Computational Geometry 47 (2014) 15–24

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Overlap of convex polytopes under rigid motion ☆

Hee-Kap Ahn [a], Siu-Wing Cheng [b,*], Hyuk Jun Kweon [a], Juyoung Yon [b]

[a] Department of Computer Science and Engineering, POSTECH, Republic of Korea
[b] Department of Computer Science and Engineering, HKUST, Hong Kong

ARTICLE INFO

Article history:
Received 3 December 2012
Accepted 1 August 2013
Available online 7 August 2013
Communicated by M. de Berg

Keywords:
Convex polyhedron
Shape matching
Rigid motion
Overlap
Approximation algorithm

ABSTRACT

We present an algorithm to compute a rigid motion that approximately maximizes the volume of the intersection of two convex polytopes P_1 and P_2 in $\mathbb{R}^3$. For all $\varepsilon \in (0, 1/2]$ and for all $n \geqslant 1/\varepsilon$, our algorithm runs in $O(\varepsilon^{-3} n \log^{3.5} n)$ time with probability $1 - n^{-O(1)}$. The volume of the intersection guaranteed by the output rigid motion is a $(1-\varepsilon)$-approximation of the optimum, provided that the optimum is at least $\lambda \cdot \max\{|P_1|, |P_2|\}$ for some given constant $\lambda \in (0, 1]$.

1. Introduction

Shape matching is a common task in many object recognition problems. A translation or rigid motion of one shape is sought that maximizes some similarity measure with another shape. Convex shape matching algorithms have been used in tracking regions in an image sequence [10] and measuring symmetry of a convex body [8]. We define the *overlap* of two convex shapes to be the volume of their intersection, which is a robust similarity measure [12]. In this paper, we consider the problem of finding the maximum overlap of two convex polytopes in $\mathbb{R}^3$ under rigid motion.

Efficient algorithms have been developed for two convex n-gons in the plane. De Berg et al. [5] developed an algorithm to find the maximum overlap of two convex polygons under translation in $O(n \log n)$ time. Ahn et al. [4] presented two algorithms to find a $(1-\varepsilon)$-approximate maximum overlap, one for the translation case and another for the rigid motion case. They assume that the polygon vertices are stored in arrays in clockwise order around the polygon boundaries. Ahn et al.'s algorithms run in $O(\varepsilon^{-1} \log n + \varepsilon^{-1} \log(1/\varepsilon))$ time for the translation case and $O(\varepsilon^{-1} \log n + \varepsilon^{-2} \log(1/\varepsilon))$ time for the rigid motion case. Cheong et al. [7] gave an algorithm to align two simple polygons P_1 and P_2 by a rigid motion so that their overlap is at least the optimum minus $\varepsilon \cdot \min\{|P_1|, |P_2|\}$. The running time is $O((n^3/\varepsilon^8) \log^5 n)$. Cheng and Lam [6] recently improved the running time to $O((n^3/\varepsilon^4) \log^{5/3} n \log^{5/3} \frac{n}{\varepsilon})$. Finding the exact maximum overlap under rigid motion seems difficult. A brute force approach is to subdivide the space of rigid motion $(-\pi, \pi] \times \mathbb{R}^2$ into cells so that the intersecting pairs of polygon edges do not change within a cell. The hope is to obtain a formula for the maximum overlap within a cell as the intersection does not change combinatorially, and then compute the maximum of the formula. Unfortunately, the subdivision of $(-\pi, \pi] \times \mathbb{R}^2$ has curved edges and facets. Also the formula is a sum of a large number

☆ Research of Cheng and Yon was partly supported by the Research Grant Council, Hong Kong, China, project no. 611711. Research of Yon was also supported by the Hong Kong PhD Fellowship. Research by Ahn was partly supported by the NRF grant 2011-0030044 (SRC-GAIA) funded by the government of Korea.

* Corresponding author.
E-mail addresses: heekap@postech.ac.kr (H.-K. Ahn), scheng@cse.ust.hk (S.-W. Cheng), kweon7182@postech.ac.kr (H.J. Kweon), jyon@cse.ust.hk (J. Yon).

0925-7721/$ – see front matter
http://dx.doi.org/10.1016/j.comgeo.2013.08.001

of fractions, and optimizing the formula seems to require solving a high-degree polynomial system. These issues make it a challenge to optimize the formula in a cell.

Fewer algorithmic results are known concerning the maximum overlap of two convex polytopes in $\mathbb{R}^d$ for $d \geqslant 3$. Let n be the number of hyperplanes defining the convex polytopes. Ahn et al. [2] developed an algorithm to find the maximum overlap of two convex polytopes under translation in $O(n^{(d+1-3/d)\lfloor d/2\rfloor}\log^{d+1} n)$ expected time. Recently, Ahn, Cheng and Reinbacher [3] have obtained substantially faster algorithms to align two convex polytopes under translation in $\mathbb{R}^d$ for $d \geqslant 3$. The overlap computed is no less than the optimum minus μ, where μ is any small constant fixed in advance. The running times are $O(n\log^{3.5} n)$ for $\mathbb{R}^3$ and $O(n^{\lfloor d/2\rfloor+1}\log^d n)$ for $d \geqslant 4$, and these time bounds hold with probability $1-n^{-O(1)}$. There is no specific prior result concerning the maximum overlap of convex polytopes under rigid motion. Vigneron [13] studied the optimization of algebraic functions and one of the applications is the alignment of two possibly non-convex polytopes under rigid motion. For any $\varepsilon \in (0,1)$ and any two convex polytopes with n defining hyperplanes, Vigneron's method can return in $O(\varepsilon^{-\Theta(d^2)} n^{\Theta(d^3)} (\log \frac{n}{\varepsilon})^{\Theta(d^2)})$ time an overlap under rigid motion that is at least $1-\varepsilon$ times the optimum. Finding the exact overlap is even more challenging in $\mathbb{R}^3$.

In this paper, we present a new algorithm to approximate the maximum overlap of two convex polytopes P_1 and P_2 in $\mathbb{R}^3$ under rigid motion. For the purpose of shape matching, it often suffices to know that two input shapes are very dissimilar if this is the case. Therefore, we are only interested in matching P_1 and P_2 when their maximum overlap under rigid motion is at least $\lambda \cdot \max\{|P_1|, |P_2|\}$ for some given constant $\lambda \in (0,1]$, where $|P_i|$ denotes the volume of P_i. Under this assumption, for all $\varepsilon \in (0,1/2]$ and for all $n \geqslant 1/\varepsilon$, our algorithm runs in $O(\varepsilon^{-3} n \log^{3.5} n)$ time with probability $1-n^{-O(1)}$ and returns a rigid motion that achieves a $(1-\varepsilon)$-approximate maximum overlap. The assumption can be verified as follows. Run our algorithm using $\lambda/2$ instead of λ. Check if the overlap output by our algorithm is at least $(1-\varepsilon)\lambda \cdot \max\{|P_1|, |P_2|\}$. If not, we know that the assumption is not satisfied. If yes, the maximum overlap is at least $(\lambda/2)\cdot\max\{|P_1|, |P_2|\}$ and our algorithm's output is a $(1-\varepsilon)$-approximation because we used $\lambda/2$ in running the algorithm. Our high-level strategy has two steps. First, sample a set of rotations. Second, for each sampled rotation, apply it and then apply the almost optimal translation computed by Ahn et al.'s algorithm [3]. Finally, return the best answer among all rigid motions tried.

If one uses a very fine uniform discretization of the rotation space, it is conceptually not difficult to sample rotations so that the resulting approximation is good. The problem is that such a discretization inevitably leads to a running time that depends on some geometric parameters of P_1 and P_2. In order to obtain a running time that depends on n and ε only, we cannot use a uniform discretization of the entire rotation space. Indeed, our contribution lies in establishing some structural properties that allow us to discretize a small subset of the rotation space, and exploiting this discretization in the analysis to prove the desired approximation. This approach is also taken in the 2D case in [4], but our analysis is not an extension of that in [4] as the three-dimensional situation is very different.

2. Similar polytopes

In this section, we show that P_1 and P_2 are "similar" under the assumption that their maximum overlap is at least $\lambda \cdot \max\{|P_1|, |P_2|\}$. We use the Löwner–John ellipsoid [11] to identify three axes of P_1 and P_2. For every convex body P in $\mathbb{R}^d$, it is proven by Löwner that there is a unique smallest ellipsoid E that contains P. Then John proved that $\frac{1}{d}E$ is contained in P. There are various algorithms for finding an ellipsoid of this flavor.

Lemma 1. *(See [11].) Let P be a convex body with m vertices in $\mathbb{R}^3$. For every $\eta > 0$, an ellipsoid $\mathcal{E}(P)$ can be computed in $O(m/\eta)$ time such that $\frac{1}{3(1+\eta)}\mathcal{E}(P) \subset P \subset \mathcal{E}(P)$.*

For $i \in \{1,2\}$, we use $\mathcal{E}(P_i)$ to denote the ellipsoid guaranteed by Lemma 1 for P_i, using the setting of $\eta = 1/3$. There are three mutually orthogonal directed lines α_i, β_i and γ_i through the center of $\mathcal{E}(P_i)$ such that $|\alpha_i \cap \mathcal{E}(P_i)|$ and $|\gamma_i \cap \mathcal{E}(P_i)|$ are the shortest and longest, respectively, among all possible directed lines through the center of $\mathcal{E}(P_i)$. After fixing α_i and γ_i, there are two choices for β_i and any one will do. We call these directed lines the α_i-, β_i-, and γ_i-axes of P_i. The lengths $a_i = |\alpha_i \cap \mathcal{E}(P_i)|$, $b_i = |\beta_i \cap \mathcal{E}(P_i)|$, and $c_i = |\gamma_i \cap \mathcal{E}(P_i)|$ are the three *principal diameters* of $\mathcal{E}(P_i)$. Notice that $a_i \leqslant b_i \leqslant c_i$. Define $a_{\min} = \min\{a_1, a_2\}$, $b_{\min} = \min\{b_1, b_2\}$, and $c_{\min} = \min\{c_1, c_2\}$. The following result gives an upper bound on the maximum overlap of P_1 and P_2.

Lemma 2. *For $i \in \{1,2\}$, let R_i be a box with side lengths a_i, b_i, and c_i. The maximum overlap of R_1 and R_2 under rigid motion is at most $\sqrt{2}a_{\min}b_{\min}c_{\min}$.*

Proof. Without loss of generality, we suppose that a_1 is $a_{\min}$, that is, $a_1 \leqslant a_2$. If $b_{\min} = b_1$ and $c_{\min} = c_1$, then the maximum overlap of R_1 and R_2 under rigid motion is $|R_1| = a_{\min}b_{\min}c_{\min}$. There are three cases left: (1) $b_{\min} = b_2$ and $c_{\min} = c_2$, (2) $b_{\min} = b_1$ and $c_{\min} = c_2$, and (3) $b_{\min} = b_2$ and $c_{\min} = c_1$. For $i \in \{1,2\}$, let the ab-plane of R_i be the plane through the center of R_i and parallel to the facets of side lengths a_i and b_i. The bc-plane and ac-plane of R_i are defined analogously. Let L_i^{ab} be the line through the center of R_i and perpendicular to the ab-plane of R_i. The lines L_i^{bc} and L_i^{ac} are defined analogously. In the rest of the proof, assume that R_1 and R_2 have been placed such that their overlap is maximum.