

Approximating Convex Shapes With Respect to Symmetric Difference Under Homotheties

Juyoung Yon^{*1}, Sang Won Bae^{†2}, Siu-Wing Cheng^{‡3},
Otfried Cheong^{‡4}, and Bryan T. Wilkinson^{§5}

- 1 KAIST, Daejeon, Korea
- 2 Kyonggi University, Suwon, Korea
- 3 HKUST, Hong Kong
- 4 KAIST, Daejeon, Korea
- 5 MADALGO, Aarhus, Denmark

Abstract

The symmetric difference is a robust operator for measuring the error of approximating one shape by another. Given two convex shapes P and C , we study the problem of minimizing the volume of their symmetric difference under all possible scalings and translations of C . We prove that the problem can be solved by convex programming. We also present a combinatorial algorithm for convex polygons in the plane that runs in $O((m+n)\log^3(m+n))$ expected time, where n and m denote the number of vertices of P and C , respectively.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases shape matching, convexity, symmetric difference, homotheties

Digital Object Identifier 10.4230/LIPIcs.SoCG.2016.63

1 Introduction

The *shape matching* problem is the task of evaluating the similarity of two objects, and the *shape approximation* problem is to find a simpler representation of a given shape. These two problems arise in various fields and have numerous applications. Both problems call for a quantification of the approximation error, which requires a definition of the “distance” between a pair of shapes. The symmetric difference is one such measure, and it is robust against noise [16]. The symmetric difference of two sets X and Y is defined as $X \Delta Y = (X \setminus Y) \cup (Y \setminus X) = (X \cup Y) \setminus (X \cap Y)$. We use $|X|$ to denote the volume of a set X . The symmetric difference is symmetric, $X \Delta X = \emptyset$ for every set X , and the volume of the symmetric difference satisfies the triangle inequality, that is $|X \Delta Y| \leq |X \Delta Z| + |Z \Delta Y|$ as $X \Delta Y = (X \Delta Z) \Delta (Z \Delta Y) \subseteq (X \Delta Z) \cup (Z \Delta Y)$. If we restrict ourselves to a family of sets where $|X \Delta Y| = 0$ only if $X = Y$, then the symmetric difference is a metric over that family. This is, for instance, the case for convex compact shapes [16].

* Yon is supported by ICT R&D program of MSIP/IITP [R0126-15-1108] and by the Research Grant Council, Hong Kong, China, under the HKPF and GRF project 611711, respectively.

† Bae was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2013R1A1A1A05006927) and by the Ministry of Education (NRF-2015R1D1A1A01057220).

‡ Cheong is supported by ICT R&D program of MSIP/IITP [R0126-15-1108] and by the Research Grant Council, Hong Kong, China, under the HKPF and GRF project 611711, respectively.

§ Wilkinson is supported in part by the Danish National Research Foundation grant DNRF84 through the Center for Massive Data Algorithmics and in part by the Natural Sciences and Engineering Research Council of Canada. Part of this research was done while he visited KAIST.



The shape approximation problem often comes in the following form: Given shapes P and C and a family of transformations Φ (translations, rigid motions, scalings, etc.), find the transformation $\varphi \in \Phi$ that minimizes the distance between P and φC . In this paper, we consider the class Φ of *homotheties* and (a generalization of) the symmetric difference distance. (Two sets are *homothets* if they differ by a scaling factor and possibly a translation.) In other words, we compute $\min_{\varphi \in \Phi} |P \Delta \varphi C|$, for two convex shapes P and C in \mathbb{R}^d .

Our results are the first that allow transformations beyond translations in minimizing the volume of the symmetric difference of two convex shapes. In fact, scaling is rarely considered in previous shape matching results.

We optimize a more general function Δ defined as follows. Let κ be a parameter from $[0, 1]$ chosen *a priori*. We use rX to denote the set X scaled by a factor r .

$$\begin{aligned} \text{dist}(P, Q) &:= (1 - \kappa) |P \setminus Q| + \kappa |Q \setminus P| \\ \Delta(\lambda, t) &:= \text{dist}(P, \lambda C + t). \end{aligned}$$

The problem is finding $\lambda \geq 0$ and $t \in \mathbb{R}^d$ that minimize $\Delta(\lambda, t)$. For $\kappa = \frac{1}{2}$, $\text{dist}(P, Q)$ is equal to $\frac{1}{2}|P \Delta Q|$. There is a bias for P or Q if $\kappa < \frac{1}{2}$ or $\kappa > \frac{1}{2}$, respectively.

The parametrized measure $\text{dist}(\cdot)$ may find applications in determining a simple data description in machine learning (e.g. [15]). Assume that we are given a set of uniformly distributed examples (data points) in some high dimensional feature space. The positive examples cover some region P and the examples outside P are negative. If we seek a simpler model C to classify the positive examples, we may minimize the number of false positives in $C \setminus P$ and the number of false negatives in $P \setminus C$, which can be approximated by $|C \setminus P|$ and $|P \setminus C|$, respectively, because of the uniform distribution assumption. If P is a convex body enclosing the positive examples (e.g., convex hull) and C is a simpler convex shape (e.g. ellipsoid), then the measure $\text{dist}(\cdot)$ is applicable. False positives and false negatives are often not equally important; for example, false positives are more serious in spam filtering because they interfere with mail delivery. This difference in importance can be captured by using an appropriate κ in the definition of $\text{dist}(\cdot)$.

Main results

Although Δ is not even quasiconvex, we prove that Δ can be minimized by convex programming. Moreover, if P and C are convex polytopes, then every local minimum of Δ is also a global minimum. We also present a combinatorial algorithm that minimizes Δ for a pair of convex polygons P and C in the plane. The algorithm runs in $O((m+n)\log^3(m+n))$ expected time, where n and m denote the number of vertices in P and C , respectively. We also establish several necessary conditions for (λ, t) to be a local minimum of Δ .

Related work

A lot of previous results consider bounds on $|P \Delta Q|$ for convex shapes P and Q that are fixed in place, or finding a good approximation of P in a given class of convex shapes. Groemer [10] studied some inequalities that relate different similarity measures. Let $\delta(P, Q)$ denote the Hausdorff distance between P and Q . Groemer proved that for a given pair of compact convex sets P and Q in \mathbb{R}^d , if $P \cap Q \neq \emptyset$, then $|P \Delta Q| = O(\delta(P, Q))$ and $\delta(P, Q) = O(|P \Delta Q|^{1/d})$ for $d \geq 2$, where the big-O constants depend on P , Q , and d .

Brass and Lassak [5] described several known bounds on $|P \Delta T|$, where T is a specific triangle chosen for an arbitrary convex polygon P : (1) for all P , there exists $T \subseteq P$ such that $|P \Delta T| \leq \left(1 - \frac{3\sqrt{3}}{4\pi}\right) |P| \approx 0.586|P|$, and (2) for all P , there exists T containing P

such that $|P \triangle T| \leq 2|P|$. Schymura [14] expressed the effect of two rigid motions φ_1 and φ_2 on the overlap of two bounded sets X and Y in \mathbb{R}^d (not necessarily convex) in terms of $|\varphi_1(X) \triangle \varphi_2(X)|$. For $i \in \{1, 2\}$, φ_i consists of a translation v_i and a rotation ρ_i . Let α denote the norm of the vector $v_1 - v_2$, and β be the maximum distance between $\rho_1(x)$ and $\rho_2(x)$ among the points x in the boundary of X . Schymura showed that the difference between $|\varphi_1(X) \cap Y|$ and $|\varphi_2(X) \cap Y|$ is at most $\frac{1}{2}|\varphi_1(X) \triangle \varphi_2(X)|$, which is at most $\frac{1}{2}(\alpha + \beta)$ times the $(d - 1)$ -volume of the boundary of X .

Fleischer et al. [8] studied the problem of approximating a convex polygon by a homothetic pair of circumscribing and inscribed k -gons (not necessarily regular). They are interested in two quantities: $\lambda(P, Q) = \min\{\frac{s}{r} \mid rQ + t \subseteq P \subseteq sQ + t' \text{ for some translations } t \text{ and } t'\}$ and $\lambda_k = \min\{\lambda(P, Q_k) \mid \text{convex polygon } P \text{ and convex } k\text{-gon } Q\}$. They showed that $1 + \frac{\sqrt{5}}{2} = 2.118 \dots \leq \lambda_3 \leq 2.25$ and $\lambda_k = 1 + \Theta(k^{-2})$. They developed an algorithm that, for any convex polygon P with n vertices, finds a triangle T in $O(n^2 \log^2 n)$ time such that $\lambda(P, T)$ is minimized. They also presented another algorithm that finds a triangle T in $O(n)$ time such that $\lambda(P, T) \leq 2.25$.

Alt et al. [2] studied the problem of determining an axis-parallel rectangle or circle Q to minimize $|P \triangle Q|$ for any convex polygon P . Suppose that P has n vertices and they are stored in an array in clockwise order around the polygon boundary. Alt et al.'s algorithms find a solution rectangle in $O(\log^3 n)$ time and a solution circle in $O(n^5 \log n)$ time.

Before our work in this paper, previous algorithms for computing the exact similarity of two convex shapes with respect to the symmetric difference allow translations only. These algorithms solve the equivalent problem of maximizing the volume of the intersection of the two shapes. Given two convex polygons in the plane with a total of n vertices, de Berg et al. [7] proposed an algorithm that finds the translation that maximizes the area of the intersection of the two polygons in $O(n \log n)$ time. For two convex polytopes in \mathbb{R}^d with a total of n facets, Ahn, Cheng, and Reinbacher [1] presented an algorithm that finds the translation that maximizes the intersection volume in $O(n \log^{3.5} n)$ time for $d = 3$ and $O(n^{\lfloor d/2 \rfloor + 1} \log^d n)$ time for $d \geq 4$ under some genericity assumption. In the plane, if one settles for a constant factor upper bound on the similarity of two convex shapes, Alt et al. [3] showed that a much larger class Φ of transformations can be allowed. The class Φ can be as general as the class of affine mappings, the class of similarities (combinations of scaling and rigid motion), the class of homotheties, the class of rigid motions, or just the class of translations. Alt et al. showed that there exists a transformation $\sigma \in \Phi$ such that the centroids of P and $\sigma(Q)$ are aligned, and $|P \triangle \sigma(Q)| \leq \frac{11}{3} \min_{\varphi \in \Phi} |P \triangle \varphi(Q)|$. For two convex polygons with a total of n vertices, Alt et al. developed two algorithms that find such a transformation σ , one for the class of homotheties and another for the class of rigid motions. The running times are $O(n)$ and $O(n^4)$ for homotheties and rigid motions, respectively.

2 Minimizing Δ by convex programming

We show how to minimize Δ by convex programming. Without loss of generality, we assume that the origin lies in C . Let us define

$$f(\lambda, t) := |(\lambda C + t) \cap P|, \quad g(\lambda, t) := (f(\lambda, t))^{1/d}.$$

The function $g(\lambda, t)$ is well defined as $f(\lambda, t) \geq 0$. We rewrite $\text{dist}(P, Q)$ as $\text{dist}(P, Q) = \kappa|Q| + (1 - \kappa)|P| - |Q \cap P|$. Let us define

$$q(\lambda, t) := f(\lambda, t) - \kappa|C|\lambda^d. \tag{1}$$

We then have

$$\begin{aligned} \Delta(\lambda, t) &= \text{dist}(P, \lambda C + t) = \kappa|\lambda C + t| + (1 - \kappa)|P| - f(\lambda, t) \\ &= \kappa|C|\lambda^d + (1 - \kappa)|P| - f(\lambda, t) \\ &= (1 - \kappa)|P| - q(\lambda, t). \end{aligned} \tag{2}$$

$$\tag{3}$$

Since $(1 - \kappa)|P|$ is a constant, minimizing $\Delta(\lambda, t)$ is equivalent to maximizing $q(\lambda, t)$. We can always make $\text{dist}(P, \lambda C + t)$ equal to $(1 - \kappa)|P|$ by setting λ to zero. It follows that the minimum of $\Delta(\lambda, t)$ is at most $(1 - \kappa)|P|$, and any (λ, t) that makes $q(\lambda, t) < 0$ is uninteresting because $\Delta(\lambda, t)$ would then be greater than $(1 - \kappa)|P|$. Hence, we can restrict the domain to the following subset when looking for the minimum of Δ :

$$\mathfrak{D} := \{(\lambda, t) \mid q(\lambda, t) \geq 0\}.$$

We need to establish several properties of the domain \mathfrak{D} and the functions f and q in order to conclude that convex programming is applicable.

2.1 Convexity of \mathfrak{D}

We first show that g is a concave function, that is the volume below the graph of g is convex. The lemma follows quite easily from the Brunn-Minkowski theorem [13], and the same argument has been used before [7].

► **Lemma 2.1.** *The function $g(\lambda, t)$ is concave on the domain where it is non-zero.*

Proof. It suffices to prove the claim along a line through the (λ, t) -space. Let this line be parametrized as $(\lambda(\xi), t(\xi))$, for $\xi \in \mathbb{R}$, with linear functions $\lambda(\xi)$ and $t(\xi)$. We extend our d -dimensional space to $d + 1$ dimensions by adding a coordinate axis z , and write a point as (x, ξ) , where $x \in \mathbb{R}^d$ and $\xi \in \mathbb{R}$. Let $\mathfrak{P} = \{(x, \xi) \in \mathbb{R}^d \times \mathbb{R} \mid x \in P\}$ and let $\mathfrak{C} = \{(x, \xi) \in \mathbb{R}^d \times \mathbb{R} \mid x \in \lambda(\xi)Q + t(\xi)\}$. Both \mathfrak{P} and \mathfrak{C} are convex sets, and so $\mathfrak{P} \cap \mathfrak{C}$ is a convex set as well. We note that the d -dimensional volume of the intersection of $\mathfrak{P} \cap \mathfrak{C}$ with the hyperplane $z = \xi$ is exactly $f(\lambda(\xi), t(\xi))$. By the Brunn-Minkowski Theorem in \mathbb{R}^{d+1} , the function $\xi \mapsto (f(\lambda(\xi), t(\xi)))^{1/d} = g(\lambda(\xi), t(\xi))$ is therefore a concave function. ◀

We show that \mathfrak{D} is convex which is necessary for applying convex programming.

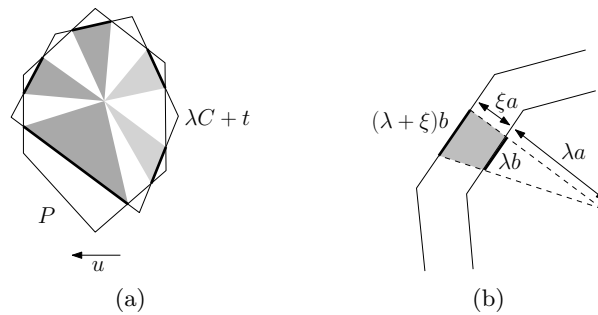
► **Lemma 2.2.** *The set $\mathfrak{D} = \{(\lambda, t) \mid q(\lambda, t) \geq 0\}$ is convex.*

Proof. It suffices to show that any line ℓ in the (λ, t) -space intersects \mathfrak{D} in a single interval. Consider such a line ℓ . We parametrize it by λ , that is, we write $t = t(\lambda)$ for some linear function $t(\lambda)$. Recall that $g(\lambda, t) = (f(\lambda, t))^{1/d}$. (We ignore the case where λ remains constant over ℓ , as this case follows immediately from the concavity of the function g .) We use $q(\lambda)$, $f(\lambda)$, and $g(\lambda)$ to denote $q(\lambda, t(\lambda))$, $f(\lambda, t(\lambda))$, and $g(\lambda, t(\lambda))$, respectively. We have $(\lambda, t(\lambda)) \in \mathfrak{D}$ if and only if $0 \leq q(\lambda) = f(\lambda) - \kappa|C|\lambda^d = (g(\lambda))^d - \kappa|C|\lambda^d$, which is equivalent to $(g(\lambda))^d \geq \kappa|C|\lambda^d$. Since both sides are positive, this is equivalent to $g(\lambda) \geq (\kappa|C|)^{1/d}\lambda$. Since g is concave by Lemma 2.1, the set of λ satisfying this inequality is an interval. ◀

2.2 Derivatives of the function f

For every choice of (λ, t) , define the *rhizome* of C to be

$$\rho(\lambda, t) := \{x \in \partial C \mid \lambda x + t \in P\},$$



■ **Figure 1** (a) Let B_1 and B_2 be the sets of points on the bold segments of the darkly and lightly shaded triangles, respectively. Let R be the union of all shaded triangles. The front and back rhizomes with respect to u are $\{\frac{1}{\lambda}(x - t) \mid x \in B_1\}$ and $\{\frac{1}{\lambda}(x - t) \mid x \in B_2\}$, respectively. And $\mathfrak{R}(\lambda, t) = \{\frac{1}{\lambda}(x - t) \mid x \in R\}$. (b) The shaded region is the volume swept by an infinitesimal expansion of a boundary element λb of λC .

where ∂A for any subset $A \subset \mathbb{R}^d$ denotes the boundary of A . In other words, the rhizome $\rho(\lambda, t)$ of C is the part of ∂C that lies in P under the homothety (λ, t) . We will now argue that the partial derivatives of $f(\lambda, t)$ is closely related to the size of the rhizome.

Consider first a unit direction vector $u \in \mathbb{R}^d$ and the partial derivative of f at (λ, t) in translation direction u . We distinguish the *front rhizome*, which are the points $x \in \rho(\lambda, t)$ such that $\lambda x + t + \delta u \notin \lambda C + t$ for an arbitrarily small positive δ , and the *back rhizome*, which are the points $x \in \rho(\lambda, t)$ such that $\lambda x + t + \delta u$ lies in the interior of $\lambda C + t$ for an arbitrarily small positive δ . Figure 1(a) gives an illustration.

► **Lemma 2.3.** *Suppose that the intersection of the boundaries of $\lambda C + t$ and P has zero $(d - 1)$ -dimensional volume. For every unit direction vector $u \in \mathbb{R}^d$, $\frac{\partial}{\partial u} f(\lambda, t) = F - B$, where F and B denote the $(d - 1)$ -dimensional volumes of the projections in direction u of the front and back rhizomes at (λ, t) with respect to direction u , respectively.*

Proof. Fix some choices of $u \in \mathbb{R}^d$, $\lambda > 0$, and $t \in \mathbb{R}^d$. We denote by γ^+ and γ^- the front and back rhizomes of C at (λ, t) with respect to direction u , respectively. Consider $\lambda C + t + \delta u$. As δ increases infinitesimally from 0, a point on the front rhizome γ^+ contributes to a positive change, while a point on the back rhizome γ^- contributes to a negative change in $f(\lambda, t)$. On the other hand, if δ decreases infinitesimally from 0, then a point on the front rhizome γ^+ contributes to a negative change, while a point on the back rhizome γ^- contributes to a positive change in $f(\lambda, t)$. Thus, $\frac{\partial}{\partial u} f(\lambda, t) = f - b$, where f and b denote the $(d - 1)$ -dimensional volumes of the projections of γ^+ and γ^- in direction u , respectively. ◀

We now turn to the partial derivative $\frac{\partial}{\partial \lambda} f(\lambda, t)$. This is closely related to the union of segments that connect the points in the rhizome $\rho(\lambda, t)$ to the origin, which we assume to lie inside C . We denote this union of segments by $\mathfrak{R}(\lambda, t)$. Figure 1(a) gives an example.

► **Lemma 2.4.** *Suppose that the origin lies in the interior of C , and the intersection of the boundaries of $\lambda C + t$ and P has zero $(d - 1)$ -dimensional volume. Then, $\frac{\partial}{\partial \lambda} f(\lambda, t) = d\lambda^{d-1}|\mathfrak{R}(\lambda, t)|$.*

Proof. It suffices to prove the claim for a convex polytope C as the general case of C being a compact convex set then follows by a limit argument. Fix some choices of $\lambda > 0$ and $t \in \mathbb{R}^d$. Consider a boundary element b of $\rho(\lambda, t)$ that lies on a facet ℓ of C at distance a from the origin. Let β be the $(d - 1)$ -dimensional volume of b . The $(d - 1)$ -dimensional

volume of the boundary element λb on λC is $\lambda^{d-1}\beta$. For any $\xi \neq 0$ arbitrarily close to zero, being positive or negative, the contribution of this boundary element b to the difference $f(\lambda + \xi, t) - f(\lambda, t)$ is exactly $\frac{1}{d}(\lambda + \xi)^d a\beta - \frac{1}{d}\lambda^d a\beta$. Figure 1(b) shows an illustration. Taking the limit of this difference divided by ξ as ξ goes to 0, we have $\lim_{\xi \rightarrow 0} \frac{(\lambda + \xi)^d - \lambda^d}{d\xi} a\beta = \lambda^{d-1} a\beta$. Summing up this limit over all such boundary elements b of $\rho(\lambda, t)$ results in the derivative $\frac{\partial}{\partial \lambda} f(\lambda, t)$. On the other hand, $b \subseteq \rho(\lambda, t)$ contributes $\frac{1}{d}a\beta$ to the volume $|\mathfrak{R}(\lambda, t)|$, so we have $\frac{\partial}{\partial \lambda} f(\lambda, t) = d\lambda^{d-1}|\mathfrak{R}(\lambda, t)|$. ◀

We will need one more observation:

► **Lemma 2.5.** *Suppose that C and P are convex polytopes such that no facet of C is parallel to any facet of P . Then f is continuously differentiable at every point (λ, t) .*

Proof. Since no facet of C is parallel to any facet of P , the intersection of the boundaries of $\lambda C + t$ and P has zero $(d-1)$ -dimensional volume for all (λ, t) . By Lemmas 2.3 and 2.4, the partial derivatives of f depend continuously on the rhizome $\rho(\lambda, t)$, which is a continuous function of (λ, t) under the assumption about the facets. ◀

2.3 Convex programming

Let us define the following function s over \mathfrak{D} :

$$s(\lambda, t) = (q(\lambda, t))^{1/d}. \quad (4)$$

Note that $q(\lambda, t) \geq 0$ over \mathfrak{D} , so $s(\lambda, t)$ is well defined. By (3), minimizing $\Delta(\lambda, t)$ is equivalent to maximizing $q(\lambda, t)$ which is equivalent to maximizing $s(\lambda, t)$. We will prove in the following that $s(\lambda, t)$ is concave over \mathfrak{D} , which implies that we can maximize $s(\lambda, t)$ by convex programming.

To show the concavity of s in the domain \mathfrak{D} , it is sufficient to show concavity along all lines through \mathfrak{D} . Let $(\lambda, t(\lambda))$ be such a line, for a linear function $t(\lambda)$. For simplicity, we use $s(\lambda)$, $f(\lambda)$, $g(\lambda)$, and $q(\lambda)$ to denote $s(\lambda, t(\lambda))$, $f(\lambda, t(\lambda))$, $g(\lambda, t(\lambda))$, and $q(\lambda, t(\lambda))$, respectively. We first show that $s''(\lambda) \leq 0$ assuming that f is smooth enough.

► **Lemma 2.6.** *If $f(\lambda)$ is twice differentiable at λ , then $s''(\lambda) \leq 0$.*

Proof. We first collect some derivatives:

$$\begin{aligned} f(\lambda) &= (g(\lambda))^d \\ f'(\lambda) &= d(g(\lambda))^{d-1}g'(\lambda) \end{aligned} \quad (5)$$

$$f''(\lambda) = d(g(\lambda))^{d-1}g''(\lambda) + d(d-1)(g(\lambda))^{d-2}(g'(\lambda))^2 \quad (6)$$

$$q(\lambda) = f(\lambda) - \kappa|C|\lambda^d = (s(\lambda))^d \quad (7)$$

$$q'(\lambda) = f'(\lambda) - d\kappa|C|\lambda^{d-1} = d(s(\lambda))^{d-1}s'(\lambda) \quad (8)$$

$$q''(\lambda) = f''(\lambda) - d(d-1)\kappa|C|\lambda^{d-2} = d(d-1)(s(\lambda))^{d-2}(s'(\lambda))^2 + d(s(\lambda))^{d-1}s''(\lambda) \quad (9)$$

From (8) we have

$$s'(\lambda) = \frac{q'(\lambda)}{d(s(\lambda))^{d-1}}$$

and substituting this into (9) we have

$$d(s(\lambda))^{d-1}s''(\lambda) = q''(\lambda) - d(d-1)(s(\lambda))^{d-2} \frac{(q'(\lambda))^2}{d^2(s(\lambda))^{2d-2}} = q''(\lambda) - \frac{d-1}{d} \frac{(q'(\lambda))^2}{q(\lambda)}$$

and therefore

$$d^2(s(\lambda))^{2d-1}s''(\lambda) = dq(\lambda)q''(\lambda) - (d-1)(q'(\lambda))^2.$$

We will show that the right hand side of the above equation is at most 0, which implies $s''(\lambda) \leq 0$. By (7)–(9), we have

$$\begin{aligned} & dq(\lambda)q''(\lambda) - (d-1)(q'(\lambda))^2 \\ &= d(f(\lambda) - \kappa|C|\lambda^d)(f''(\lambda) - d(d-1)\kappa|C|\lambda^{d-2}) - (d-1)(f'(\lambda) - d\kappa|C|\lambda^{d-1})^2 \\ &= df(\lambda)f''(\lambda) - d\kappa|C|\lambda^d f''(\lambda) - d^2(d-1)\kappa|C|\lambda^{d-2}f(\lambda) + d^2(d-1)\kappa^2|C|^2\lambda^{2d-2} \\ &\quad - (d-1)f'(\lambda)^2 + 2(d-1)d\kappa|C|\lambda^{d-1}f'(\lambda) - d^2(d-1)\kappa^2|C|^2\lambda^{2d-2}. \end{aligned}$$

Hence,

$$\begin{aligned} dq(\lambda)q''(\lambda) - (d-1)(q'(\lambda))^2 &= (df(\lambda)f''(\lambda) - (d-1)(f'(\lambda))^2) \\ &\quad - d\kappa|C|\lambda^{d-2}(\lambda^2 f''(\lambda) - 2(d-1)\lambda f'(\lambda) + d(d-1)f(\lambda)) \end{aligned} \quad (10)$$

By (6) and (5) we have

$$\begin{aligned} d^2(g(\lambda))^{2d-1}g''(\lambda) &= df(\lambda)f''(\lambda) - d^2(d-1)(g(\lambda))^{2d-2}(g'(\lambda))^2 \\ &= df(\lambda)f''(\lambda) - d^2(d-1)(g(\lambda))^{2d-2} \frac{(f'(\lambda))^2}{d^2(g(\lambda))^{2d-2}} \\ &= df(\lambda)f''(\lambda) - (d-1)(f'(\lambda))^2. \end{aligned}$$

Plugging this into (10) and applying again (5) and (6) we have

$$\begin{aligned} & dq(\lambda)q''(\lambda) - (d-1)(q'(\lambda))^2 \\ &= (d^2(g(\lambda))^{2d-1}g''(\lambda)) - d\kappa|C|\lambda^{d-2}(\lambda^2 d(g(\lambda))^{d-1}g''(\lambda) + d(d-1)(g(\lambda))^{d-2}(g'(\lambda))^2\lambda^2 \\ &\quad - 2(d-1)\lambda d(g(\lambda))^{d-1}g'(\lambda) + d(d-1)(g(\lambda))^d) \\ &= d^2(g(\lambda))^{d-1}(f(\lambda) - \kappa|C|\lambda^d)g''(\lambda) \\ &\quad - d^2(d-1)\kappa|C|(g(\lambda))^{d-2}\lambda^{d-2}((g'(\lambda))^2\lambda^2 - 2g'(\lambda)\lambda g(\lambda) + (g(\lambda))^2) \\ &= d^2g(\lambda)^{d-1}q(\lambda)g''(\lambda) - d^2(d-1)\kappa|C|(g(\lambda))^{d-2}\lambda^{d-2}(g'(\lambda)\lambda - g(\lambda))^2 \\ &\leq 0, \end{aligned}$$

since $q(\lambda) \geq 0$, $g(\lambda) \geq 0$, and $g''(\lambda) \leq 0$ on \mathfrak{D} . ◀

We are ready to prove that $s(\lambda, t)$ is concave over \mathfrak{D} .

► **Lemma 2.7.** *The function $(\lambda, t) \mapsto s(\lambda, t) = (f(\lambda, t) - \kappa|C|\lambda^d)^{1/d}$ is concave on the domain \mathfrak{D} where $f(\lambda, t) - \kappa|C|\lambda^d \geq 0$.*

Proof. We first prove the theorem for the case where P and C are convex polytopes such that no facet of C is parallel to a facet of P . By Lemma 2.5, this implies that $f(\lambda, t)$ is continuously differentiable at every point (λ, t) . In particular, this means that the function $\lambda \mapsto f(\lambda) = f(\lambda, t(\lambda))$ is continuously differentiable at any λ , and therefore $s(\lambda)$ is continuously differentiable everywhere on the domain \mathfrak{D} . In order to prove the lemma, it suffices to show that $s'(\lambda)$ is monotonously decreasing.

Consider the intersection polytope $I(\lambda) = (\lambda C + t(\lambda)) \cap P$. For a fixed λ , we can triangulate this polytope $I(\lambda)$ into d -dimensional simplices and express its volume $|I(\lambda)|$ as

the sum of volumes of those simplices. Note that the volume of a d -dimensional simplex is represented as a determinant involving $d + 1$ vertices of the intersection polytope $I(\lambda)$.

As λ changes, the combinatorial structure of $I(\lambda)$ and its triangulation only changes at certain breakpoints. Between two consecutive breakpoints, the expression of the volume $|I(\lambda)|$ of the intersection polytope $I(\lambda)$ remains the same. On the other hand, since $t(\lambda)$ is a linear function of λ , each vertex of $I(\lambda)$ between any two consecutive breakpoints is represented by a linear function of λ . This implies that the volume $|I(\lambda)|$ is expressed by a polynomial in λ with degree at most d . The intersection volume $|I(\lambda)|$ is thus twice differentiable at any λ between two consecutive breakpoints, so by Lemma 2.6 we have $s''(\lambda) \leq 0$ between two consecutive breakpoints. This implies that $s'(\lambda)$ is monotonously decreasing in any maximal interval defined by two consecutive breakpoints. Since $s'(\lambda)$ is also continuous, $s'(\lambda)$ is monotonously decreasing for any λ on the domain \mathfrak{D} . This completes the proof of the lemma when C and P are polytopes such that no facet of C is parallel to any facet of P .

We then turn to the case where C and P are arbitrary convex shapes. Suppose to the contrary that the claim was false, that is, the function $s(\lambda)$ is not concave. This implies the existence of three values $\lambda_0 < \lambda_1 < \lambda_2$, where $\lambda_1 = (1 - \tau)\lambda_0 + \tau\lambda_2$ with $0 < \tau < 1$, such that $s(\lambda_1) < (1 - \tau)s(\lambda_0) + \tau s(\lambda_2)$, that is, the concavity of s is violated.

Let $\varepsilon = (1 - \tau)s(\lambda_0) + \tau s(\lambda_2) - s(\lambda_1) > 0$. Since $s(\lambda)$ depends continuously on the objects P and C , we can approximate them by convex polytopes P' and C' such that $s(\lambda)$ changes by at most $\varepsilon/3$ for $\lambda \in \{\lambda_0, \lambda_1, \lambda_2\}$. This approximation can be done such that no facet of C' is parallel to any facet of P' . Since the concavity of s at the values $\lambda_0, \lambda_1, \lambda_2$ is still violated, this is a contradiction to our proof for the polytope case. \blacktriangleleft

The above lemma immediately implies one of the main results of this paper.

► **Theorem 2.8.** *The function $\Delta(\lambda, t)$ attains its global minimum within the domain $\mathfrak{D} = \{(\lambda, t) \mid f(\lambda, t) - \kappa|C|\lambda^d \geq 0\}$, and this minimum can be computed using convex programming.*

Proof. Recall that $\Delta(\lambda, t) = (1 - \kappa)|P| - q(\lambda, t)$ by (3) and $s(\lambda, t) = (q(\lambda, t))^{1/d}$. Since Δ is greater than $(1 - \kappa)|P|$ outside \mathfrak{D} , the global minimum of Δ is in \mathfrak{D} . This minimum corresponds to a maximum of s . Since \mathfrak{D} is convex (Lemma 2.2) and s is concave over \mathfrak{D} (Lemma 2.7), one can maximize s by convex programming to obtain the minimum of Δ . \blacktriangleleft

3 Additional properties of the intersection volume function

We refine the analysis of the rhizome and the partial derivatives of f for the case that P and C are convex polytopes. We show that every local minimum of Δ lies in the domain \mathfrak{D} , and is therefore a global minimum of Δ . We also prove that the function $\lambda \mapsto \min_{t \in \mathbb{R}^d} \Delta(\lambda, t)$ is unimodal. The partial derivatives of f and the unimodality of $\min_{t \in \mathbb{R}^d} \Delta(\lambda, t)$ will be useful for developing the combinatorial algorithm for convex polygons in the next section.

Recall that the rhizome $\rho(\lambda, t)$ of C is the part of ∂C that lies in P under the homothety (λ, t) . Some portion of the rhizome $\rho(\lambda, t)$ can be mapped to points in ∂P . We define

$$\rho_0(\lambda, t) := \{x \in \partial C \mid \lambda x + t \in \partial P\},$$

and call it the *critical rhizome* of C at (λ, t) . Note that $\rho_0(\lambda, t) \subseteq \rho(\lambda, t)$.

Consider first a unit direction vector $u \in \mathbb{R}^d$, and the partial derivative of f at (λ, t) in direction u , denoted by $\frac{\partial}{\partial u} f(\lambda, t)$. As we will see, the derivative $\frac{\partial}{\partial u} f(\lambda, t)$ of f in direction u is not always well defined, while both its left and right derivatives are well defined everywhere. Specifically, the left and the right partial derivatives of f at (λ, t) in direction

u are defined to be the following one-sided limits: $\frac{\partial_-}{\partial u} f(\lambda, t) = \lim_{\delta \rightarrow 0^-} \frac{f(\lambda, t + \delta u) - f(\lambda, t)}{\delta}$ and $\frac{\partial_+}{\partial u} f(\lambda, t) = \lim_{\delta \rightarrow 0^+} \frac{f(\lambda, t + \delta u) - f(\lambda, t)}{\delta}$. If the left and right derivatives are equal, then $\frac{\partial}{\partial u} f(\lambda, t)$ is well defined as $\frac{\partial_-}{\partial u} f(\lambda, t) = \frac{\partial_+}{\partial u} f(\lambda, t) = \frac{\partial}{\partial u} f(\lambda, t)$.

Like front and back rhizomes, we also distinguish between *front and back critical rhizomes*. The front critical rhizome consists of points $x \in \rho_0(\lambda, t)$ such that $\lambda x + t + \delta u \notin \lambda C + t$ for an arbitrary small positive δ , and the back critical rhizome consists of points $x \in \rho_0(\lambda, t)$ such that $\lambda x + t + \delta u$ lies in the interior of $\lambda C + t$ for an arbitrary small positive δ . The proof of this and the following lemma can be found in the full version of our paper.

► **Lemma 3.1.** *Assume that C and P are convex polytopes. For every unit direction vector $u \in \mathbb{R}^d$, $\frac{\partial_-}{\partial u} f(\lambda, t) = F - B + B_0$, $\frac{\partial_+}{\partial u} f(\lambda, t) = F - B - F_0$ and where F , B , F_0 , and B_0 denote the $(d - 1)$ -dimensional volumes of the projections in direction u of the front, back, front critical, and back critical rhizomes of C at (λ, t) with respect to direction u , respectively.*

We now turn to the left and right partial derivatives of f at (λ, t) with respect to the scaling λ . Recall that $\mathfrak{R}(\lambda, t)$ denotes the union of segments that connect the points in the rhizome $\rho(\lambda, t)$ to the origin, which we assume to lie inside C . Analogously, we denote by $\mathfrak{R}_0(\lambda, t)$ the union of segments that connect the points in $\rho_0(\lambda, t)$ to the origin.

As done above, we analyze the partial derivative of f with respect to λ by looking into its one-sided versions $\frac{\partial_-}{\partial \lambda} f(\lambda, t) = \lim_{\delta \rightarrow 0^-} \frac{f(\lambda + \delta, t) - f(\lambda, t)}{\delta}$ and $\frac{\partial_+}{\partial \lambda} f(\lambda, t) = \lim_{\delta \rightarrow 0^+} \frac{f(\lambda + \delta, t) - f(\lambda, t)}{\delta}$.

► **Lemma 3.2.** *Assume that C and P are convex polytopes. Suppose that the origin lies in the interior of C . Then, $\frac{\partial_-}{\partial \lambda} f(\lambda, t) = d\lambda^{d-1}|\mathfrak{R}(\lambda, t)|$ and $\frac{\partial_+}{\partial \lambda} f(\lambda, t) = d\lambda^{d-1}(|\mathfrak{R}(\lambda, t)| - |\mathfrak{R}_0(\lambda, t)|)$.*

Lemma 3.2 leads to bounds for the volume of the rhizome at a local minimum of Δ .

► **Lemma 3.3.** *Assume that C and P are convex polytopes. Suppose that $t \in \mathbb{R}^d$ is fixed at some point. If Δ achieves a local minimum under scaling at (λ, t) , then $\kappa|C| \leq |\mathfrak{R}(\lambda, t)| \leq \kappa|C| + |\mathfrak{R}_0(\lambda, t)|$, and this holds for all choices of the origin inside C .*

Proof. A local minimum of $\Delta(\lambda, t)$ under scaling is a local maximum of $q(\lambda, t)$ under scaling. By Lemma 3.2, the one-sided derivatives $\frac{\partial_-}{\partial \lambda} f(\lambda, t)$ and $\frac{\partial_+}{\partial \lambda} f(\lambda, t)$ of f are always well defined, so those of q are well defined as well since $q(\lambda, t) = f(\lambda, t) - \kappa|C|\lambda^d$. Suppose that q attains a local maximum at (λ, t) . Then, either one of the one-sided derivatives of q is zero or they have opposite signs. Therefore, $\left(\frac{\partial_-}{\partial \lambda} q(\lambda, t)\right) \cdot \left(\frac{\partial_+}{\partial \lambda} q(\lambda, t)\right) \leq 0$. By Lemma 3.2, we have $\frac{\partial_-}{\partial \lambda} q(\lambda, t) = \frac{\partial_-}{\partial \lambda} (f(\lambda, t) - \kappa|C|\lambda^d) = d\lambda^{d-1}|\mathfrak{R}(\lambda, t)| - \kappa d|C|\lambda^{d-1}$, and $\frac{\partial_+}{\partial \lambda} q(\lambda, t) = \frac{\partial_+}{\partial \lambda} (f(\lambda, t) - \kappa|C|\lambda^d) = d\lambda^{d-1}(|\mathfrak{R}(\lambda, t)| - |\mathfrak{R}_0(\lambda, t)|) - \kappa d|C|\lambda^{d-1}$. Since $|\mathfrak{R}_0(\lambda, t)| \geq 0$, we must have that $\frac{\partial_-}{\partial \lambda} q(\lambda, t) \geq 0$ and $\frac{\partial_+}{\partial \lambda} q(\lambda, t) \leq 0$. This implies that $\kappa|C| \leq |\mathfrak{R}(\lambda, t)| \leq \kappa|C| + |\mathfrak{R}_0(\lambda, t)|$. ◀

This immediately implies that any local minimum of Δ under scaling is attained in \mathfrak{D} .

► **Lemma 3.4.** *Assume that C and P are convex polytopes. Suppose that $t \in \mathbb{R}^d$ is fixed at some point. If Δ achieves a local minimum under scaling at (λ, t) , then $q(\lambda, t) \geq 0$.*

Proof. Without loss of generality, we can choose an origin in C such that its image t lies in P . By Lemma 3.3, if Δ achieves a local minimum under scaling at (λ, t) , then $|\mathfrak{R}(\lambda, t)| \geq \kappa|C|$, and therefore $f(\lambda, t) = |(\lambda C + t) \cap P| \geq \lambda^d |\mathfrak{R}(\lambda, t)| \geq \kappa|C|\lambda^d$. So, $q(\lambda, t) = f(\lambda, t) - \kappa|C|\lambda^d \geq 0$. ◀

Combining Lemma 3.4 with Lemma 2.7 and Theorem 2.8 gives the following result.

► **Theorem 3.5.** *Assume that C and P are convex polytopes. Every local minimum of $\Delta(\lambda, t)$ is also a global minimum, and it is attained in the convex domain $\mathfrak{D} = \{(\lambda, t) \mid f(\lambda, t) - \kappa|C|\lambda^d \geq 0\}$. This global minimum can be computed using convex programming.*

We can also analyze the dependence of the optimal approximation on the scaling factor λ .

► **Theorem 3.6.** *Assume that C and P are convex polytopes. The function $\Delta_{\min}(\lambda) = \min_{t \in \mathbb{R}^d} \Delta(\lambda, t)$ decreases monotonously from $\lambda = 0$ to its minimum at $\lambda = \lambda^*$, and then increases monotonously for $\lambda \geq \lambda^*$.*

Proof. The function Δ_{\min} is clearly decreasing for λ close to zero, and increasing for large λ . Let λ' be a local minimum of Δ_{\min} , and let t' be such that $\Delta(\lambda', t')$ assumes this minimum. Then (λ', t') is a local minimum of Δ , and by Theorem 3.5 this is the unique value $\lambda' = \lambda^*$. The claim follows. ◀

4 Optimality under translations in the plane

Suppose that λ is fixed at some value. A placement (λ, t) is a local minimum of $\Delta(\lambda, t)$ under translation if and only if (λ, t) is a local maximum of $f(\lambda, t)$ under translation. We observed in Lemma 3.1 that if C and P are convex polytopes, both the left and right partial derivatives $\frac{\partial_-}{\partial u} f(\lambda, t)$ and $\frac{\partial_+}{\partial u} f(\lambda, t)$ with respect to any unit vector u are always well defined. This gives a general criterion for a local minimum under translations.

In this section we consider the planar case, and assume that the boundaries of $\lambda C + t$ and P intersect in a finite number of points. This is always true, for instance, when C and P are polygons without parallel edges or when C is a circle and P is a polygon. Under this assumption, the rhizome $\rho(\lambda, t)$ consists of a finite number of intervals of ∂C that we will denote $\rho_i(\lambda, t)$, for $1 \leq i \leq k$ and some constant k . For each rhizome interval $\rho_i(\lambda, t)$, let its endpoints be p_i and q_i such that $\rho_i(\lambda, t)$ is the counter-clockwise interval on ∂C from p_i to q_i . We define the *rhizome vector* $\vec{\rho}_i(\lambda, t)$ as the vector $q_i - p_i$.

► **Lemma 4.1.** *Let C and P be convex polygons in the plane such that no edge of C is parallel to any edge of P . Suppose that $\lambda > 0$ is fixed at some value. If Δ achieves a local minimum under translation at (λ, t) , then the rhizome vectors sum to zero: $\sum_{i=1}^k \vec{\rho}_i(\lambda, t) = 0$.*

Proof. Let $p_i = (x_i, y_i)$ and $q_i = (x'_i, y'_i)$. Consider first the direction vector $u = (1, 0)$, that is, a translation in the positive x -direction. Let $F \subset \{1, 2, \dots, k\}$ be the set of indices such that $\rho_i(\lambda, t)$ is part of the front rhizome. Similarly, let $B \subset \{1, 2, \dots, k\}$ be the set of indices such that $\rho_i(\lambda, t)$ is part of the back rhizome.

For $i \in F$, we have $y'_i > y_i$, while for $i \in B$, we have $y'_i < y_i$. The length of the projection of $\rho_i(\lambda, t)$ on the y -axis is $y'_i - y_i$ for $i \in F$ and $y_i - y'_i$ for $i \in B$. (For $i \notin F \cup B$ we have $y_i = y'_i$.) By Lemma 2.3, the partial derivative $\frac{\partial}{\partial u} f(\lambda, t)$ is well defined, and it should be zero. Thus we have $\sum_{i \in F} y'_i - y_i = \sum_{i \in B} y_i - y'_i$, which is equivalent to $\sum_{i=1}^k y'_i - y_i = 0$.

We consider next the direction vector $u = (0, 1)$, that is, a translation in the positive y -direction. Arguing as before, we obtain $\sum_{i=1}^k x'_i - x_i = 0$. Putting both statements together the lemma follows. ◀

For the special case where C is a circle, we obtain another elegant characterization:

► **Lemma 4.2.** *Let C be a circle, and let P be a convex polygon in the plane. Suppose that $\lambda > 0$ is fixed at some value. If Δ achieves a local minimum under translation at (λ, t) , then the centroid of the rhizome $\rho(\lambda, t)$ coincides with the center of C .*

Proof. We can assume that C is the unit circle and the center of C is $(0, 0)$. Let α_i and β_i be the angular position of p_i and q_i on the circle. We then have $\vec{\rho}_i(\lambda, t) = q_i - p_i = (\cos \beta_i - \cos \alpha_i, \sin \beta_i - \sin \alpha_i)$. As in the proof of Lemma 4.1, we have $\sum_{i=1}^k (\cos \beta_i - \cos \alpha_i) = 0$ and $\sum_{i=1}^k (\sin \beta_i - \sin \alpha_i) = 0$.

On the other hand, the centroid of $\rho_i(\lambda, t)$ is

$$\left(\int_{\alpha_i}^{\beta_i} \cos \theta \, d\theta, \int_{\alpha_i}^{\beta_i} \sin \theta \, d\theta \right) = (\sin \beta_i - \sin \alpha_i, \cos \alpha_i - \cos \beta_i).$$

Therefore, the centroid of the entire rhizome coincides with $(0, 0)$, the center of circle C . ◀

5 Exact algorithm

Let (λ^*, t^*) be the global minimum of $\Delta(\lambda, t)$. In Section 2, we showed that one can compute numerical approximations of λ^* and t^* via convex programming for arbitrary convex bodies C and P in \mathbb{R}^d . There are efficient convex programming solvers such as CVX and MOSEK, and they would be the solution of choice in practice. However, the convex programming routines are numerical in nature and their running times depend on the input bit complexity and the solution precision required. Therefore, from an algorithmic theory standpoint, it is justifiable to ask whether there is a fast, exact combinatorial algorithm in the RAM model (a common computation model for computational geometry). In this section, we sketch such an algorithm when P and C are convex polygons with n and m vertices, respectively. We assume without loss of generality that the origin lies in C . The details of the algorithm can be found in the full version of our paper.

It is not difficult to design an algorithm with running time polynomial in m and n . For example, when λ is fixed, the achievement by de Berg et al. [7] is to find the best translation in $O((m+n)\log(m+n))$ time despite the complexity of the translation configuration space (for a fixed λ) being $O(m^2n^2)$. Similarly, our goal is to find (λ^*, t^*) in $O((m+n)\text{polylog}(m+n))$ time. Since there is a known algorithm by de Berg et al. [7] for a fixed λ under translation, it is natural to apply parametric search. However, in order to use Meggido’s generic parametric search technique as a black box [12], one would need to parallelize the algorithm of de Berg et al., which is rather complicated. To keep a simpler and self-contained description, we present a direct randomized parametric search algorithm.

For simplicity, we assume that no vertex of C or P has the same y -coordinate as another vertex of C or P . This can always be enforced by a slight rotation.

5.1 Overview of the maximum overlap algorithm

Assume that λ is fixed. We outline de Berg et al.’s algorithm for finding the translation that maximizes the overlap (and hence minimizes the symmetric difference). There are two stages and one key subroutine for which we will develop parametric versions later.

Let a_1, a_2, \dots, a_n and $\lambda b_1, \lambda b_2, \dots, \lambda b_m$ be the y -coordinates of the vertices of P and λC . There are mn vertical translations $a_i - \lambda b_j$ that align two vertices of P and λC horizontally. These vertical translations correspond to mn horizontal lines $y = \ell_j$ for $j = 1, 2, \dots, mn$ from top to bottom in the translation configuration space. These lines cut the configuration space into $mn + 1$ horizontal strips.

A key primitive of the algorithm is the following **line search**: Let ℓ be a given line in the translation configuration space. Compute the translation $t \in \ell$ that maximizes the overlap of $\lambda C + t$ with P . This can be done in $O(m+n)$ time [4].

The **first stage** of the algorithm localizes the horizontal strip, among the $mn + 1$ candidates in the translation configuration space, that contains the optimal placement of λC . We take $k_{\min} = 1$, $k_{\max} = mn$, $k = mn/2$, and find the optimal translations along the lines $y = \ell_j$ for $j \in \{k_{\min}, k, k + 1, k_{\max}\}$. The maximum overlap values then tell us whether we can ignore strips below $y = \ell_{k+1}$ or those above $y = \ell_k$. A binary search will lead us to the desired horizontal strip in the translation configuration space in $O((m + n) \log(m + n))$ time.

In the **second stage**, the algorithm starts with the strip S that contains the goal placement of λC . Consider an arbitrary vertex λv of λC . For every horizontal line h within S , the line $\lambda v + h$ intersects the same edge(s) of P (at most two). Let uw be such an edge. Then, the translations that move λv onto uw form the line segment that joins $u - \lambda v$ and $w - \lambda v$ in the translation configuration space. Similarly, for every vertex w of P , the line $w + h$ intersects the same edge(s) of λC for any horizontal line h within S , and w and these edge(s) of λC induce at most two line segments in the translation configuration space. By taking the supporting lines of these line segments, we get an arrangement \mathcal{A} of $O(m + n)$ lines in the configuration space. If we can identify which cell σ of \mathcal{A} contains the optimal placement, then for all translations $t \in \sigma$, the combinatorial structure of $(\lambda C + t) \cap P$ is fixed. That is, each vertex of $(\lambda C + t) \cap P$ is the intersection of two fixed edges of λC and P , and thus we can obtain its coordinates as linear functions in t . The area of $(\lambda C + t) \cap P$ is a quadratic polynomial in t whose maximum value can be calculated using standard calculus. One can zoom into σ as follows. Compute a $(1/r)$ -cutting for \mathcal{A} for some appropriate constant $r > 1$. By solving the one-dimensional optimal placement of λC along the supporting line of each edge in the cutting, we zoom into one cell of the cutting and then retrieve the $(m + n)/r$ lines of \mathcal{A} that intersect this cell. Afterwards, we recurse on these $(m + n)/r$ lines. We spend $O(m + n)$ time at each recursion level, resulting in a total of $O((m + n) \log(m + n))$ time to find the cell σ .

We extend de Berg et al.'s algorithm to a parametric search version in which λ is not fixed. We will also go through the same two stages. Since λ is not fixed, there is an extra dimension to the configuration space (the configuration space is now three-dimensional). We need to work with an arrangement of planes in three dimensions. However, it is time-consuming to deal with a three-dimensional configuration space in its full generality. The trick is to continuously constrain the possible interval of λ values that contains the optimal λ^* .

5.2 Decision algorithm: comparing λ with λ^*

We will need a method to test whether a given value λ' is less than, equal to, or greater than the optimal value λ^* . This decision needs to be done without knowing λ^* . The proof of the following lemma can be found in the full version.

► **Lemma 5.1.** *Suppose that the function $t \mapsto \Delta(\lambda', t)$ achieves the minimum at t' . Then, we have*

- $\lambda' = \lambda^*$ if and only if $(\kappa|C| - |\mathfrak{R}(\lambda', t')|)(\kappa|C| - |\mathfrak{R}(\lambda', t')| + |\mathfrak{R}_0(\lambda', t')|) \leq 0$.
- $\lambda' < \lambda^*$ if and only if $\kappa|C| - |\mathfrak{R}(\lambda', t')| + |\mathfrak{R}_0(\lambda', t')| < 0$.
- $\lambda' > \lambda^*$ if and only if $\kappa|C| - |\mathfrak{R}(\lambda', t')| > 0$.

Moreover, $\kappa|C| - |\mathfrak{R}(\lambda', t')|$ and $\kappa|C| - |\mathfrak{R}(\lambda', t')| + |\mathfrak{R}_0(\lambda', t')|$ can be computed in $O((m + n) \log(m + n))$ time given λ' .

5.3 Parametric version of searching along a line

The parametric version of the line search primitive is the following: We are given a plane h in configuration space, and search for $(\lambda, t) \in h$ that maximizes $|(\lambda C + t) \cap P|$. We represent the plane h as $l(\lambda)$ —for a fixed λ , $l(\lambda)$ is a line in the translation space.

We do not solve the problem in its full generality. Instead, we assume that the input satisfies some conditions that we explain below.

Consider a fixed λ . As we vary $t \in l(\lambda)$, the edges of P that can be hit by a vertex λv of λC are those edges of P intersected by the line $\lambda v + l(\lambda)$. There are at most two such edges, but in general their identities vary depending on the value of λ . The same can be said for the edges of λC that can be hit by the vertices of P as we vary t along $l(\lambda)$. We assume that this does not happen within a range $[\lambda_1, \lambda_2]$ specified in the input. Precisely, as we vary $\lambda \in [\lambda_1, \lambda_2]$, for every vertex λv of λC , the line $\lambda v + l(\lambda)$ does not sweep over any vertex of P , and for every vertex w of P , the line $w - l(\lambda)$ does not sweep over any vertex of λC . Under this condition, we want to find $(\lambda, t) \in l(\lambda)$ that maximizes $|\lambda C + t \cap P|$.

There are events at which the combinatorial structure of $P \cap (\lambda C + t)$ changes. These are moments at which a vertex of λC runs into an edge of P or vice versa. Notice that a vertex λv of λC runs into at most two edges of P . The same is true for the events at which vertices of P running into edges of λC . There are $O(m + n)$ such events in total. These events happen in a fixed order as we vary t along $l(\lambda)$ when $\lambda = \lambda_1$. However, when λ is varied linearly within $[\lambda_1, \lambda_2]$, the occurrence time of such an event is a linear function in λ . That is, the occurrence times of the events as functions of λ form an arrangement of $O(m + n)$ lines. Each intersection of two linear functions gives a value of λ . We are only interested in the pair of intersections that are successive in the increasing order of λ value and sandwich λ^* . These two successive values $\lambda_3 \leq \lambda_4$ can be identified using expected $O(\log(m + n))$ binary search probes at the vertices of the arrangement by following the framework of randomized slope selection [11]. Each probe involves deciding whether the corresponding λ value is greater than or less than λ^* using Lemma 5.1. Therefore, we can find $[\lambda_3, \lambda_4]$ in $O((m + n) \log^2(m + n))$ expected time. Within $[\lambda_3, \lambda_4]$, the ordering of the events is fixed, and these events/lines divide the vertical strip $[\lambda_3, \lambda_4] \times (-\infty, \infty)$ into $O(m + n)$ event trapezoids. Sort these event trapezoids in vertical order in $O((m + n) \log(m + n))$ time.

We perform a binary search on the sorted event trapezoids as follows. Let I be the current event trapezoid that we are probing. Let $t = h_1(\lambda)$ and $t = h_2(\lambda)$ be the two lines that bound I . We first spend $O(m + n)$ time to identify the intersecting pairs of edges of $\lambda C + t$ and P , and this fixes the combinatorial structure of $P \cap (\lambda C + t)$. The area of $P \cap (\lambda C + t)$ is a quadratic function $F(\lambda, t)$ within this event trapezoid. The maximum of F is achieved only when $\partial F / \partial t = 0$, which gives a line $H: t = h(\lambda)$. If H does not intersect I within $[\lambda_3, \lambda_4]$, then $\partial F / \partial t$ has a fixed sign within I which tells us how to continue the binary search. If H intersects I within $[\lambda_3, \lambda_4]$, then H intersects H_1 and/or H_2 at one or two values of λ , say λ_5 and λ_6 . Using Lemma 5.1, we can decide whether $\lambda_i < \lambda^*$ or $\lambda_i > \lambda^*$ for $i \in \{5, 6\}$. This allows us to constrain $[\lambda_3, \lambda_4]$ to a smaller interval $[\lambda', \lambda'']$ such that, within $[\lambda', \lambda'']$, H lies either completely outside or inside I . In the former case, $\partial F / \partial t$ has a fixed sign within I and it tells us how to continue the binary search. The latter case is the terminating case of the binary search, that is, F achieves its maximum within I , and we can obtain this maximum by substituting $t = h(\lambda)$ into $F(\lambda, t)$ and optimize the resulting quadratic function in λ . In all, we can compute the maximum overlap of $P \cap (\lambda C + t)$ over $t \in l(\lambda)$ and $\lambda \in [\lambda_1, \lambda_2]$ in $O((m + n) \log^2(m + n))$ expected time.

5.4 Parametric version of first stage

In $O(m \log m + n \log n)$ time, we order the vertices of P and C from top to bottom. Let their y -coordinates be $a_1 < a_2 < \dots < a_{n-1} < a_n$ and $b_1 < b_2 < \dots < b_{m-1} < b_m$, respectively.

Let $\ell_{ij}(\lambda)$ denote the set of translations that put the vertex of P with y -coordinate a_i and the vertex of λC with y -coordinate λb_j at the same height. Recall that for a fixed λ ,

$\ell_{ij}(\lambda)$ is a horizontal line in de Berg et al.'s algorithm. Since λ can vary, $\ell_{ij}(\lambda)$ is a plane in the three-dimensional configuration space.

In the first stage we will identify a region in the three-dimensional configuration space that contains (λ^*, t^*) by binary search among the planes $\ell_{ij}(\lambda)$'s.

Consider a plane $\ell_{ij}(\lambda)$. We first constrain the range of λ to an interval $[\lambda_1, \lambda_2]$ such that $\lambda^* \in [\lambda_1, \lambda_2]$ and, as we vary $\lambda \in [\lambda_1, \lambda_2]$, for every vertex λv of λC , the line $\lambda v + \ell_{ij}(\lambda)$ does not sweep over any vertex of P and for every vertex w of P , the line $w + \ell_{ij}(\lambda)$ does not sweep over any vertex of Q . We determine λ_1 and λ_2 by binary search (using Lemma 5.1) on the critical values of λ . Since we cannot afford to produce the entire list of these roughly mn critical values for λ , we arrange these values in a matrix, and make use of searching a totally sorted matrix in linear time [9].

We are then in a position to apply the algorithm of Section 5.3 to determine the position (λ^*, t^*) with respect to $\ell_{ij}(\lambda)$.

To summarize, in the first stage we identify a range $[\lambda_1, \lambda_2]$ and two planes $\ell_{ij}(\lambda)$ and $\ell_{rs}(\lambda)$ such that $\lambda^* \in [\lambda_1, \lambda_2]$, $\ell_{ij}(\lambda)$ and $\ell_{rs}(\lambda)$ sandwich (λ^*, t^*) , and no plane $\ell_{kl}(\lambda)$ in the configuration space lies between $\ell_{ij}(\lambda)$ and $\ell_{rs}(\lambda)$ or intersects one of them within the range $[\lambda_1, \lambda_2]$. The expected run time for this stage is $O((m+n)\log^3(m+n))$.

5.5 Parametric version of second stage

Let R be the region computed by the first stage. It lies within $[\lambda_1, \lambda_2]$ and is bounded by two planes in the configuration space. Consider a vertex v of C and a horizontal line h in the configuration space that stabs R . The first stage guarantees that for all $\lambda \in [\lambda_1, \lambda_2]$, the line $\lambda v + h$ intersects the same edge(s) of P . Similarly, for every vertex w of P , the line $w + h$ intersects the same edge(s) of λC for all $\lambda \in [\lambda_1, \lambda_2]$. Thus, there are at most $E \leq 2(m+n)$ events corresponding to translations that put a vertex of λC on an edge of P or vice versa. These events form subsets of planes in the three-dimensional configuration space. We call these planes *event planes*, and we can compute their equations in $O(m+n)$ time.

We compute in $O(m+n)$ time a $(1/4)$ -cutting of these E event planes [6], which is a simplicial complex of constant size. Each cell of the cutting intersects at most $E/4$ other event planes. For every supporting plane H of cells in the cutting, since we have the output range $[\lambda_1, \lambda_2]$ of the first stage, we can apply the subroutine in Section 5.3 to find $(\lambda_H, t_H) \in H$ that minimizes Δ and hence maximizes s . This takes $O((m+n)\log^2(m+n))$ expected time. Let L be the supporting plane such that $s(\lambda_L, t_L)$ is maximum among all supporting planes. Due to the concavity of the function s , for every supporting plane $H \neq L$, the side of H that contains (λ^*, t^*) is the side that contains (λ_L, t_L) . Let \mathcal{A} be the arrangement of all supporting planes other than L . It means that we can identify the cell σ in \mathcal{A} that contains (λ^*, t^*) . The plane L may split σ into two pieces, each of which is contained in some cell in the cutting. Therefore, the number of event planes that intersect σ is at most $2(E/4) \leq E/2$. We collect the event planes that intersect σ and then recurse on σ . In $O(\log(m+n))$ rounds we can narrow down to a convex subset K of the configuration space in which there is no event. That is, the combinatorial structure of $P \cap (\lambda C + t)$ is fixed within K , and this structure can be determined in $O(m+n)$ time using any point $(\lambda, t) \in C$. Let $t = (t_x, t_y)$. Then, we can express the area of the overlap within K as a quadratic function in λ , t_x and t_y in $O(m+n)$ time. Finally, we can compute the three partial derivatives, set them to zero to obtain three linear equations in λ , t_x and t_y , and solve the linear system to obtain (λ^*, t^*) . The overall expected running time is $O((m+n)\log^3(m+n))$.

► **Theorem 5.2.** *Let P and C be two convex polygons in the plane with n and m vertices, respectively. We can compute the optimal scaling factor $\lambda^* \geq 0$ and the optimal translation $t^* \in \mathbb{R}^2$ such that the area of $P \cap (\lambda^*C + t^*)$ is maximum in $O((m+n) \log^3(m+n))$ expected time. Hence, the area of $P \Delta (\lambda^*C + t^*)$ is minimum.*

Acknowledgment. We thank Sergio Cabello for suggesting this problem and conjecturing our Theorem 3.6. We also thank Emo Welzl for many helpful discussions during O. C.'s visit to ETH in 2015.

References

- 1 H.-K. Ahn, S.-W. Cheng, and I. Reinbacher. Maximum overlap of convex polytopes under translation. *Computational Geometry: Theory and Applications*, 46:552–565, 2013.
- 2 H. Alt, J. Blömer, M. Godau, and H. Wagener. Approximation of convex polygons. In *Automata, Languages and Programming*, volume 443 of *LNCS*, pages 703–716. Springer, 1990.
- 3 H. Alt, U. Fuchs, G. Rote, and G. Weber. Matching convex shapes with respect to the symmetric difference. *Algorithmica*, 21:89–103, 1998.
- 4 D. Avis, P. Bose, T. Shermer, J. Snoeyink, G. Toussaint, and B. Zhu. On the sectional area of convex polytopes. In *Communication at the 12th Annual Symposium on Computational Geometry*, page C, 1996.
- 5 P. Brass and M. Lassak. Problems on approximation by triangles. *Geombinatorics*, 10:103–115, 2001.
- 6 B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete & Computational Geometry*, 9:145–158, 1993.
- 7 M. de Berg, O. Cheong, O. Devillers, M. van Kreveld, and M. Teillaud. Computing the maximum overlap of two convex polygons under translations. *Theory of Computing Systems*, 31:613–628, 1998.
- 8 R. Fleischer, K. Mehlhorn, G. Rote, E. Welzl, and C. Yap. Simultaneous inner and outer approximation of shapes. *Algorithmica*, 8:365–389, 1992.
- 9 G. N. Frederickson and D. B. Johnson. Generalized selection and ranking: sorted matrices. *SIAM Journal on Computing*, 13:14–30, 1984.
- 10 H. Groemer. On the symmetric difference metric for convex bodies. *Beiträge zur Algebra und Geometrie*, 41:107–114, 2000.
- 11 J. Matoušek. Randomized optimal algorithm for slope selection. *Information Processing Letters*, 39:183–187, 1991.
- 12 N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *Journal of ACM*, 30:852–865, 1983.
- 13 R. Schneider. *Convex Bodies – the Brunn-Minkowski theory*. Cambridge University Press, 1993.
- 14 D. Schymura. An upper bound on the volume of the symmetric difference of a body and a congruent copy. *Advances in Geometry*, 14:287–298, 2014.
- 15 D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54:45–66, 2004.
- 16 R. Veltkamp and M. Hagedoorn. Shape similarity measures, properties and constructions. In *Advances in Visual Information Systems*, volume 1929 of *LNCS*, pages 467–476. Springer, 2000.