

中图法分类号: TP391.4 文献标志码: A 文章编号: 1006-8961(2011)04-0686-07

论文索引信息: 张淑军, 吴威. 体素可视外壳并行优化建模方法 [J]. 中国图象图形学报, 2011, 16(4): 686-692

体素可视外壳并行优化建模方法

张淑军^{1), 2)}, 吴威²⁾

¹⁾ (青岛科技大学信息科学技术学院, 青岛 266061)

²⁾ (北京航空航天大学 虚拟现实技术与系统国家重点实验室, 北京 100191)

摘要: 基于体的可视外壳重建不需要复杂的几何计算, 易于实现, 但其精度和效率仍不理想。提出一种基于统一计算架构(CUDA)和行进立方体(MC)的体素可视外壳并行优化建模方法, 将体素相交判断和等值面抽取过程并行分解, 利用 CUDA 的内核多线程机制加速建模; 在 MC 网格化阶段, 提取准确交点并消除冗余边计算; 利用像素着色器实现混合权重的纹理映射, 提高模型精度。在多摄像机采集装置 DreamWorld 上的实验结果表明, 本文方法能够对多个动态物体进行实时准确的 3 维建模。

关键词: 可视外壳; 实时建模; 统一计算架构(CUDA); 行进立方体; 体素

A parallel and optimized volumetric visual hull reconstruction method

Zhang Shujun^{1), 2)}, Wu Wei²⁾

¹⁾ (College of Information Science & Technology, Qingdao University of Science and Technology, Qingdao 266061 China)

²⁾ (State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191 China)

Abstract: Volumetric visual hull reconstruction is easy to be implemented without complicated geometrical computation, but its accuracy and efficiency are still unsatisfactory. A parallel optimized visual hull modeling method is proposed based on CUDA and marching cubes. It parallelizes the voxel intersection judgment procedure and the isosurface extraction procedure, and implements both of them using multiple threads of CUDA to accelerate the reconstruction. During marching cubes process, exact intersection points are calculated and redundant computation of edges is eliminated. The pixel shader is used to carry out the hybrid weight texture mapping so as to enhance the quality of the reconstructed mesh model. Experimental results on the multi-camera platform DreamWorld have proved that this method can reconstruct multiple dynamic objects accurately in real time.

Keywords: visual hull; real time modeling; computer unified device architecture (CUDA); marching cubes; voxel

0 引言

基于视频/图像的 3 维建模 (IBM) 是计算机视觉、计算机图形学、虚拟现实等多个领域的研究热点^[1], 其目的是通过目标物体的多视点图像来恢复

物体的 3 维几何与纹理信息。目前常见的 IBM 方法有基于立体视觉的方法、运动恢复结构的方法、基于光流的方法、基于侧影轮廓的方法等, 其中基于侧影轮廓的方法 (SFS) 是利用多相机拍摄物体的光锥相交的原理来恢复物体的形状信息, 由于其输出为物体的可视外壳 (visual hull), 因此也称为可视外壳

收稿日期: 2010-09-03; 修回日期: 2010-10-22

基金项目: 国家自然科学基金项目 (60903064, 61040047); 国家重点基础研究发展计划 (973) 项目 (2009CB320805)。

第一作者简介: 张淑军 (1980—), 女, 讲师。2007 年于中国海洋大学获计算机应用技术专业博士学位, 主要研究方向为图形图像处理。E-mail: zhangsj@vrlab.buaa.edu.cn。

方法。该方法是目前研究较为广泛的一种建模方法。

可视外壳方法总体上分为基于面的方法和基于体的方法。基于面的可视外壳方法是直接对可视锥体求交^[2],可直接得到物体的3维网格,但这种方法需要复杂的几何计算,存在数值不稳定的问题^[3],特别是当对具有复杂拓扑结构的物体建模时。基于体的可视外壳方法能够重建具有复杂拓扑结构的物体,鲁棒性较好^[4-5]。但由于其结果是体素描述,需要辅以网格化算法才能得到三角化的模型,行进立方体(MC)算法是比较常用的一种网格化算法,可以在表面立方体中生成三角面片,以三角面片来拟合真实的物体表面。传统的MC算法存在三角形交点不准确、二义性等问题,若直接用于可视外壳网格化,效果较差。Raman^[6]提出了扩展的MC查找表及SnapMC算法,将每个立方体顶点的二值状态扩展为正、负和零3个状态,解决了传统的MC算法易出现退化的零面积的面片问题,但是该算法会改变等值面的拓扑,也会产生非流形的等值面。Liang^[7]针对传统MC算法各体素边的交点缺乏插值信息的问题,提出了一种具有准确交点的MC算法,但其生成的网格可能存在空洞,且算法效率较低。由于MC算法是逐体素计算,各体素的数据独立性也使其易于通过GPU并行实现,因此也有学者在研究基于GPU并行的MC算法^[8-9],以提高算法效率。

可视外壳建模在虚实混合环境、远程沉浸、远程交流等领域中有良好的应用前景,但这些应用同时也对建模算法提出了更高的要求,实时性是首先要保证的。只有达到实时建模,才能支持虚实交互、异地沉浸等应用。目前提高建模算法效率的主要思路是通过计算机集群来实现,通过多机、多GPU进行大规模并行计算,降低计算量,提高计算效率。Franco提出一种准确的多面体可视外壳算法EPVH^[10],并提出了EPVH的并行版本以实现实时建模,该方法依赖于3个步骤的流水线,将每个步骤做并行任务分解,在计算机集群上实现^[11-12]。集群的实现手段虽然有效,但导致系统规模过大,不易移植。

以游戏加速和图形处理为初衷设计的图形硬件处理器(GPU)以超出摩尔定律3倍的速度发展,为开发实现各种图形和仿真应用提供了前所未有的新手段,同时,GPU超越CPU的浮点计算能力也使得基于GPU进行各种算法设计和加速成为可能。

CUDA(compute unified device architecture)是显卡厂商NVIDIA推出的运算平台,作为一个新的基础架构,它可以用GPU解决商业、工业以及科学方面的复杂计算问题,尤其适合于大规模的并行数据计算^[13-15]。目前,将GPU加速用于3维建模算法的尝试初露端倪,Lee等人^[16]使用GPU加速解决了可视外壳渲染的锯齿问题,但无法得到物体的显式3维模型。Yous等人^[17]利用GPU进行视锥体求交计算,通过深度优先遍历对每个摄像机分别计算可见边,并使用计算机集群使基于面的可视外壳建模达到实时效果。Ladikos等人^[18]提出了4种基于体的可视外壳算法并初步进行了CUDA并行,实现了较好的建模速度。但是,该文献也是基于4个PC的集群,并且速度测试结果只停留在体素描述层,没有进行网格化与纹理赋值。

本文以实时准确的3维重建为目标,提出一种体素可视外壳并行优化建模方法POVH(parallel optimized visual hull),将体素相交判断和等值面抽取过程并行分解并利用CUDA的内核多线程实现,有效提高建模速度;对MC算法进行准确交点计算和边冗余消除,提高建模精度,并通过混合权重的纹理映射,实现基于多摄像机的无标识物体实时采集、建模和绘制系统,可用于虚实交互、远程沉浸等多个领域。

1 可视外壳优化建模算法思想

可视外壳优化建模算法POVH,其优化体现在3个方面:1)CUDA并行加速:利用CUDA对体素和侧影轮廓图的求交计算并行化分解,通过CUDA的内核多线程机制实现;2)MC网格化优化:对得到的体素描述的模型通过MC算法进行表面等值面抽取,并通过计算准确的立方体边交点以及消除冗余边计算来优化MC算法,得到3维网格模型;3)纹理优化:利用图形硬件的像素着色器对网格模型进行混合权重的纹理映射,在保证模型精度的同时实现摄像机采集帧速的1倍实时建模。

为描述可视外壳建模原理,给出如下符号定义:设系统用 N 个同帧速的相机进行多视点视频采集,在第 t 帧,摄像机 C_j 捕获的图像记为 $I_j, j = 1, 2, \dots, N, I_j$ 经前景提取后,得到其侧影轮廓 S_j, S_j 所在视锥体记为 VC_j ,则物体在第 t 帧的可视外壳即为 $VH = \{\cap VC_j, j = 1, 2, \dots, N\}$ 。如图1所示。

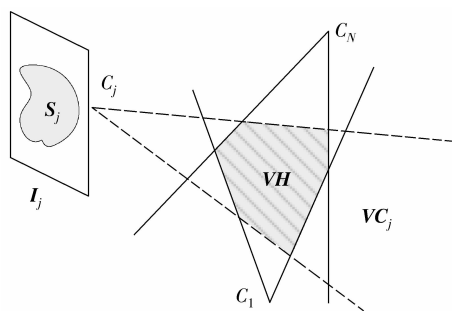


图 1 可视外壳方法建模原理

Fig. 1 Principle of visual hull modeling

图 1 中,虚线所示为相机 C_j 对应的视锥体 VC_j , 阴影部分为多个视锥体的交集,即可视外壳。算法利用从 3 维向 2 维投影的正向处理思想,求解 VH 空间。

由于摄像机是连续采集,每秒钟会有多帧图像数据被捕获,对动态物体而言,建模是对每帧数据进行的。因此,为提高在线建模的速度,首先进行预处理,为逐帧建模作数据准备。预处理主要包括生成体素查找表与体素投影查找表两方面。

1) 生成体素查找表 将物体所在空间沿 X, Y, Z 方向进行立方体剖分,如每边剖分 n 份,则空间剖分的分辨率为 n^3 ,共生成 n^3 个体素(立方体)数据。

2) 生成体素投影查找表 对所有成像平面,计算每个体素在其上的投影,并保存到查找表中。由于体素是立方体,其在图像平面上的投影是一个六边形。

预处理之后,即开始接收每帧获取的多视点图像数据,并进行在线的 3 维建模。建模过程分为基于 CUDA 的体素重建、体素模型网格化和纹理映射 3 个部分。

2 可视外壳优化建模算法

2.1 基于 CUDA 的体素重建

由于可视外壳是视锥体的最大交集,在初始的空间立方体基础上,逐渐削掉那些不属于物体的体素,即可得到构成物体的体素,这种思想类似于雕塑的创作过程。因此,需要对每个体素进行其投影与所有侧影轮廓图的相交判断,得到体素的占据状态。体素状态的判断准则为如果体素投影完全落在某个前景轮廓之外,则该体素标志为 White(不占据);如果体素投影完全落在所有侧影轮廓内,则该体素标

志为 Black(完全占据);否则,标志为 Gray(部分占据)。由于体素重建是对所有体素与所有的侧影轮廓图进行相交判断,此过程具有指令密集和数据独立性,若在 CPU 中循环执行,将耗费大量的计算时间;而这种数据独立、指令单一的特点非常适合 GPU 通用计算,因此,本文将此计算分布到 CUDA 的内核多线程机制并行实现,以提高重建效率,并行化步骤如下:

1) 对每个视点采集到的图像 I_j 进行二值化;

2) 将查找表中的数据以及每个相机的图像数据(原始图像和二值化图像)复制到 GPU 的全局显存;

3) 对每幅图像,进行体素投影与其侧影轮廓图的相交判断。利用 CUDA 的内核与线程组实现,每个内核包含一组线程,每个线程负责处理一部分体素。

所有体素判断完毕之后,此时,标志为 Black 和 Gray 的体素就构成了物体的可视外壳,而标志为 Gray 的体素位于可视外壳的边界。由于只有外围或者边缘的体素才对形状有贡献,因此将得到的标志为 Black 的体素,即在可视外壳内部的体素去除,减少网格化阶段待处理的体素数目。

2.2 体素模型网格化

由于体素重建之后的结果以立方体表示,原物体的轮廓信息无法被精确地反映出来。要得到准确的 3 维网格,还需要在模型表面应用一次网格化处理。MC 算法是被广泛应用的表面网格构造算法,其基本思想是逐个处理数据场中的立方体(体素),分类出与等值面相交的立方体,采用插值计算出等值面与立方体边的交点。根据立方体每一顶点与等值面的相对位置,将等值面与立方体边的交点按一定方式连接生成等值面,作为等值面在该立方体内的一个逼近表示。因此,MC 算法中每一单元内等值面抽取的两个主要计算是:1) 体素中由三角面片逼近的等值面计算;2) 三角面片法向量计算。

MC 算法的基本假设是等值面沿着立方体的边数据场呈连续线性变化,即如果一条边的两个顶点分别投影于二值化图像的不同值域,则这条边与等值面必有也仅有一个交点。确定立方体体素中等值面的分布是该算法的基础。

首先对立方体的 8 个顶点进行分类,以判定其顶点是位于等值面之外,还是位于等值面之内;再根据 8 个顶点的状态,确定等值面的剖分模式;进而计

算三角面片的顶点位置。本文对文献[7]提出的MC算法进行了改进,通过CUDA实现并行网格化,同时对其中涉及的边计算进行简化,能够快速精确地计算出三角面片顶点的位置,得到优化的网格模型。利用CUDA和OpenGL的互操作性,注册缓存对象(vertex buffer object)并映射,通过CUDA的内核计算得到相交点的坐标和纹理坐标信息,并通过缓存对象传给OpenGL。其中,计算三角面片顶点的步骤如下:

1) 将三角形顶点所在的立方体边 E 的两顶点 v_1, v_2 投影到各相机成像平面中,记为 w_1, w_2 (为便于后续描述,假设顶点 v_1 在等值面内);

2) 对每个侧影轮廓图:利用二分法,在投影线段 w_1w_2 上线性遍历光栅化后的各点,直到找到该侧影轮廓与 E 投影的交点 $S_j(j=1,2,\dots,N)$;

3) 利用各相机的投影矩阵,将交点 S_j 反投影到3维,得到其对应的空间点 λ_j ;

4) 比较各个侧影轮廓图计算得到的交点坐标 $\lambda_j(j=1,2,\dots,N)$,选取距离等值面内的边顶点即 v_1 最近的那个交点作为最终该边 E 与可视外壳的交点。

通常,在MC算法的执行过程中,需要对立方体进行逐层扫描,对每层、每个立方体的边逐条计算与等值面的交点,得到每个立方体内的三角面片,所有立方体的三角面片相连,得到最终的3维网格。这导致了立方体共享边的重复计算,据实验统计,平均每条边要计算4次,最少也要计算2次交点。因此,为提高MC算法的效率,对其进行冗余边计算的消除,在保持网格质量的前提下,有效减少计算量。具体实现是通过设置一个边表,来存储已经处理过的边序号。每当有一条新的边需要计算时,就检查该边表,如果该边已经被处理,就直接跳过;否则,对该边进行交点计算。

2.3 混合权重的纹理映射

对于生成的3维网格模型进行逼真的纹理映射是提高模型视觉效果的关键。视点无关的纹理映射仅使用模型网格表面法线进行纹理的选择,可以得到独立的模型纹理,但容易出现纹理抖动和锯齿,效果较差。视点相关的纹理映射结合了网格法线及当前观察视点的信息进行纹理选择,效果更加真实自然,但若对每个三角面片只选择与当前视点最接近的相机拍摄的图像进行纹理映射,则在比较靠近相机方向的视点处效果好,而在与相机方向差别较大

的视点处效果差。为克服以上问题,提出一种混合权重的视点相关纹理映射方法,综合考虑三角面片法向、当前视点与各相机光轴方向,并利用像素着色器进行纹理渲染,从而生成具有较高真实感的纹理。

具体算法为遍历网格化之后物体模型上的每个三角形面片,计算出所有可以观察到该面片的相机视点,再通过混合权重公式进行纹理映射,在像素着色器中渲染。

2.3.1 相机可见性判断

对于某个相机视点 C_i ,影响面片可见性的因素有两个:面片的朝向以及原始物体的自遮挡。对于面片的朝向,可通过面片自身法线与相机光轴向量的关系进行判断。设其夹角为 θ_i ,则有:当 $-\frac{\pi}{2} \leq \theta_i \leq \frac{\pi}{2}$ 时,面片对相机可见;否则,面片对相机不可见。

对于自遮挡,需要在各相机视点的观察窗口中生成诸像素的 Z 缓冲图,缓冲图记录投影到该像素的空间顶点的最小深度值,记为 Z_{\min} ,如果三角面片顶点的深度值大于 Z_{\min} ,则表示该面片由于模型本身的自遮挡不可见。

通过面片的法向与物体自遮挡的双重判断,得到对于当前面片可见的相机集合,记为 $\{C_i\}(i=1,2,\dots,K)$ 。此即当前面片的纹理源。

2.3.2 基于像素着色器的混合权重纹理映射

可见相机集合 $\{C_i\}$ 中每个相机的光轴方向与当前观察视线都有一个夹角 a_i ,通过下式求得各相机图像对当前三角面片的权重

$$w_i = \cos a_i / \sum_{i=1}^K \cos a_i \quad (1)$$

计算出权重之后,便可根据此值将三角面片的每个纹理源进行纹理混合,并通过法向量获得纹理坐标,进行纹理映射。将顶点数据和纹理数据都传递到像素着色器中,进行逐像素的渲染。

3 实验与结果分析

为验证本文提出的建模方法,设计并建立了基于多摄像机的无标识物体实时采集装置DreamWorld,采集空间分别为 $1\text{ m} \times 1\text{ m} \times 1\text{ m}$ 及 $3\text{ m} \times 3\text{ m} \times 3\text{ m}$,可实现动态物体及多人数据的数据采集、建模与交互,如图2所示。

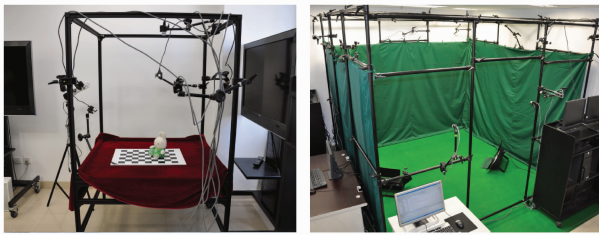


图 2 多摄像机采集平台 DreamWorld

Fig. 2 Multi-camera acquisition platform-DreamWorld

采用 5 个摄像机与客户端进行图像采集和预处理、1 台图形工作站进行建模,系统配置及软件环境如表 1 所示。

表 1 实验环境

Tab. 1 Experiment environment

类别	描述
网络环境	千兆以太网
主机环境	客户端: Intel Core 2 Duo E6400 3.00 GHz, 2GB DDR2-667 memory 服务器端: Intel Xeon X5450 3.00 12MB, 16GB RAM, NVIDIA Quadro FX5600 1.5GB
摄像机	FL2G-13S2C-C, 分辨率 640 × 512, 15 ~ 30 帧/s, 镜头 FV0420
软件环境	VS2005 (C + +), OpenGL 1.0, Glut 3.7.0

在小型 DreamWorld 平台上进行真实物体的视频采集与重建,实验场景与重建结果如图 3 所示。

图 3 中,分别给出了不同视点下的视频图像帧、对应时刻重建出的网格线模型及实体模型。可以看出,我们的重建方法和纹理映射达到了较好的精度,并能够很好地支持多个物体及动态物体(手一直在运动)的建模。

选取玩具兔 Miffy 和真实手 Hand 作为目标物体,将本文方法与串行八叉树建模方法相比较,在不同分辨率下的性能指标如表 2 所示。

从表 2 中可见,基于 CUDA 的优化建模方法比串行方法的效率平均提高了 1 倍以上,能够实现摄像机采集帧速的 1 倍实时建模(15 帧/s 以上);随着建模分辨率的提高,加速比也随之增大,说明 CUDA 的加速能力在高精度的建模时更为明显。

在实时建模的基础上,实现了虚实交互,真实手与虚拟水面实时交互的视频截图与重建结果如图 4 所示。

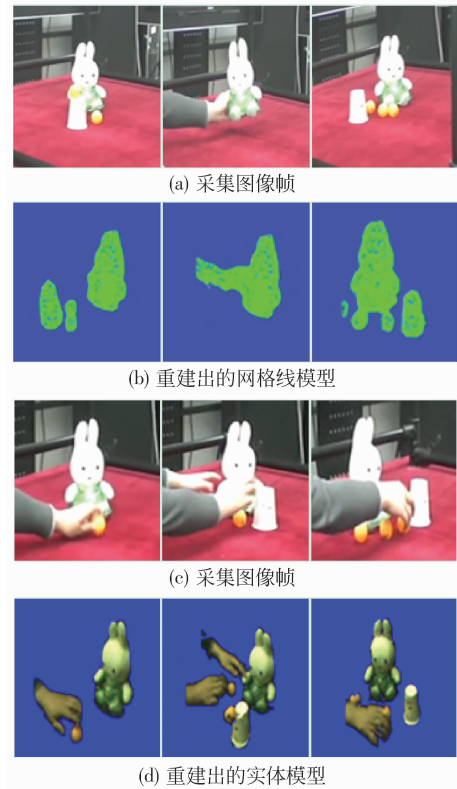


图 3 采集图像帧、重建出的网格线模型及实体模型
Fig. 3 Acquired image frames, reconstructed wireframe models and solid models

表 2 本文方法与串行方法比较

Tab. 2 Comparison between the proposed method and the serial method

指标	Hand		Miffy	
	32 ³	64 ³	32 ³	64 ³
串行方法/ms	61.34	179.20	71.78	209.42
POVH/ms	28.05	68.32	33.45	68.97
POVH 帧速/(帧/s)	36	15	30	15
加速比/%	54.27	61.88	53.40	67.07
面片数	1 021	3 983	1 140	5 464

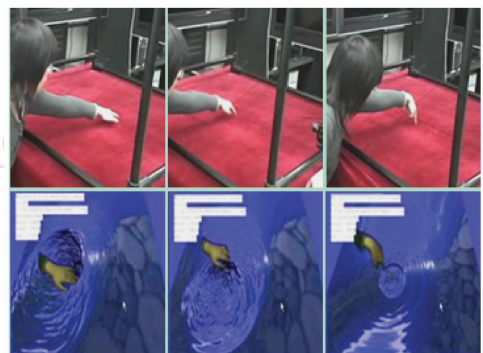


图 4 真实手与虚拟水面交互

Fig. 4 Real hand interaction with the virtual water

图4中,用户在工作空间内动作,手就会被实时重建到虚拟环境中,并与虚拟水面产生交互,水面出现涟漪、尾迹及反射与折射效果,由此实现了一个沉浸式混合现实环境。

在 $3\text{ m} \times 3\text{ m} \times 3\text{ m}$ 的 DreamWorld 装置中,基于以上建模方法,实现了多人互动虚实共享的混合现实应用,如图5所示。



图5 多人互动实时建模结果

Fig. 5 Reconstruction results of multiple users' interaction

图5分别展示了在多人数据采集装置中的视频图像帧与重建出的人体模型与虚拟海洋交互的场景。从图5可见,本文所提出的方法能够支持动态物体及多人体的实时3维建模,在满足较好逼真度的前提下实现虚实交互、远程沉浸等实时应用。

4 结 论

针对虚拟现实、混合现实中广泛的实时应用问题,提出并实现了一种体素可视外壳并行优化建模方法,将体素重建过程和等值面抽取过程并行分解,利用 CUDA 的通用计算能力,提高建模效率;将 MC 算法与可视外壳本身的特点相结合,提取准确交点,消除冗余边计算,提高网格化的速度和准确度;通过权重混合方法在像素着色层次实现较高准确度的纹理映射。在多摄像机采集系统 DreamWorld 上的实验结果表明,该方法能够有效地实现动态物体的实时准确建模。

参考文献 (References)

- [1] Teresa C S Azevedo, João Manuel R S Tavares, Mário A P Vaz. 3D object reconstruction from uncalibrated images using an off-the-shelf camera [J]. Advances in Computational Vision and Medical Image Processing Methods and Applications, Springer Netherlands, 2009, 13(2) : 117-136.
- [2] Matusik W, Buehler C, McMillan L. Polyhedral visual hulls for real-time rendering [C] // Proceedings of Eurographics Workshop on Rendering. Berlin, Germany: Springer, 2001:115-126.
- [3] Lazebnik S, Boyer E, Ponce J. On how to compute exact visual hulls of object bounded by smooth surfaces [C] // Proceedings of CVPR. New York, USA: IEEE Press, 2001, 1 : 156-161
- [4] Hasenfratz J M, Lapierre M, Gascuel J D, et al. Real-time capture, reconstruction and insertion into virtual world of human actors [C] // Vision, Video and Graphics Conference. Berlin, Germany: Springer, 2003 : 49-56.
- [5] Slabaugh G, Culbertson B, Malzbender T, et al. A survey of methods for volumetric scene reconstruction from photographs [C] // Proceedings of International Workshop on Volume Graphics. New York, USA: IEEE Press, 2001 : 81-100.
- [6] Raman S, Wenger R. Quality isosurface mesh generation using an extended marching cubes lookup table [J]. Computer Graphics Forum, 2008, 27(3) : 791-798.
- [7] Liang C, Wong K-Y K. Exact visual hull from marching cubes [C] // Proceedings of the third International Conference on Computer Vision Theory and Applications. Berlin, Germany: Springer, 2008, 2 : 597-604.
- [8] Xie Y, Heng P A, Wang G, et al. GPU-friendly marching cubes for visualizing translucent isosurfaces [C]. Proceedings of Medicine Meets Virtual Reality 2007 (MMVR). Long Beach, California, USA: IOS Press, 2007 : 500-503.
- [9] Johansson G, Carr H. Accelerating marching cubes with graphics hardware [C] // Proceedings of the Conference of the Center for Advanced Studies on Collaborative Research Table of Contents (CASCON 06'). New York, USA: ACM Press, 2006 : 378-383.
- [10] Franco J S, Boyer E. Exact polyhedral visual hulls [C] // Proceedings of the Fourteenth British Machine Vision Conference (BMVC). Berlin, Germany: Springer, 2003 : 329-338.
- [11] Franco J S, Menier C, Boyer E, et al. A distributed approach for real-time 3D modeling [C] // Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '04). New York, USA: IEEE Press, 2004 : 31-31.
- [12] Menier C, Allard J, Franco J S, et al. Markerless real time 3D modeling for virtual reality [C] // Proceedings of the 8th Immersive Projection Technology Workshop. Ames, Iowa: John Wiley & Sons Publisher, 2004 : 1-8.

- [13] Luebke D. CUDA: scalable parallel programming for high-performance scientific computing [C]. Proceedings of the 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI). New York, USA: IEEE Press, 2008: 836-838.
- [14] Garland M, Grand S L, Nickolls J, et al. Parallel computing experiences with CUDA [J]. IEEE Micro, 2008, 28(4): 13-27.
- [15] Che S, Boyer M, Meng J, et al. A performance study of general-purpose applications on graphics processors using CUDA [J]. Journal of Parallel and Distributed Computing, 2008, 68(10): 1370-1380.
- [16] Lee C H, Cho J H, Oh K S. Hardware-accelerated jaggy-free visual hulls with silhouette maps [C]// Proceedings of the ACM Symposium on Virtual Reality Software and Technology, New York, USA: ACM Press, 2006: 87-90.
- [17] Yous S, Laga H, Kidode M, et al. GPU-based shape from silhouettes [C]// Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast. New York, USA: ACM Press, 2007: 71-77.
- [18] Ladikos A, Benhimane S, Navab N. Efficient visual hull computation for real-time 3D reconstruction using CUDA [C]// Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. New York, USA: IEEE Press, 2008: 1-8.