

## 多站全景图像犯罪现场三维模型纹理生成方法研究

刘猛<sup>1\*</sup>, 申思<sup>1</sup>, 段动宾<sup>2</sup>, 姚文英<sup>2</sup><sup>1</sup>浙江警察学院刑事科学技术系, 浙江 杭州 310053;<sup>2</sup>浙江安正科技股份有限公司, 浙江 杭州 310008

**摘要** 针对三维模型传统顶点着色渲染方法存在数据体量大、渲染效果差等问题,提出了一种利用点云与多站全景图像空间融合技术生成清晰纹理图像的方法。首先计算全景纹理像素点对应的空间映射点到全景球中心的遮挡;其次进行纹理图像像素点透视情况下多站全景的空间颜色域反距离加权融合;然后对纹理图像空白区域进行反距离加权填充,生成清晰纹理图像;最后采集全景图像和激光点云作为数据源,进行生成纹理图像实验。实验结果表明:所提方法经过并行设计,实现了多站全景图像到高清纹理图像的快速生成;相比传统顶点着色方法,视觉效果更加真实,更加清晰,提升了犯罪现场三维模型可视化效果。

**关键词** 成像系统; 三维重建; 全景图像; 激光点云; 三角化

中图分类号 TP391.41;D918.4 文献标志码 A

DOI: 10.3788/LOP202259.1611008

## Texture Generation Method of Three-Dimensional Model of Crime Scene Based On Multi Panoramic Images

Liu Meng<sup>1\*</sup>, Shen Si<sup>1</sup>, Duan Dongbin<sup>2</sup>, Yao Wenyong<sup>2</sup><sup>1</sup>Department of Criminal Science and Technology, Zhejiang Police College, Hangzhou 310053, Zhejiang, China;<sup>2</sup>Zhejiang Angelcomm Technology Co., Ltd., Hangzhou 310008, Zhejiang, China

**Abstract** To solve the problems of considerable data volume and poor rendering effect in the traditional vertex shading rendering method of three-dimensional (3D) mesh, this study proposes a method to generate clear texture images using the fusion technology of point cloud and multiple-station panoramic images. First, the occlusion from the spatial mapping point of the panoramic texture pixel to the center of the panoramic sphere was calculated. Subsequently, the spatial color domain inverse distance weighted fusion of multiple-station panorama was conducted under texture image pixel point visibility. Additionally, the texture image blank area neighborhood weighted filling was conducted to generate a clear texture image. Finally, panoramic images and laser point clouds were collected as data sources to generate texture images. The experimental results show that the algorithm designed in parallel can quickly generate highly clear textures from multiple-station panoramic images. The overall effect of the model is more realistic and clearer than vertex colored model. The visualization effect of the 3D model of a crime scene is also improved.

**Key words** imaging systems; three-dimensional reconstruction; panoramic image; point cloud; triangulation

## 1 引言

犯罪现场勘查是侦查破案的源头,是诉讼活动的起点,是公安机关打击、防范、控制犯罪的重要基础性工程,勘查质量决定着办案质量。面对打击犯罪的新形势、司法改革的新要求、人民群众的新期待、科技发展的新机遇,随着智慧新刑技、刑事技术信息化建设理

念深入人心,以三维激光扫描为代表的高新技术越来越多地应用到犯罪现场勘查中来,为犯罪现场的记录、重建分析、展示汇报带来了新的手段和方法。及时、客观、全面、细致、合法是犯罪现场勘查的基本要求,利用三维激光扫描技术对犯罪现场的记录、固定不仅要做到快速、客观,可视化效果要做到真实细致,精确客观地还原犯罪现场的原始状态,还要可以进行快速数据

收稿日期: 2021-09-23; 修回日期: 2021-10-13; 录用日期: 2021-11-01

基金项目: 全国教育信息技术研究课题(186140083)、浙江省高等教育“十三五”第一批教学改革研究项目(jg20180381)、浙江省教育科学规划 2021 年度一般规划课题(高校)(2021SCG281)

通信作者: liumeng@zjcxxy.cn

处理以便在线、近距离查看,这样才能够满足三维激光扫描技术应用到犯罪现场勘查的基本要求,才可以为后续的犯罪现场的分析与重建提供客观的依据和较好的可视化效果。

近些年,三维重建一直是热门的课题,目前获取三维数据的常用方法主要有以下几种:RGBD 深度相机移动 simultaneous localization and mapping (SLAM)<sup>[1-2]</sup>、多线激光 SLAM、站点式三维激光、纯视觉 SLAM<sup>[3]</sup>及激光和视觉结合移动 SLAM 等。为更好地展示、表达三维空间,这些方式获取的数据均需进行一系列处理得到三维 Mesh 模型<sup>[4]</sup>;主要处理工序包括点云数据空间抽稀、平滑去噪、剔除离群点、法线估计、点云三角化等<sup>[5-6]</sup>。激光点云数据仅有空间离散采样点数据,没有图像纹理细节;而纯图像数据仅有丰富纹理,缺少空间尺度信息。因此,本实验组采用站点式三维激光与全景图像结合的方式来获得最终结果,主要包括点云数据的预处理、三角化、Mesh 模型的简化、Mesh 模型参数化、纹理计算等<sup>[7]</sup>。

目前,通过点云三角化生成 Mesh 的算法相对较多,如 Hoppe 等<sup>[8]</sup>提出的移动立方体表面重建算法、Li 等<sup>[9]</sup>提出的网格投影算法、Dickerson 等<sup>[10]</sup>提出的贪婪三角形算法、Bernardini 等<sup>[11]</sup>提出的 Ball-Pivoting 算法等。但这些算法都有各自的参数设置,欲适配各种数据生成较好模型,调参复杂,对于包含一定噪声的点云表现一般,如 Ball-Pivoting 算法在点云不均匀分布情况下生成的 Mesh 包含很多孔洞。因此,本实验组选取了经典 Poisson 重建算法<sup>[12]</sup>进行点云的三角化, Poisson 重建算法属于隐式表面重建法<sup>[13]</sup>,对点云数据噪声和空间分布要求较低。

三维模型的参数化<sup>[7]</sup>即建立三维模型与纹理贴图坐标(UV)平面的映射关系,具体可分为 3 个过程:首先将三维模型分割成多个 Chart;然后参数化每个 Chart;最后打包 Chart 到纹理图集。常用算法为 least square conformal maps (LSCM)、as rigid as possible (ARAP) 等<sup>[14-15]</sup>,LSCM 在映射的过程中尽可能地保持三角形的角度相同,ARAP 参数化算法在 LSCM 的基础上尽可能地保证三角形没有扭曲地映射在二维平面上。

在以上相关研究的基础上,本文提出了一种多张全景空间颜色域融合的方法。所提方法通过计算纹理像素点的空间映射点到全景球中心的遮挡情况,进行纹理像素点通视情况下的多站全景的空间颜色域融合,并对纹理图像空白区域进行邻域加权填充。最后对该方法进行实验测试。实验结果表明,所提方法实现了多站全景图像到高清图理图像的快速生成,视觉效果更加真实、更加清晰,提升了犯罪现场三维模型可视化效果。

## 2 基本原理

### 2.1 三角形重心坐标公式

平面内三角形  $P_1P_2P_3$  如图 1 所示,面积为  $s$ ,内部

一点  $P$ ,三角形  $PP_2P_3$  的面积为  $s_1$ ,三角形  $PP_1P_3$  的面积为  $s_2$ ,三角形  $PP_1P_2$  的面积为  $s_3$ ;若已知 3 个顶点对应的属性向量  $V_1、V_2、V_3$ ,则  $P$  的属性向量  $V$  的表达式为

$$V = V_1 \times \frac{s_1}{s} + V_2 \times \frac{s_2}{s} + V_3 \times \frac{s_3}{s}, \quad (1)$$

式中: $s$  为  $s_1、s_2、s_3$  的和。

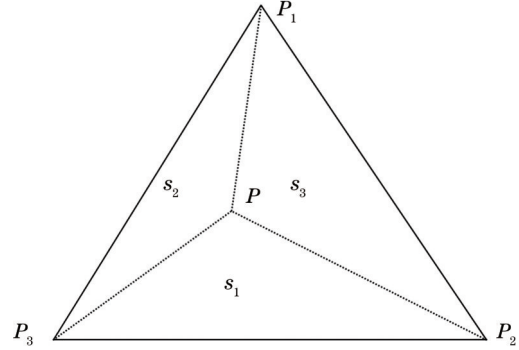


图 1 三角形  $P_1P_2P_3$

Fig. 1 Triangle  $P_1P_2P_3$

利用上面三角形面积比例不变性,计算平面三角形 Triangle2D 中内一点  $P$  点对应在空间三角形 Triangle3D 中的坐标,即

$$\begin{cases} x_P = x_{P_1} \times \frac{s_1}{s} + x_{P_2} \times \frac{s_2}{s} + x_{P_3} \times \frac{s_3}{s} \\ y_P = y_{P_1} \times \frac{s_1}{s} + y_{P_2} \times \frac{s_2}{s} + y_{P_3} \times \frac{s_3}{s} \\ z_P = z_{P_1} \times \frac{s_1}{s} + z_{P_2} \times \frac{s_2}{s} + z_{P_3} \times \frac{s_3}{s} \end{cases} \quad (2)$$

### 2.2 全景球面透视投影模型

全景球面透视投影模型<sup>[16-17]</sup>不同于传统平面透视投影模型,如图 2 所示,采用预设的单位球球面作为投影基准面,球中心  $O$  为坐标原点,半径  $R=1$ ,原点  $O$  到球面上一点  $P$  连线为投影射线,过球中心的赤道面为坐标系  $XOY$  平面,过原点  $O$  与  $XOY$  平面垂直向上为  $Z$  轴; $\alpha$  为  $OP$  到  $XOY$  平面投影与  $X$  轴夹角, $\beta$  为  $OP$  与  $XOY$  平面夹角;通过  $R、\alpha、\beta$  可以计算球面任一点  $T$  的三维坐标  $x, y, z$ 。

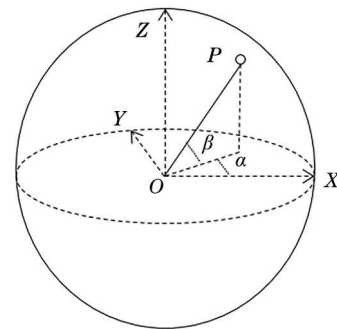


图 2 球面几何

Fig. 2 Spherical geometry

$$\begin{cases} x=R \times \cos \alpha \times \sin \beta \\ y=R \times \cos \alpha \times \cos \beta, 0 \leq \alpha \leq 360, -90 \leq \beta \leq 90。 \\ z=R \times \sin \beta \end{cases} \quad (3)$$

同理,根据式(3),当已知球面点的三维空间坐标时,可以解算出 $\alpha$ 和 $\beta$ ,需要注意的是解算 $\alpha$ 时需要注意 $\alpha$ 的象限确定问题,其表达式分别为

$$\begin{cases} \alpha = \arctan\left(\frac{x}{y}\right) \\ \beta = \arctan\left(\frac{z}{\sqrt{(x \times x + y \times y)}}\right) \\ 0 \leq \alpha \leq 360, -90 \leq \beta \leq 90 \end{cases} \quad (4)$$

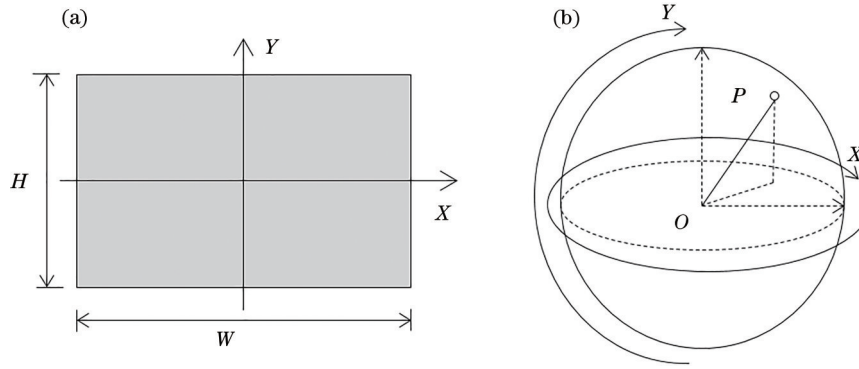


图 3 平面到球面的关系。(a)全景图像的平面坐标系;(b)全景图像的球坐标系

Fig. 3 Relation between panoramic image and panoramic sphere. (a) Coordinate system of panoramic image; (b) coordinate system of panoramic sphere

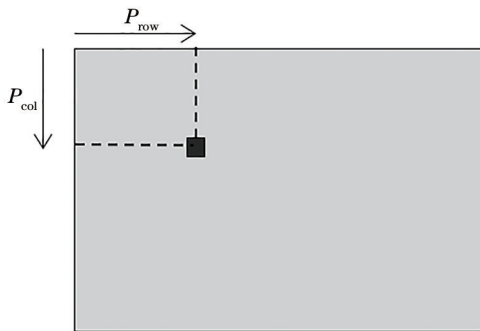


图 4 全景图像像素坐标系

Fig. 4 Coordinate system of texture image

同理,根据式(5),已知全景图像中任意一点的像素位置时,也可以解算出其对应球面上点的位置。

### 2.3 全景到纹理图的映射

全景到纹理图的映射计算过程包括纹理图像与UV平面的映射<sup>[18]</sup>、三维模型与UV平面的映射<sup>[19]</sup>、三维模型与全景球的映射、全景球与全景图像的映射,如图5所示。

三维模型经过参数化后生成UV平面坐标,三维模型的每个顶点都有一个二维UV坐标值与其对应,同理三维模型的每个三角面都有UV平面上三角面与其对应,

由全景球面透视投影模型可知,全景图像被看作球面几何<sup>[17]</sup>,但常见全景图像是平面的二维图,在水平方向是360°视角,垂直方向为180°视角,标准全景图像的宽(W)为高(H)的2倍,总像素数为 $W \times H$ 。因此,需要建立平面全景与球面全景的映射对应关系,具体如图3所示。球面任一点 $T(x, y, z)$ ,其对应的全景图像中像素点位置为 $(P_{col}, P_{row})$ , $P_{col}$ 代表图像的列号, $P_{row}$ 代表图像的行号,如图4所示,假设图像的第1列与球面XOZ对应,顺时针依次对应起来,同时第1行与球面的顶部对应,有

$$\begin{cases} P_{col} = \alpha \times 2\pi / W \\ P_{row} = (\beta + 0.5\pi) \times \pi / H \end{cases} \quad (5)$$

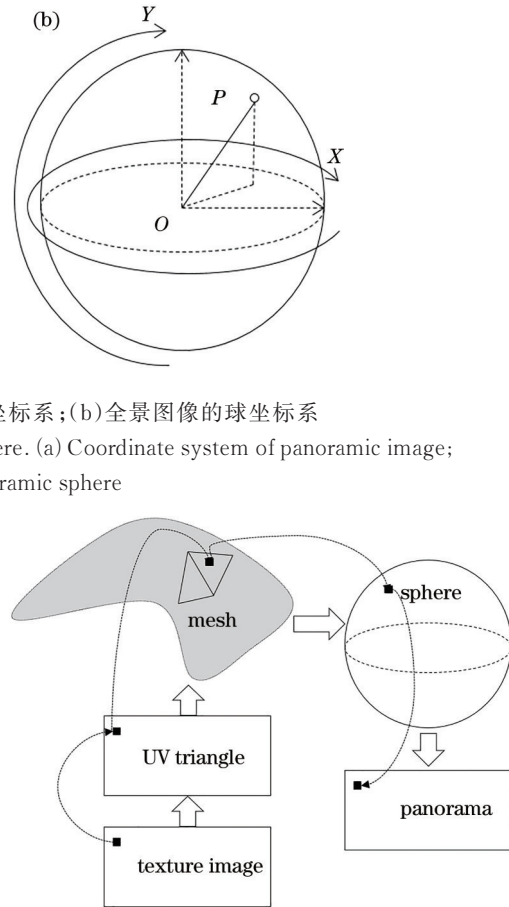


图 5 全景图像与纹理图像的关系

Fig. 5 Relation between panoramic image and texture image

即完成三维模型与平面UV的映射计算,如图6所示。

则三维模型通过参数化后,UV坐标的阈值范围为 $[0.0, 1.0]$ ;假设纹理图像宽为 $w$ ,高为 $h$ ,可根据式(6)计算纹理图像的任意像素点 $i(P_{col}, P_{row})$ 的UV坐标;通过该UV坐标检索到该点所处UV平面上的三角面Triangle2D,同时可得对应的三维模型空间的三角面Triangle3D,利用式(2)计算其在Triangle3D中的坐标,然后利用球面透视投影模型计算像素点 $i$ 所在全景图像中的位置,即实现了全景图像到纹理图像的映射。

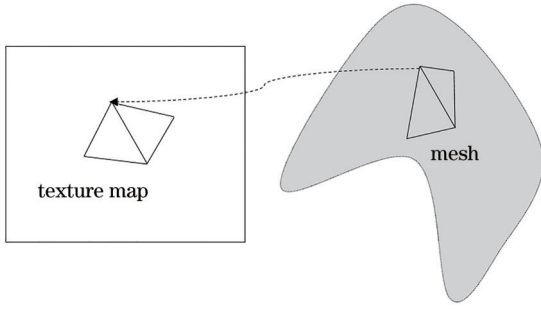


图 6 模型到纹理图集

Fig. 6 Mesh to texture map

$$\begin{cases} u = P_{col} / \tau w \\ v = P_{row} / h \end{cases} \quad (6)$$

### 3 所提算法流程和实现

所提算法基于 Mesh 模型数据、Mesh UV<sup>[20]</sup>展开数据及全景图像实现 Mesh 模型纹理贴图的自动生成。首先,利用式(3)、(4)、(5)、(6)计算纹理图每个像素点到全景图像的映射,同时计算纹理像素对应的三维空间点到全景球中心的通视遮挡;然后,对该像素点通视的多个全景进行色彩域融合,生成彩色纹理图;最后,利用像素临近域反距离加权方法填充彩色纹理图的空白区,导出最终的纹理图像。具体流程如图 7 所示。

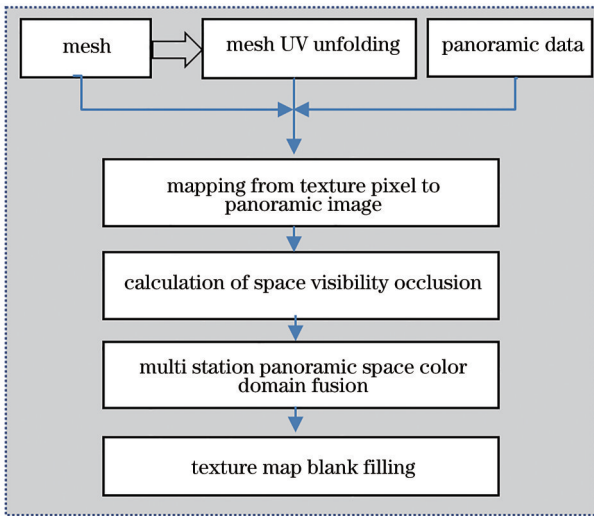


图 7 纹理生成算法流程图

Fig. 7 Flow chart of texture generating algorithm

#### 3.1 空间通视遮挡计算

全景球中心为点  $o$ , 空间地物表面存在一点  $P$ , 为确定  $OP$  光线的通视, 需要判断  $OP$  线段是否有横穿场景模型的三角面, 如图 8 所示, 当  $OP$  横穿三角形  $S_1$  时, 交点为  $K$ , 则  $K$  必在三角形  $S_1$  内或上; 同时计算向量  $\overrightarrow{KO}$  与  $\overrightarrow{KP}$  的夹角, 判断点  $K$  是否位于线段  $OP$  之间。

首先, 由第 2 小节纹理图像与 UV 平面的映射、三维模型与 UV 平面的映射, 计算得到纹理图像到三维空间的映射, 但纹理图像中并非所有像素点都有三维

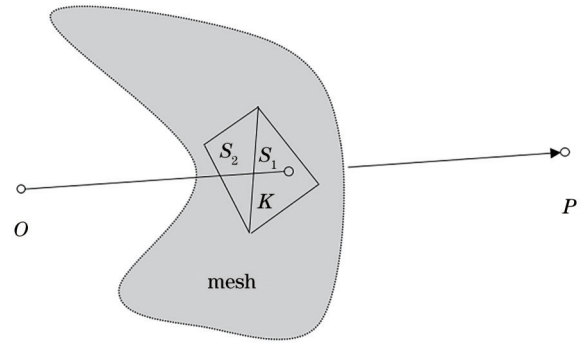


图 8 射线遮挡

Fig. 8 Ray occlusion

空间点与其对应, 只有位于 UV 平面中三角形内的像素点才可计算得其在三维空间的映射点, 剩下空白区采用别的颜色填充方法。

算法 1 解决了纹理图像在三维空间的映射点到全景球中心的通视遮挡计算问题。当纹理图像总量较大及场景中三角形过多时, 对纹理图像像素点进行遍历计算其对三角形的遮挡, 就会让计算体量变得很大; 因此, 为减小计算量设计了距离阈值 Threshold, 当三角形顶点到线段  $OP$  最短距离小于该距离阈值时, 计算相对该三角形的遮挡。

Algorithm 1: Caculate Shelter

```

1: function CaculateShelter(triangle, panorama, texture_pixel)
2:   Bool shelter_state ← false
3:   If(CaculateInTriangle(uv, texture_pixel)) then
4:     pt ← Caculate3DpointOfTecturePixel(in_triangle, texture_pixel)
5:     MinDis ← CaculateMinDistance(triangle, pt)
6:     If(MinDis < Threshold) then
7:       Caculate intersection point K of OP and triangle
8:       Caculate angle of KO and KP
9:       If(angle ≈ π) then
10:        shelter_state ← true
11:       end//if
12:     end//if
13:   end//if
14:   return shelter_state
15: end function
    
```

在 Algorithm 1 中, triangle 为输入的三维模型的三角形, 用于射线求交判断; in\_triangle 为输入的空间三角形, 用于像素点的空间映射计算; uv 为 in\_triangle 三个点对应的 uv 坐标组成的平面三角形; texture\_pixel 为纹理图中的一个像素坐标; CaculateInTriangle 用于判断 texture\_pixel 是否在 uv 平面三角形中; Caculate3DpointOfTecturePixel 是式 (2) 的实现, 用于计算 texture\_pixel 在 in\_triangle 中的坐标 pt; CaculateMinDistance 用于计算空间三角形 triangle 到线段  $OP$  的最小距离 MinDis; 当 MinDis 小于阈值 Threshold 时, 计算 triangle 与线段  $OP$  的交点及向量  $\overrightarrow{KO}$

与  $\overrightarrow{KP}$  的夹角, 如果夹角近似等于  $\pi$ , 返回遮挡状态为真。

### 3.2 全景空间颜色域融合

纹理图实际上是用全景图像中的颜色进行回填得到的, 由于相邻站点全景图像的色差、全景位姿误差及图形计算精度问题, 并不能确保纹理图生成的精度和视觉上的美观。算法 2 采用一种多站全景图像在颜色回填时进行反距离加权融合的方法避免了生成纹理图色差问题。

Algorithm 1 计算了单站全景与空间点连线的遮挡, 在 Algorithm 2 中将计算纹理图像像素点对应的三维空间点到全景球中心的距离, 每个像素点会有  $n$  个与其通视的全景站点, 如图 9 所示, 颜色融合的权重和像素点颜色的表达式为

$$W_i = \frac{(1/d_i)}{\sum_{i=0}^n (1/d_i)}, \quad (7)$$

$$C_{\text{RGB}} = \sum_{i=0}^n W_i \times C_{\text{rgb}_i}, \quad (8)$$

式中:  $d_i$  为像素空间点到全景中心的距离;  $W_i$  为该像素点在该站点全景的权重;  $C_{\text{rgb}_i}$  即为在通视全景的映射点颜色;  $C_{\text{RGB}}$  为最终计算得到的像素点颜色。

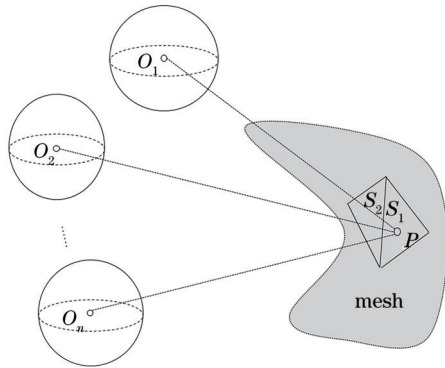


图 9 多站视点融合

Fig. 9 Multi station view fusion

Algorithm 2: Generate Texture Image Tith Multi-Panoramas

```

1: function GenerateTexture (in_triangles, panoramas)
2:   texture_image
3:   for texture_pixel in texture_image
4:     for panorama in panoramas
5:       for triangle in in_triangles
6:         shelter_state ← CaculateShelter (triangle, panorama,
texture_pixel)
7:         distance_pixel ← CaculateDistance (triangle,
panorama, texture_pixel)
8:       end for
9:     end for
10:    CaculateFillColor (shelter_states, distance_pixels,
texture_pixel)
11:  end for
12:  return texture_image
13: end function

```

在 Algorithm 2 中, in\_triangles 为三维模型的所有空间三角形; panoramas 为多站全景数据; texture\_image 为预定输出的纹理图; CaculateShelter 为 Algorithm 1 中实现的方法; CaculateDistance 用于计算像素点的空间映射点到各站全景球中心的反距离权重, 即式 (7) 的实现; CaculateFillColor 使用反距离加权计算像素点的颜色, 即式 (8) 的实现。

### 3.3 二维纹理图空白填充

在第 3.2 节已经提到, 纹理图像中并非所有像素点都有三维空间点与其对应, 这部分像素点利用 Algorithm 2 无法计算出颜色, 纹理图像这部分就会形成空白, 导致在进行纹理与三维模型渲染显示时, 相邻 UVchart<sup>[20]</sup> 之间会出现明显接缝问题。因此, Algorithm 3 采用一种邻近域反距离加权的方法进行纹理空白的填充, 首先以空白像素点  $i$  进行邻域搜索, 得到  $N$  个非空白像素点, 同样利用式 (7)、(8) 解算空白像素点  $i$  的颜色。

Algorithm 3: Fill Texture Image

```

1. function FillTexture (texture_image)
2.   texture_image_t ← texture_image
3.   buildKdtree (texture_image_t)
4.   for i in texture_image_t
5.     N ← KnnSearch (i)
6.     CalculateWight (N)
7.     CalculateWightColor (N)
8.   end for
9.   return texture_image_t
10. end function

```

在 Algorithm 3 中, texture\_image 为输入的纹理图像; buildKdtree 为纹理图像创建 Kdtree, 用于临近像素的检索; KnnSearch 是在创建的 Kdtree 基础上, 进行临近像素点的搜索; CalculateWight 计算像素点与临近非空像素点的反距离权重; CalculateWightColor 利用反距离权重计算像素点的颜色。

## 4 分析与讨论

在实际采集的数据上进行了相关实验, 以验证所提算法效率及生成纹理的美观程度。首先, 利用 faro M70 站点式激光全景扫描仪, 在模拟室内案发现场获取三维激光点云数据以及全景数据; 然后, 对点云数据进行预处理, 包括多站点云匹配、点云融合、点云平滑去噪、点云空间抽稀、点云法向估计、区域切割等, 利用 Poisson 重建算法<sup>[12]</sup> 生成三角网格数据, 通过 Quadric based Edge Collapse 算法<sup>[21-23]</sup> 对三角网格进行简化处理, 并对简化后的网格采用 LSCM 算法<sup>[15]</sup> 进行 UV 展开; 最后, 通过所提相关算法实现纹理图的生成。所用计算机的操作系统为 Windows 10, CPU 为 Intel-i7-8750H, 内存为 32 GB, GPU 为 Intel UHD Graphic 630

和 NVIDIA GeForce GTX 1050Ti。

#### 4.1 空间通视遮挡计算

实际上,在生产纹理图过程中,需要计算纹理图像每个像素点到各站全景图像的通视遮挡,当场景模型三角面、站点全景数据过多时,计算量变大,可将空间通

视遮挡计算部分,移植到 GPU 端计算,提高计算效率。

不计算遮挡情况下,与全景球中心不通视的区域会得到一些错误纹理;相反,遮挡计算后,不通视的区域不计算纹理,在多站融合时由相邻站点全景补充计算,如图 10 所示。

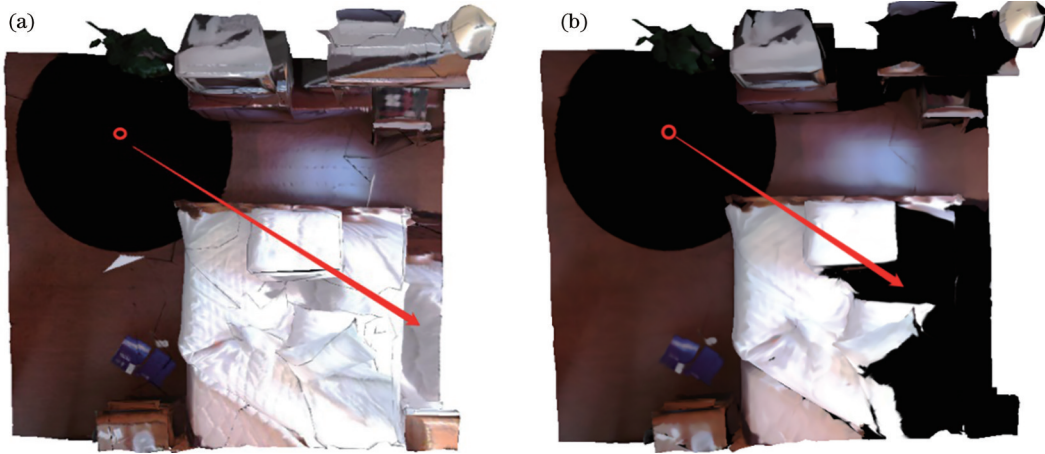


图 10 遮挡计算前后效果。(a) 遮挡计算前;(b) 遮挡计算后

Fig. 10 Effect before and after occlusion calculation. (a) Before occlusion calculation; (b) after occlusion calculation

#### 4.2 多站全景空间颜色域融合纹理图生成

所提算法生成的纹理图像如图 12 所示,图 11 为融合前的图像。从图中可以看出:所提算法不仅补充了相邻站全景盲区(图 11 右下角黑色区域)和不通视区(图 11 左上角黑色区域),而且很大程度上缓和了相邻站全景之间的过度色差。

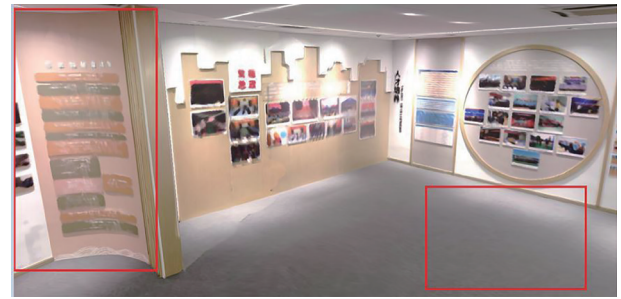


图 12 融合后

Fig. 12 After blended



图 11 融合前

Fig. 11 Before blended

#### 4.3 纹理图空白填充

选取了一张生成的纹理图作为测试图像,通过 Algorithm 3 实现纹理图像中空白区域的颜色填充,图 13、14 分别为纹理填充前后的效果,纹理填充前渲染左侧的模型时,UVchart 之间会显示出黑色裂缝,而纹理填充后,黑色裂缝消失,视觉效果得到提升。另外为提高填充计算效率,可考虑将 Algorithm 3 移

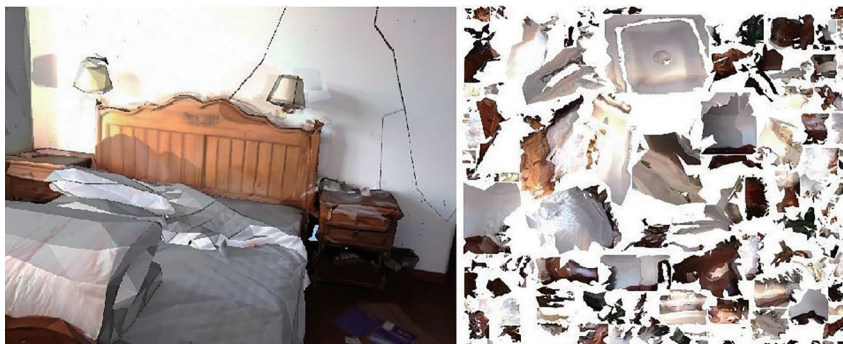


图 13 纹理填充前

Fig. 13 Before texture filled

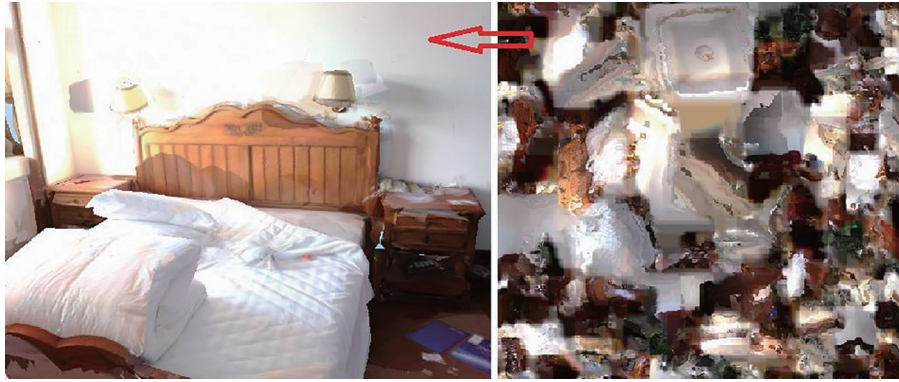


图 14 纹理填充后

Fig. 14 After texture filled

植到 GPU 端运行, 或者在 CPU 端进行多线程并行计算。

最后, 测试了一些场景数据, 图 15 为常见顶点着

色渲染的模型效果, 图 16 为所提算法实现的纹理图像渲染的模型效果; 整体效果相比顶点着色的模型, 颜色过渡平顺, 视觉效果更加清晰。

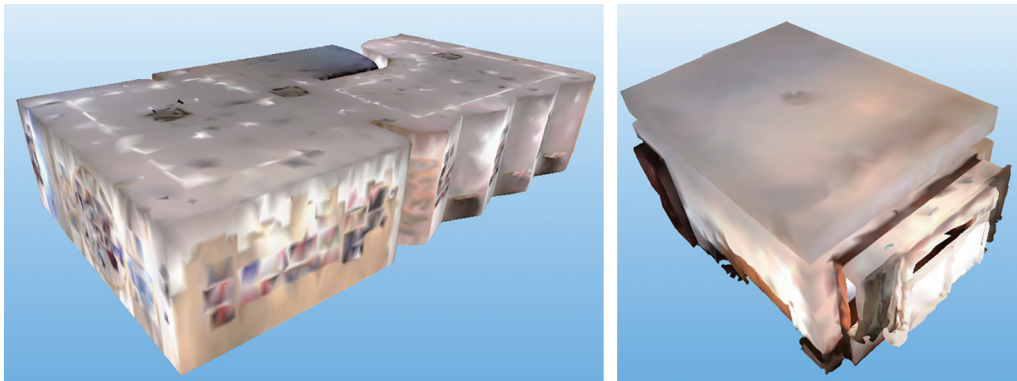


图 15 顶点着色渲染效果

Fig. 15 Vertex shading rendering effect

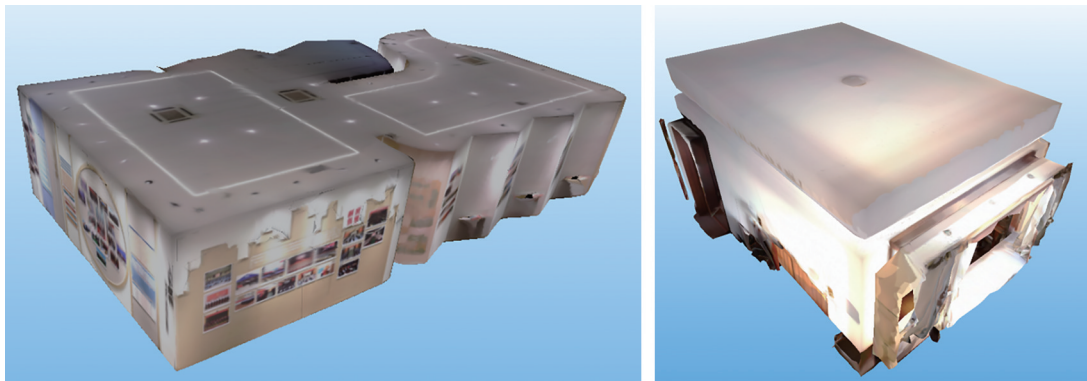


图 16 纹理贴图渲染效果

Fig. 16 Texture map rendering effect

## 5 结 论

提出了一种基于激光点云和全景图像生成清晰纹理的方法。所提方法利用全景球模型将多站全景图像在空间上利用反距离加权方法进行颜色域融合, 同时计算纹理像素点的空间映射点到全景球中心的通视遮挡, 并进行纹理图像空白填充。实验

结果表明, 该方法完成了点云与全景图像的数据融合, 能够生成清晰、真实的纹理图像, 减小了最终整体数据的体量, 提升了视觉效果, 更加符合犯罪现场勘查实战应用的要求, 为三维激光扫描技术更好地应用到犯罪现场的固定与记录、分析与重建奠定了基础, 实现了室内犯罪现场三维重建技术的突破。

## 参 考 文 献

- [1] Newcombe R A, Izadi S, Hilliges O, et al. KinectFusion: Real-time dense surface mapping and tracking[C]//2011 10th IEEE International Symposium on Mixed and Augmented Reality, October 26-29, 2011, Basel, Switzerland. New York: IEEE Press, 2011: 127-136.
- [2] Whelan T, Johannsson H, Kaess M, et al. Robust real-time visual odometry for dense RGB-D mapping[C]//2013 IEEE International Conference on Robotics and Automation, May 6-10, 2013, Karlsruhe, Germany. New York: IEEE Press, 2013: 5724-5731.
- [3] 刘浩敏, 章国锋, 鲍虎军. 基于单目视觉的同时定位与地图构建方法综述[J]. 计算机辅助设计与图形学学报, 2016, 28(6): 855-868.  
Liu H M, Zhang G F, Bao H J. A survey of monocular simultaneous localization and mapping[J]. Journal of Computer-Aided Design & Computer Graphics, 2016, 28(6): 855-868.
- [4] Zhou Q Y, Koltun V. Dense scene reconstruction with points of interest[J]. ACM Transactions on Graphics, 2013, 32(4): 1-8.
- [5] Rusu R B. Semantic 3D object maps for everyday manipulation in human living environments[J]. KI-Künstliche Intelligenz, 2010, 24(4): 345-348.
- [6] Berge M, Tagliasacchi A, Seversky L, et al. State of the art in surface reconstruction from point clouds[C]//Proceedings of Eurographics 2014-State of the Art Reports, April 7-11, 2014, Strasbourg, France. Geneve: The Eurographics Association, 2014, 24(4): 161-185.
- [7] Lai J Y, Wu T C, Phothong W, et al. A high-resolution texture mapping technique for 3D textured model[J]. Applied Sciences, 2018, 8(11): 2228.
- [8] Hoppe H, DeRose T, Duchamp T, et al. Surface reconstruction from unorganized points[J]. ACM SIGGRAPH Computer Graphics, 1992, 26(2): 71-78.
- [9] Li R S, Liu L, Phan L, et al. Polygonizing extremal surfaces with manifold guarantees[C]//Proceedings of the 14th ACM Symposium on Solid and Physical Modeling-SPM '10, September 1-3, 2010. Haifa, Israel. New York: ACM Press, 2010: 189-194.
- [10] Dickerson M T, Scot Drysdale R L, McElfresh S A, et al. Fast greedy triangulation algorithms[J]. Computational Geometry, 1997, 8(2): 67-86.
- [11] 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.
- [12] Kazhdan M, Bolitho M, Hoppe H. Poisson surface reconstruction[C]//Proceedings of Eurographics Symposium on Geometry Processing 2006, June 26-28, 2006, Cagliari, Sardinia, Italy. Geneve: The Eurographics Association, 2006: 61-70.
- [13] Moenning C, Dodgson N A. Fast marching farthest point sampling for point clouds and implicit surfaces[D]. Cambridge: University of Cambridge Computer Laboratory, 2003.
- [14] Lévy B, Petitjean S, Ray N, et al. Least squares conformal maps for automatic texture atlas generation[J]. ACM Transactions on Graphics, 2002, 21(3): 362-371.
- [15] Sorkine O, Marc A. As-rigid-as-possible surface modeling[C]//Proceedings of Eurographics Symposium on Geometry Processing 2007, July 4-6, 2007, Barcelona, Spain. Geneve: The Eurographics Association, 2007: 109-116.
- [16] 肖潇, 杨国光, 白剑. 基于球面透视投影约束的全景环形透镜畸变校正[J]. 光学学报, 2008, 28(4): 675-680.  
Xiao X, Yang G G, Bai J. Panoramic-annular-lens distortion correction based on spherical perspective projection constraint[J]. Acta Optica Sinica, 2008, 28(4): 675-680.
- [17] 闫利, 曹亮, 陈长军, 等. 分片纠正的球面全景影像匹配方法研究[J]. 测绘科学, 2015, 40(11): 23-28.  
Yan L, Cao L, Chen C J, et al. A patch rectification-based feature matching method for spherical panoramic images[J]. Science of Surveying and Mapping, 2015, 40(11): 23-28.
- [18] Liu S R, Ferguson Z, Jacobson A, et al. Seamless: seam erasure and seam-aware decoupling of shape from mesh resolution[J]. ACM Transactions on Graphics, 2017, 36(6): 1-15.
- [19] 杜瑞建, 葛宝臻, 陈雷. 多视高分辨率纹理图像与双目三维点云的映射方法[J]. 中国光学, 2020, 13(5): 1055-1064.  
Du R J, Ge B Z, Chen L. Texture mapping of multi-view high-resolution images and binocular 3D point clouds [J]. Chinese Optics, 2020, 13(5): 1055-1064.
- [20] Jeon J, Jung Y, Kim H, et al. Texture map generation for 3D reconstructed scenes[J]. The Visual Computer, 2016, 32(6/7/8): 955-965.
- [21] Li J F, Chen Y B, Wang M S, et al. Half-edge collapse simplification algorithm based on angle feature[C]//Advances in Intelligent Systems Research", "Proceedings of the 2015 International Conference on Automation, Mechanical Control and Computational Engineering, April 24-26, 2015, Shunhe, China. Paris: Atlantis Press, 2015: 367-371.
- [22] Cignoni P, Corsini M, Ranzuglia G. MeshLab: an open-source 3D mesh processing system[EB/OL]. [2021-03-06]. <https://ercim-news.ercim.eu/>.
- [23] Garland M, Heckbert P S. Surface simplification using quadric error metrics[C]//Proceedings of the 24th annual conference on Computer graphics and interactive techniques-SIGGRAPH '97, August 3-8, 1997, Los Angeles, CA, USA. New York: ACM Press, 1997: 209-216.