

SHEN, L., LI, G., XIAN, C., JIANG, Y. and XIONG, Y. 2019. Door and window detection in 3D point cloud of indoor scenes. *Jisuanji fuzhu sheji yu tuxingxue xuebao/Journal of computer-aided design and computer graphics* [online], 31(9), pages 1494-1501. Available from: <https://doi.org/10.3724/SP.J.1089.2019.17575>

# Door and window detection in 3D point cloud of indoor scenes.

SHEN, L., LI, G., XIAN, C., JIANG, Y. and XIONG, Y.

2019



## 室内 3D 点云模型的门窗检测

沈 乐<sup>1)</sup>, 李桂清<sup>2)</sup>, 冼楚华<sup>2)</sup>, 江 洋<sup>3)</sup>, 熊赞晖<sup>1)\*</sup>

<sup>1)</sup> (华南理工大学数学学院 广州 510640)

<sup>2)</sup> (华南理工大学计算机科学与工程学院 广州 510006)

<sup>3)</sup> (School of Computing Science and Digital Media, Robert Gordon University, Aberdeen AB107QB)  
(yhxiong@scut.edu.cn)

**摘 要:** 为了检测室内 3D 场景中的门窗信息, 提出一种 3D-2D-3D 的门窗检测算法. 首先在 3D 室内场景点云模型中多角度旋转拍照, 获取点云的 2D 图像; 然后对 2D 图像进行门窗目标的粗检测, 得到门窗在图像中的大致范围, 并将此 2D 信息返回到 3D 点云数据中, 得到包含门窗的局部点云数据; 最后提取局部点云数据的轮廓线及其交点, 通过优化得到门窗特征角点的位置信息. 实验结果表明, 采用这种“整体-局部”策略的算法能有效地检测出 3D 室内场景中门窗的位置信息.

**关键词:** 室内场景; 点云; 目标检测; 门窗检测; 特征角点

中图分类号: TP391.41 DOI: 10.3724/SP.J.1089.2019.17575

## Door and Window Detection in 3D Point Cloud of Indoor Scenes

Shen Le<sup>1)</sup>, Li Guiqing<sup>2)</sup>, Xian Chuhua<sup>2)</sup>, Jiang Yang<sup>3)</sup>, and Xiong Yunhui<sup>1)\*</sup>

<sup>1)</sup> (School of Mathematics, South China University of Technology, Guangzhou 510640)

<sup>2)</sup> (School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006)

<sup>3)</sup> (School of Computing Science and Digital Media, Robert Gordon University, Aberdeen AB107QB)

**Abstract:** This paper proposes a 3D-2D-3D algorithm for doors and windows detection in 3D indoor environment of point cloud data. Firstly, by setting up a virtual camera in the middle of this 3D environment, a set of pictures are taken from different angles by rotating the camera, so that corresponding 2D images can be generated. Next, these images are used to detect and identify the positions of doors and windows in the space. To obtain point cloud data containing the doors and windows position information, the 2D information are then mapped back to the origin 3D point cloud environment. Finally, by processing the contour lines and crossing points, the features of doors and windows through the position information are optimized. The experimental results show that this “global-local” approach is efficient when detecting and identifying the location of doors and windows in 3D point cloud environment.

**Key words:** indoor scene; point cloud; object detection; door and window detection; feature corner points

近年来, 随着 3D 数据获取设备的普及, 3D 场景的重建在计算机图形学领域热度不减, 其在建

筑保护、城市景观、机器人应用等方面应用广泛. 目前, 大部分建筑物重建工作集中在大尺度建筑

收稿日期: 2018-09-04; 修回日期: 2018-11-20. 基金项目: 国家自然科学基金(61572202); 广东省自然科学基金(S2013020012795, 2017A030313347). 沈 乐(1994—), 女, 硕士研究生, 主要研究方向为 3D 室内场景重建; 李桂清(1966—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究方向为数字几何处理、图像与视频编辑、可视分析等; 冼楚华(1982—), 男, 博士, 副教授, 硕士生导师, CCF 会员, 主要研究方向为智能识别与图像处理、几何建模与处理、计算机视觉等; 江 洋(1984—), 女, 博士, 讲师, 博士生导师, 主要研究方向为数字媒体的机器学习应用、计算机动画等; 熊赞晖(1972—), 男, 博士, 副教授, 硕士生导师, CCF 会员, 论文通讯作者, 主要研究方向为优化算法、图像处理、3D 室内场景重建、增材制造等.

物,如整个街区、整栋建筑物或整个建筑物群的外部重建<sup>[1-4]</sup>,也有少量集中在对建筑物内部信息重建<sup>[5]</sup>.这些算法都能用于整个室内场景的重建,但重建时对门窗等特征细节信息会有所缺失.本文的工作集中在建筑物内部信息重建上,而其中门窗检测是建筑物内部3D场景重建中的重要步骤,模型上往往因为没有门窗等信息而导致重建结果不真实,因此,门窗信息的检测尤为重要.

门窗检测是3D场景重建的重要步骤,尤其是在无人机、机器人、室内设计和建筑模型重建等方面有着举足轻重的地位.无人机从室内飞向室外,需要寻找门窗等准确的出口位置信息;机器人在室内移动,需要准确的门位置信息以方便出入;而室内家居设计需考虑门窗信息,以防被其他家具物体遮挡<sup>[6]</sup>;此外,在进行建筑模型重建方面,建筑物的细节位置信息,包括门、窗,是重建流程的关键.本文对如何识别出门窗等特征信息的工作进行了探讨,能够找出门窗在整个室内场景中的位置,以供后期场景重建中进行门窗的细节添加,并提出了一种自动提取门窗在室内场景中位置的算法.

## 1 相关工作

### 1.1 基于2D图像进行处理

Quintana等<sup>[7-8]</sup>提出了在门不同的开合状态下,检测门的3D位置的算法.其算法输入是带有颜色信息的单面矩形墙体点云及相应含有标记的体素,核心思想都是将3D信息转换到带有颜色和深度信息的2D图像,并在2D图像上利用传统图形学算法进行检测,然后映射回3D空间.不过针对门闭合时颜色和周围颜色相近的情况,该算法会失效.Kakillioglu等<sup>[9]</sup>利用无人车在室内扫描获取数据,包括点云数据和相应的彩色图像.其首先在3D点云数据上检测到主平面,然后将此平面对应的点云数据转换为相应的深度图,并在深度图上探测空洞区域作为门的候选集;接着利用预先训练好的聚合通道特征检测算子来筛选候选集,以得到真正门的位置;但是,从点云数据中获取的门候选集一旦出错,则检测不出门的位置.Chen等<sup>[10]</sup>利用机器人在不同角度下拍摄的彩色门照片作为输入,然后利用CNN训练得到在当前拍摄角度下机器人与门之间的空间位置关系.利用Kinect扫描获取的深度数据信息<sup>[11-14]</sup>进行处理也是检测的有效途径,因为这样可以最小化光照因素对数据

的影响.其中,Derry等<sup>[11]</sup>的算法根据实际需求,对于门的精确位置大小要求不高,找到门在地面上的位置即可.而Yuan等<sup>[12]</sup>则在Derry等<sup>[11]</sup>的算法基础上进行改进,将检测门的开合状态延伸到开或半开,不过其算法是以Kinect相机总是垂直于门面为前提的.Kelly等<sup>[1]</sup>利用神经网络结合相应的谷歌街道图像进行分割得到的建筑物分割立面集合,然后进行组合优化,得到了带有门窗信息的建筑模型.

### 1.2 基于3D点云进行处理

Borgesen等<sup>[15]</sup>利用机器人传感器获取数据,其算法输入是包含完整门结构的局部点云;然后利用下采样降低点云密度,减少计算量;接着采用区域增长分割算法和随机抽样一致(random sample consensus, RANSAC)算法<sup>[16]</sup>得到包含把手的门的候选平面结构,并对于此候选结构计算出其边界点就能得到相应的门框架.此外,根据门把手材质对于传感器反射特性以及欧氏聚类的算法,在门点云结构中确定了门把手大小、形状和位置.与一般的门框检测算法不同,Rusu等<sup>[17]</sup>是检测门平面的位置和大小(无论门的开合状态),而非固定在墙面上门框的位置,以及确定门把手的具体位置,其并未利用额外的辅助数据,如相机拍摄的图像.Ochmann等<sup>[18]</sup>检测重建墙面和模拟激光射线从扫描位置到测量点之间的交点,通过一个基于交叉点距离的贪婪单联动聚类算法得到墙上开口位置,即门窗.Wang等<sup>[19]</sup>基于大部分城市建筑物外表面窗户的重复性结构特点,设计了自动重建算法;首先是检测重复性结构,并提取该重复结构单元后进行规则约束优化,然后将优化结果返回到模型上,得到带有门窗纹理信息的3D模型.不过,这种算法只对与建筑表面含有大量相同结构的门窗信息才有效,一旦建筑物结构表面门窗元素太过个性化且风格各不相同,则该算法检测效果不好.

## 2 重建流程及其关键算法

本文的目标是找到室内场景中门窗在墙面上的位置,但基于整个室内场景的复杂性,本文采用“整体-局部”的策略.首先从室内场景中找到门窗的大致位置,得到有关门窗的局部点云;然后再基于局部门窗数据确定最终的门窗位置.其大致步骤是通过3D门窗点云转换成2D图像,从2D图像

中识别出门窗;然后将 2D 图像中门窗的信息转换为 3D 门窗信息,最终得到门窗的位置,如图 1 所示.

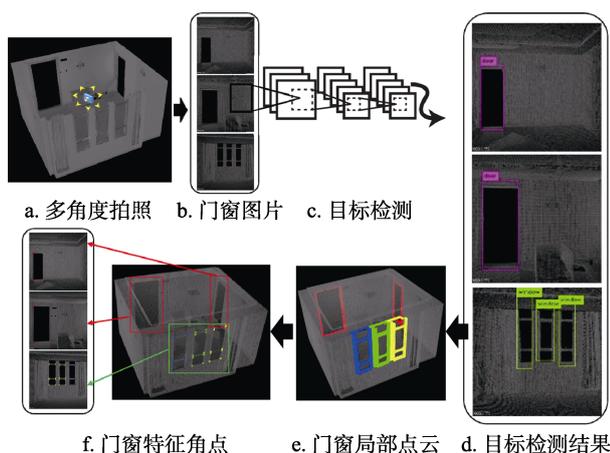


图 1 本文主要流程

在整个流程中,关键步骤如下:

Step1. 3D 与 2D 的对应. 由于从扫描设备直接获取的点云数据中存在大量重复、孤立的点云,也存在与重建目标不相关的点云数据. 因此,必须先对原始室内点云数据进行清洗、去噪等预处理工作,以获得相对完整和准确的数据. 在此基础上,本文得到室内场景不同视角的 2D 图像. 具体做法是:在房屋 3D 室内点云数据中设置一架虚拟相机,然后按照固定角度旋转进行多角度拍照,获得多幅图像. 与此同时,保存 2D 图像中像素点和 3D 室内点云信息的对应关系,以备 2D 到 3D 的重建.

Step2. 门窗范围粗检测. 获取门窗的图像后,需要对图像中的门窗进行目标检测. 考虑检测精度和检测效率,本文采用深度学习的算法——YOLOv2<sup>[20]</sup>神经网络来进行门窗图像的目标检测. 将获取的门窗图像作为神经网络的输入,并获取门窗的检测框,得到门窗粗略的范围;再利用 2D 与 3D 的对应关系,就能获取有关门窗的局部点云数据.

Step3. 门窗位置精检测. 对局部点云数据提取轮廓线后,对所有轮廓线交点做优化得到门窗特征角点的位置信息,这样就实现了逐步精细的检测过程.

### 3 关键算法实现过程

现在采集点云数据的设备如 FARO Freestyle (3D) X 手持式激光扫描仪、FARO Focus3D 激光扫描仪等越来越齐全,获取数据的精度也越来越高. 但在扫描过程中,不可避免地会引入一些噪声,并产生实物遮挡问题,这些都是导致室内 3D 场景重建困难的原因. 因此,需对获取的数据进行清洗、去噪等,以获得质量较高的点云数据<sup>[21]</sup>.

本文采用“整体-局部”的策略来完成门窗检测,将检测范围从整个室内场景转换为定位到可能出现门窗的局部数据(称为目标范围的粗检测);再基于局部数据得到精确的门窗位置,这样就降低了重建算法的计算复杂度. 具体做法是:将 3D 点云数据的位置信息转换为 2D 图像的目标检测,然后再将 2D 检测结果信息逆转回 3D,得到相应局部信息数据;接着对局部数据进行处理,以获取最终特征角点信息.

对应室内 3D 重建算法,本文做了一些基本的假设:

- (1) 墙面是平面结构,而非曲面;
- (2) 由于在室内进行扫描时,不停地进出房间,所以门保持打开状态;因此,门框的位置就是门的实际位置;

(3) 由于扫描玻璃时会导致获取点云数据的密度不均匀且有缺失;因此,对于窗户的位置信息,本文认为窗户玻璃部分的点云数据是没有的,即窗户的点云信息是由窗框组成的;

- (4) 门、窗都是矩形形状.

#### 3.1 3D 与 2D 的对应

为了计算方便,实验中所有的室内点云数据都已“摆正”,即将房屋点云平移至世界坐标系原点,相邻墙面分别平行于坐标轴  $x$  轴、 $y$  轴,方向向上( $z$  轴),如图 2 所示. 接下来,本文将实现 3D 与 2D 的对应.

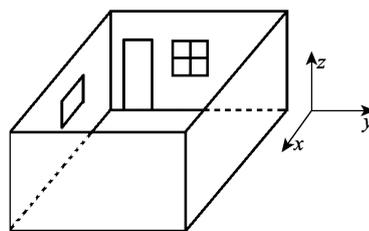


图 2 房屋点云“摆正”

本文在房间的中心设置一个虚拟相机进行多角度旋转,拍摄得到多幅有关门窗信息的点云图像;同时,记录包含相应相机参数的转换矩阵,作为 2D 与 3D 的对应关系.

室内点云的部分点在屏幕上出现的过程,实际上是世界坐标系逐步向观察坐标系、透视投影坐标系和屏幕坐标系的一系列变换过程. 为方便起见,本文记室内点云为  $C = \{p_i, i = 1, \dots, n\}$ , 其坐标(属于世界坐标系)信息为

$$[X_{\text{world}}, Y_{\text{world}}, Z_{\text{world}}]^T = \left\{ (x_{\text{world}_i}, y_{\text{world}_i}, z_{\text{world}_i})^T, i = 1, \dots, n \right\}.$$

### 3.1.1 世界坐标系与观察坐标系之间的转换

世界坐标系  $[X_{\text{world}}, Y_{\text{world}}, Z_{\text{world}}]^T$  向观察坐标系  $[X_{\text{view}}, Y_{\text{view}}, Z_{\text{view}}]^T$  进行转换的实质是 2 个坐标系间的平移旋转: (1) 平移观察坐标系原点到世界坐标系原点; (2) 进行旋转, 让观察坐标系的坐标轴分别与世界坐标系的坐标轴对应. 本文记该变换矩阵为  $M_{\text{RT}}$ , 则有

$$\begin{pmatrix} x_{\text{view}} \\ y_{\text{view}} \\ z_{\text{view}} \\ 1 \end{pmatrix} = \begin{pmatrix} R_{3 \times 3} & T_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & \mathbf{1} \end{pmatrix} \cdot \begin{pmatrix} x_{\text{world}} \\ y_{\text{world}} \\ z_{\text{world}} \\ 1 \end{pmatrix} = M_{\text{RT}} \cdot \begin{pmatrix} x_{\text{world}} \\ y_{\text{world}} \\ z_{\text{world}} \\ 1 \end{pmatrix} \quad (1)$$

其中,  $R_{3 \times 3}$  是做旋转变换的旋转矩阵;  $T_{3 \times 1}$  是做平移变换的平移矩阵.

### 3.1.2 观察坐标系与透视投影坐标系之间的转换

通过式(1), 世界坐标系中的对象点云就变换到观察坐标系中; 然后将其投影到观察平面上, 进入规范化的投影坐标系  $[X_{\text{pro}}, Y_{\text{pro}}, Z_{\text{pro}}]^T$ . 本文选用真实感较好的透视投影变换. 本文通过预置参数——投影参考点为原点, 观察平面在近裁剪平面、视场角(即相机能够观测到的最大范围的夹角)的大小  $\theta$ 、裁剪窗口的纵横比  $k_{\text{aspect}}$  以及对称的视锥体, 以确定规范化的透视投影变换矩阵

$$M_{\text{pro}} = \begin{pmatrix} \cot \frac{\theta}{2} & 0 & 0 & 0 \\ k_{\text{aspect}} & & & \\ 0 & \cot \frac{\theta}{2} & 0 & 0 \\ 0 & 0 & \frac{z_{\text{far}} + z_{\text{near}}}{z_{\text{far}} - z_{\text{near}}} & \frac{2z_{\text{far}} \cdot z_{\text{near}}}{z_{\text{far}} - z_{\text{near}}} \\ 0 & 0 & -1 & 0 \end{pmatrix}.$$

其中,  $z_{\text{far}}$  和  $z_{\text{near}}$  分别是远近裁剪平面在观察坐标系中  $z_{\text{view}}$  轴上的位置.

### 3.1.3 透视投影坐标系与屏幕坐标系之间的转换

完成规范化透视投影后, 点云信息已进入规范化的裁剪窗口. 下面把规范化窗口的内容变换到屏幕坐标, 这实际上是一个 2D 坐标变换过程, 变换矩阵为

$$M_{\text{pro2screen}} = \begin{pmatrix} \frac{u_{\text{max}} - u_{\text{min}}}{2} & 0 & 0 & \frac{u_{\text{max}} + u_{\text{min}}}{2} \\ 0 & \frac{v_{\text{max}} - v_{\text{min}}}{2} & 0 & \frac{v_{\text{max}} + v_{\text{min}}}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

其中,  $u_{\text{max}}, u_{\text{min}}, v_{\text{max}}, v_{\text{min}}$  分别是视口区域在坐标轴  $u$  上最大值和最小值, 以及在坐标轴  $v$  上的最大值和最小值.

综上, 世界坐标系到屏幕坐标系之间的转换矩阵为  $M_{\text{transform}} = M_{\text{pro2screen}} \cdot M_{\text{pro}} \cdot M_{\text{RT}}$ . 因此, 本文通过保存转换矩阵  $M_{\text{transform}}$  实现后期局部点云的确定.

## 3.2 门窗范围粗检测

在目标粗检测领域, 基于深度学习的目标检测算法主要分为 Region-Proposal 和 End-to-End. 前者需要先对原始图像进行特征提取, 找到可能含有目标的候选框; 然后再结合 CNN 进行检测, 输出目标类别和位置; 后者是输入原始图像到网络中, 直接输出图像中所有目标类别和位置. 从便捷性和检测效能上来看, 本文采用端对端的算法来直接获得检测结果, YOLO 正是这种算法的典型代表. 本文采用的是 YOLOv2 神经网络, 是 YOLO 的改进版, 其在速度和精度上都达到了较好的平衡.

将含有门窗信息的室内点云场景图像作为 YOLOv2 的输入, 输出为类别标签和包围框信息. 在得到图像中标定的门窗大致范围( $r_{\text{left}}, r_{\text{right}}, r_{\text{top}}, r_{\text{bot}}$ )之后, 根据房屋点云坐标  $[X_{\text{world}}, Y_{\text{world}}, Z_{\text{world}}]^T$  和转换矩阵  $M_{\text{transform}}$ , 确定包围框中所有像素对应的 3D 点信息, 得到局部点云数据  $C_{\text{local}}$ .

## 3.3 门窗范围精检测

在粗略得到包含门窗的局部点云后, 再对此局部数据进行优化处理得到门窗的精确位置信息.

首先用 RANSAC 算法拟合出该局部点云的平面(实际上, 这个平面是墙面); 然后将整个局部门窗点云  $C_{\text{local}}$  向这个平面进行投影, 得到投影点云在这个投影面上的轮廓点; 接着对这些轮廓点做拟合, 得到轮廓线候选集  $C'$ , 即

$$\forall l_1, l_2 \in C', \text{ 有} \begin{cases} l_1, l_2 \text{ 合并} & \text{if } (d(l_1, l_2) < \varepsilon \ \& \ \mathbf{v}_{l_1} * \mathbf{v}_{l_2} < \delta) \\ l_1, l_2 \text{ 掰直} & \text{if } (|\mathbf{v}_{l_1} * \mathbf{v}_{l_2} - 1| < \delta) \end{cases}.$$

此时对所有轮廓线求交, 得到特征点候选集  $F_C = \{p_i, i = 1, \dots, n\}$ ; 分别基于门和窗对这些特征

点候选集合做优化处理, 以得到真正的特征角点. 具体流程如算法 1 所示.

算法 1. 步骤如下:

输入. 特征点候选集  $F_C$ , 图像中门或窗包围框分类标签(label).

输出. 特征角点集合  $S_p$ .

Step1. 计算  $F_C$  中每 4 个点构成的四边形面积  $S_{ijkl}$ , 并排序, 确定面积最大的 4 个点  $p_a, p_b, p_c, p_d$ .

Step2. 若  $label="door"$ , 则对  $F_C$  进行  $k$ -means 聚类, 得到 4 个簇; 然后在每个簇筛选出到其他簇中所有点的距离最短的点作为特征点, 归入特征角点集合  $S_p$ . 其中,  $k$ -means 初始聚类中心分别为  $p_a, p_b, p_c, p_d$ .

Step3. 若  $label="window"$ , 则对  $F_C$  中所有点进行筛选, 剔除和点  $p_a, p_b, p_c, p_d$  在同一条直线上的所有点;  $F_C$  中剩下的点即构成特征角点集合  $S_p$ .

Step4. 返回特征角点集  $S_p$ .

图 3 展示了从门的局部点云数据中得到特征角点的过程. 其中, 图 3a 和图 3c 分别是门的局部点云和门的特征角点; 在图 3b 中, 蓝色的点表示轮廓线交点, 黄色点表示面积最大的 4 个交点, 红色点表示门真正的特征角点. 首先确定局部点云的轮廓线, 然后得到所有轮廓线交点; 接着, 将其中构成四边形面积最大的 4 个交点作为引导项, 引导完成  $k$ -means 聚类, 形成 4 类点集合; 最后在这 4 类点集合中确定真正的特征角点.

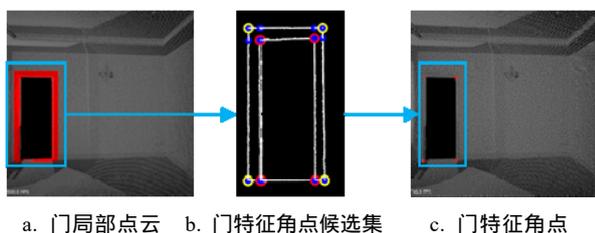


图 3 门的特征角点确定

图 4 展示了从窗的局部点云数据中得到特征角点的过程. 其中, 图 4a 和 4c 分别是窗的局部点云和窗的特征角点; 在图 4b 中, 蓝色的点表示轮廓线交点, 黄色点表示面积最大的 4 个交点, 绿色点表示窗真正的特征角点. 首先确定局部点云的轮廓线, 然后得到所有轮廓线交点; 接着, 将其中构成四边形面积最大的 4 个交点作为引导项, 完成最外围边界轮廓线及其相关交点的删除工作, 使得只剩下内边界的轮廓线和真正的特征角点.

对单个房间点云数据而言, 通过由粗到细的门窗检测过程, 得到了门窗位置的特征角点, 从而完成门窗具体位置的有效检测.

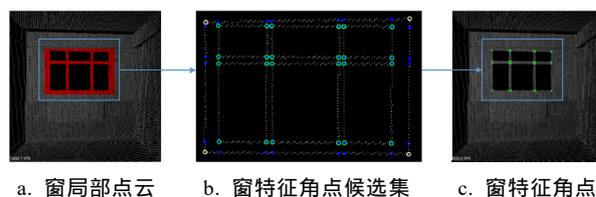


图 4 窗的特征角点确定

## 4 实验及结果分析

本文算法的主要代码在 Windows10 系统中利用 VS2015 实现, 其中 YOLOv2 的训练是在 Ubuntu 系统中基于 Darknet 框架下进行; 在实现过程中用到的库主要有 PCL(point cloud library), OpenCV 等.

### 4.1 数据类型与来源

在实验中, 本文主要采用室内场景 3D 点云数据和门窗图像 2D 数据.

室内场景点云数据集包含 2 个部分. 一个来自广州思勘测绘技术有限公司, 共 157 个室内场景点云数据, 包括酒店、民居、学校行政楼等建筑物; 另一个是来自斯坦福大学的大型 3D 室内场景数据集<sup>[22]</sup>, 并选取其中 153 个室内场景点云数据.

实验中的门窗 2D 图像数据来自室内 3D 点云数据. 相机视场角为  $90^\circ$ , 拍照旋转间隔为  $18^\circ$ , 图像大小为  $448 \times 448$ . 图像数据集一共 5000 幅, 其中, 4500 幅作为训练集, 500 幅作为验证集使用.

### 4.2 误差定义

本文采用人工标注的方式得到样本门窗角点的真实位置, 再用算法 1 的计算结果与真实值之间的欧氏距离最大误差和平均误差来评估本文算法的有效性. 这 2 个函数分别定义如下:

最大误差函数

$$e_{\max} = \max \{ \|g_i - c_i\|, i = 1, \dots, N \};$$

平均误差函数

$$e_{\text{avg}} = \frac{\sum_{i=1}^N \|g_i - c_i\|}{N}.$$

其中,  $\| \cdot \|$  为 2 点之间的欧氏距离;  $g$  表示真实特征角点集;  $c$  表示本文算法计算的特征角点集;  $N$  表示特征角点数.

### 4.3 结果分析

由于在观测角偏转较大时, 门窗在图像中显示会有变形现象, 返回到 3D 点云数据上时, 门窗的局部点云会呈现非矩形形状. 本文的局部特征角点检测算法对这种情况具有较好的鲁棒性. 表 1

为在同一视场角(算法实验中是 90°)下的不同观测角度的检测结果, 图 5 是相应的误差结果图.

表 1 不同观测角的检测结果

观测角	2D 图像	目标检测	局部点云	特征角点
正面右偏 0°				
右偏 18°				
右偏 36°				
右偏 54°				

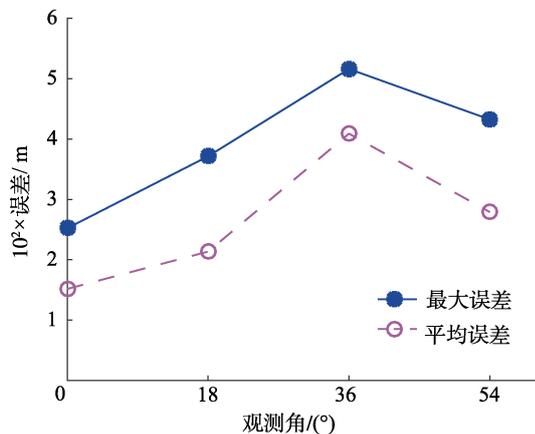


图 5 不同观测角的误差对比结果

由图 5 的对比结果显示, 算法 1 对不同观测角度具有较好的鲁棒性, 其中在正面 0° 的情况下, 误差最小.

表 2 所示为在门窗数据部分缺失的情况下, 门窗特征角点的检测结果; 表 3 是相应的误差结果. 表 4 显示了算法 1 在不同模型上的效果. 第 1 列是对应门窗的 2D 图像, 第 2 列是 YOLOv2 目标检测结果, 第 3 列是对应的局部点云数据, 其中红色点和绿色点分别对应门和窗的局部点云结果, 第 4 列是门窗最终特征角点信息. 值得注意的是, 其中模型 #7 的 III 号门的边界特征角点并没有完全被检测出来, 这是由于在拍摄时, III 号门仅部分出现在图像中, 导致在返回 3D 数据时, 仅能获取一部分门的点云数据, 从而也就只能提取出一部分轮廓线, 影响后续特征点检测. 表 5 所示为相应的误差结果.

表 2 数据缺失情况下不同模型的检测结果

模型	2D 图像	目标检测	局部点云	特征角点
#2				
#3				
#4				

表 3 数据缺失情况下不同模型的误差

模型	$e_{max}$	$e_{avg}$
#2	0.016	0.009
#3	0.045	0.023
#4	0.039	0.026

表 4 不同模型上的检测结果

模型	2D 图像	目标检测	局部点云	特征角点
#5				
#6				
#7				

表 5 不同模型上的误差结果

模型	$e_{max}$	$e_{avg}$
#5	0.038	0.019
#6	0.036	0.020
#7	1.195	0.216
#7(不含 III 门)	0.044	0.019

本文还对比了不同相机视场角下算法的结果, 对比结果如表 6 所示. 表 6 展示了算法 1 在同一观测角度(180°)的不同视场角的检测结果. 图 6 所示为相应的误差对比图, 对比结果显示, 在视场角为 90° 时, 检测效果最好.

表 6 不同视场角的检测结果

视场角	2D 图像	目标检测	局部点云	特征角点
60°				
70°				
80°				
90°				
100°				

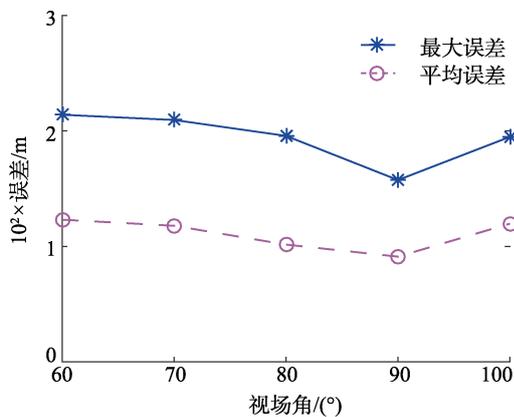


图 6 不同视场角的误差对比结果

## 5 结 语

通过采用“整体-局部”的策略,借助图像目标检测的特点,本文提出了一种基于图像自动检测 3D 室内场景中门窗特征角点的算法.该算法利用包含完整门窗信息的图像进行目标检测,将所检测的门窗包围框信息返回到点云数据上,获得局部门窗点云数据.在局部点云的门窗特征角点检测阶段,首先提取局部特征轮廓线,然后在所有轮廓线交点中提取出“最外围”4 个交点;接着分别对门和窗提取特征角点.若局部点云标签是门,则将此 4 个交点作为  $k$ -means 聚类的初始聚类中心,聚类结束后在每一簇点中找到特征角点;若局部点云标签是窗,则将此 4 个交点作为引导项来删除外

围轮廓线,留下窗的特征轮廓线及其真正特征角点.

实验结果表明,提出的特征角点检测算法能有效地检测出 3D 室内场景中门窗的特征角点.但若图像中不包含完整门窗信息,将影响后续特征点的检测;如若有关橱、箱柜、窗帘等物体大面积遮挡住门窗,当去除这些物体时,会造成门窗数据缺失严重,则无法正确地检测出门窗信息的特征角点.

本文下一步工作将利用拍照角度的前后关联性,改进门窗特征角点检测,补充优化不完整门窗信息导致特征点检测失败的结果.

## 参考文献(References):

- [1] Kelly T, Femiani J, Wonka P, *et al.* BigSUR: large-scale structured urban reconstruction[J]. *ACM Transactions on Graphics*, 2017, 36(6): Article No.204
- [2] Yi C, Zhang Y, Wu Q Y, *et al.* Urban building reconstruction from raw LiDAR point data[J]. *Computer-Aided Design*, 2017, 93: 1-14
- [3] Nan L L, Wonka P. PolyFit: polygonal surface reconstruction from point clouds[C] // *Proceedings of the IEEE International Conference on Computer Vision*. Los Alamitos: IEEE Computer Society Press, 2017: 2372-2380
- [4] Xiao J X, Furukawa Y. Reconstructing the world's museums[J]. *International Journal of Computer Vision*, 2014, 110(3): 243-258
- [5] Monszpart A, Mellado N, Brostow G J, *et al.* RAPter: rebuilding man-made scenes with regular arrangements of planes[J]. *ACM Transactions on Graphics*, 2015, 34(4): Article No.103
- [6] Chen Guangming, Li Guiqing, Liu Pei, *et al.* Hierarchical constraints with particle swarm optimization for furniture arrangement[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2014, 26(10): 1603-1613(in Chinese)  
(陈光明, 李桂清, 刘培, 等. 家居布局的层次化约束及其粒子群优化[J]. *计算机辅助设计与图形学学报*, 2014, 26(10): 1603-1613)
- [7] Quintana B, Prieto S A, Adán A, *et al.* Door detection in 3D colored laser scans for autonomous indoor navigation[C] // *Proceedings of International Conference on Indoor Positioning and Indoor Navigation*. Los Alamitos: IEEE Computer Society Press, 2016: 1-8
- [8] Quintana B, Prieto S A, Adán A, *et al.* Door detection in 3D coloured point clouds of indoor environments[J]. *Automation in Construction*, 2018, 85: 146-166
- [9] Kakillioglu B, Ozcan K, Velipasalar S. Doorway detection for autonomous indoor navigation of unmanned vehicles[C] // *Proceedings of the IEEE International Conference on Image Processing*. Los Alamitos: IEEE Computer Society Press, 2016: 3837-3841
- [10] Chen W, Qu T, Zhou Y M, *et al.* Door recognition and deep learning algorithm for visual based robot navigation[C] // *Proceedings of the IEEE International Conference on Robotics and Biomimetics*. Los Alamitos: IEEE Computer Society

- Press, 2014: 1793-1798
- [11] Derry M, Argall B. Automated doorway detection for assistive shared-control wheelchairs[C] //Proceedings of the IEEE International Conference on Robotics and Automation. Los Alamitos: IEEE Computer Society Press, 2013: 1254-1259
- [12] Yuan T H, Hashim F H, Zaki W M D W, *et al.* An automated 3D scanning algorithm using depth cameras for door detection[C] //Proceedings of International Electronics Symposium. Los Alamitos: IEEE Computer Society Press, 2016: 58-61
- [13] Zhou Y M, Jiang G L, Xu G Q, *et al.* Kinect depth image based door detection for autonomous indoor navigation[C] //Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication. Los Alamitos: IEEE Computer Society Press, 2014: 147-152
- [14] Dai D W, Jiang G L, Xin J B, *et al.* Detecting, locating and crossing a door for a wide indoor surveillance robot[C] //Proceedings of the IEEE International Conference on Robotics and Biomimetics. Los Alamitos: IEEE Computer Society Press, 2013: 1740-1746
- [15] Borgsen S M Z, Schopfer M, Ziegler L, *et al.* Automated door detection with a 3D-sensor[C] //Proceedings of Canadian Conference on Computer and Robot Vision. Los Alamitos: IEEE Computer Society Press, 2014: 276-282
- [16] Schnabel R, Wahl R, Klein R. Efficient RANSAC for point-cloud shape detection[J]. Computer Graphics Forum, 2007, 26(2): 214-226
- [17] Rusu R B, Meeussen W, Chitta S, *et al.* Laser-based perception for door and handle identification[C] //Proceedings of International Conference on Advanced Robotics. Los Alamitos: IEEE Computer Society Press, 2009: 1-8
- [18] Ochmann S, Vock R, Wessel R, *et al.* Automatic reconstruction of parametric building models from indoor point clouds[J]. Computers & Graphics, 2016, 54: 94-103
- [19] Wang J, Xu Y B, Remil O, *et al.* Automatic modeling of urban facades from raw LiDAR point data[J]. Computer Graphics Forum, 2016, 35(7): 269-278
- [20] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2017: 6517-6525
- [21] Zheng Y L, Li G Q, Wu S H, *et al.* Guided point cloud denoising via sharp feature skeletons[J]. The Visual Computer, 2017, 33(6-8): 857-867
- [22] Armeni I, Sener O, Zamir A R, *et al.* 3D semantic parsing of large-scale indoor spaces[C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos: IEEE Computer Society Press, 2016: 1534-1543