# Voronoi Diagrams – Computational Geometry's Favorite

Oswin Aichholzer and Franz Aurenhammer

*Institute for Theoretical Computer Science*
*Graz University of Technology*
*Graz, Austria*
*e-mail:* {*oaich,auren*}*@igi.tu-graz.ac.at*

## Introduction

*Computational Geometry* is the name of a young and dynamic branch of computer science. It is dedicated to the algorithmic study of elementary geometric questions, arising in numerous practically oriented areas like computer graphics, computer-aided design, pattern recognition, robotics, and operations research, to name a few. Computational geometry has attracted enormous research interest in the past two decades and is an established area nowadays. It is also one of the main research areas at our institute. The computational geometry group at IGI is well recognized in the international competition in that field.

*"Imagine a large modern-style city which is equipped with a public transportation network like a subway or a bus system. Time is money and people intend to follow the quickest route from their homes to their desired destinations, using the network whenever appropriate. For some people several facilities of the same kind are equally attractive (think of post offices or hospitals), and their wish is to find out which facility is reachable first. There is also commercial interest (from real estate agents, or from a tourist office) to make visible the area which can be reached in, say one hour, from a given location in the city (the apartment for sale, or the recommended hotel). Neuralgic places lying within this 1-hour zone, like the main square, train stations, shopping centers, or tourist attraction sites should be displayed to the customer."*
From: Quickest Paths, Straight Skeletons, and the City Voronoi Diagram [2].

The complexity (and appeal) hidden in this motivating every-day situation becomes apparent when noticing that quickest routes are inherently complex: once having accessed the transportation network, it may be too slow to simply follow it to an exit point close to the desired destination; taking intermediate shortcuts by foot walking may be advantageous at several places. Still, when interpreting travel duration as a kind of distance, a metric is obtained. *Distance problems*, as problems of this kind are called, constitute an important class in computational geometry.

In this context, there is one geometric structure which is maybe most famous in computational geometry: *Voronoi diagrams.* Intuitively speaking, a Voronoi diagram divides the available space among a number of given locations (called sites), according to the nearest-neighbor rule: each site $p$ gets assigned the region (of the plane, say) which is closest to $p$. A honeycomb-like structure is obtained; see Figure 1.

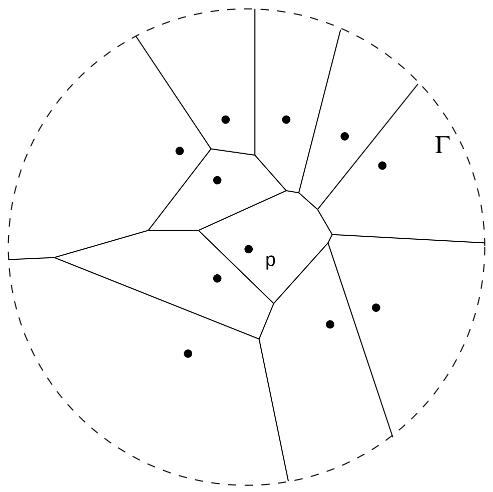
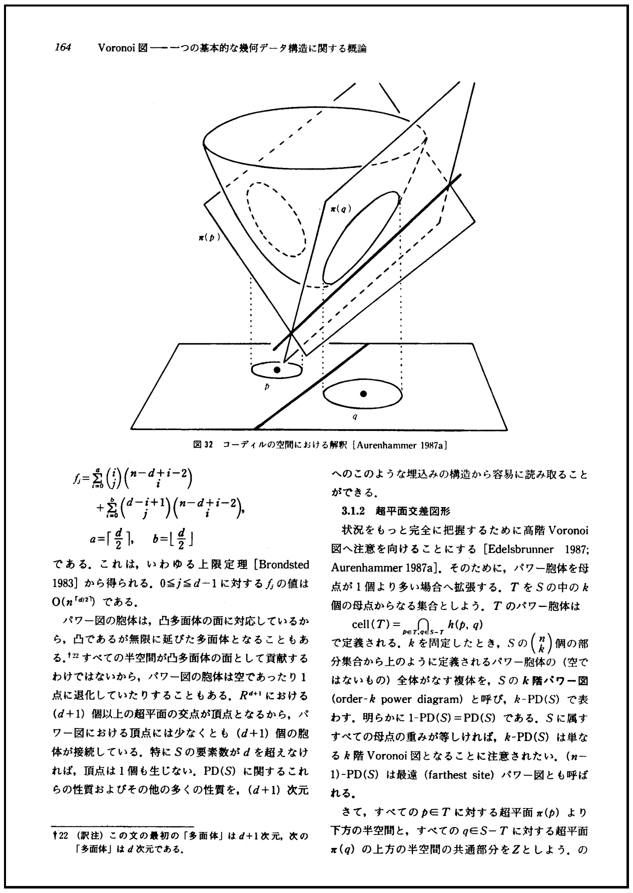


Figure 1: Voronoi regions – aesthetically pleasing ...

Christened after the Russian mathematician George Voronoi – believed to be the first to formally introduce it – this diagram has been reinvented and used in the past century in various different sciences. Area-specific names like Wigner-Seitz zones (chemics, physics), domains of action (cristallography), Thiessen polygons (geography), and Blum's transform (biology) document this remarkable fact. As of now, a good percentage of the computational geometry literature (about one out of 16 publications) is concerned with Voronoi diagrams. The computational geometry research group at Graz has been involved in this topic even before it became popular in the early 1980s (and long before our present institute has been founded). In this sense, research on Voronoi diagrams is a tradition at our place. For example, some 30 publications by the second author, including two survey articles [4] [6] (the former also available in Japanese translation [5]) have emerged from this preference.



164　Voronoi 図——一つの基本的な幾何データ構造に関する概論

図 32　コーディルの空間における解釈［Aurenhammer 1987a］

$$f_j=\sum_{i=0}^{a}\binom{i}{j}\binom{n-d+i-2}{i}+\sum_{i=0}^{b}\binom{d-i+1}{j}\binom{n-d+i-2}{i},$$

$$a=\left\lceil \frac{d}{2}\right\rceil,\quad b=\left\lfloor \frac{d}{2}\right\rfloor$$

である．これは，いわゆる上限定理［Brondsted 1983］から得られる．$0\leqq j\leqq d-1$ に対する $f_j$ の値は $O(n^{\lceil d/2\rceil})$ である．

パワー図の胞体は，凸多面体の面に対応しているから，凸であるが無限に延びた多面体となることもある．†22 すべての半空間が凸多面体の面として貢献するわけではないから，パワー図の胞体は空であったり1点に退化していたりすることもある．$R^{d+1}$ における $(d+1)$ 個以上の超平面の交点が頂点となるから，パワー図における頂点には少なくとも $(d+1)$ 個の胞体が接続している．特に $S$ の要素数が $d$ を超えなければ，頂点は1個も生じない．PD$(S)$ に関するこれらの性質およびその他の多くの性質を，$(d+1)$ 次元へのこのような埋込みの構造から容易に読み取ることができる．

**3.1.2　超平面交差図形**

状況をもっと完全に把握するために高階 Voronoi 図へ注意を向けることにする［Edelsbrunner 1987; Aurenhammer 1987a］．そのために，パワー胞体を母点が1個より多い場合へ拡張する．$T$ を $S$ の中の $k$ 個の母点からなる集合としよう．$T$ のパワー胞体は

$$\mathrm{cell}(T)=\bigcap_{p\in T,\, q\in S-T} h(p,q)$$

で定義される．$k$ を固定したとき，$S$ の $\binom{n}{k}$ 個の部分集合から上のように定義されるパワー胞体の（空ではないもの）全体がなす複体を，$S$ の **$k$ 階パワー図** (order-$k$ power diagram) と呼び，$k$-PD$(S)$ で表わす．明らかに 1-PD$(S)$＝PD$(S)$ である．$S$ に属すすべての母点の重みが等しければ，$k$-PD$(S)$ は単なる $k$ 階 Voronoi 図となることに注意されたい．$(n-1)$-PD$(S)$ は最遠 (farthest site) パワー図とも呼ばれる．

さて，すべての $p\in T$ に対する超平面 $\pi(p)$ より下方の半空間と，すべての $q\in S-T$ に対する超平面 $\pi(q)$ の上方の半空間の共通部分を $Z$ としよう．の

†22（訳注）この文の最初の「多面体」は $d+1$ 次元，次の「多面体」は $d$ 次元である．

Figure 2: 3D view of sites and distances

We devote the present article to this fascinating geometric structure, with the intention to highlight its manifold rôle in computer science. Voronoi diagrams have proved to be a powerful tool in solving seemingly unrelated computational questions, and efficient and reasonably simple techniques have been developed for their computer construction and representation. Moreover, Voronoi diagrams have surprising mathematical properties and are related to many well-known geometric structures. Finally though, human intuition is guided by visual perception: if one sees an underlying structure, the whole situation may be understood at a higher level.

## Classical applications

Voronoi diagrams capture the distance information inherent in a given configuration of sites in a compact manner. Basically, the structure is built from *edges* (portions of perpendicular bisectors between sites) and *vertices* (endpoints of edges). To represent a Voronoi diagram in a computer, any standard data structure for storing geometric graphs will do. Nonetheless, several tailor-made representations have been developed, the most popular one being the *quad-edge data structure.* Simple graph-theoretical arguments show that there are at most $3n-6$ edges and $2n-4$ vertices for $n$ sites. In other words, the storage requirement is only $O(n)$, which gives one more reason for the practical applicability of Voronoi diagrams.

Continuing from our introductory example, imagine the given sites are post offices. Then, for any chosen location of a customer, the containing Voronoi region makes explicit the post office closest to him/her. More abstractly, by performing *point location* in the data structure 'Voronoi diagram', the nearest neighbor site of any query point can be retrieved quickly. (In fact, in $O(\log n)$ time. This is optimal by a matching information-theoretic bound). Similarly, sites may represent department stores, and Voronoi neighborhood – witnessed by diagram edges – will indicate stores in strongest mutual influence (or competition). Apart from economics, this basic finding has far-reaching applications in biological and physico-chemical systems. On the other hand, Voronoi vertices are places where the influence from sites reaches a local minimum – a fact of interest in facility location.

Similar observations apply to the *robotics* scenery: sites are snapshots of moving robots, and danger of collision is most acute for the closest pair. Moreover, when planning a collision-free motion of a robot among a given set of obstacle sites, sticking to the boundaries of Voronoi regions (that is, moving along Voronoi edges and vertices) will keep the robot off the obstacles in a best possible way. (This is known as the retraction approach in motion planning.)

Numerous other applications of Voronoi diagrams exist, one notable being geometric *clustering.* The grouping of data sites into a cluster structure is reflected by their Voronoi regions. (For instance, dense clusters give rise to regions of small area.) Even more important is the fact that prominent types of *optimal* clusterings are induced by Voronoi diagrams, namely by partition with regions (which are not necessarily defined by the data sites to be clustered).

Two important and beautiful geometric structures cannot be hidden at this point. Firstly, any *shortest connection network* for the sites (think of a road or a electricity network) will solely connect sites which are Voronoi neighbors. Secondly, the graph connecting all the neighbored sites is a triangular network, called the *Delaunay triangulation*. Among all possible ways to build a triangular irregular network (TIN) the Delaunay triangulation is provably optimum, in several respects concerning the shape and size of its triangles (or tetrahedra, when it comes to higher dimensions). It is for this reason that Delaunay triangulations have been extensively used in surface generation, solid modeling, and related areas.

Beside shortest connection networks (or minimum spanning trees, as they are called in computational geometry) there are several other classes of geometric *neighborhood graphs* which are contained in the Delaunay triangulation: $\alpha$-shapes (a tool in surface modeling), $\beta$-skeletons (with applications to the famous and still unsettled minimum-weight-triangulation problem), Gabriel graphs (geographic information systems (GIS)), and nearest-neighborhood graphs (pattern recognition).

## Algorithms designer's playground

Methods for constructing Voronoi diagrams are as old as their use in the diverse areas of natural sciences. Of course, the first diagrams have been drawn with pencil and ruler. At these early times, people already complained about ambiguities if the sites come in a co-circular fashion. Nowadays, where sophisticated, efficient, and practical construction algorithms exist, robustness in the case of degenerate input sites is still an issue, and much of the program designers work goes into the implementation of 'special cases'. The heart of an algorithm, however, is the underlying paradigmatic technique, and rarely a problem has been better a playground for algorithms design than the computation of a Voronoi diagram.

Beside other intuitive construction rules, *incremental insertion* has been among the first algorithmic techniques applied to Voronoi diagrams. This technique is well known from InsertionSort, a simple sorting method that maintains a sorted list during the insertion of items. In our case, insertion of a site means integrating its Voronoi region into the diagram constructed so far – a process that involves the construction of new and the deletion of old parts. Though the approach stands out by its simplicity and obvious correctness, the resulting runtime may be bad: finding a place to start the insertion is tricky, and many already constructed parts may have to be deleted lateron. It required the advent of *randomization* to give this approach an efficiency guarantee. To be more specific, inserting $n$ sites in random order leads to an (expected) runtime of $O(n \log n)$ which is provably optimal. And only (almost-)optimal algorithms come up to a big advantage of the data structure Voronoi diagram: the linear storage requirement, $O(n)$.

*"The intrinsic potential of Voronoi diagrams lies in their structural properties, in the existence of efficient algorithms for their construction, and in their adaptability."*
From: Handbook of Computational Geometry [5], Chapter V.

Though the ancient Romans definitely knew about the power of "divide et impera", its algorithmic analog *divide & conquer* is often considered a less intuitive technique. It achieves efficiency by splitting the problem at hands, then solving the subproblems separately (and recursively), and finally merging the solutions. Voronoi diagrams are well suited to this attack. After presorting the sites (in $x$-direction, say) the merging of two subdiagrams can be done in $O(n)$ time, which calculates to a total runtime of $O(n \log n)$ by the recurrence relation $T(n) = 2 \cdot T(\frac{n}{2}) + O(n)$. Divide & conquer provided the first optimal algorithm for Voronoi diagrams, but certain peculiarities are buried in its implementation. On the other hand, it is a candidate for efficient parallelization.

Two more optimal construction techniques are known, both being specific to geometry. The *plane-sweep technique* sweeps the plane containing the $n$ input sites with a vertical line $L$, from left to right, say. Thereby, it maintains the invariant that all parts of the object to be constructed, which lie to the left of $L$, have already been completed. In this way, a 2D static problem (the construction of a Voronoi diagram) is translated into a 1D dynamic problem (the handling of the interactions near the sweep line). When utilizing the advanced data structures 'priority queue' and 'dictionary', this event-driven algorithm runs in $O(\log n)$ time per event, the number of which is proportional to the size of a Voronoi diagram, $O(n)$.

Finally, *geometric transformation* is an elegant tool to gain algorithmic efficiency, via mapping a given problem to a better understood (and preferably solved) one. Its application to Voronoi diagrams is described in the next paragraph.

## Back to geometry

Figure 2 gives a flavor of how proximity in 2D may be expressed by convexity in 3D. The surprising observation that a 2D Voronoi diagram is nothing but a projected convex 3D polyhedron opens our eyes – and a door to new construction methods. Polytope theory tells us to look for the geometric dual of that polyhedron (which now is the *convex hull* of $n$ points in 3D), and indeed there is a simple rule to obtain these hull points directly from the given Voronoi sites: project them onto the paraboloid of rotation. The careful reader may notice that this very convex hull projects back to the afore-mentioned Delaunay triangulation of the sites in the plane. Convex hull algorithms are well established in computational geometry, and practical and robust implementations (running in time $O(n \log n)$ in 3D) are available.

Noteworthy more is hidden in the geometric relation mentioned above. Firstly, the theory of convex polytopes allows us to exactly analyze the number of individual components of a Voronoi diagram. This is a highly non-trivial task in three and higher dimensions; faces of various dimensions (vertices, edges, facets, etc.) have to be counted in a thorough analysis of the storage requirement. Secondly, a connection to *hyperplane arrangements* is drawn, that is of importance when Voronoi diagrams are modified to order $k$. Here subsets of $k$ sites get assigned their Voronoi regions; they carry the information for efficiently performing $k$-nearest neighbor search.

Finally, a natural generalization arises in the light of the geometric transformation shown in Figure 2. *Power diagrams*, defined by circular or spherical sites, and retaining the convexity of the regions. They constitute *exactly* those diagrams that are projected boundaries of convex polyhedra; Voronoi diagrams are special cases where circles degenerate to point sites. (When writing his doctoral thesis, the second author was excited when discovering the beauty and versatility of this structure; its name has been coined after one of his papers [3].) Power diagrams in 3D are related to important types of Voronoi diagrams in 2D, and thus provide a unified view of these structures. Among them are diagrams for sites with individual weights, expressing their capability to influence the neighborhood. These flexible models are used in other areas of science (Johnson-Mehl model in chemics and Appolonius model in economics). Even the mathematical physicist Clerk Maxwell in 1864 (implicitly) payed attention to power diagrams: he observed that a diagram reflects the equilibrium state of a spider web just if the diagram comes from projecting a polyhedron's boundary...

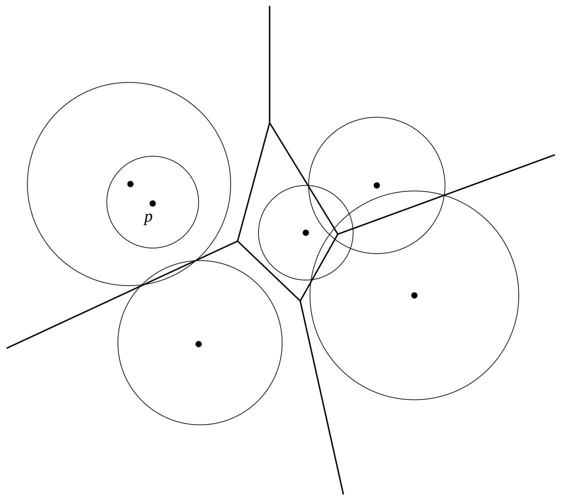


Figure 3: Power diagram for 6 circles

## A novel concept

In order to meet practical needs, Voronoi diagrams have been modified and generalized in many ways over the years. Concepts subject to change have been the shape of the sites (standard: points), the distance function used (standard: Euclidean metric), or even the underlying space (standard: the plane or 3-space). It is not appropriate here to give a systematic treatment of the various existing types of Voronoi diagrams. Instead, we would like to report on a particular type, which has been recently introduced and highlighted with success by our research group.

Let us aid the reader's intuition by giving a physical interpretation of Voronoi diagrams. Imagine a handful of small pebbles being thrown into a quiet pond, and watch the circular waves expanding. The places where waves interfere are equidistant from the pebbles' hitting points. That is, a Voronoi diagram is produced. (This is the so-called growth model in biology and chemics.)

*Straight skeletons* [1] are diagrams induced by wavefronts of more general shape. Consider a set, $F$, of simple polygons (called figures) in the plane. Each figure in $F$ is associated with a birth time, and an individual speed for each of its edges to move in a self-parallel fashion. In this way, each figure sends out a polygonal wavefront (actually two, an external and an internal one). The straight skeleton of $F$ now is the interference pattern of all these wavefronts, under the requirement that their expansion ceases at all points where wavefronts come into contact or self-contact.

During their propagation, the wavefront edges trace out planar and connected Voronoi diagram-like regions. Wavefront vertices move at (speed-weighted) angle bisectors for edges, and thus trace out straight line segments. We originally intended straight skeletons as a linearization of the *medial axis*, a widely used internal structure for polygons. (The medial axis is just the Voronoi diagram for the components of a polygon's boundary. It contains parabolically curved edges if the polygon is non-convex.) In numerous applications, e.g., in pattern recognition, robotics, and GIS, skeletonal partitions of polygonal objects are sought that reflect shape in an appropriate manner. The straight skeleton naturally suits these needs; see Figure 4. It is superior to the medial axis also because of its smaller size.

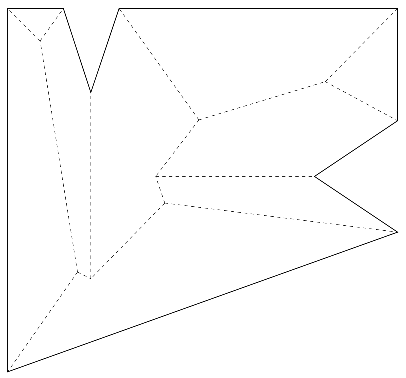


Figure 4: Internal straight skeleton

Curiously enough, straight skeletons do *not* admit a distance-from-site definition, in general (and therefore are no Voronoi diagrams in the strict sense). This counter-intuitive finding outrules the well-developed machinery for constructing Voronoi diagrams; merely a simulation of the wavefront expansion will work. The theoretically most efficient implementation runs in roughly $O(n\sqrt{n})$ time, and a triangulation-based method that maintains 'free-space' exhibits an $O(n \log n)$ observed behavior for many inputs.

Straight skeletons apply to seemingly unrelated situations. This partially stems from a nice 3D interpretation, which visualizes the movement of each wavefront edge as a facet in 3D. The expansion speed of the edge determines the slope of the facet. In this way, each figure gives rise to a polyhedral cone in 3D, whose intersection with the plane is just the figure itself. The surface made up from these cones projects vertically to the straight skeleton. See Figure 5 for an illustration.

A problem from architectural design is constructing a roof that rises above a given outline of a building's

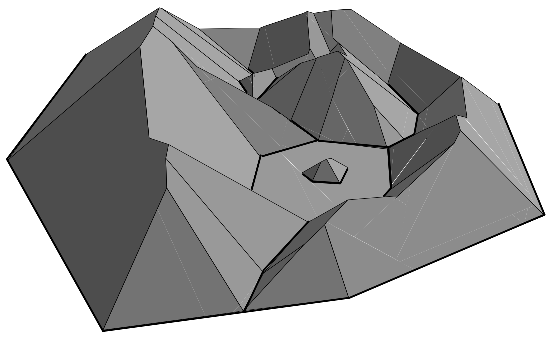


Figure 5: Terrain reconstructed from river map

groundwalls. This task is by no means trivial as roofs are highly ambigous objects. A more general question is the reconstruction of geographical terrains (say, from a given river map with additional information about elevation and slope of the terrain), which is a challenging problem in GIS. The straight skeleton offers a promising approach to both questions. Of particular elegance is the following property: the obtained 3D surfaces are characterized by the fact that every raindrop that hits the surface facet $f$ runs off to the figure edge that defines $f$. This applies to the study of rain water fall and the prediction of floodings.

Finally, there is an application to a classical question in origami design that deserves mention: is every simple polygon the silhouette of a flat origami? An affirmative answer has been found recently. The used method is based on covering certain polygonal rings that arise from shrinking the polygon in a straight skeleton manner. That is, our concept of straight skeleton allows for a relatively simple proof of this longstanding open conjecture in origami theory.

## ... and the city Voronoi diagram

Whereas the standard Voronoi diagram has interpretations in both the wavefront model and the distance-from-site model, this is not true for other types. Remember that straight skeletons cannot be defined in the latter model. Conversely, the occurance of disconnected Voronoi regions (as in the Apollonius model, where distances are weighted by multiplicative constants) disallows an interpretation in the former model. We conclude this article with a non-standard and demanding structure, that we have investigated recently, and that bridges the gap between both models.

The structure in question – we called it the *city Voronoi diagram* [2] – is just the diagram matching the motivating example in the introduction. Recall that we are given a city transportation network that influences proximity in the plane. We model this network as a

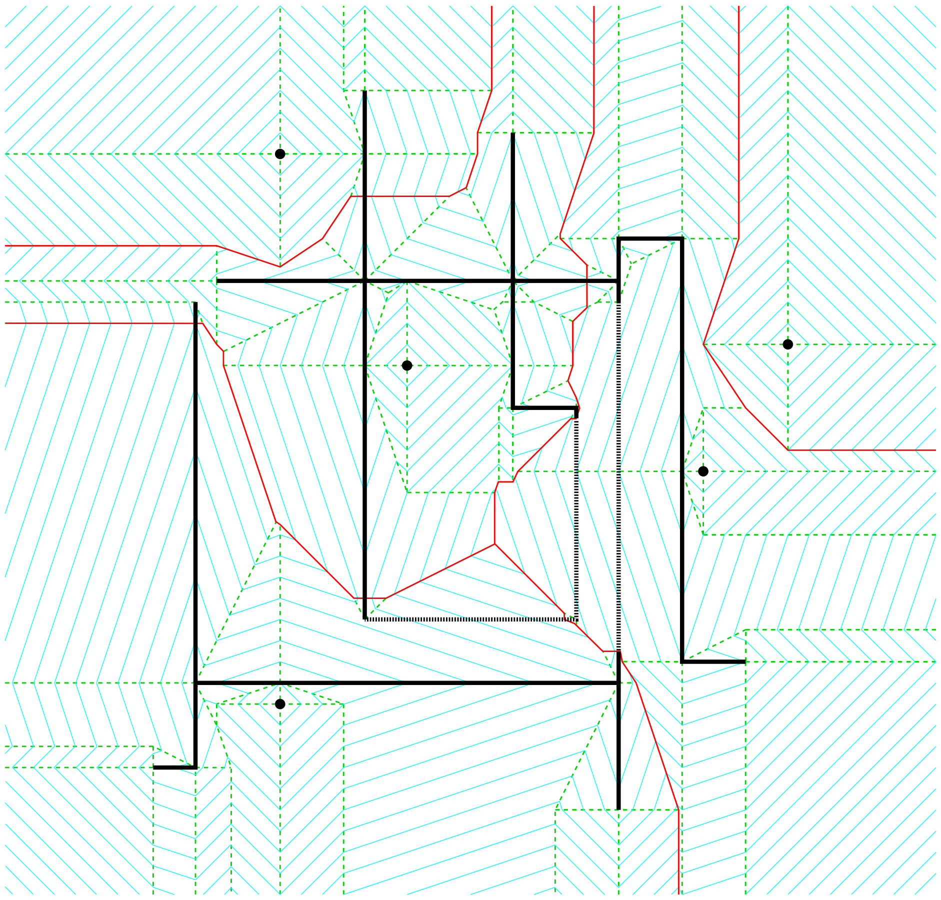


Figure 6: A complex transportation network $C$ (bold segments) and the city Voronoi diagram (full edges) for a set $S$ of five sites scattered among $C$. The diagram is refined by straight skeleton edges (dashed). Quiet ranges occur in the network, shown as dotted subsegments. They could be closed down without delaying any quickest path to $S$. Also, one site is isolated from the network (the rightmost site $s$), in the sense that from nowhere in its region the quickest path to $s$ takes advantage of $C$.

planar straight-line graph $C$ with horizontal or vertical edges. No other requirements are posed on $C$ – it may contain cycles and even may be disconnected.

By assumption, we are free to enter $C$ at any point. (This is not unrealistic for a bus system with densely arranged stops, and exactly meets the situation for shared taxis which regularly drive on predetermined routes and will stop for every customer.) Once having accessed $C$ we travel at arbitrary but fixed speed $v > 1$ in one of the (at most four) available directions. Movement off the network takes place with unit speed, and with respect to the $L_1$ (Manhattan) metric. (Again, this is realistic when walking in a modern city).

Let now $d_C(x, y)$ be the duration for quickest route (which of course may use the network) between two given points $x$ and $y$. When viewed as a distance function, this 'city metric' $d_C$ induces a Voronoi diagram as follows. Each site $s$ in a given point set $S$ gets assigned the region

$$reg(s) = \{x \mid d_C(x, s) < d_C(x, t), \forall t \in S \setminus \{s\}\}.$$

Setting equality in this term gives the *bisector* of two sites $s$ and $t$. This is the locus of all points which can be reached from $s$ and $t$ within the same (minimum) time. Bisectors are polygonal lines which, however, show undesirable properties in view of an algorithmic construction of the city Voronoi diagram. By $C$'s influence, they are of non-constant size, and even worse, they may be cyclic. These are main obstacles for efficiently applying divide & conquer and randomized incremental insertion.

The key for a proper geometric and algorithmic understanding of the city Voronoi diagram lies in the concept of straight skeletons. Let us ignore the network

$C$ for a moment. The $L_1$-metric Voronoi diagram for the sites in $S$ already is a straight skeleton. Its figures are the $L_1$ unit circles (diamonds) centered at the sites. How does the network $C$ influence their wavefronts? Their shapes change in a pre-determined manner, namely whenever a wavefront vertex runs into a network segment, or a wavefront edge slides into a network node. A new diamond will appear at such a place, along with tangent sharp-angled wedges whose peaks move at speed $v$ in all possible directions on $C$. All these diamonds and wedges are taken as new figures. Together with the original $L_1$ diamonds, their straight skeleton now gives the city Voronoi diagram.

In fact, the obtained skeleton contains a lot more edges (and information). Given a query point $q$, not only the first site $s$ in $S$ reachable from $q$ can be retrieved, but rather the quickest route from $q$ to $s$ itself – a polygonal path of possibly high complexity. Still, this refined city Voronoi diagram has a size of only $O(n + c)$, where $n$ and $c$ denote the complexity of $S$ and $C$, respectively. (This important property is lost for non-isothetic networks or for the Euclidean metric; the size blows up to quadratic.)

Two major difficulties have to be mastered before arriving at an efficient construction algorithm. Firstly the set of figures, when constructed from $S$ and $C$ as sketched above, contains high redundancy for various reasons. Secondly, when having available the set of $O(n + c)$ non-redundant figures, a fast way of computing their straight skeleton has to be found. The second goal is achieved by modifying the figures so as to fit into the framework of so-called *abstract Voronoi diagrams.*

This general and elegant framework extracts the desired algorithmic properties of a Voronoi diagram. It is based on an admissible system of bisectors, rather than on some distance from the sites. For example, for each triple of abstract sites, any point common to two bisectors must also belong to the third. (This property is trivially fulfilled for any distance-defined diagram, but is violated by straight skeletons, in general. 'No-mans lands' belonging to no site are the consequence.) In our case, a careful adaption of the figures allows a reduction to abstract Voronoi diagrams. The refined city Voronoi diagram then can be constructed in $O(n \log n + c^2 \log c)$ time and optimal space.

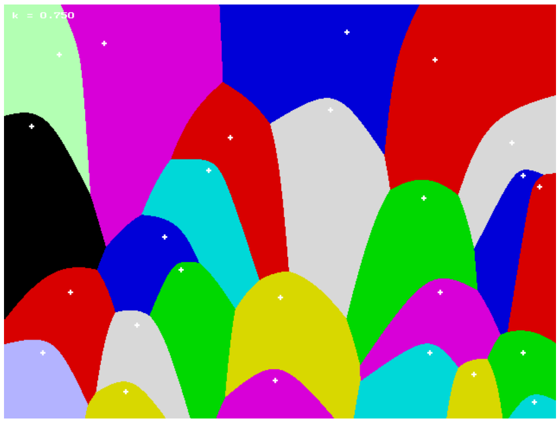


Figure 7: Direction-sensitive diagram [7]. Quo vadis?

## References

This choice of references is highly selective. We refrained from listing all our publications relevant to Voronoi diagrams; this is done elsewhere in the present issue.

1. O.Aichholzer, F.Aurenhammer, *Straight skeletons for general polygonal figures in the plane.* In: Voronoi's Impact on Modern Sciences II, A.M.Samoilenko (ed.), Proc. Institute of Mathematics of the National Academy of Sciences of Ukraine 21, Kiev, Ukraine, 1998, 7-21.

2. O.Aichholzer, F.Aurenhammer, B.Palop, *Quickest paths, straight skeletons, and the city Voronoi diagram.* Submitted to the ACM Symp. Computational Geometry 2002.

3. F.Aurenhammer, *Power diagrams: properties, algorithms, and applications.* SIAM Journal on Computing, 16 (1987), 78-96.

4. F.Aurenhammer, *Voronoi diagrams – a survey of a fundamental geometric data structure.* ACM Computing Surveys 23 (1991), 345-405.

5. F.Aurenhammer. *Voronoi diagrams – a survey of a fundamental geometric data structure (Japanese translation).* BIT – ACM Computing Surveys '91, Kyoritsu Shuppan Co., Ltd.(1993), 131-185.

6. F.Aurenhammer, R.Klein, *Voronoi diagrams.* Handbook of Computational Geometry, Chapter 5, J.Sack, G.Urrutia (eds.), Elsevier Science Publishing, 2000, 201-290.

7. O.Aichholzer, F.Aurenhammer, D.Z.Chen, D.T.Lee, E.Papadopoulou, *Skew Voronoi diagrams.* Int'l. Journal of Computational Geometry & Applications 9 (1999), 235-247.