INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Inner and Outer Rounding of Set Operations on Lattice Polygonal Regions

Olivier Devillers — Philippe Guigue

N° 5070

December 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--5070--FR+ENG

INRIA
SOPHIA ANTIPOLIS

Inner and Outer Rounding of Set Operations on Lattice Polygonal Regions

Olivier Devillers , Philippe Guigue

Thème 2 — Génie logiciel
et calcul symbolique
Projet Geometrica

Rapport de recherche n° 5070 — December 2003 — 23 pages

Abstract: Robustness problems due to the substitution of the exact computation on real numbers by the rounded floating point arithmetic are often an obstacle to obtain practical implementation of geometric algorithms. If the adoption of the *exact computation paradigm* [12] gives a satisfactory solution to this kind of problems for purely combinatorial algorithms, this solution does not allow to solve in practice the case of algorithms that cascade the construction of new geometric objects. In this report, we consider the problem of rounding the intersection of two polygonal regions onto the integer lattice with inclusion properties. Namely, given two polygonal regions A and B having their vertices on the integer lattice, the inner and outer rounding modes construct two polygonal regions $A \underline{\cap} B$ and $A \overline{\cap} B$ with integer vertices such that $A \underline{\cap} B \subseteq A \cap B \subseteq A \overline{\cap} B$. We also prove interesting results on the Hausdorff distance, the size and the convexity of these polygonal regions.

Key-words: high level geometric rounding, finite precision geometry, intersection, polygons

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Arrondi par défaut et par excès d'opérations booléennes sur des régions polygonales

Résumé : Les problèmes de robustesse liés à la substitution du calcul exact sur les réels par le calcul flottant approché de la machine sont souvent un obstacle à l'implantation pratique des algorithmes géométriques. Si l'adoption du *paradigme du calcul exact* [12] apporte une solution satisfaisante à ce type de problèmes pour les algorithmes ayant un résultat purement combinatoire, cette solution ne permet cependant pas de résoudre en pratique le cas des algorithmes qui réutilisent voire cascadent la construction de nouveaux objets géométriques. Dans ce rapport, nous abordons le problème de l'arrondi sur la grille entière de l'intersection de deux régions polygonales avec des propriétés d'inclusion. Plus précisèment, étant donné deux regions polygonales A et B ayant leurs sommets sur la grille entière, le mode par *défaut* et le mode par *excès* construisent deux régions polygonales $A \underline{\cap} B$ et $A \overline{\cap} B$ ayant leurs sommmets sur la grille entière telles que $A \underline{\cap} B \subseteq A \cap B \subseteq A \overline{\cap} B$. Nous prouvons de plus certaines propriétés intéressantes sur la distance de Hausdorff, la taille et la convexité de ces régions arrondies.

Mots-clés : arrondi géométrique, géométrie à précision fixée, intersection, polygones

1 Introduction

Many geometric algorithms are designed in the Real RAM model, and the use of rounded floating point arithmetic is well known to create robustness problems: Numerical rounding errors done the evaluation of geometric predicates lead to inconsistent results and cause trouble in computer data structures. The now classical solution of the *exact computation paradigm* [12] offers an attractive solution for algorithms that do not construct new geometric objects such as convex hulls or triangulations i.e whose results are purely combinatorial (the position of the points is not the result but the input of the algorithm). The exact computation paradigm approach takes decisions on an exact basis. To achieve reasonably efficient computation times this induces to use well defined exact representations of geometric objects: Typically, the coordinates of a point are assumed to be fixed size integers.

However, the exact computation paradigm is less satisfactory for algorithms that compute the geometric embedding of new objects. An intersection point between two line segments is a relevant example of a construction of a new geometric object. Such a point has rational coordinates and therefore is generally not representable on the integer lattice. If this point is used by the algorithm to take a decision, we must have an exact representation of that point e.g. using rational numbers or implicit representation [3] in order to ensure the exactness of that decision. One drawback of this approach is that a constructed point does not look like a data point[1] and thus in such a framework, algorithms cannot be easily *cascaded*, i.e. the (rational) output from one algorithm cannot be used as input for another algorithm designed for usual (integer) input.

An alternative consists in rounding the constructions that is replacing a geometric structure with arbitrary bit-length coordinates by an approximating structure with (short) fixed bit-length coordinates. However, rounding the coordinates of geometric objects like vertices of a polygonal region is not straightforward since incidence information may be invalidated by small perturbations of edges and vertices. For instance, a polygonal region may be initially convex or simple and can loose these properties after a simple rounding of its vertices' coordinates. Since these properties might be reused by other algorithms, this loss of information is problematic.

Yet, there exist few published work in this direction, except for rounding line segment arrangements in the plane while preserving the topology of the arrangement [6, 5, 7, 8, 10] (see Section 2) and for rounding polyhedral subdivisions in three dimensions [1]. In this report, we are concerned with rounding the result of the intersection of two planar lattice polygonal region (i.e whose vertices have integer coordinates). The result will extend trivially to any other set operations on pairs of lattice polygonal regions. Conversely to the arrangement problem, we are interested in inclusion properties between the exact object and its rounded versions. Previous works on arrangements can therefore not be used directly. We propose in this report an algorithm to guarantee the preservation of such properties (see Figure 2).

[1] It has been shown [4, 10] that no matter which exact numerical representation is used, the space required to represent a vertex or edge of a new object grows exponentially with the height of its construction tree.

Section 4 introduces the concept of inner and outer rounding of the intersection of lattice polygonal regions. Section 5 deals with the practical computation of these approximations. Sections 6 proves that a point on the boundary of a rounded version is at distance less than $\sqrt{2}$ from the exact boundary, and that convexity is somehow preserved. Finally, Section 7 and 8 generalize these rounding modes to other set operations and to general polygonal regions.

2 Related Work

Three techniques for rounding line segments arrangements to a finite precision lattice have been proposed in the literature. All methods proceed by rounding the intersection points between the input line segments to their nearest lattice point. Each original line segment is then replaced by a polygonal chain connecting the rounded version of the endpoints and visiting all its rounded intersection points. The techniques described below differ in the way that the polygonal chains are constructed in order to guarantee metric and topological properties[2].

Greene-Yao perturbation technique The first method by Greene-Yao [6] treats each lattice point as an obstacle and forbids any segment to go over an obstacle while its intersection points move to their nearest lattice point. Instead, an obstacle is introduced as a new vertex into the polygonal chain representing the segment. The authors show that with this technique, edges move by a distance at most $\frac{\sqrt{2}}{2}$. This algorithm has the disadvantage that it produces very fragmented polygonal chains, which has an adverse effect on the efficiency of algorithms and operations that use these fragmented line segments. Namely, this technique introduces $\Omega(\log|ab|)$ *excess* lattice points onto a segment ab where $|ab|$ denotes the length of the segment ab.

Later papers tried to reduce the number of additional vertices without introducing larger geometric derivations.

Snap Rounding Paradigm Various researchers [5, 7, 8] have developed the Snap Rounding technique for rounding line segments to the integer lattice. The idea behind Snap Rounding is as follows. The plane is partitioned into pixels (i.e. isothetic unit squares) centered at integer lattice points. A pixel is called *hot* if it contains a vertex of the original arrangement (that is either an endpoint or an intersection point of the input segments). The embedding is then rounded as follows: Each original line segment is replaced by a polygonal chain that connects the centers of the hot pixels crossed by the segment. This way, the number of vertices on an edge is equal to the number of hot pixels crossed by the edge.

Guibas and Marimont [7] give a very nice analysis of the properties of Snap Rounding. One of its main properties is that it does not introduce any extra lattice points. Moreover,

[2] We refer the reader to [6] for an exhaustive inventory of shortcomings of the use of a simple rounding that maps each vertex of a line segments arrangement to its nearest representable point.

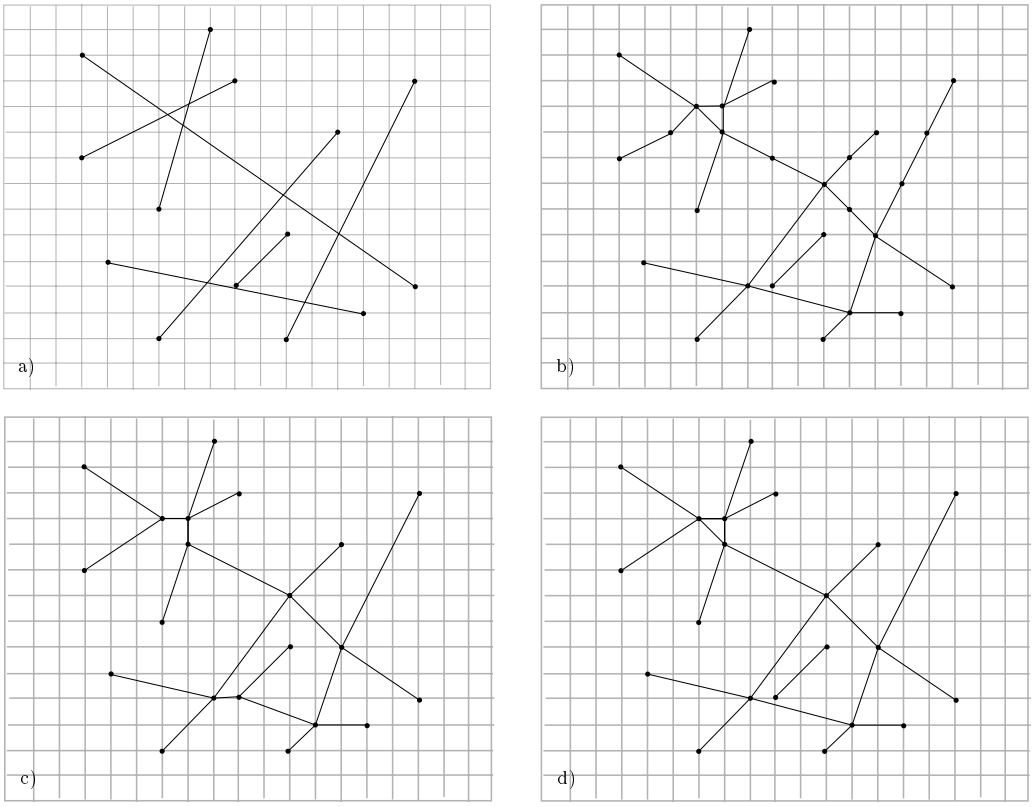

Figure 1: a) A line segment arrangement, b) Its rounded version with the Greene-Yao perturbation technique, c) Its rounded version with the Snap Rounding technique, and d) Its rounded version with the Shortest Path Rounding technique.

it can be easily shown that the polygonal chain corresponding to an original segment is contained within the Minkowski sum of the original segment with a pixel (*unit square*) centered at the origin.

Shortest Path Rounding technique Shortest Path Rounding has been introduced by Milenkovic [10, 11] and introduces even fewer additional incidences between the rounded segments than Snap Rounding. The basic idea is to round each intersection point to its nearest lattice point and to replace each edge by the shortest path connecting the rounded endpoints that keeps all other rounded vertices at the correct side. This technique has the advantage that it introduces minimum geometric and combinatorial error (it gives the same result as the Snap Rounding method in the worst case). Moreover, unlike other finite

precision geometric rounding techniques, Shortest Path Rounding can be applied to non uniform lattices.

Although these different techniques allow to preserve somehow the topology of the exact arrangement, they do not offer any inclusion or convexity guarantees if they are applied on faces (and not only edges) of the arrangement. The rounding modes proposed in this report are inspired from the presented methods however they respond to the demand of such guarantees.

3 Notations and preliminaries

By a *lattice point* or *grid point* we mean a point in $\mathbb{Z}^2$. A *lattice polygon* is a polygon that defines a well defined interior and exterior (we permit incidences of the polygons' features) and whose vertices are lattice points. A *lattice polygonal region* is a plane figure which can be expressed as a collection of lattice polygons having nested holes at any level of depth.

In the following, the complexity of a polygonal region P defined as the number of distinct vertices of P is denoted by $|P|$. The *interior* of a polygonal region P, defined as the biggest open set contained in P, is denoted by P^o. The *boundary* of P is denoted by ∂P. We will say that a point p belongs to a polygonal region P, and note $p \in P$, if p belongs either to the interior or to the boundary of P. Finally, P^C will denote the set complement of P.

Given two polygonal regions A and B, the Hausdorff distance $d_H(A, B)$ between A and B is defined as

$$d_H(A, B) = max(d_h(A, B), d_h(B, A))$$

where $d_h(A, B) = \max_{a \in A} \min_{b \in B} d(a, b)$ and $d(a, b)$ denotes the Euclidean distance between these points.

We will use the following definition of visibility. For two points p and q that belong to a polygonal region P, we say that q is *visible* from p within P, if every point of the line segment pq lies in P. The *visibility region*, $V_P(p)$, of a point $p \in P$ is defined as the locus of all points $q \in P$ that are visible from p. The *nearest visible lattice point* of p, denoted by $v_P(p)$, is defined as the nearest grid point to p that belongs to $V_P(p)$ with any tie-breaking rule if p is equidistant to several lattice points. Finally, for a vertex $v \in P$ and an edge $e \in P$, we say that v is *vertically visible* from e, if it exists a vertical line segment that connects v to e that is entirely contained in P.

We describe in the next section the scheme used to define the inner and the outer rounding of a polygonal region and state the properties of the rounding scheme in the case where the input regions result from the intersection of two lattice polygonal regions. Note that from the application of Morgan laws, all set operations reduce to the complementary operation (whose computation is trivial) and to the intersection operation. Section 7 enumerates the

properties satisfied when the exact region to be rounded comes from an union or a set difference operation.

4 Rounding Modes

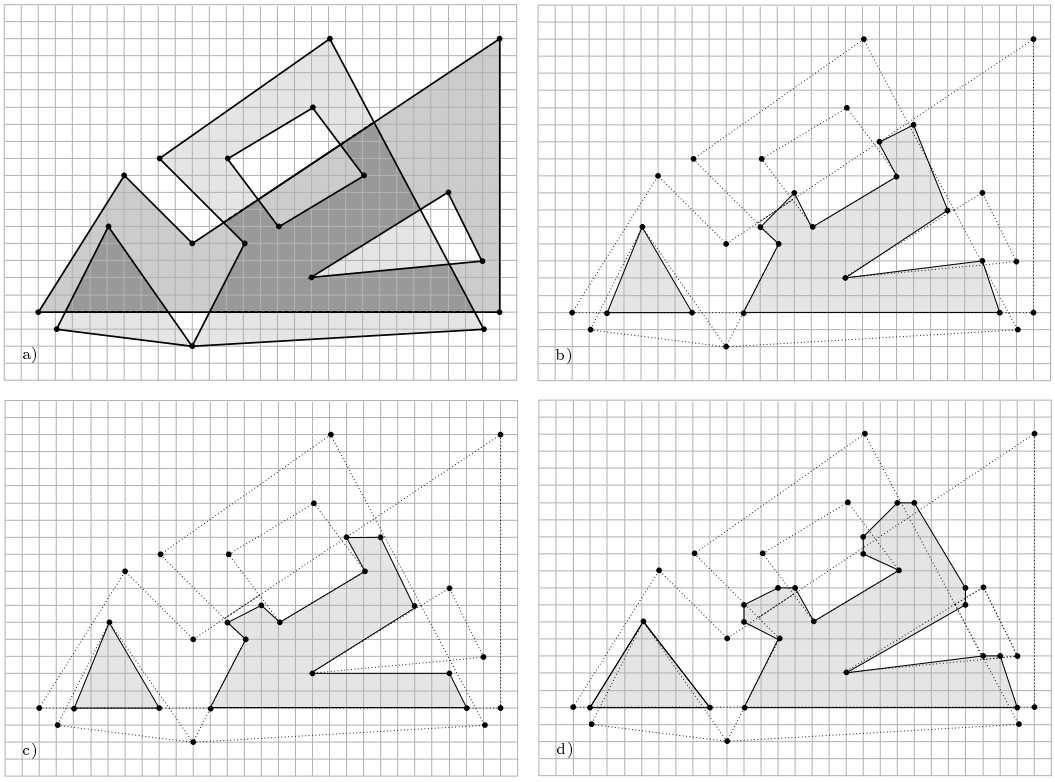

Figure 2: a) The two input lattice polygonal regions and their exact intersection region P. b) The rounded version of P with the Shortest Path Rounding technique. c) The inner rounding $\underline{P}$. d) The outer rounding $\overline{P}$.

4.1 Inner Mode

Suppose we start with two numerically consistent input lattice polygonal regions: For the given values of their coordinates, the regions have a well defined interior and exterior. One can intuitively visualize the rounding process of the polygonal region corresponding to the intersection of these two regions using the analogy used by Greene and Yao [6]. Look at the

edges of this region as if they were rubber bands rooted at their two endpoints and let every vertex of this region be marked by a rigid post. Each of these rigid post is then treated as an obstacle and we do not allow the rubber bands to go over an obstacle while each of the posts move to their nearest visible lattice point. Now, if we release rigid posts that correspond to vertices that have lost their convexity (vertices that were convex and became concave), then the resulting polygonal region gives the inner rounded polygonal region.

Theorem 1 states some properties of the obtained rounded region in the case where P corresponds to the intersection of two lattice polygonal regions A and B (the proof is postponed until the Section 6.1).

Theorem 1 *The inner rounding $\underline{P}$ of $P = A \cap B$ satisfies the following properties:*
1) $\underline{P}$ is a numerically consistent lattice polygonal region,
2) $\underline{P}$ is contained in P,
3) $d_H(\underline{P}^C, (P^o)^C) < \sqrt{2}$,
4) $|\underline{P}| \leq |P|$,
5) A concave vertex of $\underline{P}$ does always correspond to a concave vertex of P.

From property 5) we have the following corollary:

Corollary 2 *If P_i is a convex component of $P = A \cap B$ and if $\underline{P_i}$ is not empty then $\underline{P_i}$ is a convex component of $\underline{P}$.*

4.2 Outer Mode

Given two numerically consistent polygonal regions in the plane, the process leading to the computation of the outer rounding of their exact intersection region can be split in three step. The idea is to bring the problem back to an inner intersection computation (cf. Figure 3). To do so, the exact intersection region P is first computed. Then, for each vertex $v = (v_x, v_y)$ of P that is not representable on the integer lattice is associated a pixel (unit square of the grid) having respectively $(\lfloor v_x \rfloor, \lfloor v_y \rfloor)$ and $(\lceil v_x \rceil, \lceil v_y \rceil)$ as bottom left and top right vertex. The outer rounding $\overline{P}$ of P is then obtained from P and the set I of all pixels containing non representable vertices of P by carrying out the operation $((P^C) \underline{\cap} (I^C))^C$. A last pass removes all extraneous reflex vertices of the obtained polygonal region (see Section 5.3).

Theorem 3 states some properties of the obtained rounded region in the case where P corresponds to the intersection of two lattice polygonal regions A and B (the proof is postponed until the Section 6.2).

Theorem 3 *The outer rounding $\overline{P}$ of $P = A \cap B$ satisfies the following properties:*
1) $\overline{P}$ is a numerically consistent lattice polygonal region,
2) $\overline{P}$ contains P,
3) $d_H(\overline{P}, P) < \sqrt{2}$,
4) $|\overline{P}| < |P| + 3k + h$, where k is the number of non-lattice vertices of P and h is the total number of intersecting pairs between the edges of P and those of I.

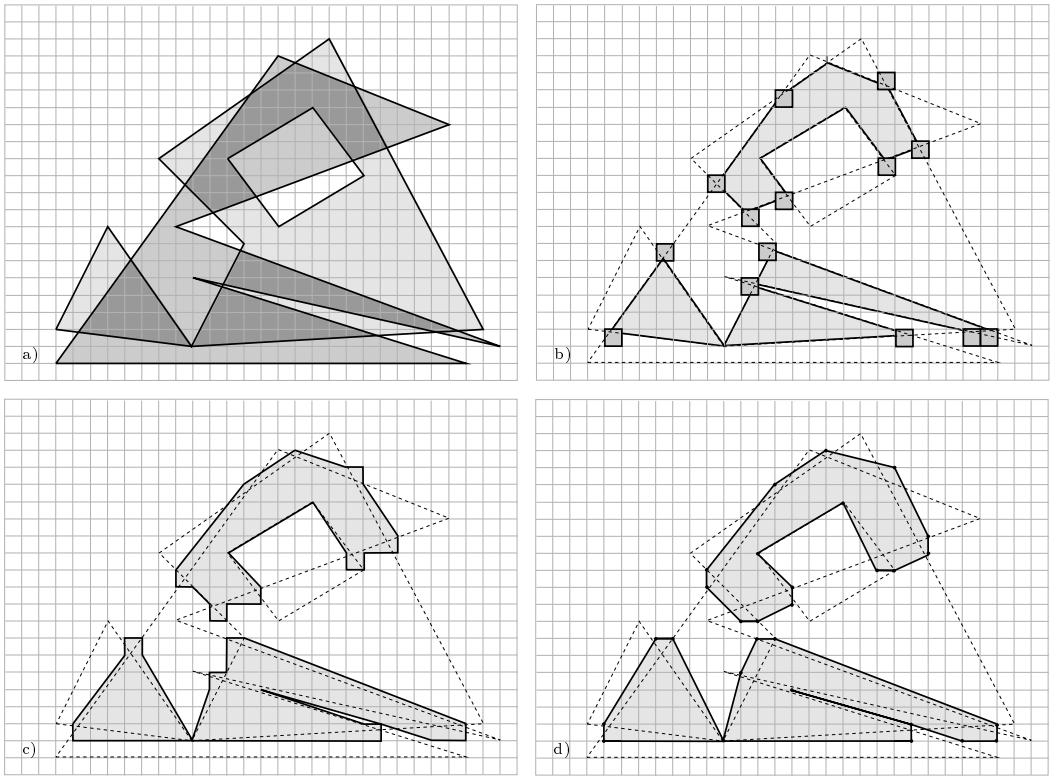

Figure 3: a) The two input lattice polygonal regions. b) Their exact intersection region P and the set of pixel I. c) The rounded version obtained by computing $(P^C \underline{\cap} I^C)^C$. d) The region $\overline{P}$ obtained by removing superfluous reflex vertices.

5 Practical Algorithms

From the analogy used in the Section 4.1, it is easy to see that each rigid post that corresponds to a vertex of P and that causes an edge of the intersection region to be broken during its perturbation corresponds to a reflex vertex of the exact intersection region. Given two lattice polygonal regions A and B, the only vertices of the polygonal region $P = A \cap B$ that are not representable onto the integer lattice (that is the only vertices that need to be rounded) correspond to the intersection points between an edge of A and an edge of B. From the definition of the intersection operation, these non representable vertices can only form a convex vertex of P. Consequently, each reflex vertex of P comes from a reflex vertex of one of the two input regions and is therefore a lattice vertex.

The algorithm for rounding the intersection of two lattice polygonal region with the inner mode is essentially based on the reflex vertical decomposition of the exact intersection region. The purpose of the construction of this map is twofold: 1) It gives a convex decomposition of the original region that will permit to avoid complex visibility calculation, 2) It determines for each edge of the region a subset of the original vertices that should be snapped in order to avoid the introduction of extraneous intersections.

5.1 The Reflex Vertical Decomposition

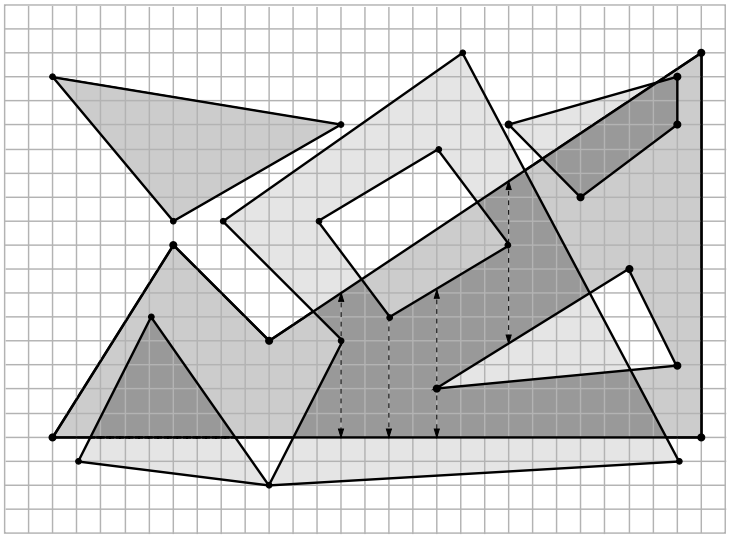

Figure 4: The reflex vertical decomposition of the intersection of two lattice polygonal regions.

The reflex vertical decomposition of a planar polygonal region is constructed by extending from each reflex vertex of the input region two vertical rays in the interior of the region in both the upward and downward directions. These rays are the maximal vertical segments such that their relative interior does not intersect any edge of the polygonal region. The reflex vertical decomposition of a polygonal region i.e. the subdivision of this region induced by the edges of the region and by the rays issued from its reflex vertices is a partition of the input region into convex cells (see Figure 4).

Before detailing the practical algorithm we first prove some properties of this decomposition.

Lemma 4 *Given P the exact intersection of planar lattice polygonal regions, p a vertex of P and C a convex cell of the reflex vertical decomposition of P having p as vertex then $v_P(p) = v_C(p)$.*

Proof: We prove this by contradiction. Suppose that $v_P(c) \neq v_C(p)$. As $v_P(p)$ and $v_C(p)$ must be distinct points $v_P(p)$ cannot belong to C. Therefore, the line segment connecting p to $v_P(p)$ must cross the boundary of C (cf. Figure 5). Since $v_P(p)$ is visible from p, the crossed boundary can only be a vertical wall emanating from a reflex vertex. Yet, this is impossible since, in this case, the two lattice points on the crossed ray immediately above and below the crossing are closer to p than $v_P(p)$. One of these two lattice points is between the crossing and the source of the ray and thus inside C and visible by convexity of C. This contradicts the fact that $v_P(p)$ cannot belong to C and therefore the claim we made in the proof. □

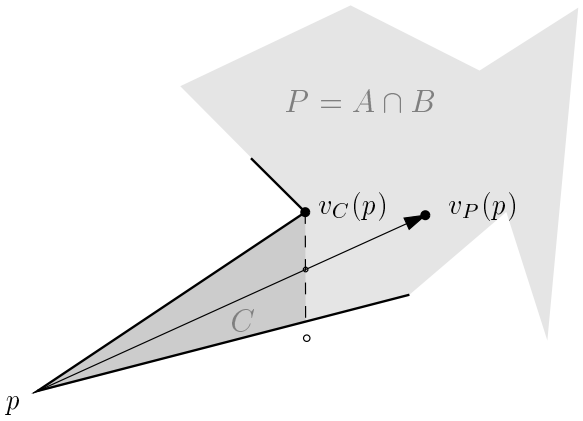

Figure 5: If p is a vertex of a convex cell C then $v_P(p) = v_C(p)$.

Lemma 5 *Let P be the exact intersection of planar lattice polygonal regions, pq be an edge of P, $\underline{P}$ be the inner rounding of P, and $\sigma(pq)$ be the polygonal chain connecting $v_P(p)$ to $v_P(q)$ that corresponds to the rounded counterpart of pq in $\underline{P}$. The set of vertices of $\sigma(pq)$ between $v_P(p)$ and $v_P(q)$ are reflex vertices of P vertically visible from pq in P.*

Proof: By construction of the polygonal chain $\sigma(pq)$, the vertices of $\sigma(pq)$ between $v_P(p)$ and $v_P(q)$ necessarily correspond to reflex vertices of P. We show in the following that these vertices are vertically visible from pq in P. Here again, we prove this by contradiction. Assume that there exists a vertex c of $\sigma(pq)$ between $v_P(p)$ and $v_P(q)$ such that c is not vertically visible from pq in P. Since c belongs to $\sigma(pq)$ and is not vertically visible from pq in P, c surely lie in one of the two x-intervals induced by the segments $pv_P(p)$ and $qv_P(q)$ (cf. Figure 6). Suppose wlog that c belongs to the x-interval induced by the segment $pv_P(p)$ and let i be the point of intersection between $pv_P(p)$ and the vertical line L passing through c. Since c is a reflex vertex of P, c must lie at a lattice site and L is a lattice vertical line. But this is impossible, since in this case there exists a lattice point r on L between i and c that is visible from P and closer to p than $v_P(p)$, which contradicts the fact that $v_P(p)$ is the nearest visible lattice point from p in P. □

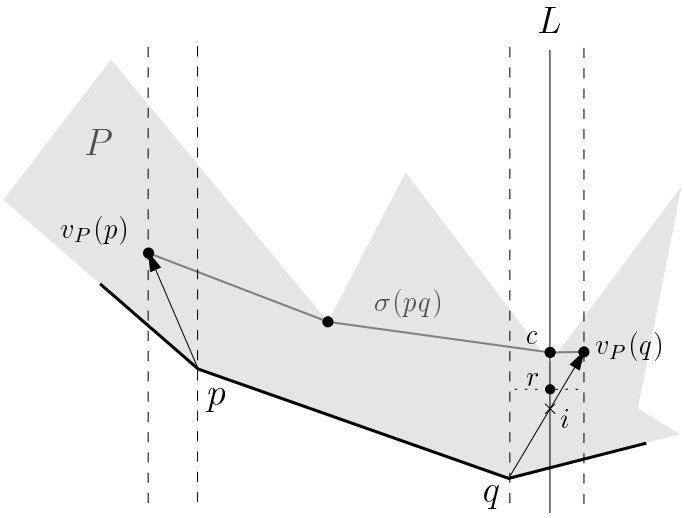

Figure 6: If c is a vertex of $\sigma(pq)$ between $v_P(p)$ and $v_P(q)$ then c is vertically visible from pq in P.

5.2 Inner Intersection Algorithm

Let A and B be two lattice polygonal regions in the plane. The algorithm works in three steps. The first step constructs the arrangement of the edges of A and B and computes the reflex vertical decomposition of the intersection region $P = A \cap B$ by the use of a Bentley-Ottmann-like sweep line algorithm.

Based on this vertical decomposition, the second step rounds each vertex of P that does not lie at lattice site to the nearest visible lattice point that belongs to its incident convex cell in the vertical decomposition. At the same time, each edge of P is replaced by a polygonal chain that connects its two rounded endpoints and passes through the set of all its vertically visible reflex vertices in the order of their vertical projection on the edge.

The last step finally performs a variant of the Graham's scan algorithm for the convex hull computation over the set of the resulting polygons (or holes). This procedure removes all the reflex vertices from each polygon/hole except the ones corresponding to original reflex vertex (that is, it removes each reflex vertex that corresponds to a rounded intersection point or a visited vertically visible reflex vertex).

Given a vertex v of P and its associated convex cell C, the computation of the nearest visible lattice point of v in C can be done using the algorithm described in [2] in time $O(m \log m \log N)$ where $m = |C|$ and $N \times N$ is the size of the lattice containing C. This algorithm, based on the continued fraction expansion technique, is inspired from the algorithm developed by H.S.Lee and R.C.Chang [9] which solves the problem in time $O(m + \log l)$, where l is the diameter of the convex cell. However, this latter needs the use of an exact

arithmetic on algebraic numbers to be implemented robustly (while our algorithm in [2] is of degree 4 and can be implemented using a rational arithmetic).

Theorem 6 *The inner rounding $\underline{P}$ of a region $P = A \cap B$ can be computed in time $O((n+k)\log n + k|P|\log|P|\log N)$ where n is the total number of edges of the two input regions, k is the number of edges of A and B that intersect and $N \times N$ is the size of the lattice.*

Proof: Given the two input regions A and B, the reflex vertical decomposition of their intersection region is a by-product of the trapezoidal map of their edges. Therefore it can be calculated in time $O((n+k)\log n)$ where n is the total number of edges of A and B and k is the number of intersecting pairs. The second step of the algorithm computes at most k nearest visible lattice points in convex cells of size at most $|P|$ in time $O(|P|\log|P|\log N)$ and produces, in the worst case, a set of polygons/holes having a total of $|P|+2r$ vertices where r is the number of reflex vertices of P (each reflex vertices being vertically visible from at most two edges of P). Given an edge of the intersection region and its two rounded endpoints, its associated polygonal chain can be constructed in time linear with the number of intersections between the edge and the vertical walls of the decomposition and thus can be done in time $O(|P|)$. Putting all together and since $r < |P|$ and $k \leq |P|$ we obtain a worst case complexity of $O((n+k)\log n + k|P|\log|P|\log N)$ for the whole algorithm. □

5.3 Outer Intersection Algorithm

The algorithm for computing the outer rounding of the intersection of two lattice polygonal regions is essentially based on the algorithm of Section 5.2 and can be directly deduced from the description given in Section 4.2. However, we discuss here a way to reduce the number of extraneous reflex vertices of $\overline{P}$, namely the extraneous reflex vertices of $\overline{P}$ issued from the vertices of the pixels of I, that derive from the straightforward computation of $\overline{P}$ as $(P^C \underline{\cap} I^C)^C$.

Contrary to the inner rounding of an intersection region, the outer rounding mode (as described in Section 4.2) does not offer any guaranty on the convexity/concavity preservation of the exact region's vertices. Some reflex vertices of P can disappear in $\overline{P}$, in the same manner some extraneous reflex vertices (that correspond to vertices of I and thus do not appear in P) can appear in $\overline{P}$. A simple improvement consists in removing all reflex vertices appearing in $\overline{P}$ if they have no counterpart in P and if their removing does not produce any topological change. Some precautions must be taken in order to preserve a maximal distance between the points of $\overline{P}$ and the points of P less than $\sqrt{2}$. A solution may consist in removing a reflex vertex r of $\overline{P}$ only if there exists an edge e of P such that r, the vertex preceding and the vertex following r on $\overline{P}$'s boundary all lie at a distance less than $\sqrt{2}$ from e. This kind of simplification permits a reduction of extraneous reflex vertices of $\overline{P}$ of a factor $O(k)$ in the best case. Moreover, this additional pass is sufficient to guarantee as a side-effect the following property: If no components of P are merged in $\overline{P}$ (that is if P and $\overline{P}$ have exactly the same number of polygons) then a convex component of P remains convex in $\overline{P}$.

Theorem 7 *The outer rounding $\overline{P}$ of a region $P = A \cap B$ can be computed in time $O((n+h)\log n + kp\log p\log N)$ where n denotes the total number of vertices of the two input regions, h denotes the number of intersection points between the edges of A and the edges of B, $N \times N$ is the size of the lattice, k denotes the number of intersection points between an edge of P and an edge of the set of pixels I and $p = \max(|P|, |P^C \cap I^C|)$.*

Proof: The computation of the exact intersection region P can be done in time $O((n+h)\log n)$ where h denotes the number of intersection points between the edges of A and the edges of B. The computation of the reflex vertical decomposition of $(P^C \cap I^C)$ can then be computed in time $O((|P|+h+k)\log(|P|+h))$ where k is the number of intersection points between the edges of P and the edges of I. Finally, the algorithm computes at most k nearest visible lattice points in convex cells of size at most equals to $m = |P^C \cap I^C|$ in time $O(m\log m\log N)$ using the algorithm described in [2] and produces a set of polygons/holes having a total number of vertices in $O(m)$. The final step of the algorithm is linear in the number of vertices of each polygons. Since $h \leq |P|$ and with $p = \max(|P|, |P^C \cap I^C|)$, we obtain a worst case complexity of $O((n+h)\log n + kp\log p\log N)$ for the whole algorithm. □

6 Proofs of Properties

6.1 Inner Intersection

We now prove that the algorithm of Section 5.2 computes an inner approximation of $A \cap B$ that satisfies the properties stated in Theorem 1. We first need the following lemmas:

Lemma 8 *The computed polygonal region is a numerically consistent polygonal region.*

Proof: We prove that no extraneous intersections are introduced in the final approximation (though new incidences are permitted). Let $C_{i=0..p}$ be the set of all convex cells of the vertical decomposition of the exact intersection. For each C_i, let l_i and r_i be the two vertical lines that pass through respectively the leftmost and the rightmost lattice point of C_i (cf. Figure 7). Now, if the intersection of C_i with l_i (resp. r_i) is a wall of C_i, let l_i^{down} and l_i^{up} (resp. r_i^{down} and r_i^{up}) be the lower and the upper intersection of l_i (resp. r_i) with C_i and let $\mathbf{l}_i$ (resp. $\mathbf{r}_i$) be the point on l_i (resp. r_i) that corresponds to the reflex vertex from where the wall is stemming from. Otherwise, let $l_i^{down} = l_i^{up}$ (resp. $r_i^{down} = r_i^{up}$) equal the leftmost (resp. the rightmost) vertex of C_i and let $\mathbf{l}_i = v_P(l_i^{down})$ (resp. $\mathbf{r}_i = v_P(r_i^{down})$).

The rounded counterparts of the polygonal chains connecting l_i^{down} to r_i^{down}, respectively r_i^{up} to l_i^{up}), are convex (by convexity of the original chains), therefore they are guaranteed not to lie above, respectively below, the edge $\mathbf{l}_i\mathbf{r}_i$ and thus cannot invert in C_i. □

Lemma 9 *All vertices of the computed polygonal region lie at lattice point within the exact intersection region.*

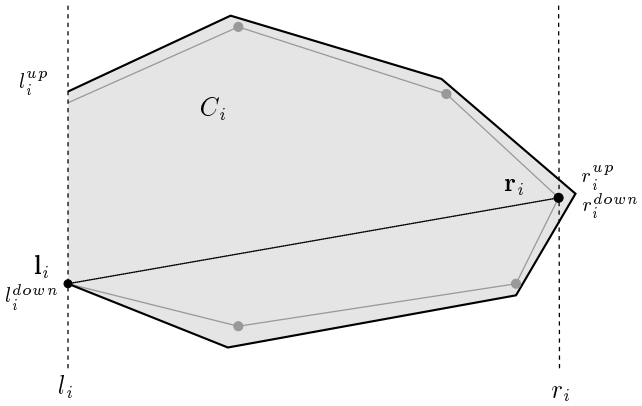

Figure 7: The rounded counterparts of the polygonal chains connecting l_i^{down} to r_i^{down} and l_i^{up} to r_i^{up} have opposite convexity and cannot invert in C_i.

Proof: There actually exist three types of vertices in the final approximation: rounded intersection points, original input vertices and snapped vertices corresponding to vertically visible input reflex vertices. Since each intersection point rounds to its nearest visible lattice point, the first type of vertex is guaranteed to lie at lattice point within the intersection region. The two other types of vertices correspond to lattice vertices of the exact intersection region. □

Observation 10 *Given pq an edge of the exact intersection region, the polygonal chain $\sigma(pq)$ that connects $v_P(p)$ to $v_P(q)$ and corresponds to the rounded counterpart of pq in $\underline{P}$ is entirely contained in P by construction.*

Lemma 11 *Let p be a vertex of P and $L(P)$ be the union of all lattice points, unit lattice segments and unit lattice squares that belong to the interior or to the boundary of P. The segment connecting p to $v_P(p)$ cannot intersect the interior of $L(P)$.*

Proof: To intersect the interior of $L(P)$, the segment $pv_P(p)$ must intersect the interior of a unit lattice segment s of $\partial L(P)$ (cf. Figure 8). The two endpoints of s are necessarily closer to p than $v_P(p)$ and therefore cannot be visible from p since they correspond, by definition of $L(P)$, to lattice points that lie inside P. Consequently, the relative interior of the segments connecting p to these endpoints must intersect the boundary of P. But this is impossible since by definition both segments $pv_P(p)$ and s cannot intersect in their interior the boundary of P and there cannot exist any visible reflex (lattice) vertex of P inside the triangle having p and the two endpoints of s as vertices since all points of this triangle are closer to p than $v_P(p)$. □

Lemma 12 *Given two lattice polygonal regions A and B of the plane, $d_H((A \underline{\cap} B)^C, (A^o \cap B^o)^C) < \sqrt{2}$.*

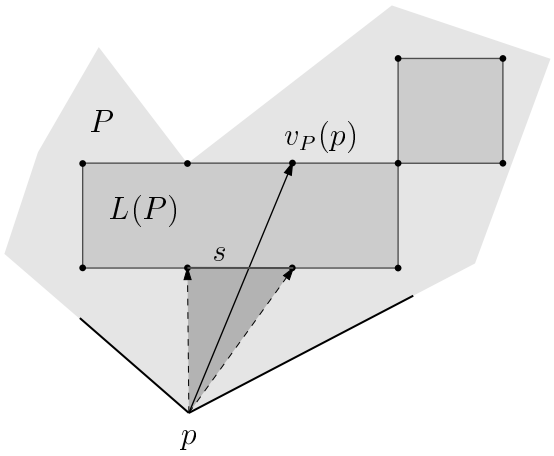

Figure 8: The segment $pv_P(p)$ cannot intersect the interior of L.

Proof: From Observation 10, $\underline{P} = A \underline{\cap} B$ is included in $P = A \cap B$ and the directional Hausdorff distance $d_h((P^o)^C, \underline{P}^C)$ is zero. We prove in the following that all points of $P \setminus \underline{P}$ are at a distance less than $\sqrt{2}$ from ∂P. Let pq be an edge of P and let E_{pq} be the polygon obtained by appending the rounded chain $\sigma(pq)$ to qp. Notice that $(\bigcup_{pq \in P} E_{pq})$ partitions $P \setminus \underline{P}$ except for the polygons p_i of P that do not have any rounded counterpart. The bound is trivially proven for the latter polygons since they do not contain any lattice point. For the non-trivial case, we conclude that it is sufficient to prove that $d_h(E_{pq}, \partial P) < \sqrt{2}$ for any edge pq of P.

Let $L(P)$ be the union of all lattice points, unit lattice segments and unit lattice squares that belong to the interior or to the boundary of P. By definition of $L(P)$, all points of $P \setminus L(P)^o$ are at a distance less than $\sqrt{2}$ of ∂P, we therefore suppose in the following that E_{pq} is not entirely included in $P \setminus L(P)^o$.

By lemma 11, the segments $pv_P(p)$ and $qv_P(q)$ cannot intersect the interior of $L(P)$ thus for E_{pq} to intersect $L(P)^o$, $\sigma(pq)$ must necessarily intersect $L(P)^o$. Moreover, by convexity of the chain $\sigma(pq)$, there must exist in this case at least one (lattice) vertex v different from $v_P(p)$ and $v_P(q)$ that lies in or on the boundary of E_{pq}. Suppose wlog that pq is oriented from left to right with a positive or zero slope and that the interior of P lies above pq. Finally, let v_l the xy-smallest point (w.r.t the lexicographic order) of the set S of all lattice points different from $v_P(p)$ and $v_P(q)$ that lie in or on the boundary of E_{pq} (cf. Figure 9).

Using the same arguments as in the proof of lemma 5 and since v_l is the xy-smallest point of S, it is easy to show that v_l is vertically visible from pq and that the vertical unit lattice segment having v_l as top vertex surely intersects pq in a point i_p. Similarly, since pq is oriented from left to right and has a positive or zero slope and since by lemma 11 the segments $pv_P(p)$ and $qv_P(q)$ cannot intersect the interior of $L(P)$, the horizontal unit

lattice segment having v_l as right vertex surely intersects $\sigma(pq)$ in a point j_p. Notice that, by construction, both i_p and j_p belong to the boundary of a same unit lattice square so that $\|i_p j_p\| < \sqrt{2}$.

Replacing p by q and applying a symmetry operation on E_{pq} such that qp is oriented from left to right and has a positive or zero slope with the interior of P above qp, we define similarly two points i_q and j_q on pq and $\sigma(pq)$ such that $\|i_q j_q\| < \sqrt{2}$. We conclude that $d_h(E_{pq}, \partial P) < \sqrt{2}$ since the polygons $pi_p j_p v_P(p)$ and $j_q i_q q v_P(q)$ are contained in $P \setminus L(P)^o$ (by definition of v_l) and the polygon $i_p i_q j_q j_p$ is contained in the Minkowski sum of $i_p i_q$ with the interior of a disc of radius $\sqrt{2}$ centered at the origin (cf. Figure 9), and by convexity of $\sigma(pq)$, the portion of $\sigma(pq)$ between j_p and j_q is included in $i_p i_q j_q j_p$. □

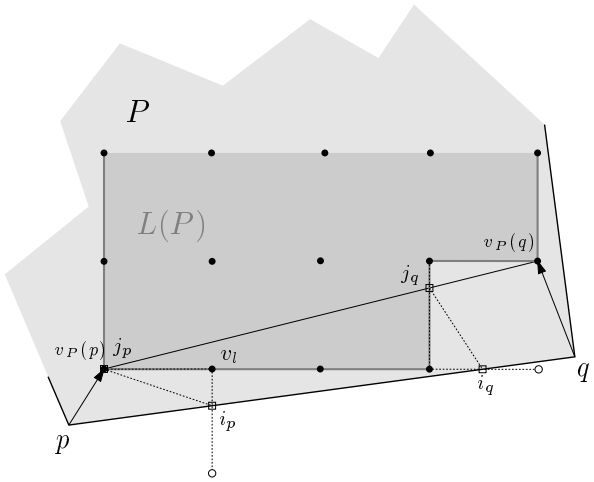

Figure 9: The polygons $pi_p j_p v_P(p)$ and $j_q i_q q v_P(q)$ are contained in $P \setminus L^o$ and the polygon $i_p i_q j_q j_p$ is at a distance less than $\sqrt{2}$ to pq.

We are now able to prove Theorem 1. Proof of property 1) comes from the combination of Lemmas 8 and 9. From 1) and by construction of the approximation the proof of Property 2) is trivial. Property 3) is proven in Lemma 12. Proof of property 4) comes from the fact that each intersection point rounds to at most one lattice point and that all extra vertices that appear in the approximation correspond to original reflex vertices of the exact intersection. The number of vertices of the final approximation is larger than the number of vertices of the original region only when a vertex of P is used several times in the approximation. Property 5) is a direct consequence of the last step of the algorithm since the convex-hull pass guarantees that no extra reflex vertices are introduced in the final approximation.

6.2 Outer Intersection

In this section, we introduce some lemmas needed for the proof of Theorem 3. Notice that property 3) cannot be deduced from Lemma 12 since we must bound the distance between the points of $\overline{P}$ to the *exact* intersection region P and not only to the region $(P^C \cap I^C)^C$. That is, we must exclude that there exist points of $\overline{P}$ that are close to a pixel of I but at a distance greater than $\sqrt{2}$ from the region P.

Lemma 13 *Given two lattice polygonal regions A and B of the plane, $d_H((A \,\overline{\cap}\, B\,), (A \cap B)) < \sqrt{2}$.*

Proof: Since $P = (A \cap B)$ is included in $\overline{P} = (A \,\overline{\cap}\, B\,)$, the (directional) Hausdorff distance from P to $\overline{P}$ is zero. Therefore, it is sufficient to show that each point of $\overline{P} \setminus P$ is at a distance less than $\sqrt{2}$ to P.

Note that since no pixel of I contains a lattice point in its interior, the union $L(P^C \cap I^C)$ of all the lattice points, lattice segments and pixels that belong to the interior or to the boundary of $(P^C \cap I^C)$ is also the union of all the lattice points, lattice segments and pixels that belong to the interior or to the boundary of P^C. Therefore, if the polygon E_{pq} (as defined in the proof of Lemma 12) is contained in $(P^C \cap I^C) \setminus L(P^C \cap I^C)^o$ it is also contained in $P^C \setminus L(P^C)^o$, and E_{pq} surely lies at a distance less than $\sqrt{2}$ to the boundary of P.

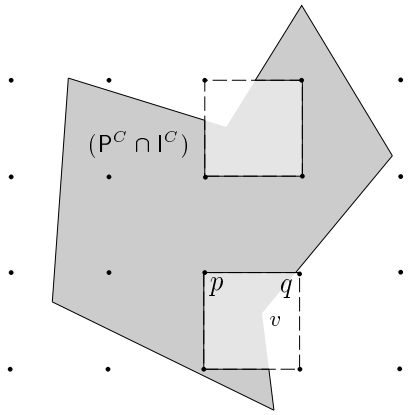

Figure 10: If pq is issued from an edge of I and at least one of its endpoints is a lattice point then $\sigma(pq) \subseteq pq$ (in the example $\sigma(pq) = p$) and $\sigma(pq)$ lies at a distance less than $\sqrt{2}$ to a vertex v of P.

Otherwise, with the same arguments as in the proof of Lemma 12, we show that the part of E_{pq} which is intersected by the interior of $L(P^C)^o$ surely lies at a distance less than $\sqrt{2}$ to pq. Therefore if pq is issued from an edge of P then the bound is trivially proven.

Otherwise, that is if pq is issued from a pixel Q of I, using the same arguments as in the proof of Lemma 12, there must exist a lattice line passing through v_l (as defined in the proof) that intersects pq. This line cannot intersect the relative interior of pq since the edge pq is included or equals a unit lattice segment. Moreover, if this line intersects pq in one of its endpoints then the intersected endpoint is necessarily a lattice point and therefore the rounded counterpart $\sigma(pq)$ of pq is included in pq (cf. Figure 10). We conclude in this case that $d_h(E_{pq}, \partial P) < \sqrt{2}$ since there must exist a vertex of P in the pixel Q (namely, the vertex of P that causes the presence of Q in I). □

Lemma 14 *The rounded region $\overline{P}$ of $P = A \cap B$ has less than $|P| + 3k + h$ distinct vertices where k is the number of non-lattice vertices of P and h is the total number of intersecting pairs between the edges of P and those of I.*

Proof: Since $\overline{P}$ corresponds to the complementary of the inner rounding $\underline{\mathsf{P}_I}$ of $P_I = (P^C \cap I^C)$, we have from Theorem 1 that $|\overline{\mathsf{P}_I}| \leq |P_I|$, and the number of vertices of $\overline{\mathsf{P}_I}$ is bounded by $|P_I|$. If P has a total of n vertices and has k vertices which are not representable on the integer lattice, $|I| \leq 4k$ and P_I has at most $(n-k) + 4k$ lattice vertices and h non integer vertices where h denotes the number of intersection point between P and I edges. □

Although the number of vertices of P_I used as an upper bound on the complexity of $\overline{P}$ can be in the worst case in $O(nk)$, an additional pass of the algorithm can be used to guarantee a total number of vertices of $\overline{P}$ which is linear in the number of vertices of the exact region P. More precisely, we show in [2] that the removal of all zero-area components (that is polygons or holes of $\overline{P}$ that have no interior) from the obtained region allows to bound the worst case number of distinct vertices of $\overline{P}$ by $2n + 3k$ without affecting the geometric error bound. In addition, experimental results obtained with an implementation of the algorithm using the C++ library CGAL [13] indicate that the number of additional vertices of $\overline{P}$ is very small in practice.

From the above lemmas, we are now able to prove Theorem 3. The proof of property 1) and 2) can be directly deduced by construction of $\overline{P}$ from Theorem 1. Property 3) is proved in Lemma 13. Property 4) is proved in Lemma 14.

7 Rounding Set Operations

Theorems 15 and 16 enumerate the set of properties satisfied when the exact region U comes from an union operation i.e. when $U = A \cup B$. These properties can be directly obtained from Theorems 1 and 3 by replacing A and B by their complementary sets.

Theorem 15 *The outer rounding $\overline{U}$ of $U = A \cap B$ satisfies the following properties:*
1) $\overline{U}$ is a numerically consistent lattice polygonal region,
2) $\overline{U}$ contains U,
3) $d_H(\overline{U}, U) < \sqrt{2}$,
4) $|\overline{U}| \leq |U|$,
5) A convex vertex of $\overline{U}$ does always correspond to a convex vertex of U.

Theorem 16 *The inner rounding $\underline{U}$ of $U = A \cup B$ satisfies the following properties:*
1) $\underline{U}$ is a numerically consistent lattice polygonal region,
2) $\underline{U}$ is contained in U,
3) $d_H(\underline{U}^C, (U^o)^C) < \sqrt{2}$,
4) $|\underline{U}| \leq |U| + k + h$,where k is the number of non-lattice vertices of U and h is the total number of intersecting pairs between the edges of U and those of I.

The result for the set difference operation can equally be deduced from Theorems 1, 2, 5 and 6 for each rounding mode.

8 Rounding General Regions

From the lemmas and algorithms presented so far in this report, the inner/outer rounding of a general polygonal region (for which we do not have any assumption on the representation of its vertices, for example a region issued from a rotation operation) can be obtained in the following manner. Let P be a general polygonal region and consider V_c (resp. V_r), the set of its convex (resp. reflex) vertices that do not lie at lattice sites. Let now I_c (resp. I_r) be the set of unit lattice squares that contain the vertices of V_c (resp. V_r), i.e. the set of quadrilaterals having respectively $(\lfloor v_x \rfloor, \lfloor v_y \rfloor)$ and $(\lceil v_x \rceil, \lceil v_y \rceil)$ as bottom left and top right vertex where $v = (v_x, v_y)$ is a vertex of I_c (resp. I_r).

We define the inner rounding $\underline{P}$ of P as the result of the rounding with the inner mode of the intersection of P and I_r, that is $\underline{P} = P \underline{\cap} I_r$. Similarly, we define the outer rounding $\overline{P}$ of P as the complementary of the rounding with the inner mode of the intersection of P^C and ${I_c}^C$, that is $\overline{P} = (P^C \underline{\cap} {I_c}^C)^C$.

The inner rounded intersection operations used in these definitions can be directly computed from the algorithm presented in Section 5. We remark indeed that all reflex vertices of the intersection regions $(P \underline{\cap} I_r)$ and $(P^C \underline{\cap} {I_c}^C)^C$ lie at lattice sites which is a sufficient condition to satisfy the properties of the reflex vertical decomposition stated in Lemmas 4 and 5 and thus to prove the correctness of the algorithms.

We notice however that the absence of lattice segments supporting the edges of the input region P requires the use of a well suited number type and arithmetic in order to evaluate the numerical primitives that appear in the algorithm. A number type and an arithmetic allowing the manipulation of algebraic numbers can be necessary for example if the input region P are issued from a rotation operation.

The properties satisfied by the rounded regions $\underline{P}$ and $\overline{P}$ can be directly derived from the lemmas of the previous sections. More precisely, the inner rounding $\underline{P}$ is a numerically consistent lattice polygonal region contained in P such that $d_H(\underline{P}^C, P^{oC}) < \sqrt{2}$. Moreover, if it exists, the rounded counterpart of a convex vertex of P is a convex vertex of $\underline{P}$. Finally, $\underline{P}$ has less than $|P| + r + h_r$ distinct vertices where r is the number of reflex vertices of P and h_r is the number of intersections between the edges of P and the edges of I_r. The outer rounding $\overline{P}$ of P is a numerically consistent lattice polygonal region that contains P such

that $d_H(\overline{P}, P) < \sqrt{2}$ and $|\overline{P}| < |P| + c + h_c$ where c denotes the number of convex vertices of P and h_c denotes the number of intersections between the edges of P and the edges of I_c.

We finally remark that the number of vertices needed to represent $\overline{P}$ can be reduced with the same kind of technique as described at the end of Section 6.2.

9 Conclusion

We have given methods for computing the inner/outer rounding of the result of set operations on two lattice polygonal regions in the plane. The guarantees that the exact result of such operations contains (or is contained in) its finite precision approximation allows to introduce the geometric analogue of interval arithmetic provided by the certified rounding modes of the IEEE 754 norm for floating point arithmetic operations. The computation of such geometric intervals with respect to the inclusion relation permits in particular to cascade various geometric constructions as set operations, convex hulls or rotations with a control on their bit complexity. This result is a first step towards the definition of a complete system for performing rounded operations on polygonal and polyhedral objects which would be of great practical interest in many CAO applications.

References

[1] S. Fortune. Vertex-rounding a three-dimensional polyhedral subdivision. *Discrete Comput. Geom.*, 22(4):593–618, 1999.

[2] P. Guigue. *Constructions géométriques à précision fixée.* Thèse de doctorat en sciences, université de Nice-Sophia Antipolis, France, 2003.

[3] S. Funke and K. Mehlhorn. Look: A lazy object-oriented kernel for geometric computation. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, pages 156–165, 2000.

[4] J. E. Goodman, R. Pollack, and B. Sturmfels. Coordinate representation of order types requires exponential storage. In *Proc. 21st Annu. ACM Sympos. Theory Comput.*, pages 405–410, 1989.

[5] 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.

[6] D. H. Greene and F. F. Yao. Finite-resolution computational geometry. In *Proc. 27th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 143–152, 1986.

[7] L. Guibas and D. Marimont. Rounding arrangements dynamically. *Internat. J. Comput. Geom. Appl.*, 8:157–176, 1998.

[8] J. D. Hobby. Practical segment intersection with finite precision output. *Comput. Geom. Theory Appl.*, 13(4):199–214, October 1999.

[9] H. S. Lee and R. C. Chang. Approximating vertices of a convex polygon with grid points in the polygon. In *Proc. 3rd Annu. Internat. Sympos. Algorithms Comput.*, volume 650 of *Lecture Notes Comput. Sci.*, pages 269–278. Springer-Verlag, 1992.

[10] V. J. Milenkovic. Shortest path geometric rounding. *Algorithmica*, 27(1):57–86, 2000.

[11] V. J. Milenkovic. Practical methods for set operations on polygons using exact arithmetic. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 55–60, 1995.

[12] C. K. Yap and T. Dubé. The exact computation paradigm. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, volume 4 of *Lecture Notes Series on Computing*, pages 452–492. World Scientific, Singapore, 2nd edition, 1995.

[13] *The CGAL Manual*, 2003. Release 3.0.

Contents

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

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 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 - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

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