

# 凸包围多面体生成算法及应用

(申请清华大学工学硕士学位论文答辩报告)

研 究 生：唐 磊

指导教师：雍 俊 海 教授



计算机辅助设计图形学与可视化研究所

二〇一五年六月

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ

# 引言

## 凸包围体技术

在计算机图形学领域里的各种算法中发挥着重要作用，如优化渲染和建模过程，加速求交、碰撞检测等算法。

## 碰撞检测问题

计算机图形学、虚拟现实等领域中的研究热点，是计算机模拟真实环境中不可或缺的技术，在物理仿真及游戏领域里应用十分广泛。

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ

# 凸包围体的种类

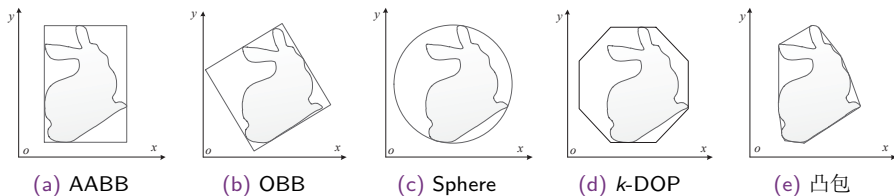


Figure 1: 不同种类的包围体

其他: Tribox、Swept-sphere、Sphere-shell、Zonotopes、圆柱形、圆锥、椭球形等等。

# 本文目标

## 综合比较

$k$ -DOP[1]: 方向固定且有限, 不同模型其截面方向一致, 不够紧致。

凸包: 很 (最) 紧致, 但面片数量太多, 复杂度  $O(n \log n)$ 。

## 本文凸包围体的目标

紧致: 能够自适应模型

快速: 利用 GPU 加速

灵活: 通过参数  $k$  调节简单性和紧致性

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ

# 碰撞检测算法

## 碰撞检测算法

许多应用的基础，例如在 3D 游戏，物理仿真，机器人，虚拟现实等领域中。

## 分类

**加速结构：** SPT（如四叉树、KD 树等） v.s BVH（OBB 树、 $k$ -DOP 树等）

**表现形式：** 刚体 v.s 可变形，凸体 v.s 凹体，CSG v.s 参数曲面 v.s 多边形网格

**碰撞环境：** 成对 v.s 多体，静止 v.s 运动，离散 v.s 连续



# 基于 BVH 的碰撞检测算法

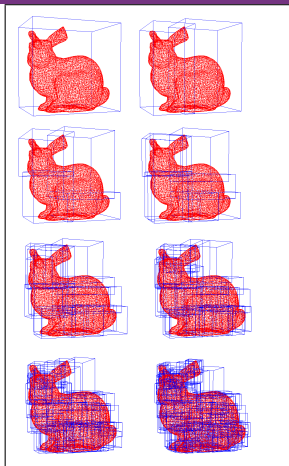


Figure 2: 八层 BVH 示例

## 算法 1 自顶向下层次遍历 BVH

输入: 两个 BVH 树的根节点  $node_1$ ,  $node_2$

输出: 模型是否相交

```

1: function TRAVERSEBVHTREE( $node_1$ ,  $node_2$ )
2:   if  $node_1.bv \cap node_2.bv = \emptyset$  then
3:     return False // 包围体重合测试, 包围体不相交直接返回
4:   else
5:     if  $node_1.children = \emptyset$  then
6:       if  $node_2.children = \emptyset$  then
7:         // 最底层叶子节点原生几何相交测试
8:         return CHECKINTERSECTION( $node_1.primitives$ ,  $node_2.primitives$ )
9:       else
10:        for all  $child \in node_2.children$  do
11:          TRAVERSEBVHTREE( $node_1$ ,  $child$ ) // 递归调用
12:        end for
13:      end if
14:    else
15:      for all  $child \in node_1.children$  do
16:        TRAVERSEBVHTREE( $child$ ,  $node_2$ ) // 递归调用
17:      end for
18:    end if
19:  end if
20: end function

```

代价函数:  $T_{cost} = n_v * C_v + n_p * C_p + (n_u * C_u)$ (运动)

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ

# 问题的定义

## 凸包围 $k$ 面体

$k$ -Convex Bounding Polyhedron, 简称  $k$ -CBP,  
可通过  $k$  个半空间定义:

$$\begin{cases} k\text{-CBP} = \bigcap_{i=1}^k H_i \\ H_i = \{ \mathbf{p} \in \mathbb{R}^3 \mid \mathbf{n}_i \cdot \mathbf{p} \leq w_i, w_i \in \mathbb{R} \}, \end{cases} \quad (1)$$

其中,  $\mathbf{n}_i$  是半空间  $H_i$  的法向, 方向指向包围体外部,  $w_i$  是输入点集中沿  $\mathbf{n}_i$  方向投影的最大值。

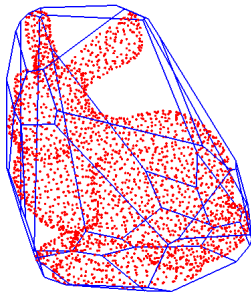
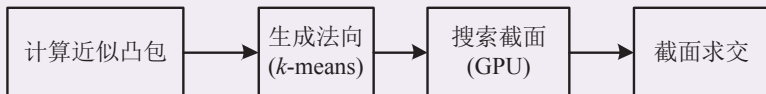


Figure 3: 34-CBP

# 算法流程

## 构造 $k$ -CBP 算法流程图



## 关键步骤

定法向： 结合近似内凸包和  $k$ -means

搜截面： GPU 中沿各法向搜索切点构造截面

求交点： 截面对偶映射求得交点

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ

# 近似凸包的构造

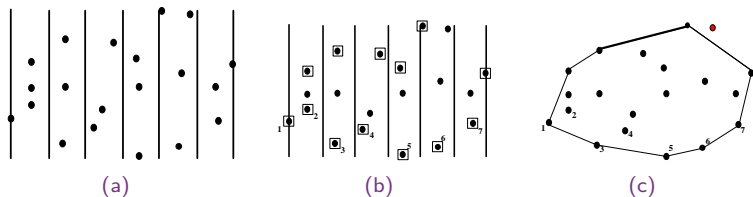


Figure 4: 二维近似内凸包的构造

构造近似内凸包 [2], 算法复杂度为  $O(n + \xi)$ , 扩展到三维为  $O(n + \xi^2 \log \xi)$ , 然后利用  $k$ -means 聚类.

## k-means 聚类

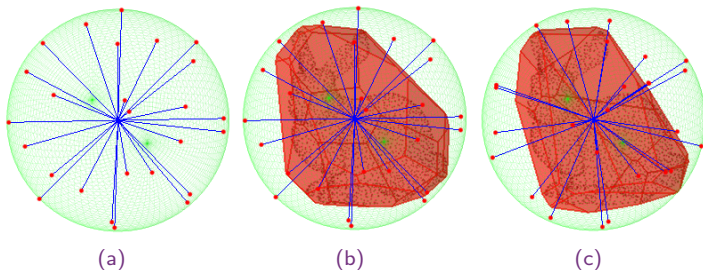


Figure 5: 通过聚类确定法向

聚类初始方向 (均匀分布). 距离度量 (余弦), 聚类更新中心点时将面片的面积作为权重即  $\mathbf{c}_i = \frac{\sum_{j=1}^n \omega_j \cdot \mathbf{n}_j}{\sum_{j=1}^n \omega_j}$ , 其中  $\mathbf{c}_i$  为第  $i$  类的中心点,  $\omega_j$  为法向  $\mathbf{n}_j$  所在面片对应的面积.

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ



# 搜索截面

等效于寻找最大投影值，即对每个法向  $\mathbf{n}_i$ ，从输入模型的所有点中寻找最大投影值的点作为切点进而确定  $\mathbf{n}_i$  对应的截面。时间复杂度为  $O(k \cdot n)$ ，其中  $k$  为法向数量， $n$  为模型所含点数。各法向的计算相互独立，借助 GPU 并行加速。

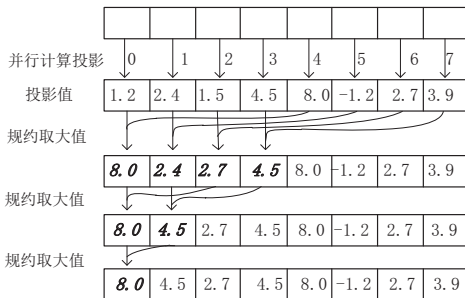


Figure 6: 并行规约求最大投影值

# 求交算法

法向  $\mathbf{n}(a, b, c)$  及平面上一点  $\mathbf{p}(x_0, y_0, z_0)$  确定, 转化为平面方程  $ax + by + cz = ax_0 + by_0 + cz_0 = d, d \neq 0$ . 对偶映射后的点为  $\mathbf{p}'(a/d, b/d, c/d)$ , 对这  $k$  个映射点求凸包, 凸包平面映射回原来的交点, 时间复杂度为  $O(k \log k)$ .

亦可直接通过枚举所有每 3 个平面交于 1 点的情况, 然后排除在平面外部的交点, 剩下的构成  $k$ -CBP 的顶点, 时间复杂度为  $O(k^3)[3]$ .

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ



## 凸包围多面体的生成速度

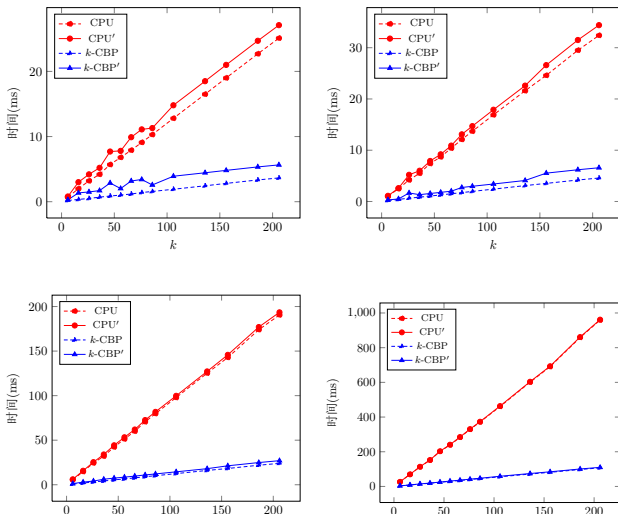


Figure 7: 本文算法与 CPU 算法对比 Budda(31k),Dinosaur(40k),Alice(224k), Bugatti(1011k)



图 7 中横纵坐标分别代表多面体面数和运行时间, 其中虚线代表搜索截面的过程, 实线为构造凸包围多面体总体耗时. 当模型点数量较大时, 搜索截面的过程占据了算法绝大多数时间, 且随着凸包围多面体的面数  $k$  值增加而线性增长, 这与搜索截面时间复杂度 ( $O(k \cdot n)$ ) 一致, 截面求交过程的时间复杂度为  $O(k \log k)$ , 当点数量极大时, 实线虚线几乎重合即求交等步骤耗时相比整体算法而言几乎可忽略.

Table 1: 本文算法与文献 [4] 算法对比

$k$	Apple(8118 points)			Bugatti(1010815 points)		
	SSE <sup>5</sup> (ms)	$k$ -CBP(ms)	Speedup	SSE(ms)	$k$ -CBP(ms)	Speedup
6	0.4	0.12	3.20	24.2	3.20	7.56
16	0.9	0.26	3.43	44.5	8.44	5.27
26	1.4	0.41	3.38	66.5	13.65	4.87
36	1.9	0.52	3.65	91.1	18.34	4.97
46	2.5	0.67	3.74	119.5	24.13	4.95
56	2.9	0.79	3.66	138.4	28.86	4.80
66	3.5	0.95	3.69	170.6	34.10	5.00
76	4.0	1.08	3.70	197.1	39.85	4.95
86	4.5	1.22	3.69	219.8	45.08	4.88
106	5.4	1.49	3.62	267.8	55.52	4.82
136	6.8	1.92	3.54	342.9	71.24	4.81
156	7.7	2.17	3.55	411.3	81.18	5.07
186	9.3	2.60	3.58	479.4	97.39	4.92
206	10.5	2.85	3.68	523.0	106.87	4.89

点数量较小时, 能够提高 3-4 倍速度, 模型变大, 加速比更大, Bugatti 模型的提速达到 4-8 倍.

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ



## 凸包围多面体的紧致程度

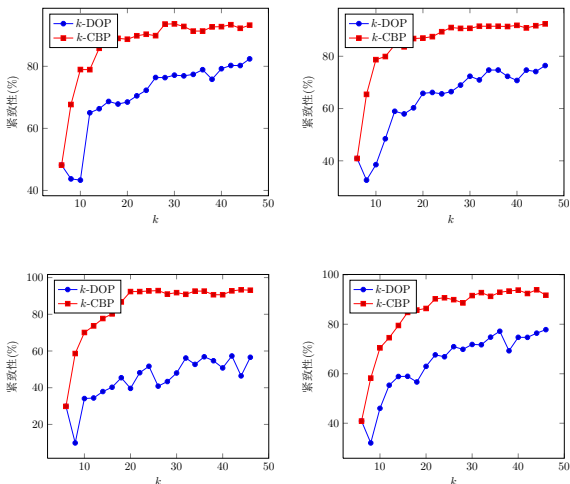


Figure 8: 紧致程度对比:  $k$ -DOP v.s  $k$ -CBP Apple(8k), Budda(31k), Dinosaur(40k), Alice(224k)  
 紧致程度用凸包与凸包围多面体的体积之比来量化.



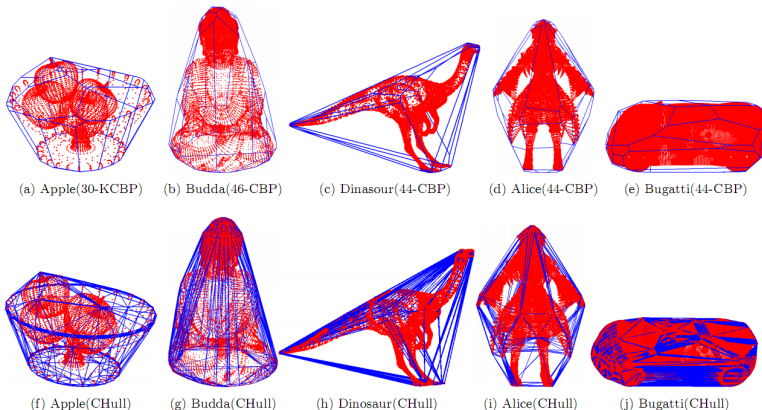
Table 2:  $k$ -CBP 与 QuickHull 凸包算法比较

Model	$f(\text{CHull})$	$f(k\text{-CBP})$	$\tau(k\text{-CBP})$	$t(\text{CHull}(\text{ms}))$	$t(k\text{-CBP}(\text{ms}))$
Apple	499	30	93.67%	5.5	1.30
Budda	1608	46	92.39%	21.3	2.86
Dinosaur	1240	44	93.34%	22.6	1.99
Alice	1332	44	93.92%	85.8	8.47
Bugatti	24654	44	95.06%	688.7	25.41

$\tau(k\text{-CBP})$  为凸包围多面体的紧致程度. 与凸包相比, 本文算法在大大简化包围体平面数量的同时能保持较好的紧致程度, 下图为可视化结果.



## 凸包围多面体的紧致程度

Figure 9:  $k$ -CBP 与凸包对比

# 目录

- 1 引言
  - 凸包围体
  - 碰撞检测算法
- 2 模型适应的凸包围多面体构造算法
  - 问题定义及算法流程
  - 截面法向的生成
  - 搜索截面及求交
- 3 实验与分析
  - 凸包围多面体的生成速度
  - 凸包围多面体的紧致程度
  - 凸包围多面体的简单应用
- 4 主要参考文献
- 5 感谢
- 6 FAQ

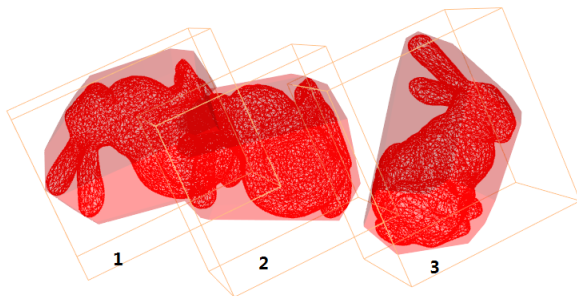


Figure 10:  $k$ -CBP 应用于碰撞检测示例

图中模型 1 与 2、2 与 3 的包围盒分别相交, 而其 16-CBP 仅 1 与 2 相交, 实际模型仅 1 与 2 相交. 不同数量的模型 (模型位置和旋转角度随机生成) 测试结果如下表所示.

Table 3:  $k$ -CBP 和包围盒应用于碰撞检测结果对比

n	c(Box)	c(16-CBP)	t(Box)	t(16-CBP)	r(Box)	r( $k$ -CBP)	n(Model)
10	0.1	1.8	26.0	0.1	0.00 %	100.00%	0
30	0.2	2.9	134.0	70.0	45.45%	83.33%	5
50	0.5	4.8	506.0	255.2	46.34%	86.36%	19
70	0.4	4.8	901.1	492.5	44.16%	80.95%	34
90	0.7	5.7	1324.0	734.7	41.82%	73.02%	46
100	0.7	7.8	1481.0	870.7	43.31%	75.34%	55
150	1.0	9.8	4153.1	2473.0	42.98%	70.75%	150
200	1.6	12.8	8049.3	4430.9	41.02%	71.32%	281

其中  $r(\text{Box})$ ,  $r(16\text{-CBP})$  分别表示包围盒、16-CBP 的命中率即用实际模型相交的数量除以包围体检测出来相交的数量. 模型和凸包围多面体是否相交都采用了普通 AABB 树的方式进行判断.

## 主要参考文献 I

- [1] J. T. Klosowski, M. Held, J. S. Mitchell, H. Sowizral, and K. Zikan, “Efficient collision detection using bounding volume hierarchies of k-dops,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 1, pp. 21–36, 1998.
- [2] J. L. Bentley, F. P. Preparata, and M. G. Faust, “Approximation algorithms for convex hulls,” *Communications of the ACM*, vol. 25, no. 1, pp. 64–68, 1982.
- [3] C. Ericson, *Real-time collision detection*. San Francisco, CA: Morgan Kaufmann Publishers, 2005.

## 主要参考文献 II

- [4] M. Karlsson, O. Winberg, and T. Larsson, “Parallel construction of bounding volumes,” in *The Annual Swedish Computer Graphics Association Conference(SIGRAD)*, 2010, pp. 65–69.

# 感谢

导师、学院



# FAQ

Thank you!