Background

January 12, 2011

As this section will contain descriptions of other work there will be equations from different sources. To retain some clarity the same nomenclature will be used throughout.

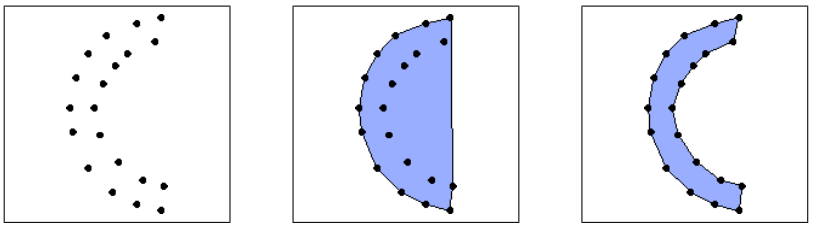
$$
\begin{aligned}
S &= \text{The set of points that constitutes the dot pattern} \\
p &= \text{A point from the set } S \\
F(S) &= \text{The footprint of } S \\
A(F(S)) &= \text{The area of } F(S) \\
n &= \text{The size of set } S \\
DT(S) &= \text{The Delaunay Triangulation of } S \\
V(S) &= \text{The Voronoi Diagram of } S \\
VC(p) &= \text{The Voronoi Cell containing } p \\
\tau &= \text{A threshold value}
\end{aligned}
$$

0.1 Footprints

There is a fairly large body of work about the generation of footprints, publications from as early as 1973 ([13]) presenting a variety of different algorithms to create representational shapes from dot patterns. Amongst this there are surprisingly few that examine the footprints created in a comparative fashion. Also conspicuous by its absence is a systematic approach to determining the quality of the produced footprint, Galton [10] makes significant inroads in to both determining how 'good' a footprint is and why this is difficult to judge.

A discussion of the footprint algorithms should probably begin with one of the first to give an efficient algorithm for its computation in 1973 Jarvis [13] presented an algorithm, since called the 'Jarvis March', to generate the convex hull of a dot pattern. The convex hull is almost a base level of footprint algorithm, it is easily computable and has distinct mathematical properties. Importantly the convex hull is unique for any particular dot pattern.

The convex hull is not without its problems as a representation.



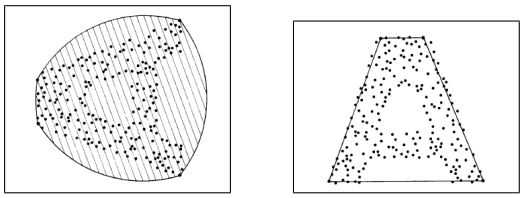
(a) Point set (b) Convex Hull (c) Non-Convex Footprint

Figure 1: When a convex hull is inappropriate

As can be seen in Fig.1(b) the convex hull can potentially lose information about the pattern, whereas Fig.1(c) may be a better approximation of the underlying data. An algorithm capable of reaching a better fit representation is a non-trivial problem and one of the earliest, and much-referenced, papers on the subject is by Edelsbrunner et al. [9]. The method produces straight-line graphs called α-shapes, obtained from a generalisation of the convex hull. For a set S the convex hull can be considered to be the intersection of all closed half-planes that contain all the points of S. Taking a half-plane to be a closed disc of infinite radius, an α-hull can be defined as the intersection of all closed discs with radius $1/\alpha$ that contain all the points of S. Using a radius of $1/\alpha$ allows an approximation of the convex hull[1], to make the α-hull more descriptive a *generalized disc of radius* $1/\alpha$ [2] is defined as a disc of radius $1/\alpha$ if $\alpha > 0$, a halfplane if $\alpha = 0$ and the complement of a disc of radius $-1/\alpha$ if $\alpha < 0$, the α-hull, then, is the intersection of all closed generalized discs of radius $1/\alpha$ that contain all the points of S.

Before the α-shape can be defined some properties of the hulls need to be noted. A point p from the set S is an *α-extreme* if there exists a closed generalized disc of radius $1/\alpha$ such that p lies on its boundary and it contains all the points of S. If two α-extreme points can share the same generalized disc the they are said to be *α-neighbours*. The α-shape is is the straight line graph with vertices at α-extreme points and edges connecting the α-neighbours.

The positive α-shape (where $\alpha > 0$) is clearly a footprint, notable in that it tends to look like an approximation of convex hull save that it is possible for it to not contain all the points Fig.2(b). However the negative α-shape (where $\alpha < 0$) produces far more interesting results as shown in Fig.3(b).



(a) α-hull (b) α-shape

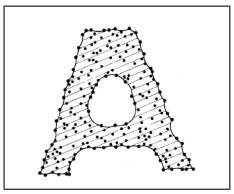
Figure 2: Positive α-hull and shape

This is one of the earliest steps toward an algorithm that is cognitively more 'appropriate' for the dot pattern than the convex hull.

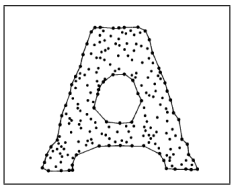
The paper showcases a method for creating α-shapes from the Delaunay triangulation, however describing this here seems unnecessary. As it happens there is one more facet introduced by this paper that is of interest to us when considering footprints and change identifiers. They make note that for a particular point set there is a finite number of α-shapes which they call the *shape spectrum* ($SP(S)$), and that by generalising their algorithm the $SP(S)$ can be found in $O(n \log n)$ time. This type of analysis of the algorithm and its emergent properties is often not found within the literature, however the majority of the work discussed in this chapter does contain it precisely because of its rarity.

[1] Where $f(\alpha)$ is the disc radius as $\lim_{\alpha \to 0} f(\alpha) = \infty$

[2] American generalized instead of the english generalised as this is how it is given in the paper.



(a) α-hull



(b) α-shape

Figure 3: Negative α-hull and shape

Edelsbrunner *et al.* do not comment on how the inherent properties of the dot pattern affect the produced shape, nor do they present any discussion on how to choose α to produce a specific shape from the shape spectrum.

Since this landmark paper much use has been made of α-shapes, in [8] Edelsbrunner and Mücke make note of two of the most interesting applications, namely molecular structure mapping and distributions of a point set. The paper presents an extension of α-shapes into 3D, but they have also been extended to take into account the natural weighting of the point set in [7].

α-shapes require a parameter from which to be formed, this is common amongst all the *non-convex* footprint algorithms. This parameter is required because the idea of the footprint is vaguely defined; any shape that can be said to represent the underlying dot pattern is a valid footprint. This vagueness is necessary because users of the algorithms have different requirements on the type of shape they need and the parameterization allows control over the detail captured/lost within the footprint.

Moving on from α-shapes there are many possible algorithms we could examine, however rather than listing them (for a larger set see [6]) the rest of this section will look for novelties within the literature.

Melkemi [14] came up with a interesting change to the idea of the parameter, for the $\mathscr{A}$-shape[3] the parameter, $\mathscr{A}$, is actually a set of dots. The footprint is then constructed from the Voronoi Diagram of the union of the original dot pattern and $\mathscr{A}$, the footprint being the borders of the cells containing the original dots. The process for choosing $\mathscr{A}$ is not expanded on until [15] in which it is defined as sampled from the union of two sets:

1. Centers of the Delaunay circles associated with the Delaunay triangulation of the original pattern having radii higher than a threshold $\tau \geq 0$.

2. For each edge $\overrightarrow{pq}$ of the convex hull of the original pattern, consider the point not belonging to the convex hull of S and which is the center of the circle passing through p and q and having sufficiently big radii.

[3]In [14] it is referred to as the $\mathcal{A}$-shape but in [15] and [16] it is called $\mathscr{A}$-shape, so we use the most common notation

What is meant by 'sufficiently big' is not elaborated on however we can assume it relates to the same threshold as in the first constraint. Interestingly this turns the original concept, of having a point set as the parameter, into the more common usage of a single numeric value instead. In [16] Melkemi and Djebali introduce the idea of the weighted $\mathscr{A}$-shape, this allows the algorithm to deal with dot patterns containing areas of different densities. Each point is given a weight based on the distance between it and its closest neighbour. The set of points $\mathscr{A}$ is found using the power diagram of the original pattern, suffice to say that it too uses a threshold value much the same as the unweighted version.

Alani et al. [1] also use a point set as their input parameter however theirs is an interesting paper as it is one of the few where the application has directly lead to the development of the algorithm. There exist gazetteers (or geographical thesauri) which combine place name data with limited locational information. Such systems are used for queries such as as requests for all the hotels in a specific area. After noting some of the current constraints on such systems (limited bandwith, differing search terms to index terms, imprecise or precise matching, etc.) they introduce the *Dynamic Spatial Approximation Method* or DSAM. Much the same as the work done by Melkemi and Djebali, it uses the union of the original dot pattern and another set of coordinates to construct the Voronoi Diagram, the footprint being the boundary edges of the union of cells containing the original pattern. Unlike the $\mathscr{A}$-shape the external points can be taken from the data as points known not to exist within the query location. In this instance as the data is already obtained for the database there is no need to give any further complexity to the parameter. Alani *et al.* note that the error of the approximation can be measured in three ways:

- Total areal error – Gives a basic approximation error.
- Visual error – Gives a measure of how different the shapes are. If we take the negative false error as the areas left out of the approximation and the positive false error to be the areas in the approximation not in the expected area then the visual error can be measured as:

  $$V = \frac{A_{pp} + A_{np}}{A_o} 100\%$$

  Where V is the visual error, A_{pp} is the positive false error, A_{np} is the negative false error and A_o is the original error.
- Quality of the spatial relationships – Consistency of the spatial relationships of the actual with the approximated area e.g., if a data point is within the actual is it within the approximation.

This level of reasoned assessment is notably absent from much of the literature. Although this is undoubtedly because Alani *et al.* have an expected shape to measure against, it seems strange that little has been done to give any form of general scoring to the footprints produced.

α-shape, $\mathscr{A}$-shape and DSAM are all of complexity $O(n \log n)$. This is common amongst the footprint algorithms, at least in part because they tend to be generalisations or modifications of existing $O(n \log n)$ algorithms (e.g., Delaunay triangulations, Voronoi diagrams and Jarvis March). Aware of this Chaudhuri et al. [3] propose two methods for extracting the 'perceptual border' of a dot

pattern that have $O(n)$ complexity; the s-shape and the r-shape. The s-shape is generated by laying a grid over the isothetic rectangle of the dot pattern. The union of all the grid cells that contain at least one dot gives the s-shape, s being the length of the grid cell sides. Chaudhuri *et al.* note that choosing s is not a simple task and the interesting component of the method is the fashion in which they deal with this problem. First they note that there are a finite number of different s-shapes that can be created for any dot pattern, by defining the sequence $\langle s \rangle$ as:

$$s_i = \overline{s} \qquad \text{when } i = 1$$

$$s_i = \sqrt{\frac{A(H(s_{i-1}))}{n}} \qquad \text{when } i > 1$$

In which:

$$\overline{s} = \sqrt{\frac{A(H(s_{i-1}))}{n}} \text{ NB:}^4$$

$H(s)$: The footprint (hull) when the grid length is s

$\langle s \rangle$ finishes when each grid cell of the footprint contains only one point. This sequence gives rise to the 's-shape spectrum' of the dot pattern (similar to the work done by Edelsbrunner for α-shapes) and can be done in $O(n)$ time. The next step was to find a way to choose an appropriate s value from the spectrum and to do this they introduce the parameter ε as a measure of disparity within the dot pattern, essentially how uniform the density is across the pattern. s is now chosen from the spectrum by:

$$s = \max\{s_k; |\frac{s_{k-1} - s_{k+1}}{s_k}| \leq \varepsilon, s_{k-1} \epsilon \langle s \rangle\}$$

For the majority of patterns they found that an ε value of 0.3-0.5 was sufficient to achieve a suitable representation. Although it should be noted that they give little discussion on how the representation is measured and do point out that the 'perceptual structure' is not necessarily unique. The s-shape is notable staircase like and lacks details that maybe required within the application. The r-shape however is a much 'smoother' representation. Placing a disc or radius r over each dot and then joining edges between dots that share an exposed point. This is clearly an entirely seperate algorithm from the s-shape and suffers from the difficulty of selecting a suitable value of r. Chaudhuri *et al.* note just this and proceed to show that using the s-shape algorithm you can retrieve a value for r where $r = \sqrt{2s_i}$, combined with the ε-measure of dispersion this gives a $O(n)$ method for producing a footprint with a single parameter and suggested ε value of 0.4. On the surface Chaudhuri *et al.* appear to have covered all the major issues (visual salience, complexity and parameter choice) but they do not explain how they determine whether or not a disc shares an external point with

[4] If the spacing of the dot pattern was completely uniform this would give an optimal value of s.

another Check this bit with their digital implementation as a result the $O(n)$ complexity may not be accurate. Even if the complexity is $O(n)$ the number of iterations and steps indicate that the algorithm may not be particularly fast, although they say for their experiments the number of iterations for the shape spectrum was often under four they don't give the time this took or the number of dots involved. They finish with an approach for dealing with patterns of mixed dispersion measures which breaks the footprint into different density blocks, dealing with each indivdually, essentially leading to a sort of contour map. This is still valid as a footprint but now provides more that just one border, essentially giving layered footprints, it would be interesting to see the idea of different footprint types developed more here.

There appears to be a division in types of footprint algorithms appearing, the α-shape, $\mathscr{A}$-shape and DSAM are all mathematically derived algorithms, they arise from the implementation of easily expressable concepts:

- α-shape – The intersection of all closed discs with radius $1/\alpha$ that contain all the points of S.
- $\mathscr{A}$-shape – The union of the cells containing dots of S from the Voronoi Diagram of $\mathscr{A} \cup S$.
- DSAM – The union of the cells containing dots of S from the Voronoi Diagram of $E \cup S$ where E is a set of dots known to be external to the query area.

However s-shape and r-shape are a bit different. On the surface the basic description of the s-shape seem to be of the same type i.e.

- s-shape – The union of the grid cells of length s containing dots of the pattern.

Except the s-shape consists of more steps than this, in fact to properly describe the algorithm is to describe each of the steps taken within it and the same is true for the r-shape, particularly when taken with the s-shape as a precursor. While this thesis is not meant to be a detailed analysis of the types of algorithms available it is interesting that there should be definable characteristics which could be put into a form of classification. Further work could entail examining which type of algorithms can be used to produce which types of footprint.

Commenting on footprint analysis we should look at the work done by Galton and Duckham in [11]. The paper approaches the concept of finding an appropriate footprint for a dot pattern by first looking at what was meant by the concept of 'appropriate'. Before examining the footprint criteria they point out that visual salience is problematic in that human intuition can play a great part in the shapes we see when we look at dot patterns, they note that the notion of gestalt perception almost certainly comes into play. Before describing their criteria for analysis they make one last caveat in that the specific application must decide the relevance of the footprint. The nine general criteria they provide are, in fact, questions for which a specific algorithm should give answers in order to be compared to other algorithms to assess suitability for use in a specific application. These criteria are as follows:

1. Should every member of S fall within $H(S)$ or are outliers permitted?

2. Should any points of S be allowed to fall on the boundary of $H(S)$ or must they all lie within its interior?

3. Should $H(S)$ be topologically regular or can it contain exposed point or line elements?

4. Should $H(S)$ be connected or can it have more than one component?

5. Should $H(S)$ be polygonal or can its boundary be curved?

6. Should $H(S)$ be simple, i.e., its boundary is a Jordan curve or can it have point connections?

7. How big is the largest circular (or other specified) subregion of $H(S)$ that contains no elements of S?

8. How easily can the method used be generalised to three (or more) dimensions?

9. What is the computational complexity of the algorithm?

The authors note that the criteria can be split into four categories. The questions (1) and (2) focus on the relationship between the footprint and the dot pattern. (3)–(6) describe the nature of the footprint itself. (7) is, in some respect, an indicator of the quality of the footprint, in that reducing the amount of 'free' space is important for a visually salient (this is exapanded on by Galton in [10]). (8) and (9) are both questions about the nature of the algorithm. They use these criteria to compare three algorithms and the general class of convex hull algorithms (for which all but question (9) will have the same answers). The three algorithms compared are the Swinging Arm, Close Pairs and a Delaunay triangulation based method. The Delaunay triangulation method is extended into the χ-hull in [5] and is examined in greater detail later in this section. The Swinging Arm method extends the 'gift-wrap' algorithm for constructing convex hulls. The 'gift-wrap' method is a renaming of the Jarvis March mentioned earlier. Taking an extremal point p_0 of S and half-line l anchored to p, l is swung in a clockwise direction about p_0 till it collides with another point of S, p_1. l is swung successively from p_i to p_{i+1} till $p_{i+1} = p_0$. The Swinging Arm is identical save that instead of a half-line a line segment of length r is used. Interestingly this change allows that an anti-clockwise direction of spin can change the footprint produced. The Close Pairs method considers simply joining all point-pairs whose distance is less than or equal to r, then taking the union of all the closed polygons as the footprint. With regard to how they compare, the authors note that in most cases they are identical, save for criteria (8) and (9). The extensions into three dimensions is not particularly obvious for the Swinging Arm, the arm can easily be conceptually thought of as a 'flap', but the edge about which to rotate the flap is not pre-determined and would need to be decided on. Closest Pairs generalises relatively easily, after including any polygon formed from the joins any polyhedrons with said polygons for borders are included. The complexity of both of the algorithms is at least $O(n^2)$ with a worst case of $O(n^3)$ for Swinging Arm and an unknown worst case for Closest Pairs. This kind of systematic comparison does not appear prior to this paper and will be looked at in greater detail in Chapter [Ref added later]. . For the moment we note that being able to compare the footprint types and

their algorithms can be useful in the assessment of suitability for any specific application.

The final algorithm that we'll examine in this section is the χ-hull by Duckham et al. [5] (expanding on work done in [11]). This paper includes a discussion of the footprint's properties, and how these are directly tied to the method by which it is created. Expand the bit about properties, it can be the link to the above part about footprint examination The method itself is simple to understand; starting with the Delaunay triangulation and successively removing the longest external edge, subject to constraints of maintaining connectedness and regularity, until either some predetermined minimum length is reached, or no more edges can be removed. The authors note that there can be no uniquely 'optimal' footprint when the application context is considered to be general, however, like Chaudhuri *et al.* , examine the parameter choice and its effect. There are practical limits on the minumum length l for any triangulation, if it is too large then no lines will be removed and if it is too small too many will be removed, and consequently l can be normalised. Duckham *et al.* propose using this normalised parameter, λp, to find a starting value which should achieve what they call a *characteristic shape* for many, if not all, dot patterns. While they conclude that there is no λp that always produces a 'good' characterization, the fact that they spend time considering this is futher proof of the desirability of a non-parameterised algorithm.

As previously mentioned there is little in the way of hard analysis of the footprints, the algorithms or the patterns. Clearly such work is relevant to the field and much of what has been done has only been done recently. In 2008 Galton wrote a paper [10], searching for objective criteria for evaluating the acceptability of any proposed footprint in relation to the 'perceived' shape of a dot pattern. The paper notes that in most of the published work, "while lip-service is generally paid to the fact that there is no objective definition of such a 'perceived shape', little is said about how to verify this, or indeed, about exactly what it means". Restricting attention to footprints in the form of *polygonal hulls*, simple polygons having vertices selected from the dot pattern, all the other dots being within the interior, the paper presents evidence that while a dot pattern may have several equally acceptable perceived shapes, they all represent optimal or near-optimal compromises between the conflicting goals of simultaneously minimising both the area and the perimeter of the hull.

This work was followed by a paper by this author and Galton [6], suggesting a method for classifying the footprints. Unlike Galton [10] it does not look at their 'fitness' but approaches the subject from a desire to be able to describe algorithms by the types of footprints they can create. The paper notes that the context in which the algorithm is being used determines the type of footprint that is satisfactory. With this in mind it proposes a method of using the application specific knowledge to limit the choice of algorithms for any particular user requirement. The classification bears some similarity to the set of criteria proposed by Galton and Duckham [11] for evaluating the footprints produced by different algorithms and will be detailed in Chapter [Ref added later]. .

0.2 Dot Patterns

Examining dot patterns has generally been within the field of geospatial information. However, if we move away from real-world phenomena, we can imagine that any data that can be represented on a 2-dimensional plane (e.g., classification data, multi-objective optimisation) can be viewed as a dot-pattern. This leads to a daunting amount of possible literature to examine so the analysis given is by no mean exhaustive but should serve to give a general overview.

O'Sullivan and Unwin [17] give a good description of the treatement of dot patterns (called point patterns) from a geographic standpoint. The chapter begins with noting that point patterns frequently occur in GIS (Geographic Information System[5]) and gives the examples of crime or death hot-spot analysis.

Within GIS events have a set of criteria that must be satisfied for them to be considered point patterns:

1. The pattern should be *mapped on the plane.*
2. The study area should be *determined objectively.*
3. The pattern should be an enumeration or *census* of the entities of interest, not a sample.
4. There should be *one-to-one correspondence* between objects in the study area and events in the pattern.
5. Event locations must be *proper.* They should not be, for example the centroids of areal units chosen as representative ... They really should represent the point locations of entities that can be sensibly be considered points at the scale of the study.

While we do not need to be so strict when considering dot patterns (they may, for example, be graph points for a classification), it is important to keep in mind these restrictions when assessing any GIS specific literature on the subject.

It is impossible to discuss the possibilities for change present in a dot pattern without first describing the pattern's properties in an analytical manner. Traditionally GIS has focused two connected approaches; point density and point seperation. While the two sound like synonyms for the same measurement the difference is important, if subtle. Density measures can be used to show first-order effects[6] while separation is indicative of second-order effects[7]. This work is less concerned with the implications of the properties, instead focusing on what they can tell us about the change the pattern has undergone. However these two approaches provide us with a useful starting point, having a reasoned basis on which to classify the change identifiers is a requisite aim.

It is worth noting that there are a variety of different ways dot pattern data can be stored. The structure containing the pattern can greatly effect the

[5]While a GIS is a specific system for storing geographic information, the field as a whole is often reffered to as GIS

[6]A first-order effect occurs when the physical location has correlation with the event, for example a study of the locations of the swans in hyde park it is likely that the clustering would occur around the bodies of water.

[7]A second-order effect occurs where an event affects the incidence of other events, for example a study of locations of a particular contagious disease would show clustering around as the probability of catching the disease increases with the number of events in the area

complexity of change measures (i.e. can the extremal points be found in $O(n)$). Worboys and Duckham [19] provide a useful overlay of some of the common structures and their properties.

Grid based structures are a simple starting point. The underlying concept is the *bucket*, a contiguous memory location, that will contain only points that the grid deems to be related. The basic grid type is the fixed grid structure, in which the grid partitions the region containing the pattern into equal sized cells and each cell constitutes its own bucket. All points within a specific cell are held in the same place. The obvious problem with this is when the dot pattern is not uniformly distributed, some cells may be empty while others may be near overflowing. An extension to the fixed grid is the grid file, in which the horizontal and vertical lines making up the cell divisions do not have to be equally spaced. They are placed based on the dot distribution and cells can be divided or amalgamated depending on the amount of free space they contain. The major benefit of the grid file is the ability for it to be easily dynamically updated, however using it to search for specific dots (extremal, median, etc) is not particularly fast.

Data structures do not have to mimic the spatial relationships of the dots, the data can be stored in any way that preserves those relationships. Tree structures are quick to both build and search. They are, however, not often designed with fast updates in mind, some changes involving re-computation of whole branches.

For the purposes of this thesis the data structure was important as it would greatly affect the speed of the change identifiers, however as the concept is supposed to be applicable regardless of application no assumptions could be made about the data arriving. As a result whichever data structure we used would have to be built from scratch, we would have no way of knowing if any dots were stationary from one timestep to the next or even which dots represented which real-world object (there would be no identity associated with any dot). Some form of tree structure seemed sensible as it would be quick to build and easy to search, however without a list of all the possible change identifier's requirements it is impossible to know which specific structure would be best. As such this will be revisited in a later chapter.

Things to reference

- Geographic Information Analysis book, check for further references
- Density measures
- Probability distributions
- Existing work on things like variance and mean
- see if there is any literature on describing dot patterns within classification or optimisation

0.3 Change

The focus of this thesis is not an examination of footprints or the underlying dot patterns but how the change of these things can be suitably measured. It seems

prudent to spend some time examining the existing approaches to dynamic updates.

First the Kinetic Data Structures (KDS), proposed by Basch et al. [2]. This is particularly appropriate because one of the applications they link it to is that of convex hulls under movement. The KDS is not a single algorithm, rather it is a system to describe how to create dynamic algorithms. A set of conditions, called certificates are geometric relations that possibly describe the shape we can therefore ascertain whether or not the convex hull needs to be redrawn based purely on whether or not these certificates have failed. Obviously a KDS is only useful if the cost involved in discovering and processing certificate failure is small. They state that the cost is small if it asymptotically of the order of $O(\text{Polylog}(n))$, or $O(n^{\epsilon})$, for some small $\epsilon > 0$. A KDS with such small costs is deemed *responsive*. Futher to this a KDS is *efficient* if there are very few internal events compared to external events[8], *compact* if it has a near linear number of certificates and *local* if no object participates in too many certificates.

The KDS uses short term motion plans for the objects, these are used to sort the events into queues such that the mosy likely certificate failures (events) are looked at first. An example on convex hulls is given where it is shown that by checking a set of certificates all using the rule $\text{ccw}(a, b, c)$ (where a, b and c are points and the relation is true if they form a counter-clockwise triangle). Further to this they provide a more robust method using the dualities of the convex hull and focusing only on the upper envelope. All of this gives an excellent base from which to work but it may not be a directly transferrable approach for footprints. This is because, unlike convex hulls, footprints are vaguely defined. As a result choosing the certificates is no longer a trivial task. Although the classification system examined in Chapter [Ref added later]. may well be able to play some part in defining if the shape is still valid, possibly replacing the certificates.

Hershberger and Suri [12] have a very different idea using adaptive sampling, by using an approximation of the extrema from the dot set they find a convex hull that approximates the 'true' convex hull, with triangles of uncertainty over each line segment. There is a little confusion over these areas of uncertainty in that they're supposedly constructed from the supporting lines of the vertices, however which supporting line (considering there are technically infinite) isn't given. Although from the given diagram it appears as though they mean the supporting line perpindicular to the direction in which the point was found. While sampling is not a new concept Hershberger and Suri [12] suggest that uniform sampling produces poor quality approximations in low curvature regions. As such they propose an adaptive sampling scheme.

The method works thusly, first they uniformly sample extrema in directions $2\pi/r$ for $j = 0, ..., r-1$ then they add up to r more extrema using their adaptive technique. Given an edge e let $\Theta(e)$ be the minimum angle between the directions the endpoints were sampled in. Using this they work out the proportion of the perimeter that is made up by e. If e is a large perimeter contribution then it is refined; the extreme point is found in the direction that bisects the angular range defined by e's endpoints. If the point found is not an endpoint of e then e is replaced by the two new edges of the vertices of e and the newly found point. This greatly reduces the error in the approximation.

[8] *External Event*: Changes the shape of the shape. *Internal Event*: Shape stays the same, certificates change.

This concept is far more easily applicable to the footprint problem although it's interesting to note just how vastly different it is in approach to the KDS idea. It does still have an issue in dealing with the fact that there is no unique valid footprint, making the area of uncertainty much harder to assess.

Chiang and Tamassia [4] present a general review of the field. While it is a little dated it serves as a good presentation of methods still in use. The first part of the paper looks at a different area to the other papers currently looked at. Instead of the updating of the structure or the ways in which it can be approximated they look at the data structure that represents the geometry.

They present a few storage methods; various forms of binary trees and *fractional cascading*. Either of which may be useful to the field of footprints if a suitable way of arranging the data can be created. Next they approach some general dynamization methods. Going through all the methods would be essentially repeating their words but there are a couple of terms worth noting:

- **Local rebuilding / Balancing** Technique applied to search trees that they maintain logarithmic height.
- **Partial rebuilding** This rebuilds entire subtrees when they become out of balance.
- **Global Rebuilding** Periodically reconstructs an entire tree, often used with 'weak' updates (like lazy deletion).
- **Lazy Deletion** Does not remove deleted item but marks it as deleted to be dealt with during the reconstruction.
- **Decomposable** A search problem is decomposable 'if for any partition (S', S'') of S the answer to a query on S can be obtained in constant time from the answers to queries in S' and S''.

Jumping forward to Section 7 convex hulls make an appearance. Chiang and Tamassia give a list of things we can reasonably expect from any dynamic algorithm for convex hulls:

- find if a given point of S is on the convex hull H of S;
- find if a query point is internal or external to the convex hull H of S;
- find the tangents to the convex hull H of S from an external query point;
- find the intersection of the convex hull H of S with a given query line;
- report the points on the convex hull H of S.

However they do point out that the set of points S is updated only by insertions and deletions so any point movement should be treated as being removed then added as a new point. This may be a perfectly valid assumption (certainly for the type of pattern we consider in this thesis) and it does make life simpler in terms of algorithm creation but whether or not a better algorithm could be created with knowledge of point identity may be worth considering.

Chiang and Tamassia describe a method by Preparata with update and query time of $O(\log v)$ and a report-query time of $O(v)$, where v is the number of vertices currently in the convex hull H of S. The method only deals with

insertions and is therefore not entirely applicable but it does introduce the concept of splitting the footprint into an upper and lower hull, the methods used for the upper are transferable to the lower. This concept seems common, appearing in the next algorithm and in the KDS example.

This leads into the discussion of an algorithm by Overmars and Leeuwen [18] which deals with fully dynamic hulls (i.e. insertion and deletion). Again this considers splitting the hull into two sets, one for left and one for right. This is interesting and undeniably something that merits further observation, however it lacks an concept of movement, dealing again with insertions and deletions.

This common approach of insertion and deletion is not a method we can use. As mentioned earlier we make no assumptions about our incoming data and have no identity associated with any dot. As a result any movement would have to be treated as a many insertions and deletions possibly over the whole set causing a complete rebuild at each time step, this is obviously the same as simply redrawing the shape each time.

Bibliography

[1] H. Alani, C. B. Jones, and D. Tudhope. Voronoi-based region approximation for geographical information retrieval with gazetteers. *International Journal of Geographical Information Science*, 15(4):287–306, 2001.

[2] Julien Basch, Leonidas J. Guibas, and John Hershberger. Data structures for mobile data. To Appear in Journal of Algorithms, 1997.

[3] A. Ray Chaudhuri, B. B. Chaudhuri, and S. K. Parui. A novel approach to computation of the shape of a dot pattern and extraction of its perceptual border. In *Computer Vision and Image Understanding*, volume 68, pages 257–275. Academic Press, 1997.

[4] Yi-Jen Chiang and Roberto Tamassia. Dynamic algorithms in computational geometry. In *Proceedings of the IEEE*, number 9, pages 1412–1434, 1992.

[5] Matt Duckham, Lars Kulik, Mike Worboys, and Antony Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. In *Pattern Recognition*, volume 41, pages 3224–3236. Elsevier, 2008.

[6] Max Dupenois and Antony Galton. Assigning footprints to dot sets: An analytical survey. In K. S. Hornsby, C. Claramunt, M. Denis, and G. Ligozat, editors, *Spatial Information Theory: Proceedings of the 9th International Conference COSIT 2009*, pages 227–244, Berlin, 2009. Springer.

[7] H. Edelsbrunner. Weighted alpha shapes. Technical Report UIUCDCS-R-92-1760, Department of Computer Science, University of Illinois, 1992.

[8] H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. In *ACM Transactions on Graphics*, volume 13, pages 43–72. 1994.

[9] Herbert Edelsbrunner, David G. Kirkpatrick, and Raimund Seidel. On the shape of a set of points in the plane. In *Computer Vision and Image Understanding*, volume IT-29, pages 551–559. IEEE, 1983.

[10] Antony Galton. Pareto-optimality of cognitively preferred polygonal hulls for dot patterns. In *Spatial Cognition*, 2008.

[11] Antony Galton and Matt Duckham. What is the region occupied by a set of points? In *GIScience*, 2006.

[12] John Hershberger and Subhash Suri. Convex hulls and related problems in data streams. In *Proceedings of ACM/DIMACS Workshop on Management and Processing of Data Streams*, pages 148–168, 2003.

[13] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. In *Information Processing Letters*, volume 2, pages 18–21. North-Holland Publishing Company, 1973.

[14] Mahmoud Melkemi. $\mathcal{A}$-shapes of a finite point set. In *Proceedings of the thirteenth annual symposium on Computational geometry*, SCG '97, pages 367–369. ACM, 1997.

[15] Mahmoud Melkemi and Mourad Djebali. Computing the shape of a planar points set. *Pattern Recognition*, 33(9):1423 – 1436, 2000.

[16] Mahmoud Melkemi and Mourad Djebali. Weighted $\mathscr{A}$-shape: a descriptor of the shape of a point set. *Pattern Recognition*, 34(6):1159 – 1170, 2001.

[17] David O'Sullivan and David J. Unwin. *Geographic Information Analysis*, chapter 4, pages 77–114. Wiley, November 2002. ISBN 0471211761.

[18] Mark H. Overmars and Jan van Leeuwen. Maintenance of configurations in the plane. *Journal of Computer and System Science*, 23(2):166–204, 1981.

[19] Michael Worboys and Matt Duckham. *GIS: A Computing Perspective*, chapter 6.4 Point Object Structures, pages 240 – 248. CRC Press, 2nd edition, 2004.