Rounding Voronoi Diagram

Olivier Devillers, Pierre-Marie Gandoin

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Rounding Voronoi diagram

Olivier Devillers — Pierre-Marie Gandoin

N° 3481

Aout 1998

THÈME 2

Rapport de recherche

ISSN 0249-6399

Rounding Voronoi diagram

Olivier Devillers , Pierre-Marie Gandoin

Thème 2 — Génie logiciel
et calcul symbolique
Projet Prisme

Rapport de recherche n° 3481 — Aout 1998 — 21 pages

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 at 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 preserve the properties. We also present heuristics to round vertices (not to the nearest) and preserve these properties.

Key-words: geometric computing, Voronoi diagram, integer coordinates, exact computations

This work was partially supported by ESPRIT LTR GALIA

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Arrondi du diagramme de Voronoï

Résumé : La géométrie algorithmique repose généralement sur l'utilisation de nombres exacts non représentables sur un ordinateur réel. Une solution consiste à utiliser des coordonnées entières et à faire du calcul exact sur celles-ci. Cette approche implique d'arrondir les résultats d'un algorithme si l'on veut pouvoir les réinjecter dans un autre algorithme.
Dans ce rapport, le cas du diagramme de Voronoï bidimensionnel est abordé: on cherche à arrondir les sommets de Voronoï aux points d'une grille en conservant les propriétés intéressantes du diagramme, telles que la planarité du plongement et la convexité des cellules. On donne une condition sur le pas de la grille pour garantir que l'arrondi des sommets de Voronoï aux points de la grille les plus proches conserve ces propriétés. On présente également des heuristiques d'arrondi pour le cas où le pas de la grille fait que l'arrondi au plus proche ne respecte pas ces propriétés.

Mots-clés : géométrie algorithmique, calcul géométrique, triangulation de Delaunay, diagramme de Voronoï, coordonnées entières, calcul exact

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 yields difficult precision problems in the implementation of geometric algorithms. Recent trends in computational geometry consists in using integer coordinates to represent the data, and exact integer arithmetic to make the computations [4, 5, 11].

Exact output. A solution to solve precision problem 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 integer, 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 chained 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 chained this approach yields an unacceptable increase in the precision used to store results.

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 results consist in points, they must be moved to coincide with the 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 rounded method preserving geometrical properties of objects. The rounding of geometric structure 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 used *snap rounding* where 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 $\mathcal{S}$ in the plane (see Definition 1). An example of chaining algorithms may be found in 3D reconstruction, where we need to compute the overlap of Voronoi diagram [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 way of rounding than snap rounding.

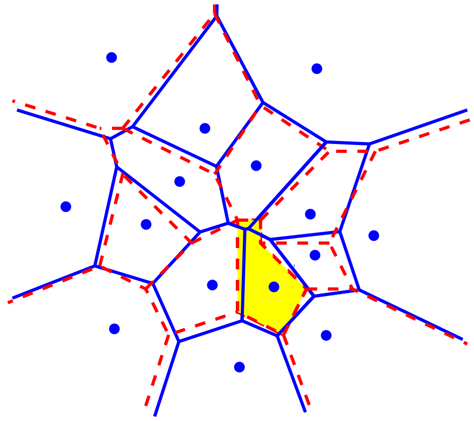


Figure 1: Rounding of Voronoi diagram

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.

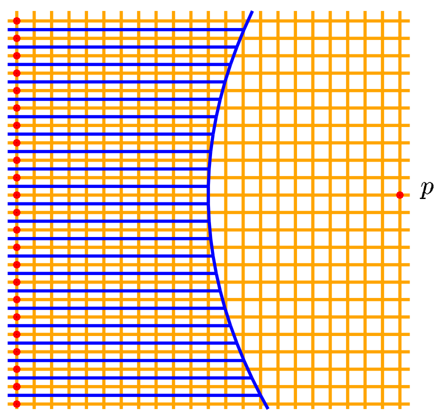


Figure 2: In this example, a "good" rounding does not exist

Rounded computational geometry versus discrete geometry The use of grid in computational geometry imply some convergence with fields such 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 16000000×16000000 (for 24 bits integer) 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 definition.

Definition 1 *Let $\mathcal{S}$ be a set of n points in the plane, $M_1, \ldots, M_n$, 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 $\mathcal{S}$:*

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

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-spaces (bounded by the perpendicular bisector of M_iM_j, $j = 1, \ldots, n$, $j \neq i$). This shows that $V(M_i)$ is a convex polygon, which may or may not be bounded. The $V(M_i)$'s and their edges form a cell complex whose domain is the whole $\mathbb{R}^2$, called Voronoi diagram of $\mathcal{S}$.

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

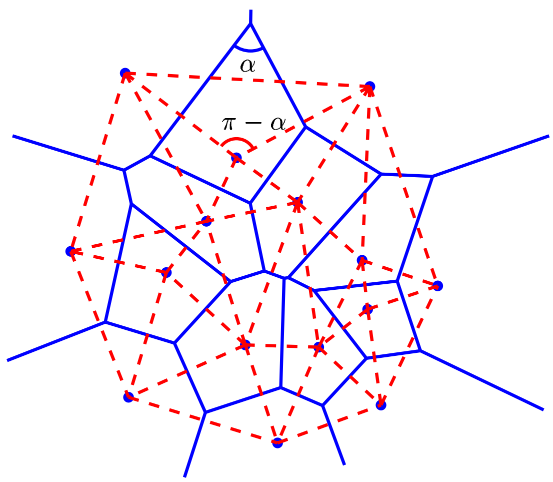


Figure 3: Duality between Voronoi (continuous lines) and Delaunay (dashed lines)

2 Sufficient condition of correct rounding

Given $\mathcal{S}$ a set of sites in the plane and p a grid step, 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 results into a planarity criterion. If the 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 depends only on the original data: the coordinates of points 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, and A', B', C' the snap rounded corresponding vertices. The angle $\alpha' = (A'B', A'C')$ has 7 degrees of freedom and thus depends of seven 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 the 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

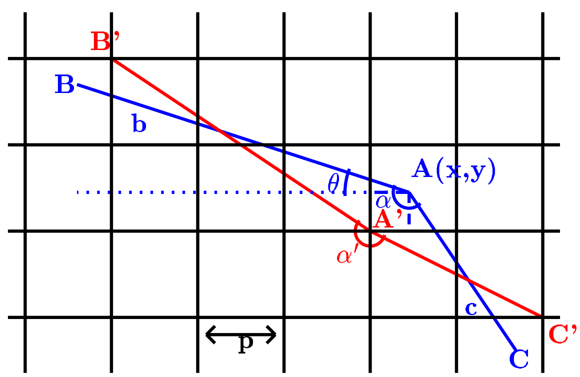


Figure 4: α' depends on 7 parameters

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-estimation.

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. $x_A \leq x_B \Rightarrow round(x_A) \leq round(x_B)$ and $y_A \leq y_B \Rightarrow round(y_A) \leq round(y_B)$) involves that the extremities B and C of the angle stay in the same quadrant of plane when they are rounded. 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 configuration*, 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.

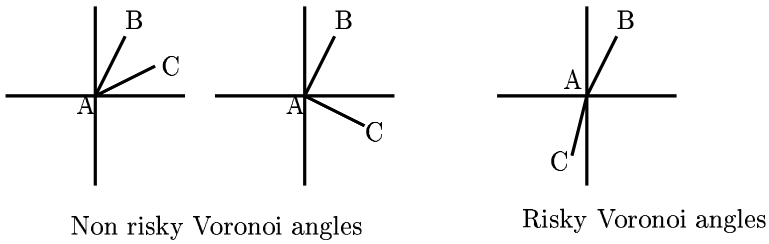


Figure 5: Use of the monotonicity of snap rounding

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 which maximize α'. A simple way to consider the worst case is to assume that the three points A, B, C lie in the center of a grid square and thus have a maximal move of the half diagonal of a pixel. 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.

2.4 Main criterion

The rounding of a point can be viewed as a translation of a certain vector $\vec{u}$. We will now use the fact that with the assumptions of previous paragraph, the vector of translation of points A, B, C have the same direction and the same norm. More precisely, if A is translated of $\vec{u}$, B and C are translated of $\vec{-u}$. But these 3 translations are equivalent - concerning the value of the rounded angle α' - to translate the point A of $2\vec{u}$ with B and C fixed. Since

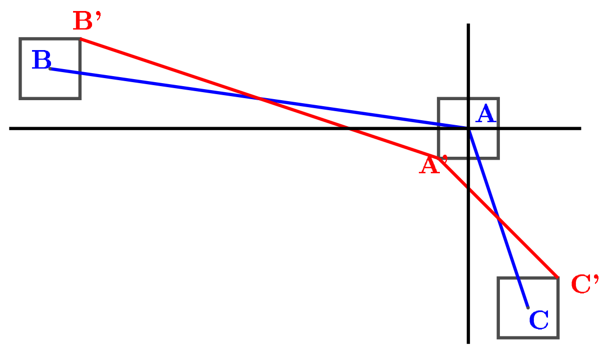


Figure 6: Position of A', B', C' maximizing α'

$\| \vec{u} \| = \sqrt{2}\,p$, it follows that the maximum grid size p_{MAX} such as α' is convex, is given by the length $h'/\sqrt{2}$ (see Figure 7). To calculate h', we will use the height h of the triangle

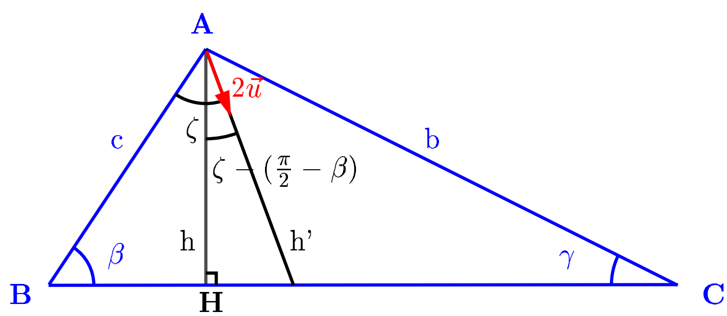


Figure 7: Calculation of a convexity criterion

ABC and angle $\zeta = \theta + \frac{\pi}{4}$. Basic relations of triangle geometry give:

$$h = c\,\sin(\beta) \tag{1}$$

$$h' = \frac{h}{\cos(\frac{\pi}{2} - \beta - \zeta)} \tag{2}$$

After reduction, we obtain:

$$h' = \frac{c}{\cos(\zeta) + \sin(\zeta)\,\cot(\beta)} \tag{3}$$

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:

$$\cot(\beta) = \frac{c - b\,\cos(\alpha)}{b\,\sin(\alpha)} \tag{4}$$

So finally, the maximum grid size warranting convexity of rounded Voronoi diagram is:

$$p_{MAX} = \frac{c/\sqrt{2}}{\cos(\zeta) + \sin(\zeta)\,\frac{c-b\,\cos(\alpha)}{b\,\sin(\alpha)}} \tag{5}$$

2.5 Semi-infinite Voronoi angles

We call semi-infinite Voronoi angles the angles containing an infinite point, i.e. composed 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 of $2\,\vec{u}$. Therefore, we have to express the length h' as a function of b, c, α and ζ:

$$h = c\,\sin(\alpha) \tag{6}$$

$$h' = \frac{h}{\cos(\frac{\pi}{2} - (\alpha - \zeta))} \tag{7}$$

After reduction, we obtain:

$$h' = \frac{c}{\cos(\zeta) - \sin(\zeta)\,\cot(\alpha)} \tag{8}$$

It follows that the maximum grid size such as the rounded semi-infinite angle remains convex is:

$$p'_{MAX} = \frac{c/\sqrt{2}}{\cos(\zeta) - \sin(\zeta)\,\cot(\alpha)} \tag{9}$$

2.6 From Voronoi to Delaunay

We have thus obtain a criterion expressed on some length 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 (see Figure 3) by

$$\alpha = \pi - \alpha_1 \tag{10}$$

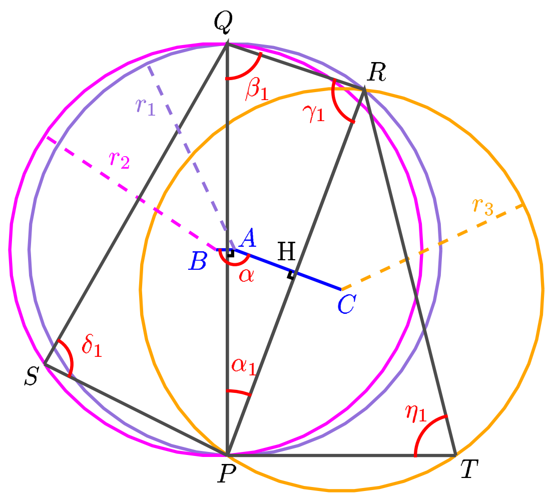


Figure 8: How express Voronoi data from Delaunay data

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 radius of circumscribed circles are respectively r_1, r_2 and r_3.
We obtain for b and c:

$$b = r_1 \cos(\beta_1) + r_3 \cos(\eta_1), \tag{11}$$
$$c = r_1 \cos(\gamma_1) + r_2 \cos(\delta_1). \tag{12}$$

For the radius of the circumscribed circles, we have the relation:

$$r = \frac{a}{2 \sin(\alpha)} = \frac{b}{2 \sin(\beta)} = \frac{c}{2 \sin(\gamma)}. \tag{13}$$

Applied to triangles (PQR), (PQS), and (PQT), this gives finally for b and c:

$$b = \frac{PR}{2} (\cot(\beta_1) + \cot(\eta_1)), \tag{14}$$
$$c = \frac{PQ}{2} (\cot(\gamma_1) + \cot(\delta_1)). \tag{15}$$

At last, the orientation $\zeta = (\vec{AB}, \vec{u})$ of the angle α in the grid is given too by:

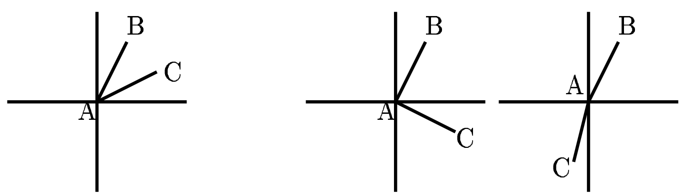
$$\zeta = (\vec{PQ}, (-1, 1)). \tag{16}$$

Inserting in 5 and 9 the relations 10, 14, 15 and 16, 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 simples 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} which guaranty the preserving of the orientation. This problem being similar to the problem of convexity, we will treat it more briefly.

We use again the monotonicity of snap rounding in order to obtain the following pre-criterion: given an angle whose extremities do not lie strictly in the same quadrant of plane, the corresponding rounded angle is necessarily $\geq \pi/2$. Therefore, the main criterion (for the preserving of orientation) will be computed only for the angles which are in a risky configuration concerning the orientation (see Figure 9).



Risky Voronoi angles

Non risky Voronoi angles

Figure 9: Use of the monotonicity of snap rounding

Now, we assume again that the vertices of the Voronoi angle have a maximal move of an half diagonal of a unit square. Comparing to convexity criterion, the worst case rounding is obtained with different translation s than in the convexity case. Without loss of generality, if $c \leq b$, the worst case is obtained by translating A and C of $-\vec{v}$ and B of $\vec{v}$ (see Figure 10). Finally we get:

$$p_{MAX} = \frac{\min(b,c)/\sqrt{2}}{\cos(\zeta) + \sin(\zeta)\ \cot(\alpha)} \tag{17}$$

It can be easily shown that the case of semi-infinite angles gives exactly the same expression of p_{MAX}.

At last, to pass from the criterion on Voronoi diagram to a criterion on Delaunay triangulation, the whole expressions 10, 14, 15 can be reused. The only change is the expression

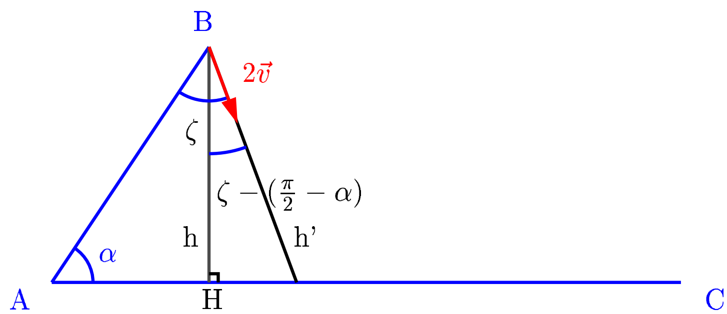


Figure 10: Calculation of an orientation criterion

of ζ, that becomes:

$$\zeta = (\vec{AB}, \vec{v}) = (\vec{PQ}, (-1, -1)) \tag{18}$$

3 Use of the criterion

3.1 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. [10], [9] have shown that the probability density function of 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.

3.2 Statistics on the pre-criterion

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 depends on the density or the number of the sites.

3.3 Complexity of the main criterion

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. We show that above criterion can be computed in about half the time of the computation of the Voronoi diagram and its rounding.

3.3.1 Complexity of direct detection

For each Voronoi angle, we have to compute:

- Computation of the 3 vertices coordinates:

  A Voronoi vertex being the center of the circumscribed circle of the corresponding Delaunay triangle, it is define by the intersection of two perpendicular bisectors. To compute these perpendicular bisectors equations: Mid-point of the segment $[PQ]$:

  $$2\,x_I = x_P + x_Q, \quad 2\,y_I = y_P + y_Q$$

  Direction of the line (PQ):

  $$x_{\vec{u}} = x_Q - x_P, \quad y_{\vec{u}} = y_Q - y_P$$

  The equation of the perpendicular bisector is $y = a\,x + b$, with:

  $$a = -\frac{x_{\vec{u}}}{y_{\vec{u}}}, \quad 2\,b = 2\,y_I - a\,2\,x_I$$

  For 3 Voronoi vertices, we need to compute 6 line equations, that is to say: 30 additions, 6 multiplications, 6 divisions, 12 affectations.

  The coordinates of the intersection point of two lines $y = a\,x + b$ and $y = a'\,x + b'$ are given by:

  $$x_A = \frac{2b' - 2b}{2\,(a - a')}, \quad y_A = a\,x_A + \frac{2\,b}{2}$$

  Hence, for the 3 Voronoi vertices: 9 additions, 6 multiplications, 6 divisions, 6 affectations.

- Rounding of the coordinates to the nearest grid:

  This cost depends on the grid size. We can consider that this is about the cost of an addition for an integer grid.

- Comparison of 2 orientation tests of 3 points (the vertices of the angle before and after rounding):

  An orientation test consists in the computation of a 3×3 determinant, that is to say 10 additions et 12 multiplications for the two tests.

Finally, we count 49 additions, 24 multiplications, 12 divisions, 24 affectations and 6 scalar roundings.

Remark 1 *The complexity calculated here corresponds to the computation of the coordinates of the Voronoi vertices only. If we want construct a planar map with Voronoi edges and ray, the complexity can increase significantly due to the management of the planar map.*

3.3.2 Complexity of the criterion

For a given Voronoi angle, we only have to compute one of the two criterions (convexity and orientation). We will study here the complexity of the criterion of convexity for a finite angle. The others cases (semi-infinite angles, criterion of orientation) are slightly cheaper.

To avoid the calculation of square roots, we will use the following variables:

$$b_1 = \frac{b}{PR} = \frac{1}{2}\left(\cot(\beta_1) + \cot(\eta_1)\right), \quad c_1 = c\,PQ = \frac{PQ^2}{2}\left(\cot(\gamma_1) + \cot(\delta_1)\right)$$

$$n_{\cos(\zeta)} = x_{PQ} - y_{PQ}, \quad n_{\sin(\zeta)} = x_{PQ} + y_{PQ}$$

$$n_{\cos(\alpha)} = x_{PQ}\,x_{PR} + y_{PQ}\,y_{PR}, \quad n_{\sin(\alpha)} = x_{PQ}\,y_{PR} - y_{PQ}\,x_{PR}$$

Then the criterion becomes:

$$p_{MAX} = \frac{c_1/(PQ\sqrt{2})}{\frac{n_{\cos(\zeta)}}{PQ\sqrt{2}} + \frac{n_{\sin(\zeta)}}{PQ\sqrt{2}}\,\frac{\frac{c_1}{PQ} - b_1\,PR\,\frac{n_{\cos(\alpha)}}{PQ\,PR}}{b_1\,PR\,\frac{n_{\sin(\alpha)}}{PQ\,PR}}} \tag{19}$$

That is to say, after reduction:

$$p_{MAX} = \frac{c_1}{n_{\cos(\zeta)} + n_{\sin(\zeta)}\,\frac{c_1 - b_1\,n_{\cos(\alpha)}}{b_1\,n_{\sin(\alpha)}}} \tag{20}$$

Hence for each Voronoi angle, we have to compute:

- coordinates of vector PQ:

  $$x_{\vec{PQ}} = x_Q - x_P, \quad y_{\vec{PQ}} = y_Q - y_P$$

  So for the vectors PQ, PR, QR, PS, QS, PT, RT, we count 14 additions and 14 affectations.

- squared distance PQ^2: 1 addition, 2 multiplications, 1 affectation.

- 4 cotangents (for b_1 and c_1): 8 additions, 16 multiplications, 4 divisions and 4 affectations.

- numerators of $\sin(\alpha)$, $\cos(\alpha)$, $\sin(\zeta)$, $\cos(\zeta)$: 4 additions, 8 multiplications, 4 affectations.

Finally, we count 31 additions, 31 multiplications, 6 divisions and 25 affectations.

3.3.3 Global complexity of the criterion with the pre-criterion

To know the configuration of a Delaunay angle in respect with the plane quadrants, we have to sort lexicographically the 3 vertices of the angle, which costs 6 tests in the worst case. But sorting the 3 vertices of a Delaunay face allows to treat 3 angles. The mean complexity of the pre-criterion is thus 2 tests for an angle.

We have seen in 3.2 that the pre-criterion eliminated about 2/3 of the Delaunay triangulation angles. Therefore, with 2 tests for each angle, the main criterion complexity is reduce of 2/3 in mean. Hence the mean complexity for each Delaunay angle is about: 2 tests, 10 additions, 10 multiplications, 2 divisions and 8 affectations.

Remark 2 *The pre-criterion can also be used with the actual rounding of the diagram in order to detect bad angles. That is why we have done the comparison without using the pre-criterion. The practical results show that the computation of the criterion on the Delaunay triangulation takes about half the time of the computation of the Voronoi diagram itself from the triangulation.*

3.4 Results on the criterion

The implementation of the criterion 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 estimate the grid size needed for a correct rounding. The criterion detect as risky some angles that do not create problems when they are snap rounded. The following graphs show the correspondence between the the criterion and the actual rounding.

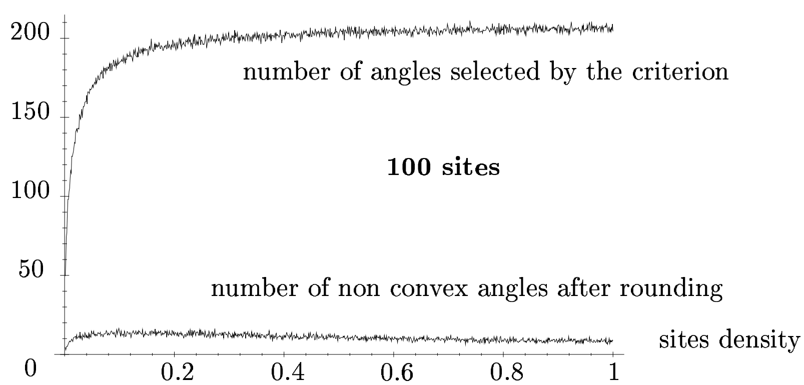


Figure 11: Comparison of the criterion with the actual rounding

3.5 Adaptative grid size

As described in Section 2, the criterion 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 criterion 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 13) shows 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.

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 in the center 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 define). 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 12):*

- *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 12, 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.

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

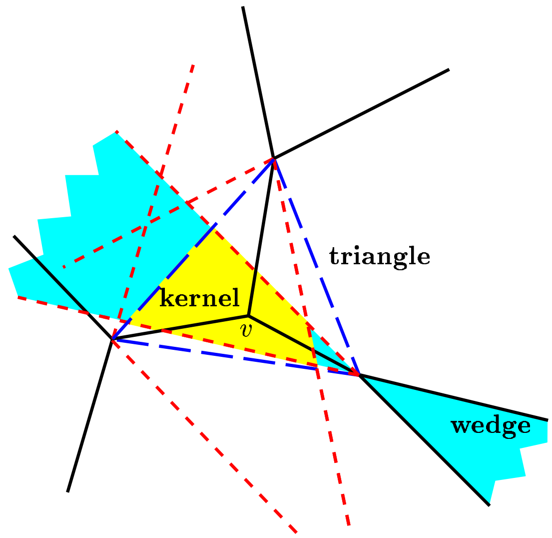


Figure 12: Kernel polygon of a Voronoi vertex v (Voronoi diagram in continuous line)

Algorithm *Rounded Voronoi*(S)

$\mathcal{VD} \leftarrow$Voronoi diagram of $\mathcal{S}$
$\mathcal{RVD} \leftarrow$Snap rounding($\mathcal{VD}$)
while $\mathcal{RVD}$ not planar and convex
    do
        $v \leftarrow$A vertex of $\mathcal{RVD}$ involved in a bad angle
        $K(v) \leftarrow$kernel of v in $\mathcal{RVD}$
        $v' \leftarrow$A grid point inside $K(v)$ (if it exists)
        Modify $\mathcal{RVD}$ by moving v to v'
return $\mathcal{RVD}$

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 to 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 last three lines of figure 13 correspond to data coming from a GIS database(raw data in the first line, dilated data on a scale of 24 bits integers in the next two lines), while the others lines correspond to an uniform distribution of points.

Generalizations **Edge contraction.** As noticed above, the algorithm still have a bit a freedom to choose v and v'. We first can remark that the problems usually comes 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 result in the contraction of edge vw.

Vertex rounding propagation Since the difficulty may comes 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 14).

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 ensure 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 diagram.

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 prove that, with reasonable hypotheses on the data distribution, snap rounding and some heuristic will actually succeed to round the Voronoi diagram with good probability.

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
1000	10	598	249/50	0/0	Infinity	0/0	0/0
1000	100	5627	2131/255	1119/1	very small	86/1	0/0
1000	1000	5937	2187/208	484/2	3.1e-03	37/0	0/0
1000	10000	5939	2184/204	52/0	3.0e-02	5/0	0/0
1000	100000	5939	2185/206	5/0	2.7e-01	1/0	0/0
1000	1000000	5940	2186/206	1/0	2.7e+00	0/0	0/0
10000	10	600	249/49	0/0	Infinity	0/0	0/0
10000	100	37636	14885/2339	846/0	very small	9/1	0/0
10000	1000	59594	21914/2050	11327/31	very small	816/2	1/0
10000	10000	59918	21919/1946	1726/6	7.3e-04	143/0	0/0
10000	100000	59922	21914/1940	170/1	8.1e-03	16/0	0/0
10000	1000000	59921	21920/1946	18/0	7.8e-02	2/0	0/0
100000	10	600	245/45	0/0	Infinity	0/0	0/0
100000	100	59989	25014/5018	0/0	Infinity	0/0	0/0
100000	1000	570623	214332/24124	115592/88	very small	8823/73	25/0
100000	10000	599566	219103/19248	51002/164	very small	3940/6	4/0
100000	100000	599904	219152/19184	5619/21	1.3e-04	489/0	0/0
100000	1000000	599893	219049/19085	540/2	2.9e-03	51/0	0/0
1000000	10	600	244/44	0/0	Infinity	0/0	0/0
1000000	100	60000	24959/4959	0/0	Infinity	0/0	0/0
1000000	1000	3790866	1497194/233572	78151/0	very small	368/27	1/0
1000000	10000	5969757	2189097/199178	1149273/3160	very small	83233/213	139/0
1000000	100000	5999598	2190626/190760	175988/620	very small	14664/14	4/0
1000000	1000000	5999883	2190143/190182	17336/66	7.6e-05	1489/1	0/0
120973	58093	725727	304901/62992	76564/859	very small	8669/38	14/0
120973	11618600	725727	304901/62992	461/4	1.1e-03	52/0	0/0
906347	11698600	5437908	2056172/243536	2801/73	very small	155/0	0/0

Figure 13: Table of results.
Columns of risky or bad angles give: number of angles for convexity / for planarity

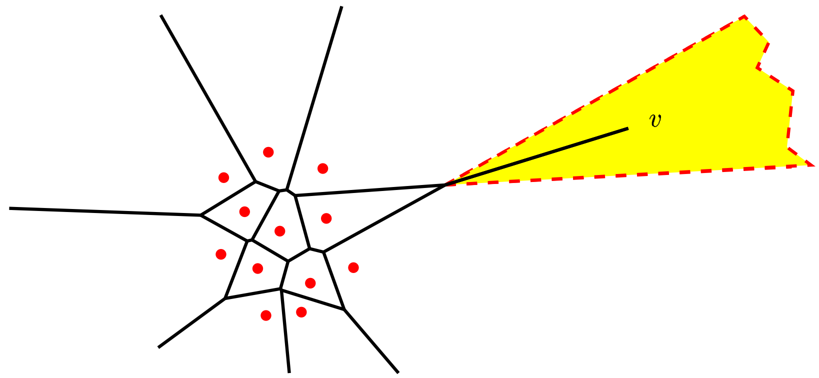


Figure 14: Kernel polygon of an infinite Voronoi vertex

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.

Acknowledgement
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 Processing 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. (invited paper).

[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.*, pages 284–293, 1997.

[8] Leonidas Guibas and David Marimont. Rounding arrangements dynamically. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 190–199, 1995.

[9] R. E. Miles. On the homogenous planar Poisson point-process. *Math. Biosci.*, 6:85–127, 1970.

[10] Atsuyuki Okabe, Barry Boots, and Kokichi Sugihara. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams.* John Wiley & Sons, Chichester, UK, 1992.

[11] C. Yap. Towards exact geometric computation. *Comput. Geom. Theory Appl.*, 7(1):3–23, 1997.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399