4	Order-2 triangle & Order-3 triangle	O1VD ~ O4VD	O3VD & O4VD
…	…	…	…
k	Order-(*k*-2) triangle & Order-(*k*-1) triangle	O1VD ~ O*k*VD	O(*k*-1)VD & O*k*VD

The Delaunay triangulation is a dual graph of the OVD. It can be obtained by connecting two adjacent Voronoi generators when they share a Voronoi edge together. The circumcircle of each Delaunay triangle in the triangulation does not contain any other generator in it. Delaunay triangles for the OVD are named Order-0 triangles. However, there could be triangles whose circumcircles contain a certain number of generators within them. Order-1 triangles are those triangles whose corresponding circumcircles contain only one generator in it. Similarly, Order-*k* triangles are those ones whose corresponding circumcircles contain *k* generators in it.

Figure 1 displays a screen capture of various HOVDs for a given *P*. The framework is implemented in such a way that it enables users to open as many windows as they want to compare and contrast different HOVDs for various *k* values. Since all HOVDs are drawn from the same dataset, the framework enables users to explore various relations within HOVDs. It supports what-if analysis, and also *k*NNQ and *k*ORQ. Users could interact with the program to retrieve *k*NN, *k*OR, Voronoi vertices, and Voronoi edges for any user-interested location within the study region.

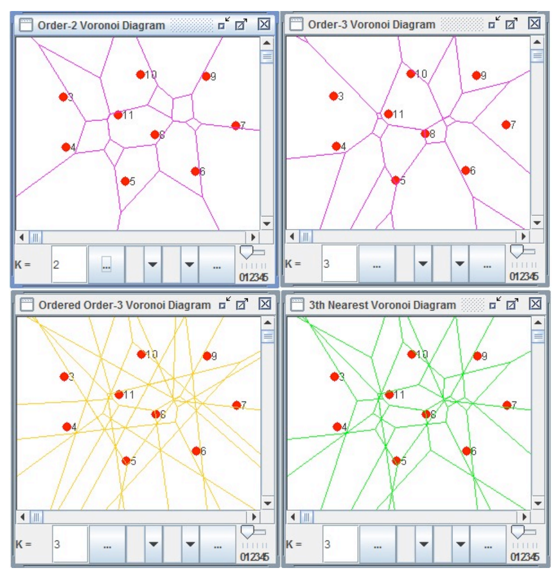

Fig. 1. Various HOVDs ($|P| = 9$).

B. Visualization Elements

There are two elements for visualization: geometrical information and topological information. The former encompasses generators, Voronoi regions, Voronoi edges, and Voronoi vertices whilst the latter consists of three relations: intra-, inter-, and cross-relations. *Intra-relations* are those within one type of HOVD for a given *k*. That is, given O3VD all topological relations within O3VD are intra-relations. For instance, a relationship between unordered 3NN and unordered 3OR is an example of intra-relation. *Inter-relations* are those within the same type of HOVD families for all *k* values. For instance, a relationship between O4VD and O6VD, or between 2NVD and 3NVD is an instance of inter-relation. *Cross-*

relations are those between different types of HOVD families. For instance, a relation between O2VD and 5NVD is a cross-relation example.

IV. VISUALIZATION APPROACHES

A. Voronoi Areas

Voronoi areas are resources for both *k*NNQ and *k*ORQ. This paper introduces three approaches.

1) *K-Region:* This approach utilizes colors and texture to highlight regional information for a user clicked reference point. It is designed to provide information for both *k*NNQ and *k*ORQ at the same time. Given a user clicked reference location (yellow in color in figures), it displays *k*NNs with a highlighted color (in this case green) for *k*NNQ, and shows highlighted Voronoi regions for *k*ORQ. The program is background Voronoi diagram aware (dependent). This approach shows unordered *k*NNs for O*k*VD, ordered *k*NNs for OO*k*VD, and k^{th} NN for *k*NVD. This context-aware interactive visualization supports what-if analysis and comparative analysis. This texture approach also displays Voronoi regions for *k*ORQ along with *k*NN details. For O*k*VD, it highpoints corresponding order-*k* Voronoi region along with order-1 Voronoi region for a given user clicked reference point. This enables users to observe how the order-1 Voronoi region is involved in forming the order-*k* Voronoi region. Similar observations could be made in OO*k*VD and *k*NVD. For OO*k*VD, this approach highlights order-k Voronoi region and order-1 Voronoi region whilst for *k*NVD it shows order-(*k*-1) Voronoi region and order-1 Voronoi region. This context-aware information is summarized in Table II.

TABLE II. K-REGION VISUALIZATION

	*k*NNQ	*k*ORQ
O*k*VD	Unordered *k*NN	Order-*k* region & Order-1 region
OO*k*VD	Ordered *k*NN	Order-*k* region & Order-1 region
*k*NVD	k^{th} order NN	Order-(*k*-1) region & Order-1 region

Figure 2 displays K-Region outputs for various HOVDs

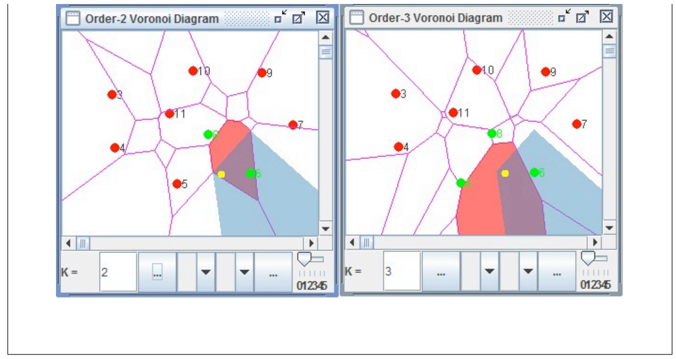

Fig. 2. K-Region representations: O2VD (left); O3VD (right).

2) *K-Circle*: In some cases, users might be interested in understanding the relative difference between the first nearest neighbor and k^{th} nearest neighbor. The region and contour approaches do not meet this requirement. K-Circle approach is to display *k* nearest circles (circles with radii between a clicked point and *k*NN) with a darker color for the nearest neighbor. Filled circle colors become gradually lighter as *k* grows. Figure 3 (right) shows an example with 3NVD.

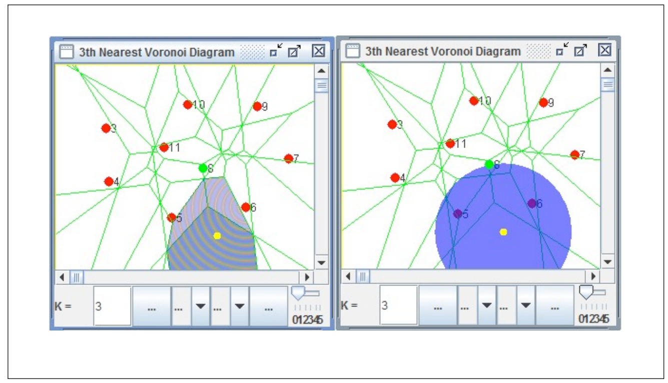

Fig. 3. K-Contour and K-Circle representations.

B. Voronoi Edges

1) *K-Edge*: This K-Edge approach is similar to the K-Region approach in the sense that both use the same information to visualize. The K-Region approach focuses on Voronoi areas whilst the K-Edge approach emphasizes Voronoi edges along with Voronoi vertices. This approach reveals relations between *k*NNs and corresponding Voronoi edges and vertices. This approach utilizes different colors and line widths to spotlight. Figure 4 shows K-Edge representations with O2VD and OO3VD.

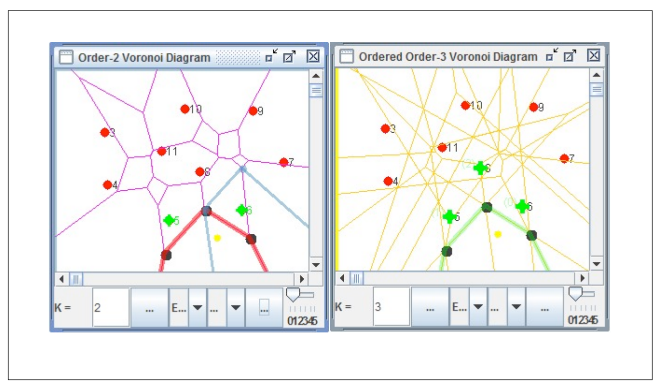

Fig. 4. K-Edge representations of O2VD (left) and OO3VD (right).

2) *K-VoEdge*: As shown in Table I, there are tight relationships among HoVDs. Aforementioned approaches are limited to intra-/inter-relations. K-VoEdge is designed to highlight cross-relations.

It superimposes O*k*VD to all opened windows to highlight cross-relations between O*k*VD and HOVDs in opened windows. The value of *k* in O*k*VD is determined by where a user clicks a reference point. If the user clicks in 4NVD, then O4VD is overlaid. Figure 5 illustrates an example of K-VoEdge with 6 windows created: O2VD (top left), O3VD (top middle), O4VD (top right), OO3VD (bottom left), 2NVD (bottom middle) and 3NVD (bottom right). Each Voronoi diagram in each window is overlaid with highlighted O2VD and *k*NNs.

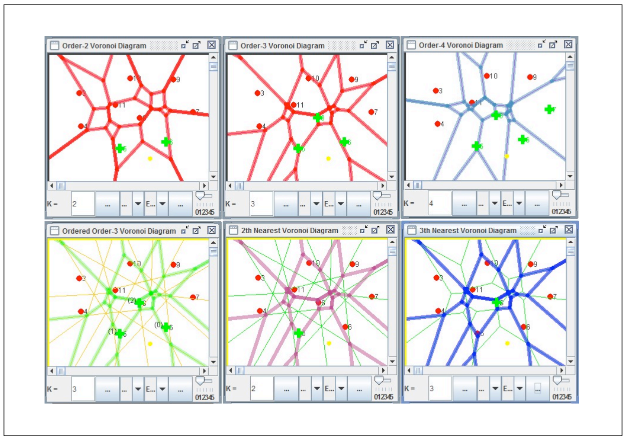

Fig. 5. K-VoEdge representations.

C. Voronoi Vertices

1) *K-Vertex*: K-Vertex approach is designed to show relationships between corresponding Voronoi vertices and *k*NNs. Figure 6 depicts K-Vertex representations with the same number and type of diagrams in Figure 5.

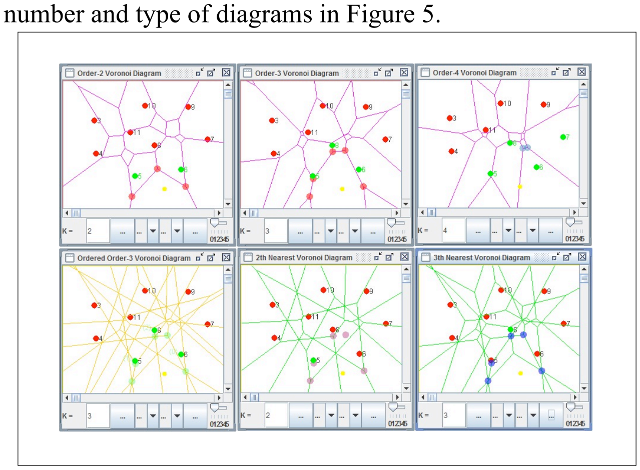

Fig. 6. K-Vertex representations.

D. Mix

In this subsection, approaches are combined to provide more information. Three approaches are developed. Since K-Circle approach uniquely represents differences among *k*NN, it is combined with K-Region, K-Contour, and K-Edge approaches. However, any combination of previous approaches is possible and can be implemented if needed.

1) *K-CiRegion*: K-CiRegion approach combines K-Circle and K-Region together. This combined approach clearly reveals the relationship between the *k* nearest circle and K-Region information. Figure 7 (left) displays an example with O2VD.

2) *K-CiContour*: K-CiContour approach combines K-Circle and K-Contour. Similar to K-CiRegion, this approach discloses the relationship between the *k* nearest circle and K-Contour details as shown in Figure 7 (middle).

3) *K-CiEdge*: K-CiEdge combines K-Circle and K-Edge. This approach also unveils the relationship between the *k* nearest circle and Voronoi vertices. Thus, users could easily explore how many Voronoi vertices lie within the *k* nearest circle that reveals the relationship between the user clicked reference point and its *k*NN. It is shown in Figure 7 (right).

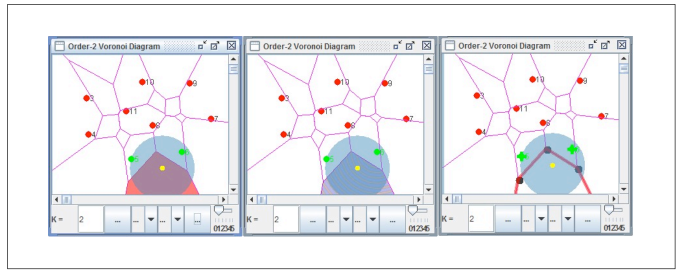

Fig. 7. Mixed approach representation.

V. Final remarks

As we move from data-poor environments to data-rich environments, visual analytics has become more important. This paper proposes various visual analytical approaches for two popular queries *k*NNQ and *k*ORQ. These approaches enable a user to prune irrelevant unnecessary information, but to focus on interested information. The proposed approaches are based on the unified data structure, thus they effectively and efficiently support what-if analysis and comparative analysis. These data structure based approaches not only provide a rich set of geometrical information, but support topological intra-/inter-/cross-relations.

Future work includes its extension to various other types of kNNQ such as reverse nearest neighbor queries [2]. Tight integration to visual data mining is another area of exploration in order to enhance efficiency side of HOVD visualization.

References

[1] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, Second Edition, John Wiley & Sons, New York, 2001.

[2] N. Bhatia, and Vandana, "Survey of Nearest Neighbor Techniques", *International Journal of Computer Science and Information Security*, 2010, Vol. 8(2), pp: 302-305.

[3] T.M. Cover, and P.E. Hart, "Nearest Neighbor Pattern Classification", *IEEE Transactions on Information Theory*, 1967, Vol. 13(1), pp: 21-27.

[4] D. Liu, E.P. Lim, and W.K. Ng, "Efficient *k* Nearest Neighbor Queries on Remote Spatial Databases Using Range Estimation", In *Proceedings of the 14th International Conference on Scientific and Statistical Database Management*, IEEE Computer Society, 2002, pp: 121-130.

[5] 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*, 2009, Vol. 33(2), pp: 90-99.

[6] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.H. Zhou, M. Steinbach, D.J. Hand, and D. Steinberg, "Top 10 Algorithms in Data Mining", *Knowledge Information Systems*, 2007, Vol. 14, pp: 1-37.

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

[8] A. Okabe, B.N. Boots, and K. Sugihara, "Nearest Neighbourhood Operations with Generalized Voronoi Diagrams: A Review", *International Journal of Geographical Information Systems*, 1994, Vol. 8(1), pp: 43-71.

[9] 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*, ACM Press, 1991, pp: 142-151.

[10] 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*, Lecture Notes in Computer Science 154, Springer, 1983, pp: 160-172.

[11] D.T. Lee, "On k-Nearest Neighbor Voronoi Diagrams in the Plane", *IEEE Transactions on Computers*, 1982, Vol. 31(6), pp: 478-487.

[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*, 2002, Vol. 6(2), pp: 89-114.

[13] M. Gahegan, and I. Lee, "Data Structures and Algorithms to Support Interactive Spatial Analysis Using Dynamic Voronoi Diagrams", *Computers, Environments and Urban Systems*, 2000, Vol. 24(6), pp: 509-537.

[14] J.D. Palmer, "Using Line and Texture to Visualize Higher-Order Voronoi Diagram", In *Proceedings of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering*, IEEE Computer Society, 2006, pp: 166-172.

[15] A. Telea, and J. van Wijk, "Visualization of Generalized Voronoi Diagrams", In *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*, Springer, 2001.

[16] Y. Wang, K. Lee, and I.Lee, "Representing Ordered Order-k Voronoi Diagrams", In *Proceedings of the 7th International Conference on Advanced Information Management and Service*, IEEE Computer Society, 2012, pp: 71-75.

[17] M.T. de Berg, M.J. van Kreveld. M.H. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Second Edition, 2000, Springer-Verlag, Heidelberg.

[18] J. Sander, M. Ester, H.P. Kriegel, M. Wimmer, and X. Xu, "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications", *Data Mining and Knowledge Discovery*, 1998, Vol. 2(2), pp. 169-194.