

## 基于可视外壳的 cage 生成\*

陈雪, 刘涛, 冯结青

(浙江大学 CAD & CG 国家重点实验室, 浙江 杭州 310058)

通讯作者: 冯结青, E-mail: jqfeng@cad.zju.edu.cn, http://www.cad.zju.edu.cn/home/jqfeng

**摘要:** 借助 cage 作为代理几何来处理高精度复杂模型, 正成为计算机动画与几何造型中的一种重要方法。目前, 已有的 cage 生成算法尚缺乏普适性, 很大程度上依赖于几何的表示和模型的复杂度。因此, 提出一种基于可视外壳的 cage 生成算法。通过逆向模拟计算机视觉领域可视外壳的生成过程获取 cage, 以实现 cage 生成与几何表示和模型复杂度无关的目标。实验结果表明, 该方法易于实现且效率高。

**关键词:** cage 生成; 可视外壳; 网格优化

**中图法分类号:** TP391      **文献标识码:** A

中文引用格式: 陈雪, 刘涛, 冯结青. 基于可视外壳的 cage 生成. 软件学报, 2013, 24(10): 2379-2390. <http://www.jos.org.cn/1000-9825/4413.htm>

英文引用格式: Chen X, Liu T, Feng JQ. Cage generation based on visual hull. Ruan Jian Xue Bao/Journal of Software, 2013, 24(10): 2379-2390 (in Chinese). <http://www.jos.org.cn/1000-9825/4413.htm>

### Cage Generation Based on Visual Hull

CHEN Xue, LIU Tao, FENG Jie-Qing

(State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310058, China)

Corresponding author: FENG Jie-Qing, E-mail: jqfeng@cad.zju.edu.cn, <http://www.cad.zju.edu.cn/home/jqfeng>

**Abstract:** Cage, which is adopted as the proxy geometry, plays an important role in processing high-precision and complex models in computer animation and geometric modeling. Currently, cage generation algorithms are heavily dependent on representation and complexity of the shape, which could not be applied to various models universally. To decouple the cage generation from the representation and complexity of the shape, the paper proposes a cage generation algorithm based on visual hull in computer vision. The algorithm can generate tight and coarse cage by inverse simulation of visual hull construction procedure. The results demonstrate usability and efficiency of the proposed algorithm.

**Key words:** cage generation; visual hull; mesh optimization

随着三维模型获取技术的飞速发展, 计算机图形学处理的模型精度和复杂度越来越高, 以往的几何处理算法在普适性和计算复杂度上都面临新的挑战。采用与原模型形状相似但顶点数较少的包围网格作为代理几何来衔接复杂模型与算法是一个趋势。在几何编辑与空间变形框架下, 代理几何需要包裹原模型, 被称为 cage。Cage 具有包裹性、相似性和简单性。其优势在于把直接处理高精度复杂模型的棘手问题转化为对 cage 的间接处理, 以 cage 为媒介, 把算法的复杂度与模型复杂度与其表示方式分离, 可大大降低算法复杂度并扩展算法的适用性。因此, cage 被广泛应用于空间变形、变形迁移、碰撞检测以及细分的控制网格。

直观上, cage 是一个低分辨率网格, 要求用尽可能少的面片逼近目标模型, 并保证目标模型嵌于 cage 内, 如图 1 所示。已有的 cage 生成算法<sup>[1-5]</sup>都致力于自动生成满足 cage 定义的控制网格, 但是它们忽略了 cage 出现的

\* 基金项目: 国家自然科学基金(60933007, 61170138); 国家重点基础研究发展计划(973)(2009CB320801); 新世纪优秀人才支持计划(NCET-10-0728)

收稿时间: 2012-08-13; 定稿时间: 2013-04-09

缘由:cage 的生成应独立于模型的几何表示和复杂度.

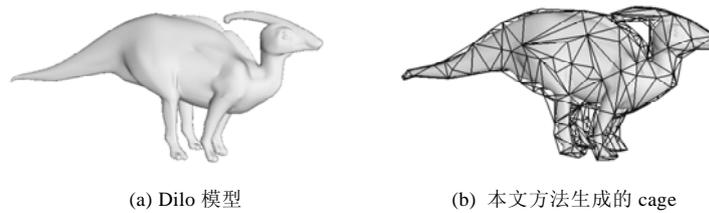


Fig.1 Cage generation for the Dilo model

图 1 Dilo 模型及其 cage

本文从引入 cage 用于形状编辑的动机出发,提出了具有普适性且高效的 cage 生成算法.图 1 是采用本文算法生成的 cage.为了克服已有 cage 生成方法对模型表示和复杂性的依赖,本文另辟蹊径,尝试着将计算机视觉领域的可视外壳(visual hull)技术引入到 cage 生成算法中.在计算机视觉领域,可视外壳是由多个视点和对应图像中物体的侧影轮廓线构成锥体的交集<sup>[6,7]</sup>,其目的是基于图像的轮廓信息重构几何物体.我们已知几何物体和视点,逆向模拟可视外壳的生成过程:首先,在不同视点下绘制物体,得到二值图像,对图像进行膨胀得到其简化的多边形轮廓,并与视点进行反投影得到视锥,多个视锥进行布尔交运算并经优化最终形成 cage.由于生成 cage 所需的视锥是由视点和绘制图像所决定,理论上凡是可绘制的几何物体都可以用本文方法生成其 cage,因此,本算法与模型的表示无关.对于计算机图形学中常用的几何表示形式,例如点云、网格、参数曲面、隐式曲面以及各种体模型,均可在给定视点下进行绘制以得到其绘制图像,进而通过视锥求交生成其 cage;其次,本文算法复杂度与视锥的数目相关,视锥的构造仅与模型在平面投影的二维轮廓线有关,而与模型几何信息无关,因此,算法的复杂度独立于模型的复杂度.综上所述,本文算法符合 cage 引入的初衷.本文的主要贡献有:

- (1) Cage 的生成与几何表示无关:可处理多边形网格、点云、多边形集合(polygon soup)、参数曲面、隐式曲面等.三维扫描的输出或图像序列重建的结果多为点云或多边形集合,本算法无需预处理,可直接应用;
- (2) Cage 的生成与模型的复杂度无关:通过将生成 cage 的主要过程转化到图像空间,实现了算法复杂度与模型复杂度的分离.Cage 的作用就是高精度模型的代理,只有以小的计算代价生成 cage 才能体现其价值;
- (3) 算法效率高且易于实现.

## 1 相关工作

作为一种高效、灵活的形状编辑和柔性物体动画生成方法,自由变形技术(free-form deformation,简称 FFD)最早由 Sederberg 和 Parry<sup>[8]</sup>提出,它以 Bézier 体的控制网格作为目标模型的代理几何.尽管各种扩展的 FFD 方法可使用非平行六面体网格,但受插值和光滑性约束的限制,控制网格的固定拓扑形状制约了变形的灵活性.于是,引发了对具有更多自由度的控制网格的空间变形技术的探寻.例如,基于细分体和细分曲面的 FFD<sup>[9,10]</sup>,基于广义重心坐标插值的空间变形方法<sup>[11-14]</sup>等.在这些方法中,控制网格不再有固定的拓扑和几何结构,而是逼近目标模型形状、顶点数较少的 2-流形网格,被称为 cage.随着模型精度、复杂度以及实时编辑要求的不断提高,cage 作为代理几何受到了广泛关注.例如,文献[15]结合广义均值坐标与梯度域方法进行网格编辑,文献[16]将格林坐标引入基于物理变形的框架中,文献[2,17]分别提出了基于 cage 的变形迁移等.

虽然 cage 在几何处理方面得到广泛应用,但生成满足要求的 cage 一直是一个难题.文献[11-14]采用的 cage 都由手工构造,借助建模工具的方法繁琐、耗时,对于非专业人士很难掌握,从而影响基于 cage 的应用和推广.鉴于自动生成 cage 困难重重,文献[18-20]分别提出了有关 cage 序列恢复的算法.这些方法推动了基于 cage 变形在动画中的应用.但初始 cage 的质量在很大程度上影响动画效果,因此,研究自动生成高质量 cage 的算法具有

重要意义。

Xian 等人<sup>[1]</sup>提出了基于体素化的自动 cage 生成方法:首先,将模型的 AABB(axis aligned bounding box)包围盒体素化(voxelization);将体素(voxel)分为体外、体内以及与模型相交的特征体素 3 类;抽取所有与特征体素相邻的体外体素的面,并三角化后得到初始包围网格;最后,对包围网格进行光顺处理得到 cage。该方法本质上对模型进行了均匀采样处理,采样的频率就是体素化的分辨率。体素化的分辨率直接影响 cage 的形状表达能力:分辨率过高会增加计算负担,分辨率低则难以准确捕捉模型的形状。李琳等人<sup>[4]</sup>也通过体素化抽取三维距离场等值面形成 cage。虽然他们生成的 cage 的网格质量较高,但计算距离场前需要简化模型来提高效率,抽取等值面重构的网格也需简化才能形成最终的 cage。两阶段的网格简化,对 cage 的包裹性缺乏理论上的保证。

文献[2,3,15]都是基于保凸渐进简化的思想<sup>[21]</sup>:缩边操作后,新点的位置必须保证在该边一环邻域的外侧。Ben-Chen 等人<sup>[2]</sup>利用不断迭代偏移、重构、简化的过程生成 cage。均匀采样模型的顶点,并沿法向方向偏移,泊松重构后再简化重构网格,直到生成满意的结果。但泊松重构的网格不保证一定包围目标模型,而基于简化的方法容易在模型 U 型区域(凹陷部位)发生自交。

体素化和简化法生成的 cage 都不具有语义信息,难以反映用户的编辑意图。Xian 等人<sup>[5]</sup>使用 OBB(oriented bounding box)包围盒自动生成 cage:首先,构造目标模型的树状 OBB 包围盒;然后,无缝拼接所有 OBB 包围盒;最后,抽取包围盒的外表面形成 cage。上述方法生成的 cage 能够体现模型的大致形状,在期望变形的区域(如关节)提供相应的控制顶点。但 cage 与原模型不够贴近,使得难以在细节丰富区域反映用户的编辑意图。针对关节模型, Ju 等人<sup>[22]</sup>引入了可重用蒙皮模板的 cage 变形, cage 在骨架的指导下进行构造:在关节和骨骼处预定义包围模型的模板,然后连接相邻模板形成初始的 cage。该方法生成的 cage 具有语义信息,但需要额外交互以处理 cage 自交的情况。文献[23]采用类似的思想构建 cage,用于角色模型的肌肉建模,并通过在参数空间求解优化问题以解决 cage 自交问题。

在几何处理与动画中,基于 cage 变形的初衷就是将几何处理算法的复杂度与模型的复杂度和表示形式分离,因此, cage 的生成也应该遵循这一原则。但目前的 cage 生成算法都不能完全具备这两点属性。从算法复杂度来衡量,除文献[22]外,其余算法复杂度都与模型复杂度密切相关。对输入模型的要求,文献[22,23]基于骨架信息,体素化和简化法<sup>[1,3,4]</sup>只能处理三角网格,文献[2]需要顶点的法向信息,文献[5]要求必须是封闭的模型。本文正是以上述两个要求为出发点,提出了一种新的 cage 生成方法。

## 2 基于可视外壳的 cage 生成

### 2.1 算法概述

可视外壳是一种从轮廓线获取形状的三维重建技术<sup>[6,7]</sup>,简单地讲,就是从不同视角观察空间物体,由物体的侧影轮廓线与对应视点形成的空间区域的交集,即为可视外壳。显然,可视外壳是物体外形的逼近,且包含该物体。Cage 是包围模型的低分辨率 2-流形网格。因此,传统的可视外壳可以看作是一种与模型部分相切的特殊 cage。由于图形的绘制流水线本身就是模拟相机的成像过程,所以在绘制流水线下,模拟可视外壳的构造过程是可行的。本文提出的基于可视外壳的 cage 生成算法主要包含如下步骤:

- (1) 在给定视点下,对输入模型进行二值绘制。图 3 中 3 个视点分别为沿 3 个坐标轴的无穷远,即模型在 3 个正交视图下的二维投影;
- (2) 分别对投影的二值图像进行距离变换操作,根据 cage 的包裹程度选择恰当的距离值,提取距离变换图像中相对于该距离值的等值线,并对等值线进行简化,形成模型在投影平面的二维包围盒,具体流程如图 4 所示。其中,图 2(b)的折线为简化等值线;
- (3) 每个视点和对应的二维包围盒在空间的投影构成视锥。图 2(c)中,平行于坐标轴的 3 个柱面即为 3 个无穷远视点下的视锥(平行投影);
- (4) 上述所得多个视锥进行布尔求交运算生成可视外壳,形成初始的 cage。如图 2(d)所示;
- (5) 对初始 cage 进行网格优化生成最终的 cage。如图 2(e)所示。

图2所示为本算法的流程,图3进一步描述了可视外壳的构造方式.

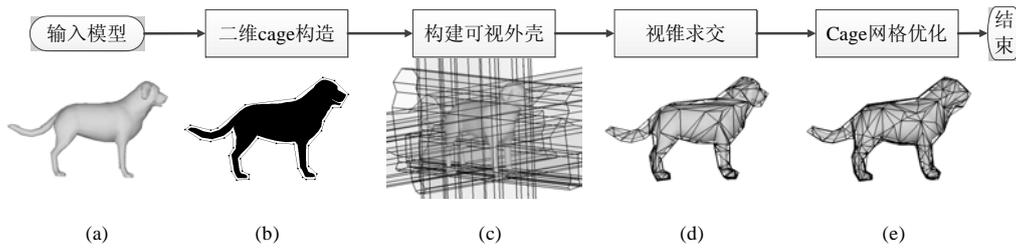


Fig.2 Flowchart of cage generation algorithm based on visual hull

图2 基于可视外壳的 cage 生成算法流程图

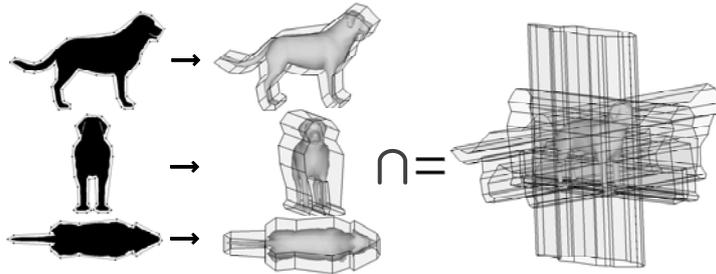
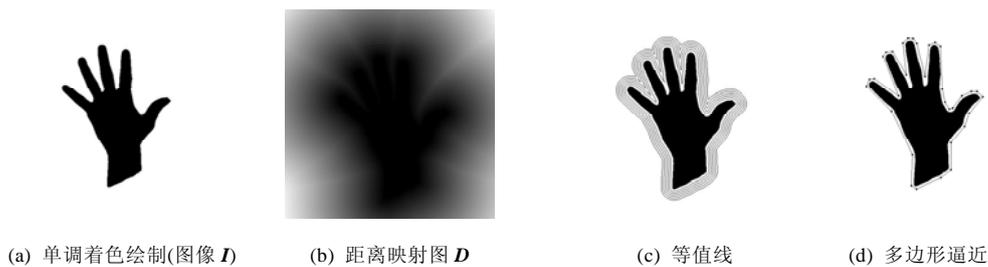


Fig.3 Construction of visual hull

图3 可视外壳的构造



(a) 单调着色绘制(图像  $I$ )

(b) 距离映射图  $D$

(c) 等值线

(d) 多边形逼近

Fig.4 Construction of 2D bounding box

图4 二维包围盒的构造

## 2.2 图像处理:二维cage的构造

类比于计算机视觉,在图形绘制流水线中观察者的位置相当于相机位置,绘制图像相当于所拍摄的照片.这里采用二值着色绘制即满足要求,绘制代价非常低.直接用三维模型在屏幕空间投影的外轮廓生成可视外壳与模型相切,不满足 cage 包裹性要求.因此,需膨胀二维投影区域来构造 cage.膨胀程度决定了 cage 的紧致度.为了增加对 cage 大小的可控性,对绘制图像采用距离变换,用距离值(用  $dist$  表示)作为控制参数.距离变换的定义为

$$D(p) := \min\{d(p,q) | I(q)=0\} \quad (1)$$

其中,图像  $I$  为上述给定视点下绘制模型所得的二值图像,这里,目标物体的灰度值设置为 0,  $d(\cdot, \cdot)$  表示任意背景像素到目标物体像素间的欧氏距离,距离变换图像中每一个像素值  $D(p)$  反映每一个背景像素  $p$  与目标物体的远近程度.本文采用文献[24]提出的在线性时间内求解欧氏距离变换的方法.根据距离映射图可以构成一系列等距离值的曲线(如图 4(c)所示),以实现用距离值参数控制轮廓大小的需求.

Cage 要用尽可能少的面片描述模型的形状.简单地距离等值线曲线进行反投影,会产生过于稠密的网格.

为此,我们采用平面多边形逼近等值线.曲线多边形逼近的方法有很多,其中,Douglas-Peucker(DP)<sup>[25]</sup>算法能够准确反映轮廓曲线的特征,且鲁棒性好.算法首先连接曲线上距离最远的两个点,再查找曲线上距离该线段最远的点,形成初始的多边形;反复迭代,不断将曲线上距离多边形最远的点(该点到垂足的线段只与多边形有一个交点)加入多边形顶点序列,直到曲线上所有点到多边形的最短距离小于阈值(用 *approx* 表示),算法终止.

图 4 所示为二维包围盒的构造过程:首先,采用二值着色绘制三维模型;绘制所得图像经距离变换得到距离映射图,不同的距离值对应于一系列等值线;最后,对选取的等值线进行多边形逼近.

### 2.3 构造可视外壳:初始cage的生成

可视外壳的生成算法主要分为两大类<sup>[7]</sup>:基于体素和基于表面的方法.体素方法首先将模型的包围盒体素化,体素与视锥进行求交测试,所有位于视锥内部的体素形成对原模型的逼近表示.该方法的优点是算法简单且易于实现,但其时间和空间复杂度较高,且体素的大小直接影响重建的精度.若采用该方法生成的 cage 会产生与文献[1]同样的缺陷.基于表面的方法是对视点与物体的侧影轮廓线形成的锥体进行布尔求交运算,得到多面体逼近.视点越多,重建的结果越逼近真实模型,但重建模型的复杂度也越高.Cage 的目的是构造与模型相似且分辨率低的网格,这与可视外壳重建的目标不同,所以只要保证所选视点能够完整描述模型形状即可.此外,从重建结果看,体素化方法需要从边界体素中进一步提取外表面才能得到网格表示.因此,本文选择视锥求交的方法生成初始 cage.

在第 2.2 节中,我们已经获得了物体的轮廓信息,构造视锥还需要视点的空间位置.在计算机视觉领域,通过棋盘格标定求出拍摄相机的内部参数,再由内部参数确定相机位置.而在绘制流水线中,视点位置可由用户指定,且模型本身位于近裁剪平面和远裁剪平面之间,锥体可扩展为棱台.因此,无需计算视点信息,使得视锥的构造更为简单.利用投影变换简单地确定二维 cage 分别在远、近裁剪平面上的对应点,连接同一点反投影的两个空间点,形成棱台状的视锥.本文的重点是构造 cage,重建可视外壳不是本文关注的焦点,所以采用实体造型中的多面体正则化布尔运算<sup>[26]</sup>进行视锥求交得到初始的 cage.

### 2.4 网格优化

经由正则化布尔运算得到的初始 cage 可能包含退化三角形和细长三角形,但不会存在自相交现象.Cage 的质量会影响后续几何处理算法的鲁棒性和有效性,因此,需要采取恰当的网格优化方法以改善网格质量.但传统的优化方法会破坏 cage 的包裹性.本文选取的优化算法不会引入自相交,且会保持初始 cage 的拓扑.由于 cage 是对原始模型非常粗糙的逼近,所以 cage 不会保留原始模型的细节,仅仅刻画其宏观形状.而其中退化三角形和细长三角形的面积接近于 0,承载的几何信息较少,消除后对网格整体几何形状的影响甚微.所以在满足 cage 的各种要求的前提下,网格优化主要处理退化和细长三角形.其中,退化三角形包括 3 个顶点重合、2 个顶点重合和 1 个顶点位于对边上 3 类,可以简单地识别并消除,这里不再赘述,下面重点介绍细长三角形的处理方法.

细长三角形可以分为两类,如图 5 所示:(a) needle:三角形最短边与最长边的比值小于 0.2;(b) cap:三角形最大内角大于  $170^\circ$ ,且不为 needle.对于 needle 三角形,容易通过边收缩消除;但 cap 三角形难以直接处理,需经过分割转化为 needle 三角形.若在局部范围内通过分割处理一个 cap 三角形,那么共享被分裂边的邻接三角形也会被一分为二,此时容易产生新的 cap 三角形(如图 6(b)所示).为避免再次产生 cap,必须从全局出发,用过 cap 点的平面对初始 cage 进行分割和三角化,如图 6(c)所示.

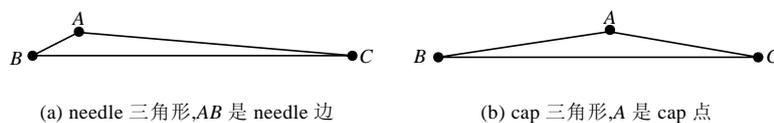


Fig.5 Classification of slim triangles

图 5 细长三角形分类

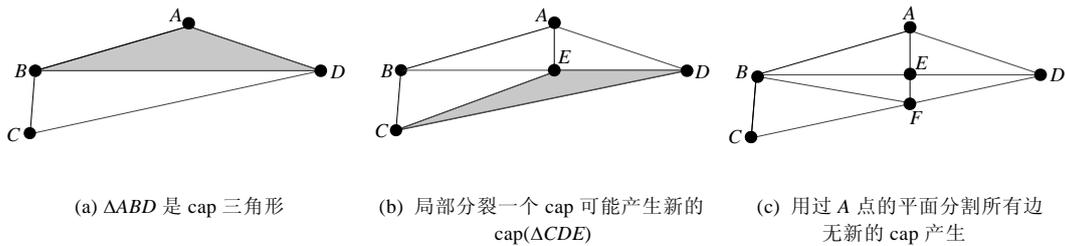


Fig.6 Elimination of cap triangle

图 6 cap 三角形的消除

全局分割的思想类似于文献[27],算法大致分为两步:

- (1) 顶点分类(如图 7(b)所示).切割平面为过 cap 点且垂直于 cap 三角形的平面.网格上所有顶点按到切割平面的有向距离分为平面以上的点(above)、平面以下的点(below)和平面上的点(on);
- (2) 三角形分割(如图 7(c)所示).根据边两个端点的属性判断该边是否与分割平面相交:两个端点分属于 above 和 below 的边与剖面相交.只有一条边与剖面相交的三角形分裂为两个三角形,有两条边相交的分裂为 3 个三角形;否则保持不变.以上步骤保证每次至少处理一个 cap,且不增加新的 cap.

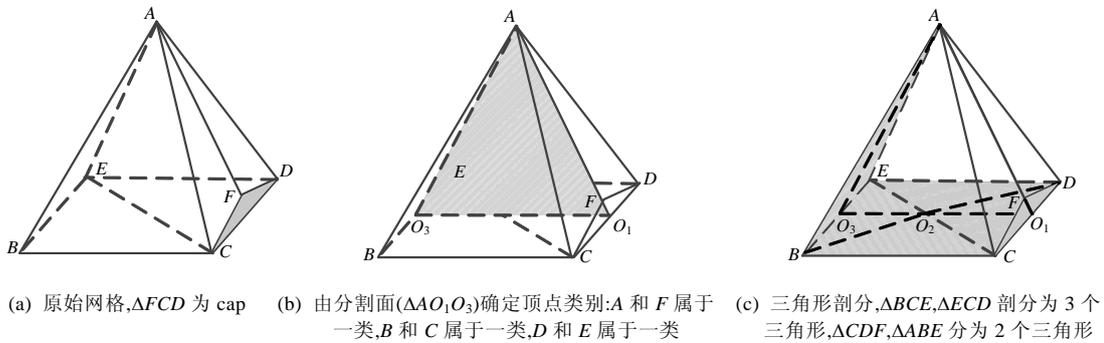


Fig.7 Globally cutting a cap triangle

图 7 全局分割处理

通过上述全局分割,所有 cap 三角形都转化为了 needle 三角形.我们在保证 cage 包裹性的前提下,通过收缩 needle 边可以达到消除细长三角形的目的.图 8 描述了 needle 边的处理方式.假设边 BC 是 needle 边,存在两种边收缩方式:一种是 B 固定,C 合并到 B;另一种是 C 固定,B 合并到 C.我们选择固定点位于移动点的一环邻域所有面的外侧的收缩方式,该方向的边收缩是可行的,保证新生成的 cage 不与模型相交,如图 8(b)所示.

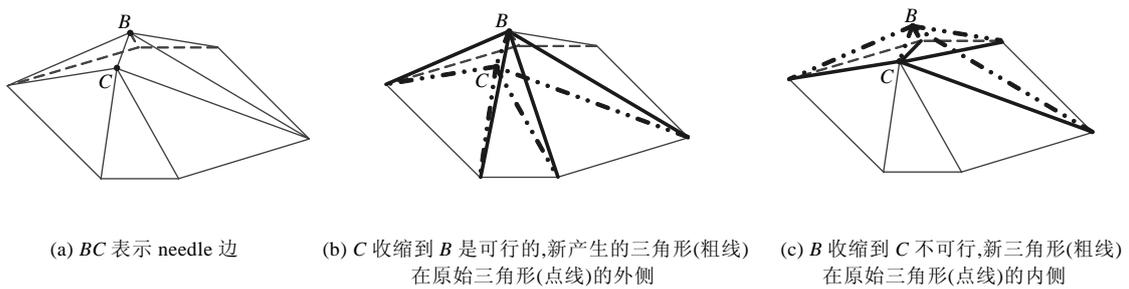


Fig.8 Elimination of needle triangles by contracting needle edge

图 8 收缩 needle 边以消除 needle 三角形

### 3 实验结果与分析

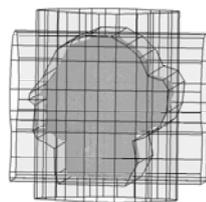
本文在配置为 Intel Core 2 Quad 2.5GHz CPU,4GB 内存的平台上用 VC++ 2008 和 OpenGL 实现了基于可视外壳的 cage 生成算法.视点的位置、数目以及选取等值线的距离值作为用户可调参数.视点数目一定程度上决定了 cage 与原模型的相似程度.根据 cage 的定义和要求,应采用尽可能少的视点来描述模型形状.利用平行投影(视点无穷远)可以减少对称方向的视点,所以本节的实验结果均采用平行投影.视点位置的组合决定了 cage 的初始形状.对于复杂模型,至少需要两个方向的视锥描述 U 型区域分支的形状.通过测试我们发现,3 个正交视点位置足够刻画对称模型的形状;而非对称模型根据 U 型区域原则,可在其他方向引入视点.本文采用 IRIT 建模库(<http://www.cs.technion.ac.il/~irit/>)中的布尔运算进行视锥求交.本节大部分模型的 cage 都是采用三视点,表 1 列出了所有实验数据及结果数据.

**Table 1** Data on the visual hull based algorithm

**表 1** 本文测试数据及运行时间统计

模型名称	模型顶点数	Cage 顶点数	视点数目	初始 cage 生成时间(s)	网格优化时间(s)
Dilo (Fig.1/Fig.15(a))	27 174	217	3	0.218	4.524
Dog (Fig.2/Fig.18)	35 474	220	3	0.156	2.168
Head (Fig.9)	80 859	218	3	0.094	0.936
Moai (Fig.10)	14 475	193	3	0.203	1.060
Utah teapot (Fig.11)	-	128	3	0.078	0.047
Man (Fig.12)	10 050	164	3	0.110	0.530
Dilo (Fig.15(b))	27 174	205	3	0.313	1.435
Dilo (Fig.15(c))	27 174	201	3	0.094	0.842
Hand (Fig.16(a))	35 074	561	3	0.376	110.184
Hand (Fig.16(b))	35 074	262	3	0.422	7.847
Hand (Fig.16(c))	35 074	141	3	0.110	1.856

图 9~图 12 所示为具有不同几何表示形式的模型的 cage 生成实例.



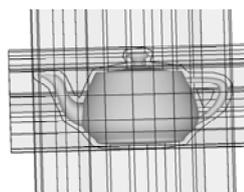
(a) 构建可视外壳的 3 个视锥,  $dist=4, approx=5$



(b) 生成的 cage

**Fig.9** Point cloud model Head and its cage

图 9 点云 Head 模型及其 cage



(a) 构建可视外壳的 3 个视锥,  $dist=6, approx=5$



(b) 生成的 cage

**Fig.10** Parametric surface model Utah teapot and its cage

图 10 参数曲面表示的 Utah teapot 模型及其 cage

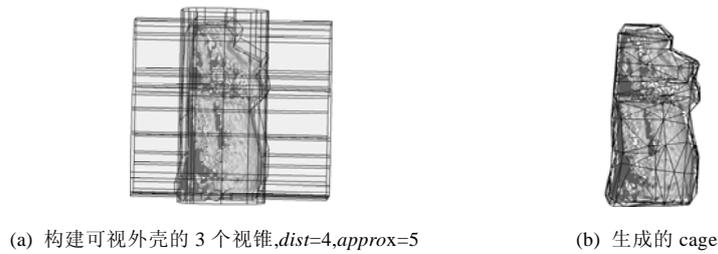


Fig.11 Polygon soup model Moai and its cage  
图 11 多边形集合 Moai 模型及其 cage

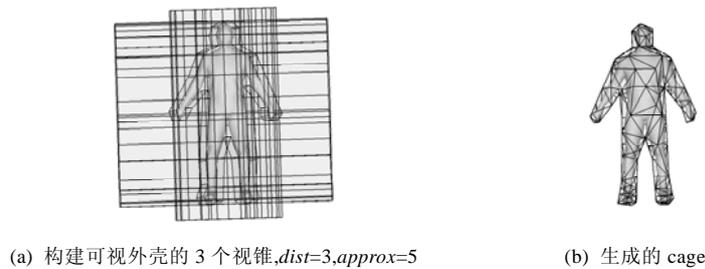


Fig.12 Triangle mesh model Man and its cage  
图 12 三角网格 Man 模型及其 cage

图 9 是点云模型及其 cage,若模型为稀疏的点云,只需对投影图像进行膨胀和收缩形态学操作来填补空洞.图 10 测试了由 32 个双三次 Bézier 曲面表示的 Utah teapot,我们使用 OpenGL 的求值器函数来绘制该曲面.图 11 是多边形集合模型的结果,多边形集合是由一些无拓扑连接关系的三角形面片集合组成的非封闭网格,目前已有 cage 生成方法都不能直接处理这类表示.图 12 是常见的三角网格模型的结果.理论上,本算法也能处理隐式曲面、体数据,但这两类模型通常转化为网格来绘制.上述例子说明,本算法适用于任何可以绘制的几何表示形式.

图 13、图 14 分别为改进 OBB 方法<sup>[5]</sup>、基于保凸简化法<sup>[3]</sup>以及本文方法生成的 cage.本文方法生成的结果接近保凸简化法的结果,比 OBB 法更贴近模型.因为改进 OBB 的 cage 是由长方体堆砌构成,长方体的固有形状限制了其整体的表达能力;而本文方法借助模型的侧影轮廓线信息,能够准确地表达模型形状,更符合 cage 形似的要求.

表 2 为本文方法与文献[3,5]中两种方法运行时间的对比.根据 Dilo 模型和 Dancer 模型的 cage 生成时间及结果可以看出:本算法运行时间远小于文献[3,5]中的运行时间,说明本算法不依赖于模型的复杂度,只与模型的轮廓和视点信息有关,计算代价小.文献[3]采用缩边简化,新点的位置计算需要求解一个线性规划问题,所以时间复杂度与模型待收缩边数成正比关系.文献[5]需要体素化,则算法的效率与模型的复杂度相关,从图 13(a)和图 14(a)可以看出,要使 cage 更贴近模型,需要增加层次 OBB 树的深度,运行时间也随之增长.

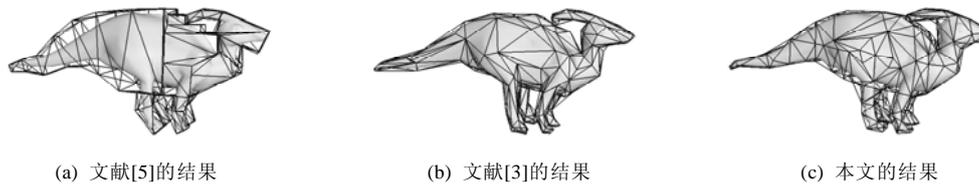


Fig13 Comparison with two methods<sup>[3,5]</sup> for the Dilo model  
图 13 与文献[3,5]中两种方法对比 Dilo 模型的结果



Fig14 Comparison with two methods<sup>[3,5]</sup> for the Dancer model

图 14 与文献[3,5]中两种方法对比 Dancer 模型的结果

Table 2 Runtime comparison with other methods<sup>[3,5]</sup>

表 2 与文献[3,5]中方法的运行时间对比

模型名称	模型顶点数	OBB 方法 <sup>[5]</sup>		简化方法 <sup>[3]</sup>		本文可视外壳方法	
		Cage 顶点数	运行时间(s)	Cage 顶点数	运行时间(s)	Cage 顶点数	运行时间(s)
Dilo (Fig.13)	5 000	145	362	217	432.872	217	4.665
Dancer (Fig.14)	2 396	41	20	221	101.9	221	2.09

下面对二维 cage 构造中的两个参数 *dist* 和 *approx* 的作用进行讨论。*dist* 表示等值线的值,其大小决定 cage 与模型的贴近程度,*dist* 值越大,等值线越钝化。对于物体表面曲率较大的区域,只有设置较小的 *dist* 值才能区分该部分,如图 13(a)所示。*approx* 是 DP<sup>[25]</sup>算法的参数,用于控制二维等值线曲线的多边形逼近度。*approx* 值越小,曲线的离散度越低,导致求交后退化面数目增加,降低网格优化率,且最终 cage 的面片数也随之增加。实验结果表明,针对大部分模型,当 *approx* 值取 5 时,生成的 cage 满足简洁、形似以及不与模型相交的要求。图 15 所示为 *approx* 值相同(取 5)、*dist* 值分别取 1,3,6 时生成 Dilo 模型的 cage 的情况。图 16 描述了固定 *dist* 值(取 2),*approx* 值对结果的影响。此外,两个参数对算法运行时间也有影响,*dist* 值越大,运行时间越短;*approx* 值越小,运行时间越长。尤其是优化时间增长,见表 1。本文算法的空间复杂度主要体现在两方面:一是视锥数据的存储开销,仅与视锥数目成正比,而与模型本身无关;二是网格优化模块对初始 cage 以及中间优化结果的存储开销,其数据量远小于输入模型。因此,本算法的空间复杂度与模型复杂度无关。



Fig.15 Effects of parameter *dist*

图 15 *dist* 参数的作用

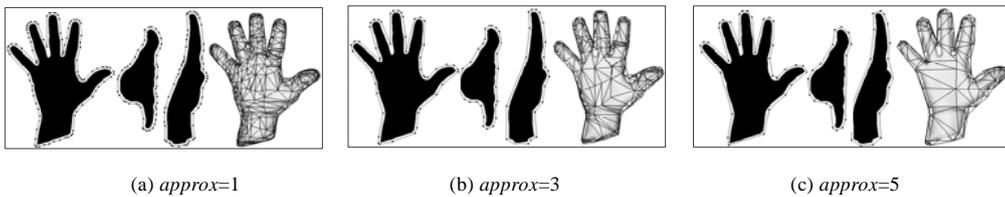


Fig.16 With the increase of parameter *approx*, the resolution of generated cage decreases

图 16 随着参数 *approx* 的增大,生成的 cage 分辨率降低

此外,本算法还可以选择绘制模型的部分组件实现局部 cage 的生成,为局部编辑提供可能.如图 17 所示.



Fig.17 Hand model and its partial cage

图 17 生成 Hand 模型的局部 cage

最后,基于本文生成的 cage,结合格林坐标<sup>[14]</sup>对模型进行交互式形状编辑.图 18 所示为对模型的头、四肢和尾巴编辑后的结果.

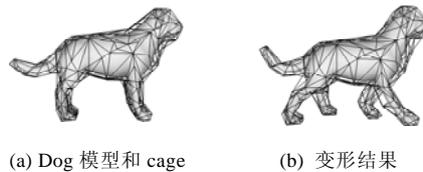


Fig.18 Dog and its cage-based deformation

图 18 基于 cage 的变形结果

#### 4 总结与展望

本文提出了一种新的 cage 生成算法.通过将计算机视觉领域的可视外壳技术引入几何建模中,解决了已有的基于几何信息的 cage 生成算法中存在的几何表示相关和模型复杂度相关的问题.基于可视外壳生成 cage 的算法思想简单且易于实现.通过逆向模拟可视外壳生成过程,以不同视点下输入模型在屏幕上的投影看作相机拍摄的照片,利用距离变换扩充目标区域以保证 cage 的包裹性,用多边形逼近轮廓以保证 cage 的简单性,二维多边形轮廓反投影形成的空间区域求交后得到初始 cage,运用网格优化获得最终结果.本文算法除了具有普适性的优点外,还可通过等值线的值来控制 cage 的逼近程度以及生成局部 cage.

针对本文算法的进一步研究应集中于如下几个方面:

- 首先,如何证明网格优化过程中可以有效地保持 cage 的包裹性,将为本文方法提供重要的理论保证.在我们进行的大量实验中,优化算法对于网格调整幅度很小,所得到的 cage 均具有良好的包裹性.事实上,如果网格优化过程中对于初始 cage 的调整幅度小于距离变换的距离值,那么 cage 的包裹性可以得到保证;
- 其次,如何在生成的 cage 引入语义信息,满足用户高效地选择控制点以获取期望的变形效果的需求,是 cage 生成中另一个重要问题.众所周知,计算机无法预先获取用户的交互意图,在 cage 生成过程引入简单、直观的交互意图十分重要;
- 最后,如何自动选取恰当数目的视点,高效地生成具有良好包裹性的 cage,是另一个重要的问题.在本文实验中的标准姿势模型,采用 3 个正交视点通常能够生成质量较高的 cage.对于自遮挡较多的模型,如章鱼模型,3 个正交视点难以准确地刻画模型的形状,则需要更多视点的辅助.

**致谢** 本文工作得到了浙江大学 CAD&CG 国家重点实验室蔺宏伟副教授和华南理工大学计算机系冼楚华博士的支持和帮助,在此表示衷心的感谢.

**References:**

- [1] Xian CH, Lin HW, Gao SM. Automatic generation of coarse bounding cages from dense meshes. In: Yong JH, Spagnuolo M, eds. Proc. of the IEEE Int'l Conf. on Shape Modeling and Applications. Los Alamitos: IEEE Computer Society Press, 2009. 21–27. [doi: 10.1109/SMI.2009.5170159]
- [2] Ben-Chen M, Weber O, Gotsman C. Spatial deformation transfer. In: Tamstorf R, Fellner D, eds. Proc. of the 2009 ACM SIGGRAPH/Eurographics Symp. on Computer Animation. New York: ACM Press, 2009. 67–74. [doi: 10.1145/1599470.1599479]
- [3] Deng ZJ, Luo XN, Miao XP. Automatic cage building with quadric error metrics. Journal of Computer Science and Technology, 2011,26(3):538–547. [doi: 10.1007/s11390-011-1153-4]
- [4] Li L, Li GQ, Huang LW. Generation of equidistance cages. Journal of Computer-Aided Design & Computer Graphics, 2011,23(6): 956–963 (in Chinese with English abstract).
- [5] Xian CH, Lin HW, Gao SM. Automatic cage generation by improved OBBs for mesh deformation. The Visual Computer, 2012, 28(1):21–33. [doi: 10.1007/s00371-011-0595-6]
- [6] Laurentini A. The visual hull concept for silhouette based image understanding. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1994,16(2):150–162. [doi: 10.1109/34.273735]
- [7] Franco JS, Boyer E. Exact polyhedral visual hulls. In: Aldridge R, Bangham A, eds. Proc. of the 14th British Machine Vision Conf. London: BMVA Press, 2003. 329–338.
- [8] Sederberg TW, Parry SR. Free-Form deformation of solid geometric models. SIGGRAPH Computer Graphics, 1986,20(4):151–160. [doi: 10.1145/15886.15903]
- [9] MacCracken R, Joy KI. Free-Form deformation with lattices of arbitrary topology. In: Blau B, Dodsworth C, eds. Proc. of the 23rd Annual Conf. on Computer Graphics and Interactive Techniques. New York: ACM Press, 1996. 181–188. [doi: 10.1145/237170.237247]
- [10] Feng JQ, Shao J, Jin XG, Peng QS, Forrest AR. Multiresolution free-form deformation with subdivision surface of arbitrary topology. The Visual Computer, 2006,22(1):28–42. [doi: 10.1007/s00371-005-0351-x]
- [11] Ju T, Schaefer S, Warren J. Mean value coordinates for closed triangular meshes. ACM Trans. on Graphics, 2005,24(3):561–566. [doi: 10.1145/1186822.1073229]
- [12] Lipman Y, Kopf J, Cohen-Or D, Levin D. GPU-Assisted positive mean value coordinates for mesh deformation. In: Belyaev A, Garland M, eds. Proc. of the 5th Eurographics Symp. on Geometry Processing. Aire-la-Ville: Eurographics Association Press, 2007. 117–123.
- [13] Joshi P, Meyer M, DeRose T, Green B, Sanocki T. Harmonic coordinates for character articulation. ACM Trans. on Graphics, 2007, 26(3):71. [doi: 10.1145/1276377.1276466]
- [14] Lipman Y, Levin D, Cohen-Or D. Green coordinates. ACM Trans. on Graphics, 2009,27(3):78. [doi: 10.1145/1360612.1360677]
- [15] Huang J, Shi XH, Liu XG, Zhou K, Wei LY, Teng SH, Bao HJ, Guo BN, Shum HY. Subspace gradient domain mesh deformation. ACM Trans. on Graphics, 2006,25(3):1126–1134. [doi: 10.1145/1179352.1142003]
- [16] Martin S, Huber C, Kaufmann P, Gross M. Shape-Preserving animation of deformable objects. In: Magnor M, Rosenhahn B, eds. Proc. of the Vision, Modeling, and Visualization. Braunschweig: DNB, 2009. 65–72.
- [17] Chen L, Huang J, Sun HQ, Bao HJ. Cage-Based deformation transfer. Computer & Graphics, 2010,34(2):107–118. [doi: 10.1016/j.cag.2010.01.003]
- [18] Xu WW, Zhou K, Yu YZ, Tan QF, Peng QS, Guo BN. Gradient domain editing of deforming mesh sequences. ACM Trans. on Graphics, 2007,26(3):84. [doi: 10.1145/1275808.1276482]
- [19] Savoye Y, Franco JS. Cage-Based tracking for performance animation. In: Kimmel R, Klette R, eds. Proc. of the 10th Asian Conf. on Computer Vision. Berlin, Heidelberg: Springer-Verlag, 2011. 599–612. [doi: 10.1007/978-3-642-19318-7\_47]
- [20] Thiery JM, Tierny J, Bouhekeur T. CageR: Cage-Based reverse engineering of animated 3D shapes. Computer Graphics Forum, 2012,31(8):2303–2316. [doi: 10.1111/j.1467-8659.2012.03159.x]
- [21] Platis N, Theoharis T. Progressive hulls for intersection applications. Computer Graphics Forum, 2003,22(2):107–116. [doi: 10.1111/1467-8659.00653]

- [22] Ju T, Zhou QY, van de Panne M, Cohen-Or D, Neumann U. Reusable skinning templates using cage-based deformations. ACM Trans. on Graphics, 2008,27(5):1–10. [doi: 10.1145/1457515.1409075]
- [23] Yang XY, Chang J, Southern R. Automatic cage construction for retargeted muscle fitting. The Visual Computer, 2013,29(5): 369–380. [doi: 10.1007/s00371-012-0739-3]
- [24] Felzenszwalb FP, Huttenlocher DP. Distance transforms of sampled functions. Theory of Computing, 2012,8(19):415–428.
- [25] Douglas DH, Peucker TK. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Canadian Cartographer, 1973,10(2):112–122.
- [26] Hoffmann CM. Geometric and Solid Modeling: An Introduction. San Francisco: Morgan Kaufmann Publishers, 1989. 80–93.
- [27] Botsch M, Kobbelt LP. A robust procedure to eliminate degenerate faces from triangle meshes. In: Ertl T, Girod B, eds. Proc. of the Vision Modeling and Visualization Conf. Stuttgart: Aka GmbH, 2001. 283–289.

#### 附中文参考文献:

- [4] 李琳,李桂清,黄力慰.等距 cage 生成.计算机辅助设计与图形学学报,2011,23(6):956–963.



陈雪(1985—),女,陕西宝鸡人,博士生,CCF 学生会员,主要研究领域为几何造型.  
E-mail: chenxue@zjucadcg.cn



冯结青(1970—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机图形学,几何造型,科学数据可视化.  
E-mail: jqfeng@cad.zju.edu.cn



刘涛(1987—),男,博士生,CCF 学生会员,主要研究领域为三维重建.  
E-mail: liutao@zjucadcg.cn