Deconstructing Approximate Offsets

Eric Berberich
Max-Planck-Institut für Informatik
Saarbrücken, Germany
eric@mpi-inf.mpg.de

Dan Halperin
Tel Aviv University
Tel Aviv, Israel
danha@post.tau.ac.il

Michael Kerber
Institute of Science and Technology (IST) Austria
Klosterneuburg, Austria
mkerber@ist.ac.at

Roza Pogalnikova
Tel Aviv University
Tel Aviv, Israel
rozapoga@post.tau.ac.il

ABSTRACT

We consider the *offset-deconstruction problem*: Given a polygonal shape Q with n vertices, can it be expressed, up to a tolerance ε in Hausdorff distance, as the Minkowski sum of another polygonal shape P with a disk of fixed radius? If it does, we also seek a preferably simple-looking solution shape P; then, P's offset constitutes an accurate, vertex-reduced, and smoothened approximation of Q. We give an $O(n \log n)$-time exact decision algorithm that handles any polygonal shape, assuming the real-RAM model of computation. An alternative algorithm, based purely on rational arithmetic, answers the same deconstruction problem, up to an uncertainty parameter δ, and its running time depends on the parameter δ (in addition to the other input parameters: n, ε and the radius of the disk). If the input shape is found to be approximable, the rational-arithmetic algorithm also computes an approximate solution shape for the problem. For convex shapes, the complexity of the exact decision algorithm drops to $O(n)$, which is also the time required to compute a solution shape P with at most one more vertex than a vertex-minimal one. Our study is motivated by applications from two different domains. However, since the offset operation has numerous uses, we anticipate that the reverse question that we study here will be still more broadly applicable. We present results obtained with our implementation of the rational-arithmetic algorithm.

Categories and Subject Descriptors:
I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling
F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems - *Geometrical problems and computations*

General Terms: Algorithms, Theory

Keywords: offsets, Minkowski sums, polygonal smoothing, deconstruction

1. INTRODUCTION

The r-*offset* of a polygon, for a real parameter $r > 0$, is the set of points at distance at most r away from the polygon. Computing the offset of a polygon is a fundamental operation. The offset operation is, for instance, used to define a tolerance zone around the given polygon [1] or to dilute details for clarity of graphic exposition [2, 3, 4]. Technically, it is usually computed as the *Minkowski sum* of the polygon and a disk of a certain radius. The resulting shape is bounded by straight-line segments and circular arcs. However, a customary practice is to model the disk in the Minkowski sum with a (tight) polygon, which yields a piecewise-linear approximation of the offset. Our study is motivated by two applications, where such an approximation forms the legacy data which a program has to deal with—the original shape before offsetting is unknown. This leads us to the question what is the original polygon whose approximate offset we have at hand. Of course, finding the exact original polygon, or even its topology, is impossible in general, because the offset might have blurred small features like holes or dents. However, a reasonable choice can lead to a more compact and smooth representation of the approximate offset.

The first relevant problem concerns cutting polygonal parts out of wood. A wood-cutting machine, which can smoothly cut along straight line segments and circular arcs, is given a plan to cut out a certain shape. This shape was designed as a polygon expanded by a small offset, but with circular arcs approximated by polygonal lines comprising many tiny line segments. Thus instead of moving smoothly along circular arcs, the cutting tool has to move along a sequence of very short segments, and make a small turn between every pair of segments. The process becomes very slow, the tool heats up, and occasionally it causes the wood to burn. Moving the cutting tool smoothly and fast enough is the way to keep it cool. If this were the only issue, other smoothing techniques like *arc-spline approximation* [5, 6] may have been applicable. However, we may also wish to reduce the offset radius if a more accurate cutting machine is available—in this case, it seems desirable to find the original shape first and then to re-offset with a smaller radius.

A motivation to study this question from a different domain is to recover shapes sketched by a user of a digital pen and tablet. The pen has a relatively wide tip, and the input obtained is in fact an approximate offset (with the radius of the pen tip) of the intended shape. The goal is to give a good polygonal approximation of the intended shape. More broadly, as the offset operation is so commonplace, it seems natural to ask, given only an (approximated) offset shape, what could be the original shape before the offsetting. Therefore, we pose

Problem 1: the (offset-)deconstruction problem or **decision problem** Given a polygonal shape Q, and two real parameters $r, \varepsilon > 0$, decide if Q is within (symmetric) Hausdorff-distance ε to the r-offset (i.e., offset with radius r) of some other (yet unknown) polygonal shape P.

Problem 2: finding a valid source If the answer to Problem 1 is YES, compute a valid polygonal shape P. We refer to P as a *valid source* of the approximate offset Q. Note that P might be disconnected, even if Q is connected (Figure 1.1).

Problem 1 can be seen as a special case of the *Minkowski decomposition problem* which asks whether a set can be composed in a non-trivial way as the Minkowski sum of two sets—disallowing a summand to be a homothetic copy of the input set. A general criterion for decomposability of convex sets in arbitrary dimension has been presented in [7]. A particularly well-studied case are planar lattice polygons, because of their close relation to problems in algebra, for instance, polynomial factorization [8]. It has been shown that deciding decomposability is NP-complete for lattice polygons [9]. In [10], decomposability is investigated under the constraint that one of the summands is a line segment, a triangle, or a quadrangle. However, all these approaches discuss the exact decomposition problem; our scenario of being Hausdorff-close to a particular decomposition seems to not have been addressed in the literature. Allowing tolerance raises interesting algorithmic questions and at the same time makes the tools that we develop more readily suitable for applications, which typically have to deal with inaccuracies in measuring and modeling.

Contributions. We first present an efficient algorithm to decide Problem 1: For a shape Q with n vertices, the algorithm reports the correct answer in $O(n \log n)$ time in the real-RAM model of computation [11]. It constructs offsets with increasing radii in three stages; the intermediate shapes arising during the computation are in general more difficult to offset than polygons, as they are bounded by straight-line segments and "indented" circular arcs (namely, the shape is locally on the concave side of the arcs). The main observation is that for certain classes of such shapes, these circular arcs can be ignored when computing the next offset (see Theorem 5 for the precise statement). This observation bounds the time required by each offset computation by $O(n \log n)$, which is the key to the efficiency of the decision algorithm. Our proof is constructive, that is, if a valid source (Problem 2) exists, a solution can be computed with the same running time.

The computation of the exact decision procedure requires the handling of algebraic coordinates of considerably high degree. As alternative, we give an approximation scheme that works exclusively with rational numbers instead. The scheme proceeds by replacing the offset disks by polygonal shapes of similar diameter, whose precision is determined by another parameter $\delta < \varepsilon$. We prove a bound Δ that depends on $\hat{\varepsilon}$, the minimal ε for which the answer to the decision problem is YES, such that the rational approximation returns the exact result for all $\delta \leq \Delta$. If the input shape is found to be approximable, this algorithm also outputs a valid source. The computation of $\hat{\varepsilon}$ up to any desired precision is still possible. We believe that our investigation of the relation between δ and $\hat{\varepsilon}$ is of independent relevance, mostly to the study of certified algorithms that approximate geometric objects with algebraic coordinates by means of rational arithmetic.

For a convex shape Q with n vertices, we reduce the running time for solving Problem 1 to the optimal $O(n)$ (in the real-RAM model). Moreover, we describe a greedy algorithm within the same time complexity that returns a valid source $P^\star$ which minimizes, up to one extra vertex, the number of vertices among all valid choices, if there are any. Our algorithm technically resembles an approach for the different problem of finding a vertex-minimal polygon in the annulus of two nested polygons [12]. We also remark that the r-offset of $P^\star$ has a tangent-continuous boundary and therefore constitutes a special case of an arc-spline approximation of Q where all circular arcs have the same radius.

Organization. We describe an exact decision algorithm for the deconstruction problem (solving Problem 1 above) in Section 2. In Section 3 we describe a rational-approximation algorithm for the deconstruction problem. Both algorithms output a valid source in case the input is deconstructible (solving Problem 2). For convex input, Section 4 exposes a specialized deconstruction and the computation of an almost vertex-minimal valid source. Details on our implementation and some illustrative cases are depicted in Section 5. We conclude in Section 6 by pointing out open problems.[1]

2. THE DECISION ALGORITHM

For a set $X \subset \mathbb{R}^2$ denote its boundary by ∂X and its complement by $X^C := \mathbb{R}^2 \setminus X$. A *polygonal region* or *polygonal shape* $X \subset \mathbb{R}^2$ is a set whose boundary consists of finitely many line segments with disjoint interiors. The endpoints of these straight-line segments are the *vertices* of the polygonal region. We assume henceforth that the input shapes that we deal with are bounded (but not necessarily connected). Although the techniques seem to go through also for unbounded shapes, this assumption simplifies the exposition and is sufficient for the real-life applications we have in mind. If X is a bounded polygonal region, ∂X is the union of (weakly) simple polygons. For two sets X and Y, we denote their *Minkowski sum* by $X \oplus Y := \{x + y \mid x \in X, y \in Y\}$. With $d(\cdot,\cdot)$ the Euclidean distance function, and any $c \in \mathbb{R}^2, r \in \mathbb{R}$, we write $D_r(c) := \{p \in \mathbb{R}^2 \mid d(c,p) \leq r\}$ for the (closed) r-disk around c, and $D_r := D_r(O)$ for the disk centered at the origin. The *r-offset* of a set X, offset(X, r), is the Minkowski sum $X \oplus D_r$.

For $p \in \mathbb{R}^2$ and X a closed set, we write $d(p, X) := \min\{d(p,x) \mid x \in X\}$. The *(symmetric) Hausdorff distance* of two closed point sets X and Y is

$$H(X,Y) := \max\{\max\{d(x,Y) \mid x \in X\}, \max\{d(y,X) \mid y \in Y\}\}.$$

We say that X is ε-close to Y (and Y to X) if $H(X,Y) \leq \varepsilon$, which can also be expressed alternatively:

[1] A preliminary and abridged version of this paper, excluding the results of Sections 3 and 5 in particular, appeared in Abstracts of the 26th European Workshop on Computational Geometry [13].

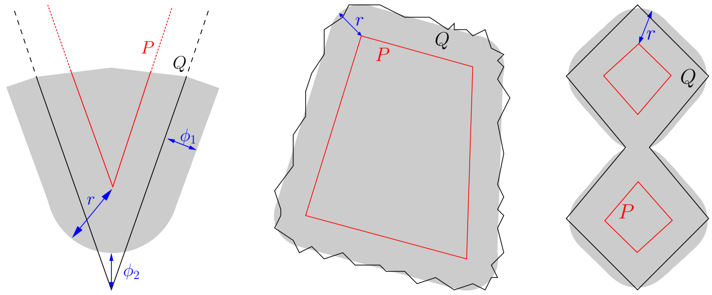


Figure 1.1: For a given Q, the red P is a candidate summand whose exact r-offset is shaded. Left: For a given ε, deconstruction is ensured iff $\phi_1 \leq \varepsilon$ and $\phi_2 \leq \varepsilon$. Note that, when r decreases, ϕ_1 decreases, but ϕ_2 increases. Middle: Example where Q can be approximated by an r-offset of a P that has much fewer vertices than Q. Right: Example where Q can be approximated by the r-offset of a disconnected shape P.

Proposition 1. *For X, Y closed, X is ε-close to Y if and only if $Y \subseteq \operatorname{offset}(X, \varepsilon)$ and $X \subseteq \operatorname{offset}(Y, \varepsilon)$.*

Decision algorithm. We fix $r > 0$, $\varepsilon > 0$, and a polygonal region Q, and consider the following question: Is there a polygonal region P such that Q and the r-offset of P have Hausdorff-distance at most ε? First of all, we can assume that $r > \varepsilon$; otherwise, we can choose $P := Q$, because $\operatorname{offset}(Q, r)$ and Q have Hausdorff-distance at most ε. We define another operation, *r-inset* (a.k.a. "erosion"), which is computationally similar to an offset:

Definition 2. *For $r > 0$, and $X \subset \mathbb{R}^2$, the* r-inset *of X is the set* $\operatorname{inset}(X, r) := \operatorname{offset}(X^C, r)^C = \{x \in \mathbb{R}^2 \mid D_r(x) \subseteq X\}$.

We are now ready to present the decision algorithm:

Algorithm 1 Decide(Q, r, ε)

(1) $Q_\varepsilon \leftarrow \operatorname{offset}(Q, \varepsilon)$
(2) $\Pi \leftarrow \operatorname{inset}(Q_\varepsilon, r)$
(3) $Q' \leftarrow \operatorname{offset}(\Pi, r + \varepsilon)$
(4) return $Q \subseteq Q'$

We next prove that Decide (Algorithm 1) correctly decides whether Q is ε-close to some r-offset of a polygonal region. A first observation is that for any polygonal region P, $\operatorname{offset}(P, r) \subseteq Q_\varepsilon$ if and only if $P \subseteq \Pi$. This is an immediate consequence of the definition of the inset operation. This shows that for any $\operatorname{offset}(P, r)$ that is ε-close to Q, P must be inside Π. Moreover, it shows that any choice of $P \subseteq \Pi$ already satisfies one of Proposition 1's inclusions. It is only left to check whether $Q \subseteq \operatorname{offset}(\operatorname{offset}(P, r), \varepsilon) = \operatorname{offset}(P, r+\varepsilon)$. We summarize:

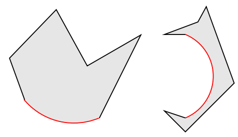
Proposition 3. *Q is ε-close to $\operatorname{offset}(P, r)$ if and only if $P \subseteq \Pi$ and $Q \subseteq \operatorname{offset}(P, r + \varepsilon)$.*

To prove correctness of Decide, we have to show that $Q \subseteq \operatorname{offset}(\Pi, r + \varepsilon)$ already implies that there also exists a polygonal region $P \subseteq \Pi$ with $Q \subseteq \operatorname{offset}(P, r + \varepsilon)$. Indeed, Π is not polygonal in general; we have to study its shape closer to prove that we can approximate it by a polygonal region, maintaining the property that the offset remains ε-close to Q.

The shape of offsets and insets. For a polygonal region Q, it is not hard to figure out the shape of $Q_\varepsilon = \operatorname{offset}(Q, \varepsilon)$: It is a closed set bounded by straight-line segments and by circular arcs, belonging to a circle of radius ε. It is important to remark that all circular arcs are *bulges*:

Definition 4. *Let $X \subset \mathbb{R}^2$ be a closed set with some circular arc γ on its boundary. Then, γ is called a* dent *with respect to X, if each line segment connecting two distinct points on γ is not fully contained in X. Otherwise, the arc is called a* bulge.

We call X a bulged *(resp. an* indented*) region with radius r, if ∂X consists of finitely many straight-line segments and bulges (resp. dents) that are all of radius r, interlinked at the* vertices *of the region.*



Note that a bulged region (left) is not necessarily convex. The r-offset of a polygonal region P is a bulged region with radius r. The heart of this section is Theorem 5 showing that the same also holds if P is an indented region (right) with radius smaller than r:

Theorem 5. *Let P be an indented region with radius r_1, and let $r_2 > r_1$. Then, there is a polygonal region $P_L \subseteq P$ such that $\operatorname{offset}(P, r_2) = \operatorname{offset}(P_L, r_2)$. In particular, $\operatorname{offset}(P, r_2)$ is a bulged region with radius r_2.*

Proof. After possibly splitting circular arcs into at most four parts, we can assume that each circular arc spans at most a quarter of the circle. For such a circular arc, we define its endpoints by x_1 and x_2, and denote the *linear cap* of the circular arc as the (closed) indented region enclosed by the circular arc, and the two lines tangent to the circle through x_1 and x_2 (the shaded area in Figure 2.1a). The *extended linear cap* is the (polygonal) region spanned by the two tangents just mentioned, and the two corresponding normals at x_1 and x_2. Clearly, the normals meet in the center of the circle that defines the arc.

We iteratively replace a indented arc of an indented region P' with radius r_1 (initially set to P) by a polyline ℓ ending in the endpoints of the circular arc, such that ℓ does neither leave P' nor the linear cap of the circular arc, and such that other boundary parts of P' are not intersected. This yields another indented region P'' with radius r_1, where one indented arc is replaced by a polyline, as depicted in Figure 2.1a. Iterating this construction, starting with P, until all indented arcs are replaced, we obtain a polygonal region P_L.

We show that in each iteration, the r_2-offsets of P' and P'' are the same. For that we consider any point $x' \in P' \setminus P''$, in the region that is cut off by P'', and consider $y = x' + v'$ for an arbitrary $v' \in D_{r_2}$. We show that in all cases, y can also be written by $y = x'' + v''$, with $x'' \in P''$, and $v'' \in D_{r_2}$.

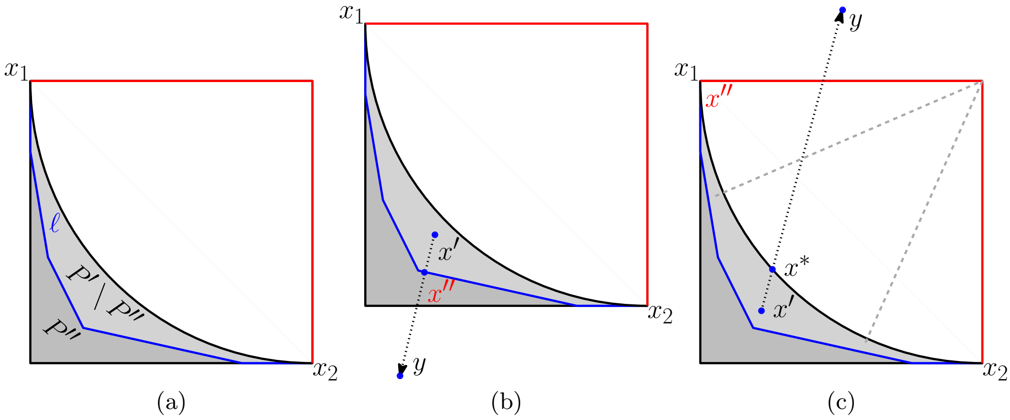


Figure 2.1: (a) The (extended) linear cap split by the polyline ℓ (b-c) the two cases in the proof of Theorem 5.

Since the circular arc spans at most a quarter of the circle, it is easily seen that $D_{r_1}(x_1) \cup D_{r_1}(x_2)$ covers the whole extended linear cap. Therefore, for any y that lies within the extended linear cap, selecting $x'' = x_1$ or $x'' = x_2$, we get $y = x'' + v''$ with $v'' \in D_{r_1}$.

We distinguish two other cases: for y that lies outside of the extended linear cap $v' = \overline{x'y}$ crosses either ℓ or the circular arc. In the former case, we can simply pick the crossing point as x'', and set $v'' \in D_{r_2}$ accordingly (Figure 2.1b). In the latter case, let's denote the crossing point as x^* (Figure 2.1c). We consider the set of points that is closer to x^* than to x_1 and x_2. Clearly, that region is bounded by the two corresponding bisectors, which meet in the center of the circle that defines the circular arc and is therefore completely contained within the extended linear cap. It follows that y is closer to one of the endpoints of the arc, say x_1, than to x^*. Selecting $x'' = x_1$ we ensure y is closer to x'' than to x', which proves that $y = x'' + v''$ with some $v'' \in D_{r_2}$ in this case as well. □

The proof of Theorem 5 implies that offset(P, r_2) for such a region P is completely determined by the offset of its linear segments, and the offset of the endpoints of circular arcs: the interior of the indented circular arcs can be ignored.

Corollary 6. Decide *returns* YES *if and only if there exists a polygonal region P such that* offset(P, r) *is ε-close to Q.*

Proof. Q_ε is a bulged region with radius ε. Therefore, Q_ε^C is an indented region with the same radius. Since $r > \varepsilon$, Theorem 5 implies that offset(Q_ε^C, r) is a bulged region with radius r, and so, offset$(Q_\varepsilon^C, r)^C$ = inset$(Q_\varepsilon, r) = \Pi$ is an indented region with the same radius. Using $r + \varepsilon > r$ and applying Theorem 5 once more, there exists a polygonal region $P \subseteq \Pi$ such that offset$(\Pi, r+\varepsilon)$ = offset$(P, r+\varepsilon)$. It follows that, if the algorithm returns YES, there is indeed a polygonal region P whose r-offset is ε-close to Q. If the algorithm return NO, it is clear that no such polygonal region can exist. □

Theorem 7. *Let P be an indented region with radius r_1 having n vertices, and assume $r_2 > r_1$. Then,* offset(P, r_2) *has $O(n)$ vertices and it can be computed in $O(n \log n)$ time.*

Proof. By Theorem 5, it suffices to consider a polygonally bounded P_L instead of P. We use trapezoidal decomposition of P to construct such a P_L with only $O(n)$ vertices. The Voronoi diagram of P_L's vertices and (open) edges can be computed in $O(n \log n)$ time and has size $O(n)$ [14]. From it, the offset polygon with the same asymptotic complexity can be obtained in linear time [15]. □

Corollary 8. Decide *decides ε-closeness with $O(n \log n)$ operations.*

Proof. Apply Theorem 7 in each step of Algorithm 1. The fourth step runs in $O(n \log n)$ time as well using a simple sweep-line algorithm. □

Note that Π_L, if constructed for Π as in the proof of Theorem 5 during step (3) of Algorithm 1, is a valid source for Q if Decide returns YES.

3. RATIONAL APPROXIMATION

A direct realization of Algorithm 1 runs into difficulties since vertices of the resulting offsets are algebraic numbers (and their degree increases for cascaded offset computation). We next describe two approximation variants of Algorithm 1, each producing a certified one-sided decision by approximating all disks in the algorithm with k-gons. In order to make guaranteed statements about the exact ε-approximability by r-offsets, we have to approximate the disks by a "working precision" δ which is even smaller than ε. Recall that D_r is the disk of radius r centered at the origin. For $a, b \in \mathbb{R}$, $a < b$ define $\bar{D}_{a,b}$ to be a polygon with rational vertices whose boundary lies in the annulus $D_b \setminus D_a$. In the approximation algorithms, every disk is replaced with such a polygon lying inside a δ-width annulus.

Interior approximation. In the first part of our algorithm, we ensure that the final approximation of Q' (see line (3) of Algorithm 1), called $\widehat{Q'}$, will be a subset of the exact Q'. We achieve this by approximating D_s by $\bar{D}_{s-\delta,s}$ when an offset is computed; and by approximating D_s by $\bar{D}_{s,s+\delta}$ when an inset is computed; see Algorithm 2.

Lemma 9. *If* ApproxDecideInterior $(Q, r, \varepsilon, \delta)$ *returns* YES*, then* Decide(Q, r, ε) *returns* YES *as well, which means that there exists a polygonal region P such that* offset(P, r) *is ε-close to Q. In particular, $P := \widehat{\Pi}$ is a valid source.*

Proof. Compare the execution of Algorithm 2 with the corresponding call of its exact version, Algorithm 1. It is straight-forward to check that for any δ, $\widehat{Q_\varepsilon} \subset Q_\varepsilon$, $\widehat{\Pi} \subset \Pi$,

Algorithm 2 ApproxDecideInterior $(Q, r, \varepsilon, \delta)$

(1) $\widehat{Q_\varepsilon} \leftarrow Q \oplus \widehat{D_\varepsilon}$ with $\widehat{D_\varepsilon} \leftarrow \bar{D}_{\varepsilon-\delta,\varepsilon}$
(2) $\widehat{\Pi} \leftarrow \left(\widehat{Q_\varepsilon}^C \oplus \widehat{D_r}\right)^C$ with $\widehat{D_r} \leftarrow \bar{D}_{r,r+\delta}$
(3) $\widehat{Q'} \leftarrow \widehat{\Pi} \oplus \widehat{D_{r+\varepsilon}}$ with $\widehat{D_{r+\varepsilon}} \leftarrow \bar{D}_{r+\varepsilon-\delta,r+\varepsilon}$
(4) if $Q \subseteq \widehat{Q'}$, return YES,
otherwise, return UNDECIDED

and $\widehat{Q'} \subset Q'$. The last inclusion shows that if $Q \subseteq \widehat{Q'}$, also $Q \subseteq Q'$. $\square$

Definition 10. *For fixed Q and r, define $\hat{\varepsilon} := \inf\{\varepsilon \mid$ Decide(Q, r, ε) returns* YES$\}$.

Note that $\hat{\varepsilon} \in [0, r]$, and that Decide$(Q, r, \varepsilon)$ returns YES for every $\varepsilon \geq \hat{\varepsilon}$ and returns NO for every $\varepsilon < \hat{\varepsilon}$. We do not have a way to compute $\hat{\varepsilon}$ exactly. However, we show next that ApproxDecideInterior $(Q, r, \varepsilon, \delta)$ returns YES for every $\varepsilon > \hat{\varepsilon}$ for δ small enough, and that the required precision δ is proportional to the distance of ε to $\hat{\varepsilon}$.

Theorem 11. *Let $\varepsilon > \hat{\varepsilon}$, and $\delta < \frac{\varepsilon-\hat{\varepsilon}}{2}$. Then,* ApproxDecideInterior $(Q, r, \varepsilon, \delta)$ *returns* YES.

Proof. Let ε_0 be such that $\hat{\varepsilon} < \varepsilon_0 < \varepsilon_0 + 2\delta \leq \varepsilon$. Let Q_{ε_0}, Π and Q' denote the intermediate results of Decide (Q, r, ε_0) and let $\widehat{Q_\varepsilon}$, $\widehat{\Pi}$, $\widehat{Q'}$ denote the intermediate results of ApproxDecideInterior $(Q, r, \varepsilon, \delta)$. By the choice of ε_0, YES is returned, and thus $Q \subseteq Q'$. The theorem follows from $Q' \subseteq \widehat{Q'}$, which we prove in three substeps:

(1) offset$(Q_{\varepsilon_0}, \delta) \subseteq \widehat{Q_\varepsilon}$: Indeed, offset$(Q_{\varepsilon_0}, \delta) =$ offset$(Q, \varepsilon_0 + \delta) \subseteq$ offset$(Q, \varepsilon - \delta) \subset Q \oplus \widehat{D_\varepsilon} = \widehat{Q_\varepsilon}$.

(2) $\Pi \subseteq \widehat{\Pi}$: Starting with (1), we obtain

$$\begin{aligned} & \text{offset}(Q_{\varepsilon_0}, \delta) \subseteq \widehat{Q_\varepsilon} \\ \Rightarrow\ & \text{offset}(Q_{\varepsilon_0}, \delta)^C \oplus D_{r+\delta} \supseteq \widehat{Q_\varepsilon}^C \oplus \widehat{D_r} \\ \Rightarrow\ & \text{inset}(\text{offset}(Q_{\varepsilon_0}, \delta), r + \delta) \subseteq \widehat{\Pi}\,. \end{aligned}$$

On the left-hand side, we use that inset(offset$(A, r), r) \supseteq A$ to obtain:

$$\begin{aligned} & \text{inset}(\text{offset}(Q_{\varepsilon_0}, \delta), r + \delta) \\ =\ & \text{inset}(\text{inset}(\text{offset}(Q_{\varepsilon_0}, \delta), \delta), r) \\ \supseteq\ & \text{inset}(Q_{\varepsilon_0}, r) = \Pi. \end{aligned}$$

(3) $Q' \subseteq \widehat{Q'}$: Using (2), we have that

$$\text{offset}(\Pi, r + \varepsilon - \delta) = \Pi \oplus D_{r+\varepsilon-\delta} \subseteq \widehat{\Pi} \oplus \widehat{D_{r+\varepsilon}} = \widehat{Q'}.$$

Note that $r + \varepsilon - \delta > r + \varepsilon_0$, and therefore, offset$(\Pi, r + \varepsilon - \delta) \supset$ offset$(\Pi, r + \varepsilon_0) = Q'$. $\square$

Exterior approximation. In Algorithm 3, we ensure that $\widehat{Q'}$ becomes a superset of the exact Q' by appropriately choosing approximate disks. Specifically, we approximate D_s by $\bar{D}_{s,s+\delta}$ when an offset is computed, and D_s by $\bar{D}_{s-\delta,s}$ when an inset is computed. Not surprisingly, we get a certified answer in the other direction, and a certified answer is guaranteed when δ is sufficiently small. The proofs of the following two statements are similar to Lemma 9 and Theorem 11 and thus omitted.

Algorithm 3 ApproxDecideExterior $(Q, r, \varepsilon, \delta)$

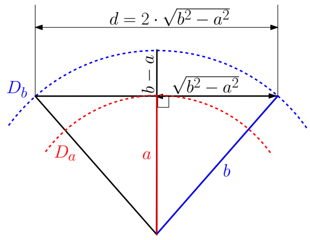
(1) $\widehat{Q_\varepsilon} \leftarrow Q \oplus \widehat{D_\varepsilon}$ with $\widehat{D_\varepsilon} \leftarrow \bar{D}_{\varepsilon,\varepsilon+\delta}$
(2) $\widehat{\Pi} \leftarrow \left(\widehat{Q_\varepsilon}^C \oplus \widehat{D_r}\right)^C$ with $\widehat{D_r} \leftarrow \bar{D}_{r-\delta,r}$
(3) $\widehat{Q'} \leftarrow \widehat{\Pi} \oplus \widehat{D_{r+\varepsilon}}$ with $\widehat{D_{r+\varepsilon}} \leftarrow \bar{D}_{r+\varepsilon,r+\varepsilon+\delta}$
(4) if $Q \subseteq \widehat{Q'}$, return UNDECIDED,
otherwise, return NO

Lemma 12. *If* ApproxDecideExterior $(Q, r, \varepsilon, \delta)$ *returns* NO, *then* Decide(Q, r, ε) *returns* NO *as well, which means that there exists no polygonal region P such that* offset(P, r) *is ε-close to Q.*

Theorem 13. *Let $\varepsilon < \hat{\varepsilon}$ and $\delta < \frac{\hat{\varepsilon}-\varepsilon}{2}$. Then,* ApproxDecideInterior $(Q, r, \varepsilon, \delta)$ *returns* NO.

In combination with Theorem 11, it follows that the exact answer can always be found for $\delta < \Delta := \frac{|\varepsilon-\hat{\varepsilon}|}{2}$ by combining ApproxDecideExterior or ApproxDecideInterior.

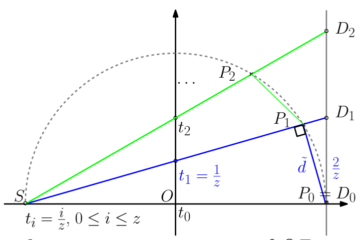
Complexity analysis. The main task is to bound the number of vertices of $\bar{D}_{a,b}$. We will create a $\bar{D}_{a,b}$ with the additional property that all vertices lie on ∂D_b. As depicted on the right, two such points on D_b are connected by a chord of the boundary circle that does not intersect D_a if and only if the angle induced by the two points is at most $\psi := 2 \arccos \frac{a}{b}$, or equivalently, the length of the chord is less than $2\sqrt{b^2 - a^2}$. Note that we need at least $\frac{2\pi}{\psi}$ points on ∂D_b for a valid $\bar{D}_{a,b}$, and $\frac{2\pi}{\psi} \in \Theta(\sqrt{\frac{b}{b-a}})$ as easily shown by L'Hopital's rule.



Rational points on ∂D_b can be constructed for an arbitrary $t \in \mathbb{Q}$ as $Q_t := (b\frac{1-t^2}{1+t^2}, b\frac{2t}{1+t^2})$ [16]. For some positive $z \in \mathbb{Z}$, we define $P_i := Q_{i/z}$ for $i = 0, \ldots, z$.

Lemma 14. *For every $i = 1, \ldots, z-1$, the chord $P_{i-1}P_i$ is longer than the chord P_iP_{i+1}. In particular, the length of each chord is bounded by the length of P_0P_1 which is shorter than $\frac{2b}{z}$.*

Proof. W.l.o.g., we assume $b = 1$ for the proof, since the chord length scales proportionally when scaling the circle by a factor of b. The point $Q_t = (\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2})$ can be constructed geometrically as the intersection point of ∂D_b with the line ℓ_t through $S = (-1, 0)$ and slope t (see the figure enclosed in this paragraph). In particular, the line SP_i has slope $\frac{i}{z}$; we let D_i denote the intersection point of that line with the line $x = 1$. We observe that the segment D_iD_{i+1} has length $\frac{2}{z}$, and that $SD_i < SD_{i+1}$ for $i = 0, \ldots, z-1$.



We are showing next that the chord $P_{i-1}P_i$ is longer than P_iP_{i+1}. For that, we consider the triangle $SD_{i-1}D_{i+1}$, and its bisector at S. This bisector intersects the line $x = 1$ at some point B. By the Angle bisector theorem, B divides the segment $D_{i-1}D_{i+1}$ proportionally to the corresponding triangle sides, that is, $\frac{SD_{i-1}}{SD_{i+1}} = \frac{BD_{i-1}}{BD_{i+1}}$. Because the left-hand side is smaller than 1, it follows that BD_{i-1} is shorter than BD_{i+1}. Therefore, B lies below D_i, and therefore, the angle $\alpha_{i-1} = \angle D_{i-1}SD_i = \angle P_{i-1}SP_i$ is larger than $\alpha_i = \angle D_iSD_{i+1} = \angle P_iSP_{i+1}$. But the chord lengths $P_{i-1}P_i$ and P_iP_{i+1} are defined by $2\sin(\alpha_{i-1})$ and $2\sin(\alpha_i)$, respectively, which proves that the chord lengths are indeed decreasing.

Finally, by Thales' theorem, the triangle SP_0P_1 has a right angle at P_1. Therefore, the longest chord P_0P_1 is shorter than the segment D_0D_1, which has length $\frac{2}{z}$. □

Note that all P_i's lie in the first quadrant of the plane and that $P_0 := (b, 0)$ and $P_z := (0, b)$. Therefore, we can subdivide the other three quarters of the circle symmetrically such that the length of each chord is bounded by $\frac{2b}{z}$, using $4z$ vertices altogether. To compute a valid $\bar{D}_{a,b}$, it suffices to choose a z such that $\frac{2b}{z} \leq 2\sqrt{b^2 - a^2}$, that is $z \geq \sqrt{\frac{b^2}{b^2-a^2}}$. We choose $z_0 := \left\lceil \sqrt{\frac{b}{b-a}} \right\rceil$, indeed, since $0 < a < b$, we have that $z_0 \geq \sqrt{\frac{b}{b-a}} > \sqrt{\frac{b}{b-a} \cdot \frac{b}{b+a}} = \sqrt{\frac{b^2}{b^2-a^2}}$. As stated above, we need at least $\Omega(\sqrt{\frac{b}{b-a}})$ points, so z_0 is an asymptotically optimal choice. We summarize the result

LEMMA 15. *For $a < b$, a polygonal region $\bar{D}_{a,b}$ as above with $O(\sqrt{\frac{b}{b-a}})$ (rational) points can be computed using $O(\sqrt{\frac{b}{b-a}})$ arithmetic operations.*

The Minkowski sum of an arbitrary polygonal region with n vertices and a convex polygonal region with k vertices has complexity $O(kn)$ and it can be computed in $O(nk\log^2(nk))$ operations by a simple divide-and-conquer approach, using a sweep line algorithm in the conquer step [17]. Using generalized Voronoi diagrams where the distance is based on the convex summand of the Minkowski sum operation [18], we obtain an improved algorithm, which requires only $O(kn\log(kn))$ operations. In combination with Lemma 15, this leads to the following complexity bound for the two approximation algorithms.

THEOREM 16. *Algorithms* APPROXDECIDEINTERIOR *and* APPROXDECIDEEXTERIOR *each requires*

$$O(n\frac{r}{\delta}\sqrt{\frac{\varepsilon}{\delta}} \cdot \log(n\frac{r}{\delta}\sqrt{\frac{\varepsilon}{\delta}}))$$

arithmetic operations with rational numbers.

We remark that the $O(n \log n)$ bound for DECIDE refers to operations with real numbers instead. In Section 5 we present examples of approximative decisions.

Searching $\hat{\varepsilon}$. Consider the problem of computing $\hat{\varepsilon}$, the minimal ε such that DECIDE returns YES for fixed Q and r. We have no algorithm to compute that value exactly; however, a simple binary search strategy using DECIDE lets us approximate $\hat{\varepsilon}$ to arbitrary precision.

We demonstrate that this approximation can be achieved also by combining the two approximative one-sided decisions. Again, we perform a binary search on ε. We chose some initial $\delta_0 < \varepsilon$. In case a clear answer is derived (YES for APPROXDECIDEINTERIOR or NO for APPROXDECIDEEXTERIOR) we proceed as if we were using DECIDE. In the other case both approximative algorithms fail to bisect the search range. From Theorems 11 and 13 it follows that the chosen ε is only at most 2δ away from $\hat{\varepsilon}$. If 2δ is smaller than the desired precision we stop. Otherwise, we halve δ and continue. That is, the approximative algorithms conduct the subdivision of ε with the current δ or we halve δ again. A similar binary-search-like strategy can be used to find the maximal radius r for which DECIDE returns YES for fixed Q and ε.

4. DECONSTRUCTING CONVEX POLYGONS

Assume that the input Q to Algorithm 1 is a convex polygon. We first improve the decision algorithm such that it runs in linear time (Algorithm 4). Then we look for a polygon P with a minimal number of vertices (OPT) such that Q is ε-close to offset(P, r). We give a simple linear-time approximation algorithm that produces a polygon with at most OPT + 1 vertices.

LEMMA 17. *If Q is a convex polygonal region, then Π, as computed by* DECIDE *(Algorithm 1), is also a convex polygon, and it can be computed in $O(n)$ time.*

PROOF. Q is the intersection of the half-planes bounded by lines that support the polygon edges. Observe that Π can be directly constructed from Q by shifting each such line by $r - \varepsilon$ inside the polygon, which shows that Π is convex. For the time complexity, we divide the shifted edges of Q into those bounding Q from above, and those bounding Q from below (we assume w.l.o.g. that no edge is vertical). Consider the former edges; the lines supporting those edges have slopes that are monotonously decreasing when traversing the edges from left to right. We have to compute their lower envelope; for that, we dualize by mapping $y = mx + c$ to $(m, -c)$, which preserves above/below relations, and compute the upper hull of the dualized points. Since we already know the order of the points in their x-coordinate, this can be done in linear time using Graham's scan [19, 20]. The same holds for the edges bounding from below, taking the upper envelope/lower hull. □

DECIDE first computes Π and checks whether $Q \subseteq \text{offset}(\Pi, r+\varepsilon)$. We replace the latter step for convex polygons: Let $q_1, \ldots, q_n$ be the vertices of Q (in counterclockwise order) and define $K_i = D_{r+\varepsilon}(q_i)$, namely the disk of radius $r + \varepsilon$ centered at q_i. We check whether all these disks intersect Π:

Algorithm 4 CONVEXDECIDE(Q, r, ε)

(1) $Q_\varepsilon \leftarrow \text{offset}(Q, \varepsilon)$
(2) $\Pi \leftarrow \text{inset}(Q_\varepsilon, r)$
(3) if $K_i \cap \Pi \neq \emptyset$ for all $i = 1, \ldots, n$, return YES, otherwise return NO

LEMMA 18. CONVEXDECIDE *agrees with* DECIDE *on convex input polygons Q and runs in $O(n)$ time.*

PROOF. For correctness, it suffices to prove that offset(Π, r) is ε-close to Q if and only if each K_i intersects Π: Indeed, if any K_i does not intersect Π, then q_i has distance more than $r + \varepsilon$ to Π, so Q is not ε-close to the offset. Otherwise, if each disk K_i intersects Π, offset$(\Pi, r + \varepsilon)$ contains each vertex of Q. Since it is a convex set (as the Minkowski sum of two convex sets), it also covers each edge of Q. Thus, $Q \subseteq \text{offset}(\Pi, r + \varepsilon)$, which ensures that Q is ε-close to the offset by Proposition 3.

For the complexity, Lemma 17 shows that the computation of Π runs in linear time. We still have to deminstrate that the last step of the algorithm (checking for non-empty intersections) also takes a linear time. Let $e_1, \ldots, e_m$ be the edges of Π (with $m < n$). To check for an intersection of K_i with Π, we traverse the edges and check for an intersection, returning NO if no such edge is found. However, if such an edge, say e_j was found, we start the search for an intersection of the next disk K_{i+1} at e_j, again traversing the edges in counterclockwise order. Using this strategy, and noting that $K_1, \ldots, K_n$ are arranged in counterclockwise order around Π, it can be easily seen that we iterate at most twice through the edges of Π. $\square$

Reducing the number of vertices. We assume that offset(Π, r) is ε-close to Q. We prefer a simple-looking approximation of Q, thus we seek a polygon $P \subseteq \Pi$ whose offset is ε-close to Q, but with fewer vertices than Π. Any such P intersects each of the bulged regions of radius $r + \varepsilon$: $\kappa_i := K_i \cap \Pi, i = 1, \ldots, n$. We call these bulged regions Π's *eyelets*. The converse is also true: Any *convex* polygon $P \subseteq \Pi$ that intersects all eyelets $\kappa_1, \ldots, \kappa_n$ has an r-offset that is ε-close to Q.

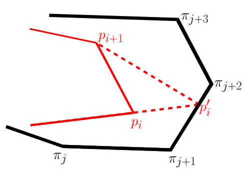
The following observation is a simple consequence of Proposition 3:

PROPOSITION 19. *If* offset(P, r) *is ε-close to Q, and $P \subseteq P' \subseteq \Pi$, then* offset$(P', r)$ *is ε-close to Q.*

We call a polygonal region P *(vertex-)minimal*, if its r-offset is ε-close to Q, and there exists no other such region with fewer vertices. Necessarily, a minimal P must be convex – otherwise, its convex hull CH(P) has fewer vertices and it can be seen by Proposition 19 that offset(CH$(P), r)$ is also ε-close to Q. By the next lemma, we can restrict our search to polygons with vertices on $\partial\Pi$.

LEMMA 20. *There exists a minimal polygonal region $P \subseteq \Pi$ the vertices of which are all on $\partial\Pi$.*

PROOF. We pull each vertex $p_i \notin \partial\Pi$ in the direction of the ray emanating from p_{i-1} towards p_i until it intersects $\partial\Pi$ in the point p'_i (dragging p_i's incident edges along with it); see the enclosed illustration.



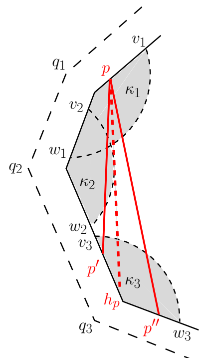
For $P' = (p_1, \ldots, p_{i-1}, p'_i, p_{i+1}, \ldots, p_m)$: $P \subseteq P' \subseteq \Pi$, offset$(P', r)$ is ε-close to Q by Proposition 19. $\square$

We call a polygonal region P *good*, if $P \subseteq \Pi$, all vertices of P lie on $\partial\Pi$, and P intersects each eyelet $\kappa_1, \ldots, \kappa_n$. Note that any good P is convex.

DEFINITION 21. *For two points $u, u' \in \partial\Pi$, we denote by $[u, u'] \subset \partial\Pi$ all points that are met when travelling along $\partial\Pi$ from u to u' in counterclockwise order. Likewise, we define half-open and open intervals $[u, u')$, $(u, u']$, (u, u').*

Let $\kappa_i = K_i \cap \Pi$ be q_i's eyelet as before. Consider $\kappa_i \cap \partial\Pi$. The portion of that intersection set that is visible from q_i (considering Π as an obstacle) defines a (ccw-oriented) interval $[v_i, w_i] \subset \partial\Pi$. We call v_i the *spot* of the eyelet κ_i. Finally, for $u, u' \in \partial\Pi$, we say that the segment $\overline{uu'}$ is *good*, if for all spots $v_i \in (u, u')$, $\overline{uu'}$ intersects the corresponding eyelet κ_i.

The figure to the right illustrates these definitions: The segment $\overline{pp'}$ is good, whereas $\overline{pp''}$ is not good, because $v_2 \in (p, p'')$, but the segment does not intersect κ_2.



THEOREM 22. *Let P be a convex polygonal region with all its vertices on $\partial\Pi$. Then, P is good if and only if all its bounding edges are good.*

PROOF. We first prove that if all the edges of P are good, then P is good. It suffices to argue that it intersects all eyelets $\kappa_1, \ldots, \kappa_n$. Let $p_1, \ldots, p_k$ be the vertices of P in counterclockwise order. Any spot v_i of an eyelet κ_i either corresponds to some vertex p_ℓ of P, or lies inside some interval $(p_\ell, p_{\ell+1})$. Since $\overline{p_\ell p_{\ell+1}}$ is good, it intersects κ_i.

For the converse, assume that $\overline{p_\ell p_{\ell+1}}$ is not good, which encloses with the interval $(p_\ell, p_{\ell+1})$ a polygonal region $R \subseteq \Pi \setminus P$. Hence, there is a spot $v_i \in R$ such that $\overline{p_\ell p_{\ell+1}}$ does not intersect the eyelet κ_i. It follows that the entire κ_i is inside R (see the illustration above, considering $\overline{pp''}$ and κ_2). Thus, $P \cap \kappa_i = \emptyset$, and so P cannot be good. $\square$

For $u \in \partial\Pi$, we define its *horizon* $h_u \in \partial\Pi$ as the endpoint of the longest good segment $\overline{uh_u}$ going on $\partial\Pi$ in counterclockwise direction. Consider again the figure above: The segment $\overline{ph_p}$ is tangential to κ_2, so if going any further than h_p on $\partial\Pi$ from p, the segment would miss κ_2 and thus become non-good.

LEMMA 23. *Let P be a good polygonal region, and $u \in \partial\Pi$. Then, P has a vertex $p \in (u, h_u]$.*

PROOF. Assume to the contrary that P has no such vertex, and let $p_1, \ldots, p_\ell$ be its vertices on $\partial\Pi$. Let p_j be the vertex of P such that $u \in (p_j, p_{j+1})$. Then, also $h_u \in (p_j, p_{j+1})$, because otherwise, $p_{j+1} \in (u, h_u]$. Since P is good, the segment $\overline{p_j p_{j+1}}$ is good, too. It is not hard to see that, consequently, both $\overline{p_j u}$ and $\overline{up_{j+1}}$ are good. However, the latter contradicts the maximality of the horizon h_u. $\square$

For an arbitrary initial vertex $s \in \partial\Pi$, we finally specify a polygonal region P^s by iteratively defining its vertices. Set $p_1 := s$. For any $j \geq 1$, if the segment $\overline{p_j s}$, which would close P^s, is good, stop. Otherwise, set $p_{j+1} := h_{p_j}$. Informally, we always jump to the next horizon until we can reach s again without missing any of the eyelets. By construction, all segments of P^s are good, so P^s itself is good. The (almost-)optimality of this construction mainly follows from Lemma 23.

THEOREM 24. *Let P be a minimal polygonal region for Q, having* OPT *vertices. Then, for any $s \in \partial\Pi$, P^s has at most* OPT + 1 *vertices.*

PROOF. We first prove that P^s has the minimal number of vertices among all good polygonal regions that have s as a vertex. Let $s := p_1, \ldots, p_m$ be the vertices of P^s. There are $m - 1$ segments of the form $\overline{p_\ell h_{p_\ell}}$. By Lemma 23, any good polygonal region has a vertex inside each of the intervals $(p_\ell, h_{p_\ell}]$. Together with the vertex at s, this yields at least m vertices, thus P^s is indeed minimal among these polygonal regions.

Next, consider any minimal polygonal region $P^\star$. We can assume that all its vertices are on $\partial\Pi$ by Lemma 20. If s is not a vertex of $P^\star$, we add it to the vertex set and obtain a polygonal region P' with at most $\mathrm{OPT}+1$ vertices that has s as a vertex. P^s has at most as many vertices as P', so $m \leq \mathrm{OPT}+1$. $\square$

As each visit of an eyelet requires constant time, the construction of a horizon is proportional to the number of visited eyelets, and there are only linearly many eyelets. Thus, we can state:

THEOREM 25. *For an arbitrary initial vertex s, computing P^s requires $O(n)$ time.*

PROOF. We prove that computing the horizon of a point u takes a number of operations proportional to the number of eyelets that are visited by the segment $\overline{uh_u}$. Let us consider an arbitrary $u \in \partial\Pi$. By rotating appropriately, we can assume, without loss of generality, that u lies on a vertical edge of Π (or, if u is a vertex, that the next edge in counterclockwise order is vertical), and that the edge is traversed top-down. The horizon is determined by the slope of the edge at u. Note that for each eyelet $\kappa_1, \ldots, \kappa_n$, there is an interval of slopes $I_1^{(u)}, \ldots, I_n^{(u)}$ such that the segment from u with slope m intersects κ_i if and only if $m \in I_i^{(u)}$. Furthermore, each single $I_i^{(u)}$ can be computed with a constant number of arithmetic operations. Assuming that the next eyelet to be travelled from the current p_i is κ_j, we can iteratively compute the intersections $I_j \cap I_{j+1} \cap I_{j+2}, \ldots$ until $I_j \cap \ldots \cap I_{j+k}$ is empty. In this case, we choose $m_i := \max(I_j \cap \ldots \cap I_{j+k-1})$ as the slope for the next segment, which must be $\overline{p_i h_{p_i}}$ since it is good by construction, and any larger slope would produce a non-good segment. Based on this property, it is easy to show that computing P^s needs a number of operations which is proportional to n, the number of eyelets. $\square$

5. IMPLEMENTATION

We have implemented the algorithms described in the paper that are based on rational arithmetic using ready-made procedures of CGAL's[2] polygon [21], Minkowski sum[3] [23] and Boolean set-operation [24] packages. We focus on the two-sided decision algorithm with rational δ-approximate constructions described in Section 3: APPROXDECIDEINTERIOR $(Q, r, \varepsilon, \delta)$ and APPROXDECIDEEXTERIOR $(Q, r, \varepsilon, \delta)$.

Recall that the input to the algorithms is a polygonal region Q, a radius r, a solution precision ε and a working precision δ. By construction of approximate offsets and insets with the working precision δ, the algorithms combined determine whether Q is ε-close to the r-offset of an unspecified polygonal shape P and compute a three-valued answer: YES if a valid P exists, NO if it does not, and UNDECIDED if a certified YES or NO cannot be determined for given parameters. How the choice of ε and δ influences this outcome is illustrated in Figure 5.1.

The rational δ-approximation of the disk (for the intermediate constructions) is created as described in Section 3, that is, by selecting vertices with rational coordinates on a circle. In the implementation we reduce the number of vertices by greedily choosing points on the circle at maximal possible distance from each other, keeping the resulting polygon inside the δ-annulus. This reduces the number of points by about 20% to 40%. The decision procedures for the cases depicted in Figure 5.1 (a), (b), (c) and (d) took 0.252, 0.480, 0.388 and 0.844 seconds respectively on a 3GHz Intel Dual Core processor in our tests. See also Figure 5.2 for results on a larger polygon.

Remark: Rational-vertex approximation of offset polygons. We also devised and implemented a novel algorithm for the approximate construction of offset polygons, namely for the direct counterpart of the problems studied thus far in the paper. As mentioned in the Introduction, the exact offset of a polygonal shape whose vertex coordinates are rational, with rational offset radius, can have non-rational vertices. In contrast our new algorithm constructs for given polygonal shape P, offset r, and tolerance ε, an ε-approximation of offset(P, r) whose vertex-coordinates are rational and that prefers circular arcs over piecewise-linear approximation where the exact offset also shows circular arcs. We refer the interested reader to the Supplementary Material[4] for algorithmic details and experimental results.

6. OPEN PROBLEMS

We have shown how to decide whether a given arbitrary polygonal shape Q is composable as the Minkowski sum of another polygonal region and a disk of radius r, up to some tolerance ε. Many related questions remain open. (i) Deconstruction of Minkowski sums seems more difficult when both summands are more complicated than a disk; many practical scenarios may raise this general deconstruction problem. (ii) It would be interesting to analyze the deconstruction not only under the Hausdorff distance but for other similarity measures, such as the Frêchet or the symmetric distance. (iii) Can one remove the extra vertex when seeking an optimal (vertex minimal) polygonal summand P in the convex case. (iv) Finding an optimal or near-optimal polygonal summand in the non-convex case seems challenging. (v) As in polygonal simplification, we could also search for the polygonal region with a given number of vertices whose r-offset minimizes the (Hausdorff) distance to the given shape. (vi) The offset-deconstruction problem can be reformulated in higher dimensions. We consider especially the three-dimensional case to be of practical relevance.

Acknowledgements

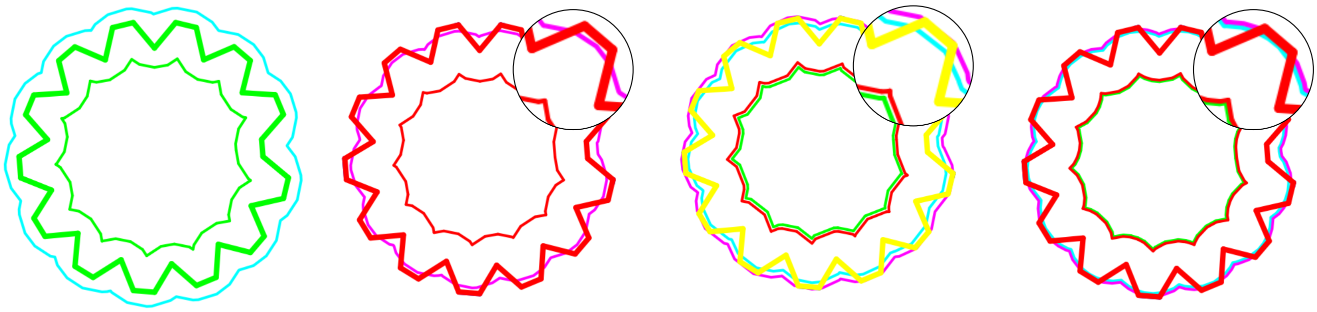
We thank Eyal Flato (Plataine Ltd.) for raising the offset deconstruction problem in connection with wood cutting. We also thank Tim Bretl (UIUC) for suggesting the digital-pen offset deconstruction problem. This work has been supported in part by the Israel Science Foundation (grant no. 236/06), by the German-Israeli Foundation (grant no. 969/07), by the Hermann Minkowski–Minerva Center for Geometry at Tel Aviv University, and by the EU Project under Contract No. 255827 (CGL—Computational Geometry Learning).

---

[2]The Computational Geometry Algorithms Library; see `www.cgal.org`.

[3]Notice that we use a convolution-cycles based implementation of Minkowski sums, which is known to perform very well in practice and in particular it was experimentally shown to perform better than divide-and-conquer based implementations described in Section 3 on many inputs [22].

[4]`http://acg.cs.tau.ac.il/projects/internal-projects/deconstructing-approximate-offsets`



(a) $\varepsilon = \frac{1}{3} \cdot r$, $\delta = \frac{1}{36} \cdot r$ (b) $\varepsilon = \frac{1}{9} \cdot r$, $\delta = \frac{1}{36} \cdot r$ (c) $\varepsilon = \frac{1}{6} \cdot r$, $\delta = \frac{1}{4} \cdot \varepsilon$ (d) $\varepsilon = \frac{1}{6} \cdot r$, $\delta = \frac{1}{10} \cdot \varepsilon$

Figure 5.1: Dependency of the algorithm outcome on ε and δ: The input polygon (wheel) appears in bold line. It is colored according to its approximability with the given parameters: green for YES, red for NO and yellow for UNDECIDED. The inner approximation of the offset source polygon and its approximate $r+\varepsilon$-offset are drawn in green and cyan respectively. Their outer-approximation counterparts are drawn in red and magenta. Figures (a) and (b) demonstrate how when ε is tightened from $\frac{1}{3} \cdot r$ to $\frac{1}{9} \cdot r$, with the same r and δ, the decision result changes from YES to NO. The green polygon inside the input polygon in (a) is a possible r-offset source. The magnification in (b) highlights the area of the input polygon that does not fit inside the outer δ-approximation (in magenta) of maximal possible $r+\varepsilon$-offset. Figures (c) and (d) show how when δ is decreased from $\frac{1}{4} \cdot \varepsilon$ to $\frac{1}{10} \cdot \varepsilon$, for the same r and ε, the decision result changes from UNDECIDED to NO, namely in the latter case the algorithm is able to produce a certified negative answer.

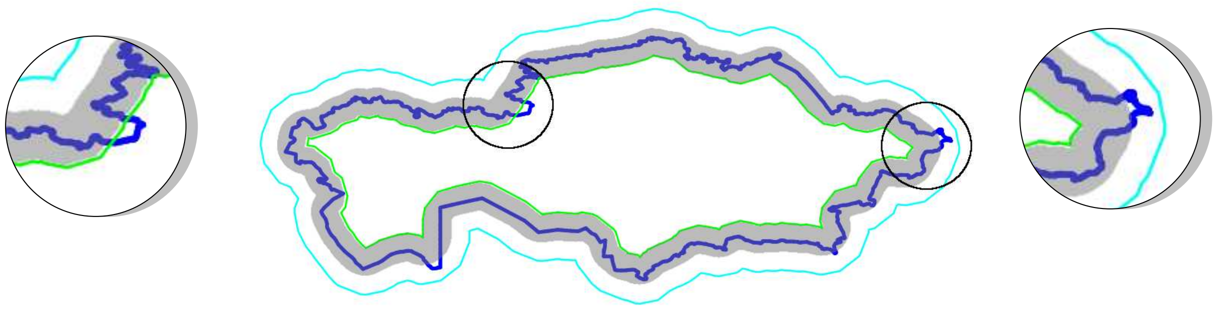


Figure 5.2: A map of Kazakhstan, represented as a polygon Q (in bold blue) with 1881 vertices, is approximable for $\varepsilon = \frac{1}{2} \cdot r$ and $\delta = \frac{1}{8} \cdot \varepsilon$. A valid source polygon P (in green) has 335 vertices. Offset(P,r) (shown as transparent gray r-strip around P) is inside the ε-offset of the input Q by construction. The δ-approximation of the ε-offset of Offset(P,r) (as computed in line (3) of ApproxDecideInterior $(Q, r, \varepsilon, \delta)$) is drawn in cyan and has 261 vertices. Since the cyan polygon contains Q, the Offset(P,r) and Q have Hausdorff distance of at most ε, that is, Q is approximable and P is a valid source polygon. Approximability computation took 3.868 seconds in this case. The magnification on the left highlights some cavities in the input polygon that have no effect on the Hausdorff distance within this tolerance ε. The magnification on the right demonstrates a sharp end that would prevent Q's approximability with a tighter ε.

7. REFERENCES

[1] Hansen, A., Arbab, F.: An algorithm for generating NC tool paths for arbitrarily shaped pockets with islands. ACM Trans. Graph. **11** (April 1992) 152–182

[2] Matheron, G.: Random sets and integral geometry. Wiley New York, (1974)

[3] Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, Inc., Orlando, FL, USA (1983)

[4] Damen, J., van Kreveld, M., Spaan, B.: High quality building generalization by extending the morphological operators. In: 11th ICA Workshop on Generalisation and Multiple Reslides. (June 2008)

[5] Drysdale, R., Rote, G., Sturm, A.: Approximation of an open polygonal curve with a minimum number of circular arcs and biarcs. Computational Geometry, Theory and Applications **41**(1-2) (2008) 31 – 47

[6] Heimlich, M., Held, M.: Biarc approximation, simplification and smoothing of polygonal curves by means of Voronoi-based tolerance bands. Int. J. Comput. Geometry Appl. **18**(3) (2008) 221–250

[7] Sallee, G.T.: Minkowski decomposition of convex sets. Israel Journal of Mathematics **12**(3) (September 1982) 266–276

[8] Ostrowski, A.: Über die Bedeutung der Theorie der konvexen Polyeder für die formale Algebra. Jahresberichte Deutsche Math. Ver. **30** (1921) 98–99 English version: "On the Significance of the Theory of Convex Polyhedra for Formal Algebra" ACM Sigsam Bulletin **33** (1999).

[9] Gao, S., Lauder, A.: Decomposition of polytopes and polynomials. Discrete & Computational Geometry **26** (2001) 89–104

[10] Emiris, I., Tsigaridas, E.: Minkowski decomposition of convex lattice polygons. In: Algebraic geometry and geometric modeling. Springer (2006) 217–236

[11] Preparata, F.P., Shamos, M.I.: Computational Geometry: An Introduction. 3rd edn. Springer-Verlag (October 1990)

[12] Aggarwal, A., Booth, H., O'Rourke, J., Suri, S., Yap, C.K.: Finding minimal convex nested polygons. Inf. Comput. **83**(1) (1989) 98–110

[13] Berberich, E., Halperin, D., Kerber, M., Pogalnikova, R.: Polygonal reconstruction from approximate offsets. In Vahrenhold, J., ed.: 26th European Workshop on Computational Geometry: Workshop Proceedings, Dortmund, Germany, TU Dortmund (March 2010) 65–68

[14] Yap, C.K.: An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments. Discrete & Computational Geometry **2** (1987) 365–393

[15] Held, M.: On the Computational Geometry of Pocket Machining. Volume 500 of Lecture Notes in Computer Science. Springer (1991)

[16] Canny, J., Donald, B., Ressler, E.K.: A rational rotation method for robust geometric algorithms. In: SCG '92: Proceedings of the Eighth Annual Symposium on Computational geometry, New York, NY, USA, ACM (1992) 251–260

[17] Kedem, K., Livne, R., Pach, J., Sharir, M.: On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles. Discrete & Computational Geometry **1**(1) (1986) 59–71

[18] Leven, D., Sharir, M.: Planning a purely translational motion for a convex object in two-dimensional space using generalized Voronoi diagrams. Discrete & Computational Geometry **2** (1987) 9–31

[19] Andrew, A.M.: Another efficient algorithm for convex hulls in two dimensions. Inform. Process. Lett. **9**(5) (1979) 216–219

[20] Graham, R.L.: An efficient algorithm for determining the convex hull of a finite planar set. Inform. Process. Lett. **1** (1972) 132–133

[21] Giezeman, G.J., Wesselink, W.: 2D polygons. In: CGAL User and Reference Manual. 3.7 edn. CGAL Editorial Board (2010) `http://www.cgal.org/Manual/3.7/doc_html/cgal_manual/packages.html#Pkg:Polygon2`.

[22] Wein, R.: Exact and approximate construction of offset polygons. Computer-Aided Design **39**(6) (2007) 518–527

[23] Wein, R.: 2D Minkowski sums. In: CGAL User and Reference Manual. 3.7 edn. CGAL Editorial Board (2010) `http://www.cgal.org/Manual/3.7/doc_html/cgal_manual/packages.html#Pkg:MinkowskiSum2`.

[24] Fogel, E., Wein, R., Zukerman, B., Halperin, D.: 2D regularized Boolean set-operations. In: CGAL User and Reference Manual. 3.7 edn. CGAL Editorial Board (2010) `http://www.cgal.org/Manual/3.7/doc_html/cgal_manual/packages.html#Pkg:BooleanSetOperations2`.