Minkowski Sums of Rotating Convex Polyhedra

Jyh-Ming Lien
Dept. of Computer Science, George Mason University
Fairfax, VA, USA.
jmlien@cs.gmu.edu

Categories and Subject Descriptors: I.3.5 [COMPUTER GRAPHICS]: Computational Geometry and Object Modeling

General Terms: Algorithms, Experimentation.

Keywords: Minkowski sum, polytope, kinematic algorithm.

1. INTRODUCTION

The Minkowski sum of two polyhedra P and Q is:

$$P \oplus Q = \{p + q \mid p \in P, q \in Q\}. \quad (1)$$

Minkowski sum is an important operation due to it's fundamental role in many geometric applications. Many methods have been proposed during the last three decades to compute Minkowski sums; see the surveys in [1,3,6]. In 3-dimensions, several methods [1,2,4,5] are known to compute the Minkowski sum of *convex* polyhedra efficiently.

In this work, we developed a method that efficiently computes the Minkowski sum of *rotating convex polyhedra*. It is well known that the Minkowski sum of two polyhedra can be dramatically different from the Minkowski sum of the same polyhedra after rotation. To the best of our knowledge, all existing methods recompute a new Minkowski sum every time when P or Q rotates. A main problem of recomputing the Minkowski sum from scratch is that it requires the same computation even when a small amount of rotation is applied to P or Q. Computing the Minkowski sum of polyhedra undergoing small rotations can be found in many problems, such as *general penetration depth* estimation for physically-based simulation and *configuration-space obstacle* approximation for robotic motion planning.

Our work is motivated by the observation described above. Thus, our objective is to compute the Minkowski sums of rotating convex polyhedra without recomputing the entire Minkowski sum repetitively. The main idea in our method is to generate the Minkowski sum from the previous Minkowski sum. More specifically, we generate the new Minkowski sum by correcting the 'errors' introduced by rotation. More details of our method are described in the following section.

The preliminary results shown in Table 1 and in the video are very encouraging. From our experiments, we observe that the number of 'errors' is small and the effort to correct these errors is far cheaper than computing the Minkowski sum from scratch. This allows us to generate a sequence of Minkowski sums (10 to 500 times) much more efficiently than re-computing the Minkowski sums. On average, the Minkowski sum can be updated at the rate of more than 1000 frames per second in our experiments.

SCG'08, June 9–11, 2008, College Park, Maryland, USA.
ACM 978-1-60558-071-5/08/04.

2. OUR APPROACH

Without loss of generality, we assume that P is movable while Q is stationary. We let P_s and P_t be two copies of P at two configurations s and t with distinct orientations. Our goal is to compute $M_t = P_t \oplus Q$ from $M_s = P_s \oplus Q$. Moreover, we will design our algorithm so that its running time is sensitive to the orientation difference between P_s and P_t, i.e., the smaller the difference between P_s and P_t, the faster the computation of M_t. Figure 1 shows an example of the Minkowski sums before and after rotating the cube.

Recall that the facet of a Minkowski sum can only come from two sources: the facets, called fv-facets, generated from a facet of P and a vertex of Q or vice versa and the facets, called ee-facets, generated from a pair of edges from P and Q, respectively. A facet f and a vertex v produce a valid fv-facet if and only if the normal of f is a convex combination of the normals of the facets incident to v. Similarly, a pair of edges e_1 and e_2 form an ee-facet if the cross product of vectors parallel to e_1 and e_2 is a convex combination of the normals of the facets incident to e_1 and e_2. These criteria allow us to test if a given pair of features (a facet/vertex pair or an edge pair) will produce a valid Minkowski sum facet by checking only the neighborhood of these features. It is obvious that when P_s becomes P_t, some facets (i.e., pairs of features) in M_s will no long satisfy these criteria. We called these facets the 'errors' introduced by rotation.

There are two main steps in our algorithm: (1) identify the errors and (2) correct the errors. We first check exhaustively all fv-facets in M_s to find some errors, which we call fv-errors. From fv-errors, we identify all other potential errors, called ee errors. For each fv-error, we perform a local search (similar to flooding) to compute a new fv-facet. Similarly, for an ee-error, a local search is conducted to compute a set of new ee-facets (since an edge can be associated with multiple edges to form multiple ee-facets). It is important to note that such searches are very efficient when the difference between P_s and P_t is small. Finally, the Minkowski sum M_t is composed of two types of facets: the facets from M_s that still satisfy the aforementioned criteria after rotation and the facets that are created due to the fv- and ee-errors.

3. RESULTS

In our experiments, the polyhedron P rotates using a sequence of random quaternions. Each quaternion is applied

Table 1: Experimental results. In this table, DD denotes dioctagonal dipyramid and GS4 denotes geodesic sphere level 4. Some models in this table are obtained from E. Fogel and D. Halperin's webpage [1]. All computation times are shown in milliseconds and are averaged over at least 100 runs.

$P \oplus Q$	cube⊕ball	ellipse⊕DD	rod⊕cone	cube⊕cylinder	GS4⊕GS4	ball⊕ball
# of facets in P	12	960	324	12	500	19,800
# of facets in Q	19,800	32	78	140	500	19,800
re-computation time	25.61	3.10	2.59	0.20	24.81	33987.40
update time	1.42	0.13	0.15	0.02	0.22	63.16

to P for a random period of time. In Table 1, we compare the proposed (update-based) method to a naïve method that re-computes the Minkowski sum in every time step. The naïve method checks the validity of all facet-vertex and edge-edge pairs every time P rotates. All the experiments are performed on a PC with Intel CPUs at 2.13 GHz with 4 GB RAM. Our implementation is coded in C++.

From Table 1, it is clear that the update-based method is always faster than the naïve method. Even for very simple cases, such as cube⊕cylinder, the update-based method is at least 10 times faster. For more complex examples, such as ball⊕ball, our update-based method is about 540 times faster than the naïve method.

4. THE VIDEO

The video starts with the definition of Minkowski sum and the issues of computing the Minkowski sums of rotating polyhedra from scratch. Next shows the Minkowski sums of two randomly rotating spheres. New facets created in the current frame are shown in color. The video goes on with another example illustrating a more clear view of the advantage of the update-based Minkowski sum computation. The video ends by showing three more examples of rotating convex polyhedra.

The video is composed using Adobe After Effects and the voice over is synthesized using Text-To-Speech from AT&T.

5. REFERENCES

[1] E. Fogel and D. Halperin. Exact and efficient construction of Minkowski sums of convex polyhedra with applications. In *Proc. 8th Wrkshp. Alg. Eng. Exper. (Alenex'06)*, pages 3–15, 2006.

[2] K. Fukuda. From the zonotope construction to the minkowski addition of convex polytopes. *Journal of Symbolic Computation*, 38(4):1261–1272, 2004.

[3] P. K. Ghosh. A unified computational framework for Minkowski operations. *Computers and Graphics*, 17(4):357–378, 1993.

[4] P. Gritzmann and B. Sturmfels. Minkowski addition of polytopes: computational complexity and applications to Gröbner bases. *SIAM J. Discret. Math.*, 6(2):246–269, 1993.

[5] A. Kaul and J. Rossignac. Solid-interpolating deformations: construction and animation of PIPs. In *Proc. Eurographics '91*, pages 493–505, 1991.

[6] G. Varadhan and D. Manocha. Accurate Minkowski sum approximation of polyhedral models. *Graph. Models*, 68(4):343–355, 2006.

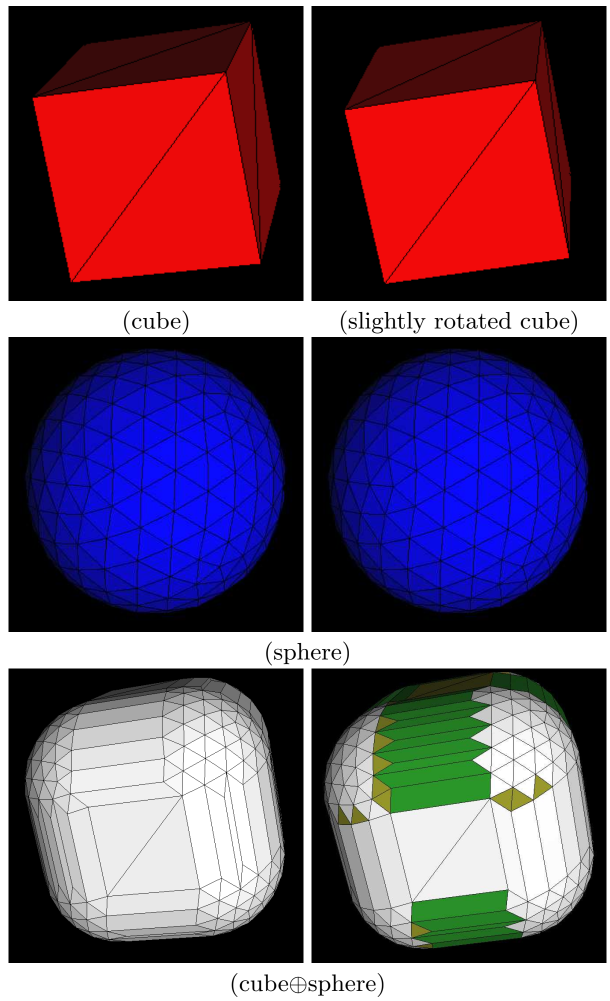

Figure 1: Left column: A cube, a sphere, and their Minkowski sum. Right column: The same cube but slightly rotated, the same sphere, and their Minkowski sum. The colored facets (appeared darker in grayscale) of the Minkowski sum are the new facets that do not exist in the Minkowski sum on the left. The yellow and green facets indicate the new fv- and ee-facets, respectively.