

结构光视觉三维点云逐层三角网格化算法

秦绪佳^{1,2} 陈楼衡¹ 谭小俊¹ 郑红波¹ 张美玉^{1,2}

(浙江工业大学计算机科学与技术学院 杭州 310032)¹

(浙江工业大学浙江省可视媒体智能处理技术研究重点实验室 杭州 310032)²

摘要 针对结构光视觉恢复的大规模三维点云的可投影特点,提出一种基于投影网格的底边驱动逐层网格化曲面重建算法。该算法首先将点云投影到一个二维平面上;然后基于点云投影区域建立规则投影网格,并将投影点映射到规则二维投影网格上,建立二维网格点与三维点云间的映射关系;接着对投影网格进行底边驱动的逐层网格化,建立二维三角网格;最后根据二维投影点与三维点的对应关系及二维三角网格拓扑关系获得最终的三维网格曲面。实验结果表明,算法曲面重建速度快,可较好地保持曲面细节特征。

关键词 三维点云,曲面重建,三角网格化,投影网格

中图分类号 TP391 文献标识码 A

Layer by Layer Triangulation Algorithm for 3D Point Clouds from Structured Light Vision

QIN Xu-jia^{1,2} CHEN Lou-heng¹ TAN Xiao-jun¹ ZHENG Hong-bo¹ ZHANG Mei-yu^{1,2}

(School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310032, China)¹

(Key Laboratory of Visual Media Intelligent Processing Technology of Zhejiang Province,

Zhejiang University of Technology, Hangzhou 310032, China)²

Abstract Aiming at the feature that large scale 3D points cloud recovered from structured light vision can be projected, a bottom edge driven layer by layer mesh surface reconstruction algorithm based on projection grid was presented. Firstly, the point cloud is projected to a 2D plane. Then a regular 2D projection grid based on the point cloud projection area is established, and the projected points are mapped to the 2D projection grid points. So the mapping between the 3D points and the 2D projection grid points is established. Thirdly, bottom edge driven layer by layer triangulates the 2D projection grid points, and a two-dimensional triangular mesh is obtained. Finally, according to the correspondences among 2D projection grid points and the 3D points, and the 2D triangular mesh topology, final 3D mesh surface is built. Experimental results show that the proposed algorithm is fast and can be used to maintain the details of the surface.

Keywords 3D point cloud, Surface reconstruction, Triangulation, Projection grid

1 引言

目前,物体三维几何表面信息的获取方式可分为接触式和非接触式两类;接触式有三坐标测量;非接触式有基于光学方法的如光激扫描、结构光法等和基于非光学方法的如 CT、超声波、核磁共振扫描等。基于光学的非接触式方法获取的物体表面信息一般以三维点云形式表达。由于基于光学的方法具有获取速度快、精度高等优点,目前已被广泛应用于物体三维几何测量及表面重建。

三维点云的曲面重建就是由点云构建曲面,通常为三角网格曲面。点云数据的曲面重建方法可分为以下 3 类:1) 基于 Delaunay 三角化的曲面重建。该方法是平面点集三角剖分在高维情形的扩展,是利用 Delaunay 三角化或 Voronoi 图的方法构建三维点云的拓扑结构。Boissonnt 最先提出了点云 Delaunay 三角化方法^[1],该方法虽然获得了物体表面的一

种逼近表达,但存在拓扑不一致问题。1998 年 Amenta 等^[2]提出了基于三维 Voronoi 图的 Crust 算法,较好地解决了重建中拓扑不一致问题。之后, Amenta 等^[3]进一步优化,引入了带权图,可以有效处理有尖锐特征和噪声的模型。Adamay 等^[4]提出了 Delaunay 剖分的 mbrellaFilter 方法, Mederos 等^[5]提出了点云存在噪声时如何进行曲面拓扑重建。上述方法没有用到点云的法向信息,重建的曲面可能出现自相交现象。董辰世等^[6]采用一致性处理之后的点法向信息,有效剔除了三角化中生成的不需要的三角面片,从而避免了曲面的自相交。2) 基于隐式曲面的重建方法。该方法是用一个三维函数来逼近离散点云,然后通过提取函数的零等值面重建曲面。Carr 等^[7]采用径向基函数拟合散乱点云数据,构造光滑曲面。基于径向基函数的方法的优点是通过对最小化薄板能量函数,来得到光滑的曲面;缺点是大规模点云数据构成的径向基函数拟合非常耗时,而且不利于模型的局部编辑。Alexa^[8]

本文受国家自然科学基金(61672462, 61672463),浙江省科技计划项目(2016C33165)资助。

秦绪佳(1968—),男,博士,教授,博士生导师,主要研究方向为计算机图形学、图像处理, E-mail: qxj@zjut.edu.cn; 陈楼衡(1992—),男,硕士生,主要研究方向为计算机图形学; 谭小俊(1986—),男,硕士生,主要研究方向为计算机图形学; 郑红波(1977—),女,博士,讲师,主要研究方向为图像处理; 张美玉(1965—),女,硕士,教授,硕士生导师,主要研究方向为图像处理。

采用移动最小二乘法定义流形曲面来拟合点云数据,通过对数据的重采样来控制逼近误差,但控制参数较难选定。Du^[9]等采用隐式 PDE 进行点云曲面重建,通过简单的距离估计替代径向基函数曲面梯度计算,减小了计算量。Ohtake^[10]提出了一种 MPU 算法,采用局部二次曲面代替径向基函数,并采用八叉树剖分点云确定紧支撑函数的参数,该方法计算速度快,但对噪声不鲁棒。Kazhdan 等^[11]将点云重建问题转化为空间泊松方程求解问题进而得到曲面,获得了较好的重建效果。3) 基于区域增长的重建方法。该类方法的核心思想是先从一个满足初始条件的种子三角形出发,通过某种扩散条件以已有边作为前沿边,逐步加入点云中的其他点,与前沿边形成新的三角形,最终重建出网格曲面。该类方法最具代表性的是 Bernardini^[12]提出的滚球法,该方法是基于 α -shape 理论的,重建结果的好坏取决于滚球半径 p 。Kuo 等^[13]提出了一种 Delaunay 三角化与区域增长结合的方法,由于该方法无法处理复杂的点云以及具有尖锐特征的情况,Kuo^[14]又对该方法进行了改进,提出了一种自适应雕刻算法,改进方法可有效处理具有尖锐特征的情况。

上述 3 类方法中,基于 Delaunay 三角化的重建算法存在计算量大、效率低,且对于噪声大和细节特征明显的物体重建效果差的缺点。基于隐式曲面重建的方法可以重建光滑、连续的网格,但很难找到一个合适的隐式函数来表示一个复杂的曲面,对噪声大、采样不均匀的点云重建效果很差。基于区域增长法原理简单,容易实现,但不易保证构造物体网格表面的拓扑一致性。

近年来,研究者们提出了一些新方法,如采用分而治之的思想将点云曲面先分割后重建^[15,16],提高了重建效率。钱归平等^[17]基于散乱点云内部特征,通过贝叶斯模型的降噪迭代来获得逼近物体的点云,然后通过曲面 Splatting 筛选采样并对采样点三角化来实现点云重建,降低了散乱点云的噪声。聂建辉等^[18]针对高密度散乱点云先构建原始点云的近似逼近三角网格曲面,然后再根据曲面变化情况对三角网格进行二次剖分,提高了曲面重建的速度和噪声点云的处理能力。另外,Huang 等^[19,20]提出了点云重建时如何保持点云的尖锐特征,并将相邻曲面重建出来,获得了较好效果。

本文针对结构光三维重建中采用投影光栅法获得的大规模点云数据研究新的重建方法。由于点云上的三维坐标信息由图片二维信息进行恢复,且由于光是沿直线传播,因此得到的点云具有单面性特征,三维点云中的点与二维照片的像素存在对应关系。本文根据结构光三维重建获得的点云特征,将三维点云的曲面重建问题转化成图片上二维像素点的网格化问题,提出了一种投影网格化法来快速构建点云间拓扑关系,建立点云的网格曲面表示。

2 点云投影网格及其拓扑关系建立

由于投影网格化法需要把点云数据投影到一个特定平面上进行拓扑关系建立。对于同时保存有三维点云及其对应的结构光图像,该图像像素阵列网格即可作为我们需要的投影网格,点云中的点对应的像素点即为投影网格中的二维点,对这些二维点进行网格化即可。

但是,对于只有结构光获取的三维点云数据,要实现光栅重建点云的网格曲面构建,首先需要计算找到进行投影的特定方向,保证投影到特定平面后,各投影点在投影平面上尽量

无重叠。其次,建立平面点拓扑关系需要估算合适的平面网格尺寸,以免网格尺寸太小时由于投影点重叠过多从而删掉过多有用点,而网格尺寸太大则会造成网格上的点稀疏从而出现过多孔洞。下面介绍点云投影方向的估算及投影网格的生成。

2.1 投影方向的估算及投影平面确定

对一片点云数据而言,确定结构光图像获取的视线方向(拍摄方向)是较关键的,如果估计的投影的平面方向与原始光栅照片的拍摄方向相差较大,则投影后点云数据很有可能在投影平面上杂糅在一起,从而导致建立错误的网格化关系图,将影响后续处理。因此,需要确保点云沿投影方向投影到平面上点云重叠度最小,使估算的投影方向尽量接近原始光栅图像的拍摄方向。

对于点云数据集 $S = \{P_i(x_i, y_i, z_i) | i = 1, \dots, n\}$,采用最小二乘法将点云整体拟合到一个平面,该平面的法向即为所需要的投影方向。

(1) 投影方向确定

假设点云拟合平面的单位法矢量为 $\vec{n} = (n_x, n_y, n_z)$,坐标原点到平面的距离为 d ,则点云 S 中任意一点 $P_i(x_i, y_i, z_i)$ 到平面的距离可表示为:

$$D_i = n_x x_i + n_y y_i + n_z z_i + d \quad (1)$$

将点云数据中的所有点用式(1)表示并联合求解 (n_x, n_y, n_z) ,为线性最小二乘问题。求解此最小二乘问题即可得到点云数据集 S 中的投影方向估计。

(2) 建立投影平面及确定投影点集

设原三维点云的全局坐标系为 OXYZ,建立局部坐标系 $oxyz$,局部坐标系原点 o 为三维点云的几何中心, z 的方向为三维点云的法向 \vec{n} , x 的方向为 \vec{n} 与全局坐标系 Z 轴的叉乘, y 的方向为 x 与 z 的叉乘得到。 xoy 平面即为我们所要的投影平面,如图 1 所示。将三维点云投影到局部坐标系的 xoy 平面上,并通过几何变换得到投影点在平面坐标系 xoy 上的二维坐标值 (x, y) 。这样便可得到原始点云 S 投影到 xoy 后的平面点集 $S' = \{Q_i(x_i, y_i) | i = 1, \dots, n\}$ 。

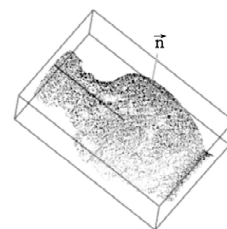


图 1 点云投影方向及局部坐标系

2.2 点云的平面投影网格生成

点云投影到平面后,三维点云数据转化为只包含二维坐标信息的平面二维点数据。此时,三维点云间的拓扑关系尚未建立,但三维点云与二维投影点间存在映射关系,通过对二维点集的网格化来得到三维点云的网格模型,这样将三维点云间的网格化问题转化为二维点集的网格化问题。为获得投影的点云数据的平面网格化拓扑关系,需要建立如图像像素阵列的网格,并将投影点就近平移到网格点上。当有多个点移动到同一个网格点上时,只取距离该网格点最近的投影点,其余投影点删除。为尽量少删除投影点,需确定适合的网格尺寸,使能平移到网格点上的投影点最大化。采用以下 3 步来建立点云的平面投影网格。

(1) 投影网格图尺寸计算

对比所有投影的平面点云数据,通过计算平面坐标包围框得到平面点云数据尺寸为 $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$,则平面上横向和纵向间隔分别为 $d_x = x_{\max} - x_{\min}$, $d_y = y_{\max} - y_{\min}$ 。

接下来需要确定投影网格 x 和 y 方向的间距 L_w 和 L_H ,以确定投影网格的网格规模。考虑到采用光栅投影重构获得的三维点云疏密度比较均匀,点云中的点数 Num 与网格尺寸成正比。取 $t = \lambda \cdot \sqrt{Num}$, λ 为原始光栅照片的纵横比,通常取 $4/3$ 。然后根据 x_{mid} 和 y_{mid} 的大小关系来确定 L_w 和 L_H 的值。如果 $x_{mid} > y_{mid}$,取 $L_H = [t]$, $L_w = [L_H \cdot \lambda]$;否则, $L_w = [t]$, $L_H = [L_w \cdot \lambda]$ ($[\cdot]$ 表示取整)。

(2) 投影网格点数据的插入

计算点云数据的映射到投影网格的缩放比。其中,宽度方向的缩放比为 $m_x = (L_w - 1)/d_x$,高度方向的缩放比为 $m_y = (L_H - 1)/d_y$ 。在网格点中插入已投影到二维平面点云数据的点,任意点 $Q_i(x_i, y_i)$,其插入到投影网格点的位置坐标为 $E_i(u_i, v_i)$,则 $u_i = [(x_i - x_{\min}) \cdot m_x]$, $v_i = [(y_i - y_{\min}) \cdot m_y]$ 。

由于2.1节估算出的投影方向可能存在微小的偏差,使得二维点云插入网格点时可能存在位置重叠的数据点。本文采用后插入为准的原则,即网格点上所在的位置以后插入的点数据为准,先插入的点数据将被覆盖。插入了数据的网格点为有值点,其余网格点为空值点。所有有值的网格点与投影二维点云之间存在对应关系,因而与三维点云间也存在映射关系。

待所有二维投影点插入到投影网格后,需要进一步调整网格尺寸。遍历并累加投影网格整行空值点的行数 t_w 和整列空值点的列数 t_H 。去掉这些空值点的行和列,投影网格尺寸更新为 $L_w = L_w - t_w$, $L_H = L_H - t_H$ 。这样,初始投影网格就建立了,去除投影重叠点后,投影网格的有值点数减小为 Num' 。

(3) 投影网格图优化

采用二值图像边界像素的方法来定义投影网格上的边界点;对投影网格图上的一个数据点而言,如果其周围8个点中,有一个点是空值点,则该点为边界点。然后计算出插入到投影网格上的点云数据点数 Num' 和边界点数 N_b 。如果满足关系 $N_b > \mu \cdot Num'$,则说明边界点数过多,重建的网格曲面存在过多孔洞。进一步缩小网格尺寸, $L_w = [L_w/\eta]$, $L_H = [L_H/\eta]$,重新返回到上述第(2)步进行网格点插入。这里 μ 取 0.5 , η 取 1.2 ,可根据实际情况自行调整取值。

通过上面3步建立的投影网格,建立了网格点与三维点云间的映射关系。图2为瓷瓶三维点云与投影网格。由于点云数据太过密集,这里按行列各间隔100个网格点显示映射关系。

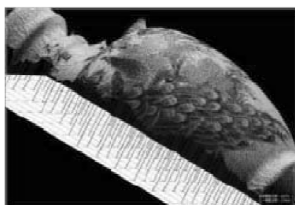


图2 三维点云投影网格化关系图

3 投影网格的逐层网格化重建算法

采用第2节方法建立的投影网格,90%以上的三维点云

会与投影网格有值点存在对应关系。下面的网格生成是基于二维投影网格进行的,对网格有值点生成三角网格,根据二维和三维的映射关系,从而获得三维点云的网格曲面。没有映射到投影网格上的三维点,则不参与三维网格的重建。

3.1 底边驱动的单层网格建立方法

对于二维投影网格图中一层点的网格建立,首先定义一层投影网格中的底边和顶边,下边顶点为 A ,上边顶点为 B ,如图3中下边界 $A_1A_2A_3A_4$ 所在的边为底边,则上边界 $B_1B_2B_3B_4$ 为顶边。通过底边上的网格点来驱动该层的网格建立。

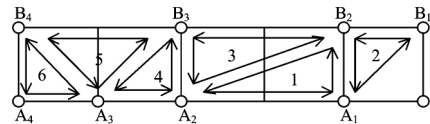


图3 底边驱动的网格构建

以图3所示的一层投影网格点为例说明底边驱动的网格建立算法过程如下。

(1)首先找到底边上的第一个数据点 A_1 ,然后在顶边中找到与 A_1 最近的数据点 B_2 ,结合底边 A_1A_2 建立该层的第一个三角面片 $A_1B_2A_2$ 。

(2)找到顶层中在 B_2 之前的数据点,这里为 B_1 ,依次连接并与 A_1 组建成三角面片 $B_2B_1A_1$ 。如果 B_1 之前还有数据点,依此构造新三角面片,直到找不到未处理的数据点为止。

(3)在顶层找到与底层第二个数据点 A_2 最近的数据点 B_2 。依次连接成三角面片 $A_1B_2A_2$ 。

(4)从 A_2 开始按步骤(2)、(3)的方法顺序建立起三角面片。对于在顶层搜索到距离相同的数据点时,为控制三角面片形状的最优,选择顶边上的位置靠后的数据点。如图3中, A_3 的最近点为 B_1 ,而不是 B_3 。

对于模型网格曲面的建立,需要考虑所建三角面片法向的统一性问题。采用光照条件下绘制,曲面模型上的三角面片法向不统一,曲面会形成很多“破洞”,影响模型的观察和实用性。

由于投影光栅法获得的三维点云模型具有单面特性,因此所建模型的三角面片法向总体朝外。在基于投影网格曲面重建中,网格点的三角化边连接采用逆时针方向,获得法向统一的三角面片。

对投影网格逐层进行三角化,得到整个投影网格的三角化网格,从而建立三维点云的网格曲面模型。

3.2 粒度可控的网格模型重建

逐层网格重建法是一种新的网格曲面重建方法,可以通过控制对投影网格行和列的采样密度来调节重建网格模型的网格精度。因此该方法在网格重建的同时,融入了网格简化的过程,通过选择适当的网格粒度 G_z 来实现对重建网格模型的精细程度控制。

定义重建网格的简化程度 L_s 和网格粒度 G_z 如下:

$$L_s = \frac{1}{G_z \cdot G_z} = \frac{1}{G_z^2} \quad (2)$$

G_z 为投影网格的行和列采样间隔。网格粒度 $G_z = 1$ 表示网格行和列采样间隔为1, $L_s = 1$,即网格重建无简化,如图4(a)所示; $G_z = 3$ 时, $L_s = 1/9$,重建网格模型简化为原来的 $1/9$ 。 $G_z = 10$ 和 $G_z = 60$ 时重建的网格模型分别如图4(b)和图4(c)所示。

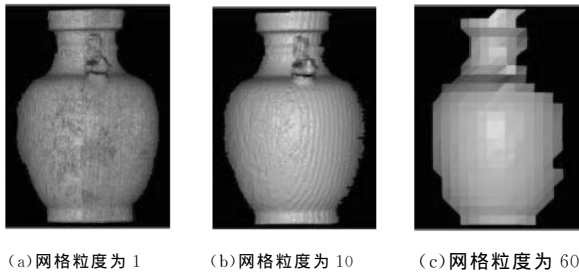


图4 不同粒度的网格曲面

3.3 底边驱动的逐层网格重建算法

上面建立了三维点云到二维投影网格点间的映射、单层投影网格点间的三角化方法,以及重建网格粒度控制方法。下面给出基于投影网格逐层网格化的三维点云网格重建算法的具体步骤。

- (1)按第2节方法建立三维点云到二维投影网格点间的映射;
- (2)根据三维点云规模及重建网格模型的具体需要,确定网格粒度大小 G_z ;
- (3)根据网格粒度大小 G_z 重采样二维投影网格,建立重采样网格点与三维点间的映射;
- (4)取出二维投影网格中最底层和次底层两行网格点,采用3.1节方法建立三角网格;
- (5)以生成的三角网格上边界为底边,逐层提取上一层网格点,建立三角网格,直到最上一层被三角化为止;
- (6)根据二维网格点与三维点间的对应关系,由二维网格点的三角面片连接关系得到三维网格模型表达。

图5为花瓶三维点云采用上述底边驱动的逐层网格重建算法生成的网格模型,其中图5(a)为三维网格重建模型,图5(b)为瓷器花瓶重建模型的局部网格细节放大图。

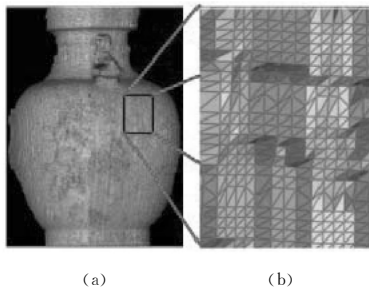


图5 瓷器花瓶模型网格曲面图

3.4 重建模型的细长面剔除

由于三维点云存在噪声、个别区域点云密度不均匀,采用上述算法生成的网格模型会出现形态较差的细长三角面片。通常采用三角形的周长与面积比来定义三角形的形态,当周长面积比大于设定值时定义为细长三角形。但是,采用结构光重建的三维点云规模往往较大,导致重建的网格模型三角面片数很大,一般会达到几十万到数百万。用周长面积比确定细长面需要计算每个三角面片的周长和面积,计算费时。

为简化计算,设三角形3条边长分别为 $l_{\min}, l_{mid}, l_{\max}$,其中 $l_{\min} \leq l_{mid} \leq l_{\max}$ 。计算最长边与最短边的比值 D :

$$D = \frac{l_{\max}}{l_{\min}} \quad (3)$$

判定细长三角面片的标准为 $D > \lambda$,本文中 λ 取6。

4 实验结果及分析

本文中的实验在PC机(Intel(R) Core(TM)2 Duo CPU 2.80GHz,内存2.00GB)上采用VC++2008编程实现的。相关实验及分析如下。

4.1 点云的投影网格生成

对由投影结构光三维重建的三维点投影网格化生成进行了实验,表1给出了点云规模从50万至200万个点的4个模型进行投影网格化的耗时。从表中可以看出,对于大规模点云数据建立投影网格的时间非常快。

表1 点云投影网格化时间像的性质

模型	模型点数	投影网格化耗时(s)
青铜马	504998	2.156
方形花瓶	720931	2.031
瓷器花瓶	989176	3.047
兵马俑	2265899	6.682

建立三维点云的投影网格关系后,为了更直观地展现点云数据间的拓扑关系,将投影网格关系以图像的形式来呈现。图像尺寸由网格尺寸决定,为 $L_w \times L_h$,图像上的像素值由点云数据与投影平面的高度差决定,即点云数据中的 Z 值确定:

$$\text{像素值} = (\text{点的 } Z \text{ 值} - \text{点云最小 } Z \text{ 值}) \times \text{缩放比}$$

其中,缩放比 = $255 / (\text{点云最大 } Z \text{ 值} - \text{点云最小 } Z \text{ 值})$ 。

图6为瓷器花瓶和青铜马两点云模型生成的投影网格关系图,其中左边的为带颜色的点云模型,右边的是其对应点云数据投影网格以后生成的网格化关系图的图像表示,图像上颜色越深的地方表示其距投影平面越远,是相对突出的地方。

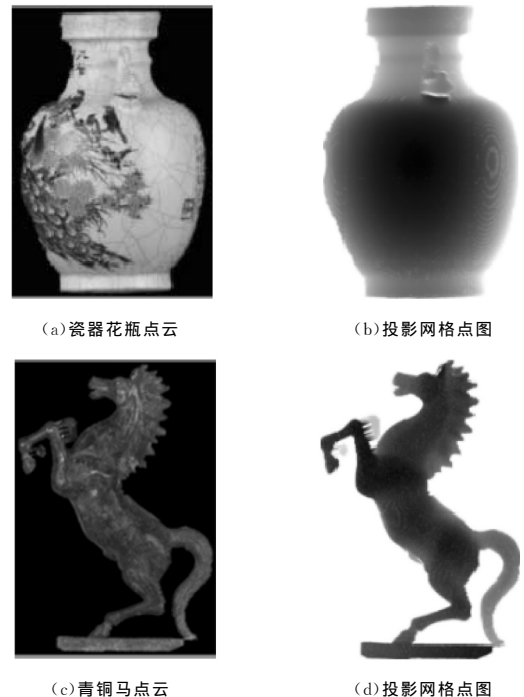


图6 点云模型及投影网格点图像表示

4.2 点云的网格重建

图7分别为本文网格重建法与商业逆向工程软件Geomagic Studio 12对方形花瓶、青铜马、瓷器花瓶、兵马俑的重建效果图,对比软件环境为Geomagic Studio 12,其中左边4幅图片为Geomagic Studio 12重建效果,右边4幅为本文方法处理效果。



(a) Geomagic 重建结果



(b) 本文重建结果



(c) Geomagic 重建结果



(d) 本文重建结果



(e) Geomagic 重建结果



(f) 本文重建结果



(g) Geomagic 重建结果



(h) 本文重建结果

图7 Geomagic 软件与本文方法重建效果比较

从图7对比可以看出,Geomagic Studio 12对网格中的小孔洞并未进行填补,而本文方法较好地填补了模型中较小的孔洞。

下面从生成的面片数和时间进行对比,分别如表2和表3所列。由表中数据可以看出,本文方法所建立的面片数比Geomagic Studio 12建立的面片数多,多出来的便是修补孔洞所用的面片,而且重建时间是随点数增加而线性增加的,且要远远快于Geomagic Studio 12的曲面重建速度。

表2 网格曲面重建三角面片数对比

模型	点数	本文逐层网格建立 (面片数)	本文坏面剔除后 (面片数)	Geomagic 建立 (面片数)
青铜马	504998	1006919	1002681	998933
方形花瓶	720931	1438383	1438031	1432084
瓷器花瓶	989176	1975387	1970878	1946996
兵马俑	2265899	4525665	4519370	4482392

表3 网格曲面重建时间对比

模型	本文投影 网格化(s)	本文逐层 网格重建(s)	本文坏面 剔除(s)	总时间 (s)	Geomagic (s)
青铜马	2.156	0.500	0.172	2.828	71.30
方形花瓶	2.031	0.625	0.235	2.891	97.10
瓷器花瓶	3.047	0.860	0.328	4.235	136.69
兵马俑	6.682	1.969	0.656	9.307	309.17

本文算法还与BEMARDINI^[12]的ball-pivoting算法、KAZHDAN^[11]的Poisson方法和聂建辉^[18]的快速算法等进行了实验对比,如表4所列。从重建速度看,本文算法大大优于其它算法。此外,本文算法重建网格的顶点均直接取自三维点云中的点,而其它基于隐式函数的方法得到的网格点均为新生成的点,是原始点云的一个光滑逼近,无法真实表达原始点云曲面上的细节特征。

表4 不同算法点云网格曲面重建时间对比

模型	模型点数	重建方法	重建时间(s)	网格模型面片数
瓷器花瓶	989176	ball-pivoting 算法	893.751	715320
		Poisson 方法	176.550	1211477
		聂建辉算法	43.778	634423
		本文算法	4.235	1946996
兵马俑	2265899	ball-pivoting 算法	1408.113	1310913
		Poisson 方法	351.645	2430117
		聂建辉算法	89.738	1098347
		本文算法	9.307	4482392

结束语 本文针对由结构光重建的三维点云进行曲面重建,提出了一种逐层网格化算法,适用于单一投影方向获得的结构光点云数据的曲面重构。本文得出如下结论:

(1)将三维点云的曲面重建问题转化成二维规则点的网格化问题,采用投影网格化法来快速构建点云间拓扑关系,建立点云的网格曲面表达。

(2)通过对二维投影网格的隔行隔列采样,即可方便地实现二维投影点的网格粒度,从而达到三维重建曲面的粒度可控。

(3)重建曲面网格顶点直接源自原始三维点云的点,而非隐函数重建方法中重采样生成的新点。因此本文方法重建曲面较好地保持了物体表面细节。

由于本文方法重建网格模型网格点直接源自点云的点,如果三维点云存在噪声,重建网格也会带有噪声。因此结构光重建的三维点云最好先去噪,然后再用本文方法重建曲面,会得到很好的效果。

参考文献

- [1] Boissonnat J D. Geometric structures for three-dimensional shape representation[J]. ACM Transactions on Graphics, 1984, 3(4):266-286
- [2] Amenta N, Bern M, Kamvyselis M. A new Voronoi-based surface reconstruction algorithm[C]//Proceedings of ACM SIGGRAPH 1998, ACM Press, Orlando, USA, 1998:415-421
- [3] Amenta N, Choi S, Kolluri R K. The power crust [C]// Proceedings of the sixth ACM Symposium on Solid Modeling and Applications(SMA2001). ACM Press, New York, USA, 2001: 249-266
- [4] Adamy U, Giesen J, John M. Surface reconstruction using umbrella filters[J]. International Journal of Computational Geometry, 2002, 21(1):63-86

(下转第410页)

来自 192.168.137.2 的回复: 字节=32 时间<1ms TTL=64
 来自 192.168.137.2 的回复: 字节=32 时间<1ms TTL=64
 来自 192.168.137.2 的回复: 字节=32 时间<1ms TTL=64
 来自 192.168.137.2 的回复: 字节=32 时间<1ms TTL=64
 192.168.137.2 的 Ping 统计信息:
 数据包: 已发送=4, 已接收=4, 丢失=0 (0% 丢失),
 往返行程的估计时间(以毫秒为单位):
 最短=0ms, 最长=0ms, 平均=0ms

图 5 DNS 重定向效果

客户端打开浏览器访问 www.sina.com 网站, 成功显示了网站的所有内容, 同时在网络安全审计系统上使用 Wireshark 软件分析 eth0 接口上的数据流量, 即可得到如图 6 所示的信息。

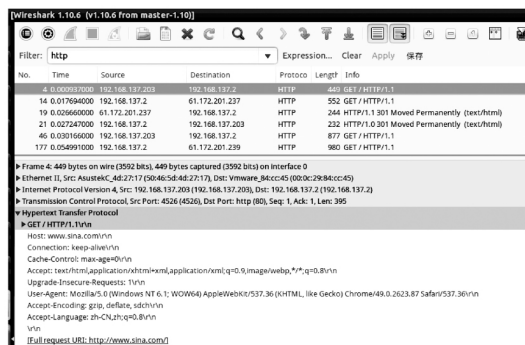


图 6 审计系统工作效果

从图 6 的信息可以看出, 客户端 192.168.137.203 首先向审计服务器 192.168.137.2 发送了访问 www.sina.com 的 HTTP 请求, 由此审计系统即审计了客户端的访问记录, 之后审计系统向真正的网页服务器 61.172.201.237 转发了客户端的 HTTP 请求, 并在得到网页服务器的回应后, 将回应内容转发回客户端, 由此保证了客户端的正常上网功能不受

任何影响。而当客户端访问了被 ACL 禁止的网站时, 由于代理服务的阻止, 无法得到有效的响应, 从而使客户端无法对该网站进行访问。

结束语 本文通过多种开源软件的结合使用, 设计了一种低成本的网络安全审计系统方案, 无需在客户端上做任何改动, 只需修改网卡的 DNS 设置, 即可对整个局域网内的网络访问行为进行安全审计, 非常适合小型网络管理员使用。未来还可以在该系统构架的基础上, 开发相应的分析及管理系统, 从而形成一个具有一定商业价值的网络安全审计方案, 发挥其更高的实用意义。

参考文献

- (上接第 387 页)
- [1] 吴承荣, 廖健, 张世永. 网络安全审计系统的设计和实现[J]. 计算机工程, 1999, 25(S1): 171-174
 - [2] 吴琼. DNS 服务应用于局域网教学的研究[J]. 吉林工程技术师范学院学报, 2010, 26(1): 77-80
 - [3] 胡小梅, 刘嘉勇. 基于 DNS 劫持的流量监测系统设计与实现[J]. 网络安全技术与应用, 2016(1): 110-112
 - [4] 百度百科. 代理服务器[EB/OL]. (2014-06-30) [2016-04-21]. [http://baike.baidu.com/link?url=8R7SISgG8VHHdLyWje5aPu2EM7kFABwodaXOkN9mKw2F2DIUnrmuMED7tqpt9cE2QFRnigfv8ZV33Kd1V094Bq#reference-\[1\]-751-wrap](http://baike.baidu.com/link?url=8R7SISgG8VHHdLyWje5aPu2EM7kFABwodaXOkN9mKw2F2DIUnrmuMED7tqpt9cE2QFRnigfv8ZV33Kd1V094Bq#reference-[1]-751-wrap)
 - [5] 何文华, 梁竞敏. Linux 下传统代理、IP 伪装与透明代理的设计与实现[J]. 计算机系统应用, 2004(10): 60-62
 - [6] 诸晔. 用 ACL 实现系统的安全访问控制[J]. 计算机应用与软件, 2005, 22(03): 111-114
 - [7] 晴刃. 访问控制列表(ACL)实例详解[EB/OL]. (2012-11-25) [2016-05-09]. <http://www.qingsword.com/qing/696.html>
 - [8] 何博远. 局域网环境下利用 DHCP 欺骗技术实现 DNS 劫持攻击[J]. 现代计算机(专业版), 2011(14): 44-47
 - [9] Mederos B, Amenta N, Velho L, et al. Surface Reconstruction from Noisy Point Clouds[C] // Proceedings of the third Eurographics Symposium Geometry Processing (SGP2005). Vienna, Austria, 2005: 53-62
 - [10] 董辰世, 汪国昭. 一个利用法矢的散乱点三角剖分算法[J]. 计算机学报, 2005, 28(6): 1000-1005
 - [11] Carr J C, Beatson R K, Cherrie J B, et al. Reconstruction and representation of 3D objects with radial basis functions[C] // Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2001). ACM Press, 2001: 67-76
 - [12] Alexa M, Behr J, Cohen-Or D, et al. Computing and rendering point set surfaces[J]. IEEE Transactions on Visualization and Computer Graphics, 2003, 9(1): 3-15
 - [13] Du H, Qin H. A shape design system using volumetric implicit PDEs[J]. Computer-Aided Design, 2004, 36(11): 1101-1116
 - [14] Ohtake Y, Belyaev A, ALEXA M, et al. Multi-level partition of unity implicits[J]. Proceedings of ACM SIGGRAPH 2003, ACM Press, San Diego, USA, 2003, 22(3): 463-470
 - [15] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction[C] // Proceedings of the Fourth Eurographics Symposium on Geometry Processing. Sardinia, Italy, 2006: 61-70
 - [16] Bernardini F, Mittleman J, Rushmeier H, et al. The ball-pivoting algorithm for surface reconstruction[J]. IEEE Transactions on Visualization and Computer Graphics, 1999, 5(4): 349-359
 - [17] Kuo C C, Yau H T. A Delaunay-based region-growing approach to surface reconstruction from unorganized points[J]. Computer-Aided Design, 2005, 37(8): 825-835
 - [18] Kuo C C, Yau H T. A new combinatorial approach to surface reconstruction with sharp features[J]. IEEE Transactions on Visualization and Computer Graphics, 2006, 12(1): 73-82
 - [19] 方林聪, 汪国昭. 基于径向基函数的曲面重建算法[J]. 浙江大学学报(工学版), 2010, 44(4): 728-731
 - [20] 赵建东, 康宝生, 康健超, 等. 改进的基于径向基函数的曲面重建算法[J]. 西北大学学报(自然科学版), 2012, 42(5): 744-748
 - [21] 钱归平, 童若锋, 彭文, 等. 基于散乱点云内部特征的网格重建[J]. 浙江大学学报(工学版), 2008, 42(5): 732-735
 - [22] 聂建辉, 马孜, 胡英, 等. 针对密集点云的快速曲面重建算法[J]. 计算机辅助设计与图形学学报, 2012, 24(5): 575-582
 - [23] Huang H, Li D, Zhang H, et al. Consolidation of unorganized point clouds for surface reconstruction[J]. ACM Transactions on Graphics, 2009, 28(5): 1-7
 - [24] Huang H, Wu S, Gong M, et al. Edge-aware point set resampling[J]. ACM Transactions on Graphics, 2013, 32(1): 1-12