THE HAUSDORFF CORE PROBLEM ON SIMPLE POLYGONS *

Reza Dorrigiv,[†] *Stephane Durocher,*[‡] *Arash Farzan,*[§] *Robert Fraser,*[‡] *Alejandro López-Ortiz,*[¶] *J. Ian Munro,*[¶] *Alejandro Salinger,*[‖] *and Matthew Skala*[‡]

Abstract. We present a study of the Hausdorff Core problem on simple polygons. A polygon Q is a k-bounded Hausdorff Core of a polygon P if P contains Q, Q is convex, and the Hausdorff distance between P and Q is at most k. A Hausdorff Core of P is a k-bounded Hausdorff Core of P with the minimum possible value of k, which we denote $k_{\min}$. Given any k and any $\varepsilon > 0$, we describe an algorithm for computing a k'-bounded Hausdorff Core (if one exists) in $O(n^3 + n^2\varepsilon^{-4}(\log n + \varepsilon^{-2}))$ time, where $k' < k + d_{\text{rad}} \cdot \varepsilon$ and d_{rad} is the radius of the smallest disc that encloses P and whose center is in P. We use this solution to provide an approximation algorithm for the optimization Hausdorff Core problem which results in a solution of size $k_{\min} + d_{\text{rad}} \cdot \varepsilon$ in $O(\log(\varepsilon^{-1})(n^3 + n^2\varepsilon^{-4}(\log n + \varepsilon^{-2})))$ time. Finally, we describe an approximation scheme for the k-bounded Hausdorff Core problem which, given a polygon P, a distance k, and any $\varepsilon > 0$, answers true if there is a $((1+\varepsilon)k)$-bounded Hausdorff Core and false if there is no k-bounded Hausdorff Core. The running time of the approximation scheme is in $O(n^3 + n^2\varepsilon^{-4}(\log n + \varepsilon^{-2}))$.

Keywords: Polygonal Approximation, Hausdorff Distance, Approximation Algorithms, d-Core.

*A preliminary version of this work appeared in the proceedings of the 2009 Algorithms and Data Structures Symposium (WADS) [DDF+09]. Funding for this research was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), partially under the Discovery Grants program and the NSERC Strategic Grant on Optimal Data Structures for Organization and Retrieval of Spatial Data.

[†] *Dalhousie University, Halifax, Nova Scotia, Canada*, rdorrigiv@cs.dal.ca

[‡] *University of Manitoba, Winnipeg, Manitoba, Canada*, {durocher,fraser,mskala}@cs.umanitoba.ca

[§] *Max-Planck-Institut für Informatik, Saarbrücken, Germany*, afarzan@mpi-inf.mpg.de

[¶] *University of Waterloo, Waterloo, Ontario, Canada*, {alopez-o,imunro}@cs.uwaterloo.ca

[‖] *Saarland University, Saarbrücken, Germany*, salinger@cs.uni-saarland.de

1 Introduction

Given two polygons P and Q, the Hausdorff distance[1] between P and Q may be expressed formally as

$$H(P,Q) = \max\left\{\sup_{p\in P}\inf_{q\in Q} \operatorname{dist}(p,q), \sup_{q\in Q}\inf_{p\in P} \operatorname{dist}(p,q)\right\}, \tag{1}$$

where $\operatorname{dist}(p,q)$ is the Euclidean distance from p to q.

Definition 1. *The Hausdorff Core Problem:* Given a simple polygon P, find a convex polygon Q such that $Q \subseteq P$ and $H(P,Q)$ is minimized, where $H(P,Q)$ denotes the Hausdorff distance between P and Q. We call Q a *Hausdorff Core* of P.

The Hausdorff Core problem is equivalent to a disc piercing problem: suppose we place discs of radius k centered on all vertices on the convex hull of a polygon P, so that each disc $D_i \in \mathcal{D}$ has center point p_i, where $\mathcal{D}$ denotes the set of discs. We wish to pierce all discs in $\mathcal{D}$ with a convex polygon Q; if $Q \cap D_i \neq \emptyset$ for each $D_i \in \mathcal{D}$, then we say that Q pierces $\mathcal{D}$. If we additionally require that $Q \subseteq P$, then Q is a k-bounded Hausdorff Core of P. We establish this relationship formally in Lemmas 1 and 3.

This work was initially motivated by the problem of path planning in the context of navigation at sea[2]. In this application, a plotted course must be tested against bathymetric soundings to ensure that the ship will not run aground. We suppose the soundings have been interpolated into contour lines and the plotted course is given as a polygonal line. Although contour lines can be arbitrarily complicated, typical shipping routes run far from potential obstacles for the majority of their trajectories, and only short segments require more careful route planning. As a result, most tests for intersection between a path and a contour line should be easy: we could subdivide the map into polygonal regions so that most intersection tests are against regions with properties that permit fast tests (e.g., see [SW87]), ideally reserving more expensive tests for the rare cases where the path comes close to intersecting the terrain.

The search for easily testable areas motivates the study of the simplification of a contour line into a simpler object which is either entirely contained within the contour line or else fully contains it. For example, it is sensible to consider the convex hull of contours corresponding to regions that are too shallow. In this work we consider the case in which the simplified polygon must be convex and contained by the contour line.

We restrict the discussion to simple polygons, where we define a *polygon* P as a closed region of the plane, whose boundary ∂P is represented as a polygonal chain on n distinct vertices $P = \{p_0, \ldots, p_{n-1}\}$ and n edges $e_i = (p_i, p_{i'})$, where $i' = (i+1) \bmod n$. A polygon is *simple* if edges only intersect at vertices, and all vertices are incident upon exactly two edges. Finally, a polygon P is *convex* if for all points p and p' in P, the line segment $\overline{pp'}$ is contained in P.

[1] The Hausdorff distance is equivalent to Blaschke's Nachbarschaftsmaß [Gru83].

[2] This research was part of the NSERC Optimal Data Structures for Organization and Retrieval of Spatial Data project. CARIS, the corporate partner on the project, develops marine GIS software.

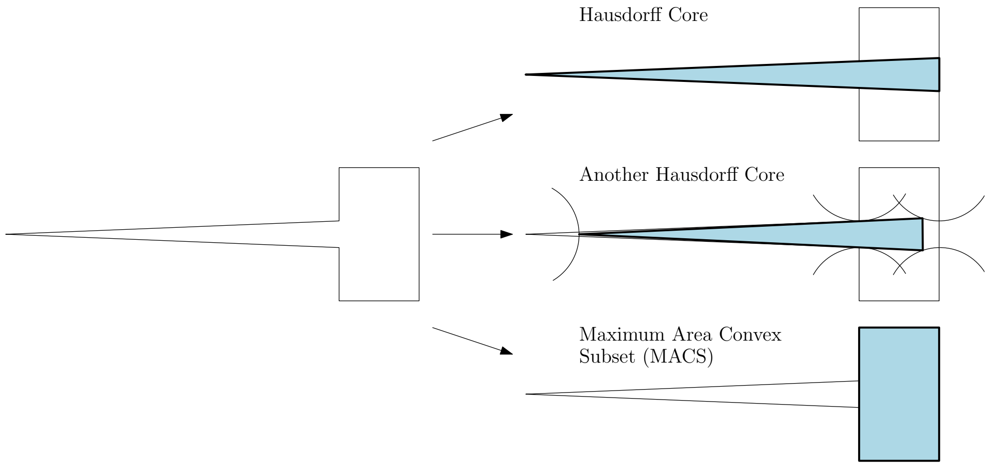


Figure 1: The optimal solutions for approximating polygons may vary significantly for different error metrics. In this example, we show optimal solutions for the Hausdorff distance and the maximum area convex subset (MACS), also known as the "potato" [CY86]). Notice that the two Hausdorff Core solutions have the same Hausdorff distance between the polygons, which corresponds to the radius of the circular arcs centered on vertices of the initial polygon.

The Hausdorff Core problem is intended to capture a sense of "closeness" between the input and approximating polygons. While there are a number of measures that may be used (we discuss several metrics in Section 2.1), the Hausdorff Core is perhaps most intuitive in that it is optimized when the maximum distance between the polygons is minimized. See Figure 1 for an illustration of convex approximating polygons: the first two minimize the Hausdorff distance between the polygons, while the third minimizes the difference in area between the polygons.

2 Related Work

Suppose that we have a contour line represented as a piecewise linear curve or polygon with n points. For our application, this corresponds to a contour of the ocean floor at a depth of x meters. We wish to approximate the contour line with a simpler curve or polygon. The simplification should be similar to the original, or close according to some measure of distance, and it should have as few points as possible. The two goals of closeness and having few points will in general conflict with each other.

If we specify a bound on the distance between the simplification and the original curve or polygon, then minimize the number of points subject to the bound on distance, that is called a min-# *problem*. Conversely, if we specify a bound on the number of points in the simplification and minimize the distance to the original subject to the bound on points, that is called a min-ε *problem*. In either kind of problem, many different measures of distance may be used. Both kinds are well-studied, with many different choices for the measure of distance. The problem we consider here is a version of the min-ε problem in which distance is measured by the Hausdorff metric and we have an additional constraint that the simplification must be contained within the original polygon.

The containment constraint comes from our application. When approximating a contour of depth x, we must ensure that for every point contained on the deeper side of the approximating contour, the actual depth is at least x. Without such a constraint, a ship following our approximated contour instead of the original could run aground expecting the ocean at a given point to be deeper than it actually is. This constraint forces us to err on the side of safety. It implies that an approximating polygon cannot cross the original contour polygon.

Without the containment constraint, approximation of piecewise linear curves with simpler piecewise linear curves is well studied [DP73, RW74, II86, PV94, KF03, DLMS12]. We review some of that work first, then discuss previous work on problems with containment constraints.

2.1 Error Metrics

The *two-strip solution* presented by Reumann and Witkam [RW74] was one of the first approaches to min-ε curve simplification. It uses a modified version of the Hausdorff metric, in which each vertex of the original curve is imagined to have a disc of radius ε centered on it and the approximating path must pass through each of these discs in order. Points from the original curve are then deleted subject to the constraints, leaving an approximating curve defined by a subset of the input points. Much subsequent work has presented polynomial time results for similar concepts [Peu76, Rob85, LY90, Hob93, Dae03]. In particular, Ballard [Bal81] described a hierarchical approximating scheme called *strip trees*. Our problem is more complex due to the convexity and containment constraints, but the idea of centering a disc on each vertex of the input is similar to how our algorithm works.

In general, the points defining the optimal simplified chain may not be points from the input. Guibas et al. [GHMS93] studied the version where simplified points need not be selected from the input for polygonal chains, and their solution entails finding a minimum link path which intersects each disc. They use a linear time ordered stabbing technique [EW91] to find a path which satisfies the ε approximation of the original curve under the Hausdorff metric, unless the path is required to be simple, in which case the problem is NP-hard [GHMS93].

The Fréchet metric describes the maximum distance between two points that follow continuous monotonic trajectories along the two curves being compared. It is much more difficult to work with than the Hausdorff metric, but may better express a useful concept of similarity for curves in some contexts. For instance, in Figure 2, P and P' have small Hausdorff distance but large Fréchet distance, and indeed they intuitively seem to be quite different curves.

However, the Fréchet metric is not intended for closed curves, and a modified version must be used for closed curves [AG92]. The Hausdorff distance is standard for polygons [Gru83, ABB95, FMR+92, Rot92, CC05, LR05], and we use the Hausdorff distance here. Other metrics of possible interest include *geodesic width* [EGHP+01]; *link*

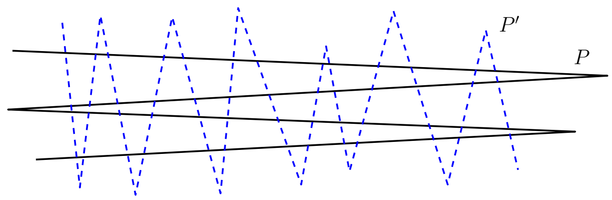


Figure 2: Two curves with small Hausdorff distance but large Fréchet distance, adapted from Alt and Godau [AG92].

width [EGHP+01]; *Eggleston measure* [Gru83]; and the difference in *area* between two polygons [CY86].

2.2 Constrained Approximation and Cores

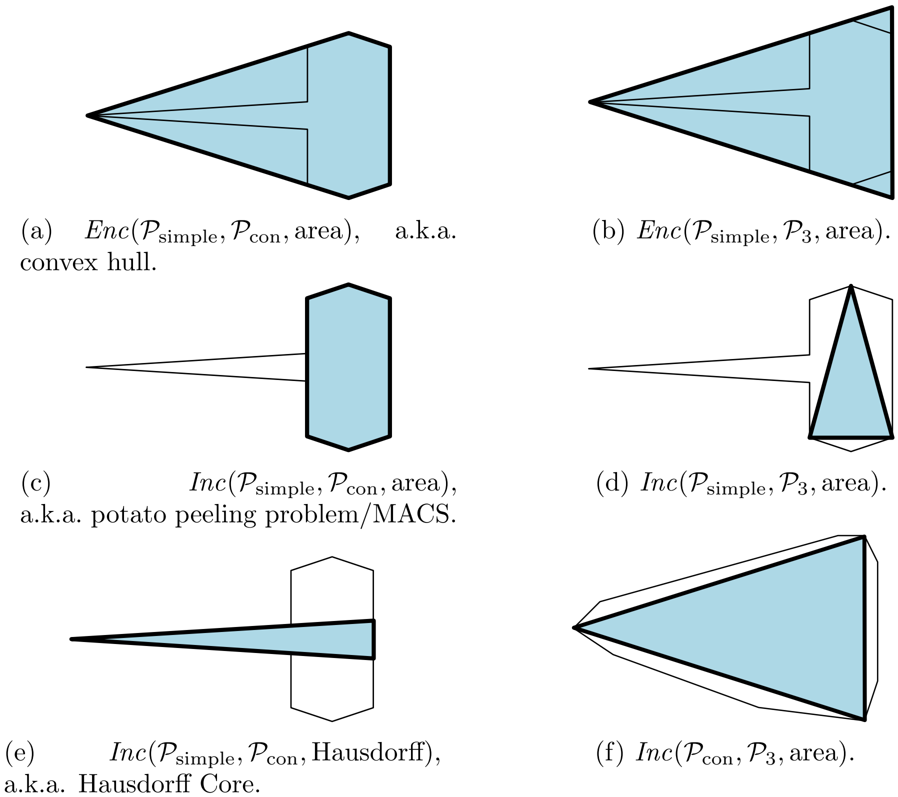
Zhang and Tian [ZT97] add the containment constraint to polygon approximation, in an application closely related to our own. They selectively remove points using the Douglas-Peucker approach [DP73], but only when the new approximation lies to the deep side of the contour line being approximated. Their technique establishes precedent, but the amount of data reduction achieved relative to the original curve is unclear from their examples [ZT97, Figures 3–5], and they provide no formal analysis.

Some previous work has placed constraints on the approximating curve in relation to other properties of the data. Estkowski and Mitchell [EM01] study an approximation problem in which adjacent polygonal curves must maintain their relative positions, in a Geographic Information System (GIS) context. They show their problem to be in a difficult complexity class (MIN PB-complete), and so propose a heuristic approximation with some experimental results showing its efficacy. Li gives a more thorough review of constrained approximation techniques for polygonal curves [Li07, Section 5.6].

Polygon approximation with a containment constraint can be divided into two broad classes of problems: inclusion problems seek an approximation contained in the original polygon, and enclosure problems determine an approximation that contains the original polygon. Formally, let $\mathcal{P}$ and $\mathcal{Q}$ be classes of polygons and let μ be a function on polygons such that for polygons P and Q, $Q \subseteq P \Rightarrow \mu(Q) \leq \mu(P)$. Chang and Yap [CY86] define the inclusion and enclosure problems as follows:

- $Inc(\mathcal{P}, \mathcal{Q}, \mu)$: Given a polygon $P \in \mathcal{P}$, find a polygon $Q \in \mathcal{Q}$ included in P, maximizing $\mu(Q)$.
- $Enc(\mathcal{P}, \mathcal{Q}, \mu)$: Given a polygon $P \in \mathcal{P}$, find a polygon $Q \in \mathcal{Q}$ enclosing P, minimizing $\mu(Q)$.

We illustrate a number of enclosure and inclusion problems in Figure 3. The best known enclosure problem is the convex hull, $Enc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{area})$, where $\mathcal{P}_{\text{simple}}$ is the family of simple polygons and $\mathcal{P}_{\text{con}}$ is the family of convex polygons. Given a convex polygon P as input, many problems are tractable in linear time: $Enc(\mathcal{P}_{\text{con}}, \mathcal{P}_3, \text{area})$ [OAMB86],



(a) $Enc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{area})$, a.k.a. convex hull.

(b) $Enc(\mathcal{P}_{\text{simple}}, \mathcal{P}_3, \text{area})$.

(c) $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{area})$, a.k.a. potato peeling problem/MACS.

(d) $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_3, \text{area})$.

(e) $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{Hausdorff})$, a.k.a. Hausdorff Core.

(f) $Inc(\mathcal{P}_{\text{con}}, \mathcal{P}_3, \text{area})$.

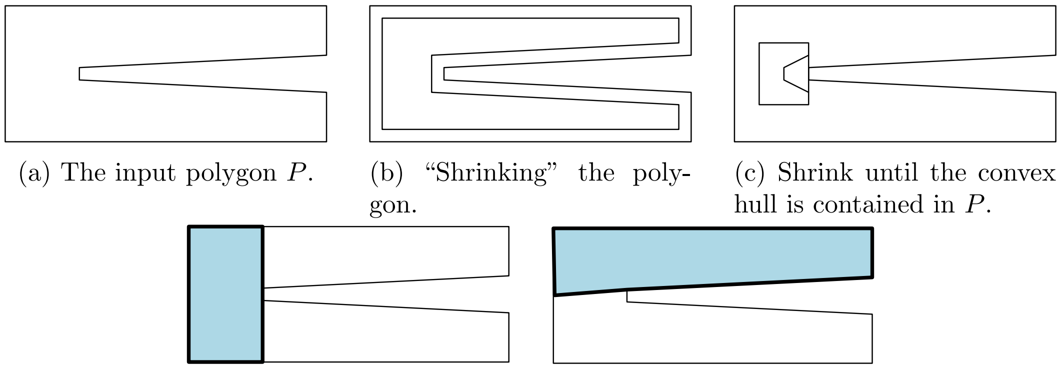
Figure 3: Some inclusion and enclosure problems for approximating polygons.

where $\mathcal{P}_3$ is the set of triangles, $Enc(\mathcal{P}_{\text{con}}, \mathcal{P}_3, \text{perimeter})$ [BM02], and $Enc(\mathcal{P}_{\text{con}}, \mathcal{P}_{\text{par}}, \text{area})$ [STV+95], where $\mathcal{P}_{\text{par}}$ is the family of parallelograms. For general k-gons, $Enc(\mathcal{P}_{con}, \mathcal{P}_k, \text{area})$ can be solved in $O(kn + n \log n)$ time [AP88].

Perhaps the best known inclusion problem is the potato-peeling problem of Chang and Yap [CY86], defined as $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{area})$. The "potato" of the potato peeling problem is formally known as the maximum area convex subset (MACS) [CC05], which is the largest area convex polygon contained in P. There is an $O(n^7)$ time algorithm for this problem, where n is the number of vertices of P, and an $O(n^6)$ time algorithm when the measure is the perimeter, i.e. $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{perimeter})$ [CY86]. The problem of finding the triangle of maximal area included in a convex polygon, $Inc(\mathcal{P}_{con}, \mathcal{P}_3, \text{area})$, can be solved in linear time [DS79]. The generalization of this problem to any k-gon can be solved in time $O(kn + n \log n)$ [AKM+87]. If the input polygon is not restricted to be convex, $Inc(\mathcal{P}_{simple}, \mathcal{P}_3, \text{area})$ can be found in time $O(n^4)$ [MS90].

The inclusion and enclosure problems can also be formulated as minimizing or maximizing a measure $d(P, Q)$ to find what we call a d-Core of P.

Definition 2 (The d-Core Problem)**.** Given a simple polygon P, determine a convex polygon Q such that $Q \subseteq P$ and $d(P, Q)$ is minimized, where $d(P, Q)$ is a measure of the difference between P and Q. We call Q a *d-Core* of P.



(a) The input polygon P.

(b) "Shrinking" the polygon.

(c) Shrink until the convex hull is contained in P.

(d) The solution returned by the Chassery and Coeurjolly algorithm [CC05].

(e) An optimal solution.

Figure 4: Chassery and Coeurjolly algorithm counterexample.

Note that in the case when $\mu(Q)$ is the area, maximizing or minimizing $\mu(Q)$ for the inclusion and enclosure problems respectively is equivalent to minimizing the difference in areas ($d(P,Q) = |\mu(P) - \mu(Q)|$).

Both the inclusion and enclosure problems using the Hausdorff distance as a measure were studied by Lopez and Reisner [LR05]. Given a convex polygon and the number of points permitted in the solution, polynomial-time algorithms were described that minimize the Hausdorff distance (i.e. the min-ε version of the problem). They also studied the min-# version of the problem. In that setting, they show that the inclusion and enclosure problems can be approximated to within one vertex of optimal in $O(n \log n)$ time and $O(n)$ time, respectively.

In the present paper we address the inclusion problem where the objective is to minimize the Hausdorff distance to a convex approximating polygon given a simple (not necessarily convex) polygon as input, i.e. $Inc(\mathcal{P}_{\text{simple}}, \mathcal{P}_{\text{con}}, \text{Hausdorff})$. Chassery and Coeurjolly [CC05] addressed this problem first. Their result is conditional on the *Euclidean 1-center*. For a polygon P the Euclidean 1-center is the point c that minimizes the maximum distance from c to any point in P. When the Euclidean 1-center of the input polygon P is contained in P, the algorithm of Chassery and Coeurjolly [CC05] finds the Hausdorff core of P. It works by shrinking P until its convex hull is contained in the original P. If the shrunken polygon P' is not convex, then the convex hull of P' contains a vertex of P which lies on an edge e of P'. The edge e is used as a cutting line upon P to obtain a new polygon P_1 to be shrunk. The procedure is repeated to obtain P_i' from P_i until P_i' is convex.

If the Euclidean 1-center of P is not contained in P, it is possible to construct examples where that algorithm would not return a Hausdorff Core of P, as shown in Figure 4. Fraser and Nicholson [FN10] provided an $O(n^3)$ time algorithm to find an exact Hausdorff Core solution if the input polygon is simple and contains at most one reflex vertex. Their solution is non-trivial, and it does not immediately extend to polygons with two or more reflex vertices.

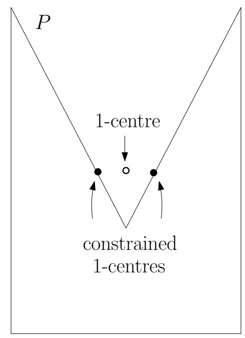


Figure 5: An example where the constrained Euclidean 1-centers (the solid points) are not unique, and are distinct from the unconstrained Euclidean 1-center (the hollow point).

2.3 LP-type Problems and the Constrained 1-Center

Certain settings for optimization problems using the Hausdorff distance metric belong to the class of problems known as *LP-type problems*. LP-type problems were formalized by Sharir and Welzl [SW92] to provide a general framework for linear programming problems with n constraints and d variables so that they may be solved in expected $O(d^2 2^d n)$ time, which translates into a linear time algorithm for many of the problems we are interested in.

Recall that the Euclidean 1-center c of a polygon P minimizes the maximum distance from c to any point of P. If we required our core to be a single point, but did not require it to be contained in P, then c would be an exact solution; finding such a c is an LP-type problem and amenable to the techniques of Sharir and Welzl.

However, even with the restriction to a single point, the Hausdorff Core is not an LP-type problem because of the requirement that the solution Q be contained within P. The *constrained* Euclidean 1-center is a point p_{1c} *contained in* P that minimizes the maximum distance from p_{1c} to any point in P. The unconstrained 1-center is unique, but the constrained 1-center might not be, as shown in Figure 5. If P is not convex, then the search space for p_{1c} cannot be described by a set of linear constraints. If P is convex, the constraints are linear, but the Hausdorff Core problem is also trivial because $Q = P$. The constrained Euclidean 1-center problem was solved by Bose and Toussaint [BT96, Algorithm 1], with an algorithm to compute a constrained Euclidean 1-center of a polygon P with n vertices in $O(n \log n + k)$ time, where k is the number of intersections between P and the furthest point Voronoi diagram of the vertices of P (for simple polygons $k \in O(n^2)$).

Throughout the rest of this work, when we refer to a 1-center, we specifically mean a constrained Euclidean 1-center. We write p_{1c} to represent a constrained Euclidean 1-center of P.

3 The Hausdorff Core

We begin our analysis by formalizing the Hausdorff Core problem, and establishing some fundamental properties of the problem.

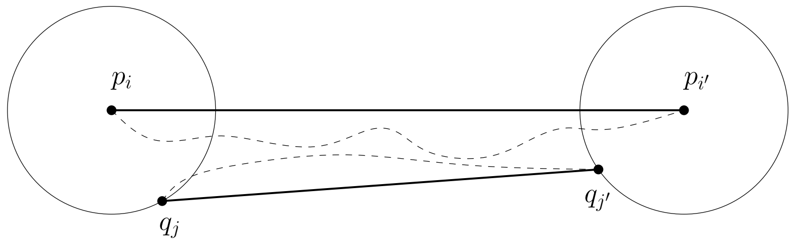


Figure 6: Illustration of Lemma 1. The dashed lines indicate a possible set of boundaries for P and Q between these vertices.

3.1 Definitions

A polygon P can have multiple distinct Hausdorff Cores; our objective is to determine any optimal solution. We consider both the minimization and decision versions of the Hausdorff Core problem for a given simple polygon P. Recall from Definition 1 that the minimization version of the Hausdorff Core problem is to determine a Hausdorff Core of P, i.e. a polygon Q covered by P where $H(P,Q)$ is minimized. The decision version is formalized below:

Definition 3. *The k-bounded Hausdorff Core Problem:* Given a simple polygon P and a non-negative real k, identify whether there exists a convex polygon Q contained in P such that $H(P,Q) \leq k$. Such a polygon Q, if it exists, is referred to as a k-bounded Hausdorff Core of P.

3.2 Hausdorff Core Properties

Given a polygon P and a convex polygon Q inside P, it suffices to measure the maximum distance from the vertices on the convex hull of P to polygon Q to obtain $H(P,Q)$. The distance from points $q \in Q$ to P need not be considered, by the following lemma. Let $CH(P)$ denote the convex hull of P, and P_V denote the set of vertices of P. We write $\overleftrightarrow{q_iq_j}$ (resp. $\overline{q_iq_j}$) to denote a line (resp. minimal line segment) containing points q_i and q_j.

Lemma 1. *Given any simple polygon P and any convex polygon Q contained in P,*

$$\max_{p\in P}\min_{q\in Q}\operatorname{dist}(p,q) \geq \max_{q\in Q}\min_{p\in P}\operatorname{dist}(q,p)\,.$$

Therefore,

$$H(P,Q) = \max_{p\in P}\min_{q\in Q}\operatorname{dist}(p,q).$$

Furthermore, the Hausdorff distance between P and Q is defined by the distance from vertices on the convex hull of P to Q:

$$H(P,Q) = H(CH(P)_V, Q).$$

Proof. Suppose $H(P,Q) = k$, and take the convex hull of P to obtain $CH(P)$. Now identify two consecutive vertices of $CH(P)$, and call them p_i and $p_{i'}$ (Figure 6). By the definition of the convex hull, we know that all vertices of P lie in one of the two half-planes defined by the line incident to p_i and $p_{i'}$; call this half-plane h^P. For ease of discussion, rotate everything so that h^P is equivalent to the $y \leq 0$ half-plane. Since P contains Q, h^P also contains Q.

Now consider two points q_j and $q_{j'}$ on the upper hull of Q, each inside a disc of radius k centered at p_i and $p_{i'}$, respectively. Since Q is convex, the edges of Q between q_j and $q_{j'}$ must lie on or above the line $\overleftrightarrow{q_j q_{j'}}$, and also remain below the boundary of P since P contains Q. Therefore, for any point on Q in this range, there is a point in P at most distance k above it. If $\overleftrightarrow{q_j q_{j'}}$ is parallel to the x-axis, then it is possible that the maximum distances are equal: $\max_{p \in P} \min_{q \in Q} \operatorname{dist}(p, q) = \max_{q \in Q} \min_{p \in P} \operatorname{dist}(p, q)$. If $\overleftrightarrow{q_j q_{j'}}$ is not horizontal, then the distance $\max_{p \in P} \min_{q \in Q} \operatorname{dist}(p, q)$ is maximized at one of the vertices of P, and so on this interval $H(P, Q) = \max_{p \in CH(P)} \min_{q \in Q} \operatorname{dist}(p, q)$.

The lemma follows, because this argument may be applied to any consecutive pair of vertices of the convex hull of P, and points of Q may be chosen so that the entire polygon is considered. □

The distance $H(P, Q)$ is determined by the vertices of P that lie on the convex hull of P, but all vertices and edges of P must be considered to determine whether Q is contained in P. Therefore, the k-bounded Hausdorff Core problem may be redefined as follows: we consider discs of radius k centered at vertices $CH(P)_V$ and ask whether there exists a convex polygon Q such that it is covered by P and intersects all such discs. Let $C(p, k)$ denote a disc of radius k centered at p. We refine the k-Hausdorff Core problem with the following corollary:

Corollary 2. *Given a simple polygon P and a convex polygon Q contained in P,*

$$H(P, Q) \leq k \Leftrightarrow \forall p \in CH(P)_V, C(p, k) \cap Q \neq \emptyset.$$

Finally, for finding a Hausdorff Core we wish to identify a point (any point) q contained in Q. For any disc $C(p, k)$ covering q, p is within distance k of Q, and so the search for a k-bounded Hausdorff Core is simplified by ignoring all such discs and searching for a solution which touches the boundaries of all remaining discs.

Lemma 3. *Given a simple polygon P and a convex polygon Q that is a Hausdorff Core of P, it is always possible to expand Q so that it contains at least one point in the set P_V (the set of vertices of P) while remaining a Hausdorff Core of P.*

Proof. It is possible for a solution to be optimal and not contain a point from the set P_V, but we show that any such solution may be expanded to include a vertex of P. An example of such a scenario is shown in Figure 7.

Suppose there exists a polygon Q which has $H(P, Q) = k$ where $P_V \cap Q = \emptyset$. We can always expand Q to create a new polygon Q', where $Q \subset Q'$, and so $H(P, Q') \leq H(P, Q)$. Note that since we assumed Q is an optimal solution, in fact $H(P, Q') = H(P, Q)$. One such algorithm is as follows:

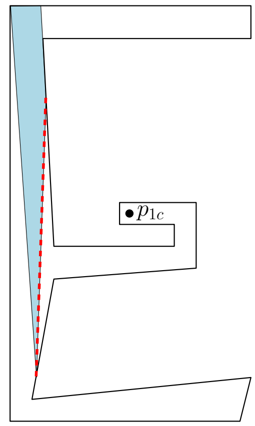


Figure 7: The thick dashed line shows an optimal Hausdorff Core for this polygon. This line may grow in a number of ways into a convex polygon which remains contained inside the original polygon, while including a vertex of P on the boundary. The shaded polygon is one such possibility. Interestingly, in this polygon the Euclidean 1-center p_{1c} is contained in the polygon even though it is not contained in any Hausdorff Core solution.

1. Choose an arbitrary edge $e_Q = (v_i, v_j)$ of Q.

2. Choose one of the vertices of e_Q arbitrarily. Say without loss of generality that v_i is chosen. Let $e'_Q = (v_i, v_{j'})$ be the other edge of Q incident to v_i (it is possible that $e_Q = e'_Q$). Expand e_Q by moving v_i away from v_j along the unique line defined by e_Q until either:

   (a) one of the edges of Q touches a vertex of P (Figure 8a);

   (b) moving v_i further would cause Q to lose convexity (Figure 8b); or

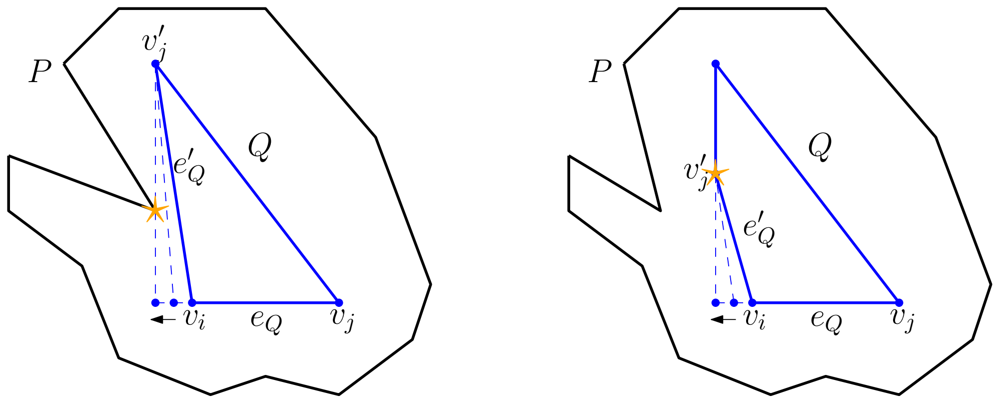
   (c) v_i encounters an edge of P, which we call e_P (Figure 8c).

3. If case 2(a) occurs, we are done. If case 2(b) occurs, we merge e'_Q with the other edge incident to $v_{j'}$, and then return to step 2. Note that this case may occur $O(|Q|)$ times before the algorithm terminates. If we arrive at case 2(c), we introduce a new vertex v' on Q at v_i, and edges $e_1 = (v', v_i)$ and $e_2 = (v', v'_j)$ (Figure 8d). Note that e_P and e_1 are collinear and e_1 initially has zero length. We move v' along e_P (in a direction which keeps Q simple), again according to the rules in the previous step. When condition 2(a) is met, one of the vertices of P must be incident to Q, and the algorithm terminates.

□

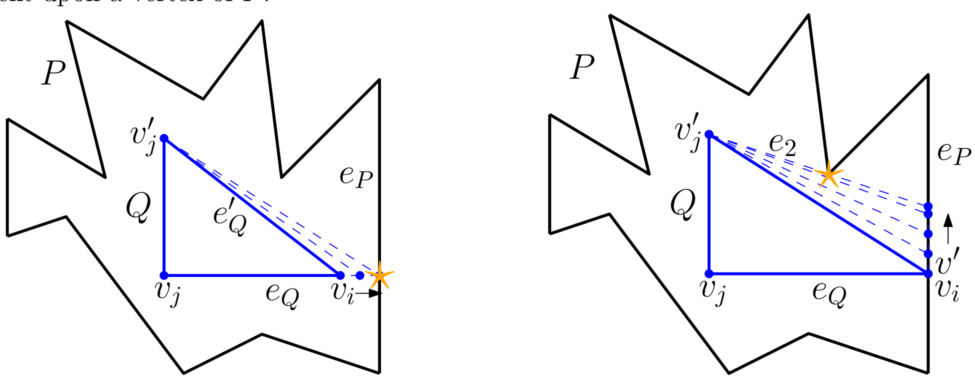
3.3 Algorithmic Challenges of the Hausdorff Core Problem

The k-bounded Hausdorff Core problem consists of determining whether we can draw a polygon Q with one vertex in or on each disc such that the vertices in each successive pair are able to see each other around the obstructions formed by the input polygon P. For any fixed choice of the obstructing vertices, this consists of a system of quadratic constraints



(a) In this example (case 2(a)), moving v_i away from v_j has resulted in e_Q' becoming incident upon a vertex of P.

(b) In this example (case 2(b)), v_i was moved until two edges of Q became collinear.



(c) In case 3(b), v_i encounters an edge of P but Q is not yet incident upon a vertex of P_V.

(d) The new vertex v' is added to Q, and the algorithm proceeds until arriving at case 2(a).

Figure 8: The cases of Lemma 3 are illustrated with examples. The stars indicate locations where interesting events have taken place.

of the form "variable point in circle" and "two variable points collinear with one constant point (an obstructing vertex)." For the optimization version we need only make the circle radius a variable and minimize that.

Solving systems that include quadratic constraints is in general NP-hard; we can easily reduce from 0–1 programming by means of constraints of the form $x(x-1) = 0$. Nonetheless, some kinds of quadratic constraints can be addressed by known efficient algorithms. Lobo et al. [LVBL98] describe many applications for second-order cone programming, a special case of semidefinite programming. The "point in circle" constraints of our problem can be easily expressed as second-order cone constraints, so we might hope that our problem could be expressed as a second-order cone program and solved by their efficient interior point method.

However, the "two variable points collinear with one constant point" constraints

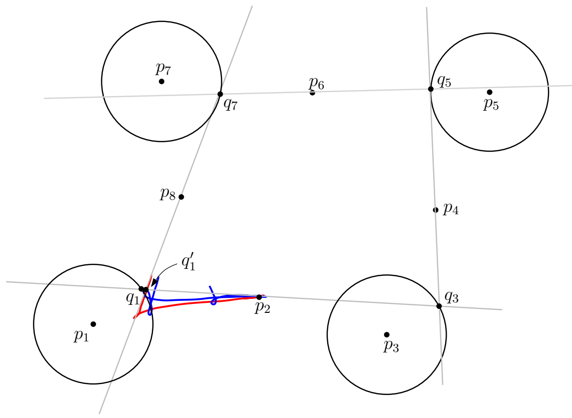


Figure 9: There can be two disconnected solution intervals for the Hausdorff Core problem. The points p_i are vertices of the polygon P, and the points q_j are points that are being selected on discs as the vertices of the solution polygon Q. The fat arcs trace the position of q_1' at the intersection point of the lines $\overleftrightarrow{q_1p_2}$ and $\overleftrightarrow{q_7p_8}$ as q_1 is moved along the boundary of the disc centered on p_1.

are not so easy to handle. With (x_1, y_1) and (x_2, y_2) the variable points and $(x_\mathrm{C}, y_\mathrm{C})$ the constant point, we have the following:

$$\frac{y_1 - y_\mathrm{C}}{x_1 - x_\mathrm{C}} = \frac{y_2 - y_\mathrm{C}}{x_2 - x_\mathrm{C}} \tag{2}$$

$$x_2y_1 - x_2y_\mathrm{C} - x_\mathrm{C}y_1 = x_1y_2 - x_1y_\mathrm{C} - x_\mathrm{C}y_2 \tag{3}$$

This constraint is hyperbolic because of its cross-product terms. The techniques of Lobo et al. [LVBL98] can be applied to some hyperbolic constraints, subject to limitations whose basic purpose is to keep the optimization region convex.

But as shown in Figure 9, it is possible for our problem to have two disconnected sets of solutions, even with as few as four discs. For a point q_1 on the first disc, we can trace the polygon through the constant point p_2 to the intersection point of that edge with the second disc at q_3, then through the constant point p_4 and so on around to p_8. The lines $\overleftrightarrow{q_1p_2}$ and $\overleftrightarrow{q_7p_8}$ intersect at q_1', which is our choice for one vertex of the solution polygon, the others being q_3, q_5, and q_7. If q_1' is on the disc, we have a feasible solution (note that q_1 is not needed as a vertex for any solution, since if q_1' is in a feasible position then q_1 lies on the edge between q_1' and q_3). The heavy curves in Figure 9 show the locus of q_1' for different choices of q_1. The set of solutions to the problem as shown is disjoint, corresponding to a slice (for a constant value of the circle-radius variable) through a non-convex optimization region. As a result, neither second-order cone programming nor any other convex optimization technique is immediately applicable.

4 Hausdorff Core Algorithms

In this section we outline approximation algorithms for solving the general Hausdorff Core problem by manipulating discs centered on selected vertices of P. We begin by describing approximate decision and optimization algorithms, and then we describe how the approximate k-bounded Hausdorff Core algorithm may be modified to obtain an approximation scheme for the decision problem.

Let d_{rad} be the distance from the constrained 1-center p_{1c} to the most distant vertex in P: $d_{\text{rad}} = \max_{p \in P} \text{dist}(p_{1c}, p)$. To simplify the discussion, we will scale the problem so that $d_{\text{rad}} = 1$, and so the algorithm finds a $(k+\varepsilon)$-bounded Hausdorff Core for some constant ε. By Corollary 2 (page 10), Invariant 1 (below) implies that there exists a k-bounded Hausdorff Core for P, i.e., given P there exists a convex polygon Q contained in P with $H(P,Q) = k$:

Invariant 1. Given a simple polygon P with convex hull $CH(P)$ and a value $k \in \mathbb{R}$, there exists a set of points $\{q_1, \ldots, q_{n'}\} \subset Q$, where n' is the number vertices of $CH(P)$, such that $\forall i,\ q_i \in C(p_i, k)$ (recall that $C(p_i, k)$ is a disc of radius k centered at p_i) and $\forall i, j,\ i \neq j,\ \overline{q_i q_j}$ does not cross outside P.

To find an approximate solution to the Hausdorff Core problem, we repeatedly apply the k-bounded Hausdorff Core solution to selected values of k. To provide insight into the ideas used in our solution, we sketch a simplified version in Algorithm 1, and we illustrate an example of the operation of the algorithm in Figure 10. In this algorithm, we use Lemma 3, which states that an optimal Hausdorff Core Q of P may always cover at least one vertex of P. We call this vertex of P covered by Q the point q_p, and we try all possible values of q_p in the algorithm. The algorithm operates by placing disc centers on the vertices of the convex hull of P and shrinking their radii uniformly as long as there exists a k-bounded Hausdorff Core which pierces all discs. We simplify this test by considering only those discs that do not cover q_p and checking for intersection between Q and the boundary of each disc. It is safe to ignore discs that cover q_p because q_p is covered by Q, and so any such disc must intersect Q.

Inside the for loop of Algorithm 1, a convex polygon Q is maintained where Q is a k-bounded Hausdorff Core and k is the radius of the discs. If $C(p_i, k)$ touches Q, then $d(p_i, Q) \leq k$. In line 6, we ensure that if a vertex $v \in CH(P)_V$ (recall that $CH(P)_V$ is the set of vertices on the convex hull of P) does not have a disc, then $\text{dist}(v, q_p) \leq k$. Since $q_p \in Q$, it follows that $\text{dist}(v, Q) \leq k$ for all such vertices. It remains to be shown that there does not exist a convex polygon Q' such that $\text{dist}(p, Q') \leq k'$, where $k' < k$. This cannot be the case, for if the discs were shrunk any further, no convex polygon could intersect some pair of the discs by Invariant 1. Therefore, the polygon would necessarily be of distance $\text{dist}(p, q') > k'$ for some vertex p and any point $q' \in Q$. Each iteration of the for loop considers a point q_p, and the body of the loop computes a k-bounded Hausdorff Core Q so that k is minimized under the constraint that $q_p \in Q$.

The optimality of the algorithm is guaranteed since the for loop exhaustively explores all possibilities for the point q_p which is known to be contained in a solution Q. By Lemma 3, we know that at least one such point q_p is contained in an optimal solution. By trying all possibilities, we ensure that a globally optimal solution is obtained, and so Q is a Hausdorff Core of P.

Our formal approximation algorithm, described in Section 4.2, operates in a similar way to Algorithm 1 except that we use a binary search on the values of possible k-

Algorithm 1 HCORE(P)

Input: A simple polygon P.
Output: Q, a Hausdorff Core of P.
$Q = \emptyset, k_{\min} = \infty$
for each $q_p \in P_V$ **do**
  Begin with discs of radius k_0 centered on the vertices $v \in CH(P)_V$, where $k_0 = 1$ (recall that $d_{\text{rad}} = 1$).
  Any disc centered at a vertex v where $\text{dist}(q_p, v) < k_0$ covers q_p; such discs are ignored for now.
  Reduce the radii such that at time $t_i \in [0, 1]$, each disc has radius $k(t_i) = 1 - t_i$. Let $Q(t_i)$ be a $k(t_i)$-bounded Hausdorff Core covering q_p at time t_i, if it exists (we discuss how this may be done approximately in Section 4.1). The radius is reduced until one of the following two events occurs:

  1. $k(t_i) = \text{dist}(q_p, v_n)$, where v_n is the farthest vertex from q_p that is not the center of a disc. Add a disc centered at v_n with radius $k(t_i)$, and continue reducing k.

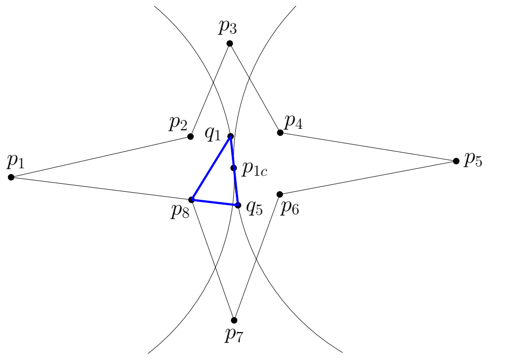
  2. A further reduction of $k(t_i)$ would prevent visibility in P between two discs. We stop reducing k, and if $k(t_i) < k_{\min}$, then set $Q = Q(t_i)$ and $k_{\min} = k(t_i)$.

end for
return Q

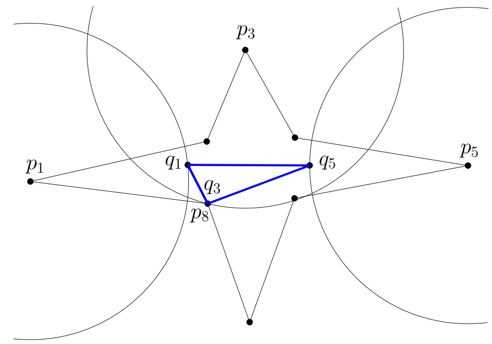
bounded Hausdorff Core solutions to obtain the approximation, rather than reducing the value continuously. We find the Euclidean 1-center p_{1c} using the technique of Bose and Toussaint [BT96] described in Section 2.3; there may be multiple such vertices, but we can choose one arbitrarily.

4.1 Discretization of the Problem

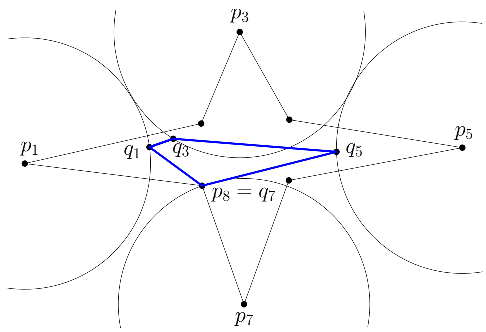
In this section, we discuss an approximate k-bounded Hausdorff Core problem, where we are given a distance k, some $\varepsilon > 0$, and a polygon P and we wish to determine whether there exists a polygon Q' contained in P so that $H(P, Q') \leq k + \varepsilon$. We extend the algorithm to provide a polygon as a certificate in the case of a positive result. The approach seeks to enlarge discs by an additive factor $\varepsilon/2$, and to determine whether there exists a solution for these expanded discs. Here ε is the fraction of d_{rad} that we wish to use as a bound on the approximation, where d_{rad} is the distance from the constrained 1-center p_{1c} to the most distant vertex in P. We grow discs by $\varepsilon/2$ rather than ε because there is some additional error that comes into the algorithm, and we wish to have a final additive approximation factor of ε. We scale the input so that $d_{\text{rad}} = 1$ to simplify the analysis. Note that this method of approximation maintains a scale-invariant approximation factor, and the size of the approximation factor for a given P is an additive constant independent of Q and the magnitude of k. We still require that the approximate solution Q' must not cross outside P, and that Invariant 1 holds.



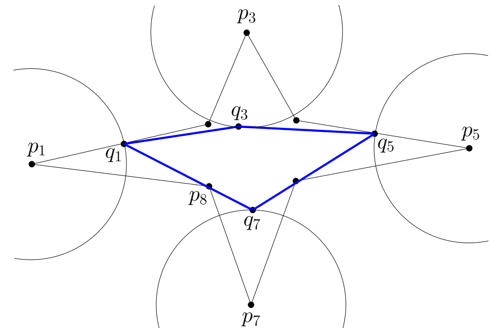
(a) Two discs are centered on p_1 and p_5, which are the critical points for determining the position of the 1-center p_{1c}. Let $q_p = p_8$.



(b) We have shrunk the discs to have radius k', and $\text{dist}(p_3, p_8) = k'$, so we add a new disc $C(p_3, k')$ to the set. The fat lines indicate a set of lines of strong visibility between the discs, and so there exists a k'-bounded Hausdorff Core (i.e. $H(P, Q) = k'$).



(c) Another disc is added centered at point p_7, so now we have four discs $C(p_i, \text{dist}(p_7, p_8))$, for each $i \in \{1, 3, 5, 7\}$.



(d) We cannot shrink the discs any further, or Invariant 1 would be violated. Therefore, a solution Q can be composed from the fat line segments such that $H(P, Q) = k$, where k is the radius of the discs. Note that all vertices of P are within distance k of Q, and vice versa.

Figure 10: Finding a Hausdorff Core by shrinking discs centered on the vertices of P, as discussed in Algorithm 1. We are using $q_p = p_8$ in this example.

We study two possible cases for a k-bounded Hausdorff Core Q: either Q may be approximated by a line segment, or the minimum angle in Q is at least $\varepsilon/4$.

4.1.1 A Line Segment Solution

We first determine whether any k-bounded Hausdorff Core Q can be approximated by a single line segment. We consider an arc segment of radius 2 (i.e. the maximum diameter of P) and arc length $\varepsilon/2$, as shown in Figure 11. The interior angle of the circular segment C_φ formed by this arc is $\varphi = \varepsilon/4$. If an interior angle of Q is less than or equal to φ, then

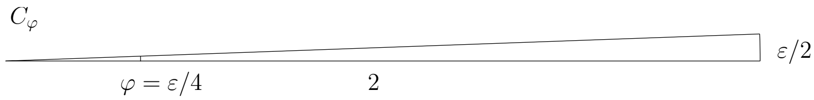


Figure 11: Determining whether there exists a straight line segment that would serve as a Hausdorff Core solution. Consider a circular segment C_φ of radius 2, arc length $\varepsilon/2$, and with an interior angle of $\varphi = \varepsilon/4$. If Q can be covered by C_φ, then a straight line segment Q_ℓ exists such that $H(Q, Q_\ell) < \varepsilon/2$.

Q may be fully covered by C_φ since Q is convex. In this case, there exists a line segment Q_ℓ which approximates Q where $H(Q, Q_\ell) < \varepsilon/2$.

We now describe how to determine such a line segment Q_ℓ, assuming there exists some C_φ that covers Q. First, we enlarge all relevant discs (i.e. those not covering q_p) by $\varepsilon/2$, so that they have radius $k_g = k + \varepsilon/2$. Since Q is convex, this operation means that any line segment which approximates Q will now intersect at least one arc from each disc if a solution exists where $H(P, Q) \leq k$. By Lemma 3, we know that $q_p \in P_V$ is contained in Q. Therefore, we attempt to find a line intersecting a point q_p and a point on each disc of radius k_g for each q_p. For a selected q_p, we build an interval graph in the range $[0 \ldots \pi]$. For each disc $C(p_i, k_g)$, if a line at angle $\theta \bmod \pi$ from an arbitrary reference line intersects a segment of $C(p_i, k_g)$ contained in P before intersecting P itself, then $C(p_i, k_g)$ covers θ in the interval graph. Note that since P may intersect a disc, any disc may have $O(n)$ intervals in the interval graph. If there is a non-zero intersection between all discs in the interval graph at $\theta^\star$, then the solution is a line segment Q_ℓ at angle $\theta^\star$ to the reference line, intersecting q_p with endpoints on the last discs intersected by Q_ℓ. Therefore, if there exists a solution $H(P, Q) \leq k$ where Q can be approximated by a line segment Q_ℓ with $H(Q, Q_\ell) < \varepsilon/2$, then we will find Q_ℓ.

4.1.2 A Polygonal Solution

If we have not found a line segment solution Q_ℓ, we know that all interior angles of some k-bounded Hausdorff Core Q are greater than $\varphi(= \varepsilon/4)$ if a solution exists. If we divide the boundaries of the expanded discs of radius $k_g = k + \varepsilon/2$ into $16\pi^2(k + \varepsilon/2)/\varepsilon^2$ equal intervals, then at least one such interval on each disc is fully contained in an optimal solution Q regardless of where the intervals are placed on the discs. In this section, we describe a dynamic programming approach for approximating the k-bounded Hausdorff Core problem.

Lemma 4. *If there exists a k-bounded Hausdorff Core Q for P, then by Invariant 1, Q pierces all discs of radius k centered on the vertices of the convex hull of P. If discs of radius $k + \varepsilon/2$ are placed on all the same vertices and the minimum angle of Q is at least φ, then the expanded discs may be divided into $O(\varepsilon^{-2})$ disjoint intervals so that at least one interval on every expanded disc is on the interior of Q.*

Proof. The smallest arc length of an expanded disc that may be covered by Q is realized when the bisector of the angle at a vertex q_1 is collinear with the line formed by q_1 and the center of the disc p_1 (see Figure 12ba). The minimum angle at a vertex in Q is at least

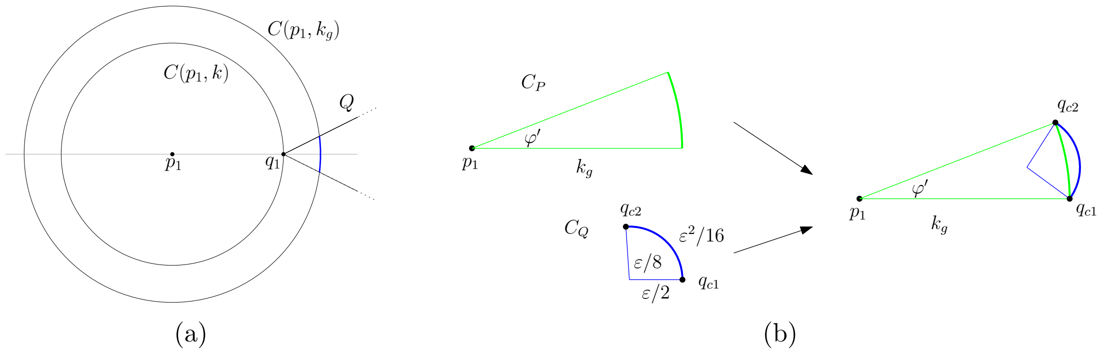


Figure 12: To find the minimum-length interval for the discretization of the expanded discs, we need to ensure that at least one full interval is always covered by Q. We know the minimum angle at any vertex of Q is at least φ, since there was no single line segment that could approximate Q. (a) The minimum arc length of $C(p_1, k_g)$ spanned by Q is realized when the bisector of the edges incident to q_1 is collinear with the line $\overleftrightarrow{p_1q_1}$. (b) Using the arc segments C_P and C_Q to determine a lower bound on the angle φ'.

$\varphi = \varepsilon/4$, and we want to determine an angle φ' for a circular segment C_P of radius k_g centered at p_1 (see Figure 12b b) so that two interior disjoint such circular segments may always be covered by Q. A circular segment C_Q of radius $\varepsilon/2$ and angle $\varepsilon/8$ (i.e. $\varphi/2$) has arc length $\varepsilon^2/16$, and two interior disjoint such segments may be covered by $Q \cap C(p_1, k_g)$. Therefore, given points q_{c1} and q_{c2} as the endpoints of the arc on C_Q, we may place these points on C_P to determine a lower bound on φ'. The arc length of C_P between q_{c1} and q_{c2} is at least $2/\pi$ times that of C_Q on these points. Since the arc length of C_Q is $\varepsilon^2/16$, the arc length of C_P is at least $\varepsilon^2/8\pi$. The interior angle φ' of C_P is given by $\varphi' = \varepsilon^2/8\pi k_g$. The number of such circular segments in $C(p_1, k_g)$ is $16\pi^2 k_g/\varepsilon^2$. Since k_g is at most 1, the number of segments is in $O(\varepsilon^{-2})$. □

Each disc is divided in $O(\epsilon^{-2})$ intervals. Consider the following observations pertaining to Q and Q', where Q' is a polygon approximating a k-bounded Hausdorff Core Q:

- $\exists Q \Rightarrow \exists Q'$, $\neg\exists Q' \Rightarrow \neg\exists Q$. The intervals are defined so that at least one interval from each disc will be contained in Q' if Q exists.
- $\exists Q' \not\Rightarrow \exists Q$. The existence of Q' does not imply the existence of Q because the optimal Hausdorff Core solution may have distance $H(P, Q) = k + \nu$, where $\nu < \varepsilon/2$.

We choose a representative point in each interval for every relevant disc on P; any arbitrary such point suffices. We write $q_{i,j}$ to represent a point chosen in the jth interval on disc c_i. As with Algorithm 1, we will explore every choice of $q_p \in P_V$ and we need only consider discs not covering q_p ($q_p \in P_V$ is a vertex of P that we are assuming is part of the solution, see Lemma 3); for any iteration say there are $n' \leq n$ such discs. Relabel the set of discs $C = \{c_1, \ldots, c_{n'}\}$ so that the discs are numbered in clockwise order relative

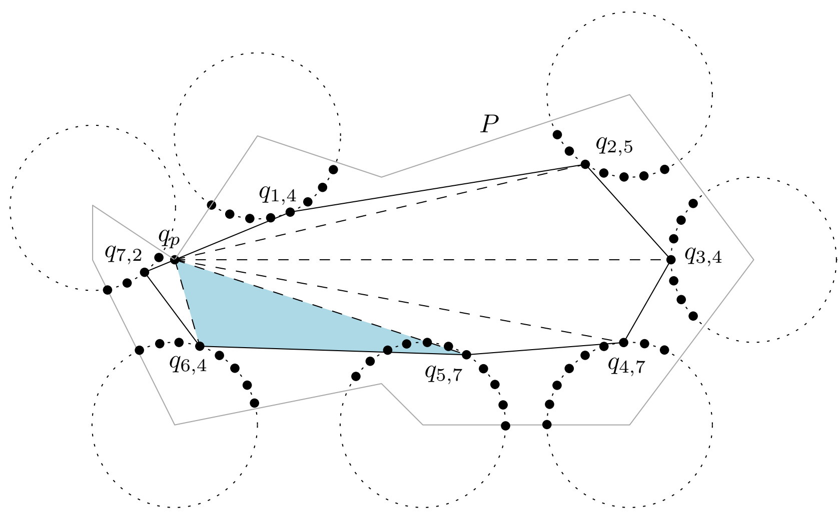


Figure 13: The convex polygon that includes points $q_{5,7}$ and $q_{6,4}$ (for example) is built by combining the shaded triangle defined by these points and q_p with a polygon containing q_p, $q_{5,7}$ and a point on c_4, $q_{4,7}$ in this case. A solution is a convex polygon composed of triangles arranged as a fan, with q_p as the vertex shared by all triangles.

to the vertex q_p. Note that the ordering is unique and Q' intersects the discs in order since all discs are centered on vertices of the convex hull of P. A solution has the form $Q' = \{q_p, q_{1,j}, \ldots, q_{n',j'}\}$, where $q_{i,j}$ is the point that was chosen on the jth interval of $C(p_i, k_g)$.

Any convex polygon may be triangulated as a fan so that every triangle shares one vertex; consider a triangulation of Q' where every triangle shares vertex q_p and triangle T_i is defined by the vertices q_p, $q_{i,a}$ and $q_{i+1,b}$. The dynamic programming solution proceeds by adding triangles $T_1, \ldots, T_{n'-1}$ to the fan iteratively. The two essential properties that Q' must possess are that it is contained in P and that it is convex. Suppose we have a valid partial solution Q_i (containing triangles $T_1, \ldots, T_i$) to which we wish to add one more triangle to create a new polygon Q_{i+1}. To check that Q_{i+1} is contained in P, we perform ray shooting queries with $\overrightarrow{q_{i+1,b}q_p}$ and $\overrightarrow{q_{i+1,b}q_{i,a}}$ on P to determine whether the corresponding edges would intersect P. To check for convexity, we compare the vertex $q_{i+1,b}$ with the lines $\overline{q_p q_{1,c}}$ and $\overline{q_{i-1,d}q_{i,a}}$, for some points $q_{1,c}$ and $q_{i-1,d}$ (convexity may be determined by checking whether a point is to the left or right of a line in $O(1)$ time, e.g. [O'R98, p.28 & §3.5]).

We write an index value as a pair $\langle i, a\rangle$ to preserve the information that this is a point in the ath interval of disc c_i, with the intention that this notation simplifies the discussion. In practice, this pair could be replaced with an integer. We keep partial solutions in a table A, in which an entry $A[\langle j, b\rangle, \langle j+1, c\rangle]$ represents a convex polygon that touches points q_p, $q_{j,b}$, and $q_{j+1,c}$, for some $j \geq 1$, and includes points on all discs $c_1, ..., c_{j+1}$. If there exists such a polygon, then the table entry $A[\langle j, b\rangle, \langle j+1, c\rangle]$ stores:

1. the pair $(\langle j-1, d\rangle, \langle j, b\rangle)$, which is the index of $A[\langle j-1, d\rangle, \langle j, b\rangle]$, where the polygon

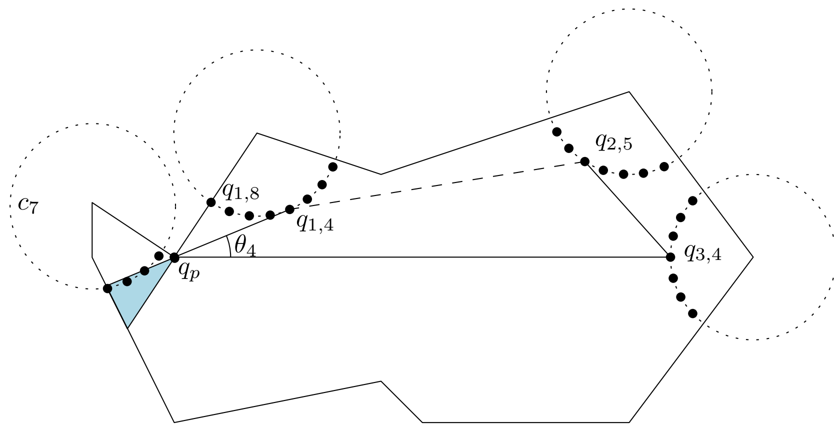


Figure 14: Consider the partial solution composed of the point q_p and points on discs c_1, c_2, c_3, and let θ_a be the angle $\angle q_{1,a}q_pq_{3,d}$ for $a \in \{1, \ldots, 8\}$. Point $q_{1,4}$ minimizes the angle θ_a, which consequently maximizes the area of the shaded region. No solution exists using point $q_{1,8}$ and a point in the shaded region.

represented by $A[\langle j, b\rangle, \langle j+1, c\rangle]$ is equal to the union of a polygon represented by $A[\langle j-1, d\rangle, \langle j, b\rangle]$ and the triangle with vertices $q_p, q_{j,b}, q_{j+1,c}$ (see Figure 13); and

2. the point $q_{1,a}(\in c_1)$ used in the polygon, where the angle $\angle q_{1,a}q_pq_{j+1,c}$ is minimized over all choices of a that are feasible for this entry. Note that choosing a point $q_{1,a}$ to minimize this angle is equivalent to minimizing $\angle q_{1,a}q_pq_j$ for any choice of q_j since the solution must be convex (see Figure 14).

The base cases consist of the entries $A[\langle 1, a\rangle, \langle 2, b\rangle]$ for all values of a and b. We set $A[\langle 1, a\rangle, \langle 2, b\rangle] = \{(), q_{1,a}\}$ if there is visibility in P between each pair of points in $\{q_p, q_{1,a}, q_{2,b}\}$, and if these vertices occur in clockwise order on the triangle defined by the points. Otherwise, there is no valid solution for these points, in which case we set $A[\langle 1, a\rangle, \langle 2, b\rangle] = \emptyset$. For each subproblem in the dynamic program, the objective is to obtain valid subpolygons where the edge $(q_pq_{1,a})$ is rotated clockwise as much as possible for all choices of a, as this maximizes the size of the region where the next point after the subproblem may lie while maintaining convexity, as illustrated in Figure 14.

The dynamic programming algorithm is described in detail in Algorithm 2. We compute the base cases in lines 5 to 11. The loop in lines 13–22 successively adds triangles to existing partial solutions: in the ith iteration of the main loop, for each triple of points $(q_p, q_{i-1,b}, q_{i,c})$ we check if it is possible to construct a convex polygon that touches these points by adding the corresponding triangle solution that uses the points $(q_p, q_{i-2,d}, q_{i-1,b})$ over all possible values of d. Note that though there may be many solutions to form the subproblem, it is enough to keep that minimizes the angle θ_a described in Figure 14. At the end of this loop, if there exists any non-empty entry $A[\langle n'-1, b\rangle, \langle n', c\rangle]$ then we have found a solution. If this is the case, we then trace table A in lines 24 to 28 to find the vertices of a polygon Q'.

The dynamic programming algorithm must be run iteratively for each $q_p \in P_V$, using only discs centered on vertices $v \in P_V$ where $\text{dist}(v, q_p) \geq k$. If no solution Q' is found for any q_p, then there is no solution where $H(P,Q) = k$.

Algorithm 2 FindConvexPolygon(C, P)

Input: Discs $C = \{c_1, \ldots, c_{n'}\}$ with $c_i = C(p_i, k_g)$ and $q_p \notin c_i$, and a simple polygon P.
Output: Q', a convex polygon approximating a k-bounded Hausdorff Core of P, or *fail* if no such approximation exists.
For each interval on each disc in P, choose a point. Let $q_{i,j}$ be the jth point on disc c_i.
{Base cases}
for each $q_{1,a}$ and $q_{2,b}$ **do**
  if $\overline{q_p q_{1,a}} \subset P$ **and** $\overline{q_{1,a} q_{2,b}} \subset P$ **and** $\overline{q_p q_{2,b}} \subset P$ **and** q_p, $q_{1,a}$ and $q_{2,b}$ appear in clockwise order on the triangle they define **then**
    $A[\langle 1, a\rangle, \langle 2, b\rangle] = \{(), q_{1,a}\}$
  else
    $A[\langle 1, a\rangle, \langle 2, b\rangle] = \emptyset$
  end if
end for
{Build partial solutions}
for $i = 3$ to n' **do**
  for each triple of points $(q_p, q_{i-1,b}, q_{i,c})$ **do**
    if $\overline{q_p q_{i,c}} \subset P$ **and** $\overline{q_{i-1,b} q_{i,c}} \subset P$ **then**
      Over all values of d in $A[\langle i-2, d\rangle, \langle i-1, b\rangle]$, find an entry with a $q_{1,a}$ so that θ_a is minimal and the points $q_p, q_{1,a}, q_{i-2,d}, q_{i-1,b}$, and $q_{i,c}$ are in convex position.
    end if
    if no valid entry was found **then**
      $A[\langle i-1, b\rangle, \langle i, c\rangle] = \emptyset$
    end if
  end for
end for
{Return one polygon, if found}
if $\exists b, c$ such that $A[\langle n'-1, b\rangle, \langle n', c\rangle] \neq \emptyset$ **then**
  Backtrack on solution stored in $A[\langle n'-1, b\rangle, \langle n', c\rangle]$ to build Q'
else
  return fail
end if

4.2 The Minimization Problem

If a Hausdorff Core Q of a given polygon P has Hausdorff distance $H(P,Q) = k_{\min}$, our algorithm finds an approximate solution Q' such that $H(P,Q') < k_{\min} + \varepsilon$. To determine a value of k' where $k' \leq k_{\min} + \varepsilon$ ($k_{\min}$ is unknown), it suffices to perform a binary search over possible values for k' in the range of $[0 \ldots 1]$, executing the approximation algorithm for the k-bounded Hausdorff Core problem at each iteration. At the ith iteration of the algorithm, let the current radius be k_i. If the algorithm finds a k_i-bounded Hausdorff Core solution Q_i, we shrink the discs and use $k_{i+1} = k_i - 1/2^i$. If the algorithm fails to find a solution, we use $k_{i+1} = k_i + 1/2^i$. Initially, $k_0 = 1$. The stopping condition for the binary search is met when we find a k-bounded Hausdorff Core, and the algorithm fails to find a $(k - \varepsilon/2)$-bounded Hausdorff Core. Thus, the approximation algorithm requires $O(\log(\varepsilon^{-1}))$ iterations of the k-bounded Hausdorff Core algorithm to find a solution. For the latter algorithm, we showed that $H(Q, Q') < \varepsilon/2$, if Q exists. For the Hausdorff Core approximation algorithm, an optimal solution has a Hausdorff distance to P of up to $\varepsilon/2$ less than a solution returned by the approximation algorithm. Therefore, the minimization approximation algorithm returns a solution Q' where $H(P, Q') < k_{\min} + \varepsilon$, where $k_{\min}$ is the Hausdorff distance of an optimal solution.

4.3 Running Time and Space Requirements

Theorem 5. *There exists an approximation algorithm for the general Hausdorff Core problem on simple polygons with $O(\log(\varepsilon^{-1})(n^3 + n^2\varepsilon^{-4}(\log n + \varepsilon^{-2})))$ running time. Given an input polygon P, the algorithm computes a convex polygon Q, where $Q \subseteq P$, and with Hausdorff distance $H(P, Q) < k + d_{rad} \cdot \varepsilon$, where k is the value of the optimal solution, and d_{rad} is the distance from a constrained 1-center of P to the most distant vertex in P.*

We begin by analyzing the space requirements and running time of the approximate k-bounded Hausdorff Core algorithm. We compute the 1-center using the technique of Bose and Toussaint [BT96], which takes $O(n^2)$ time (see Section 2.3). The single line solution tests a line against $O(n)$ discs, each of which may have $O(n)$ intervals in the interval graph. This procedure is repeated $O(n)$ times, so it takes $O(n^3)$ time in total.

In the dynamic programming algorithm, there are $O(n)$ discs and the number of intervals on each disc is bounded by $O(\varepsilon^{-2})$. A point is chosen in each interval, which may be done by incrementing by $\varphi'(= \varepsilon^2/8\pi(k + \varepsilon/2))$ around the boundary of the disc, and performing a ray shooting query on P to determine whether P contains the point. This can be done in $O(\log n)$ time per point [CEG+94], and there are $O(n)$ discs with $O(\varepsilon^{-2})$ points per disc, so choosing all of the points may be done in $O(n\varepsilon^{-2}\log n)$ time. For each disc $c_i \in \{c_2, \ldots, c_{n'}\}$, we store in the dynamic programming table an entry for every combination of points with one point on each of c_1 and c_i. Since there are $O(\varepsilon^{-4})$ possible combinations and each entry takes constant space, the space for the dynamic programming table is $O(n\varepsilon^{-4})$. To fill in a table entry, we check for containment in P and then possibly $O(\varepsilon^{-2})$ subproblems are inspected and checked for convexity with the new points. Checking for containment of an edge can performed using a ray shooting query on P in $O(\log n)$ time [CEG+94], while the convexity test may be done in constant time for each entry in the table, so each entry may take $O(\log n + \varepsilon^{-2})$ time to fill. Therefore, filling in the table takes $O(n\varepsilon^{-4}(\log n + \varepsilon^{-2}))$ time. Finding the actual vertices of the approximate polygon by tracing the table backwards takes $O(n)$ time, and thus the dynamic programming algorithm takes $O(n\varepsilon^{-4}(\log n + \varepsilon^{-2}))$ time.

The algorithm may require $O(n)$ iterations to test each value of q_p, so the approximate decision algorithm requires $O(n^3 + n^2\varepsilon^{-4}(\log n + \varepsilon^{-2}))$ time. Finally, the minimization version of the algorithm performs $O(\log(\varepsilon^{-1}))$ iterations of the approximate decision algorithm, so the complete algorithm requires $O(\log(\varepsilon^{-1})(n^3 + n^2\varepsilon^{-4}(\log n + \varepsilon^{-2})))$ time to find a Hausdorff Core with Hausdorff distance $k + \varepsilon$, where k is the value of an optimal solution.

4.4 An Approximation Scheme for the k-bounded Hausdorff Core Problem

Modifying the decision algorithm of Section 4.1 to admit an approximation scheme rather than an algorithm with an additive approximation factor is straightforward. In this context, we define an approximation scheme for the decision problem with parameters k and ε to be an algorithm which returns false if there exists no solution for k and true if there exists a solution for $(1+\varepsilon)k$, while the algorithm cannot provide a definitive answer about the existence of solutions in the range $(k, (1+\varepsilon)k)$ (in our case, we may get a false positive result). We begin by determining whether there exists a circular segment C_φ which covers a Hausdorff Core solution for P, but now we use arc length $k\varepsilon/2$. This way, our algorithm will find a line segment which is a Hausdorff Core solution with Hausdorff distance $(1+\varepsilon)k$ if one exists. The dynamic program requires $4\pi^2(1+\varepsilon)/\varepsilon^2 \in O(\varepsilon^{-2})$ equal intervals so that at least one interval is contained in any solution when the discs are enlarged to have radius $(1 + \varepsilon/2)k$. The running time is asymptotically the same as the previous approximation algorithm. However, we are not able to perform a binary search to find an approximate Hausdorff Core in the same way as before, because it could have an arbitrarily large running time relative to the sizes of the inputs. Since the running time of the decision algorithm remains $O(n^3 + n^2\varepsilon^{-4}(\log n + \varepsilon^{-2}))$, this is an approximation scheme for the k-bounded Hausdorff Core problem.

5 Conclusions and Future Work

We have described an algorithm which computes an approximate Hausdorff Core Q of a simple input polygon P with Hausdorff distance $H(P,Q) < k_{\min} + \varepsilon$, where $k_{\min}$ is the value of an optimal solution, and ε is a fraction of d_{rad} (the distance from a constrained 1-center to the most distant vertex in P). The running time of the algorithm is $O(\log(\varepsilon^{-1})(n^3 + n^2\varepsilon^{-4}(\log n + \varepsilon^{-2})))$ time. The approximate Hausdorff Core algorithm makes use of an approximate algorithm for the k-bounded Hausdorff Core problem. We extended this by describing an approximation scheme for the k-bounded Hausdorff Core problem. These are the first known algorithms for the Hausdorff Core problem on general simple polygons.

As future work, it would be interesting to explore other metrics. We studied the Hausdorff metric, but any of the other metrics discussed in Section 2.1 could be used. In our original application, we envisioned the creation of a hierarchy of simplified polygons, from full-resolution contour lines down to the simplest possible approximations. This would permit testing paths against progressively more accurate (and more expensive) approximate representations of polygons until we found a definitive answer regarding whether the path and polygon intersect. Our definition of the d-core, including the Hausdorff Core, requires the solution to be convex. While convexity has many useful consequences, it represents a compromise to the original goal because it only provides one non-adjustable level of approximation. It would be interesting to consider other related problems that might provide more control over the approximation level. Another direction for further work would be to define some other constraint upon the simplified polygon. For instance, we could require that it be star-shaped, i.e. there exists some point $p \in P$ such that every $q \in P$ can see p. A similar but even more general concept might be defined in terms of link width. Another extension would be to explore the Hausdorff Core problem with additional objectives, such as finding a Hausdorff Core with optimal (i.e. maximal or minimal) area, perimeter or number of vertices.

Acknowledgements

This paper has benefitted from significant improvement due to the careful reading and corrections provided by a number of anonymous reviewers as well as Thomas Erlebach, Jochen Koenemann, and Anna Lubiw, and we are grateful for their hard work and input. We also thank Diego Arroyuelo, Barbara Macdonald and Patrick Nicholson for participating in discussions on the Hausdorff Core problem.

References

[ABB95] H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes. *Annals of Mathematics and Artificial Intelligence*, 15:186–193, 1995.

[AG92] H. Alt and M. Godau. Measuring the resemblance of polygonal curves. In *Proc. of the Symposium on Computational Geometry (SoCG)*, pages 102–109, New York, NY, USA, 1992. ACM.

[AKM+87] A. Aggarwal, M.M. Klawe, S. Moran, P. Shor, and R. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(1):195–208, 1987.

[AP88] A. Aggarwal and J. Park. Notes on searching in multidimensional monotone arrays. In *Proc. of the Symposium on Foundations of Computer Science (FOCS)*, pages 497–512. IEEE, 1988.

[Bal81] D. H. Ballard. Strip trees: a hierarchical representation for curves. *Communications of the ACM*, 24(5):310–321, 1981.

[BM02] B.K. Bhattacharya and A. Mukhopadhyay. On the minimum perimeter triangle enclosing a convex polygon. In *Proc. of the Japan Conference on Discrete and Computational Geometry*, volume 2866 of *LNCS*, pages 84–96. Springer, 2002.

[BT96] P. Bose and G. Toussaint. Computing the constrained Euclidean, geodesic and link centre of a simple polygon with applications. In *Proc. of the Pacific Graphics International*, pages 102–111. IEEE, 1996.

[CC05] J.-M. Chassery and D. Coeurjolly. Optimal shape and inclusion. In *Mathematical Morphology: 40 Years On*, volume 30, pages 229–248. Springer, 2005.

[CEG+94] B. Chazelle, H. Edelsbrunner, M. Grigni, L. Guibas, J. Hershberger, M. Sharir, and J. Snoeyink. Ray shooting in polygons using geodesic triangulations. *Algorithmica*, 12(1):54–68, 1994.

[CY86] J. S. Chang and C. K. Yap. A polynomial solution for the potato-peeling problem. *Discrete and Computational Geometry*, 1(1):155–182, 1986.

[Dae03] O. Daescu. New results on path approximation. *Algorithmica*, 38(1):131–143, 2003.

[DDF+09] R. Dorrigiv, S. Durocher, A. Farzan, R. Fraser, A. López-Ortiz, J.I. Munro, A. Salinger, and M. Skala. Finding a Hausdorff core of a polygon: On convex polygon containment with bounded Hausdorff distance. In *Proc. of the Workshop on Algorithms and Data Structures (WADS)*, volume 5664 of *LNCS*, pages 218–229. 2009.

[DLMS12] S. Durocher, A. Leblanc, J. Morrison, and M. Skala. Robust nonparametric data approximation of point sets via data reduction. In *Algorithms and Computation*, volume 7676 of *LNCS*, pages 319–331. 2012.

[DP73] D.H. Douglas and T.K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112–122, 1973.

[DS79] D.P. Dobkin and L. Snyder. On a general method for maximizing and minimizing among certain geometric problems. In *Proc. of Foundations of Computer Science (FoCS)*, pages 9–17, 1979.

[EGHP+01] A. Efrat, L.J. Guibas, S. Har-Peled, J.S.B. Mitchell, and T.M. Murali. New similarity measures between polylines with applications to morphing and polygon sweeping. *Discrete and Computational Geometry*, 28(4):535–569, 2001.

[EM01] R. Estkowski and J. S. B. Mitchell. Simplifying a polygonal subdivision while keeping it simple. In *Proc. of the Symposium on Computational Geometry (SoCG)*, pages 40–49. ACM, 2001.

[EW91] P. Egyed and R. Wenger. Ordered stabbing of pairwise disjoint convex sets in linear time. *Discrete Applied Mathematics*, 31(2):133–140, 1991.

[FMR+92] R. Fleischer, K. Mehlhorn, G. Rote, E. Welzl, and C. Yap. Simultaneous inner and outer approximation of shapes. *Algorithmica*, 8(1):365–389, 1992.

[FN10] R. Fraser and P.K. Nicholson. Hausdorff core of a one reflex vertex polygon. In *Proc. of the Canadian Conference on Computational Geometry (CCCG)*, pages 183–186, 2010.

[GHMS93] L.J. Guibas, J.E. Hershberger, J.S.B. Mitchell, and J.S. Snoeyink. Approximating polygons and subdivisions with minimum-link paths. *International Journal of Computational Geometry & Applications*, 3(4):383–415, 1993.

[Gru83] P. M. Gruber. Approximation of convex bodies. In *Convexity and its Applications*, pages 131–162. Birkhauser Verlag, 1983.

[Hob93] J. D. Hobby. Polygonal approximations that minimize the number of inflections. In *Proc. of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 93–102. SIAM, 1993.

[II86] H. Imai and M. Iri. Computational-geometric methods for polygonal approximations of a curve. *Computer Vision, Graphics, and Image Processing*, 36(1):31–41, 1986.

[KF03] A. Kolesnikov and P. Fränti. Polygonal approximation of closed contours. In *Proc. of the Scandinavian Conference on Image Analysis (SCIA)*, pages 409–417, 2003.

[Li07] Z. Li. *Algorithmic Foundation of Multi-Scale Spatial Representation.* CRC Press, Boca Raton, 2007.

[LR05] M.A. Lopez and S. Reisner. Hausdorff approximation of convex polygons. *Computational Geometry: Theory and Applications*, 32(2):139–158, 2005.

[LVBL98] M.S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1–3):193–228, 1998.

[LY90] M. K. Leung and Y.-H. Yang. Dynamic two-strip algorithm in curve fitting. *Pattern Recognition*, 23(1-2):69–79, 1990.

[MS90] E.A. Melissaratos and D.L. Souvaine. On solving geometric optimization problems using shortest paths. In *Proc. of the Symposium on Computational Geometry (SoCG)*, pages 350–359. ACM, 1990.

[OAMB86] J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin. An optimal algorithm for finding minimal enclosing triangles. *Journal of Algorithms*, 7:258–269, 1986.

[O'R98] J. O'Rourke. *Computational geometry in C.* Cambridge University Press, 1998.

[Peu76] T. K. Peucker. A theory of the cartographic line. *International Cartographic Yearbook*, 16:134–143, 1976.

[PV94] J.-C. Perez and E. Vidal. Optimum polygonal approximation of digitized curves. *Pattern Recognition Letters*, 15(8):743–750, 1994.

[Rob85] J. Robergé. A data reduction algorithm for planar curves. *Computer Vision, Graphics, and Image Processing*, 29:168–195, 1985.

[Rot92] G. Rote. The convergence rate of the sandwich algorithm for approximating convex functions. *Computing*, 48(3-4):337–361, 1992.

[RW74] K. Reumann and A.P.M. Witkam. Optimizing curve segmentation in computer graphics. In *Proc. of the 1973 International Computing Symposium*, pages 467–472, 1974.

[STV+95] C. Schwarz, J. Teich, A. Vainshtein, E. Welzl, and B.L. Evans. Minimal enclosing parallelogram with application. In *Proc. of the Symposium on Computational Geometry (SoCG)*, pages 434–435. ACM, 1995.

[SW87] J. Singh and M. Wagh. Robot path planning using intersecting convex shapes: Analysis and simulation. *Robotics and Automation, IEEE Journal of*, 3(2):101–108, 1987.

[SW92] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. of the Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 577 of *LNCS*, pages 567–579. 1992.

[ZT97] L. Zhang and Z. Tian. Refinement of Douglas-Peucker algorithm to move the segments toward only one side. In *Proc. of the International Cartographic Conference*, pages 831–835, 1997.