Available online at www.sciencedirect.com

ScienceDirect

Computational Geometry 40 (2008) 171–177

Computational Geometry
Theory and Applications

www.elsevier.com/locate/comgeo

Maximum overlap and minimum convex hull of two convex polyhedra under translations ☆

Hee-Kap Ahn [a,*], Peter Brass [b], Chan-Su Shin [c,1]

[a] *Department of Computer Science and Engineering, POSTECH, Pohang, South Korea*
[b] *Department of Computer Science, City College, New York, USA*
[c] *School of Electrical and Information Engineering, Hankuk University of Foreign Studies, Yongin, South Korea*

Received 26 February 2007; received in revised form 6 August 2007; accepted 7 August 2007

Available online 9 October 2007

Communicated by M. de Berg

Abstract

Given two convex polyhedra P and Q in three-dimensional space, we consider two related problems of shape matching: (1) finding a translation $t_1 \in \mathbb{R}^3$ of Q that maximizes the volume of their overlap $P \cap (Q + t_1)$, and (2) finding a translation $t_2 \in \mathbb{R}^3$ that minimizes the volume of the convex hull of $P \cup (Q + t_2)$. For the maximum overlap problem, we observe that the dth root of the objective function is concave and present an algorithm that computes the optimal translation in expected time $O(n^3 \log^4 n)$. This method generalizes to higher dimensions $d > 3$ with expected running time $O(n^{d+1-\frac{3}{d}} (\log n)^{d+1})$. For the minimum convex hull problem, we show that the objective function is convex. The same method used for the maximum overlap problem can be applied to this problem and the optimal translation can be computed in the same time bound.

Keywords: Shape matching; Similarity; Convex polyhedron; Maximum overlap; Minimum convex hull

1. Introduction

In geometric shape matching, we are given two geometric objects and decide how much they resemble each other. In most cases, we first choose a good similarity measure that reflects the actual similarity. Then we transform one of two objects over the other, by scaling, translation, and rotation, and compute their similarity based on the measure. A typical example is that we have a shape database, and we are given a query shape, asking "find the best matched shape".

In this paper, we consider two related problems of shape matching: how to find, for two given convex polyhedra, the translation that maximizes their overlap, or the translation that minimizes their convex hull. Both questions can be

☆ This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD, Basic Research Promotion Fund) (KRF-2006-331-D00456).

* Corresponding author.
E-mail addresses: heekap@postech.ac.kr (H.-K. Ahn), peter@cs.ccny.cuny.edu (P. Brass), cssin@hufs.ac.kr (C.-S. Shin).

[1] The work by Shin was supported by Research Grant 2007, Hankuk University of Foreign Studies.

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2007.08.001

seen as different ways to formalize the concept of placing the polyhedra atop each other in such a way that they match best.

We are given two convex polyhedra P and Q in three-dimensional space, with n vertices in total, and want to find the translation $t \in \mathbb{R}^3$, that maximizes the volume function of the overlap

$$f(t) = \mathrm{Vol}\big(P \cap (Q + t)\big).$$

Our main result is

Theorem 1. *Given two convex polyhedra P and Q with n vertices in three-dimensional space, we can compute the translation vector t of Q that maximizes*

$$\mathrm{Vol}\big(P \cap (Q + t)\big)$$

in expected time

$$\mathrm{O}(n^3 \log^4 n).$$

The same problem has already been solved for two-dimensional convex polygons by de Berg et al. [3] in $\mathrm{O}(n \log n)$ time. Without the convexity assumption, this problem was studied by Mount et al. [10], who gave for simple polygons or sets of n triangles an $\mathrm{O}(n^4)$-method to find the maximum overlap under translations, by constructing the arrangement in translation space, and the intersection area function on each cell. That method generalizes to $\mathrm{O}(n^{2d})$ for two sets of n simplices in d-dimensional space. This was recently rediscovered and extended to the intersection of k simple polygons by Fukuda and Uno [9]. They also observed that the concavity of the dth root of the intersection volume function allows them to use the machinery of convex optimization [4]. For the special case of convex polytopes, our method generalizes to higher dimensions as in the following.

Theorem 2. *Given two convex polytopes P and Q of total complexity n in d-dimensional space, we can compute the translation vector t that maximizes*

$$\mathrm{Vol}\big(P \cap (Q + t)\big)$$

in expected time

$$\mathrm{O}\big(n^{d+1-\frac{3}{d}} (\log n)^{d+1}\big).$$

Here the total complexity of the polytope is the sum of the numbers of faces of all dimensions, or the number of simplices necessary in a triangulation. Without the restriction to translations, we do not know of any algorithm to exactly compute the maximum overlap of, e.g., two sets of triangles under rigid motions. Ahn et al. [2] recently gave a fast approximation algorithm to find a rigid motion such that the overlap of two planar convex sets is maximized.

For convex sets, we can measure the quality of fit of our matching also by minimizing the convex hull, to see how much sticks out in an optimal placement. Here we want to minimize the function

$$g(t) = \mathrm{Vol}\big(\mathrm{conv}\big(P \cup (Q + t)\big)\big).$$

The same techniques can be applied to this problem, and give the following result.

Theorem 3. *Given two convex polyhedra P and Q in three-dimensional space, we can compute the translation vector t of Q that minimizes*

$$\mathrm{Vol}\big(\mathrm{conv}\big(P \cup (Q + t)\big)\big)$$

in expected time

$$\mathrm{O}(n^3 \log^4 n).$$

The d-dimensional problem can be solved in expected time

$$\mathrm{O}\big(n^{d+1-\frac{3}{d}} (\log n)^{d+1}\big).$$

The two-dimensional version of this problem for two convex polygons has been studied by Ahn and Cheong [1]: with a fixed orientation they proposed an exact near-linear time algorithm for finding an optimal translation, and without the restriction to translations they gave a fast algorithm to find a rigid motion that gives $(1+\varepsilon)$-approximation to the optimal. The problem with multiple convex polygons finds an application in a lower bound for Lebesgue's universal cover problem [5].

2. Maximizing the overlap under translation

We first consider the problem of translating one convex polyhedron over the other, such that the overlap of two polyhedra is maximized. More precisely, given two convex polyhedra P and Q with n vertices in three-dimensional space, let

$$f(t) = \mathrm{Vol}\big(P \cap (Q+t)\big)$$

be the volume function of the overlap for a vector $t \in \mathbb{R}^3$. Our goal is to find a vector t where $f(t)$ is maximized. Once t is fixed, we can evaluate $f(t)$ in linear time [6]. Indeed, we can construct the intersection polyhedron $P \cap (Q+t)$ for given t in $O(n)$ time; triangulate it, and compute the volume of the $O(n)$ simplices. Since these volume functions are cubic polynomials in the coordinates, as can be seen from the determinant expression for the volume of a simplex, this volume function $f(t)$ is locally a cubic function, in some neighborhood of t for which the combinatorial structure of the intersection polyhedron $P \cap (Q+t)$ does not change.

A fundamental property of the intersection volume function $f(t)$ is that $(f(t))^{1/3}$ is concave on its domain $D = P - Q$. We say a function is concave if the volume below the graph in its domain is a convex set. This is a known consequence of the Brunn–Minkowski theorem on an inequality for mixed volumes in $\mathbb{R}^d$ [11,12].

Lemma 4. *Let P and Q be convex polytopes in $\mathbb{R}^d$. Then the function $t \to f(t)^{1/d}$ is concave in the domain $D = P - Q$.*

Let $Q_t := Q + t$, and imagine that we translate Q in three-dimensional space. Then we encounter events of the following three types that might result in different combinatorial structures of $P \cap Q_t$:

1. a vertex of P hits a facet of Q_t,
2. a vertex of Q_t hits a facet of P, or
3. an edge of P crosses an edge of Q_t.

Fig. 1 shows events of types 2 and 3: a vertex of Q_t hits a facet of P in the left of Fig. 1(a), and an edge of P crosses an edge of Q_t in the left of Fig. 1(b). If we translate Q_t upward slightly, the combinatorial structures of $P \cap Q_t$ change as in the right ones of Fig. 1(a) and (b).

Each of these events defines a plane in three-dimensional translation space, called an *event plane*. These $O(n^2)$ event planes partition the space into $O(n^6)$ polyhedral cells. The trivial method would be to construct this arrangement, and for each cell of the arrangement compute the function $f(t)$, and find the maximum over all that function pieces. Function evaluations in $O(n)$ time lead to an $O(n^7)$-time algorithm; this could be improved to $O(n^6)$ time if we use the coherence of the functions between the cells. We will present a much faster algorithm, using the concavity of the function $f(t)^{1/3}$, and exploiting the special structure of the event planes in the translation space.

The special structure of the event planes that we use is that they can be grouped into $O(n)$ groups, where within a group, either all planes are parallel, or all planes share a common line. Consider the three types of events described above. For a fixed facet of P, the $O(n)$ vertices of Q_t define $O(n)$ parallel planes of event type 2. In the same way, for a facet of Q_t, the $O(n)$ vertices of P define $O(n)$ parallel planes of event type 1. Therefore the events of types 1 and 2 can be grouped into $O(n)$ groups of $O(n)$ parallel planes each. For events of type 3, an edge e of P can cross $O(n)$ edges of Q_t, which define $O(n)$ planes around the common line induced by e. So the events of type 3 can be grouped into $O(n)$ groups of $O(n)$ planes around a common line.

As a first step of our algorithm, we generate all the event planes, and sort them independently within their groups. In each group, the planes are stored in an array in their sorted sequence. This step takes $O(n^2 \log n)$ time.

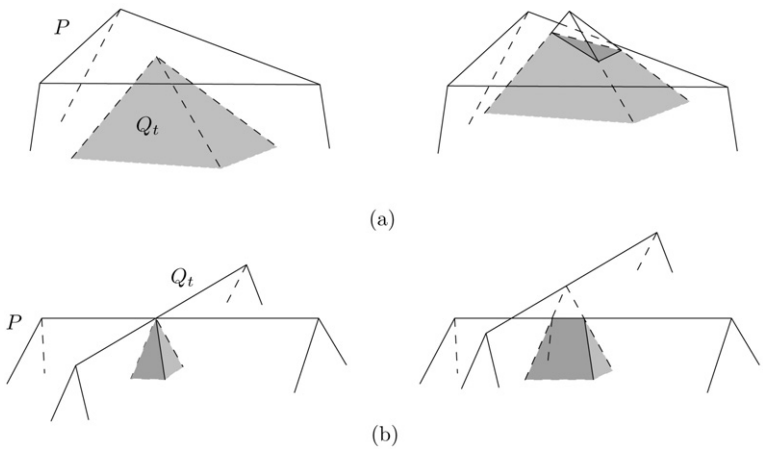

Fig. 1. Events of type 2 and 3: (a) a vertex of Q_t hits a facet of P in the left, and (b) an edge of P crosses an edge of Q_t in the left. The combinatorial structure of $P \cap Q_t$ change by upward translations of Q_t as in the right figures.

2.1. Orientation tests

To find the maximum of f, we first find the cell containing the maximum in the arrangement of event planes, then we construct the cubic function for that cell, and find its maximum. We want to find the cell without constructing the whole arrangement.

To find the cell, we assume that we have already realized a plane orientation test, which decides for a given plane, on which side of the plane the maximum of f lies. For each of the $O(n)$ groups, we apply binary search, using the plane orientation test, to find the pair of planes in that group, between which the maximum lies. This calls $O(n \log n)$ plane orientation tests, and gives us a set of $O(n)$ halfspaces. The cell containing the maximum is the intersection of these halfspaces. We have to find a maximum point in the intersection of the $O(n)$ halfspaces, but that is a problem of linear optimization in three-dimensional space, which can be solved in $O(n)$ time. Thus the time used to find the maximum is $O(n^2 \log n + (n \log n \times T_{\text{pot}}))$, where T_{pot} is the time needed for a plane orientation test.

Plane orientation tests

To realize the plane orientation test for a given query event plane H, we notice that we can know the orientation of the plane if we have the maximum of the function f restricted to the plane H, and the gradient of f at that maximum point. Thus one way to realize it is to proceed in the same way as for the three-dimensional maximum finding: we consider the arrangement of the event planes, restricted to H; then it becomes an arrangement of intersection lines of the event planes with H. The lines are again classified into $O(n)$ groups of $O(n)$ parallel lines each, and another $O(n)$ groups of $O(n)$ copunctual lines. The function is again piecewise cubic on the cells of this arrangement, and $(f(t))^{1/3}$ is still concave. So if we have a way for a line orientation test, which for a given query line in that plane tells us on which side the maximum lies, then we can apply the same method as before, in order to know the maximum restricted to H. We perform binary search on the groups of the lines, using the sorted order already constructed earlier, then we can select $O(n)$ halfplanes, and compute their intersection to find a point in the cell containing the maximum, and finally find the maximum. Thus we can reduce a single plane orientation test to $O(n \log n)$ line orientation tests; so if T_{lot} denotes the time for a line orientation test, then $T_{\text{pot}} = O(n \log n \times T_{\text{lot}})$.

Line orientation tests

The $O(n)$ groups, each of which previously consisted of $O(n)$ parallels lines or $O(n)$ copunctual lines, are now just points on the query line L. To realize the line orientation test, we should know a cell (i.e., an interval) on L which contains the maximum in each group. For this, we could again perform the same reduction, and indeed we will determine the maximum as way to find the orientation. This requires $O(n)$ binary searches for the different groups independently, which takes $O(n^2 \log n)$ time in total. This time bound is far from what we want.

Instead, we perform a randomized binary search on all these $O(n^2)$ points of the $O(n)$ groups in $O(n \log^2 n)$ expected time as follows. We maintain $O(n)$ sorted lists of points for the different groups in the search range along L.

Initially, the search range is $(-\infty, +\infty)$ and there are $O(n^2)$ points on the query line. In each round, we select a point p from the set of points in the range, uniformly distributed over them, and evaluate the function at p in $O(n)$ time. From that we gain the information whether or not the maximum is smaller than the function value at p. If the function realizes the maximum at p, we are done. If the maximum is smaller than the function value at p, we set the search range into $(-\infty, p)$. Otherwise, we set the search range into $(p, +\infty)$. We now look at each of the $O(n)$ sorted lists and update the list with the new end p such that it contains only the points in the new range. This can be done in $O(\log n)$ time for a group. A round takes $O(n \log n)$ time in total; $O(n)$ time for the function evaluation and $O(n \log n)$ time for the update of the still possible lists. Repeatedly, we select again a point uniformly distributed from the set of points in the range.

This randomized binary search consists of expected $O(\log n)$ rounds, and needs $O(n \log n)$ time per round. Thus we can answer the line orientation test in expected time of $T_{\text{lot}} = O(n \log^2 n)$.

A plane orientation test can be done in expected time $T_{\text{pot}} = O(n \log n \times T_{\text{lot}}) = O(n^2 \log^3 n)$, thus the total time to solve the maximum overlap problem in 3-dimensional space is $O(n^2 \log n + (n \log n \times T_{\text{pot}})) = O(n^3 \log^4 n)$.

A similar method has been used by Fukuda and Uno in [8] for the two-dimensional case, but the earlier result in [3] already gives a better bound.

2.2. Generalization to higher dimensions

We observe that we really used only two properties of the problem: (1) the function is concave, and (2) the event planes in translation space occur in groups which allow us to use the binary search. The first property, the concavity of the function $(f(t))^{1/d}$, holds also in d-dimensional space, but the second property breaks down from dimension four on, since we have groups of three-dimensional event hyperplanes caused by an one-dimensional face of P crossing a two-dimensional face of Q, which are not sorted linearly for binary searches. Generally, in d-dimensional space, we have groups of event hyperplanes caused by k-dimensional faces of P intersecting $(d-k-1)$-dimensional faces of Q, and these groups do not have any simple linear sorted structure for $k \neq 0, d-1$. So in the dimension reduction argument, we cannot just perform $O(n)$ binary searches. Instead we partition the event hyperplanes in groups, and build Chazelle's point location structure [7] for each group, then find the cell of the optimum point in each group by performing hyperplane orientation queries as required for the point location queries. Thus we only exchange the simple binary search structure for the more complex structure for point location in arrangements of hyperplanes. We partition the $O(n^2)$ event hyperplanes into $n^{1+\frac{3}{(d-1)d}}$ groups of $O(n^{1-\frac{3}{(d-1)d}})$ hyperplanes each, and build the point location structure for each group in

$$O\big((n^{1-\frac{3}{(d-1)d}})^d\big) = O(n^{d-\frac{3}{d-1}})$$

time [7]. For all $O(n)$ groups together, this gives a preprocessing time of $O(n^{d+1-\frac{3}{d}})$. Then we apply in each group the $O(\log n)$ hyperplane orientation tests required by the point location algorithm for that group; each hyperplane orientation test is an instance of the same maximization problem in $(d-1)$-dimensional space, so it takes $O(n^{d-\frac{3}{d-1}}(\log n)^d)$ time by induction on d. Since we need $O(n \log n)$ hyperplane orientation tests for all groups, the total time becomes $O(n^{d+1-\frac{3}{d}}(\log n)^{d+1})$, which completes the proof of Theorem 2.

3. Minimizing the convex hull under translation

The same properties we need also hold for our second problem, to find the translation $t \in \mathbb{R}^d$ that minimizes the volume of the convex hull of P and $Q+t$. Let $g(t) = \text{Vol}(\text{conv}(P \cup (Q+t)))$. For simplicity, we consider only three dimension; as we did in the problem of maximizing the overlap under translation in the last section, we can extend to four or higher dimensions

Lemma 5. *Let P and Q be convex polyhedra in $\mathbb{R}^d$. Then the function $t \mapsto g(t)$ is convex.*

Proof. We describe only when $d = 3$. To show that the function $g(t)$ is convex, it suffices to consider translations along a fixed line ℓ and show that $g(t)$ is convex for $t \in \ell$. Without loss of generality, we translate Q by t along the

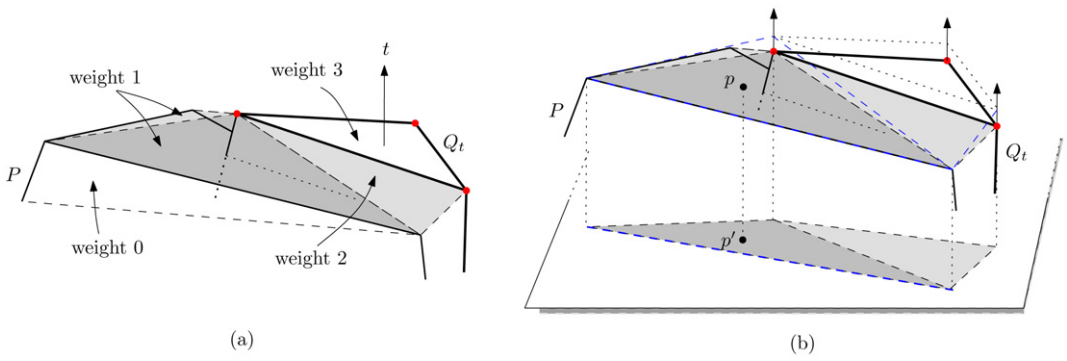

Fig. 2. (a) Triangles on the convex hull of $P \cup Q_t$ and their weights. (b) Projection of triangles on the convex hull onto a plane orthogonal to x-axis, i.e., to the direction of t.

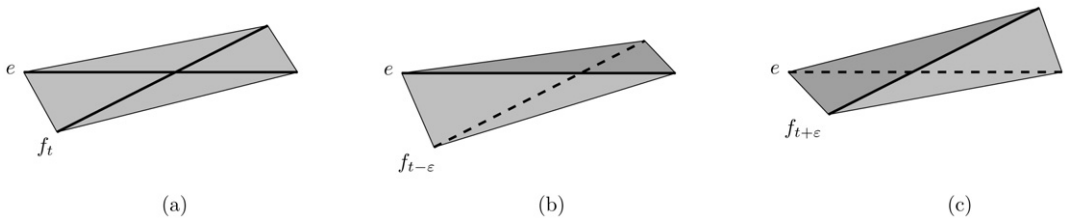

Fig. 3. (a) An edge e of P crosses an edge f_t of Q_t. Both polyhedra lie below the plane spanned by e and f_t. If we translate Q_t slightly by $\varepsilon > 0$ downward (b) or upward (c), the convex hull changes.

x-axis, that is, for $t \in \mathbb{R}$, let $Q_t := Q + (t, 0, 0)$. We will show that $\mathrm{Vol}(\mathrm{conv}(P \cup Q_t))$ is convex by showing that its derivative (the volume change) is non-decreasing.

We consider the projection of $\mathrm{conv}(P \cup Q_t)$ on a plane orthogonal to the x-axis. As Q_t moves along the x-axis, the convex hull $\mathrm{conv}(P \cup Q_t)$ has a front side, along which volume is gained by points entering the convex hull, and a back side, along which volume is lost by points leaving the convex hull.

Both sides of the convex hull consist of facets, which we may assume to be triangles. The rate with which points pass through such a triangle, that is, the rate of volume gain or loss through that triangle, is proportional to the area of the projection of that triangle onto the plane orthogonal to the x-axis, and the number of moving vertices, that is, the number of vertices from Q_t which this triangle has; one of $\{0, 1, 2, 3\}$. We call the number the *weight* of the (projected) triangle. See Fig. 2(a).

This area does not change by the motion as long as the triangle exists on the boundary of the convex hull, but the triangle might change in the following two cases. It can be subdivided when a new point appears as a vertex of the boundary, or it can be joined together with other triangles when one of its vertices disappears in the interior of the convex hull. The volume change of the convex hull, $g'(t)$, is thus the integral over the projection of the convex hull front side of the points, weighted by the weight of their triangle, minus the same integral over the projection of the back side (see Fig. 2(b)).

Consider a fixed point p on the convex hull and its projected point p' on the plane orthogonal to x-axis. If p is in the front side, the projected triangle containing p' has an increasing weight from 0 up to 3. If p is on the back side, the projected triangle containing p' has a decreasing weight from 3 to 0. So $g'(t)$ is along that line an increasing step function, hence $g(t)$ is convex. □

Finally we need to describe the event planes. But these are indeed exactly the same event planes as in the previous problem: a vertex of Q passes the supporting plane through a facet of P, a vertex of P passes the supporting plane through a facet of Q, or an edge of P crosses an edge of Q. Of the vertex-facet events, there are really only $O(n)$ possible ones, since a facet can cause an event only with the extreme vertex in the normal direction of the facet where the combinatorial structure of the convex hull changes.

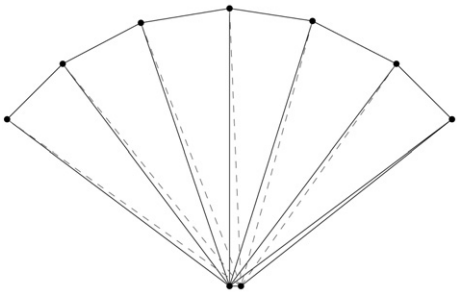

Fig. 4. The "fan-shaped" object and its rotated copy by 90° around a vertical line causes $\Omega(n^2)$ edge-edge events.

But $\Theta(n^2)$ edge–edge events are possible: the combinatorial structure of the convex hull changes when an edge e of P crosses an edge f_t of Q_t and both polyhedra lie in the same halfspace of the plane spanned by e and f_t. Fig. 3(a) shows two crossing edges e and f_t such that both polyhedra lie below the plane spanned by e and f_t. If we translate Q_t slightly by $\varepsilon > 0$ downward (b) or upward (c), the convex hull changes as in the figures.

Consider now the "fan-shaped" object in Fig. 4, which is the convex hull of nine points: one at $(1, 0, 0)$, another at $(-1, 0, 0)$, and the other seven points along a circular arc in the yz-plane as in the figure. Imagine that we are given the object and its rotated copy by 90° around a vertical line. Then each edge along the top chain causes an edge–edge event with every edges in the top chain of the rotated copy. This completes the proof of our last theorem.

References

[1] H.-K. Ahn, O. Cheong, Stacking and bundling two convex polytopes, in: Proc. 16th International Symposium on Algorithms and Computation (ISAAC 2005), in: LNCS, vol. 3827, Springer, 2005, pp. 882–891.
[2] H.-K. Ahn, O. Cheong, C.-D. Park, C.-S. Shin, A. Vigneron, Maximizing the overlap of two planar convex sets under rigid motions, in: Proc. 21st Annual ACM Symposium on Computational Geometry (SoCG 2005), 2005, pp. 356–363. Also in Computational Geometry: Theory and Applications 37 (2007) 3–15.
[3] M. de Berg, O. Cheong, O. Devillers, M. van Kreveld, M. Teillaud, Computing the maximum overlap of two convex polygons under translations, Theory of Computing Systems 31 (1998) 613–628.
[4] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge Univ. Press, 2004.
[5] P. Brass, M. Sharifi, A lower bound for Lebesgue's universal cover problem, International Journal of Computational Geometry & Applications 15 (2005) 537–544.
[6] B. Chazelle, An optimal algorithm for intersecting three-dimensional convex polyhedra, SIAM Journal on Computing 21 (1992) 671–696.
[7] B. Chazelle, Cutting hyperplanes for divide-and-conquer, Discrete & Computational Geometry 9 (1993) 145–159.
[8] K. Fukuda, T. Uno, Algorithms for maximizing the volume of intersection of polytopes, in: Proc. 22nd European Workshop on Computational Geometry (EWCG 2006), 2006, pp. 197–200.
[9] K. Fukuda, T. Uno, Polynomial time algorithms for maximizing the intersection volume of polytopes, Pacific Journal of Optimization 3 (2007) 37–52.
[10] D.M. Mount, R. Silverman, A.Y. Wu, On the area of overlap of translated polygons, Computer Vision and Image Understanding: CVIU 64 (1996) 53–61.
[11] J.R. Sangwine-Yager, Mixed volumes, in: P.M. Gruber, J.M. Wills (Eds.), Handbook on Convex Geometry A, Elsevier, 1993, pp. 43–71.
[12] R. Schneider, Convex Bodies: The Brunn–Minkowski Theory, Encyclopedia of Mathematics and its Applications, vol. 44, Cambridge Univ. Press, 1993.