

Rounding Voronoi Diagram^{*}

Olivier Devillers and Pierre-Marie Gandoin

INRIA, BP93, 06902 Sophia Antipolis

Abstract. Computational geometry classically assumes real-number arithmetic which does not exist in actual computers. A solution consists in using integer coordinates for data and exact arithmetic for computations. This approach implies that if the results of an algorithm are the input of another, these results must be rounded to match this hypothesis of integer coordinates. In this paper, we treat the case of two-dimensional Voronoi diagrams and are interested in rounding the Voronoi vertices to grid points while interesting properties of the Voronoi diagram are preserved. These properties are the planarity of the embedding and the convexity of the cells. We give a condition on the grid size to ensure that rounding to the nearest grid point preserves the properties. We also present heuristics to round vertices (not to the nearest grid point) and preserve these properties.

Keywords: geometric computing, Voronoi diagram, integer coordinates, exact computations

1 Introduction

Theoretical computational geometry often assumed real-number arithmetic to model computations. In practice, this model is not implementable on an actual computer and the use of floating point computation as an approximation of real arithmetic is well known to yield difficult precision problems in the implementation of geometric algorithms. Recent trends in computational geometry consist in using integer coordinates to represent the data, and exact integer arithmetic to make the computations [4,5,12].

Exact output: A solution to solve precision problems consists in using some exact representation for the input and exact computations to take decisions inside the algorithm. For example, if point coordinates are b -bits integers, then the intersection of line segments can be solved using an exact $2b$ -bits arithmetic [3]. Unfortunately, this approach works only for a single algorithm. If two algorithms have to be cascaded and the output of one must be the input of another, then we need an exact representation of the output. In the example of intersection of line segments, rational numbers with $2b$ -bits (numerator and denominator) are needed to represent intersection points coordinates. If several algorithms have to be cascaded this approach yields an unacceptable increase in the precision used to store results.

^{*} This work was partially supported by ESPRIT LTR GALIA

Rounding: The alternative approach consists in rounding the result of an algorithm before starting the next one. Rounding means to move the exact results to some fixed sized integer representation. If the results consist in points, they must be moved to coincide with vertices of a grid. In that case, a brute force rounding may alter the properties of the result, for example a convex polygon may loose its convexity after rounding, and the second algorithm may not work on the rounded result; thus it is necessary to define rounding methods preserving the geometrical properties of objects. The rounding of geometric structures is a new concern in the domain, to the knowledge of the authors, previous works deal only with the rounding of arrangement of line segments [8,7] and arrangement of triangles in 3D [6]. These works use *snap rounding*: a point is rounded to its nearest grid point.

Voronoi: In this paper, we address the particular problem of rounding the Voronoi diagram of a set of points S in the plane (see Definition 1). An example of cascading algorithms may be found in 3D reconstruction, where we need to compute the overlap of two Voronoi diagrams [1,2]. Given a *diagram* (a graph embedded in the plane), we will say that this diagram is *planar* if its edges do not cross and *convex* if all cells are convex. It is well known that a Voronoi diagram is planar and convex and there exist algorithms to compute the overlay of two convex planar diagrams, but convexity and planarity may not be preserved if the vertices of the Voronoi diagram are snap-rounded to grid points. In Figure 1, the rounded diagram (in dashed lines) has a non convex cell (the shaded cell). In this paper we will investigate on which conditions the snap rounding of a Voronoi diagram will preserve planarity and convexity. We will also develop other ways of rounding than snap rounding.

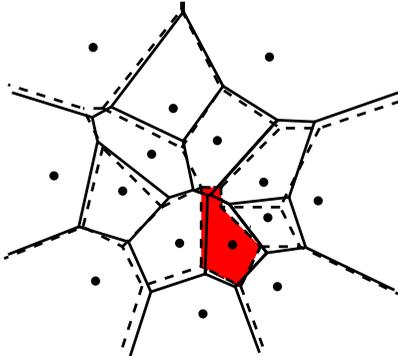


Fig. 1. Rounding of Voronoi diagram

Counter example: A good rounding does not exist in the worst case. As shown by Figure 2, it is not possible to have simultaneously in a rounded version of the diagram the convexity of cell of p and a fixed bound on the distance between Voronoi vertices and their rounded versions.

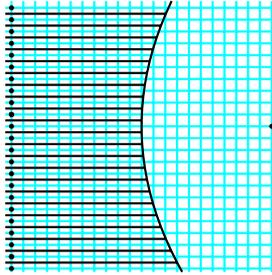


Fig. 2. In this example, a “good” rounding does not exist

Rounded computational geometry versus discrete geometry: The use of a grid in computational geometry implies some convergence with fields such as discrete geometry or computer vision. The main difference between the approaches used in these domains is a question of order of magnitude. In an image a grid point is a pixel, and the typical size of an image is 1000×1000 pixels. In computational geometry, the number of grid points depends on the integer arithmetic, a grid of $16.000.000 \times 16.000.000$ (for 24 bits integers) is a lower bound. This difference of scale has consequences on the algorithm, for example storing the entire grid is not possible with that sizes.

Voronoi diagram definition:

Definition 1 Let \mathcal{S} be a set of n points in the plane, M_1, \dots, M_n , which we call the sites to avoid confusion with the other points in the plane. To each site M_i we attach the region $V(M_i)$ that contains the points closer to M_i than to any other point in \mathcal{S} :

$$V(M_i) = \{X \in \mathbb{R}^2 \mid \delta(X, M_i) \leq \delta(X, M_j) \quad \forall j \neq i\}$$

where δ denotes the Euclidean distance in \mathbb{R}^2 .

The region $V(M_i)$ is the intersection of a finite number of closed half-planes (bounded by the perpendicular bisectors of $M_i M_j$, $j = 1, \dots, n, j \neq i$). This shows that $V(M_i)$ is a convex polygon, which may be bounded or unbounded. The $V(M_i)$'s and their edges form a cell complex called Voronoi diagram of \mathcal{S} , whose domain is the whole \mathbb{R}^2 .

The Delaunay triangulation of \mathcal{S} is a complex dual to the Voronoi diagram of \mathcal{S} , and can be obtained from it by joining the sites whose Voronoi cells are adjacent (see Figure 3).

2 Sufficient condition of correct rounding

Given a set \mathcal{S} of sites in the plane and a grid step p , the main purpose is to know if the Voronoi diagram of \mathcal{S} will keep, after snap rounding of its vertices, its properties of convexity (each angle of the diagram should stay $\leq \pi$ after rounding) and planarity (each angle of the diagram must keep its orientation after rounding). We will first focus on a convexity criterion and then adapt the

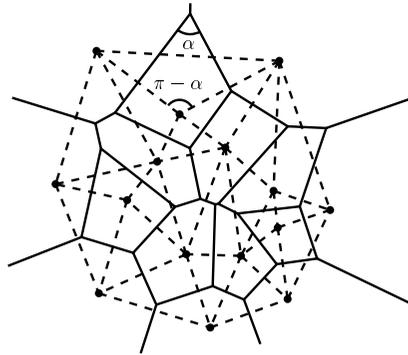


Fig. 3. Duality between Voronoi (continuous lines) and Delaunay (dashed lines)

results into a planarity criterion. If the criterion condition is not satisfied, we want to compute p_{MAX} the distance between two consecutive grid points or *grid size* which guaranty the preserving of these properties. The condition must depend only on the original data: the coordinates of the sites of \mathcal{S} , and the topology of the diagram (which is equivalent to the knowledge of the Delaunay triangulation); the criterion should not depend on an explicit computation of the Voronoi vertices coordinates.

2.1 Parameterization

Let A, B, C be 3 Voronoi vertices forming an angle α in A , and A', B', C' the snap rounded corresponding vertices. The angle $\alpha' = (\mathbf{A'B'}, \mathbf{A'C'})$ has 7 degrees of freedom and thus depends on 7 parameters: the coordinates of the points A, B, C plus the grid size p for example. In order to make some simplifications, we can use an other choice of parameters. α' can be expressed as a function of the angle α , the lengths $b = AC$ and $c = AB$, the orientation θ and the position (x_A, y_A) of the triangle (ABC) in the grid, and the grid size p (see Figure 4). But in order to know if, for a given grid size p , a given angle will be rounded correctly regarding its convexity, we will have to solve the equation $\alpha' = \pi$ in the variable p . This equation is too intricate and we will make some simplifications and over-estimations.

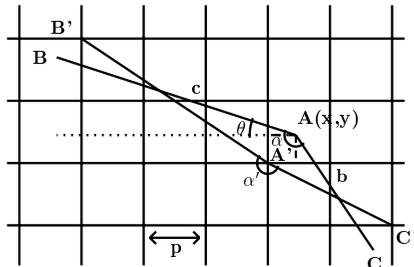


Fig. 4. α' depends on 7 parameters

2.2 Monotonicity of snap rounding: pre-criterion

Let ABC be an angle, and let divide the plane in four quadrants from the vertex A . The monotonicity of snap rounding (i.e. $x_A \leq x_B \Rightarrow \text{round}(x_A) \leq \text{round}(x_B)$ and $y_A \leq y_B \Rightarrow \text{round}(y_A) \leq \text{round}(y_B)$) involves that the extremities B and C of the angle stay in the same quadrant when they are rounded (see Figure 5). Therefore, given an angle whose extremities do not lie strictly in opposite quadrants, the corresponding rounded angle is necessarily $\leq \pi$. Such an angle is declared as *not risky* for the convexity and no further verifications are needed. The main criterion for convexity described below will be computed only for *risky configurations*, i.e. whose extremities lie strictly in opposite quadrants of plane (see Figure 5). We obtain thus a preselecting criterion that we will call pre-criterion in the sequel.

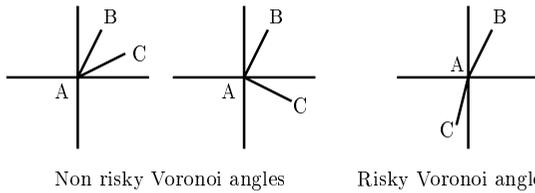


Fig. 5. Use of the monotonicity of snap rounding

2.3 Simplification

To reduce the number of variables in the expression of the rounded angle α' , we will suppress the parameters (x_A, y_A) by considering the worst case, i.e. the case that maximizes α' . A simple way to consider the worst case is to assume that the three points A, B, C lie at centerpoints of a grid squares and thus have a maximal move of half a pixel diagonal. So given 3 points A, B, C centers of 3 unit-squares the problem is to choose A', B', C' on squares vertices so as to maximize α' . The general case is not obvious, because this choice depends on the values of α and θ . However in a risky configuration, the points B and C are in opposite quadrants and the worst position of the points A', B', C' is clearly the one shown in Figure 6.

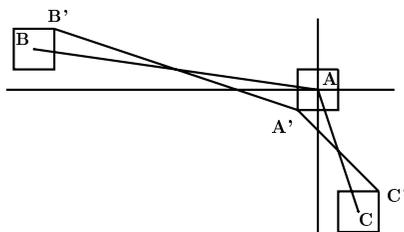


Fig. 6. Position of A', B', C' maximizing α'

2.4 First criterion: preserving the convexity of angles

The rounding of a point can be viewed as a translation by some vector \mathbf{u} . We will now use the fact that with the assumptions of the previous paragraph, the vectors of translation of points A, B, C have the same direction and the same norm. More precisely, if A is translated by \mathbf{u} , B and C are translated by $-\mathbf{u}$. But these 3 translations are equivalent — considering the value of the rounded angle α' — to translating the point A by $2\mathbf{u}$ with B and C fixed. Since $\|\mathbf{u}\| = \sqrt{2}p$, it follows that the maximum grid size p_{MAX} such that α' is convex, is given by the length $h'/\sqrt{2}$ (see Figure 7). To calculate h' , we will use the

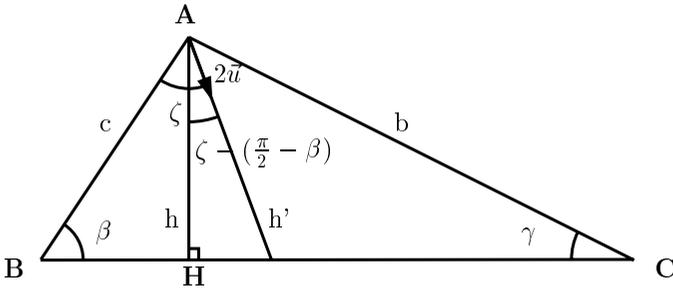


Fig. 7. Calculation of a convexity criterion

height h of triangle (ABC) and angle $\zeta = \theta + \frac{\pi}{4}$. Basic relations of triangle geometry give: $h = c \sin(\beta)$ and $h' = \frac{h}{\cos(\frac{\pi}{2}-\beta-\zeta)}$. After reduction, we obtain: $h' = \frac{c}{\cos(\zeta)+\sin(\zeta) \cot(\beta)}$. It remains to be expressed $\cot(\beta)$ as a function of b, c and α . Once again, the relations of triangle geometry give, after reductions: $\cot(\beta) = \frac{c-b \cos(\alpha)}{b \sin(\alpha)}$. So finally, the maximum grid size warranting convexity of the rounded Voronoi diagram is:

$$p_{MAX} = \frac{c/\sqrt{2}}{\cos(\zeta) + \sin(\zeta) \frac{c-b \cos(\alpha)}{b \sin(\alpha)}} \tag{1}$$

2.5 Semi-infinite Voronoi angles

We call *semi-infinite Voronoi angles* the angles containing an infinite point, i.e. defined by a segment $[AB]$ and a ray $[AC)$. The most natural way to round this kind of angles is to snap round normally the segment $[AB]$ in a segment $[A'B']$, and to transform the ray $[AC)$ in a parallel ray $[A'C')$. This case can be treated like the general case, with a few modifications. With the same notations as previously, the worst case consists here in translating the point B by $-2\mathbf{u}$. Therefore, we have to express the length h' as a function of c, α and ζ . It follows that the maximum grid size such as the rounded semi-infinite angle remains convex is:

$$p'_{MAX} = \frac{c/\sqrt{2}}{\cos(\zeta) - \sin(\zeta) \cot(\alpha)} \tag{2}$$

2.6 From Voronoi to Delaunay

We have thus obtained a criterion expressed on some lengths and angles in the Voronoi diagram. We now have to explain how these quantities can be computed from the original data, that is from the sites coordinates without an explicit computation of the Voronoi vertices. We will assume that the Delaunay triangulation, which encodes all the combinatorial information of the Voronoi diagram, is known. A Voronoi angle α can be deduced from its dual α_1 in Delaunay by $\alpha = \pi - \alpha_1$ (see Figure 3).

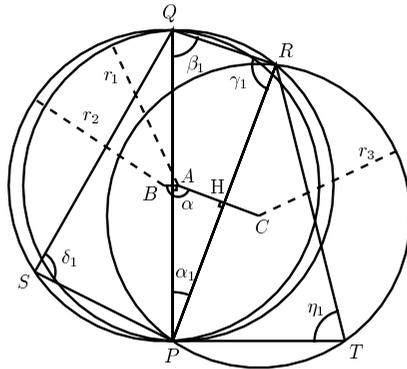


Fig. 8. How express Voronoi data from Delaunay data

The only difficulty is to express the edge lengths b and c using only Delaunay angles and Delaunay edges lengths. On Figure 8, A, B and C are the vertices of a Voronoi angle, the triangles $(PQR), (PQS)$, and (PQT) are the corresponding Delaunay faces whose radii of circumscribed circles are respectively r_1, r_2 and r_3 .

We get $b = r_1 \cos(\beta_1) + r_3 \cos(\eta_1)$ and $c = r_1 \cos(\gamma_1) + r_2 \cos(\delta_1)$. For the radii of the circumscribed circles, we have the relation (with the usual notations in a triangle (ABC)): $r = \frac{a}{2 \sin(\alpha)} = \frac{b}{2 \sin(\beta)} = \frac{c}{2 \sin(\gamma)}$. Applied to triangles $(PQR), (PQS)$, and (PQT) , this gives for b and c : $b = \frac{PR}{2} (\cot(\beta_1) + \cot(\eta_1))$ and $c = \frac{PQ}{2} (\cot(\gamma_1) + \cot(\delta_1))$. At last, the orientation $\zeta = (\mathbf{AB}, \mathbf{u})$ of the angle α in the grid is given too by $\zeta = (\mathbf{PQ}, (-1, 1))$. Inserting in (1) and (2) the preceding relations we obtain finally a criterion that apply to Delaunay triangulation data only.

2.7 Second criterion: preserving the orientation of angles

We have seen that an angle close to π could become non convex after rounding. Likewise, the rounding of an angle close to 0 can change its orientation, which can cause the overlapping of Voronoi cells and the appearance of non simple polygons. To avoid this, we want to know, given \mathcal{S} a set of sites in the plane and p a grid size, if each angle of the rounded Voronoi diagram of \mathcal{S} will keep

its orientation, or to obtain a grid size p_{MAX} that guaranties the preserving of the orientation. This problem is similar to the problem of convexity, and can be solved using the same methods. Finally, we obtain $p_{MAX} = \frac{\min(b,c)/\sqrt{2}}{\cos(\zeta)+\sin(\zeta) \cot(\alpha)}$.

3 Use of the criterions

Poisson Voronoi diagrams: Poisson Voronoi diagrams are diagrams generated by a set of sites uniformly distributed in the plane, which is relatively well approximated by an uniform distribution in a finite surface. Okabe et al. [11], [10] have shown that the probability density function of a randomly selected angle α of a typical Voronoi cell in \mathbb{R}^2 is $f(\alpha) = \frac{4}{3\pi} \sin(\alpha) (\sin(\alpha) - \alpha \cos(\alpha))$. In particular, it follows from this distribution that $p(\alpha < \pi/2) = \frac{1}{6}$, which shows that the convexity problem is much more frequent than the orientation problem.

Statistics on the pre-criterions: Experimentally, we found that, for uniform distributions of sites, about 1/3 of the angles are in a risky configuration regarding the convexity or the orientation of the rounded angles. This ratio does not depend on the density of the sites.

Criterions versus effective computation: One of the main purposes that we want to achieve is to have a way to detect the problems that occur when rounding a Voronoi diagram without computing the diagram itself. Obviously, this way must be cheaper than the computation of the rounded Voronoi diagram and testing its planarity and convexity. A careful study of the basic operations needed shows that above criterion can be computed in about half the time of the computation of the Voronoi diagram and its rounding.

Results on the criterions: The implementation of the criterions allowed us to make some statistics on its efficiency. The criterion is a sufficient but non necessary condition. Indeed, assuming a maximal move of half a pixel during rounding clearly under estimates the grid size needed for a correct rounding. The criterions detect as risky some angles that do not create problems when they are snap rounded. The figure 9 shows the correspondence between the criterion and the actual rounding.

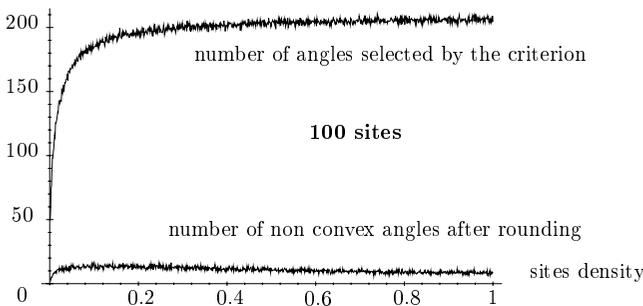


Fig. 9. Comparison of the criterion with the actual rounding

Adaptative grid size: As described in Section 2, the criterions can be used to certify that the snap rounding will preserve planarity and convexity or to determine a small enough grid size to ensure these properties. The interest of the criterions is to give directly a grid size relatively close to the maximal size, and thus avoid to test systematically the rounding of the Voronoi diagram with different grid sizes.

Our experiments (see Figure 1) show the mean value (for 100 experiences) of the maximal grid size allowed by the 2 criterions (convexity and planarity), as a function of the sites distribution.

Basic property of the Voronoi Diagram: The main advantage of the criterions is to validate the use of snap rounding. Indeed, the rounding of each vertex to the nearest grid point is the best way to preserve the tessellation to the closest points. It is easy to show than for a proximity query on the snap rounded diagram, the difference between the distance to the site given and the distance to the site that would have been given by the original diagram is less than $\sqrt{2}p$, where p is the grid size. If the point is to obtain the exact nearest site, a possibility is to use robust proximity queries methods [9].

4 Rounding heuristics

When the grid size cannot be chosen according to the distribution of the sites, the snap rounding can create bad angles. To preserve the fundamental properties of the Voronoi diagram, we will have to move some vertices not to the center point of the pixel which contains it, but in the center of a pixel in the “neighborhood” of the vertex (this notion of neighborhood remains to be defined). The main difficulty comes from the adjacency relations between angles. Indeed, moving a vertex in order to convexify a given angle, can generate non convex angles in the neighborhood of this angle (each Voronoi vertex belongs to 9 angles of the diagram).

We define briefly a rounding heuristic in the following way:

- the choice of a set of points candidates for the rounding of a given vertex,
- the choice of a cost function which gives the best candidate (in a way) of the set.

Definition 2 *We call kernel polygon associated to a Voronoi vertex v the convex polygon obtained as the intersection of (see Figure 10):*

- *The triangle joining the extremities of the angles containing v as apex,*
- *The 3 wedges defined by the pair of edges incident to the 3 neighbors of v and to v .*

As shown in Figure 10, the kernel polygon of a vertex v is the set of points where v can be moved preserving the convexity and the positive orientation of any angle involving v , when the others vertices are fixed. A diagram has the properties of convexity and planarity if and only if any vertex belongs to its kernel polygon.

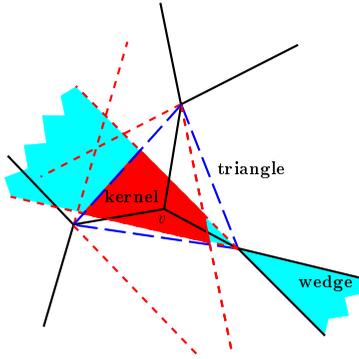


Fig. 10. Kernel polygon of a Voronoi vertex v (Voronoi diagram in continuous line)

If the snap rounding of the Voronoi diagram of \mathcal{S} is not convex and planar, the following heuristic can be used:

Algorithm *Rounded Voronoi*(\mathcal{S})

1. $\mathcal{VD} \leftarrow$ Voronoi diagram of \mathcal{S}
2. $\mathcal{RV}\mathcal{D} \leftarrow$ Snap rounding(\mathcal{VD})
3. **while** $\mathcal{RV}\mathcal{D}$ not planar and convex
4. **do**
5. $v \leftarrow$ A vertex of $\mathcal{RV}\mathcal{D}$ involved in a bad angle
6. $K(v) \leftarrow$ kernel of v in $\mathcal{RV}\mathcal{D}$
7. $v' \leftarrow$ A grid point inside $K(v)$ (if it exists)
8. Modify $\mathcal{RV}\mathcal{D}$ by moving v to v'
9. **return** $\mathcal{RV}\mathcal{D}$

At this point, this algorithm is not yet completely clear, we need to know how v is chosen among the bad vertices and how v' is chosen among the grid points inside $K(v)$. Furthermore, in case of failure, the algorithm, as presented above, fail in an infinite loop instead of ending with a clear status.

First implementation: A first simple approach to choose v is to use all bad angles in turn. The apex of the angle is tried first, then if its kernel contains no grid points the extremities of the angle are tried. The problem of the choice of v' can be solved easily by taking v' the grid point of $K(v)$ nearest to v . If for a given angle, none of the three vertices involved can be moved inside its kernel, then we end with a failure status. A rounding preserving convexity and planarity has not been found.

Such an approach works well in practice. If the density of points is not too high, then the bad angles are sparse in the diagram and the order used to examine them has no influence. Very often $K(v)$ contains grid points, more precisely, quite often at least one of the vertices of the triangle used in the kernel definition, say w , belongs to $K(v)$; rounding v to w has the effect of contracting the edge vw and increases the degree of the resulting vertex.

This algorithm has been implemented and gives very good results since in the worst case, it corrects all the bad angles but less than 0.05% of them. The

first two lines of table 1 correspond to data coming from a GIS database(raw data in line 1, dilated data on a scale of 24 bits integers in line 2), while the third line corresponds to an uniform distribution of points.

Number of points	Side of the square	Total number of angles	Risky angles for pre-criterions	Risky angles for criterions	Maximum grid size	Bad angles for actual rounding	Bad angles after heuristic
120973	58093	725727	304901/62992	76564/859	very small	8669/38	14/0
120973	11618600	725727	304901/62992	461/4	0.0011	52/0	0/0
1000000	100000	5999598	2190626/190760	175988/620	very small	14664/14	4/0

Table 1. Excerpt from the table of results (columns of risky or bad angles give: number of angles for convexity / for planarity)

Generalizations

Edge contraction: As noticed above, the algorithm still has some freedom to choose v and v' . We first can remark that the problems usually come from an initial angle which is close to π (convexity) or to 0 (orientation), or from an initial edge whose length is small. Problems created by small edges can be solved by promoting vertex merging, if $K(v)$ contains w , a neighbor of v close to v , then we can choose w as new position for v . This choice results in the contraction of edge vw .

Vertex rounding propagation: Since the difficulty may come from the propagation of the bad angles, it would be interesting to try to direct this propagation along a centrifugal axis, i.e. in direction of the convex hull of the initial sites. Indeed, the semi-infinite Voronoi angles are less constraining than the finite ones since they contain an infinite point. In particular, the “kernel polygon” for an infinite vertex is an angular sector, therefore it contains an infinite number of grid points (see Figure 11).

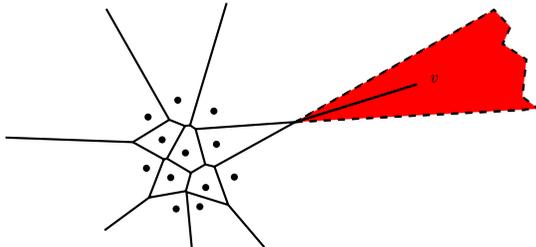


Fig. 11. Kernel polygon of an infinite Voronoi vertex

5 Conclusion

We have presented several results about the rounding of a Voronoi diagram preserving its planarity and the convexity of its cells. The first idea consists in snap rounding (rounding to the nearest grid point) all the vertices of the

diagram. We have established a reasonably cheap condition which ensures that the snap rounding preserves these properties. We have studied experimentally the efficiency of the snap rounding and the tightness of our condition on random Voronoi diagrams.

We have proposed an heuristic algorithm, which may round the Voronoi vertices further away in the grid, while preserving the desired properties. This algorithm works very well in practice as it is shown by our experimental results.

We have proven that, in the worst case, it is impossible to preserve planarity and convexity and to guarantee a fixed bound on the distance between a vertex and its rounded version. This paper proves that, with reasonable hypotheses on the data distribution, snap rounding and some heuristic will actually succeed to round the Voronoi diagram with good probability.

Another, more theoretical, direction of research consists in searching a deterministic algorithm of rounding preserving convexity and planarity such that something is provable on the distance between a Voronoi vertex and its rounded correspondent. Since a constant bound is not achievable, a bound depending on the distance to the nearest site would be a good result.

Acknowledgements

The authors would like to thank Jack Snoeyink for having graciously provided points sets from his GIS database covering the region of Vancouver.

References

1. J.-D. Boissonnat. Shape reconstruction from planar cross-sections. *Comput. Vision Graph. Image Process.*, 44(1):1–29, October 1988.
2. J.-D. Boissonnat and B. Geiger. Three dimensional reconstruction of complex shapes based on the Delaunay triangulation. In R. S. Acharya and D. B. Goldgof, editors, *Biomedical Image Proc. and Biomedical Visualization*, volume 1905, pages 964–975. SPIE, 1993.
3. J.-D. Boissonnat and F. P. Preparata. Robust plane sweep for intersecting segments. Research Report 3270, INRIA, 1997.
4. O. Devillers. Computational geometry and discrete computations. In *Proc. 6th Discrete Geometry for Computer Imagery conf.*, 1996.
5. S. Fortune and C. J. Van Wyk. Efficient exact arithmetic for computational geometry. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 163–172, 1993.
6. Steven Fortune. Vertex-rounding a three-dimensional polyhedral subdivision. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 116–125, 1998.
7. M. Goodrich, L. J. Guibas, J. Hershberger, and P. Tanenbaum. Snap rounding line segments efficiently in two and three dimensions. In *Proc. 13th Annu. ACM Sympos. Comput. Geom.*, page 284, 1997.
8. L. Guibas and D. Marimont. Rounding arrangements dynamically. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, page 190, 1995.

9. G. Liotta, F. P. Preparata, R. Tamassia. Robust proximity queries in implicit Voronoi diagrams. Technical Report CS-96-16, Center for Geometric Computing, Comput. Sci. Dept., Brown Univ., Providence, RI, 1996.
10. R. E. Miles. On the homogenous planar Poisson point-process. *Math. Biosci.*, 6:85–127, 1970.
11. A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley & Sons, Chichester, UK, 1992.
12. C. Yap. Towards exact geometric computation. *Comput. Geom. Theory Appl.*, 7(1):3–23, 1997.