

## 改进的空间体素融合方法及其在线重建\*

王剑飞<sup>1</sup>, 林金花<sup>1,2,3†</sup>, 王璐<sup>1</sup>

(1. 长春工业大学 应用技术学院, 吉林 长春 130012; 2. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033; 3. 中国科学院大学 机械电子工程系, 吉林 长春 130033)

**摘要:**目前,随着深度传感器的更新发展,使得三维重建的在线方法得以实现.在线重建的难点在于如何能够在保证较好的重建效果和重建规模的前提下,实时地将多个动态重叠深度图转化为一个三维模型.针对这一问题,本文算法基于空间索引策略构建了一种存取高效的体素数据结构,改进了传统体积融合方法中的规则分层网格数据结构,对隐含表面数据进行实时存取与更新的同时,实现大尺度的在线三维重建.这些表面数据仅在观测时存储,并且高效地流入和流出索引表,实现了传感器运动过程中的在线场景重建.算法通过在GPU(Graphics Processing Pipeline)上完成深度图预处理,摄像姿态估计,深度图融合和表面绘制几个核心阶段,实现对细粒度的细节和大尺度的环境进行交互式实时重建.通过实验表明,改进后的重建系统能够在保证重建质量与重建性能的同时,平均重建时间为21.6 ms,ICP姿态估计15次,用时8.0 ms,平均帧绘制速率提高13.3%,实现了对大尺度场景的实时三维重建.

**关键词:**在线; 空间索引; 数据结构; GPU; 实时重建

中图分类号:TP391.41

文献标志码:A

## Improved Spatial Voxel Fusion Method and Its Online Reconstruction

WANG Jianfei<sup>1</sup>, LIN Jinhua<sup>1,2,3†</sup>, WANG Lu<sup>1</sup>

(1. Computer Application Technology, Changchun University of Technology, Changchun 130012, China; 2. Mechanical and Electrical Engineering, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; 3. Department of Mechanical and Electrical Engineering, University of Chinese Academy of Sciences, Changchun 130033, China)

**Abstract:** At present, with the development of the depth sensor, the on-line method of 3D reconstruction is realized. The difficulty of the online reconstruction is how to convert multiple dynamic overlapping depth maps into a three-dimensional model in real time under the premise of better reconstruction effect and reconstruction scale. To solve this problem, this paper constructed an efficient spatial voxel data structure based on the spatial index strategy, improving the hierarchical grid data structure of the traditional volume fusion method and realizing the real-time access to the hidden surface data. These surface data are only stored during the observation to achieve the sensor movement in the process of online scene re-

\* 收稿日期:2017-04-11

基金项目:国家高技术研究发展计划(863)资助项目(2014AA7031010B), National High-tech R&D 863 Program(2014AA7031010B); 吉林省“十三五”科学技术研究资助项目(2016345), Science and Technology Project of the Thirteenth Five-Year Plan of Jilin Province(2016345)

作者简介:王剑飞(1970—),男,吉林长春人,长春工业大学教授,博士生导师

† 通讯联系人, E-mail:ljh3832@163.com

construction. The algorithm performs interactive real-time reconstruction of fine-grained detail and large-scale environment by completing several core stages, such as depth map preprocessing, camera pose estimation, depth map fusion and surface rendering, on the GPU (Graphics Processing Pipeline). The experimental results show that the average reconstruction time is 21.6 ms, the ICP pose estimation is 15 times, and the average frame rendering rate is increased by 13.3% when the reconstruction time is 8.0 ms and the reconstruction time is 21.6 ms, realizing the real-time 3D reconstruction of large scale scenes.

**Key words:** online; spatial index; data structure; GPU; real-time reconstruction

三维重建是计算机视觉与图形学的重要研究领域,随着深度摄像机的广泛使用,使得三维重建获得了新的发展势头.由于这些设备能够实时输出带有噪声的深度图,使得最近的研究重点是如何使用这种深度摄像机获得在线表面重建.在线表面重建在计算机交互式应用领域表现良好,例如,增强现实(AR),其中真实世界的几何信息可以与3D(three dimension)图形融合并实时渲染;自主指导机器人重建及其对周围环境的快速响应;或者在3D打印过程中实现向用户提供即时反馈<sup>[1-2]</sup>.

传统重建方法<sup>[3-4]</sup>大多基于一组完整的三维点坐标实现重建,而在线重建是将多个重叠深度图增量融合成一个不断精进的三维表示.为了达到较好的实时重建性能,重建质量与重建规模不受影响,出现了许多先进的基于GPU硬件加速的在线方法,这些方法通常采用不同类型的底层数据结构,在重建速度、规模和质量方面都具有不同程度的相互抵消.

参数重建方法<sup>[5-6]</sup>简单地平均重叠样本,并通过表面对拓扑的简单假设来实现点和局部多边形的连接.同时出现的网格拉链方法<sup>[7]</sup>为每个表面区域选择一个深度图,并去除重叠区域中的冗余三角形进而实现网格缝合.这些方法通过点的局部平均化来处理噪声,但是在异常值和高曲率的区域表现不佳.因此出现了多种非直连多边形网格的三维重建方法.

基于点的重建方法<sup>[8]</sup>通过合并重叠点来实现重建,从而避免推断连通性.由于大多数深度传感器的输出是三维点样本,使得基于点的三维重建更加容易,例如针对单个小对象的扫描系统<sup>[9]</sup>.由于重建规模较小,使得小规模高质量的三维重建得以实现.较大尺度场景的三维重建通常需要降低重建速度或质量来实现<sup>[10]</sup>.这种方法使用简单的非结构化表示,可以紧密地映射到深度传感器,但是缺乏直接重建表面的能力.对小物体的高质量在线重建已经得以

实现<sup>[11]</sup>,但是大尺度重建需要以重建质量或速度为代价<sup>[12]</sup>.

基于高度图的表示方法<sup>[13]</sup>支持连接表面数据的高效压缩,使用更紧凑的2.5维度的连续表面表示来实现三维重建,这种方法尤其适用于大型建筑物的重建<sup>[14]</sup>,并且可以有效扩展到更大尺度的场景,但是无法实现对复杂模型的三维重建.

基于体积数据结构的重建方法(VRM, Volume Reconstruction Method)对三角测量的传感器噪声特性进行有效估计,支持增量更新,不对表面数据进行拓扑压缩,对于给定的样本进行累加平均,可以得到较好的重建效果和重建速度<sup>[15-17]</sup>.然而,这种方法依赖于存取低效的规则体素网格,反而限制了重建规模.

为解决体积重建方法的空间限制问题,出现了多种重建方法.Keller等人<sup>[18]</sup>使用基于点的表示方法来加强体积融合的质量,虽然摆脱了空间数据结构的限制,但重建质量随之降低<sup>[19]</sup>.

同时出现了移动体积方法<sup>[20]</sup>,扩展了基于深度传感器融合的GPU渲染管线<sup>[21]</sup>.这种方法在摄像机运动过程中使用GPU管线并行处理体素流,释放了用于存储新近数据的空间.然而这些数据流是单向有损的,即当表面数据被压缩到网格时,无法从主机回流到GPU,并且算法仍然使用规则网格结构,这意味着数据流必须保持较小,才能确保精细的重建质量.这使得重建被限制在邻近几何体的场景重建,由深度传感器得到的全范围数据没有得到充分利用.

规则体素网格的限制使得许多研究人员进一步寻求更加有效的体积数据结构.例如,基于稀疏空间体素的体积表示方法<sup>[22]</sup>,用于大数据集的外核流架构<sup>[23]</sup>.这些分层数据结构被应用到三维重建的难点在于:在重建过程中需要对基础数据进行实时更新.例如,Zeng等人<sup>[24]</sup>实现了一个9到10级的八叉树结构,将传感器融合管线扩展到 $8\text{m} \times 8\text{m} \times 2\text{m}$ 的室

内空间. Pätzold 等人<sup>[25]</sup>进一步提出了一种基于 GPU 的八叉树体素网格表示方法,实现了对无网格外核体素的实时重建.然而,这些方法需要构建复杂的八叉树结构,计算复杂度随之增加,指针访问的额外开销较大<sup>[26-27]</sup>.

基于深度传感器融合的三维重建方法使用体积重建方法中的规则体素网格来重建表面,存储空间占用较大,并需要同时密集地表示空白的空间和表面,重建的难点是如何在不降低质量的情况下重建大尺度的场景.

针对分层数据结构在体积融合过程中存在的计算过度复杂的问题,本文提出了一种新的实时表面重建方法,支持大尺度高质量的三维重建.本文算法兼具体积方法的优点,但不依赖于受存储约束的体素网格,同时无需分层数据结构的计算开销.与传统分层数据结构不同,本文算法基于一种存取高效的空间索引策略,对隐含表面数据进行实时融合.表面数据被密集存储在观测单元格中,同时数据可以高效地流入或流出索引表,从而在传感器运动过程中进一步实现可伸缩重建.本文给出了一种基于空间索引策略的空间体素数据结构,通过对基础数据的不断更新,实现了对细粒度和大尺度场景的交互式重建.

## 1 相关概念与理论

体积重建方法(VRM, Volume Reconstruction Method)将深度图转换成带符号的距离场,通过叠加平均得到一个规则体素网格,并使用等值面多边形或光线投射来生成隐式函数的零级集合,从中提取最终的重建表面,实现对三维物体的重建.

体积重建方法在体数据结构中存储一个隐式带符号距离场(SDF, Signed Distance Field),这里假设输入一组深度图,且有一个规则的密集体素网格,深度传感器在网格的中心点处被初始化,并使用 ICP 变量<sup>[28]</sup>来估计传感器的刚性六自由度(6DoF, Six Degree-of-Freedom).网格中的每个体素都包含两个数值:带符号距离和权值.对于单个的深度图,传感器均匀地扫描物体表面,剔除掉视锥体以外的体素,将全部体素的中心点投影到深度图中,接着更新体数据结构中的 SDF 值,实现深度图到网格的数据融合.投影到同一像素点的所有体素成为深度样本的贡献分支.每个体素都存储了一个从体素中心到观测表面测量的带符号距离值,其中带符号距离

为正值,表示观测在前,反之为负,零值表示在观测表面的边界面.

为了降低计算成本,VRM 方法引入了截断式带符号距离场(TSDF, Truncated Signed Distance Field)概念,仅在被观测表面周围的区域中存储带符号距离.该区域的尺寸可以调节,并将传感器噪声近似表示成一个基于深度的高斯方差.只有该区域内的体素的 TSDF 值被加权平均后更新,从而获得一个表面估计.最后,作为每个深度样本的贡献分支,但在截断区域之外的体素被明确的标记为自由空间.

VRM 方法截断表面周围的 SDF,存储在规则体素网格中的大部分数据被标记为自由空间或不可见空间.因此问题的关键是:基于 TSDF 表示方法的稀疏特性建立一种存取高效的数据结构.

本文算法对传统的体积重建方法进行扩充,通过建立一种快速有效的空间查找数据结构,递增地将噪声深度图融合到指定存储单元,实现对隐含表面数据的存储、访问和更新,在保证重建质量与重建速度的同时,实现了对三维曲面的大尺度重建.

## 2 本文算法概述

本文算法的核心是建立一个基于空间索引策略<sup>[29]</sup>的空间体素数据结构,并在传感器运动过程中,使用 GPU 渲染管线实现对大尺度场景的实时重建.由于重建物体的几何特性是未知的并且连续变化的,本文提出的空间体素数据结构能够对 TSDF 进行动态存储与更新,可以高效处理索引表的冲突问题,进而实现可伸缩的体积重建过程.本文提出的空间体素数据结构的核心功能主要有以下几个方面:

1)在保持重建表面分辨率的情况下,对 TSDF 进行有效压缩,不需要建立复杂的分层空间数据结构.

2)将新生成的 TSDF 样本有效融合到索引表中,实现动态插入与更新,并将冲突降低到最小.

3)当删除体素块时,无需大量重组数据结构,降低了计算时间复杂度.

4)在 CPU 与 GPU 之间实现了轻量级的双向体素流,实现了传感器运动过程中的大范围重建.

5)使用标准光线投射和多边形操作来提取等值面,用于重建表面绘制和摄像机姿态估计.

本文算法的系统实现如图 1 所示,系统核心是

空间体素数据结构,用于存取体素块.系统使用索引表结构来存储被分配的体素块,每个体素由 TSDF、权值和颜色值组成.索引表结构采用非结构化的形式存储,即在空间上相邻的体素块,被存储在索引表中的不相邻位置.系统采用的索引函数,通过指定的世界坐标来高效查找体素块,并将冲突降到最低.本文系统的主要工作步骤详述如下:

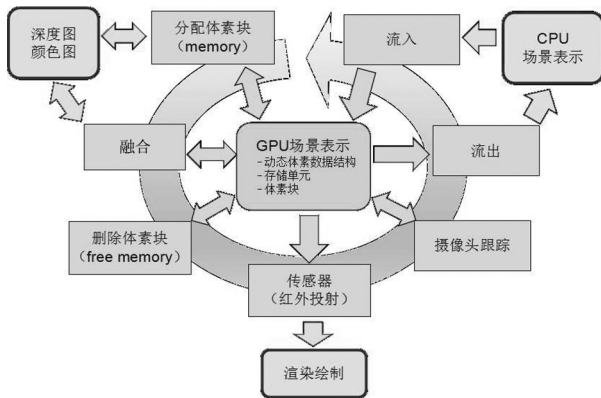


图 1 本文重建系统框图

Fig.1 Reconstruction procedure of our algorithm

第一步,基于输入的深度图,系统开始执行融合.分配新的体素块,并将体素块描述符插入到索引表中.这里仅分配被占用的体素而不考虑空的空间.扫描每个分配的体素块,更新块内体素的 TSDF,权值和颜色值,删除离等值面较远并且权值为零的体素块.同时释放该体素块所占用的存储单元,并从索引表中删除其对应的数据项.

第二步,从当前估计的摄像机姿态来投射隐含表面,用来提取等值面(包含颜色值).该提取的深度和颜色信息被用作摄像机姿态估计的输入,即给定一个输入深度图,提取投影点平面的 ICP 值来计算得出新的传感器 6DoF 值,实现对相机姿态的估计.对于小型场景,这种帧到模型的姿态估计减少了漂移问题.

第三步,在 CPU 和 GPU 之间执行双向体素流处理,当体素块的世界位置离开估计的相机视锥体时,索引表中的体素块被传送给 CPU,当区域重新出现在视锥体内时,之前流出的体素块被重新传送(流回)到 GPU 的空间体素数据结构中.

### 3 空间体素数据结构

本文采用索引表来对传感器取得的深度图进行

精进式的索引存储表示,并使用 GPU 渲染管线进行实时绘制.在世界坐标表示中,场景被理论上无限均匀的网格细分成体素块,每个体素块由更小的规则体素网格组成,本文算法将体素块分成  $8^3$  个体素,每个体素存储了 TSDF、颜色和权值,占用 8 字节存储单元.体素块仅围绕重建的表面几何进行分配(实现稀疏性),使用高效的 GPU 加速来处理体素块的分配与检索.索引表存储空间体素项,每个体素项包含所分配的体素块的指针,并使用整数世界坐标  $(x, y, z)$  从索引表中检索体素块.索引函数如公式(1)所示:

$$H(x, y, z) = (x \cdot p_1 \oplus y \cdot p_2 \oplus z \cdot p_3) \bmod n \tag{1}$$

式中:  $p_1, p_2$  和  $p_3$  是大素数;  $n$  是索引表长度.索引表数据项包含所对应的体素块指针、相关联的世界坐标值以及处理冲突的偏移指针.数据结构定义如图 2 所示.

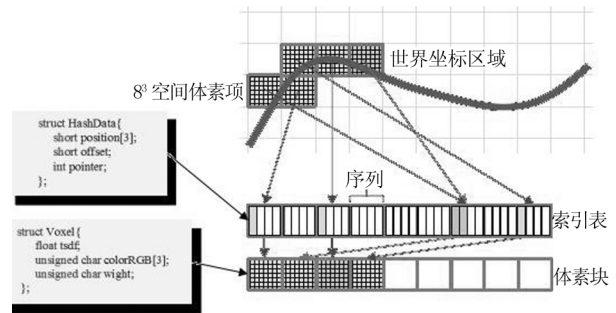


图 2 空间体素项的存储定义

Fig.2 Storage definition of space voxels

#### 3.1 体素项冲突处理

当多个体素块被映射到相同的索引单元,则出现冲突,如图 2 所示.本文将索引单元划分成由多个空间体素项构成的序列,每个序列分配唯一的索引地址,每个体素项对应世界坐标的唯一体素,即同一体素块内的体素被映射到同一个序列中.当发生冲突时,体素块的指针被存储到序列中的下一个未被占用的存储单元中.当需要查找世界坐标中的某个体素块时,首先根据坐标值计算索引函数,查找并遍历索引表中对应的序列,直到找到为止.

本文通过合理地选择索引表的长度和序列的大小,来避免序列溢出现象.但是,一旦发生溢出,本文算法通过在序列发生溢出的体素项处附加链表来实现链接.链接表的相对指针存储在空间体素项的偏移量单元,即最后存入序列的体素项的偏移单元存放的是下一个链接表在索引表中的偏移地址值,其

他体素项的偏移单元存放的是序列内下一个体素项偏移地址值. 序列的最后一个体素项存储单元设为空, 用来表示链表的头. 本文算法通过合理设定索引表和序列的大小来保持冲突数, 适用于大多数场景, 并将附加链表的数量降低最低, 以免影响重建性能.

### 3.2 空间体素存取操作

#### 3.2.1 体素项插入到索引表

首先, 计算体素块对应的索引函数值, 确定目标序列. 其次, 对序列进行遍历, 包括附加到序列的链表. 当找到与世界坐标位置相同的元素, 则返回一个索引, 否则, 继续寻找序列中的第一个空位置. 当序列中的位置可用时, 则插入新的体素项. 最后, 当序列已满时, 将该体素项附加到该序列的链表中, 如图 3 所示.

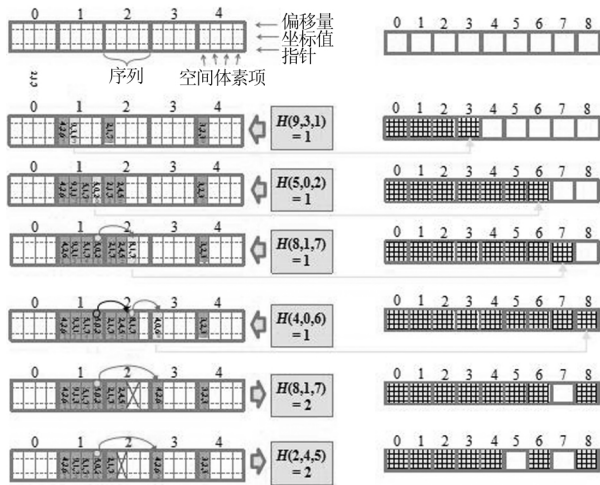


图 3 索引表的存取操作

Fig. 3 Access to the index table

当 GPU 并行地将体素项插入到序列中时, 为了避免抢占现象, 当某个体素项找到空位置时, 算法锁定该项所在的序列, 此时其它体素项无法写入. 当序列被锁定写入时, 对序列内的其它体素单元的分配迟缓, 直到开始处理下一帧的数据. 本文实践证明, 由于体积重建方法支持与顺序无关的更新, 些微的延迟不会引起重建质量的降低.

#### 3.2.2 从索引表读取体素项

为了查找体素项在索引表中的位置, 首先, 计算体素项的索引地址值; 其次, 根据索引值在序列中进行线性遍历, 当序列中找不到指定项时, 则继续查找序列最后一个体素项的偏移值, 顺序遍历与该序列关联的链表; 最后, 由于删除体素项会导致碎片的产生, 因此当空位置被找到之前, 遍历会循环进行.

#### 3.2.3 删除索引表中的体素项

删除操作与插入操作类似, 对于一个给定的世界坐标位置, 首先, 计算出索引地址值, 接着线性遍历该体素项对应的序列(包含链表遍历). 然后, 当查找到指定体素项, 并且该项无链接表时, 则直接删除该索引表项. 当该体素项是序列最后一个元素时, 并且存在一个非零的偏移量, 表示该体素项是一个链表的表头, 此时将偏移量指向的索引表项的地址复制到序列的最后一个元素中, 并从序列中删除该体素项. 最后, 由于要对链接表进行修改, 算法将该序列锁定并延缓链表的相关操作, 指定下一帧位置.

### 3.3 分配体素块存储单元

在融合下一个新体素块的 TSDF 值之前, 需要先对每个输入的深度样本区内的体素块分配存储单元, 这些体素块也在测量物体表面的截断区域内. 算法同时对深度样本进行处理, 插入空间体素项, 并分配存储单元给截断区域内的体素块. 采用深度方差来调整截断区域的大小, 用来补偿传感器远距离测量引起的不确定性.

对于每个输入的深度样本, 本文算法对与截断区域关联的射线进行实例化处理. 对于给定的体素分辨率和块大小, 使用 DDA 来确定与射线相交的所有体素块, 然后向索引表中插入一个新的空间体素项. 一般情况下, 每个深度样本是由整个平截头体所确定, 而不是单个射线, 再对与该平截头体相交的截断区域内的所有体素块分配各自的存储单元, 这样会导致性能的下降. 本文算法基于射线的近似方法实现了重建性能和精度之间的平衡. 基于重建的连续性, 传感器的帧速率和摄像头的移动变换三个因素, 本文算法在较大距离间隔外的体素块之间实现了无缝连接.

当索引表被成功插入一个空间体素项后, 本文算法在 GPU 上分配一部分堆存储单元来存储体素块. 堆是内存的线性数组, 在初始化时分配存储单元并确定大小. 它被划分为连续的区域, 用于映射到指定体素块的大小, 并且通过动态维护空闲块的列表来管理. 该列表是包含所有未分配块索引的线性缓冲区, 最后一个索引用于每次新块的分配. 当体素块被释放时, 其索引被附加到列表的末尾. 由于列表采用并行访问的方式, 需要通过使用元操作来增加或减少列表指针的同步运算.

## 4 体素块融合

融合是指对当前摄像截断范围内的已分配的体

素块的更新操作. 当可见表面的截断区域中的所有体素块被成功分配存储单元后, 索引表的大部分区域是空闲的, 大部分体素块在平截头体外. 由于仅选取当前摄像平截头体的块处理, 使得 TSDF 的融合过程非常高效.

#### 4.1 体素块的选取操作

为了选取用于融合的体素块, 算法首先并行访问索引表中所有体素项, 并将被占用的体素块(可见区域内的)的二进制标志存储在数组中. 然后使用并行预滤方法来扫描数组. 为了实现大尺度的扫描, 使用三级上下扫描. 最后将索引表压缩存储到另一个缓冲区中, 该缓冲区包含指向视锥体内的体素块对应的所有体素项, 如图 4 所示.

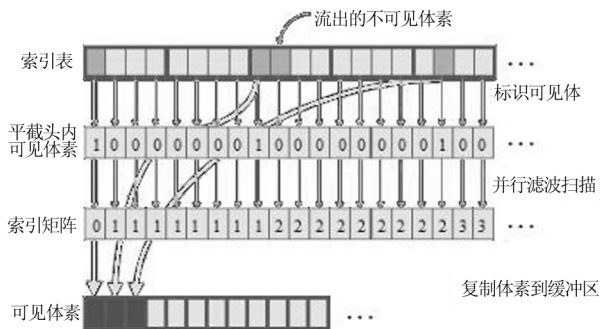


图 4 可见体素块融合过程  
Fig. 4 The voxel fusion procession

#### 4.2 隐含表面更新操作

更新操作是指对新生成的索引表中的体素项的 TSDF 值、权值和颜色进行重新计算. 对应每个项关联的块调用单个 GPGPU 内核, 即每个体素分配一个线程. 这意味着在 GPU 多个处理器上处理体素块, 从而最大化高速缓存的命中率. 实践证明, 这比分配单个线程来处理整个体素块更加有效.

更新体素块涉及与之关联的 TSDF、权值和颜色的重新计算. 深度值使用移动平均值进行整合, 再根据深度值设置融合权重, 以整合传感器的噪声特性, 即对较少噪声的近处深度测量分配较大的权重. 颜色也随之更新, 对最近的颜色样本分配较大的权重.

融合的一个重要部分是更新出现在当前平截头体内的所有体素块, 无论这些体素块是否会驻留在当前截断区域中. 由于表面移动或深度图中的异常值被分配, 使得体素块不再可见, 这些体素块会被同等地连续更新. 算法会在下一步对融合后的部分体素块进行删除处理.

#### 4.3 体素块的清除操作

清除操作是指删除由于噪声异常和移动表面所分配的体素块, 释放体素块存储单元. 对于索引表中每个体素项相关联的体素块, 算法计算出最小 TSDF 和最大权值. 当体素块的最大权值为零或最小 TSDF 大于指定阈值时, 则清除该体素块. 同时删除索引表中对应的体素项. 当索引表项被成功删除后, 再将体素块指针链接到堆, 实现对应体素块的存储单元的释放.

### 5 体素表面提取(传感器投射与跟踪)

本文算法通过光线投射来提取隐含存储的等值面. 首先, 光栅化处理当前平截头体中所有已分配的体素块的边界框, 计算每个射线的开始与结束点, 如图 5 所示. 同时, 遍历融合计算后的紧凑索引缓冲区, 并行的光栅化每个体素块, 生成用于最小和最大深度的两个 Z 缓冲区. 由于算法采用线性索引表数据结构, 可以快速并行访问所有分配的块, 在重建效率上明显优于传统的分层数据结构.

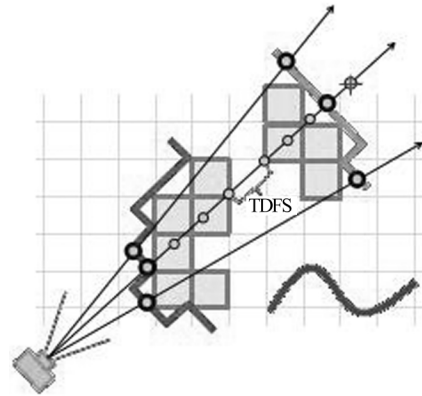


图 5 光线投射到体素块  
Fig. 5 The light is projected onto the voxel block

首先, 对于每个输出像素, 从相关联的最小到最大深度值的射线轨迹前进, 同时计算当前射线的相邻世界坐标位置的 TSDF 值. 此时, 未分配的体素块被认定为空白空间, 被占用的体素块则通过结合相邻的八个体素来进行三次线性内插计算. 对于体素块边界处的采用, 通过查找索引表中的相邻体素来处理, 而不是直接采用体素块, 即在重建过程中, 查找索引表而无论体素块是否在边界处. 由于算法基于 GPU 高速缓存实现体素的存取, 每个线程减少了寄存器个数, 提高了块采用效率.

其次,为了定位表面数据,比较当前块与上一个体素块的 TSDF 值的符号变化,忽略从负到正的零交叉,即剔除背面表面几何信息。

最后,一旦表面被光线投射提取,便可被遮挡渲染或用于相机姿态估计。为在深度融合之前估计新姿态,本文使用下一个输入帧以及射线投射深度图来估计姿态。姿态使用 ICP 的点平面变体和投影数据关联来实现估计。点平面能量函数在 GPU 上被线性化为  $6 \times 6$  的矩阵,通过在 GPU 上使用并行缩减和奇异值求解来实现。同时,在点平面误差度量中并入加权因子,提取和输入 RGB 颜色值来实现对体素块颜色数据的存取。

## 6 CPU 与 GPU 之间的体素流机制

空间体素数据结构支持高分辨率体素的建模,超出深度摄像机的分辨率范围。当表面数据远离平截头体时,GPU 的内存和性能直接影响重建速率。本文使用双向 GPU 体素流传输机制,来处理大尺度的重建范围。通过创建一个包含当前摄像机视平面和周围安全区域的球形活动区,基于空间体素数据结构的非结构化存储机制,使得流入和流出的体素块不需要重组索引表,适用于大尺度重建。对于标准深度传感器,深度范围可达 8 m,将球体的中心定位在离相机位置 4 m 处,并使用 8 m 半径测量,如图 6 所示,体素块的双向流动开始于姿态估计之后的每一帧。

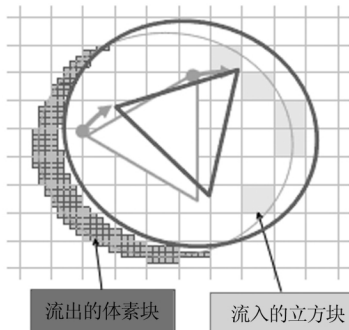


图 6 摄像头从左向右移动时的流操作  
Fig. 6 The camera moves from left to right

### 6.1 从 GPU 到 CPU 的体素流

为了将体素块流出活动区,首先,并行地遍历索引表,标记流出活动区的体素块,删除相应的体素

项,并将这些体素附加到中间缓冲区。然后对于所有体素项对应的体素块,复制其到中间缓冲区后,清除原始体素块,并将释放出的存储单元附加到堆,便于重新使用。最后,这些中间缓冲区被复制回 CPU。在 CPU 中,体素数据不再被组织成索引表,而是将世界空间统一细分为块(块的大小设置为  $1 \text{ m}^3$ ),这里统称为立方块(为区别体素块)。使用链接表将体素块附加到这些块,对应每个体素块,存储对应的空间体素项描述符合体素数据。

### 6.2 从 CPU 到 GPU 的体素流

对应 CPU 到 GPU 的流,首先识别由于摄像机移回到先前重建区域,而重新出现在球体区域中的体素块。与像素块的 GPU 到 CPU 的流动相反,CPU 到 GPU 的流是以每个立方块为基本操作单元。当块被识别为流操作,则该立方块中的所有体素块将被流传送到 GPU。由于 GPU 的带宽和剔除视锥体外的体素块的并行处理速度,使得大尺寸重建得以实现。

其次,从 CPU 到 GPU 的流是交错进行的,选择标记为最靠近相机平截头体中心的块,每帧一个立方块。通过之前创建的中间缓冲区,将块复制到 GPU。再将体素块描述符插入到索引表,分配堆存储体素块,并同时复制体素数据。这与分配阶段类似(参见本文第 5 节),不同点在于流传输过程中,所有索引表项必须插入到单个帧中,而不是交错插入。

最后,对于流处理过的体素块,检测描述符确定该位置是否在索引表中已经被占用。当被占用时,继续遍历序列中的下一个可以空闲位置,否则,将该位置的体素项写入索引表。当序列已满时,将该体素项附加到表的末尾链表中。

### 6.3 体素块的流动与分配同步问题

流传输要确保体素块不在 GPU 或 CPU 上复制传输,以免出现内存泄漏现象。当流从 CPU 传输到 GPU 时,存在等待被流传输的体素块进入平截头体的罕见情况,为了保证这些交错区域中没有对这些体素块的再次分配,本文算法在 GPU 上存储了一个二进制网格,每个数据项对应一个块。该位为 1 表示该块驻留在 GPU 上,为块分配存储单元,为 0 表示该块在 CPU 上,应避免分配。二进制网格标记  $2^{563}$  个立方块,占用 512 kB 的 GPU 内存开销,可以快速重新分配,实现大尺度场景的快速重建。

## 7 实验分析

### 7.1 实验的软硬件环境

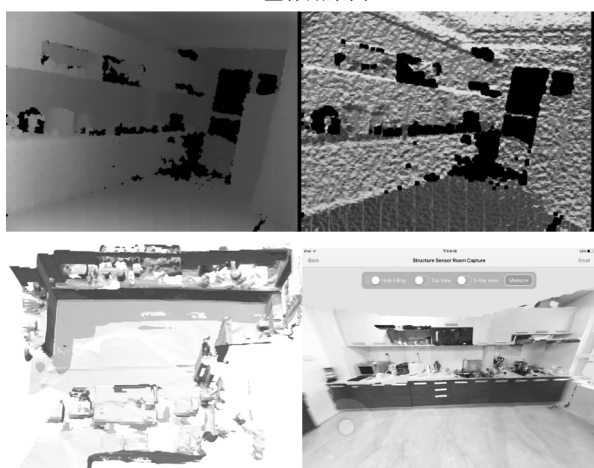
操作系统: Windows 8.1; 开发语言: C++; 开发工具: Microsoft Visual Studio 2014; 三维图形编程接口: Direct3D 11; CPU 类型/主频: INTEL Core i5/3.0 GHz; RAM: 8G; 图形处理器: NVIDIA GTX 960; 传感器: Asus Xtion Pro, RGB-D 30 Hz.

### 7.2 性能分析

在各种光照条件下拍摄了多个场景,对本文重建系统进行性能测试,如图7所示.对约3 m高,16 m<sup>2</sup>的餐厅场景进行在线重建,用时5 min;对12 m<sup>2</sup>的办公室场景的重建时间大约4 min(如图10).这些大尺度场景的平均重建速率达到30 Hz,本文系统在重建质量和速率方面表现良好.



(a) 整体效果图



(b) 深度图, 法相图, 融合图, 细节图

图7 本文算法对餐厅场景的重建效果图

Fig. 7 Reconstruction of din-room with our algorithm

平均重建时间为21.6 ms, ICP 姿态估计15次, 用时8.0 ms, 占系统总体重建时间的37%, 表面融合为4.6 ms(21%), 表面提取占用4.8 ms

(22%), 流处理占用4.4 ms(20%). 每个测试场景的重建时间如图8所示.

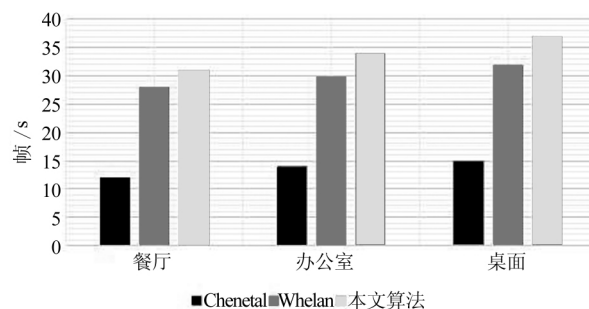


图8 Whelan, Chenetal 算法与本文算法的帧绘制速率对比图

Fig. 8 Comparison among Whelan, Chenetal and our algorithm for frame drawing rate

本文的空间体素数据结构使用34 MB的存储单元和辅助缓冲区. 包含 $2^{21}$ 个空间体素项, 每个体素项占用12 B. 序列的大小为24 B. 预先分配1 GB的堆存储单元给GPU. 每个体素块包含 $8^3$ 个体素, 每个体素8字节, 共 $2^{18}$ 个体素块. 注:  $2^{21}$ 个空间体素项索引到 $2^{18}$ 个体素块, 可降低索引冲突.

当选取体素块的大小为8 mm时(随着场景复杂程度而变化), 需要分配140 k个体素存储单元. 此时当序列大小为12 B时, 需要分配120 k个序列, 序列大小为24 B时, 则分配10 k个序列. 因此当序列为24 B, 空间体素项为 $2^{21}$ 时, 所有测试场景的溢率为0.1%. 链表长度为3, 总计大约700个链表, 与索引表大小相比是可以忽略的. 系统平均为表面数据分配不到300 MB的存储单元, 剩余的空间足够用来将RGB数据直接编码到体素存储单元.

本文系统采用的小序列-大索引的数据结构重建性能良好. 例如, 在餐厅场景中,  $2^{21}$ 个体素项占用索引表的6.4%, 运行时间是21 ms, 200 k个体素项占用率上升到65%, 性能降低大约24.5 ms, 160 k个体素项占用率上升到81%, 性能下降25.6 ms. 由于较小的索引表会导致较高的处理器占用率, 与1 GB的体素块缓冲区相比, 本文算法使用较大的索引表降低了时空复杂度. 图9给出了Whelan算法(简称WA)、Chenetal算法(简称CA)与本文算法的重建性能比较. 图中显示了本文方法在不同噪声和三维点数的情况下, 目标函数值随迭代次数的变化情况. 本文方法的收敛时间与三维点数大约呈现出线性关系, 在不同噪声强度下的收敛性能差别不大, 可以处理实际应用中的大量三维点, 适用于动态场景的实时三维重建.



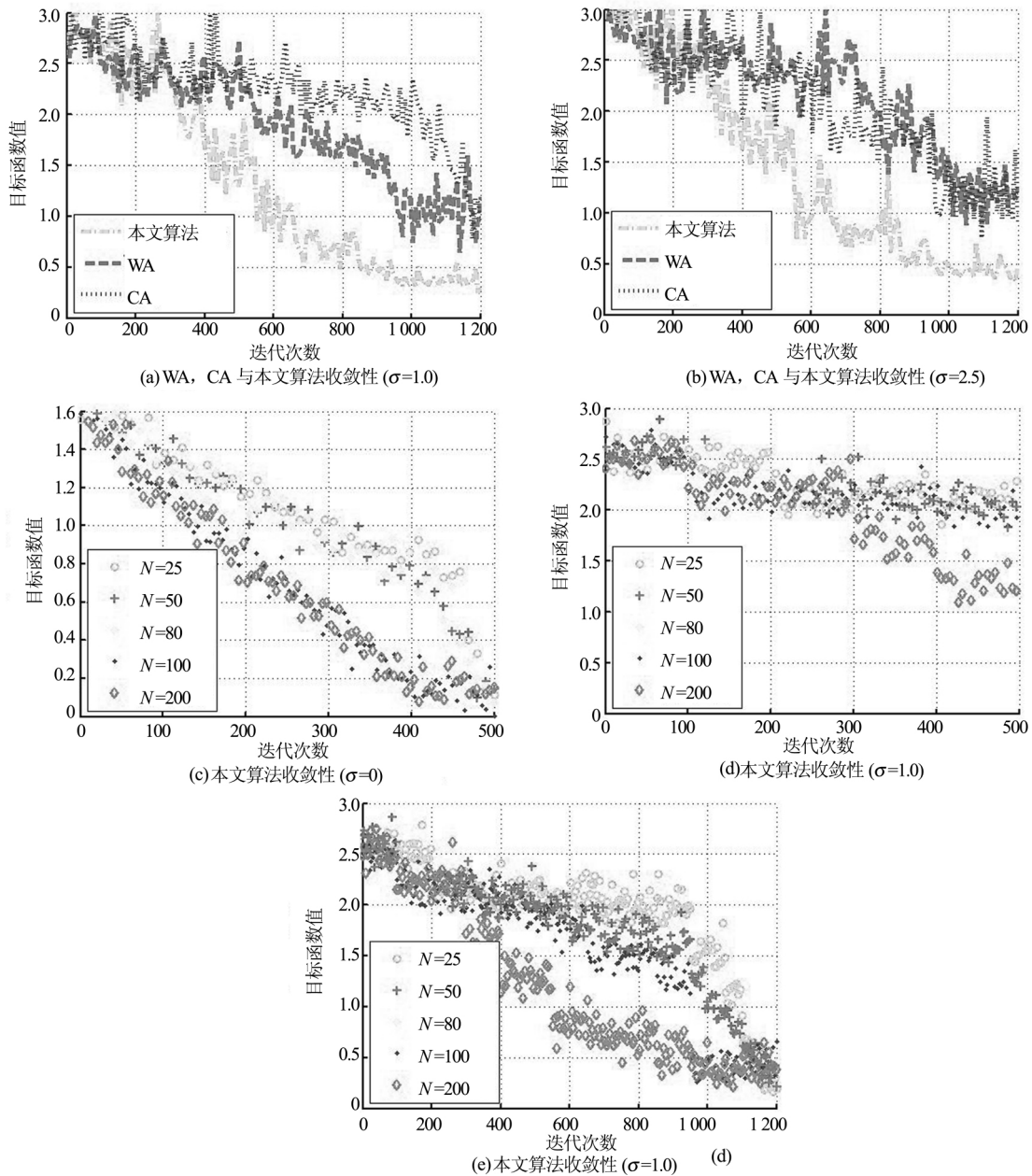


图 9 算法收敛性分析  
Fig. 9 Convergence analysis for the algorithms

### 7.3 几种重建方法的比较

在使用相同的硬件和 ICP(取 15)值的情况下,选取 Whelan 提出的实时大尺度密集融合算法和 Chenetal 提出的分层融合系统,将本文系统与同样支持实时流的这两种重建系统进行重建质量与速度的比较.分层融合受其体积结构的复杂程度限制,当重建场景较复杂时,会导致重建性能的下降.例如,对图 7 所示的几个场景进行重建,整个层次融合系统(包括姿态估计,融合和流处理)的运行速率是 15 Hz,低于传感器输入帧速率.传统体积重建方法需要对规则网格的空体素进行处理,尤其是光线投射

期间对 GPU 缓冲区的存取,耗费较大的流开销.本文系统的空间体素数据结构只对非空体素进行索引,流开销较小.

在图 10 中,统一选取体素块的大小为 4 mm,本文系统能够在较高帧频下对 4 mm 体素完成高质量的 3D 重建,如图 10(d) 所示;大尺度密集融合算法的体素分辨率较粗糙(如图 10(a) 矩形框所示),为了达到相同的重建质量,需要降低体素块的大小(4 mm),如图 10(b) 矩形框所示,但存在体素缺失现象;分层体积算法会导致更多的跟踪漂移,降低整体重建质量(如图 10(c) 矩形框所示).由于密集融

合算法的重建质量受移动体积的小空间范围限制,使得部分传感器数据超出范围没有完全融合,而分层融合算法较差的帧速率导致输入数据被跳过,影响了姿态估计准确性,出现不准确的表面融合以及漂移现象。

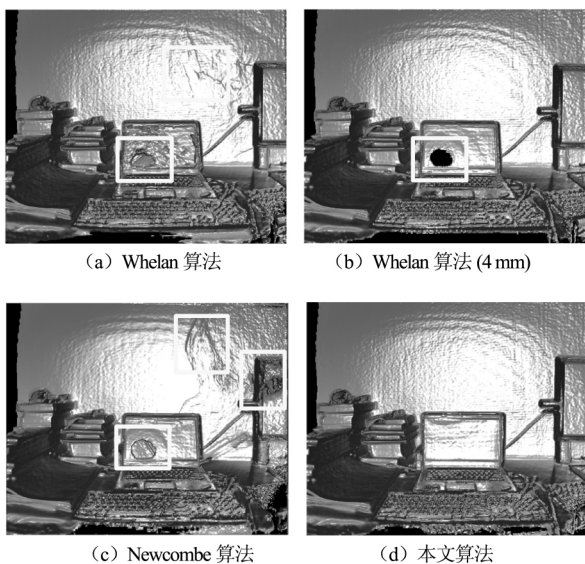


图 10 几种重建算法的性能比较  
Fig. 10 Comparison of several reconstruction algorithms

在大尺度场景中,漂移会导致重建效果的扭曲,如图 11 所示.本文系统的运行速度平均达到 30 Hz,可以通过增加 ICP 迭代的数量来提高姿态估计的精度,进而提高重建质量.从图 11 中可以看出,本文系统确实存在小漂移,但对于这种大尺度的重建系统,与以往重建系统相比,更加适用于在线应用。

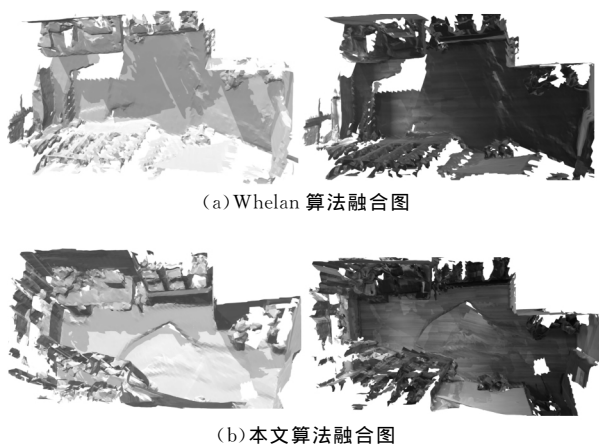


图 11 本文算法与 Whelan 算法的大尺度重建效果图  
Fig. 11 Comparison between Whelan and our algorithm for large scale reconstruction

## 8 总 结

本文提出的基于 GPU 硬件加速和深度传感器的空间体素融合系统,专门用于大尺度在线 3D 重建.本文算法对传统隐含表面的体积融合方法进行改进,使用紧凑的空间体素数据结构,有效捕获体积融合的三维体素,降低了规则体素网格和分层数据结构的开销,整体重建过程支持基于 GPU 的轻量级流操作.若结合使用更高分辨率的深度传感器,本文算法的大尺度重建效果将更加精准.相信在不久的将来,基于空间体素融合的新技术必将广泛应用于 3D 打印设备上,在 AR/VR, SLAM, 机器人视觉等领域前景看好。

## 参考文献

- [1] GURJAP S, SANJEEV S, PRABIR K P. Mobile robot localization with Kinect RGB-D sensor[C]// Proceedings of Conference on Advances in Robotics. India: ACM, 2013: 1-6.
- [2] KORNUA T, LASZKOWSKI M. Perception subsystem for object recognition and pose estimation in RGB-D images[J]. Springer International Publishing, 2016, 44(10): 995-1003.
- [3] HOPPE H, DEROSE T, DUCHAMP T, *et al.* Surface reconstruction from unorganized points[J]. ACM SIGGRAPH Computer Graphics, 1992, 26(2):71-78.
- [4] KAZHDAN M, BOLITHO M, HOPPE H. Poisson surface reconstruction[C]// Proceedings of Eurographics Symposium on Geometry Processing. Cagliari, Sardinia, Italy: DBLP, 2006:61-70.
- [5] CHEN Y, MEDIONI G. Object modelling by registration of multiple range images[J]. Image and Vision Computing, 1992, 10(3):145-155.
- [6] NEUGEBAUER P J. Geometrical cloning of 3D objects via simultaneous registration of multiple range images[C]// Proceedings of International Conference on Shape Modeling and Applications. Washington DC: IEEE Computer Society, 1997: 130-139.
- [7] TURK G, LEVOY M. Zippered polygon meshes from range images[C]// Computers and Graphics, 1994:311-318.
- [8] RUSINKIEWICZ S, HALL-HOLT O, LEVOY M. Real-time 3D model acquisition[J]. ACM Transactions on Graphics, 2002, 21(3):438-446.
- [9] ZOLLH ÖFER M, THIES J, COLAIANNI M, *et al.* Interactive model-based reconstruction of the human head using an RGB-D sensor[J]. Computer Animation and Virtual Worlds, 2014, 25(25): 213-222.
- [10] WIKOWSKI A, KORNUA T, STEFANCZYK M, *et al.* Efficient generation of 3D surfel maps using RGB-D sensors[J]. International Journal of Applied Mathematics and Com-

- puter Science, 2016, 26(1): 99–122.
- [11] KAZDAN M, HOPPE H. Screened poisson surface reconstruction[J]. ACM Transactions on Graphics, 2013, 32(3): 61–70.
- [12] WANG Y L, ZHANG Q Z, ZHOU Y L. Dense 3D mapping for indoor environment based on Kinect-style depth cameras [J]. Springer International Publishing, 2015, 345:317–330.
- [13] POLLEFEYS M, NISTER D, FRAHM J, *et al.* Detailed real-time urban 3D reconstruction from video[J]. International Journal of Computer Vision, 2008, 78(2):143–167.
- [14] NGUYEN T T, SLAUGHTER D C, MAX N, *et al.* Structured light-based 3D reconstruction system for plants[J]. Sensor, 2015, 15(8):587–612.
- [15] CURLESS B, AND LEVOY M. A volumetric method for building complex models from range images[C]// Proceedings of Conference on Computer Graphics and Interactive Techniques. New York: ACM, 1996:303–312.
- [16] LABATUT P, PONS J P, KERIVEN R. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts[C]// Proceedings of IEEE International Conference on Computer Vision. Brazil: IEEE Computer Society, 2007, 1–8.
- [17] PEASLEY B, BIRCHFIELD S, CUNNINGHAM A, *et al.* Accurate on-line 3D occupancy grids using Manhattan world constraints[C]// Proceedings of IEEE International Conference on Intelligent Robots and Systems. Portugal: IEEE Computer Society, 2012: 5283–5290.
- [18] KELLER M, LEFLOCH D, LAMBERS M, *et al.* Real-time 3d reconstruction in dynamic scenes using point-based fusion [C]// Proceedings of International Conference on 3d Vision-3dv. Seattle WA:3DV, 2013, 1–8.
- [19] MARÍA J G, FERNANDO G, DAVID M, *et al.* Intelligent surveillance of indoor environments based on computer vision and 3D point cloud fusion[J]. Expert Systems with Applications, 2015, 42(21): 8156–8171.
- [20] WHELAN T, LEUTENEGGER S, SALAS-MORENO R F, *et al.* ElasticFusion: Dense SLAM without a pose graph[C]// Proceedings of Robotics, Science and Systems (RSS). Rome Italy: 2015.
- [21] NEWCOMBE R A, IZADI S, HILLIGES O, *et al.* KinectFusion: Real-time dense surface mapping and tracking[C]// Proceedings of IEEE International Symposium Mixed and Augmented Reality. Washington DC: IEEE Computer Society, 2011: 127–136.
- [22] LOOP C, CHA Z, ZHENG Y Z. Real-time high-resolution sparse voxelization with application to image-based modeling [C]// Proceedings of the 5th High-Performance Graphics Conference. Anaheim, California: ACM, 2013: 73–79.
- [23] BAERT J, LAGAE A, DUTRÉ P. Out-of-core construction of sparse voxel octrees[C]// Proceedings of the 5th High-Performance Graphics Conference. Anaheim, California: ACM, 2013: 27–32.
- [24] ZENG M, ZHAO F, ZHENG J, *et al.* Octree-based fusion for real time 3D reconstruction[J]. Graphical Models, 2013, 75(3):126–136.
- [25] PÄTZOLD M, KOLB A. Grid-free out-of-core voxelization to sparse voxel octrees on GPU[C]//Proceedings of the 7th Conference on High-Performance Graphics. Los Angeles, California: ACM, 2015: 95–103.
- [26] CHEN J, BAUTEMBACH D, IZADI S. Scalable real-time volumetric surface reconstruction[J]. ACM Transactions on Graphics, 2013, 32(4):1–16.
- [27] WHELAN T, KAESS M, JOHANNSSON H, *et al.* Real-time large-scale dense RGB-D SLAM with volumetric fusion [J]. International Journal of Robotics Research, 2014, 34(4/5): 598–626.
- [28] 祝继华, 郑南宁, 袁泽建, 等. 基于 ICP 算法和粒子滤波的未知环境地图创建[J]. 自动化学报, 2009, 35(8):1107–1113. ZHU J H, ZHEN N N, YUAN Z J, *et al.* Creation of unknown environment map based on ICP algorithm and particle filter[J]. Acta Automatica Sinica, 2009, 35(8):1107–1113. (In Chinese)
- [29] TESCHNER M, HEIDELBERGER B, MULLER M, *et al.* Optimized spatial hashing for collision detection of deformable objects[C]// Proceedings of Vision, Modeling, Visualization VMV. München Germany: DFG, 2003:47–54.