保持特征的点云自适应网格重建

钱归平 童若锋 彭 文 董金祥
(浙江大学计算机科学与技术学院,杭州 310027)

摘 要 由于3维扫描点云通常存在噪音和缺失数据,提出了一种鲁棒的点云网格重建算法。对张量矩阵方法估计的点云法向进行增强特征处理,在频域中进行3维快速傅里叶变换,提取粗糙离散等值面。原始点云经梯度方向迭代移动后,过滤噪音和剔除离群点,并修补点云缺失数据。点云被自适应筛选后,利用圆球相交的方法生成新的三角形。实验表明,该算法具有快速、稳定可靠和内存消耗小的优点。

关键词 逆向工程 傅里叶变换 网格化 降噪
中图法分类号: TP391.9 **文献标识码**: A **文章编号**: 1006-8961(2009)01-0148-07

Adaptive Mesh Reconstruction of Point Cloud with Feature Preserved

QIAN Gui-ping, TONG Ruo-feng, PENG Wen, DONG Jin-xiang
(*College of Computer Science and Technology, Zhejiang University, Hangzhou* 310027)

Abstract There is noise and defective data on the 3D scanning point cloud. A robust mesh reconstruction algorithm is proposed. Surface normals are estimated by tensor matrix with enhanced features. By computing 3D fast Fourier transform (FFT), discrete iso-surface is extracted. Points are moved onto the iso-surface by an iterative clustering along gradient field, where the noise and outliers are removed and defective data are repaired. Point cloud is decimated adaptively, and then a new triangle is generated using sphere-intersected method. The experimental results have shown that the algorithm is fast, robust and use low memory.

Keywords reverse engineering, Fourier transform, meshing, denoising

1 引 言

在逆向工程中,对客观世界中的物体扫描,得到离散的3维采样点,进而进行曲面重建是其中的一个重要研究内容。当物体被各个方向扫描后,得到多张不完整的离散点云采样曲面。经过叠加配准,形成带有噪音,离群点和缺失数据的数字化点云模型。如何寻找一种快速可靠的方法对这个数字化点云模型进行曲面重建,是当前的研究热点。

对散乱点云进行重建,通常分为两种,点模型和网格模型。从没有方向信息的3维坐标点集,推测出曲面法向,0等值面或网格曲面,并非易事。主要面临如下几个关键问题:

(1)有效过滤采样噪音,并剔除离群点;

(2)有效降低内存消耗;

(3)能够快速重建;

(4)逼近真实物体形状。

造成这些困难的原因主要是扫描客观物体后,由于机器探头的精度和各种电子干扰等因素,产生噪音和较多离群点。同时得到的几十万个,几百万个甚至上亿个点,造成曲面重建计算效率降低,内存

基金项目:国家重点基础研究发展计划(973)项目(2006CB303106);国家科技部科技支撑计划项目(2007BAH11B04);浙江省科技计划(2007C223050)
收稿日期:2007-05-29;**改回日期**:2007-08-14

第一作者简介:钱归平(1975~),男。浙江传媒学院讲师。2008年于浙江大学计算机科学与技术学院获博士学位。主要研究方向为计算机图形学、3维CAD造型、曲面重建。E-mail:qianguiping@163.com

消耗过高。近年来,产生了较多的点云网格模型重建方法。这些点云网格重建算法主要归纳为 2 个方向:

(1)基于计算几何的方法,直接在稠密的点云上产生三角网格。这些方法以 Power Crust[1]算法和 Tight Cocone[2]为代表,直接利用 Voronoi 图来进行 Delaunay 三角化。这些方法的主要缺点就是计算量偏大,而且产生的三角网格比较粗糙,无法有效抵制稍大的噪音,并且难以填补空洞。因而常常需要更进一步的处理。

(2)曲面拟合方法,这是应用最广泛的方法。通过构造一个隐式曲面方程对点云进行拟合,从而最佳逼近真实物体表面。如 Ohtake 等人的 MPU 方法[3]和圆球相交方法[4],这两种方法都是构造 2 次误差函数来进行拟合实现的,其主要缺点就是无法剔除离群点,稍大的噪音就导致重建失败。移动最小二乘(moving least square)曲面[5-7]也是使用较多的方法。文献[5]提出了有效剔除离群点的方法,能够光滑点云,同时保持物体的尖锐特征,但是最小二乘中值法计算量相当大,效率低下。Kazhdan[8]巧妙利用 Stokes 定理,将体积分转换成曲面积分,利用快速傅里叶变换(FFT)来提取物体表面的 0 等值面。这种方法有一个主要缺点,当需要表示物体细节特征时,过大的内存需求导致在一般的计算机上难以实现。后来他又提出了泊松曲面重建[9]的算法,但是算法效率明显降低。

本文提出的方法不但能够有效降低内存消耗,而且执行效率相当高,弥补 FFT 方法需要描述物体细节特征时,过高的分辨率引起超大的内存消耗。可以在较低分辨率和很少内存消耗时,充分保持物体的尖锐细节特征,并且计算速度有了明显提高。对于点云空洞区域,能够进行充分的采样,形成封闭的网格模型。由于 Ohtake[4]的三角网格化方法,对噪音特别敏感,在稍大的噪音或者存在离群点时,就会导致重建失败或者网格非流形,并且对点云空洞只能实现简单的平面封闭。本文的算法由于有效克服了噪音和离群点的影响,从而使用圆球相交方法进行网格化时,能够很好地保证网格流形,同时对点云空洞的填补做了有效的改进。

2　法向提取

通常扫描得到的经过配准的点云,没有法向信息,张量选举法能够利用空间点之间的位置关系获得法向。在形成尖锐特征的地方,进行增强特征的法向扩散处理,以消除法向在边或角处的光滑过渡。

2.1　初始法向

首先对点云 $\boldsymbol{P}=\{\boldsymbol{p}_i\in R^3\}$, $i=1,2,\cdots,M$。建立正方体包围盒,在这个包围盒内部建立 KD 树。对这个正方体进行均匀剖分(分辨率为 B)得到 B 个小立方体,称为分辨率格子。对于每个点 $\boldsymbol{p}_i$,计算周围包含不小于 $N_{\min}$(一般设定为 15)个点的包围圆球半径 R_i。在包围圆球内,设定 $\boldsymbol{p}_i$ 相关 $\boldsymbol{p}_j$ 的单位曲面切向分量为 $\boldsymbol{t}_j=(t_{jx},\ t_{jy},\ t_{jz})^{\mathrm{T}}$。$\boldsymbol{t}_j=(\boldsymbol{p}_j-\boldsymbol{p}_i)/\|\boldsymbol{p}_j-\boldsymbol{p}_i\|$。建立 $\boldsymbol{p}_i$ 的曲面切向张量矩阵[10]为

$$\boldsymbol{T}_i=\sum_{j\in SPT(\boldsymbol{p}_i)}\begin{bmatrix}t_{jx}^2 & t_{jx}t_{jy} & t_{jx}t_{jz}\\ t_{jy}t_{jx} & t_{jy}^2 & t_{jy}t_{jz}\\ t_{jz}t_{jx} & t_{jz}t_{jy} & t_{jz}^2\end{bmatrix} \tag{1}$$

其中,$SPT(\boldsymbol{p}_i)$为 $\boldsymbol{p}_i$ 邻近的最近 $N_{\min}$ 个点,$\boldsymbol{T}_i$ 是 3 × 3 的正定实对称矩阵,在 3 维空间中可以直观地被看作一个椭球体。$\boldsymbol{T}_i$ 张量的 3 个特征值为 $\lambda_1>\lambda_2>\lambda_3$,最小特征值 λ_3 所对应的特征向量 $\boldsymbol{e}_3=(e_x,\ e_y,\ e_z)^{\mathrm{T}}$,可以被看作曲面在 $\boldsymbol{p}_i$ 的初始法向 $\boldsymbol{n}_i$(图 1)。

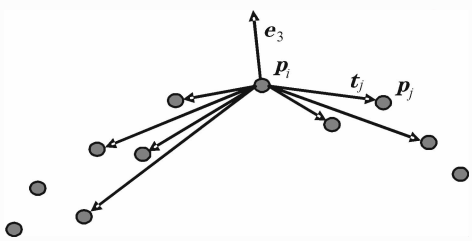


图 1　张量构造

Fig. 1　Tensor constructed

2.2　法向扩散

由于张量构造形成的法向光滑过渡,在形成尖锐特征的地方,需要进行增强特征的法向扩散,使用如下 4 步解决:

(1)查找 $\boldsymbol{n}_i$ 的邻近点法向,如果 $\|\boldsymbol{n}_i\cdot\boldsymbol{n}_j\|\geqslant 0.9$,则 $\delta_{ij}=1$,否则 $\delta_{ij}=0$。

(2)计算局部尖锐特征度量 ω_i,对于 $SPT(\boldsymbol{p}_i)$中的点,

$$\omega_i=\frac{1}{\|SPT(\boldsymbol{p}_i)\|}\sum_{j\in SPT(\boldsymbol{p}_i)}\alpha^2(\boldsymbol{n}_i-\boldsymbol{n}_j) \tag{2}$$

式中,$\alpha(\boldsymbol{n}_i-\boldsymbol{n}_j)$为取两法向夹角(锐角)的弧度函数值,$\|SPT(\boldsymbol{p}_i)\|$为包含的邻近点数。尖锐特征

度量值 ω_i 取值范围 0 ~ π/2,在法向过渡变化大的区域,如边或角,取得较大的值,而在平坦区域取相对较小值。

(3)计算新的法向

$$\boldsymbol{n}_i' = (1 - G(\omega_i))\boldsymbol{n}_i + G(\omega_i)\frac{\sum_{j\in SPT(\boldsymbol{p}_i)}\delta_{ij}\boldsymbol{n}_j}{\left\|\sum_{j\in SPT(p_i)}\delta_{ij}\boldsymbol{n}_j\right\|} \tag{3}$$

式中,$G(\omega_i)$ 是用户定义的权值函数,$G(x) = (1-2x/\pi)^3$。

(4)使用式(3)对每一个法向进行更新,重复进行迭代,直至满足条件 $\|\boldsymbol{n}_i \cdot \boldsymbol{n}_i'\| < \varepsilon$。根据实验效果,通常 ε 取值 0.02 ~ 0.05,太小取值会使得迭代次数过多而降低效率。

3 点云滤波

对于散乱点云的曲面重建,最重要的一点是如何快速有效地降噪和剔除离群点。Kazhdan[8] 利用 Stokes 定理,将体积分转化为曲面积分。有向点云则作为体边界的采样点来逼近曲面积分。通过这种方法,将点云在频域中进行高斯滤波,然后在时域中进行进一步处理。

3.1 频域高斯滤波

为了提高计算效率,采用较低的分辨率 B 来对包围立方体进行剖分,如 $B = 128^3$ 或 $B = 256^3$,将每个点到分辨率格子 8 个顶点的距离看作图像灰度。如果格子内有较多的点,则进行灰度叠加,同时相应的法向量像几何向量和,并归一化。这样处理后,将散乱点云转化为 3 维灰度图像处理。采用 Kazhdan[8] 的特征函数

$$\chi_V(\boldsymbol{p}_i) = \begin{cases} >0 & \boldsymbol{p}_i \in \boldsymbol{V} \\ <0 & \text{其他} \end{cases} \tag{4}$$

式中,$\boldsymbol{V}$ 表示点云所代表的物体。将点云所在的时域空间转化到频域空间。由于高斯滤波方法能够过滤频域中的噪音,因此在频域中将 FFT 变换和高斯函数做卷积,可以得到频域的特征函数 χ_V。然后逆 FFT 变换转换到时域空间。通过该方法,获得离散的粗糙 0 等值面。

3.2 时域滤波

原始点云分布在离散等值面 $\chi_V = 0$ 周围,由于分布不均匀,而且可能存在大量离群点,所以必须进行清理和移动。首先计算点云 $\boldsymbol{P}$ 中每个点的特征函数值的绝对值平均值,记为 $\bar{a}$。如果 $|\chi_V(\boldsymbol{p}_i)| > 3\bar{a}$,那么 $\boldsymbol{p}_i$ 判定为离群点,在 $\boldsymbol{P}$ 中删除离群点。如果 $|\chi_V(\boldsymbol{p}_i)| \leqslant 3\bar{a}$,则 $\boldsymbol{p}_i$ 初步判定为曲面上的点(含噪音)。

在 level set[11] 思想中,将重构表面看成 3 维向量空间的可变形封闭曲面,在某种力量的作用下,逐步逼近目标模型。由于已知粗糙离散 0 等值面,将 level set 方法扩展,使散乱点在自身特性及其梯度场的作用下,逼近离散 0 等值面。考虑隐式曲面 $\chi_V(\boldsymbol{p}_i)$,点在梯度场力 F 的作用下,沿着梯度法向 N 移动

$$\frac{\partial \chi_V(\boldsymbol{p}_i)}{\partial t} = FN \tag{5}$$

梯度场力 F 应当与采样点至 0 等值面的距离成正比。F 则定义为

$$F = \frac{\chi_V(\boldsymbol{p}_i)}{\bar{\alpha}} \cdot \frac{L}{5\sqrt[3]{B}} \tag{6}$$

式中,L 为包围立方体的对角线长,B 为剖分分辨率。整个噪音点云沿法向迭代移动的过程可以表示为

$$\boldsymbol{p}_i^{k+1} = \boldsymbol{p}_i^k + \frac{\chi_V(p_i)}{\bar{\alpha}} \cdot \frac{L}{5\sqrt[3]{B}}\boldsymbol{n}_i \tag{7}$$

迭代过程直到 $|\chi_V(\boldsymbol{p}_i)| < 0.7\bar{a}$ 停止。经过迭代处理,剔除离群点的点云就移动到特征函数 0 等值面附近。

4 网格可视化

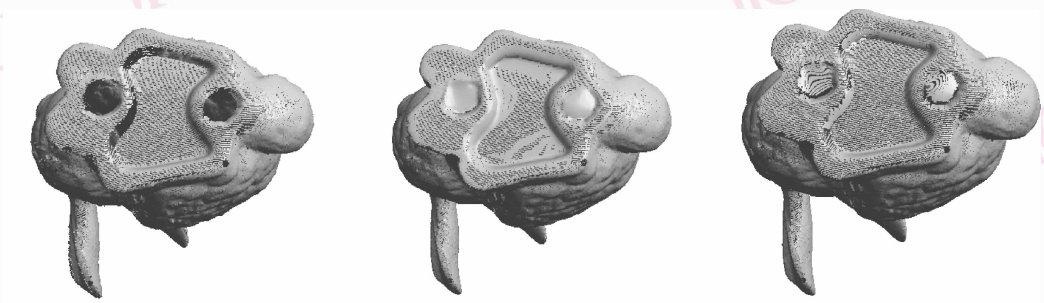
经过滤波后的点云,已经完全逼近物体表面,噪音和离群点得到有效抑制。虽然有了特征函数,但是如果使用常用的隐式曲面网格提取算法,如 Marching Cubes 或 Bloomenthal's polygonizer 方法,则必须和 FFT 变换包围立方体的分辨率保持一致,否则无法正确提取网格。因而本文采用点云的局部拟合网格化方法[3-4] 思想。为了更好地处理点云采样不足引起的网格分布异样,必须对滤波后的点云进行有效的修补。

4.1 点云修补

在点云频域 FFT 变换前,事先建立并保持一个空洞检测数组,数组长度和分辨率格子的数目一致,即 B 个。每个数组元素记录对应分辨率格子内是否包含有原始采样点。当采样分布不均匀或者分布

稀疏时,含有点的格子周围相邻的 26 个立方体也同时标记为包含有点。这样做的目的是为了避免在检测特征函数 0 等值面是否和分辨率格子相交时,过多采样。过采样的例子见图 2(b),过采样通常会造成计算效率显著降低。

当等值面和分辨率格子相交,同时该格子对应的空洞检测数组元素为非,则对分辨率格子进行细分,通常取分辨率为 $b=5^3$ 或 10^3,这取决于点云的密度。计算这 b 个顶点的特征函数值 χ_V,如果 $\chi_V < \zeta\bar{a}$(通常 ζ 取值为 0.05 ~ 0.1),则该顶点被看作曲面的空洞采样点,加入点云 P 中并计算法向。图 2(c)是合适采样的结果,相比原始 Bunny 模型的 361 000个点,增加了 21 000 个采样点。而图 2(b)则增加了 190 000 个点。



(a) 原始Bunny扫描点云　　(b) 采样点过多　　(c) 采样点合适分布

图 2　空洞采样的不同情况

Fig. 2　Different samples in hole

4.2　自适应三角网格化

基于 Delaunay 的算法和绝大部分基于函数拟合的算法都无法很好地应用于噪音数据(含离群点)。经过本文滤波处理后的点云,能够很好地应用于上述算法中,在后面的实验部分,我们将给出一些实验说明比较。

由于张量选举计算法向时,同时也体现了点云的分布特性。在式(1)中,$\boldsymbol{T}_i$ 张量分解的 3 个特征值为 $\lambda_1>\lambda_2>\lambda_3>0$。这 3 个特征值可以作为计算自适应三角网格的判断标准。

使用 Ohtake[4] 的自适应圆球相交网格化的思想来三角化。定义局部二次误差函数为

$$Er_k=\sum_{i\in SPT(\boldsymbol{p}_k)}\left(\frac{\lambda_{1i}-\lambda_{3i}}{\lambda_{1i}}\boldsymbol{n}_i(\boldsymbol{p}_i-\boldsymbol{p}_k)\right)^2 \quad (8)$$

则建立点云子采样的自适应三角网格如下:

(1)计算每个点的二次函数误差 sqrt(Er_k)/L 作为物体的曲率度量进行累加,按照设定的曲率度量误差 $\Delta=Er$,得到每个局部点云的包围球。

(2)按照曲面 splatting 方法,选择包围球的最优子集[12],该子集覆盖整个原始点云。

(3)对于子集中的每个包围球,使用 min(Er_k) 来提取圆球内的三角网格顶点,组成筛选顶点集。

(4)根据包含这些顶点的包围球的相交关系,3 个球两两相交,则包围球中的 3 个顶点构成三角形。

在这圆球覆盖相交过程中,较小的 Δ 意味着更小的圆球覆盖,从而产生更小的三角网格和更高的网格化分辨率。通过圆球的相交关系将这些顶点相连形成三角形后,得到初步的网格拓扑图。按照流形网格的标准,对冗余网格进行清理,对于只有一个三角形相邻的边,则搜寻闭环边界,填补微小空洞。

5　实验分析和比较

运用本文的方法对扫描散乱点云作自适应网格重建,算法保证整个网格是封闭的。整个算法使用 Visual C ++ 实现,所有点云重建在带有 1 G 内存的 Pentium4 2.8 PC 机上运行。表 1 是对不同的扫描模型进行滤波、网格化所需的计算时间比较。表 2 是对由 10 张扫描图合成的 361 000 点 Bunny 模型使用不同算法进行的计算时间和内存消耗比较。同目前公开发表的散乱点云网格化论文中的方法相比,本文算法的速度仅次于 FFT[8] 方法中使用低分辨率 256^3 计算时所需的时间,但消耗内存相当有限。对 1 800 000 的 Bimba 扫描模型进行网格化生

成不同分辨率的三角面片,最大内存消耗小于 350 M。

表 1 不同扫描模型的网格化计算时间

Tab. 1 Computational time measurements of different range scanning models

点云名称	点数(10^3)	滤波后(10^3)	三角数(10^3)	重建时间(s)
Bunny scans	361	346	87.7	15.5
Bunny scans	361	346	218	19.3
Carter scans	546	461	69.8	18.4
Carter scans	546	461	129.5	20.7
Dragon scans	1 406	1 332	141	34.5
Dragon scans	1 406	1 332	272	39.1
Bimba scans	1 873	1 833	130	37.8
Bimba scans	1 873	1 833	255	44.9

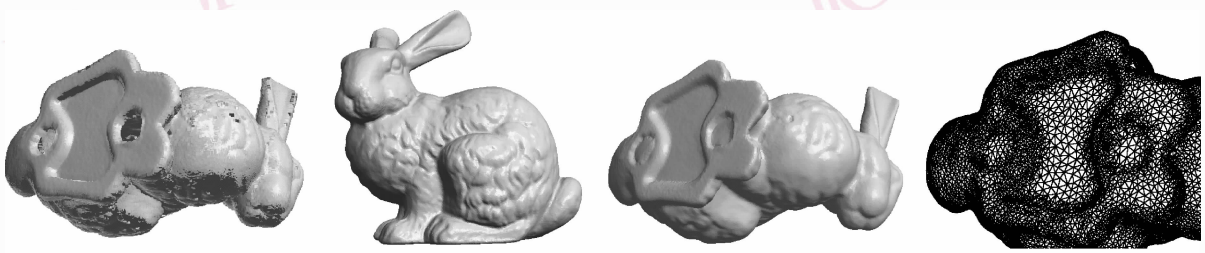
表 2 不同网格化算法的计算时间和内存消耗比较

Tab. 2 Computational time and memory consumption comparison of different mesh algorithm

不同方法	三角形数(10^3)	内存消耗(K)	计算时间(s)
本文方法	188	210	18
本文方法	374	230	26
FFT[8]	167	200	11
FFT[8]	920	1 600	1 350
Ohtake[4]	187	90	31
MPU[3]	185	110	47
IPD[13]	188	120	83
TightCocone[2]	272	570	1 239

在运行时间上,Kazhdan 的 FFT[8] 方法是目前公认的最快的散乱点重建方法,但是对于图形数据的网格重建,要得到合适的效果,仅仅提供 2 ~ 3 种有限的网格分辨率(res),通常为 256^3 或者 512^3。对于大的点云(点数超过 1 000 000 以上),需要精细的模型时,利用 512^3 的分辨率重建则需要 1 000 s 以上的时间和不少于 1.6 G 的内存,内存消耗极大(由于消耗内存太大,导致使用了较多虚拟内存,严重降低了效率。如果内存加到 2 G 以上,则运算时间可以减到 3 min 左右)。

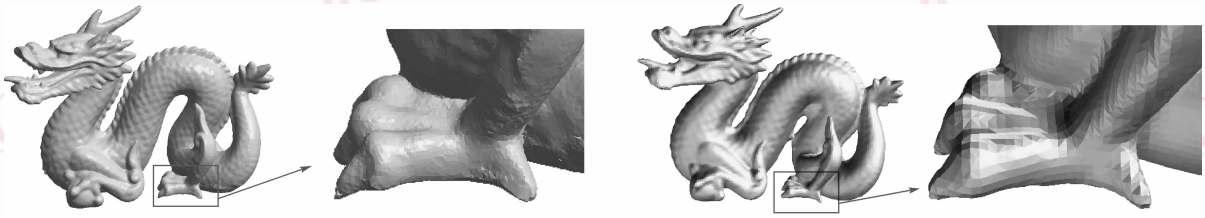
图 3,图 4(a),图 5(b)、图 5(c)和图 6(a)是对不同模型使用本文算法实现的结果。图 3 为曲面曲率度量误差 $\Delta = 0.000\ 2$ 生成的 Bunny 网格模型,三角面片 87 000。其中图 3(a)是 10 张扫描数据合成的 Bunny 原始点云模型,底部有 3 个较大的空洞。图3(b)、图 3(c)是本文的算法生成的网格模型,从图 3(c)中可以看出,3 个空洞得到了很好的修补。图 3(d)是模型底部的部分网格拓扑图,底部的空洞也相应进行了自适应的网格化。图 4 是本文方法和 FFT[8] 的点云网格化方法的比较,很明显,本文的算法效果非常好。特别在图 4(b)中,FFT[8] 方法过多地对物体的细节特征进行了光滑,同时在龙体部相互靠近的地方,产生了形状的扭曲和粘连。图 5 是和 Ohtake[4] 的网格化方法比较,从图 5(b)和图 5(d)比较可以看出,本文方法在过滤离群点方面效果明显优于 Ohtake 的方法,从而使得网格化效果极佳。图 6 使用 Bimba 模型来和 FFT[8] 方法的不同分辨率进行比较。本文方法明显优于 FFT 方法使用 256^3 的分辨率得到的效果,在保持细节特征上,使用较少的网格就可以和 512^3 的 FFT 网格化方法效果相当,而且细节特征保留的更好,在运行时间上,也体现了高效率。



(a) 原始点云图 (b) 网格模型正面 (c) 网格模型底部 (d) 网格拓扑图

图 3 Bunny 模型网格化

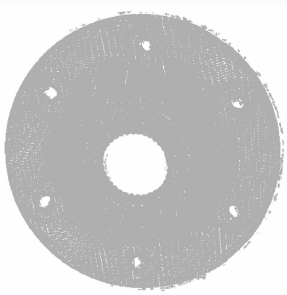
Fig. 3 Meshing Bunny model



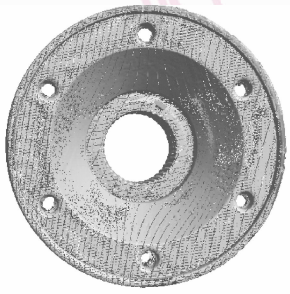
(a) 本文方法，Δ=0.000 2　　(b) FFT方法，$res=256^3$

图 4　和 FFT[8] 的方法对比

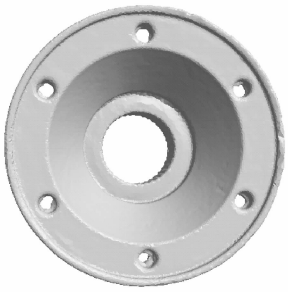
Fig. 4　Comparison to method of FFT[8]



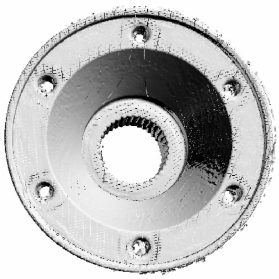
(a) 原始Carter扫描模型



(b) 过滤噪音和离群点后的点云图



(c) Δ=0.000 2生成的69.8×10^3模型

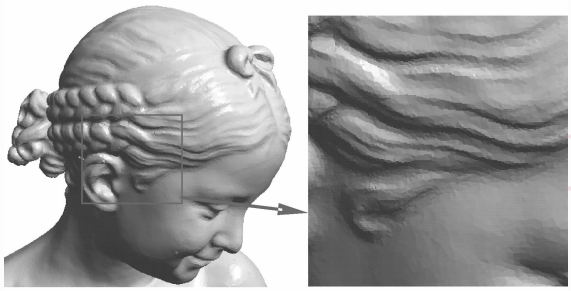


(d) Ohtake方法进行类似mean-shift迭代滤波后的点云图

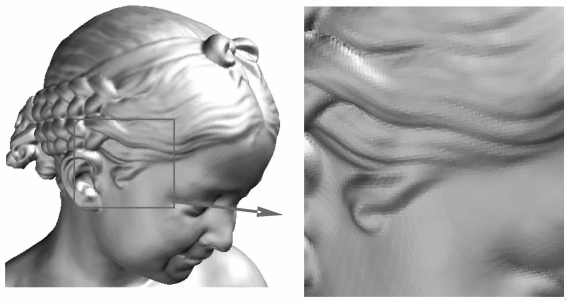
(e) Ohtake方法,Terr=1.5e–6生成68.9×10^3模型

图 5　和 Ohtake[4] 的圆球相交法对比

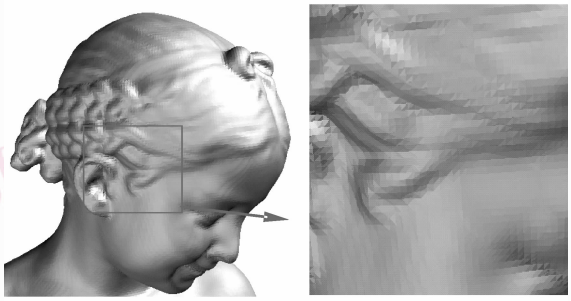
Fig. 5　Comparison to method of Ohtake' sphere-intersected[4]



(a) 本文方法，Δ=0.000 7, 三角形342000



(b) FFT[8]方法,$res=512^3$,三角形937 000



(c) FFT[8]方法,$res=256^3$,三角形231000

图 6　Bimba 扫描模型的网格化

Fig. 6　Meshing Bimba model

6 结 论

本文实现了一种新的散乱点云网格化算法。算法主要利用频域和时域滤波方法过滤噪音和剔除离群点,采用一种新的非 Delaunay 三角网格化方法来实现点云重建。对于点云空洞区域,进行充分的采样形成封闭模型,由于 FFT 变换无法提取尖锐特征,因而本文的空洞填补只能适用在平坦或者光滑过渡的区域。对于模型尖锐特征的修补,则是今后进一步深入研究的方向。

实验表明,算法能够非常稳定可靠地实现点云重建,能够有效降低散乱点云的噪音,并且消耗内存相当低。该算法能够很好地应用到扫描点云的曲面重建中。

参考文献(References)

1 Amenta N, Choi S, Kolluri R. The power crust[A]. In:Proceedings of 6th ACM Symposium on Solid Modeling[C], Ann Arbor, Michigan:ACM Press, 2001:249-260.

2 Dey T K, Gooswami S. Tight cocone: A water-tight surface reconstructor[A]. In:Proceedings of 8th ACM Symposium on Solid Modeling and Applications[C], Washington: ACM Press, 2003: 127-134.

3 Ohtake Y, Belyaev A, Alexa M, *et al*. Multi-level partition of unity implicits[J]. ACM Transactions on Graphics, 2003, **22**(3): 463-470.

4 Ohtake Y, Belyaev A, Seidel HP. An integrating approach to meshing scattered point data[A]. In: Proceedings of 9th ACM Symposium on Solid Modeling and Applications[C], Massachusetts: ACM Press, 2005: 61-69.

5 Fleishman S, Cohen-or D, Silva C T. Robust moving least-squares fitting with sharp features[A]. In: Proceedings of ACM SIGGRAPH [C], New York: ACM Press, 2005: 544-552.

6 Amenta N, Kil Y J. The domain of a point-set surface[A]. In: Proceedings of Eurographics Workshop on Point-based Graphics[C], Zurich Switzerland, 2004: 139-147.

7 Amenta N, Kil Y J. Defining point set surfaces[J]. ACM Transactions on Graphics, 2004, **23**(3): 264-270.

8 Kazhdan M. Reconstruction of solid models from oriented point sets [A]. In: Proceedings of Eurographics Symposium on Geometry Processing[C], Vienna, Aodili,2005: 73-82.

9 Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction [A]. In: Proceedings of Eurographics Symposium on Geometry Processing[C], Cagliari,Italy,2006: 61-70.

10 Knutsson H. Representing local structure using tensors[A]. In:The 6th Scandinavian Conference on Image Analysis[C],Oulu, Finland, 1989:19-22.

11 Osher S, Sethian J A. Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulation[J]. Journal of Computer Physics, 1988, **79**(1):12-49.

12 Wu J, Kobbelt L P. Optimized sub-sampling of point sets for surface splatting[J]. Computer Graphics Forum, 2004, **23**(3): 643-652.

13 Lin Hong-wei, Tai Chiew-lan, Wang Guo-jin. A mesh reconstruction algorithm driven by an intrinsic property of a point cloud[J]. Computer Aided Design, 2004, **36**(1): 1-9.