Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs

James T. Klosowski

1998 IEEE Transactions on Visualization and Computer Graphics

Presenter: Yi-Ning Chang

March 29, 2017

Outline

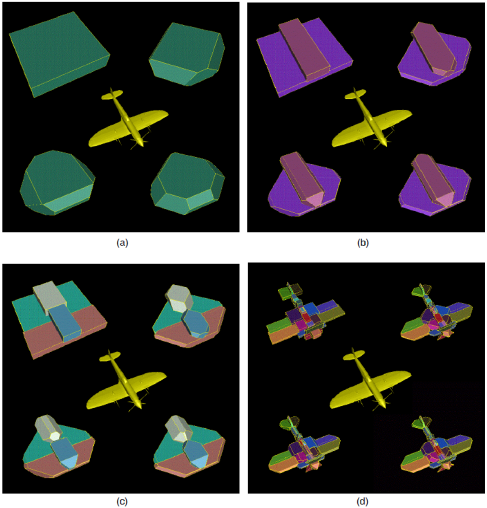
Introduction

- Collision detection is of paramount importance for many applications in computer graphics and visualization.
- In this paper, author develop and analyze a method, based on bounding-volumn hierarchies, for efficient collision detection for objects moving within highly complex environments.
- Previous work
  - Octrees, k-d trees, BSP trees, and so on.
  - *RAPID* (Robust and Accurate Polygon Interference Detection) system based on *OBBTrees*.

BV-Tree

- $BVT(S)$: a BV-tree specifies a bounding volumn hierarchy on S.
- Set S: a set of n geometry triangles in 3D that specify the boundary of polygonal models.
- Each node v of $BVT(S)$ corresponds to a subset, $S_v \subseteq S$.
- $b(S_v)$: a *bounding volume*, associated with node v of $BST(S)$, is an approximation to the set S_v using a smallest instance of specified class of shapes.

BV-tree



(a) (b) (c) (d)

Design Criteria

$$T = N_v \times C_v + N_p \times C_p + N_u \times C_u$$

- T: total cost function for collision detection.
- N_v: number of pairs of bounding volumes tested for overlap.
- C_v: cost of testing a pair of bounding volumes for overlap.
- N_p: number of pairs of primitives tested for contact.
- C_p: cost of testing a pair of primitives for contact.
- N_u: number of nodes of the flying hierarchy must be updated.
- C_u: cost of updating such node.

k-DOP

- k-dop: *k-discrete orientation polytope* is a convex polytope whose facets are determined by k *fixed* orientations.
- *6-dop*(AABB), *14-dop*, *18-dop*, *26-dop*

	Sphere	AABB	OBB
N_v	big	median	small
C_v	$O(1)$	$O(k)$	$O(\log^2 n)$
C_u	$O(1)$	$O(n)$	$O(1)$

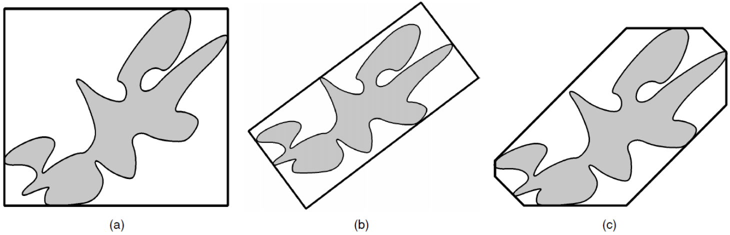


Fig. 1. Approximations of an object by three bounding volumes: an axis-aligned bounding box (AABB), an oriented bounding box (OBB), and a k-dop (where $k = 8$).

Splitting Rule

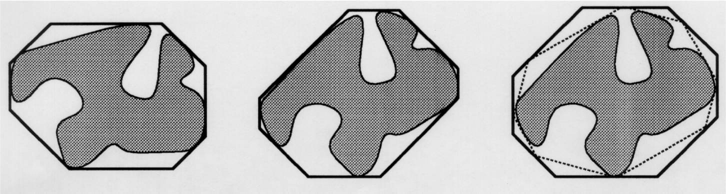
- Choice of Axis
  - *Min Sum*: $O(k|S_v|) + O(k \log k)$
  - *Min Max*: $O(k|S_v|) + O(k \log k)$
  - *Splatter*: $O(|S_v|)$
  - *Longest Side*: $O(1)$
- Choice of Split Point
  - *median*: more balanced.
  - *mean*: less total volume while not harming the balance of the tree severely.

Collision Detection Using BV-Trees

- The method of *updating* the k-dops.
- The *algorithm* for comparing two BV-trees to determine if there is a collision.

Updating the BV-Trees

- *Hill climbing algorithm*
  - Store the boundary representation of convex hull of S_v.
  - $O(n)$
- *Approximation method*
  - store the boundary vertices of the k-dop $b(S_v)$.
  - $O(k \log k)$



(a) (b) (c)

Fig. 2. Illustration of the approximation method of handling a rotating object.

Tree Traversal Algorithm

Algorithm *TraverseTrees*(v_F, v_E)
Input: A node v_F of the flying hierarchy, a node v_E of the environment hierarchy

1. **if** $b(v_F) \cap b(v_E) \neq \emptyset$ **then**
2. **if** v_E is a leaf **then**
3. **if** v_F is a leaf **then**
4. **for** each triangle t_E of S_{v_E}
5. **for** each triangle t_F of S_{v_F}
6. check test triangles t_E and t_F for intersection
7. **else**
8. **for** each child v_f of v_F
9. TraverseTrees(v_f, v_E)
10. **else**
11. **for** each child v_e of v_E
12. TraverseTrees(v_F, v_e)
13. **return**

Algorithm 1: Pseudocode of the tree traversal algorithm.

Experiment for different k

TABLE 1
AVERAGE COSTS OF C_v AND C_u (IN MS), FOR DIFFERENT CHOICES OF k

	6-dop	14-dop	18-dop	26-dop
C_v	0.0008	0.0016	0.0020	0.0028
C_u	0.0045	0.0174	0.0235	0.0509

TABLE 2
AVERAGE CD TIME (IN MS), USING OUR "SPLATTER" SPLITTING RULE

	Pipes	Torus	747	Swept	Interior
Env. Size (no. tri.)	143,690	98,114	100,000	40,000	169,944
Object Size (no. tri.)	143,690	20,000	14,646	36	404
No. of Steps	2,000	2,000	10,000	1,000	2,528
No. of Contacts	2,657	1,472	7,906	0	84,931
Hier. Method (ms per check)					
6-dop	0.487	0.294	1.639	0.582	4.375
14-dop	0.392	0.191	0.760	0.153	2.701
18-dop	0.366	0.184	0.356	0.109	2.754
26-dop	0.525	0.210	0.415	0.076	2.639
RAPID	0.934	0.242	0.494	0.556	4.375

Experiment for preprocessing time

TABLE 3
PREPROCESSING TIME (IN MINUTES),
USING OUR 18-DOP METHOD

Construction Method	Pipes	Torus	747	Swept	Interior
Longest Side	3.61	1.61	1.69	0.31	5.78
Min Sum	26.75	19.12	20.98	7.17	31.03
Min Max	28.03	19.16	20.87	7.19	31.13
Splatter	3.63	1.62	1.71	0.32	5.71
RAPID	1.05	0.69	0.71	0.26	1.31

Experiment for dividing method

TABLE 4
AVERAGE CD TIME (IN MS), USING OUR 18-DOP METHOD, DIVIDING AT THE MEAN

Construction Method	Pipes	Torus	747	Swept	Interior
Longest Side	0.384	0.192	0.366	0.111	3.036
Min Sum	0.356	0.185	0.330	0.108	2.667
Min Max	0.391	0.191	0.439	0.111	2.783
Splatter	0.366	0.184	0.356	0.109	2.754

TABLE 5
AVERAGE CD TIME (IN MS), USING OUR 18-DOP METHOD, DIVIDING AT THE MEDIAN

Construction Method	Pipes	Torus	747	Swept	Interior
Longest Side	0.476	0.193	0.412	0.116	3.164
Min Sum	0.450	0.192	0.359	0.111	2.822
Min Max	0.530	0.196	0.450	0.114	3.080
Splatter	0.481	0.194	0.396	0.113	2.774

1. Introduction
2. Preprocessing
3. Collision Detection
4. Experiment
5. Conclusion

Conclusion

- Proposed a method for efficient collision detection among polygonal models, based on a bounding volumes hierarchy (BV-tree) whose bounding volumes are k-dops.
- Future work
    - Multiple flying objects.
    - Server and client system.