软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(10):2612−2621 [doi: 10.13328/j.cnki.jos.005088]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

耳廓三维网格去补丁合并算法*

孙晓鹏[1,2], 盖 宇[1], 徐 南[1], 李 志[1]

[1](辽宁师范大学 计算机与信息技术学院 计算机系统研究所,辽宁 大连 116029)

[2](智能通信软件与多媒体北京市重点实验室(北京邮电大学),北京 100876)

通讯作者: 孙晓鹏, E-mail: CADCG2008@hotmail.com

摘 要: 针对耳廓多角度扫描获取的三维网格合并问题,提出了一种新的三维网格合并方法——去补丁合并法.首先,基于 kd-tree 算法将三维耳廓配准后的两幅网格快速分割为重叠区域与非重叠区域;然后,根据连通性对重叠区域和非重叠区域进行分块,并从重叠区域分块中去除冗余的补丁块、构建边界点;最后,基于边界点将保留的重叠区域网格与邻接的非重叠区域网格缝合.实验结果表明,与同类算法相比,该方法具有较好的合并效果与较高的计算效率.

关键词: 三维耳廓;kd-tree;网格合并;重叠区域;补丁去除

中图法分类号: TP391

中文引用格式: 孙晓鹏,盖宇,徐南,李志.耳廓三维网格去补丁合并算法.软件学报,2016,27(10):2612−2621. http://www.jos.org.cn/1000-9825/5088.htm

英文引用格式: Sun XP, Gai Y, Xu N, Li Z. 3D ear mesh merge algorithm based on patch removing. Ruan Jian Xue Bao/Journal of Software, 2016,27(10):2612−2621 (in Chinese). http://www.jos.org.cn/1000-9825/5088.htm

3D Ear Mesh Merge Algorithm Based on Patch Removing

SUN Xiao-Peng[1,2], GAI Yu[1], XU Nan[1], LI Zhi[1]

[1](Computer System Institute, School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China)

[2](Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia (Beijing University of Posts and Telecommunications), Beijing 100876, China)

Abstract: To address the problem of merging the multiple meshes of the same human ear, which is scanned from different perspectives, this paper proposes a novel approach—patch removing, for merging 3D ear meshes. First, kd-tree is used to segment the two registration meshes of 3D ear into un-overlapping area and overlapped area. Then, the un-overlapping area and overlapping are segmented into blocks according to their connectivity. Next, the redundant patch in the overlapping area is discarded, thus establishing the points on the boundary of blocks. Finally, to merge the 3D ear meshes, the mesh blocks in the un-overlapping area and the rest of the overlapping area are stitched along the boundary points. Experimental results show that, comparing with other works, the proposed algorithm can merge the 3D ear meshes more effectively with higher computational efficiency.

Key words: 3D ear; kd-tree; mesh merge; overlapping area; patch removing

近年来,作为一种新的生物信息技术,三维耳廓的形状分析与识别得到了国际生物信息识别研究领域的关注.Iannarelli 等人的研究和实践已经证实,人类的耳廓具有较高的独特性、唯一性和稳定性,没有任何两个人(即

* 基金项目: 国家自然科学基金(61472170, 61170143, 60873110); 北京邮电大学智能通信软件与多媒体北京市重点实验室开放课题(ITSM201301)

Foundation item: National Natural Science Foundation of China (61472170, 61170143, 60873110); Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia (Beijing University of Posts and Telecommunications) (ITSM201301)

收稿时间: 2016-01-21; 修改时间: 2016-03-25; 采用时间: 2016-05-09; jos 在线出版时间: 2016-08-08

CNKI 网络优先出版: 2016-08-09 15:37:55, http://www.cnki.net/kcms/detail/11.2560.TP.20160809.1537.007.html

© 中国科学院软件研究所 http://www.jos.org.cn

使是双胞胎)的耳朵外形是完全一样的,而且在 8~70 岁之间耳廓外形不会发生显著变化[1].

目前,获取耳廓几何外形的基本方法是使用三维激光扫描仪对耳廓进行多角度扫描以获取耳廓的深度信息,然后进行三维点云配准、三维网格合并等基本数字几何处理,最终得到完整的、高质量的耳廓三维单层网格数据[2].

由于耳廓的物理尺寸较小且具有丰富的沟壑外形特征,为了保证最终耳廓网格数据的完整性,多幅扫描深度数据之间必须满足重叠区域规模较大、非重叠区域规模相对较小的要求.如何快速、有效地对重叠情况严重的多幅耳廓网格数据进行快速、有效的删减和重新网格化、合并为单层无重叠的完整耳廓网格,便成为基于耳廓身份识别的关键问题[3−6].

针对多幅网格间的过渡区域内存在网格多次相互交叉、平行叠盖并存在扫描间隙等复杂问题,本文工作在 Kaushik 等人[5]的 Patch 思想启发下,提出了一种新的网格重构方法——去补丁合并法.首先,对齐配准待合并的两幅耳廓网格数据,确定两幅网格之间的重叠区域和非重叠区域;然后,对重叠区域与不重叠区域进行分块,确定连续的重叠区域和不重叠区域的数目,保留重叠区域内顶点数目较多的网格块,剔除顶点数目较少的网格块(即补丁),并构造边界点;最后,应用简化的网格缝合算法,沿边界点将非重叠区域缝合到保留的重叠区域网格块上,实现两幅网格的合并.

本文首先概述耳廓三维形状可用于生物信息识别以及耳廓网格合并存在的困难.第 1 节对近年来国际国内的相关工作进行分析介绍,考虑到计算效率因素,没有采用点云曲面网格重构方法,而将耳廓网格合归为网格合并重构类问题.第 2 节对本文提出的算法进行详细的介绍,具体包括区域分块、去除补丁、网格缝合这 3 部分.第 3 节给出部分合并实验结果,以说明本文算法的有效性,并与同类工作进行对比分析,给出本文算法的计算效率.最后总结全文,并对未来值得关注的研究方向进行初步探讨.

1 相关工作

多幅扫描数据合并是数字几何处理领域的基本重构问题之一,需要在保持三维模型的基本形状信息不变的前提下对多幅扫描数据的几何信息进行重采样,对拓扑信息进行重连接,一般可以分为点云曲面网格重构[7−9]和网格合并重构[10−12]这两类.

点云的曲面网格重构即对不附加任何几何信息与拓扑信息(如法矢、曲面边界、连通性、曲面孔洞数)的散乱点云进行重构曲面的过程.近年来,在此方向上取得了较好的研究进展,如连续全局优化方法基于最小化能量重构离散噪音点云的全局最优曲面[13];基于粒子群演化算法对离散点参数化并进行非均匀有理 B 样条拟合[14];逐层外扩原始点云的近似网格,并通过层次 B 样条方法对近似网格进行细分降噪[15];结合三维 Delaunay 三角剖分与局部增长算法,自适应地动态搜索增长点,避免人为阈值带来的不可靠性,以及投影法引起几何形变问题等.此类工作从三维点云中重建网格的算法精度较高,计算效率较低,与本文工作的相关度较低.

网格合并的重构算法则面向以多边形网格描述的几何模型.较为经典的网格合并算法是 1994 年 Turk 等人[16]提出的直接在网格上进行操作的合并算法——Mesh Zippering,首先交替去除重叠部分,再对两幅网格的不重叠部分进行夹合裁剪,以新的顶点分割两幅网格的边界三角片并删除冗余三角片.该算法的核心工作分为 3 个步骤:剔除两幅网格重叠部分、两幅网格互相剪切、删除剪切产生的碎小三角片.该算法适用于多幅网格重叠区域,不过,随着重叠区域的增多,合并效果逐渐降低,计算效率随之降低;在合并扫描角度差异较大的多幅网格时,裁剪操作将产生大量的细碎三角片,人工算法痕迹明显.2008 年,Hui 等人[17]基于 Zipper 算法提出网格缝合算法,通过构造缝合线(stich line)建立待合并网格的公共边界,以缝合线为骨架构造连接两幅网格的缝合面.该方法着重介绍了两幅相交网格的缝合过程,未对重叠问题展开讨论,更适用于只存在网格边界重叠的情况.2012 年,Du 等人[18]针对多表面模型提出了自动网格划分与合并技术.首先将模型转换成三角网格,保留模型的表面特征、边界和体积层次结构;计算模型的相交三角面片,生成相交环;基于约束 Delaunay 算法将相交环中的多边形剖分为三角片;最后删除冗余三角片完成模型合并.该算法的不足之处在于,仅通过面片之间的位置关系判断是否删除,合并效果较差.2013 年,Jonathan 等人[19]提出多边形网格 MeshGit 合并算法.首先建立网格模型之间的

© 中国科学院软件研究所 http://www.jos.org.cn

匹配点对;以贪心算法迭代计算匹配顶点与面片之间的最小距离,作为网格模型之间的差异测度;最后合并差异测度较低的局部网格.该算法的缺陷是需要人工交互干涉,实用性较低.2014 年,Yue 等人[20]基于泊松克隆理论,采用中值坐标,对任意不规则的多幅网格进行合并,不足之处是需要构造参数映射,因此难以处理网格空洞问题.2014 年,Jun 等人[21]提出改进的四面体网格生成算法,该方法能够抑制扁平面体,生成优质四面体网格,并保证边界一致性,计算复杂度显然高于一般的曲面网格模型合并.该算法的缺陷是结合平衡八叉树和空间质心差值算法构造尺度控制,迭代阈值的选择影响了网格生成效率,该方法仅对封闭的三维表面网格有效.显然,上述算法均难以处理耳廓三维网格的合并问题.

近年来,国际国内针对网格合并展开的研究工作较少,除 1994 年 Turk 等人[16]提出的 Mesh Zippering 外,2008 年 Hui 等人[17]提出网格缝合算法具有重要的参考价值,并已基于该算法研发出经济实用的手持式三维激光扫描设备.本文对 Hui 等人[17]的工作进行了改进,借鉴深度图像处理中的 Patch 思想[5],针对多幅耳廓网格数据之间存在大规模重叠的显著特征,提出去补丁合并法.检查网格数据重叠状况,生成重叠分割块和分重叠分割块,删除补丁块,确定并合并区域网格,提高了大规模重叠区域检测的计算效率,并保留适当的重叠边界以实现网格缝合.本文的贡献在于:基于重叠区域和非重叠区域块分割的思想提出补丁的概念,定义了边界点,并提出了新的网格缝合算法.

2 网格合并

对于配准后的任意两个耳廓三维网格模型,设其顶点集合分别为 $M_a=\{\mathbf{pa}_i,i=1,2,\ldots,ma\}$、$M_b=\{\mathbf{pb}_j,j=1,2,\ldots,mb\}$,其中,$ma$ 和 mb 分别表示 M_a 和 M_b 的顶点数目.本文给出的耳廓三维网格合并过程可以分为区域分块、补丁去除、网格缝合这 3 步.

2.1 区域分块

记 M_a 和 M_b 的重叠区域分别为 O_a 和 O_b,并记非重叠区域分别为 U_a 和 U_b.对于 M_a 上的任意一点 $\mathbf{pa}_i$,设在 M_b 上与其距离最近的对应点为 $\mathbf{pb}_j$,则令:

$$\begin{cases}\mathbf{pa}_i\in O_a,\mathbf{pb}_j\in O_b & \text{when}\left\|\mathbf{pa}_i-\mathbf{pb}_j\right\|<\varepsilon\\ \mathbf{pa}_i\in U_a,\mathbf{pb}_j\in U_b & \text{else}\end{cases}.$$

即:当距离 $d=\|\mathbf{pb}_j-\mathbf{pa}_i\|$小于指定的阈值$\varepsilon$时,分别将 $\mathbf{pa}_i$ 标记为重叠区域 O_a 内的点,将点 $\mathbf{pb}_j$ 标记为重叠区域 O_b 内的点,否则,分别将 $\mathbf{pa}_i$ 标记为非重叠区域 U_a 内的点,将 $\mathbf{pb}_j$ 标记为非重叠区域 U_b 内的点.

在保持 M_a 和 M_b 原有连通性的前提下,根据重叠区域 O_a 和 O_b、非重叠区域 U_a 和 U_b 的分割情况,对各点连通性更新如下:对于 M_a 或 M_b 的任意两个不连通的顶点 $\mathbf{p}_i$ 和 $\mathbf{p}_j$,保持其不连通性;对于 M_a 或 M_b 的任意两个连通的顶点 $\mathbf{p}_i$ 和 $\mathbf{p}_j$,如果同被标记为重叠区域点或非重叠区域点,则保持其连通性;若分别被标记为重叠区域点和非重叠区域点,则删除两顶点间原有的连接边,两顶点间不再连通,同时将此两顶点标记为分块边界点.

根据各顶点所在网格的连通性,可对重叠区域 O_a 和 O_b、非重叠区域 U_a 和 U_b 分别进行分块,每个分块 O_{ai},O_{bi}、U_{ai} 和 U_{bi} 都连通顶点构成,分别记 $O_a=\{O_{ai},i=1,2,\ldots,oa\}$,$O_b=\{O_{bi},i=1,2,\ldots,ob\}$,$U_a=\{U_{ai},i=1,2,\ldots,ua\}$,$U_b=\{U_{bi},i=1,2,\ldots,ub\}$,其中,$oa$、$ob$、$ua$ 和 ub 分别表示重叠区域 O 和非重叠区域 U 分块的数目.各连通分块分别记为 $O_{ai}=\{\mathbf{pa}_j,j=1,2,\ldots,ona\}$,$O_{bi}=\{\mathbf{pb}_j,j=1,2,\ldots,onb\}$,$U_{ai}=\{\mathbf{pa}_j,j=1,2,\ldots,una\}$,$U_{bi}=\{\mathbf{pb}_j,j=1,2,\ldots,unb\}$,其中,$ona$,$onb$,$una$ 和 unb 分别表示 O_{ai},O_{bi}、U_{ai} 和 U_{bi} 中顶点的数目.如果分块 O_{ai},O_{bi}、U_{ai} 和 U_{bi} 中顶点的数目较少,则将该分块归并到最邻近的分块中,以保证不产生过分割.

图 1 所示为区域分块的基本流程.在三维耳廓网格模型中提取点云(即网格顶点集合),得到蓝色模型 M_a 和浅绿色模型 M_b;M_a 和 M_b 配准后的重叠区域以红色表示;根据模型重叠区域与非重叠区域的连通性对 M_a 和 M_b 进行分块,得到浅绿色、深绿色、蓝色、深紫色、浅紫色、红色等 7 个分块;最后根据模型分块区域的连通进行网格拓扑连接还原,其中,在被标记为分块边界点的顶点邻域内,在原网格拓扑连接中,跨越分块 O_{ai} 和 U_{ai}、O_{bi}

© 中国科学院软件研究所 http://www.jos.org.cn

和 U_{bi} 边界的边被删除.

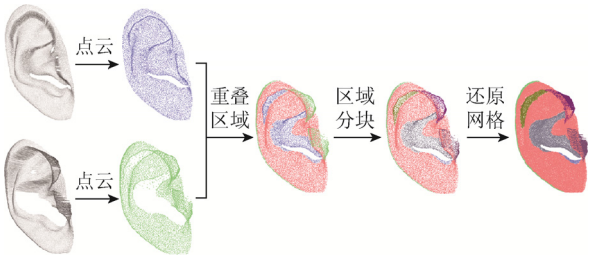


Fig.1 Pipeline of segment ear to regional patch

图 1 耳廓区域分块流程

在 M_b 上搜索距离 M_a 的 $\mathbf{pa}_i$ 距离最近的对应点 $\mathbf{pb}_j$ 的过程计算效率较低,本文基于 CUDA 对该搜索过程进行 kd-tree 并行加速[22],在每个 kernel 中只需对网格上的两个点进行一次距离计算并判断是否在阈值内即可,从而大幅度缩短了运算时间.

2.2 去除补丁

由于重叠区域的网格分块有两层,存在信息冗余,所以需要去除一层,本文称被去除的一层重叠区域网格分块为补丁.保留层内的重叠区域分块将与周边的非重叠区域分块合并,构成最终的单层网格.

例如,网格分块 M_a 和 M_b 边界重叠如图 2 所示情况.对于 M_a 的任意重叠分块 O_{ai},设其在 M_b 中的对应重叠分块为 O_{bi},比较 O_{ai} 和 O_{bi} 的顶点数目 ona 和 onb,若 $ona>onb$,则将重叠分块 O_{bi} 作为补丁,否则将重叠分块 O_{ai} 作为补丁.

设重叠分块 O_{ai} 保留,则将其邻接的所有不重叠区域分块 U_{ai} 以及与补丁 O_{bi} 邻接的所有不重叠区域分块 U_{bi} 与 O_{ai} 直接缝合.对于不重叠分块 U_{bi} 的边界点 $\mathbf{pb}_i$,设其在 O_{bi} 中的对应边界点为 $\mathbf{pb}_j$;设 $\mathbf{pb}_j$ 在 O_{ai} 中的对应点为 $\mathbf{pa}_j$,则记缝合点对为 $MP_j=(\mathbf{pa}_j,\mathbf{pb}_j)$.同理,对于 U_b 的 ub 个分块,记全部缝合点对的集合为 $MP=\{MP_i,i=1,2,\ldots,ub\}$.在完成缝合点对的集合 MP 构造后,将补丁 O_{bi} 删除.

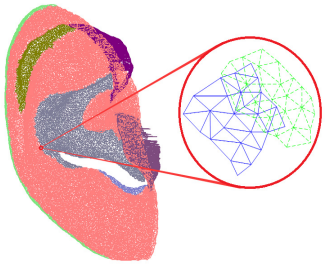
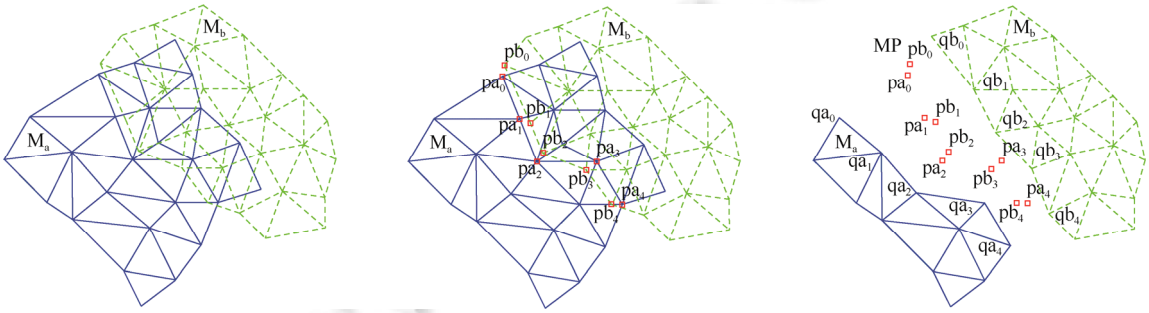


Fig.2 Overlapping border of patch M_a and patch M_b

图 2 网格分块 M_a 和 M_b 的重叠边界

图 3 所示为补丁删除过程.其中,网格分块重叠区域边界点及其对应点 $\mathbf{pa}_0$~$\mathbf{pa}_4$,$\mathbf{pb}_0$~$\mathbf{pb}_4$ 标记为红色(如图 3(b)所示).在删除缝合点的邻接边以及与 M_b 的重叠区域后,记网格分块 M_a 的新边界点为 $\mathbf{qa}_0$~$\mathbf{qa}_4$,记网格分块 M_b 的新边界点为 $\mathbf{qb}_0$~$\mathbf{qb}_4$(如图 3(c)所示).



(a) 网格分块重叠边界 (b) 重叠边界对应点 (c) 删除重叠网格及重叠边界

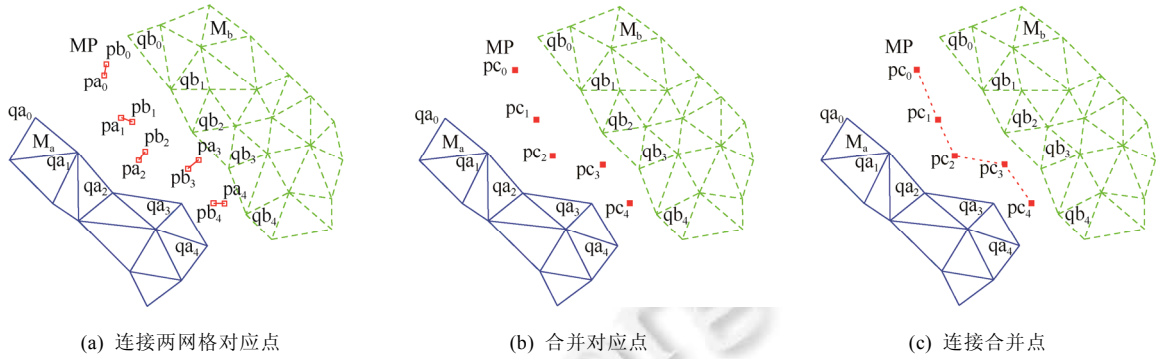
Fig.3 Procedure of removing patch

图 3 补丁去除过程

© 中国科学院软件研究所 http://www.jos.org.cn

2.3 网格缝合

网格缝合最终完成两层重叠网格数据的合并.重叠区域分块与不重叠区域分块之间的缝合点集合 MP_i,其缝合线构造过程如图 4 所示.图 4(a)中,每个缝合分组均包括两个网格重叠区域边界点的对应点 $\mathbf{pa}_0$~$\mathbf{pa}_4$,$\mathbf{pb}_0$~$\mathbf{pb}_4$;图 4(b)中的点 $\mathbf{pc}_0$~$\mathbf{pc}_4$ 分别为点对 $\mathbf{pa}_0\mathbf{pb}_0$~$\mathbf{pa}_4\mathbf{pb}_4$ 的中点;顺序连接 $\mathbf{pc}_0$~$\mathbf{pc}_4$ 即可得到一条缝合线(如图 4(c)所示).



(a) 连接两网格对应点 (b) 合并对应点 (c) 连接合并点

Fig.4 Procedure of constructing stitch line

图 4 生成缝合线

图 5 为网格缝合过程,本文基于 Delaunay 三角剖分算法,基于网格分块 M_a 和 M_b 的新边界点集 $\mathbf{qa}_0$~$\mathbf{qa}_4$ 和 $\mathbf{qb}_0$~$\mathbf{qb}_4$ 以及缝合线的点集 $\mathbf{pc}_0$~$\mathbf{pc}_4$,构造新的缝合网格,实现网格分块 M_a 和 M_b 的合并.本文使用 CUDA 并行对每个 MP_i 使用网格缝合算法,优化了合并速度.

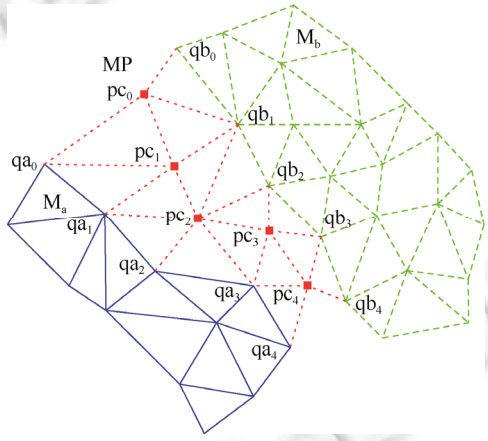


Fig.5 Result of mesh merging

图 5 缝合网格

3 实验结果及分析

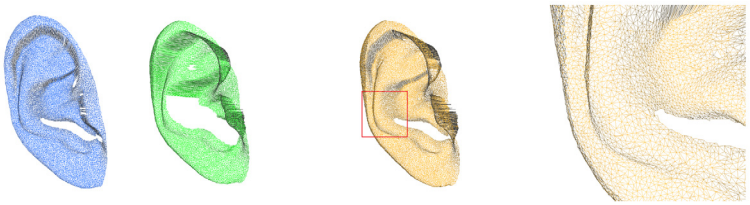
本文实验的软硬件环境为 Intel Xeon E5-2609@ 2.40Ghz,16GB RAM,64 位 Windows 7 操作系统.所用的 GPU 为 NVIDIA Quadro 2000,核心数 192,显存 1GB GDDR5.编程平台为 Microsoft Visual Studio 2008,CUDA6.5 以及 OpenGL.实验数据像素分辨率为 640×480 三维耳廓网格模型.

3.1 合并效果

图 6 所示为利用本文算法对三维耳廓的两幅网格进行合并的效果.图 6(a)为两幅已采用 ICP 算法配准、待

© 中国科学院软件研究所 http://www.jos.org.cn

合并的三维耳廓网格,图 6(b)为合并后的单幅三维耳廓网格效果,图 6(c)是图 6(b)合并区域的局部放大.可以看出,合并后的单层网格几乎看不到人工痕迹,合并效果较为理想.



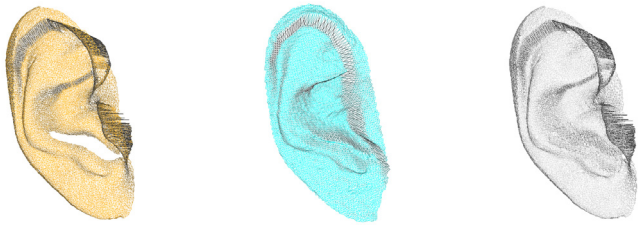
(a) 待合并的两个网格模型 (b) 合并后的网格模型 (c) 合并处局部效果

Fig.6 Result of merging two mesh of ear

图 6 两幅耳廓网格合并

如图 7 所示,将图 6(b)合并后得到的单层网格模型(如图 7(a)所示),与同一人耳的第 3 幅网格(如图 7(b)所示)再次合并,最终得到一个完整的单层的耳廓网格(如图 7(c)所示).

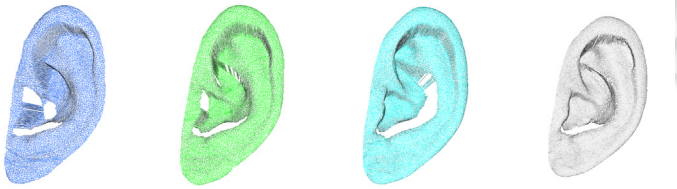
图 8 给出了另外 3 幅耳廓网格的合并效果.可以看出,由于扫描角度存在较小差别,导致前 3 幅网格(图 8(a)~图 8(c))之间存在较大的重叠区域,图 8(d)为最终合并得到的单层耳廓三维网格.



(a) 两层数据合并后的单层网格 (b) 与同一人耳的第 3 幅网格再次合并 (c) 3 层数据合并后的单层网格

Fig.7 Stitch the merging mesh to 3rd ear mesh

图 7 多次渐增合并效果



(a) 第 1 层网格 (b) 第 2 层网格 (c) 第 3 层网格 (d) 最终的合并网格

Fig.8 Merging mesh of three mesh from same ear

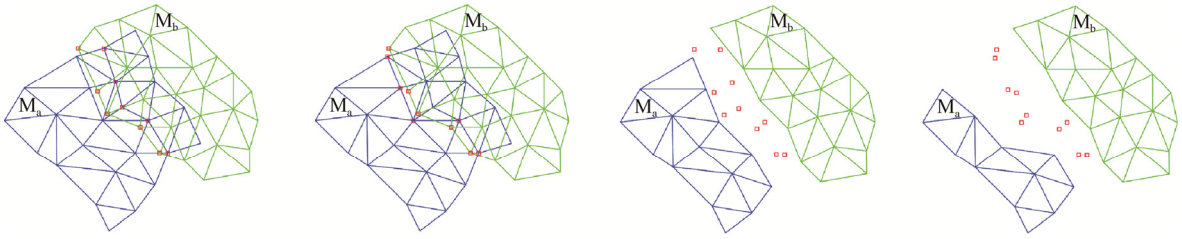
图 8 同一耳廓的 3 幅耳廓网格合并

3.2 网格缝合对比

本文算法是对文献[17]工作的改进.文献[17]合并位置的选择直接使用了 zipper 算法的网格裁剪方法,本文则以最近边界点为合并位置,故产生的缝合点及最终缝合线位置有所不同(如图 9 所示).从图 9 中可以看出,文献[17]算法给出的缝合点比较分散,点与点之间的对应关系及位置不确定,需要额外两两计算距离再选择最近顶点合并.本文算法给出的缝合点是两幅网格的重叠对应点,缝合点的位置分布更为合理,所得网格三角形更趋近于等边三角形,合并网格的质量更高.由于缝合点对具有已知的两两对应关系,不需要进行额外的计算和选择判

© 中国科学院软件研究所 http://www.jos.org.cn

断即可直接合并,因此计算效率较高.



(a) 文献[17]算法的缝合位置 (b) 本文算法的缝合位置 (c) 文献[17]算法的缝合点 (d) 本文算法的缝合点

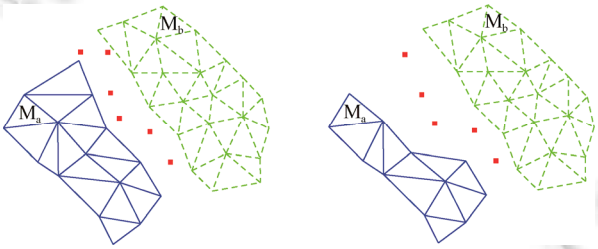
Fig.9 Comparison of the location of the stitch line

图 9 缝合位置对比

图 10 为文献[17]算法与本文算法合并缝合点后得到的新的缝合点.可以看出,文献[17]算法给出的合并缝合点的位置分布得不够理想,个别缝合点距离网格边界较近,需要进一步使用最近点扩展法删除多余的缝合点.本文算法的合并缝合点基本位于两幅网格的中间位置,不再需要对缝合点进行判断和删除等操作,计算效率较高.

图 11 所示为文献[17]算法与本文算法生成的缝合线对比情况.可以看出,由于文献[17]算法缝合点的位置偏向一侧,给出的缝合线平滑性较低.另外,由于缝合点的不确定性,生成缝合网格时需逐一判断选择适当的边界点与缝合点连接.本文算法给出的缝合线则相对平滑,因为本文算法的缝合点由两幅网格的对应点合并而得,保留着与两幅网格之间的拓扑连接信息,故可以直接将缝合点与两幅网格的边界点进行连接,计算效率较高.

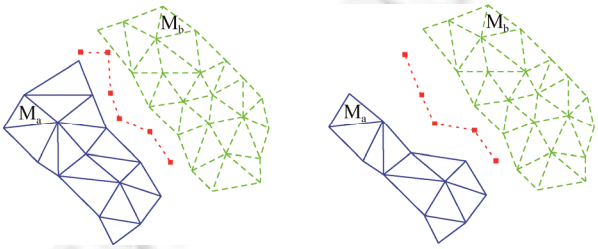
图 12 所示为文献[17]算法与本文算法得到的最终合并网格效果对比情况.可以看出,本文算法在网格分布和密度方面都更为合理,合并效果较好.而文献[17]算法在合并后还需要进一步优化,才能达到较好的合并效果.故,由总体对比可知,本文算法在计算时间以及合并效果上都优于文献[17]算法.



(a) 文献[17]算法合并后的缝合点 (b) 本文算法合并后的缝合点

Fig.10 Comparison of the location of the point on stitch line

图 10 合并后的缝合点对比

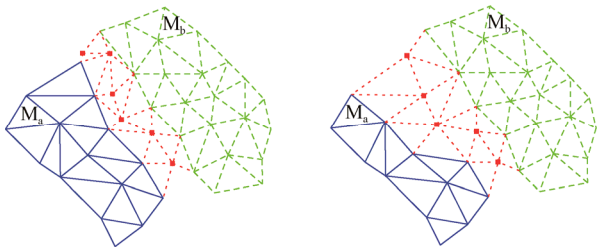


(a) 文献[17]算法生成的缝合线 (b) 本文算法生成的缝合线

Fig.11 Comparison of the stitch line

图 11 生成缝合线对比

© 中国科学院软件研究所 http://www.jos.org.cn



(a) 文献[17]算法的最终合并网格 (b) 本文算法的最终合并网格

Fig.12 Comparison of the result of mesh merging

图 12 最终合并效果对比

3.3 计算效率

图 13 对比了本文算法、本文加速算法以及文献[17]算法的计算时间,由于本文算法在分块 M_b 内搜索距离分块 M_a 的点 $\mathbf{pa}_i$ 距离最近的对应点 $\mathbf{pb}_j$ 的过程计算效率较高,与文献[17]工作相比,在相关网格面片规模下,本文算法具有较高的计算效率.另外,本文给出了基于 CUDA 和 kd-tree 对该搜索过程进行了并行加速,在每个 kernel 中只需对网格上的两个点进行一次距离计算,并判断是否在阈值内即可,因而大幅度缩短了运算时间.并行加速前后的运行时间对比可见表 1.对于网格顶点集合规模 7 000 左右的数据,本文算法在并行加速前用时 1.568s,并行加速后用时 0.354s,加速比约为 2.4 倍,当网格顶点集合规模为 40 000 左右时,并行加速提高计算效率约 20 倍.

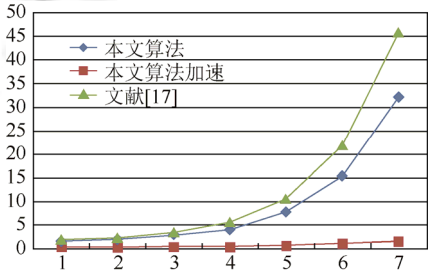


Fig.13 Comparison of the efficiency

图 13 计算效率对比

Table 1 Time comparison of regional search of series and parallel

表 1 查找重叠区域的串/并行时间对比

网格 *a* 点数	网格 *a* 面数	网格 *b* 点数	网格 *b* 面数	重叠点数	加速前(s)	加速后(s)	文献[17](s)
6 995	12 450	7 535	13 120	3 085	1.568	0.354	1.943
7 535	13 120	8 541	16 582	4 264	1.878	0.391	2.365
8 948	18 852	9 702	19 704	5 504	2.924	0.418	3.494
9 702	19 704	15 879	25 827	7 951	4.014	0.471	5.635
15 879	25 827	16 348	28 582	9 632	7.753	0.652	10.546
23 905	41 732	24 806	43 228	13 925	15.312	1.043	21.934
37 854	79 822	44 924	89 165	21 512	32.164	1.533	45.753

4 结束语

本文提出了基于补丁块去除思想的三维耳廓网格模型合并算法,针对三维耳廓扫描网格重叠区域规模较大的特点,基于重叠区域和非重叠区域分块,有效地去除了补丁分块;基于合并点对间的对应关系,快速、有效地构造了缝合点和缝合线,提高了网格缝合效果和计算效率.

© 中国科学院软件研究所 http://www.jos.org.cn

耳廓数据处理可以分为两部分:第 1 阶段为耳廓点云配准,第 2 阶段为耳廓网格融合.本文算法属于第 2 阶段,仅针对理想配准后的耳廓网格的融合问题,通过局部改变网格边缘的拓扑结构,并通过局部微调网格边缘的顶点位置实现多幅网格的融合.在耳廓点云配准不理想的情况下,本文所提出的网格融合算法虽然能够正常实现网格的有效融合,但不会改进多幅耳廓网格的配准效果.下一步将考虑构建反馈机制,考虑如何基于网格融合过程中出现的质量指标反向改进第 1 阶段的配准效果.

另外,现有合并算法往往采用首先对网格两两进行局部的合并,然后网格逐一添加的增量式多次合并模式,如何实现多幅网格的实时整体并行合并是今后深入分析研究的重要问题之一.

References:

[1] Prakash S, Gupta P. An efficient ear recognition technique invariant to illumination and pose. Telecommunication Systems, 2011, 52(2):1435–1448. [doi: 10.1007/s11235-011-9621-2]

[2] Bi ZM, Wang LH. Review: Advances in 3D data acquisition and processing for industrial applications. Robotics and Computer-Integrated Manufacturing, 2010,26(5):403–413. [doi: 10.1016/j.rcim.2010.03.003]

[3] Nagai Y, Ohtake Y, Suzuki H. Tomographic surface reconstruction from point cloud. Computers & Graphics, 2015,46:55–63. [doi: 10.1016/j.cag.2014.09.034]

[4] Albarelli A, Rodolà E, Torsello A. Fast and accurate surface alignment through an isometry-enforcing game. Pattern Recognition, 2015,48(7):2209–2226. [doi: 10.1016/j.patcog.2015.01.020]

[5] Kaushik R, Xiao JZ. Accelerated patch-based planar clustering of noisy range images in indoor environments for robot mapping. Robotics and Autonomous Systems, 2012,60(4):584–598. [doi: 10.1016/j.robot.2011.12.001]

[6] Fuhrmann S, Goesele M. Floating scale surface reconstruction. ACM Trans. on Graphics (TOG), 2014,33(4):1–11. [doi: 10.1145/2601097.2601163]

[7] Zhou QY, Koltun V. Dense scene reconstruction with points of interest. ACM Trans. on Graphics (TOG), 2013,32(4):96. [doi: 10.1145/2461912.2461919]

[8] Bergström P, Edlund O. Robust registration of point sets using iteratively reweighted least squares. Computational Optimization and Applications, 2014,58(3):543–561. [doi: 10.1007/s10589-014-9643-2]

[9] Wu SH, Huang H, Gong ML, Zwicker M, Cohen-Or D. Deep points consolidation. ACM Trans. on Graphics (TOG), 2015,34(6): 1–13. [doi: 10.1145/2816795.2818073]

[10] Nießner M, Zollhöfer M, Izadi S, Stamminger M. Real-Time 3D reconstruction at scale using voxel hashing. ACM Trans. on Graphics (TOG), 2013,3(6):1–11. [doi: 10.1145/2508363.2508374]

[11] Fuhrmann S, Goesele M. Fusion of depth maps with multiple scales. ACM Trans. on Graphics (TOG), 2011,30(6):61–64. [doi: 10.1145/2070781.2024182]

[12] Zou BJ, Zhou HY, Wang L, Liang YX. 3D mesh merging and stitching with large overlaps. Acta Electronica Sinica, 2012,40(5): 1005–1010 (in Chinese with English abstract). [doi: 10.3969/j.issn.0372-2112.2012.05.023]

[13] Pan RJ, Skala V. Continuous global optimization in surface reconstruction from an oriented point cloud. Computer Aided Design, 2011,43(8):896–901. [doi: 10.1016/j.cad.2011.03.005]

[14] Gálvez A, Iglesias A. Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. Information Sciences, 2012,192:174–192 (in Chinese with English abstract). [doi: 10.1016/j.ins.2010.11.007]

[15] Nie JH. Rapid surface reconstruction algorithm from dense point cloud. Journal of Computer-Aided Design and Computer Graphics, 2012,24(5):574–582 (in Chinese with English abstraact). [doi: 10.3969/j.issn.1003-9775.2012.05.002]

[16] Turk G, Levoy M. Zippered polygon meshes form range images. In: Proc. of the SIGGRAPH. 1994,(94):311–318. [doi: 10.1145/192161.192241]

[17] Liu H, Xiang SM, Chen R, Li H. Merged and regularized polygon meshes from range images. Computer Engineering and Applications, 2004,40(29):28−31 (in Chinese with English abstract).

[18] Du JJ, Guo XY, Lu SL, Xiao BX, Wen WL. Automatic mesh split-and-merge technique for multiple surface models. In: Proc. of the 11th ACM SIGGRAPH Int'l Conf. on Virtual-Reality Continuum and Its Applications in Industry. 2012. 227–230. [doi: 10.1145/2407516.2407571]

© 中国科学院软件研究所 http://www.jos.org.cn

[19] Denning JD. MeshGit: Diffing and merging meshes for polygonal modeling. In: Proc. of the ACM Trans. on Graphics (TOG) SIGGRAPH. 2013. [doi: 10.1145/2461912.2461942]

[20] Wang Y, Deng YK, Qian GP. Robust algorithm for real-time mesh geometric editing. Journal of Image and Graphics, 2014,19(5): 764–770 (in Chinese with English abstract). [doi: 10.11834/jig.20140515]

[21] Liu J, Chen WQ, Xiong BS. Three-de mesh generation through volume data reconstructed form surface. Journal of Image and Graphics, 2014,19(5):771–780 (in Chinese with English abstract). [doi: 10.11834/jig.20140516]

[22] Wei H, Du Y, Liang F. A kd tree-based algorithm to parallelize kriging interpolation of big spatial data. GIScience & Remote Sensing, 2015,52(1):40–57. [doi: 10.1016/j.cad.2011.03.005]

附中文参考文献:

[12] 邹北骥,周浩宇,王磊,梁毅雄.大交叠区域的三维网格的融合与拼接.电子学报,2012,40(5):1005–1010. [doi: 10.3969/j.issn.0372-2112.2012.05.023]

[15] 聂建辉.针对密集点云的快速曲面重建算法.计算机辅助设计与图形学学报,2012,24(5):574–582. [doi: 10.3969/j.issn.1003-9775.2012.05.002]

[17] 刘晖,向世明,陈睿,李华.三维扫描网格的合并和优化.计算机工程与应用,2004,40(29):28–31.

[20] 汪悦,邓元凯,钱归平.鲁棒的网格实时几何编辑算法.中国图象图形学报,2014,19(5):764–770. [doi: 10.11834/jig.20140515]

[21] 刘君,陈伟强,熊邦书.从表面重构的体数据实现三维网格剖分.中国图象图形学报,2014,19(5):771–780. [doi: 10.11834/jig.20140516]

孙晓鹏(1968－),男,辽宁大连人,博士,教授,CCF 高级会员,主要研究领域为计算机图形学.

徐南(1991－),女,硕士生,CCF 学生会员,主要研究领域为计算机图形学.

盖宇(1987－),男,硕士,主要研究领域为计算机图形学.

李志(1991－),男,硕士生,主要研究领域为计算机图形学.

© 中国科学院软件研究所 http://www.jos.org.cn