Approximating the Maximum Overlap of Polygons under Translation[*]

Sariel Har-Peled[†] Subhro Roy[‡]

June 24, 2014

Abstract

Let P and Q be two simple polygons in the plane of total complexity n, each of which can be decomposed into at most k convex parts. We present an $(1-\varepsilon)$-approximation algorithm, for finding the translation of Q, which maximizes its area of overlap with P. Our algorithm runs in $O(cn)$ time, where c is a constant that depends only on k and ε.

This suggest that for polygons that are "close" to being convex, the problem can be solved (approximately), in near linear time.

1. Introduction

Shape matching is an important problem in databases, robotics, visualization and many other fields. Given two shapes, we want to find how similar (or dissimilar) they are. Typical problems include matching point sets by the Hausdorff distance metric, or matching polygons by the Hausdorff or Fréchet distance between their boundaries. See the survey by Alt and Guibas [AG00].

The maximum area of overlap is one possible measure for shape matching that is not significantly effected by noise. Mount *et al.* [MSW96] studied the behavior of the area of overlap function, when one simple polygon is translated over another simple polygon. They showed that the function is continuous and piece-wise polynomial of degree at most two. If the polygons P and Q have complexity m and n, respectively, the area of overlap function can have complexity of $\Theta(m^2n^2)$. Known algorithms to find the maximum of the function work by constructing the entire overlap function. It is also known that the problem is 3SUM-Hard [BH01b], that is, it is believed no subquadratic time algorithm is possible for the problem.

Approximating maximum overlap of general polygons. Cheong *et al.* [CEH07] gave a $(1-\varepsilon)$-approximation algorithm for maximizing the area of overlap under translation of one simple polygon over the other using random sampling techniques. However, the error associated with the algorithm is additive, and the algorithm runs in near quadratic time. Specifically, the error is an ε fraction of the area of the smaller of the two polygons. Under rigid motions, the running time deteriorates to being

[*]Work on this paper was partially supported by NSF AF awards CCF-0915984 and CCF-1217462.

[†]Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sariel@uiuc.edu; http://sarielhp.org.

[‡]Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sroy9@illinois.edu; http://web.engr.illinois.edu/~sroy9/.

near cubic. More recently, Cheng and Lam [CL13] improved the running times, and can also handle rigid motions, and present a near linear time approximation algorithm if one of the polygons is convex.

Maximum overlap in the convex case under translations. de Berg *et al.* [dBCD+98] showed that finding maximum overlap translation is relatively easier in case of convex polygons. Specifically, the overlap function in this case is unimodal (as a consequence of the Brunn-Minkowski Theorem). Using this property, they gave a near linear time exact algorithm for computing the translation that maximizes the area of overlap of two convex polygons. The complexity of the graph of the overlap function is only $O(m^2 + n^2 + \min(m^2 n, mn^2))$ in this case. Alt *et al.* [AFRW98] gave a constant-factor approximation for the minimum area of the symmetric difference of two convex polygons.

Approximating maximum overlap in the convex case. As for $(1-\varepsilon)$-approximation, assuming that the two polygons are provided in an appropriate form (i.e., the vertices are in an array in their order along the boundary of the polygon), then one can get a sub-linear time approximation algorithm. Specifically, Ahn *et al.* [ACP+07] show an $(1-\varepsilon)$-approximation algorithm, with running time $O((1/\varepsilon)\log(n/\varepsilon))$ for the case of translation, and $O((1/\varepsilon)\log n + (1/\varepsilon)^2 \log 1/\varepsilon))$ for the case of rigid motions. (For a result using similar ideas in higher dimensions see the work by Chazelle *et al.* [CLM05].)

Overlap of union of balls. de Berg *et al.* [dBCG+04] considered the case where X and Y are disjoint unions of m and n unit disks, with $m \leq n$. They computed a $(1-\varepsilon)$ approximation for the maximal area of overlap of X and Y under translations in time $O((nm/\varepsilon^2)\log(n/\varepsilon))$. Cheong *et al.* [CEH07] gave an additive error ε-approximation algorithm for this case, with near linear running time.

Other relevant results. Avis *et al.* [ABT+96] computes the overlap of a polytope and a translated hyperplane in linear time, if the polytope is represented by a lattice of its faces. Vigneron [Vig14] presented $(1-\varepsilon)$-approximation algorithms for maximum overlap of polyhedra (in constant dimension) that runs in polynomial time. Ahn *et al.* [ACKY14] approximates the maximum overlap of two convex polytopes in three dimensions under rigid motions. Ahn *et al.* [ACR13] approximates the maximum overlap of two polytopes in $\mathbb{R}^d$ under translation in $O\left(n^{\lfloor d/2 \rfloor + 1} \log^d n\right)$ time.

Our results

As the above indicates, there is a big gap between the algorithms known for the convex and non-convex case. Our work aims to bridge this gap, showing that for "close" to convex polygons, under translation, the problem can be solved approximately in near linear time.

Specifically, assume we are given two polygons P and Q of total complexity n, such that they can be decomposed into k convex parts, we show that one can $(1-\varepsilon)$-approximate the translation of Q, which maximizes its area of overlap with P, in linear time (for k and ε constants). The translation returned has overlap area which is at least $(1-\varepsilon)\mu_{\max}(\mathsf{P},\mathsf{Q})$, where $\mu_{\max}(\mathsf{P},\mathsf{Q})$ is the maximum area of overlap of the given polygons.

Approach. We break the two polygons into a minimum number of convex parts. We then approximate the overlap function for each pair of pieces (everywhere). This is required as one cannot just approximate the two polygons (as done by Ahn *et al.* [ACP+07]) since the optimal solution does not realize the maximum overlap of each pair of parts separately, and the alignment of each pair of parts might be arbitrary.

To this end, if the two convex parts are of completely different sizes, we approximate the smaller part, and approximate the overlap function by taking slices (i.e., level sets) of the overlap function. In the other case, where the two parts are "large", which is intuitively easier, we can approximate both convex parts, and then the overlap function has constant complexity. Finally, we overlap all these functions together, argue that the overlap has low complexity, and find the maximum area of overlap.

Our approach has some overlap in ideas with the work of Ahn *et al.* [ACP+07]. In particular, a similar distinction between large and small overlap, as done in Section 4.1 and Section 4.2 was already done in [ACP+07, Theorem 17].

Why the "naive" solution fails? The naive solution to our problem is to break the two polygons into k convex polygons, and then apply to each pair of them the approximation of Ahn *et al.* [ACP+07]. Now, just treat the input polygon as the union of their respective approximations, and solve problem using brute force approach. This fails miserably as the approximation of Ahn *et al.* [ACP+07] captures only the maximum overlap of the two polygons. It does not, and can not, approximates the overlap if two convex polygons are translated such that their overlap is "far" from the maximum configuration, especially if the two polygons are of different sizes. This issue is demonstrated in more detail in the beginning of Section 4.1. A more detailed counterexample is presented in Appendix B.

Paper organization. We start in Section 2 by defining formally the problem, and review some needed results. In Section 3, we build some necessary tools. Specifically, we start in Section 3.1 by observing that one can get $O(1/\varepsilon)$ approximation of a convex polygon, where the error is an ε-fraction of the width of the polygon. In Section 3.2, we show how to compute a level set of the overlap function of two convex polygons efficiently. In Section 3.3, we show that, surprisingly, the polygon formed by the maximum overlap of two convex polygons, contains (up to scaling by a small constant and translation) the intersection of any translation of these two convex polygons. Among other things this implies an easy linear time constant factor approximation for the maximum overlap (which also follows, of course, by the result of Ahn *et al.* [ACP+07]). In Section 4, we present the technical main contribution of this paper, showing how to approximate, by a compact representation that has roughly linear complexity, the area overlap function of two convex polygons. In Section 5 we put everything together and present our approximation algorithm for the non-convex case.

2. Preliminaries

For any vector $\mathsf{t} \in \mathbb{R}^2$ and a set Q, let $\mathsf{t} + \mathsf{Q}$ denote the translation of Q by t; formally, $\mathsf{t} + \mathsf{Q} = \left\{ \mathsf{t} + \mathsf{q} \;\middle|\; \mathsf{q} \in \mathsf{Q} \right\}$. Also let $\mu(\mathsf{P}, \mathsf{Q}) = \mathrm{area}(\mathsf{P} \cap \mathsf{Q})$, which is the ***area of overlap*** of sets P and Q. We are interested in the following problem.

Problem 2.1. *We are given two polygons X and Y in the plane, such that each can be decomposed into at most k convex polygons. The task is to compute the translation t of Y, which maximizes the area of overlap between X and $\mathsf{t} + \mathsf{Y}$. Specifically our purpose is to approximate the quantity*

$$\mu_{\max}(\mathsf{X}, \mathsf{Y}) = \max_{\mathsf{t} \in \mathbb{R}^2} \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y}) \,.$$

For a polygon P, let $|\mathsf{P}|$ denote the number of vertices of P. For $\mathsf{X}, \mathsf{Y} \subseteq \mathbb{R}^d$, the set X is ***contained under translation*** in Y, denoted by $\mathsf{X} \sqsubseteq \mathsf{Y}$, if there exists $\vec{x}$ such that $\vec{x} + \mathsf{X} \subseteq \mathsf{Y}$.

Unimodal. A function $f : \mathbb{R} \to \mathbb{R}$ is ***unimodal***, if there is a value α, such that f is monotonically increasing (formally, non-decreasing) in the range $[-\infty, \alpha]$, and f is monotonically decreasing (formally, non-increasing) in the interval $[\alpha, +\infty]$.

From width to inner radius. For a convex polygon P, the ***width*** of P, denoted by $\omega(\mathsf{P})$, is the minimum distance between two parallel lines that enclose P.

Lemma 2.2 ([GK92]). *For a convex shape* X *in the plane, we have that the largest disk enclosed inside* X*, has radius at least* $\mathrm{width}(\mathsf{X})\,/2\sqrt{3}$*.*

Convex Decomposition of Simple Polygons. A vertex of a polygon is a ***notch*** if the internal angle at this vertex is reflex (i.e. $> 180°$). For a non-convex polygon P with n vertices and r notches, Keil and Snoeyink [KS02] solves the minimal convex decomposition problem in $O(n + r^2 \min(r^2, n))$ time, that is, they compute a decomposition of P into minimum number of convex polygons. Observe, that if the number of components in the minimum convex decomposition is k, the number of notches r is upper bounded by $2k$.

Scaling similarity between polygons. For two convex polygons X and Y, let us define their ***scaling similarity***, denoted by $\mathrm{ssim}(\mathsf{X}, \mathsf{Y})$, as the minimum number $\alpha \geq 0$, such that $\mathsf{X} \sqsubseteq \alpha\mathsf{Y}$. Using low-dimensional linear programming, one can compute $\mathrm{ssim}(\mathsf{X}, \mathsf{Y})$ in linear time. In particular, the work by Sharir and Toledo [ST94] implies the following.

Lemma 2.3 (ssim). *Given two convex polygons* X *and* Y *of total complexity* n*, one can compute, in linear time,* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y})$*, and the translation that realizes it.*

3. Building blocks

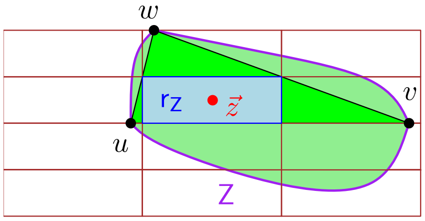
3.1. A better convex approximation in the plane

Let B be the minimum volume bounding box of some bounded convex set $K \subseteq \mathbb{R}^d$. We have that $v + c_d B \subseteq K \subseteq B$ [Har11], for some vector v and a constant c_d which depends only on the dimension d. This approximation can be computed in $O(n)$ time [BH01a], where n is the number of vertices of the convex-hull of K. The more powerful result showing that a convex body can be approximated by an ellipsoid (up to a scaling factor of d), is known as John's Theorem [Har11].

We need the following variant of the algorithm of Barequet and Har-Peled [BH01a].

Lemma 3.1. *Given a convex polygon* Z *in the plane, with* n *vertices, one can compute, in linear time, a rectangle* $\mathsf{r_Z}$ *and a point* $\vec{z}$*, such that* $\vec{z} + \mathsf{r_Z} \subseteq \mathsf{Z} \subseteq \vec{z} + 5\mathsf{r_Z}$*.*

Proof. This is all well known, and we include the details for the sake of completeness. Using rotating caliper [Tou83] compute the two vertices u and v of Z realizing its diameter. Let w be the vertex of Z furthest away from uv, Consider the rectangle r'_{Z} having its base on uv, having half the height of $\triangle uvw$, and contained inside this triangle. Now, let $\vec{z}$ be the center of r'_{Z}, and set $\mathsf{r_Z} = \mathsf{r}'_{\mathsf{Z}} - \vec{z}$, see figure on the right. It is now easy to verify that the claim holds with $\mathsf{r_Z}$ and $\vec{z}$. ■



Observation 3.2. *Given two bodies* $\mathsf{X}, \mathsf{Y} \subseteq \mathbb{R}^2$ *and a non-singular affine transformation* M*, we have* $\dfrac{\operatorname{area}(\mathsf{X})}{\operatorname{area}(\mathsf{Y})} = \dfrac{\operatorname{area}(\mathsf{M}(\mathsf{X}))}{\operatorname{area}(\mathsf{M}(\mathsf{Y}))}$.

Since a similar construction is described by Ahn *et al.* [ACP+07], we delegate the proof of this lemma to Appendix A.

Lemma 3.3 (approxPolygon). *Given a convex polygon* P*, and a parameter* $m > 0$*, we can compute, in* $O(|\mathsf{P}|)$ *time, a convex polygon* P' *with* $O(m)$ *vertices, such that (i)* $\mathsf{P}' \subseteq \mathsf{P}$*, and (ii) for any point* $\mathsf{p} \in \mathsf{P}$*, its distance from* P' *is at most* $\omega(\mathsf{P})/m$*, where* $\omega(\mathsf{P})$ *is the width of* P.

3.2. The level set of the area of overlap function

Definition 3.4. *The superlevel set of a function* $f : \mathbb{R}^d \rightarrow \mathbb{R}$*, for a value* α *is the set* $L_\alpha(f) = \left\{ \mathsf{p} \in \mathbb{R}^d \;\middle|\; f(\mathsf{p}) \geq \alpha \right\}$*. We will refer to it as the* **α-slice** *of* f.

Lemma 3.5. *Given two convex polygons* X *and* Y*, the slice* $\mathsf{Z} = L_\alpha(\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y}))$ *is convex, and has complexity* $O(m)$*, where* $m = |\mathsf{X}|\,|\mathsf{Y}|$*. Furthermore, given a point* $\mathsf{p} \in \mathsf{Z}$*, the convex body* Z *can be computed in* $O(m \log m)$ *time.*

Proof. Along any line, the function $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$ is unimodal [dBCD+98]. This directly implies that on any segment joining two points of the boundary of an α-slice, the function will have values greater than α, and hence, it will lie inside the α-slice. Therefore, Z is convex.

As one translates Y over X, as long as the same pairs of edges intersect, the function governing the overlap function will remain the same quadratic polynomial – the function changes form whenever a vertex is being swept over.

Now, consider an edge e of X and an edge e' of Y. All the translations t for which e intersects $e' + \mathsf{t}$ map out a parallelogram $\pi(e, e')$, which has edges parallel to e and e'. The edges of these parallelograms correspond to translations, where some vertex of Y lies on an edge of X, or vice versa, and the vertices correspond to those translations where some vertex of X coincides with some vertex of Y.

So, consider the arrangement $\mathcal{A}$ of all lines passing through the edges of these parallelograms – it is defined by $O(|\mathsf{X}|\,|\mathsf{Y}|)$ lines, and as such has complexity $O\big(|\mathsf{X}|^2\,|\mathsf{Y}|^2\big)$ overall. By convexity, the boundary $\gamma = \partial\mathsf{Z}$ intersects every line at most twice, and hence the curve γ visits at most $O(|\mathsf{X}|\,|\mathsf{Y}|)$ faces of this arrangement (we count a visit to the same face with multiplicity). Thus, γ can be broken into $O(|\mathsf{X}|\,|\mathsf{Y}|)$ arcs, each one of them of constant descriptive complexity (i.e., its the boundary of a slice of a quadratic function, between intersection points of this curve with two lines).

As for the algorithm, set $\mathsf{t} = \mathsf{p}$, and compute the overlap function value at p by sweeping the two polygons, computing their intersection polygon, and then computing the vertical decomposition of this intersection polygon. It is easy to verify, that the overlap function is a quadratic function of t, and its the sum of the areas of the vertical trapezoids, each one of them can be decomposed into two triangles, and the coordinates of the vertices of each triangle are affine function of t. As such, the area of each triangle is a quadratic function, and adding them up give us the formula for the area of the whole overlap polygon. Now, as t moves, every time the vertical decomposition changes as a vertex of one polygon is swept over by the boundary of the other polygon, this corresponds to a local change in the vertical decomposition of the overlap polygon, and the overlap function can be changed in constant time. Thus, starting from p, we move t up, updating the overlap function as necessary till we reach a point, where the overlap function has value α. At that point, we trace out the outer zone of $\partial\mathsf{Z}$ – since this zone

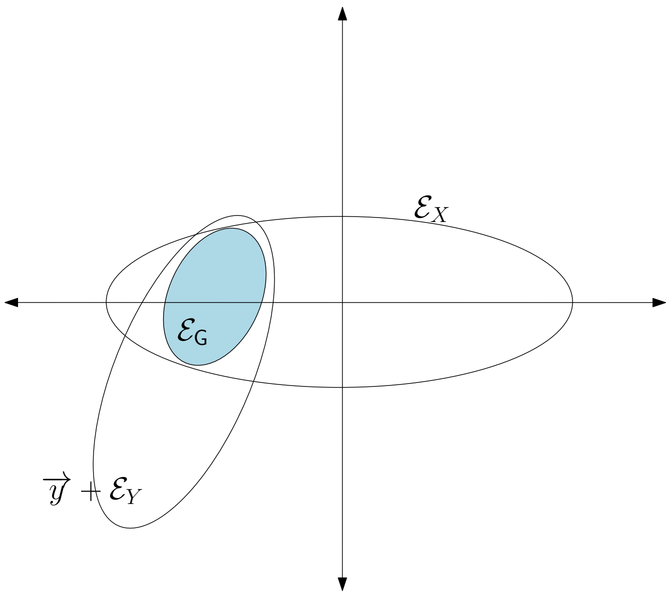

$\mathcal{E}_X$
$\mathcal{E}_{\mathsf{G}}$
$\overrightarrow{y}+\mathcal{E}_Y$

Figure 3.1:

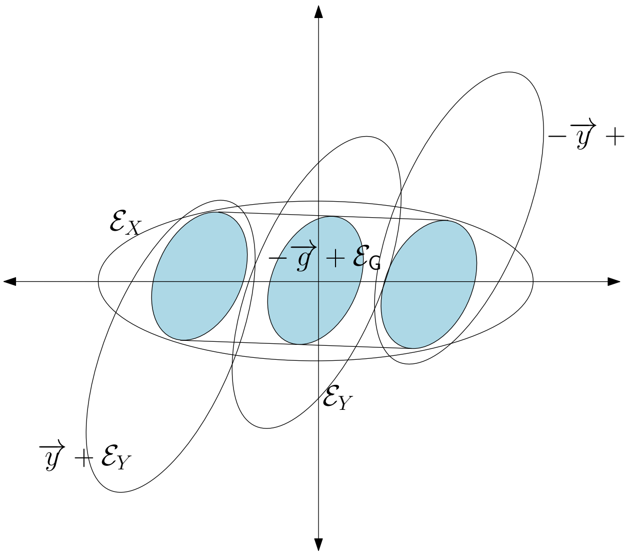

$-\overrightarrow{y}+\mathcal{E}_Y$
$\mathcal{E}_X$
$-\overrightarrow{g}+\mathcal{E}_{\mathsf{G}}$
$\mathcal{E}_Y$
$\overrightarrow{y}+\mathcal{E}_Y$

Figure 3.2:

is defined by $m = O(|\mathsf{X}|\,|\mathsf{Y}|)$ rays, this zone has complexity $O(m)$ (note, that the exact bound on the worst case complexity of the faces of the zone inside a convex region is still open, and it is between $O(m)$ and $O(m\alpha(m))$, where $\alpha(n)$ is the inverse Ackerman function). Specifically, we are walking on the arrangement of $\mathcal{A}$ starting on a point on a line that is on $\partial\mathsf{Z}$, go clockwise on the edges of the face till we encounter $\partial\mathsf{Z}$ again, move to the next face at this point, and repeat. To perform this, we need an efficient data-structure for doing a walk in a planar arrangement [Har00] – in this case since this is an arrangement of lines, we can use a data-structure for dynamic maintenance of convex-hull. In particular, Brodal and Jacob presented a data-structure [BJ02] that supports this kind of operations in $O(\log m)$ amortized time per update. As such, computing $\partial\mathsf{Z}$ takes $O(m \log m)$ time overall. ■

3.3. The shape of the polygon realizing the maximum area overlap

In the following, all the ellipses being considered are centered in the origin.

Lemma 3.6. *Given two ellipses $\mathcal{E}_1$ and $\mathcal{E}_2$, the translation which maximizes their area of overlap is the one in which their centers are the same points.*

Proof. Translate $\mathcal{E}_1$ and $\mathcal{E}_2$ such that their centers are at the origin. Consider any unit vector $\vec{u}$, translate $\mathcal{E}_2$ along the direction of $\vec{u}$, and consider the behavior of the overlap function $f(x) = \mu\Big(\mathcal{E}_1, \mathcal{E}_2 + x\vec{u}\Big)$, where x varies from $-\infty$ to $+\infty$. The function f is unimodal [dBCD$^+$98]. By symmetry, we have

$$f(x) = \mu\Big(\mathcal{E}_1, \mathcal{E}_2 + x\vec{u}\Big) = \mu\Big(-\mathcal{E}_1,\ -(\mathcal{E}_2 + x\vec{u})\Big) = \mu\Big(\mathcal{E}_1, \mathcal{E}_2 - x\vec{u}\Big) = f(-x),$$

as $\mathcal{E}_i = -\mathcal{E}_i$. If the maximum is attained at $x \neq 0$, we will get another maximum at $-x$, which implies, as f unimodal, that $f(0) = f(x) = f(-x)$, as desired. ■

Lemma 3.7. *Consider two ellipses $\mathcal{E}_{\mathsf{X}}$ and $\mathcal{E}_{\mathsf{Y}}$ in the plane, and consider any two vectors $\vec{x}$ and $\vec{y}$, then there is a vector $\vec{u}$ such that $\vec{u} + (\vec{x} + \mathcal{E}_{\mathsf{X}}) \cap (\vec{y} + \mathcal{E}_{\mathsf{Y}}) \subseteq 2\mathcal{E}_{\mathsf{X}} \cap 2\mathcal{E}_{\mathsf{Y}}$.*

Proof. For the sake of simplicity of exposition, assume that $\vec{x} = 0$. Now, consider the intersection $\mathsf{G} = \mathcal{E}_\mathsf{X} \cap (\vec{y} + \mathcal{E}_\mathsf{Y})$, and let $\mathcal{E}_\mathsf{G}$ be the largest area ellipse contained inside G. John's theorem implies that there is a translation vector $\vec{g}$, such that $\vec{g} + \mathcal{E}_\mathsf{G} \subseteq \mathsf{G} \subseteq \vec{g} + 2\mathcal{E}_\mathsf{G}$, see Figure 3.1.

Observe that $\vec{g} + \mathcal{E}_\mathsf{G} \subseteq \mathcal{E}_\mathsf{X}$, and by the symmetry of $\mathcal{E}_\mathsf{G}$ and $\mathcal{E}_\mathsf{X}$, we have that $-\vec{g} + \mathcal{E}_\mathsf{G} = -\vec{g} - \mathcal{E}_\mathsf{G} \subseteq -\mathcal{E}_\mathsf{X} = \mathcal{E}_\mathsf{X}$. This by convexity implies that $\mathcal{E}_\mathsf{G} \subseteq \mathcal{E}_\mathsf{X}$. A similar argument implies that $\mathcal{E}_\mathsf{G} \subseteq \mathcal{E}_\mathsf{Y}$. As such, $\mathcal{E}_\mathsf{G} \subseteq \mathcal{E}_\mathsf{X} \cap \mathcal{E}_\mathsf{Y}$.

Thus, we have that $\mathsf{G} \subseteq \vec{g} + 2\mathcal{E}_\mathsf{G} \subseteq \vec{g} + 2\mathcal{E}_\mathsf{X} \cap 2\mathcal{E}_\mathsf{Y}$, as desired. ∎

Lemma 3.8 ([Har11, Lemma 22.5]). *Any convex set $K \subseteq \mathbb{R}^d$ contained in a unit square, contains a ball of radius* $\mathrm{area}(K)/8$

The following lemma is one of our key insights – the maximum area of intersection of two polygons contains any intersection of translated copies of these polygons up to translation and a constant factor scaling.

Lemma 3.9. *Let X and Y be two convex polygons, and let M be the polygon realizing their maximum area of intersection under translation. Let $\vec{u}$ be any vector in the plane, and consider the polygon $\mathsf{D} = \mathsf{X} \cap (\vec{u} + \mathsf{Y})$, then there exists a vector $\vec{v}$ such that, $\vec{v} + \mathsf{D} \subseteq c_0\mathsf{M}$, for some fixed constant c_0.*

Proof. Let $\mathcal{E}_\mathsf{X}$ (resp., $\mathcal{E}_\mathsf{Y}$) denote the maximum area ellipse (centered at the origin) contained inside X (resp. Y). By John's Theorem, we have $\vec{x} + \mathcal{E}_\mathsf{X} \subseteq \mathsf{X} \subseteq \vec{x} + 2\mathcal{E}_\mathsf{X}$ and $\vec{y} + \mathcal{E}_\mathsf{Y} \subseteq \mathsf{Y} \subseteq \vec{y} + 2\mathcal{E}_\mathsf{Y}$, where $\vec{x}, \vec{y}$ are some vector. Let $\mathsf{B} = \mathcal{E}_\mathsf{X} \cap \mathcal{E}_\mathsf{Y}$, and let $\mathcal{E}_\mathsf{B}$ be the maximum area ellipse contained inside B. Observe that B is symmetric and centered at the origin, and by John's theorem $\mathcal{E}_\mathsf{B} \subseteq \mathsf{B} \subseteq 2\mathcal{E}_\mathsf{B}$.

By Lemma 3.7, there are vectors $\vec{z}$ and $\overrightarrow{w}$, such that

$$\begin{aligned}\mathsf{D} = \mathsf{X} \cap (\vec{u} + \mathsf{Y}) &\subseteq (\vec{x} + 2\mathcal{E}_\mathsf{X}) \cap (\vec{z} + \vec{y} + 2\mathcal{E}_\mathsf{Y}) \subseteq \overrightarrow{w} + 4\mathcal{E}_\mathsf{X} \cap 4\mathcal{E}_\mathsf{Y} = \overrightarrow{w} + 4\mathsf{B}\\ &\subseteq \overrightarrow{w} + 8\mathcal{E}_\mathsf{B}.\end{aligned}$$

Applying a similar argument, we have that $\mathsf{M} \subseteq \overrightarrow{m} + 8\mathcal{E}_\mathsf{B}$, for some vector $\overrightarrow{m}$.

Apply the linear transformation that maps $\mathcal{E}_\mathsf{B}$ to $\mathrm{disk}(1/16)$, where $\mathrm{disk}(r)$ denotes the disk of radius r centered at the origin. By Observation 3.2, we can continue our discussion in the transformed coordinates. This implies that $\mathsf{M} - \overrightarrow{m} \subseteq \mathrm{disk}(1/2)$ (which is contained inside a unit square). By Lemma 3.8, there is a vector $\vec{x_1}$, such that $\vec{x_1} + \mathrm{disk}(\mathrm{area}(\mathsf{M})/8) \subseteq \mathsf{M}$.

Observe that $\mathsf{B} = \mathcal{E}_\mathsf{X} \cap \mathcal{E}_\mathsf{Y} \subseteq (-\vec{x} + \mathsf{X}) \cap (-\vec{y} + \mathsf{Y})$. As such, the area of B must be smaller than the area of M (by the definition of M). We thus have $\mathrm{area}(\mathsf{M}) \geq \mathrm{area}(\mathsf{B}) \geq \mathrm{area}(\mathcal{E}_\mathsf{B}) = \mathrm{area}(\mathrm{disk}(1/16))$ which is a constant bounded away from zero. Therefore,

$$\begin{aligned}\mathsf{D} &\subseteq \overrightarrow{w} + 8\mathcal{E}_\mathsf{B} = \overrightarrow{w} + \mathrm{disk}\left(\frac{1}{2}\right) = \overrightarrow{w} + \frac{4}{\mathrm{area}(\mathsf{M})} \cdot \mathrm{disk}\left(\frac{\mathrm{area}(\mathsf{M})}{8}\right)\\ &\subseteq \overrightarrow{w} + \frac{4}{\mathrm{area}(\mathsf{M})}(\mathsf{M} - \vec{x_1}),\end{aligned}$$

which implies the claim. ∎

3.3.1. Constant approximation to the maximum overlap

Lemma 3.10 (constApproxByRect). *Let X and Y be two convex polygons, and let M be the polygon realizing their maximum area intersection under translation. Then, one can compute, in $O(|\mathsf{X}| + |\mathsf{Y}|)$ time, a rectangle r, such that $\mathsf{r} \subseteq \vec{u} + \mathsf{M} \subseteq c_r\mathsf{r}$, where c_r is a constant. That is, one can compute a constant factor approximation to the maximum area overlap in linear time.*

Furthermore, for any translation t_Y, we have that $\mathsf{X} \cap (\mathsf{Y} + \mathsf{t}_\mathsf{Y}) \sqsubseteq c_r\mathsf{r}$.

Proof. We are going to implement the algorithmic proof of Lemma 3.9. Instead of John's ellipsoid we use the rectangle of Lemma 3.1. Clearly, the proof of Lemma 3.9 goes through with the constants being somewhat worse. Specifically, we compute, in linear time, vectors $\vec{x}, \vec{y}$, and rectangles $\mathsf{r}_\mathsf{X}, \mathsf{r}_\mathsf{Y}$, such that $\vec{x} + \mathsf{r}_\mathsf{X} \subseteq \mathsf{X} \subseteq \vec{x} + 5\mathsf{r}_\mathsf{X}$ and $\vec{y} + \mathsf{r}_\mathsf{Y} \subseteq \mathsf{Y} \subseteq \vec{y} + 5\mathsf{r}_\mathsf{Y}$. Again, compute a rectangle r_M, such that $\mathsf{r}_\mathsf{M}/5 \subseteq \mathsf{r}_\mathsf{X} \cap \mathsf{r}_\mathsf{Y} \subseteq \mathsf{r}_\mathsf{M}$. Arguing as in Lemma 3.9, and setting $\mathsf{r} = \mathsf{r}_\mathsf{M}/c_3$, for some constant c_3, is the desired rectangle. ■

4. Approximating the overlap function of convex polygons

Definition 4.1. *Given two convex polygons* X *and* Y *in the plane, of total complexity* n*, and parameters* $\varepsilon \in (0,1)$*,* ν*,* ρ*, a function* $\psi(\mathsf{t})$ *is* **(ε, ν, ρ)-approximation** *of* $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$*, if the following conditions hold:*

(A) $\forall \mathsf{t} \in \mathbb{R}^2$*, we have* $|\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y}) - \psi(\mathsf{t})| \leq \varepsilon \mu_{\max}(\mathsf{X}, \mathsf{Y})$.

(B) There are convex polygons $\mathsf{P}_1, \ldots, \mathsf{P}_\nu$*, each of maximum complexity* ρ*, such that inside every face of the arrangement* $\mathcal{A} = \mathcal{A}(\mathsf{P}_1, \ldots, \mathsf{P}_\nu)$*, the approximation function* $\psi(\mathsf{t})$ *is the same quadratic function.*

That is, the total descriptive complexity of $\psi(\cdot)$ *is the complexity of the arrangement* $\mathcal{A}$*.*

Algorithm 4.2. *The input is two convex polygons* X *and* Y *in the plane, of total complexity* n*, and a parameter* $\varepsilon \in (0,1)$*. As a first step, the algorithm is going to approximate* X *and* Y *as follows:*

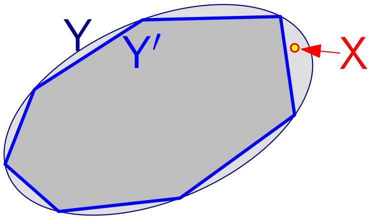
(A) $\mathsf{r}_\mathsf{M} \leftarrow$ **constApproxByRect**(X, Y)*, see Lemma 3.10.*

(B) $\mathcal{T} \leftarrow$ *affine transformation that maps* $2c_r\mathsf{r}_\mathsf{M}$ *to* $[0,1]^2$*.*

(C) $\mathsf{X}'_\mathcal{T} \leftarrow$ **approxPolygon**$(\mathcal{T}(\mathsf{X}), N)$ *and* $\mathsf{Y}'_\mathcal{T} \leftarrow$ **approxPolygon**$(\mathcal{T}(\mathsf{Y}), N)$*.*
See Lemma 3.3, here $N = \lceil c_4/\varepsilon \rceil$*, and* c_4 *is a sufficiently large constant.*

(D) $\mathsf{X}' \leftarrow \mathcal{T}^{-1}(\mathsf{X}'_\mathcal{T})$ *and* $\mathsf{Y}' \leftarrow \mathcal{T}^{-1}(\mathsf{Y}'_\mathcal{T})$*.*

4.1. If one polygon is smaller than the other



Assume, without loss of generality, that X is smaller than Y, that is, X can be translated so that it is entirely contained inside Y (i.e., $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) \leq 1$, see Lemma 2.3). The maximum area of overlap is now equal to $\mathrm{area}(\mathsf{X})$. The challenge is, that for any approximation of Y, we can always have a sufficiently small X which can be placed in $\mathsf{Y} \setminus \mathsf{Y}'$, as shown in the figure on the right. Therefore for all those translations for which X is placed inside $\mathsf{Y} \setminus \mathsf{Y}'$, our approximation will show zero overlap, even though the actual overlap is $\mathrm{area}(\mathsf{X})$.

To get around this problem, we will first approximate the smaller polygon X, using our approximation scheme, to get polygon X', then we will compute level sets of the overlap function and use them to approximate it.

Lemma 4.3. *Given convex polygons* X *and* Y*, such that* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) < 1$*, and parameter* $\varepsilon > 0$*, and let* X' *be the approximation to* X*, as computed by Algorithm 4.2. Then, we have, for all translations* $\mathsf{t} \in \mathbb{R}^2$*, that* $\Bigl|\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\Bigr| \leq \varepsilon \mu_{\max}(\mathsf{X}, \mathsf{Y})$.

Proof. Consider the overlap of $\mathsf{X}_\mathcal{T} = \mathcal{T}(\mathsf{X})$ and $\mathsf{Y}_\mathcal{T} = \mathcal{T}(\mathsf{Y})$. Lemma 3.10 implies that any intersection polygon of $\mathsf{X}_\mathcal{T}$ and $\mathsf{Y}_\mathcal{T}$ can be contained (via translation) in $\mathcal{T}(c_r\mathsf{r}_\mathsf{M})$ (which is a translation of the square $[0, 1/2]^2$). Clearly, in this case, $\mathsf{X}_\mathcal{T}$ and $\mathsf{X}'_\mathcal{T}$ can both be translated to be contained in this square, both contain a disk of constant radius, the maximum distance between $\mathsf{X}_{\mathcal{T}}$ and $\mathsf{X}'_{\mathcal{T}}$ is $O(\varepsilon)$, and the total area of $\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}}$ is $O(\varepsilon)$, as the perimeter of $\mathsf{X}_{\mathcal{T}} \leq 4$. Thus, setting c_4 to be sufficiently large, implies that $\mathrm{area}(\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}}) \leq \varepsilon \mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}})$, as $\mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}}) = \Omega(1)$. This implies that $\left|\mu(\mathsf{X}'_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}}) - \mu(\mathsf{X}_{\mathcal{T}}, \mathsf{t} + \mathsf{Y}_{\mathcal{T}})\right| \leq \varepsilon \mu_{\max}(\mathsf{X}_{\mathcal{T}}, \mathsf{Y}_{\mathcal{T}})$, which implies the claims by applying $\mathcal{T}^{-1}$ to both sides. ∎

Therefore, $\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y})$ is a good approximation for $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$. However, $\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y})$ has complexity $O\left(|\mathsf{X}'|^2 |\mathsf{Y}|^2\right)$ [dBCD+98], in the worst case, which is still too high.

Lemma 4.4 (approxLevelSet). *Given two convex polygons* X *and* Y*, of total complexity* n*, and a parameter* ε*, such that* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) < 1$*, then one can construct in* $O\left(n/\varepsilon^2\right)$ *time, a* $(\varepsilon, O(1/\varepsilon^2), O(n/\varepsilon^2))$*-approximation* $\psi(\cdot)$ *to* $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$*.*

Proof. There is a translation of X such that it is contained completely in Y. Approximate X from the outside by a rectangle r, using Lemma 3.1. Next, spread a grid in r by partitioning each of its edges into $O(1/\varepsilon)$ equal length intervals. Let S be the set of points of the grid that are in X. It is easy to verify, that for any convex body Z and a translation t, we have

$$\left|\mu(\mathsf{X}, \mathsf{t} + \mathsf{Z}) - \frac{|(\mathsf{t} + \mathsf{Z}) \cap \mathsf{S}|}{|\mathsf{S}|}\right| \leq \varepsilon\, \mathrm{area}(\mathsf{X})$$

Namely, to approximate the overlap area for $\mathsf{t} + \mathsf{Y}$, we need to count the number of points of S that it covers. To this end, for each point $\mathsf{p} \in \mathsf{S}$, we generate a 180° rotated and translated copy of Y, denoted by Y'_{p}, such that $\mathsf{p} \in \mathsf{t} + \mathsf{Y}$ if and only if $\mathsf{t} \in \mathsf{Y}'_{\mathsf{p}}$.

Clearly, the generated set of polygons is the desired $(\varepsilon, O(1/\varepsilon^2), O(n/\varepsilon^2))$-approximation $\psi(\cdot)$ to $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$.

The time to build this approximation is $O(n/\varepsilon^2)$. ∎

We next describe a slightly slower algorithm that generates a slightly better approximation.

Lemma 4.5 (approxLevelSet). *Given two convex polygons* X *and* Y*, of total complexity* n*, and a parameter* ε*, such that* $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) < 1$*, then one can construct in* $O\left(\varepsilon^{-2} n \log n\right)$ *time, a* $(\varepsilon, O(1/\varepsilon), O(n/\varepsilon))$*-approximation* $\psi(\cdot)$ *to* $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$*.*

Proof. We compute X', as above with approximation parameter $\epsilon = \varepsilon/4$. Next, using the algorithm of [dBCD+98], we compute $\mu_{\max}(\mathsf{X}', \mathsf{Y})$ in $O(n' \log n') = O(n \log n)$ time, where $n' = |\mathsf{X}'| + |\mathsf{Y}| = O(n)$.

We approximate the function $\mu(\mathsf{X}', \mathsf{Y} + \mathsf{t})$ by constructing the α_i-slices, where $\alpha_i = \min(1, (i + 1)\epsilon)\mu_{\max}(\mathsf{X}', \mathsf{Y})$, for $i = 0, 1, \ldots, M$, where $M = \lceil 1/\epsilon \rceil$. To this end, we deploy the algorithm of Lemma 3.5 for each slice, which takes $O(m \log m)$ time, where $m = |\mathsf{X}'|\,|\mathsf{Y}| = O(n/\varepsilon)$.

Let us denote the ith region constructed, by S_i, that is, S_i is a convex figure whose boundary corresponds to all translations t such that $\mu(\mathsf{X}', \mathsf{Y} + \mathsf{t}) = \alpha_i$. Clearly, S_{i+1} lies entirely within S_i. Given the description of S_i's, for any translation t, we define

$$\psi(\mathsf{t}) = \begin{cases} \alpha_i & \mathsf{t} \in \mathsf{S}_i \setminus \mathsf{S}_{i+1} \\ 0 & \mathsf{t} \notin \mathsf{S}_0. \end{cases}$$

It is now straightforward to verify this is the desired approximation. Indeed, for $\mathsf{t} \notin \mathsf{S}_0$, we have by Lemma 4.3, that

$$\begin{aligned} \left|\psi(\mathsf{t}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\right| &\leq \left|\psi(\mathsf{t}) - \mu(\mathsf{X}', \mathsf{t} + \mathsf{Y})\right| + \left|\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\right| \leq \alpha_0 + \epsilon \mu_{\max}(\mathsf{X}, \mathsf{Y}) \\ &= \epsilon(\mu_{\max}(\mathsf{X}', \mathsf{Y}) + \mu_{\max}(\mathsf{X}, \mathsf{Y})) \leq 2\epsilon \mu_{\max}(\mathsf{X}, \mathsf{Y}) = (\varepsilon/2)\mu_{\max}(\mathsf{X}, \mathsf{Y}), \end{aligned}$$

as desired. Similarly, if $\mathsf{t} \in \mathsf{S}_i \setminus \mathsf{S}_{i+1}$ then $\alpha_i \leq \mu(\mathsf{X}', \mathsf{t}+\mathsf{Y}) \leq \alpha_{i+1}$, and

$$\begin{aligned}\left|\mu'(\mathsf{t}) - \mu(\mathsf{X}, \mathsf{t}+\mathsf{Y})\right| &\leq \left|\mu'(\mathsf{t}) - \mu(\mathsf{X}', \mathsf{t}+\mathsf{Y})\right| + \left|\mu(\mathsf{X}', \mathsf{t}+\mathsf{Y}) - \mu(\mathsf{X}, \mathsf{t}+\mathsf{Y})\right| \\ &\leq \alpha_{i+1} - \alpha_i + \epsilon\mu_{\max}(\mathsf{X}, \mathsf{Y}) = \epsilon\mu_{\max}(\mathsf{X}', \mathsf{Y}) + \epsilon\mu_{\max}(\mathsf{X}, \mathsf{Y}) \\ &\leq (\varepsilon/2)\mu_{\max}(\mathsf{X}, \mathsf{Y})\,.\end{aligned}$$

Clearly, this function is defined by an onion like set of $\nu = O(1/\varepsilon)$ polygons, and the maximum complexity of these polygons is $\rho = O(m) = O(n/\varepsilon)$.

The overall running time is dominated by computing the slices, which takes overall $O(\nu m \log m)$ $= O(n\varepsilon^{-2} \log n\varepsilon^{-1})$ time. ■

4.2. If the two polygons are incomparable

The more interesting case, is when the maximum intersection of X and Y is significantly smaller than both polygons; that is, $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) \geq 1$ and $\mathrm{ssim}(\mathsf{Y}, \mathsf{X}) \geq 1$. Surprisingly, in this case, we can approximate both polygons simultaneously.

Lemma 4.6. *Given convex polygons X and Y, such that $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) \geq 1$ and $\mathrm{ssim}(\mathsf{Y}, \mathsf{X}) \geq 1$, then the widths of $\mathsf{X}_{\mathcal{T}} = \mathcal{T}(\mathsf{X})$ and $\mathsf{Y}_{\mathcal{T}} = \mathcal{T}(\mathsf{Y})$, as computed by Algorithm 4.2, are bounded by 7.*

Proof. Let $\omega_{\mathsf{X}} = \mathrm{width}(\mathsf{X}_{\mathcal{T}})$ and $\omega_{\mathsf{Y}} = \mathrm{width}(\mathcal{T}(\mathsf{Y}))$. By Lemma 2.2, we have that $\mathrm{disk}(\omega_{\mathsf{X}}/3.5) \sqsubseteq \mathsf{X}_{\mathcal{T}}$ and $\mathrm{disk}(\omega_{\mathsf{Y}}/3.5) \sqsubseteq \mathsf{Y}_{\mathcal{T}}$.

So, assume for the sake of contradiction, that $\omega_{\mathsf{X}} \geq 7$. This implies that $\mathsf{X}_{\mathcal{T}}$ contains a disk of radius 2, which in turn contains the unit square. In particular, let $\ell = \mathrm{diameter}(\mathsf{Y}_{\mathcal{T}})$, if $\ell < 2$, then $\mathsf{Y}_{\mathcal{T}}$ is contained in a disk of radius 2, implies that $\mathsf{Y}_{\mathcal{T}} \sqsubseteq \mathsf{X}_{\mathcal{T}}$, a contradiction.

Otherwise, if $\ell > 2$ then there is a translation of $\mathsf{Y}_{\mathcal{T}}$ such that its intersection with $\mathsf{Y}_{\mathcal{T}}$ has length > 2 (indeed, consider the segment realizing the diameter of $\mathsf{Y}_{\mathcal{T}}$, and translate it so its middle point is in the center of the disk of radius 2 inside $\mathsf{X}_{\mathcal{T}}$). But then, this intersection is not contained in $[0, 1]^2$ under any translation, which contradicts Lemma 3.10.

The case $\omega_{\mathsf{Y}} \geq 7$ is handled in a similar fashion. ■

Lemma 4.7. *Given two convex polygons X and Y, of total complexity n, and a parameter ε, such that $\mathrm{ssim}(\mathsf{X}, \mathsf{Y}) \geq 1$ and $\mathrm{ssim}(\mathsf{Y}, \mathsf{X}) \geq 1$, then one can construct in $O\big(n + 1/\varepsilon^2\big)$ time, a $(\varepsilon, O(1/\varepsilon), O(1/\varepsilon))$-approximation $\psi(\cdot)$ to $\mu(\mathsf{X}, \mathsf{t}+\mathsf{Y})$.*

Proof. Use Algorithm 4.2 to compute X' and Y', both of complexity $O(1/\varepsilon)$. We set $\psi(\mathsf{t}) = \mu(\mathsf{X}', \mathsf{t}+\mathsf{Y}')$. [dBCD+98] describes how to describe the function $\mu(\mathsf{X}', \mathsf{t}+\mathsf{Y}')$ as an arrangement of $O(|\mathsf{X}'| + |\mathsf{Y}'|)$ polygons, each of complexity $|\mathsf{X}'|$ or $|\mathsf{Y}'|$. Thus, this result in the desired approximation, that has total complexity $O(1/\varepsilon^2)$, and that can be computed in $O(n + 1/\varepsilon^2)$ time.

In the following, we use the notation of Algorithm 4.2: $\mathsf{X}_{\mathcal{T}} = \mathcal{T}(\mathsf{X})$ and $\mathsf{Y}_{\mathcal{T}} = \mathcal{T}(\mathsf{Y})$, $\mathsf{X}'_{\mathcal{T}} =$ **approxPolygon**$(\mathcal{T}(\mathsf{X}), N)$, and $\mathsf{Y}'_{\mathcal{T}} =$ **approxPolygon**$(\mathcal{T}(\mathsf{X}), N)$.

Lemma 3.10 implies that any intersection polygon of $\mathsf{X}_{\mathcal{T}}$ and $\mathsf{Y}_{\mathcal{T}}$ can be contained in $\mathcal{T}(c_r \mathsf{r}_{\mathsf{M}})$. The error due to approximation of $\mathsf{X}_{\mathcal{T}}$, is $\mathrm{area}(\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}})$. The part of this error that can contribute to the area of overlap, is bounded by portion of $\mathsf{X}_{\mathcal{T}} \setminus \mathsf{X}'_{\mathcal{T}}$, which can be included inside $\mathcal{T}(c_r \mathsf{r}_{\mathsf{M}})$.

The length of the boundary of $\mathsf{X}'_{\mathcal{T}}$, which can be placed inside $\mathcal{T}(c_r \mathsf{r}_{\mathsf{M}})$, is bounded by the perimeter of $\mathcal{T}(c_r \mathsf{r}_{\mathsf{M}})$. Also, our approximation scheme (Lemma 3.3) ensures that the distance between $\mathsf{X}_{\mathcal{T}}$ and $\mathsf{X}'_{\mathcal{T}}$ (along the direction of shortest diameter) is bounded by $\omega(\mathsf{X}_{\mathcal{T}})\,/N$, where N is a chosen parameter.

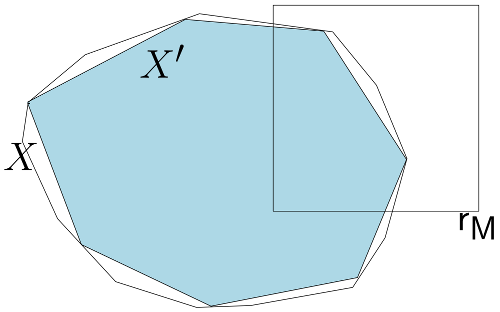


Figure 4.1: Error bound.

Consider any fixed t – for the sake of simplicity of exposition, we assume the intersection $\mathsf{X}_\mathcal{T} \cap (\mathsf{t} + \mathsf{Y}_\mathcal{T})$ as being inside the square $\mathcal{T}(c_r \mathsf{r_M})$ (if not, we can translate $\mathcal{T}(c_r \mathsf{r_M})$ so this assumption holds). As $N = \lceil c_4/\varepsilon \rceil$, and for c_4 sufficiently large constant, the error due to the approximation of $\mathsf{X}_\mathcal{T}$ is bounded by

$$\begin{aligned}\mathcal{E}_\mathsf{X} &= \Big|\mu(\mathsf{X}'_\mathcal{T}, \mathsf{t} + \mathsf{Y}_\mathcal{T}) - \mu(\mathsf{X}_\mathcal{T}, \mathsf{t} + \mathsf{Y}_\mathcal{T})\Big| = \Big|\mu(\mathsf{X}_\mathcal{T} \setminus \mathsf{X}'_\mathcal{T}, \mathsf{t} + \mathsf{Y}_\mathcal{T})\Big| \\ &\leq \text{perimeter}\Big(\mathsf{X}_\mathcal{T} \cap (\mathsf{t} + \mathsf{Y}_\mathcal{T})\Big) \times \frac{\omega(\mathsf{X}_\mathcal{T})}{N} \leq \text{Perimeter}(\mathcal{T}(c_r \mathsf{r_M})) \times \frac{\omega(\mathsf{X}_\mathcal{T})}{N} \\ &\leq 4 \times \frac{7}{N} \leq \frac{\varepsilon}{4} \mu_{\max}(\mathsf{X}_\mathcal{T}, \mathsf{Y}_\mathcal{T}) \,,\end{aligned}$$

by Lemma 4.6, and since by Lemma 3.10 we have $\mu_{\max}(\mathsf{X}_\mathcal{T}, \mathsf{Y}_\mathcal{T}) = \Omega(1)$. A symmetric argument works for the error $\mathcal{E}_\mathsf{Y}$ of the overlap caused by the approximation of $\mathsf{Y}_\mathcal{T}$ by $\mathsf{Y}'_\mathcal{T}$. We conclude that

$$\begin{aligned}\mathcal{E}_\mathcal{T} &= \Big|\mu(\mathsf{X}'_\mathcal{T}, \mathsf{t} + \mathsf{Y}'_\mathcal{T}) - \mu(\mathsf{X}_\mathcal{T}, \mathsf{t} + \mathsf{Y}_\mathcal{T})\Big| \\ &\leq \Big|\mu(\mathsf{X}'_\mathcal{T}, \mathsf{t} + \mathsf{Y}'_\mathcal{T}) - \mu(\mathsf{X}_\mathcal{T}, \mathsf{t} + \mathsf{Y}'_\mathcal{T})\Big| + \Big|\mu(\mathsf{X}_\mathcal{T}, \mathsf{t} + \mathsf{Y}'_\mathcal{T}) - \mu(\mathsf{X}_\mathcal{T}, \mathsf{t} + \mathsf{Y}_\mathcal{T})\Big| \\ &\leq \Big|\mu(\mathsf{X}'_\mathcal{T}, \mathsf{t} + \mathsf{Y}_\mathcal{T}) - \mu(\mathsf{X}_\mathcal{T}, \mathsf{t} + \mathsf{Y}_\mathcal{T})\Big| + \mathcal{E}_\mathsf{Y} \leq \mathcal{E}_\mathsf{X} + \mathcal{E}_\mathsf{Y} \leq \frac{\varepsilon}{2} \mu_{\max}(\mathsf{X}_\mathcal{T}, \mathsf{Y}_\mathcal{T}) \,,\end{aligned}$$

since $\mathsf{Y}'_\mathcal{T} \subseteq \mathsf{Y}_\mathcal{T}$. By applying $\mathcal{T}^{-1}$ to the above, we get that for any translation t, it holds

$$\mathcal{E} = \Big|\psi(\mathsf{t}) - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\Big| = \Big|\mu(\mathsf{X}', \mathsf{t} + \mathsf{Y}') - \mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})\Big| \leq \frac{\varepsilon}{2} \mu_{\max}(\mathsf{X}, \mathsf{Y}) \,. \qquad \blacksquare$$

4.2.1. The result.

By combining Lemma 4.4 and Lemma 4.7 (deciding which one to apply can be done by computing $\text{ssim}(\mathsf{X}, \mathsf{Y})$ and $\text{ssim}(\mathsf{Y}, \mathsf{X})$, which takes $O(n)$ time), we get the following.

Lemma 4.8. *Given two convex polygons* X *and* Y*, of total complexity* n*, and a parameter* ε*, one can construct in* $O\big(n/\varepsilon^2\big)$ *time, a* $(\varepsilon, O(1/\varepsilon^2), O(n/\varepsilon^2))$*-approximation* $\psi(\cdot)$ *to* $\mu(\mathsf{X}, \mathsf{t} + \mathsf{Y})$*.*

5. Approximating the Maximum Overlap of Polygons

The input is two polygons P and Q in the plane, of total complexity n, each of them can be decomposed into at most k convex polygons. Our purpose is to find the translation that maximizes the area of overlap.

The Algorithm. We decompose the polygons P and Q into minimum number of interior disjoint convex polygons [KS02], in time $O(n + k^2 \min(k^2, n))$ (some of these convex polygons can be empty). Then, for every pair P_i, Q_j, we compute an $(\epsilon, O(1/\epsilon^2), O(n/\epsilon^2))$-approximation ψ_{ij} to the overlap function of P_i and Q_j, using Lemma 4.8, where $\epsilon = \varepsilon/k^2$.

Next, as each function ψ_{ij} is defined by an arrangement defined by $O(1/\epsilon^2)$ polygons, we overlay all these arrangements together, and compute for each face of the arrangement the function $\psi = \sum_{i,j} \psi_{ij}$. Inside such a face this function is the same, and it is a quadratic function. We then find the global maximum of this function, and return it as the desired approximation.

Analysis – Quality of approximation. For any translation t, we have that

$$\begin{aligned}\left|\mu(\mathsf{P}, \mathsf{t} + \mathsf{Q}) - \psi(\mathsf{t})\right| &\leq \sum_{i=1}^{k}\sum_{j=1}^{k} \left|\mu(\mathsf{P}_i, \mathsf{t} + \mathsf{Q}_j) - \psi_{ij}(\mathsf{t})\right| \leq \sum_{i=1}^{k}\sum_{j=1}^{k} \epsilon\mu_{\max}(\mathsf{P}_i, \mathsf{Q}_j) \leq \epsilon k^2 \mu_{\max}(\mathsf{P}, \mathsf{Q})\\ &\leq \varepsilon\mu_{\max}(\mathsf{P}, \mathsf{Q})\,.\end{aligned}$$

Analysis – Running time. Computing each of the k^2 approximation function, takes $O\big((k/\varepsilon)^2 n\big)$ time. Each one of them is a $(\varepsilon/k, O(k^2/\varepsilon^2), O(k^2 n/\varepsilon^2))$-approximation, which means that the final arrangement is the overlay of $O(k^4/\varepsilon^2)$ convex polygons, each of complexity $O(k^2 n/\varepsilon^2)$. In particular, any pair of such polygons can have at most $O(k^2 n/\varepsilon^2)$ intersection points, and thus the overall complexity of the arrangement of these polygons is $N = O\Big(\big(k^4/\varepsilon^2\big)^2 \big(k^2 n/\varepsilon^2\big)\Big) = O(k^{10}\varepsilon^{-6} n)$. Computing this arrangement can be done by a standard sweeping algorithm. Observing that every vertical line crosses only $O(k^4/\varepsilon^2)$ segments, imply that the sweeping can be done in $O(\log(k/\varepsilon))$ time per operation, which implies that the overall running time is

$$O\left(k^2 \frac{k^2}{\varepsilon^2} n + N \log \frac{k}{\varepsilon}\right) = O\left(\frac{k^{10}}{\varepsilon^6} n \log \frac{k}{\varepsilon}\right).$$

The result.

Theorem 5.1. *Given two simple polygons P and Q of total complexity n, one can compute a translation which ε-approximates the maximum area of overlap of P and Q. The time required is $O(c'n)$ where $c' = \dfrac{k^1 0}{\varepsilon^6} \log \dfrac{k}{\varepsilon}$, where k is the minimum number of convex polygons in the decomposition of P and Q.*

More specifically, one gets a data-structure, such that for any query translation t, one can compute, in $O(\log n)$ time, an approximation $\psi(\mathsf{t})$, such that $|\psi(\mathsf{t}) - \mu(\mathsf{P}, \mathsf{Q})| \leq \varepsilon\mu_{\max}(\mathsf{P}, \mathsf{Q})$, where $\mu_{\max}(\mathsf{P}, \mathsf{Q})$ is the maximum area of overlap between P and Q.

Note, that our analysis is far from tight. Specifically, for the sake of simplicity of exposition, it is loose in several places as far as the dependency on k and ε.

Acknowledgments

The authors would like to thank the anonymous referees for their insightful comments. In particular, the improved construction of Lemma 4.4 was suggested by an anonymous referee.

References

[ABT$^+$96] D. Avis, P. Bose, G. T. Toussaint, T. C. Shermer, B. Zhu, and J. Snoeyink. On the sectional area of convex polytopes. In *Proc. 12th Annu. Sympos. Comput. Geom.* (SoCG), pages 411–412, 1996.

[ACKY14] H.-K. Ahn, S.-W. Cheng, H. J. Kweon, and J. Yon. Overlap of convex polytopes under rigid motion. *Comput. Geom. Theory Appl.*, 47(1):15–24, 2014.

[ACP$^+$07] H.-K. Ahn, O. Cheong, C.-D. Park, C.-S. Shin, and A. Vigneron. Maximizing the overlap of two planar convex sets under rigid motions. *Comput. Geom. Theory Appl.*, 37(1):3–15, 2007.

[ACR13] H.-K. Ahn, S.-W. Cheng, and I. Reinbacher. Maximum overlap of convex polytopes under translation. *Comput. Geom. Theory Appl.*, 46(5):552–565, 2013.

[AFRW98] H. Alt, U. Fuchs, G. Rote, and G. Weber. Matching convex shapes with respect to the symmetric difference. *Algorithmica*, 21:89–103, 1998.

[AG00] H. Alt and L. J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 121–153. Elsevier, 2000.

[BH01a] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38:91–109, 2001.

[BH01b] G. Barequet and S. Har-Peled. Polygon containment and translational min-hausdorff-distance between segment sets are 3sum-hard. *Internat. J. Comput. Geom. Appl.*, 11(4):465–474, 2001.

[BJ02] G. S. Brodal and R. Jacob. Dynamic planar convex hull. In *Proc. 43th Annu. IEEE Sympos. Found. Comput. Sci.* (FOCS), pages 617–626, 2002.

[CEH07] O. Cheong, A. Efrat, and S. Har-Peled. On finding a guard that sees most and a shop that sells most. *Discrete Comput. Geom.*, 37(4):545–563, 2007.

[CL13] S.-W. Cheng and C.-K. Lam. Shape matching under rigid motion. *Comput. Geom. Theory Appl.*, 46(6):591–603, 2013.

[CLM05] B. Chazelle, D. Liu, and A. Magen. Sublinear geometric algorithms. *SIAM J. Comput.*, 35(3):627–646, 2005.

[dBCD$^+$98] M. de Berg, O. Cheong, O. Devillers, M. van Kreveld, and M. Teillaud. Computing the maximum overlap of two convex polygons under translations. *Theo. Comp. Sci.*, 31:613–628, 1998.

[dBCG$^+$04] M. de Berg, S. Cabello, P. Giannopoulos, C. Knauer, R. van Oostrum, and R. C. Veltkamp. Maximizing the area of overlap of two unions of disks under rigid motion. In *Scandinavian Workshop on Algorithms*, volume 3111 of *LNCS*, pages 138–149, 2004.

[GK92] P. Gritzmann and V. Klee. Inner and outer j-radii of convex bodies in finite-dimensional normed spaces. *Discrete Comput. Geom.*, 7:255–280, 1992.

[Har00] S. Har-Peled. Taking a walk in a planar arrangement. *SIAM J. Comput.*, 30(4):1341–1367, 2000.

[Har11] S. Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Mathematical Surveys and Monographs*. Amer. Math. Soc., 2011.

[KS02] J. M. Keil and J. Snoeyink. On the time bound for convex decomposition of simple polygons. *Internat. J. Comput. Geom. Appl.*, 12(3):181–192, 2002.

[MSW96] D. M. Mount, R. Silverman, and A. Y. Wu. On the area of overlap of translated polygons. *Computer Vision and Image Understanding: CVIU*, 64(1):53–61, 1996.

[ST94] M. Sharir and S. Toledo. Extremal polygon containment problems. *Comput. Geom. Theory Appl.*, 4:99–118, 1994.

[Tou83] G. T. Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE MELECON '83*, pages A10.02/1–4, 1983.

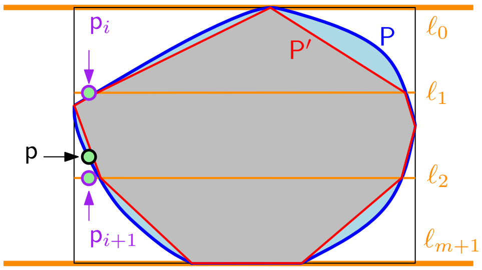
[Vig14] A. Vigneron. Geometric optimization and sums of algebraic functions. *ACM Trans. Algo.*, 10(1):4:1–4:20, 2014.

A. Proof of Lemma 3.3

Lemma 3.3 follows by the work of Ahn *et al.* [ACP+07]. For the sake of completeness we include the details here (our construction is somewhat different).

Proof. The width of P is realized by two parallel lines ℓ_0 and ℓ_{m+1} at a distance of at most $\omega(\mathsf{P})$ from each other. These two lines can be computed in linear time using rotating caliper [Tou83] (and one of them must pass through an edge of P). Similarly, it is easy to compute the two extreme points of P in the direction of ℓ. Adding the supporting lines to P through these points that are orthogonal to ℓ, results in a rectangle, that has five vertices of P on its boundary, and we mark these vertices. Next, we slice this rectangle by m parallel lines $\ell_1, \ldots, \ell_m$ into $m+1$ slices of the same width. We mark all the intersections of these lines with P. Next, let P' be the polygon formed by the convex-hull of all the marked points. It is easy to verify that P' can be computed in linear time.

Clearly, $\mathsf{P}' \subseteq \mathsf{P}$. For any point $\mathsf{p} \in \mathsf{P}$, consider the two lines ℓ_i and ℓ_{i+1} that p is contained between them. Consider the projection of p into ℓ_i and ℓ_{i+1}, denoted by p_i and p_{i+1} respectively. If any of these two projections are inside P, then we are done, as this portion of P is inside P', and the distance of projection is at most $\omega(\mathsf{P})/m$. Thus, the only remaining possibility is that both projections are outside P. But this implies that P's extreme point in the parallel direction to ℓ (or its reverse) must be in p's slice. In particular, the segment $\mathsf{p}_i\mathsf{p}_{i+1}$ (that includes p) must intersect P, and furthermore, since the aforementioned extreme point is a vertex of P', it follows that this segments must intersect P', implying the claim. ■



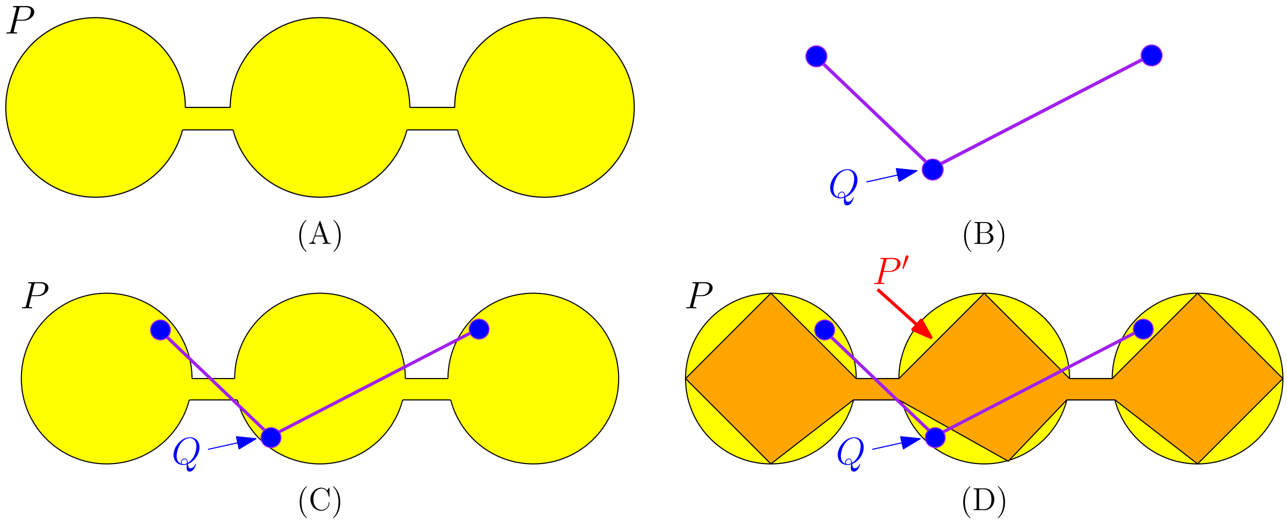


Figure B.1: Counterexample to naive approach for approximating overlap. (A), (B) Input polygons. (C) Optimal placement. (D) Approximate polygons fails to preserve optimal solution.

B. Counter example for some natural approaches

> "I have not failed. I've just found 10,000 ways that won't work." – Thomas Edison

Here we present a counterexample to the natural for our problem: Decompose the two polygons into their convex parts, approximate the parts, and then find the maximum overlap for the two reconstituted polygons.

So, consider the polygon P, depicted in Figure B.1 (A), and its arch-nemesis Q in Figure B.1 (B). The optimal overlap placement is depicted in Figure B.1 (C).

But any naive piecewise approximation of P by pieces will fail, because it creates cavities that might contain the whole overlap. For example, one possible solution for the approximate polygon P' and the maximum overlap, is depicted in Figure B.1 (D).

In particular, any placement that makes the approximate polygon P' cover two of the disks of Q, would fail to cover the third disk of Q. so, in this case, the best approximation suggested by the reviewer is (roughly) 2/3. This can be made to be arbitrarily bad by enlarging k.

Another approach that does not work. Another natural idea is to pick from each pair of polygons P_i and Q_j a set of directions such that if you approximate each pair with these directions than the pair overlap is approximated correctly. Note, however, that if the polygons are of completely different sizes then, inherently, you need unbounded number of directions if the maximum overlap of the original (non-convex polygons) align the pieces far from their piecewise overlap maximum.