Master’s Thesis

Application of $k$-order $\alpha$-shapes to
geospatial data processing and
time series analysis

Mikko Nikkilä

November 18, 2015
Often it would be useful to be able to calculate the “shape” of a set of points even though it is not clear what is formally meant by the word “shape.” The definitions of shapes presented in this thesis are generalizations of the convex hull of a set of points $P$ which is the smallest convex set that contains all points of $P$.

The $k$-hull of a point set $P$ is a generalization of the convex hull. It is the complement of the union of all such halfplanes that contain at most $k$ points of $P$. One application of the $k$-hull is measuring the data depth of an arbitrary point of the plane with respect to $P$. Another generalization of the convex hull is the $α$-hull of a point set $P$. It is the complement of the union of all $α$-disks (disks of radius $α$) that contain no points of the set $P$ in their interior. The value of $α$ controls the detail of the $α$-hull: $0$-hull is $P$ and $∞$-hull is the convex hull of $P$. The $α$-shape is the “straight-line” version of the $α$-hull. The $α$-hull and the $α$-shape reconstruct the shape in a more intuitive manner than the convex hull, recognizing that the set might have multiple distinct data clusters. However, $α$-hull and $α$-shape are very prone to outlier data that is often present in real-life datasets. A single outlier can drastically change the output shape. The $k$-order $α$-hull is a generalization of both the $k$-hull and the $α$-hull and as such it is a link between statistical data depth and shape reconstruction. The $k$-order $α$-hull of a point set $P$ is the complement of the union of all such $α$-disks that contain at most $k$ points of $P$. The $k$-order $α$-shape is the $α$-shape of those points of $P$ that are not included in any of the $α$-disks. The $k$-order $α$-hull and the $k$-order $α$-shape can ignore a certain amount of the outlier data which the $α$-hull and the $α$-shape cannot. The detail of the shape can be controlled with the parameter $α$ and the amount of outliers ignored with the parameter $k$. For example, the $0$-order $α$-hull is the $α$-hull and the $k$-order $∞$-hull is the $k$-hull.

One possible application of the $k$-order $α$-shape is a visual representation of spatial data. Multiple $k$-order $α$-shapes can be drawn on a map so that shapes that are deeper in the dataset (larger values of $k$) are drawn with more intensive colors. Example datasets for geospatial visualization in this thesis include motor vehicle collisions in New York and unplanned stops of public transportation vehicles in Helsinki.

Another application presented in this thesis is noise reduction from seismic time series using $k$-order $α$-disks. For each time tick, two disks of radius $α$ are put above and below the data points. The upper disk is pulled downwards and the lower disk upwards until they contain exactly $k$ data points inside. The algorithm’s output for each time tick is the average of the centres of these two $α$-disks. With a good choice of parameters $α$ and $k$, the algorithm successfully restores a good estimate of the original noiseless time series.
Contents

1 Introduction  .................................................. 3

2 The shape of a set of points  .................................... 5
  2.1 Convex hull .................................................. 5
  2.2 $\alpha$-hull and $\alpha$-shape  ................................. 6
     2.2.1 Voronoi diagram and Delaunay triangulation .............. 8
     2.2.2 Problems with $\alpha$-hull and $\alpha$-shape ............... 9
  2.3 $k$-hull .................................................... 10
  2.4 $k$-order $\alpha$-hull and $k$-order $\alpha$-shape ............... 11
  2.5 Demonstrating the hulls with icecream ....................... 15

3 Visualizing spatial data  ........................................ 17
  3.1 Motor vehicle collisions ................................... 17
     3.1.1 Comparing the method with heatmaps .................... 17
  3.2 Public transport ............................................ 20

4 Analyzing seismic time series  ................................. 23
  4.1 Choosing the parameters .................................... 25

5 Conclusion  .................................................. 27

References  ..................................................... 29
1 Introduction

If the points of a point set are arranged in a certain way, one might be able to recognize shapes that are formed by the points. Figure 1 shows some examples of point sets that may or may not have meaningful shapes. However, the word “shape” has a very subjective meaning and different people may interpret the word differently. It is not clear what is the formal meaning of the word “shape.” Nowadays most calculations are done with a computer, but without a formal definition, how to tell the computer what points or areas are parts of the shape of a set of points? In real life, these kinds of datasets can be, for example, measurements of abstract variables, lists of geographic coordinates, laser scanings of concrete objects, and so on.

In chapter 2, I will discuss about possible definitions of “shape” by starting with the convex hull of a set of points and then showcasing various generalizations of the convex hull that have such attributes that make them look more like the intuitive interpretation of the word “shape.” For example, the $\alpha$-hull and $\alpha$-shape are capable of recognizing more detailed features of the point set and the $k$-hull ignores the outermost layers of points, showing us what is inside the convex hull. Krasnoshekov and Polishchuk [8] have recently introduced $k$-order $\alpha$-hull and $k$-order $\alpha$-shape which combine the benefits of $\alpha$-hull and $k$-hull. They can be used to reveal the inner shape of the point set and to ignore a certain amount of unwanted outlier data points.

Chapters 3 and 4 are about some applications of the idea of $k$-order $\alpha$-shape to data from real world. In chapter 3, $k$-order $\alpha$-shapes are used to visualize geospatial data, and chapter 4 deals with reducing noise from seismic time series.

This thesis discusses mainly the two-dimensional versions of shapes but the definitions of the shapes are quite straightforwardly extendable to three dimensions as well. It is easy to come up with possible applications of shapes also in three dimensions. For example, Edelsbrunner and Mücke [4] have used the three-dimensional $\alpha$-shape to study molecular structures and the distribution of galaxies in our universe.
Figure 1: Examples of sets of points.
2 The shape of a set of points

In the following subchapters I will discuss about some tools that may be useful in reconstructing meaningful shapes from sets of points.

2.1 Convex hull

The convex hull of a set $P$ is the smallest convex set that contains all points $p \in P$. Another equivalent definition of the convex hull of the set $P$ is the complement of the union of all such halfspaces that contain no points of the set $P$. An intuitive way to think about the convex hull would be a tight rubber band surrounding all the nails that stick out from a board. A visual representation of a set of points and its convex hull are shown in figure 2.

The convex hull has many applications in various fields. For example, in computer graphics one obvious example is speeding up collision detection algorithms. Calculating intersections of convex polygons is often much faster than calculating intersections of arbitrary sets. If the convex hulls do not intersect, then the original sets do not intersect either.

There are many algorithms to find the convex hull, such as Graham’s scan, Jarvis’s march, quickhull, and mergehull. While computing the convex hull is an interesting problem by itself,

![Figure 2: A set of points (a) and the convex hull of the same set of points (b).](image)

5
it is also the first step of many other algorithms in computational geometry. [2, 12]

The convex hull obviously is not very good in describing detailed shapes of point sets but it gives some idea of where the points are. However, just a few additional points can significantly change the output of the convex hull.

Kirkpatrick and Seidel [7] have shown that using their “ultimate planar convex hull algorithm,” the worst-case time complexity of convex hull is $O(n \log h)$ where $n$ is the number of input points and $h$ is the number of points on the boundary of the hull.

2.2 $\alpha$-hull and $\alpha$-shape

Edelsbrunner, Kirkpatrick, and Seidel [3] have introduced $\alpha$-shape to eliminate the informal human perception from the meaning of the word “shape.”

A concept closely related to the $\alpha$-shape is the $\alpha$-hull. The $\alpha$-hull of a point set $P$ is the complement of the union of all $\alpha$-disks (disks of radius $\alpha$) that contain no points of the set $P$ in their interior. For an illustration of $\alpha$-hull, see figure 3 (a). The value of $\alpha$ controls the detail of the $\alpha$-hull. When $\alpha = \infty$, the $\alpha$-hull becomes the convex hull - the $\alpha$-hull therefore is a generalization of the convex hull. If $\alpha$ is very small, the $\alpha$-disks can fit between the points and the $\alpha$-hull becomes $P$.

The $\alpha$-hull may not be very satisfying interpretation of the point set’s shape because of its curvy boundary. The $\alpha$-shape of the set $P$ is a “straight-line” version of the $\alpha$-hull of $P$. For all empty $\alpha$-disks that have two distinct points of $P$ on its boundary, an edge of the $\alpha$-shape is drawn as a straight line between those two points. See figure 3 (b) for an example of $\alpha$-shape and comparison with $\alpha$-hull of figure 3 (a). The time complexity of $\alpha$-shape is $O(n \log n)$.

As can be seen from figure 4, the $\alpha$-shape reconstructs the shape in a more intuitive manner than the convex hull, recognizing that the set might have multiple distinct data clusters.

Edelsbrunner and Mücke [4] have generalized the $\alpha$-shape also for three-dimensional point sets.
Figure 3: The $\alpha$-hull (a) and the $\alpha$-shape (b) of a set of points.

Figure 4: Comparison between the convex hull (a) and the $\alpha$-shape (b) of the same 2-clustered point set. The $\alpha$-shape recognizes the two distinct data clusters while the convex hull does not.
2.2.1 Voronoi diagram and Delaunay triangulation

The Voronoi diagram of a set of points \( P \) is a partitioning of the plane so that for each point \( p_i \in P \), the corresponding Voronoi region \( V_i \) contains all points of the plane that are closer to \( p_i \) than to any other point \( p_j \in P \). A set of points with its Voronoi diagram is shown in figure 5.

![Voronoi Diagram](image)

**Figure 5:** The Voronoi diagram of a set of points.

Inserting an edge between each two points \( p_i, p_j \in P \) of adjacent Voronoi regions \( V_i, V_j \) produces the Delaunay triangulation which is a special kind of graph that consists only of triangles. The point set from figure 5 is shown with its Delaunay triangulation in figure 6. Delaunay triangulation of the point set \( P \) has a useful property that the circumscribed circle of each triangle in the Delaunay triangulation does not contain any other points of \( P \).

Edelsbrunner et al. [3] have shown that there exists an interesting relationship between \( \alpha \)-shapes and Delaunay triangulations. Any \( \alpha \)-shape of a set of points \( P \) is a subgraph of...
the Delaunay triangulation of the set $P$. This result allows to compute $\alpha$-shapes much more efficiently than doing a brute-force check for all pairs of points.

2.2.2 Problems with $\alpha$-hull and $\alpha$-shape

In the real world, data often contains unwanted outlier points. These can be caused by background noise or measurement errors and they are often better to be ignored when reconstructing the shape of the data. It turns out that $\alpha$-hull and $\alpha$-shape are very prone to these outliers. Figure 7 demonstrates how an outlier point drastically changes the features of the $\alpha$-shape of a point set. Because of this behaviour, $\alpha$-hull and $\alpha$-shape are not that good tools if the data contains outlier points. However, there exist generalizations that can be used when dealing with data that contains outliers. One of these tools will be described in chapter 2.4.
Figure 7: Subfigure (a) illustrates the $\alpha$-shape of a set of points. In subfigure (b), an outlier data point (drawn in red color) has been added and the $\alpha$-shape fails to detect the hole feature of the original point set.

2.3 $k$-hull

Another generalization of the convex hull is the $k$-hull, also known as the $k$-depth contour, introduced by Cole, Sharir, and Yap [1]. The $k$-hull of a point set $P$ is the complement of the union of all such half-planes that contain at most $k$ points of $P$. With $k = 0$, the $k$-hull is the convex hull. Figure 8 shows a set of points with its $k$-hull with different values of $k$.

One obvious application of the $k$-hull is measuring the interiorness or data depth of an

![Figure 8: The $k$-hull of a set of points with $k = 0 \ldots 2$.](image)
arbitrary point $q \in \mathbb{R}^2$ with respect to $P$. It can be defined as the largest $k$ such that $q$ belongs to the $k$-hull of $P$.

As is the case with convex hull, also $k$-hull might produce unwanted results when the data is divided into multiple clusters. A region between distinct clusters might actually have a high data depth. One possible solution to the problem will be discussed in chapter 2.4.

The worst-case time complexity of $k$-hull in two dimensions is $O(N_k(n) \log^2 k)$ where $N_k(n)$ is the maximum number of $k$-sets for a set of $n$ points in the plane. The time complexity to compute $N_k(n)$ is $O(nk^{1/2})$. [1]

2.4 $k$-order $\alpha$-hull and $k$-order $\alpha$-shape

To solve some of the problems described in chapter 2.2.2 and to provide a link between shape reconstruction and statistical data depth, Krasnoshchekov and Polishchuk [8] have introduced $k$-order $\alpha$-hull and $k$-order $\alpha$-shape that are generalizations of the hulls and shapes discussed in the previous subchapters. One of their useful features is that they can ignore a certain amount of the possibly unwanted outlier data. The amount of outliers ignored can be controlled with the parameter $k$ and the detail of the shape with the parameter $\alpha$.

The $k$-order $\alpha$-hull of a point set $P$ is the complement of the union of all such $\alpha$-disks that contain at most $k$ points of $P$. Just like with $\alpha$-hull, the detail of the output shape can be adjusted by changing the parameter values – $\alpha$ is the radius of a disk and $k$ is the amount of points allowed to be inside each disk. The 0-order $\alpha$-hull is the ordinary $\alpha$-hull and the $k$-order $\infty$-hull is the $k$-hull. As a generalization of both $\alpha$-hull and $k$-hull, $k$-order $\alpha$-hull has the benefits of the both. See figure 9 for an example of $k$-order $\alpha$-hull with different values of $k$.

With $k$-order $\alpha$-hulls, the data depth of an arbitrary point $q \in \mathbb{R}^2$ with respect to $P$ that was discussed in chapter 2.3 can be generalized in a way that takes the possible distinct data clusters into account. The $\alpha$-depth of a point $q \in \mathbb{R}^2$ with respect to $P$ is the largest $k$ such that $q$ belongs to the $k$-order $\alpha$-hull of $P$. If $\alpha = \infty$, the $\alpha$-depth is the same as the depth function from chapter 2.3.

A point $p \in P$ is called $(k, \alpha)$-outside if it can be contained in an $\alpha$-disk that contains at most $k$ points of $P$. Points of $P$ that are not $(k, \alpha)$-outside are called $(k, \alpha)$-inside. The $k$-order $\alpha$-shape of $P$ is the $\alpha$-shape of the points that are $(k, \alpha)$-inside. See figure 10 for an example of $k$-order $\alpha$-shape with various values of $k$.

The $k$-order $\alpha$-shape is a generalization of the ordinary $\alpha$-shape (0-order $\alpha$-shape) and it also has the thinning property of $k$-hull. The edges of $k$-order $\alpha$-shape are straight lines, so
Figure 9: The $k$-order $\alpha$-hull of a set of points with $k = 0 \ldots 3$.

depending on the application it may have certain advantages over $k$-order $\alpha$-hull that has curvy boundary.

Because of their thinning feature, both $k$-order $\alpha$-hull and $k$-order $\alpha$-shape can be used to solve the problems caused by outlier data discussed in chapter 2.2.2. For an example, see figure 11. However, while being better than $\alpha$-hull and $\alpha$-shape in ignoring outliers, $k$-order $\alpha$-hull and $k$-order $\alpha$-shape, depending on the value of $k$, peel off the outermost layers of the data, possibly causing loss of sharp features. This kind of thinning might be either good or bad thing depending on the situation. Using the $k$-order versions also increases computational complexity and so they might require longer time to compute in practice.

Krasnoshchekov and Polishchuk [8] have shown that $k$-order $\alpha$-shapes are related to $k$-order Voronoi diagrams in the same way in which the ordinary $\alpha$-shapes are related to Voronoi diagrams. The worst-case time complexity for both $k$-order $\alpha$-hull and $k$-order $\alpha$-shape is $O(T_{VD}^k + C_{VD}^k \log C_{VD}^k)$ where the time $T_{VD}^k$ is the time to compute the $k$-order Voronoi diagram of $P$ and $C_{VD}^k$ is the complexity of the $k$-order Voronoi diagram.
Figure 10: The $k$-order $\alpha$-shape of a set of points with $k = 0 \ldots 5$. 
Figure 11: The outlier problem of $\alpha$-shapes presented in figure 7 can be solved with $k$-order $\alpha$-shapes. In subfigure (a), an outlier data point (red) was added to the point set and the $\alpha$-shape was not able to recognize the hole feature of the dataset. Subfigure (b) is the 1-order $\alpha$-shape of the same data. The added outlier point no more affects the detection of the shape’s hole feature.
2.5 Demonstrating the hulls with icecream

Imagine an infinite plane $\mathbb{R}^2$ of icecream that has solid chocolate chips scattered around. The convex hull, the $k$-hull, the $\alpha$-hull, and the $k$-order $\alpha$-hull of the set of the chocolate chip points are what remains of the icecream after a person allergic to chocolate has eaten up as much as possible with certain kinds of kitchen utensils. This example is first introduced by Fischer [5] and then further expanded by Krasnoshechekov and Polishchuk [8]. The kitchen utensils in question and their relation to the hulls are illustrated in figure 12.

A knife (a line) cuts the plane into two halfplanes. The person can cut away and eat arbitrary halfplanes that do not contain any chocolate chips. After all possible chocolate-free halfplanes have been eaten, what remains is the convex hull of the chocolate points.

But eating with an ordinary knife is a waste of good icecream! If the eater instead uses a cheeseknife that has $k$ holes in it, it is possible to get more icecream without having to eat the chocolate chips. A halfplane can be cut out if it has up to $k$ chips in it – the chocolate chips are sieved through the holes of the cheeseknife. The icecream that remains after eating everything that is possible to cut out with the cheeseknife is the $k$-hull of the chocolate points. If $k = 0$, the cheeseknife is equal to the knife and the $k$-hull is equal to the convex hull.

Another way to improve the icecream eating efficiency is to use a scoop (a disk) of radius $\alpha$. Using a small scoop (small values of $\alpha$) allows the eater to scoop between the chocolate chips while a big scoop (big values of $\alpha$) might be too large to scoop between the chips. Eating everything that is possible to eat with the scoop produces the $\alpha$-hull. If the scoop is small enough, the eater is able to eat up all of the icecream so that only the chocolate chips remain.

Figure 12: The kitchenware family picture. [8]
If the scoop is huge, the remaining part of the icecream begin to resemble the convex hull.

The ultimate icecream eater’s tool is a colander that has radius of $\alpha$ and $k$ holes. It combines the benefits of both the cheeseknife and the scoop. The idea is that each scoopful contains at most $k$ chocolate chips which are sieved through the colander’s holes. A colander with no holes is a scoop and eating with a colander of infinite radius is like eating with a cheeseknife. What remains after eating as much icecream as possible with the colander is the $k$-order $\alpha$-hull.
3 Visualizing spatial data

Because the $k$-order $\alpha$-shape can be interpreted as the shape of a set of points, one natural way to use it in real life is to make visual representations of datasets, such as spatial data which can be nicely visualized on a map. One possible benefit of good visualizations is that it would be easier to prepare for future events when the geographic locations that are affected by certain phenomena are known.

In this chapter, I will show a few maps from the real world with visualizations where the $k$-order $\alpha$-shape has been applied to the locations of motor vehicle collisions in New York and unplanned stops of public transport vehicles in Helsinki.

3.1 Motor vehicle collisions

In this subchapter, I will present a visualization of the locations of certain motor vehicle collisions in New York using multiple $k$-order $\alpha$-shapes with different parameters.

Figure 13 shows a visualization of 635 motor vehicle collisions that happened in the city of New York on April 17, 2015. The red markers represent the locations of the collisions as provided by New York City Police Department [10]. The figure contains multiple $k$-order $\alpha$-shapes stacked on top each other, drawn with different colors. The shapes share a constant $\alpha$ but their $k$ values are ranging from 0 to 16. The yellow colors indicate small values of $k$ – only a few motor vehicle collisions happen at these parts of the city. Redder colors mean greater values of $k$. The most intensive red is the 16-order $\alpha$-shape which contains the most dangerous collision clusters of the city.

3.1.1 Comparing the method with heatmaps

Visualizing motor vehicle collisions with stacked $k$-order $\alpha$-shapes is an alternative to, for example, using heatmaps. In a simple grid-based heatmap, the region is divided into a uniform grid. Each cell of the grid has a certain number of data points in it and the more points a cell has, the more intensive color is used for it in the final visualization. For comparison, the data from figure 13 visualized with a simple uniform grid-based heatmap can be seen in figure 14.

Heatmaps are obviously faster to generate than multiple $k$-order $\alpha$-shapes, but the results might be interpretable slightly differently. The heatmap method produces shapes that consist of rectangles which might not be very convenient representation of the areas of the city. Neither of the methods takes the layout of the road network into account which might not be the ideal case when dealing with motor vehicle collisions.
Figure 13: Visual representation of motor vehicle collisions in the city of New York on April 17, 2015 using multiple $k$-order $\alpha$-shapes.
Figure 14: Visual representation of motor vehicle collisions in the city of New York on April 17, 2015 using a simple grid-based heatmap.
3.2 Public transport

Packer, Bak, Nikkilä, Polishchuk, and Ship [11] have used the $k$-order $\alpha$-shape in the domain of urban public transportation. They explored the stop patterns of public transportation vehicles, such as trams, in the city of Helsinki. Planned stops at stations are of course required but studying the unplanned stops is important when optimizing the public transportation service.

The authors gathered location data from a live web service provided by Helsinki Region Transport [6]. Whenever a vehicle stood still for at least 10 seconds, a stop was recorded. At the end of the data gathering period of 24 hours they had gathered location data of about 7,000 unplanned stops.

Their method uses $\alpha$-shapes to detect hierarchies of clusters and $k$-order $\alpha$-shape to remove outliers and noise, such as GPS errors, from the dataset.

The distribution of unplanned stops was visualized on the city map with the usual color range from light yellow to dark red, where redder areas contain more data points. See figure 15 for a close look at some of the most interesting parts and figure 16 for the visualization of the whole data on the map of Helsinki.

![Figure 15: Closer look on parts of the $k$-order $\alpha$-shape of figure 16. The arrows in subfigure (c) indicate outlier points that are part of the data but are ignored from the final shape due to the selection of $k$. [11]](image-url)
Figure 16: The distribution of public transportation vehicles in Helsinki visualized with $k$-order $\alpha$-shapes. See figure 15 for closer look on the parts of the figure that are marked with (a), (b), and (c). [11]
4 Analyzing seismic time series

This chapter is about applying the idea of \(k\)-order \(\alpha\)-shapes to analysis of seismic time series that may have a lot of background noise. The algorithm, which has been presented by Nikkilä, Polishchuk, and Krasnoshchekov [9], reconstructs a smooth curve from noisy time series and it is an alternative to moving average and other smoothing techniques.

For each time tick, two disks of radius \(\alpha\) are put above and below the data points. The upper disk is pulled downwards and the lower disk upwards until they contain exactly \(k\) data points inside. The algorithm's output for each time tick is the average of the centres of these two \(\alpha\)-disks. For an illustration, see figure 17.

\[\text{Figure 17: An application of the } k\text{-order } \alpha\text{-shape to time series. The upper disks are pushed downwards and the lower disks upwards until there are exactly } k = 4 \text{ points inside. For each time tick, the output is the average of the centre points of the disks. [9]}\]
The advantage of this algorithm is its robustness. The shape is not affected by small variations of the input points and the algorithm locally ignores the most extreme values of the seismic background noise. The amount of ignored values can be adjusted by the selection of $k$. One possible way to choose reasonable values for $\alpha$ and $k$ is described in chapter 4.1.

Figure 18 shows the output of the $k$-order $\alpha$-shape algorithm applied to a 45 minutes long synthetic seismic dataset with Rayleigh distributed noise.

**Figure 18:** A 45 minutes long synthetic dataset of a seismic event with Rayleigh distributed noise. The black graph is the original shape and the white graph is the shape restored by the $k$-order $\alpha$-shape. [9]
4.1 Choosing the parameters

Before using the previously described algorithm, one must choose reasonable values for the parameters $\alpha$ and $k$. When choosing the parameters, it is important to use a part of the data that only contains background noise. Nikkilä et al. [9] use the following procedure.

1. Set the value of $\alpha$ equal to the standard deviation of the noise.

2. For 1,000 random time ticks, place an $\alpha$-disk so that its centre is $2\alpha$ above the mean level of the noise and another $\alpha$-disk so that its centre is $2\alpha$ below that mean level.

3. Set a candidate value for $k$ as the average number of points inside the 2,000 $\alpha$-disks.

4. Run the algorithm on the noise data with the candidate value of $k$. The $k$-order $\alpha$-disks are supposed to ignore the noise so that the output would be approximately the noise’s mean level.

5. Measure the goodness of the output by using, for example, root-mean-square deviation.

6. If the output is too far from the noise’s mean level, scale the time axis and recompute the value of $k$. Larger scaling implies larger value of $k$ and smaller root-mean-square deviation. The choice of $\alpha$ does not depend on the scale of the time axis. Repeat the procedure until the root-mean-square deviation is small enough, usually between 0.05$\alpha$ and 0.005$\alpha$. Larger scales would cause oversmoothed output.
5 Conclusion

In this thesis, I have studied shapes of sets of points and their applications. In this chapter, I will briefly revisit the previous chapters and write some thoughts about possible future work concerning the subject.

In chapter 1, I discussed about the subjectiveness of the meaning of the word “shape” and the importance of having a formal definition for it.

In chapter 2, I presented some tools for shape reconstruction from a set of points. The simplest of those is the convex hull which is the smallest convex set that contains all of the data points. In most cases, that is not what is meant by the word “shape,” so I presented also $\alpha$-hull and $\alpha$-shape, that reconstruct the shape by removing all empty $\alpha$-disks from the plane, and $k$-hull, which is the complement of the union of all such halfplanes that contain at most $k$ points. I also presented the fairly new concepts of $k$-order $\alpha$-hull and $k$-order $\alpha$-shape that are generalizations of the earlier mentioned hulls and shapes. Instead of empty $\alpha$-disks, they use $\alpha$-disks that have at most $k$ points. They are good in reconstructing the shape of a set of points just like $\alpha$-hull and $\alpha$-shape, but in addition to that, they are also good in ignoring unwanted data such as outliers and measurement errors.

In chapter 3, I showed how to use multiple $k$-order $\alpha$-shapes with different values of $k$ to visualize geospatial data on a map. I made a visual representation of motor vehicle collisions in New York with $k$-order $\alpha$-shapes and I also made a simple uniform grid-based heatmap of the same data for comparison. I also presented the results of the study of the unplanned stops of the public transportation vehicles in Helsinki that I did with Packer, Bak, Polishchuk, and Ship [11].

In chapter 4, I explained a method that I used with Polishchuk and Krasnoleshchekov [9] to remove seismic background noise from seismic time series with $k$-order $\alpha$-disks. The method successfully restores a good estimate of the original time series and it can be used as an alternative to moving average and other smoothing techniques.

In the future, it would be interesting to see more research and applications of the $k$-order $\alpha$-hull and the $k$-order $\alpha$-shape, like visualizations of such data that does not depend on the layout of the road network, also in other than geospatial context. It would also be interesting to see more applications on time series analysis and especially how well the algorithm of chapter 4 would do in other fields than seismology.

Although the definition of $k$-order $\alpha$-shape easily extends into three dimensions, would it be easy to compute three-dimensional $k$-order $\alpha$-shapes in practice? There would surely be many real-world applications for the three-dimensional $k$-order $\alpha$-shape as well.
References


