

适用于无人驾驶车辆的地图构建与道路提取算法研究

于华超

2018 年 3 月

中图分类号 : TP391.41

UDC 分类号 : 621.3

适用于无人驾驶车辆的地图构建与道路提取算法研究

作 者 姓 名	<u>于华超</u>
学 院 名 称	<u>自动化学院</u>
指 导 教 师	<u>王美玲</u>
答 辩 委 员 会 主 席	<u>孙长胜教授</u>
申 请 学 位 级 别	<u>工学硕士</u>
学 科 专 业	<u>控制科学与工程</u>
学 位 授 予 单 位	<u>北京理工大学</u>
论 文 答 辩 时 间	<u>2018 年 3 月</u>

Research on map building and road extraction algorithm for intelligent vehicles

Candidate Name: Hao Li

School or Department: School of Automation

Faculty Mentor: Yi Yang

Chair, Thesis Committee: Prof. Changhai Sun

Degree Applied:

Major: Control Science and Engineering

Degree by: Beijing Institute of Technology

The Date of Defence: March, 2018

适用于无人驾驶车辆的地图构建与道路提取算法研究

北京理工大学

研究成果声明

本人郑重声明：所提交的学位论文是我本人在指导教师的指导下进行的研究工作获得的研究成果。尽我所知，文中除特别标注和致谢的地方外，学位论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京理工大学或其它教育机构的学位或证书所使用过的材料。与我一同工作的合作者对此研究工作所做的任何贡献均已在学位论文中作了明确的说明并表示了谢意。

特此申明。

签名：

日期：

关于学位论文使用权的说明

本人完全了解北京理工大学有关保管、使用学位论文的规定，其中包括：①学校有权保管、并向有关部门送交学位论文的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存学位论文；③学校可允许学位论文被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换学位论文；⑤学校可以公布学位论文的全部或部分内容（保密学位论文在解密后遵守此规定）。

签名：

日期：

导师签名：

日期：

摘要

这里是摘要。

关键词:

Abstract

This is abstract.

Keywords:

目录

第1章 绪论 ······	1
1.1 研究背景与意义 ······	1
1.2 国内外研究现状 ······	2
1.2.1 地图构建方法研究 ······	2
1.2.2 道路提取方法研究 ······	2
1.3 主要研究内容 ······	2
1.3.1 二维与三维地图构建 ······	2
1.3.2 二维图像中的道路提取 ······	2
1.4 本章小结 ······	2
第2章 基于全景图像的高精度地图构建 ······	3
2.1 引言 ······	3
2.2 全景图像与定位信息采集 ······	3
2.3 全景图像处理 ······	3
2.4 地图的构建与使用 ······	7
2.4.1 图像拼接 ······	7
2.4.2 基于GIS数据库生成无人车行驶地图 ······	8
2.5 本章小结 ······	9
第3章 基于无人机航拍图像的三维构建 ······	10
3.1 引言 ······	10
3.2 相机模型 ······	10
3.3 相关坐标系 ······	12
3.4 特征点提取 ······	16
3.5 特征点匹配 ······	20
3.6 多视图重建 ······	25
3.6.1 两视图重建 ······	25
3.6.2 三角定位法 ······	30
3.6.3 2D-3D位姿求解 ······	31
3.7 三维重建中的优化 ······	33
3.7.1 光束平差法 ······	34
3.7.2 最小化相机中心位置误差 ······	37

3.8 稀疏点云的渲染 ······	39
3.8.1 点云稠密化 ······	39
3.8.2 点云网格渲染（三角剖分） ······	39
3.9 本章小结 ······	39
第 4 章 二维图像中的道路提取 ······	40
4.1 引言 ······	40
4.2 图像二值化 ······	40
4.3 非道路区域移除 ······	40
4.3.1 开闭运算 ······	42
4.3.2 轮廓提取 ······	43
4.4 拓扑处理 ······	44
4.4.1 拓扑细化 ······	44
4.4.2 去除小枝丫 ······	45
4.5 本章小结 ······	46
第 5 章 实验结果分析 ······	47
5.1 引言 ······	47
5.2 全景图像构建地图实验结果 ······	47
5.2.1 全景图像地图实验平台搭建 ······	47
5.2.2 全景图像地图构建效果及精度分析 ······	47
5.3 地理空间三维重建实验结果 ······	47
5.3.1 地理空间三维重建实验平台搭建 ······	47
5.3.2 三维重建效果及精度分析 ······	47
5.4 道路提取精度分析 ······	47
5.5 本章小结 ······	47
总结与展望 ······	50
参考文献 ······	51
攻读硕士学位期间发表论文与研究成果清单 ······	53

第1章 绪论

1.1 研究背景与意义

近年来智能移动机器人变得越来越普及，这些智能化的地面无人车辆可以独立或辅助人类完成很多任务，这依赖其对环境的感知、导航和运动规划等过程。地面无人车辆在未知区域导航时，对感知、规划等的要求非常高，所以一般的无人车辆都依赖详细的地图进行导航规划^[5]，完成既定任务。而易于获取的卫星图像，由于更新不及时，室内无法使用，精度难以满足车道线级别的精度要求等原因一般不被地面高精度的无人系统采用。目前移动机器人的构建高精度地图方法较多，一般采用其自身车载传感器等设备，总体上可以分为基于摄像头和基于激光雷达两种方案。

实时定位与地图构建(SLAM,Simultaneous Localization and Mapping)作为移动机器人构建地图的方法今年来获得长足发展，较为著名的有ORB-SLAM^[1]，LSD-SLAM^[2]，LOAM^[3]等，图 1.1是运用车载传感器得到SLAM地图，其中左图使用激光雷达，右图使用摄像头得到的三维地图。这种采用车载传感器的构图方法已经发展地较为成熟，但是运用车载传感器得到大范围地图仍有困难，例如车载传感器视野受限，易受遮挡，在灾区发生道路阻塞时，车辆无法行驶而无法采集道路信息，甚至使用车辆对所有道路遍历的方法效率低下。然而无人机飞行灵活，视野宽阔，可以高效地对地面拍摄并建立地理空间地图，为无人地面车辆行驶提供先验信息。

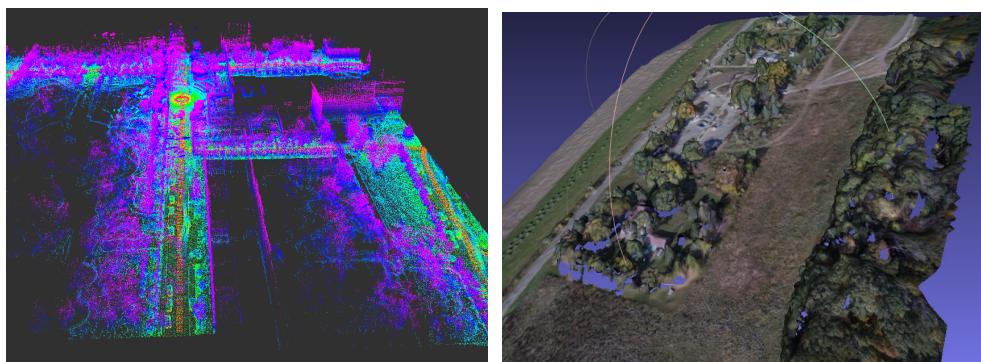


图 1.1 高精度地图

与视觉SLAM类似，SfM(Structure from Motion)也是一种构建三维地形的技术，SfM能通过2D图像得到稠密3D点云，与传统的飞行器激光扫描(ALS,Airborne Laser

Scanning)或者(LiDAR,airbone light detectiona and ranging)绘制地图, SfM可得到更高分辨率的3D模型, 包括纹理、地形等, 更重要的是SfM得到如此丰富的数据结果, 使用的设备更廉价。SfM使用算法匹配图像特征点, 计算相机位姿, 从而重建拍摄场景的“稀疏”3D点云, 通过MVS(Multi-View Stereo)将点云稠密化而后渲染, 得到逼真的三维地理空间模型^[4], 基于此SfM作为一种地形测绘技术被广泛应用于地理科学中。我们可以通过学术文献数据库Web of Knowledge观察SfM的学术研究情况, 从20世纪80年代至2015年共有1000条关于Structure from Motion主题的记录, 其中属于计算机科学范畴的最多, 地理科学占第9名^[4], 如图所示。

SfM更多使用无序的图片, 运行过程没有时间限制, 构建地图更加详细

1.2 国内外研究现状

1.2.1 地图构建方法研究

1.2.2 道路提取方法研究

1.3 主要研究内容

1.3.1 二维与三维地图构建

1.3.2 二维图像中的道路提取

1.4 本章小结

第2章 基于全景图像的高精度地图构建

2.1 引言

全景图像（panorama）是指水平角度包含完整 360° 的相机得到的二维平面图像。相对于普通单目相机，全景相机可以完整的呈现无人车周围环境。在本章节中，讨论使用全景相机构建车道级别的用于无人车行驶的高精度地图的方法。首先介绍了全景图像与定位信息的采集，而后介绍了全景相机的标定与全景图像预处理，最后阐述了如何通过全景图像构建大范围的高精度地图的方法。

2.2 全景图像与定位信息采集

为了同时采集全景图像信息与高精度定位导航系统信息，本文采用将车辆的定位信息编码为大小一定的QR code (Quick-Response code)，放置于每帧全景图像左上角位置，完成系统的信息采集任务。在此介绍一下二维码的编码与解码过程。由于全景图像的帧率为 7Hz ，惯性导航系统的帧率为 20Hz ，考虑拼接图像的精度，在没有使用硬件触发的条件下，本文采用最近邻原则，即当接受到一帧全景图像时，编码离当前时刻最近的惯性导航定位信息。由于编码的信息有时间，经纬度，方位角，偏航角与俯仰角6个整数，所以需要的编码信息量很小，本文只采用 25×25 像素的二维码编码定位信息。

QR code作为一种非常流行的矩阵式二维码，具有空间利用率高，存储数据量大，解码速度快，可以包含字符，数字等不同内容等优点。如图 2.1所示，二维码可分为两部分：功能区和编码区。功能区包含用于二维码定位的位置探测区域，用于校正二维码位置的校正图形区域等，在编码区中，包含数据和纠错码等。二维码的纠错级别越高，则实际存储的数据信息就越少。按照QR code的规则，可以很方便的为定位信息编码为固定像素大小的二维码，进而将该大小的区域放置在每帧全景图像的左上角，完成信息的采集过程。当回放该视频时，可以很容易地对大小固定的编码区进行解码，读取到每帧全景图对应的车辆当前的位置与角度信息等，为图像拼接做准备。

2.3 全景图像处理

Ladybug全景相机具有6个独立的摄像头，如图 2.2，可以在球形坐标系下将6个相机图像实时合并为一张全景图，呈现在二维平面空间中。ladybug相机提供 360°

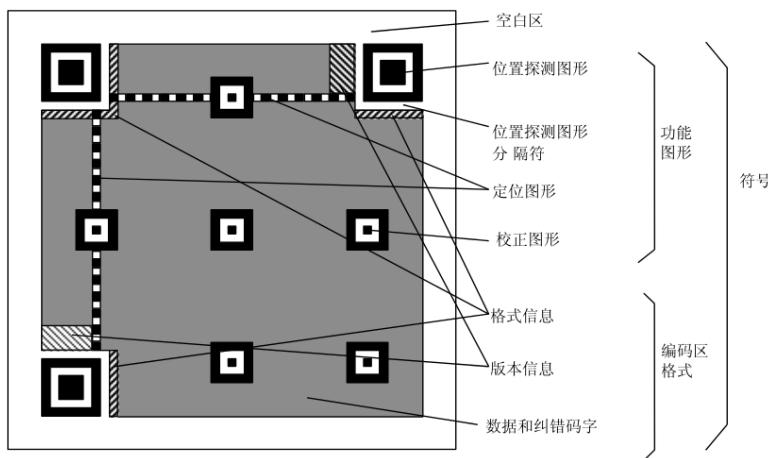


图 2.1 QR code示意图

视频串流功能，可以覆盖90%的可视球面具有图像采集、处理和校正功能。在得到Ladybug相机的全景图后，使用逆透视投影（IPM）将全景图像投影成俯视图。而后将车辆的高精度定位信息编码为二维码，放置于对应的逆透视投影得到的图片的左下角，从而完成图像与对应车体位置的采集与处理，方便后续构建大范围地图。



图 2.2 Ladybug全景相机

由于LadyBug成像在球面坐标系中描述，IPM投影需要在车体坐标系中描述，地图拼接需要在世界坐标系中完成，在此简单介绍全景图像处理过程涉及的相关坐标系的变换关系。如图 2.3所示，代表全景相机的球面坐标系与车体坐标系的关系，其中在车体坐标系中，绕自身x, y和z旋转，分别代表俯仰角，横滚角和方位角，使用道路方向形象地代表车辆的行驶方向；在全景坐标系中，球面狭缝与全景图分块成对应关系，如图中虚线箭头所示，得到的全景图的长宽，如图 2.4所示，图片长

宽分别代表相对于车体的方位角 μ 与俯仰角 ν 。为了得到全景图像的IPM俯视图，需要标定全景相机相对于车辆的外参：俯仰角 θ ，横滚角 ϕ 和方位角 φ ，以及Ladybug相机光心与地面的高度 h ，同时假定车辆周围的环境与地面高度一致。现定义车体坐标系为 $\{F_\omega\} = \{X_\omega, Y_\omega, Z_\omega\}$ ，坐标原点位于车辆后轴中点；定义全景相机坐标系为 $\{F_c\} = \{X_c, Y_c, Z_c\}$ ；定义全景图像坐标为 $\{F_i\} = \{\mu, \nu\}$ 。如图 2.3所示，车体坐标系中任一点与全景相机坐标系中的点的对应关系为：

$$P_c = R_\phi R_\theta R_\varphi^T (P_\omega + T) \quad (2.1)$$

其中

$$R_\phi = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \quad (2.2)$$

$$R_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad (2.3)$$

$$R_\varphi = \begin{bmatrix} \cos(\varphi - \frac{\pi}{2}) & \sin(\varphi - \frac{\pi}{2}) & 0 \\ -\sin(\varphi - \frac{\pi}{2}) & \cos(\varphi - \frac{\pi}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$T = \begin{bmatrix} 0 \\ 0 \\ -h \end{bmatrix} \quad (2.5)$$

进而可以得到全景图像坐标系中的坐标 $P_i = [\mu_i, \nu_i]^T$ ：

$$P_i = \begin{bmatrix} \arctan \frac{y_c}{x_c} \\ \arccos \frac{z_c}{\|P_c\|} \end{bmatrix} \quad (2.6)$$

其中 $\|P_c\| = \sqrt{X_{P_c}^2 + Y_{P_c}^2 + Z_{P_c}^2}$ ，即 P_c 的二范数。通过以上算法可以将全景图像（图 2.4）变换得到车体坐标系下的IPM图，显示效果如图 2.5所示

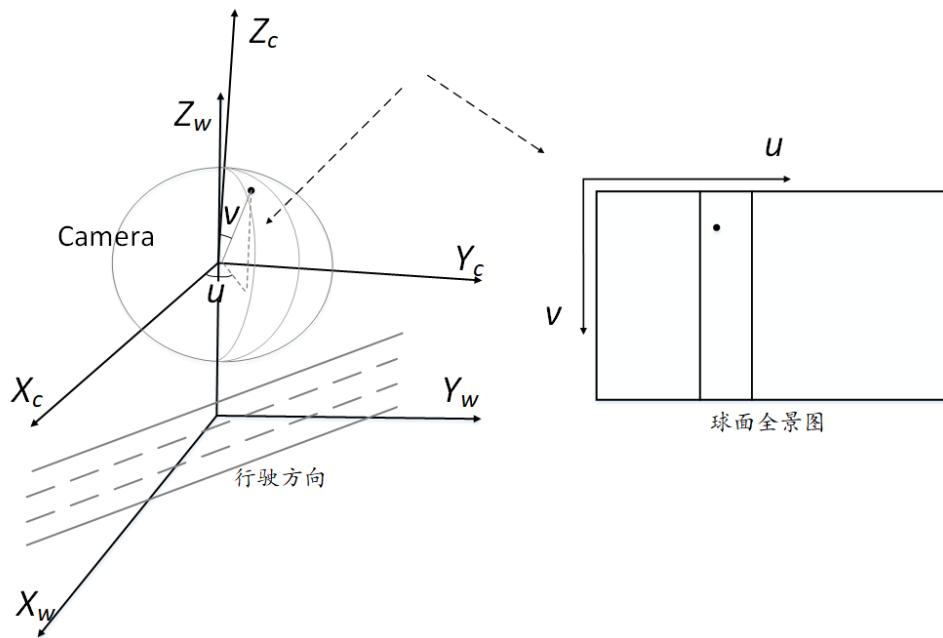


图 2.3 逆透视坐标变换关系



图 2.4 基于球面坐标的全景图

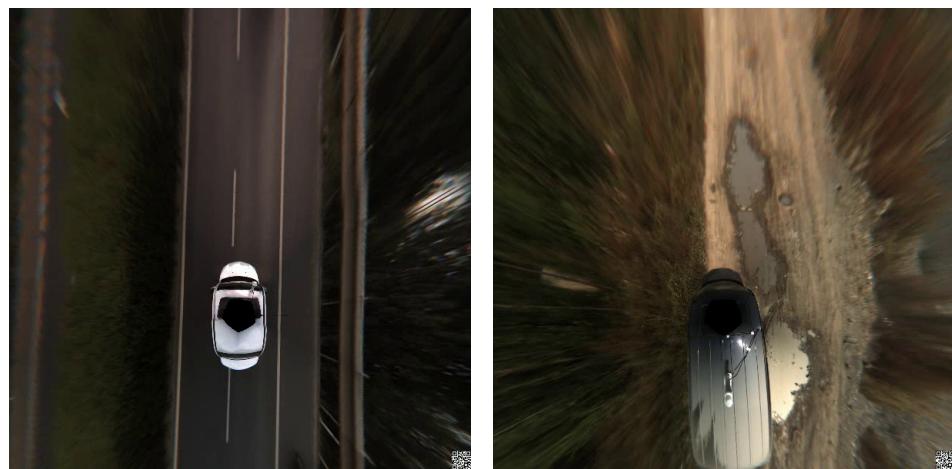


图 2.5 逆透视变换结果

2.4 地图的构建与使用

使用以上的方法可以得到全景图像的逆透视投影，以及每张投影图对应的此时车辆位置与方位等信息。进而我们可以得到大范围地图信息，制作大范围行驶地图，为无人车规划与感知提供先验信息。

2.4.1 图像拼接

图像拼接的历史可以追溯到上个世纪，使用的方法也不尽相同，总体上可以分为两类，一类是基于特征匹配的图像拼接方法，另一类是基于图片相互间位置关系的拼接方法。由于本文采用的惯性组合导航系统，在无遮挡，多路径效应较弱的空旷场地，实验精度达到 $5cm$ 以内，所以在此本文采用基于图片之间相互位置关系的方法。具体就是将视频流中每帧逆透视投影图片按照二维码解算的航向旋转一定角度，然后按照解算的位置平移一段距离，对所有视频按照上述方法处理，最终得到完整的城区地图。

对于每帧IPM俯视图，车身为无效信息且占据图像大量位置，对地图拼接造成干扰，所以在拼接地图前，需要将车身移除。由于全景图变换为IPM投影图时，将IPM俯视图变换到车体坐标系中，所以任何一张IPM俯视图的车辆位置与方位是固定的：车辆后轴中心位于图片横轴的中心，纵轴靠下的 $\frac{1}{3}$ 位置处，航向朝上。如图2.6所示。据此设计一个mask，在拼接过程中将车辆去掉，只拼接剩下的像素不为零的部分。

在拼接过程中，因为所需拼接的地理空间过大，图片过多，而内存限制了单张图片的大小，所以本文设计了区号标记的方法。具体做法为：将视频流中图片按照位置和航向拼接为多张 28000×7000 像素的图片，如图2.6中橙色方框所示，由于每个像素代表实际地理空间的大小为 $0.06m$ ，即每张拼接后的图片代表的物理空间大小为 $1680m \times 420m$ 。设置整个地图的坐标原点（图2.6红色圆点所示）为某一固定经纬度坐标，IPM图对应的经纬度与该点对比后，计算出该帧IPM图应该放置于哪个区，而后可以拼接成多张橙色大小的图片。一个橙色方框代表一个区域，对应一个区号，这种存在重叠区域的区号法则有效的解决了图像拼接过程中出现的边缘缺失的问题，所以即使橙色方框代表的区域出现边缘缺失现象，由于重叠区域的存在可以有效解决该问题。最后只需要保留图2.6的绿色框表示的区域和该区域的区号，这样就可以按照区号和图2.6的样式将拼接的图片在Photoshop中拼接为更大的图片，为后

续GIS数据库中构建无人车底图服务。

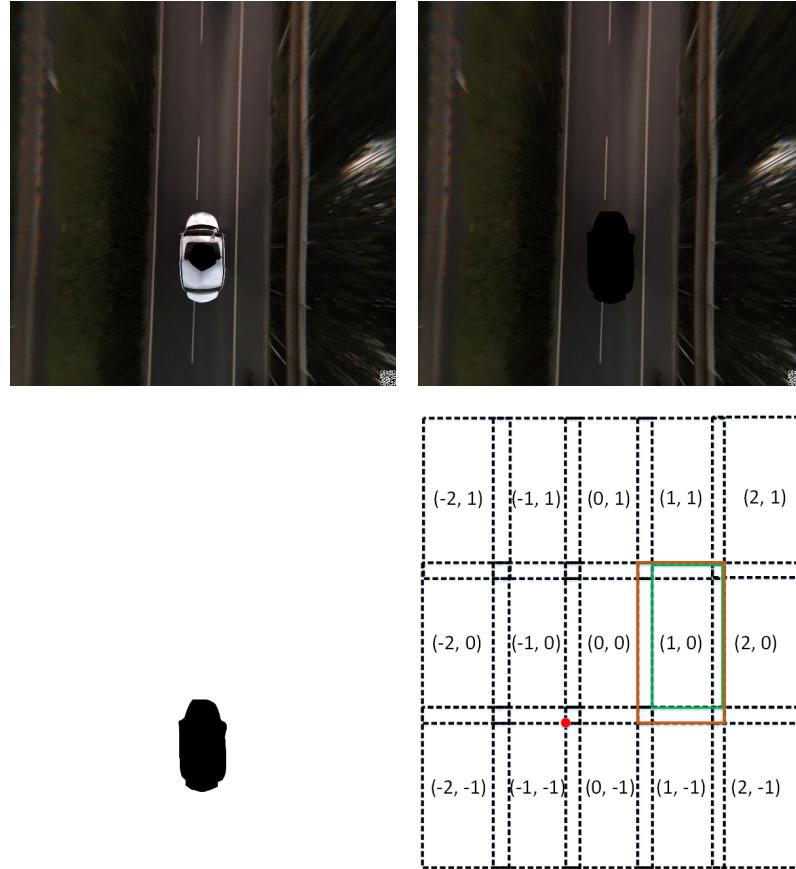


图 2.6 去掉车辆

2.4.2 基于GIS数据库生成无人车行驶地图

在得到数张Photoshop拼接的更大的图片的同时，按照图 2.6的分区原则也很容易得到每张图片四个顶点的地理坐标。依托Supermap(GIS)软件，可以很容易完成金字塔地图的生成，作为GIS数据库的一个图层。如图 2.7所示。这样构建的地图并不能直接用于无人车导航或规划，我们将得到的底图作为GIS数据库的一个图层，根据底图可以得到车道级别的道路拓扑结构，该拓扑作为GIS数据库的另一个图层，如图 2.7中的绿色线段，在线段与线段的节点上添加各种属性，例如车道数量，限速，红路灯和路口等，即可实现全局路径规划等功能，这些功能不是本文讨论的内容。

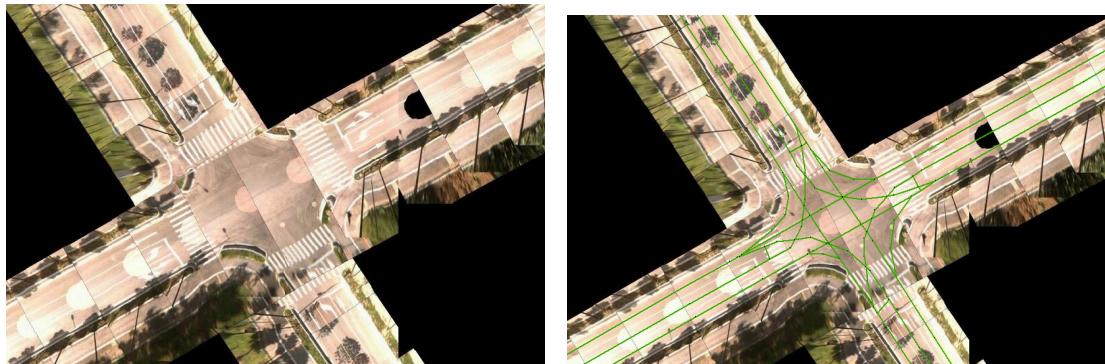


图 2.7 拼接的路口展示

2.5 本章小结

本章介绍了如何通过Ladybug全景相机生成的球面全景图与高精度定位导航组合，构建车道级别地图的过程。根据Ladybug与车体的固定位置关系，将全景图像旋转与平移转换到车体坐标系中，并完成逆透视投影的过程，并将每个IPM 投影的图像通过车辆的高精度定位导航信息，旋转平移拼接为完整的道路地图，最后将地图生成GIS数据库的图层，完成道路拓扑构建与道路属性添加的任务，即可将该地图用于无人驾驶导航与规划等任务中。

第3章 基于无人机航拍图像的三维构建

3.1 引言

大部分无人车辆行驶依赖环境地图导航与感知^[5]，这限制了无人车在未知环境中的行驶。为了解决该问题，当前许多技术利用车载传感器可以构建很大范围的地图，例如SLAM(Simultaneous Localization and Mapping) 和第二章提到的用车载Ladybug构建地图的方法。然而车载传感器仅能获取车辆周边环境信息且易被阻挡，而且在野外或灾害地区等环境中，车辆可通行区域变少。然而无人机拥有比无人车更广阔的视野，不受地面障碍物的影响，可以灵活地从空中俯视地面环境。本章介绍了应用无人机构建三维环境地图，为无人车导航提供先验信息。

三维重建涉及到的内容可以分为三部分：稀疏点云的获取，点云稠密化与点云的网格渲染。其中稀疏点云的获取使用的是Structure from Motion(SfM)，在技术层面上与当前热门的Visual-SLAM(VSLAM)比较类似，只是SfM侧重于三维重建，而SLAM侧重于实时定位，并需要预测下一时刻的位置。SfM可以分为视觉前端与优化后端，前端涉及到的是相机模型，坐标转换，特征提取与匹配以及多视图几何相关的内容，最后得到粗略的相机位姿与3D点坐标，后端涉及到的是优化前端得到的相机位姿与3D点坐标，去除错误的结果等。点云稠密使用的技术称为Multi-View Stereo(MVS)，MVS的方法很多，本文使用的是深度图融合的方法。网格渲染主要用的技术为三角剖分，网格渲染后的三维重建结果比稠密点云存储量小，更易于后续的仿真与处理。

3.2 相机模型

数码相机拍摄的过程中，实际上是一个光学成像的过程，这涉及到摄像机最基本的原件——透镜——的成像原理，如图 3.1 所示，这是最基本的透镜成像原理：Z 是物体距透镜光心的距离，简称物距；f 是焦距；b 是相距，即成像平面与透镜光心的距离。三者满足式 3.1

$$\frac{1}{f} = \frac{1}{Z} + \frac{1}{b} \quad (3.1)$$

在机器视觉中，利用摄像机可以将三维场景记录在二维图形上，不同的相机模型导致不同的成像效果，比如第二章中Ladybug使用的球面成像模型，使不同相机的拼

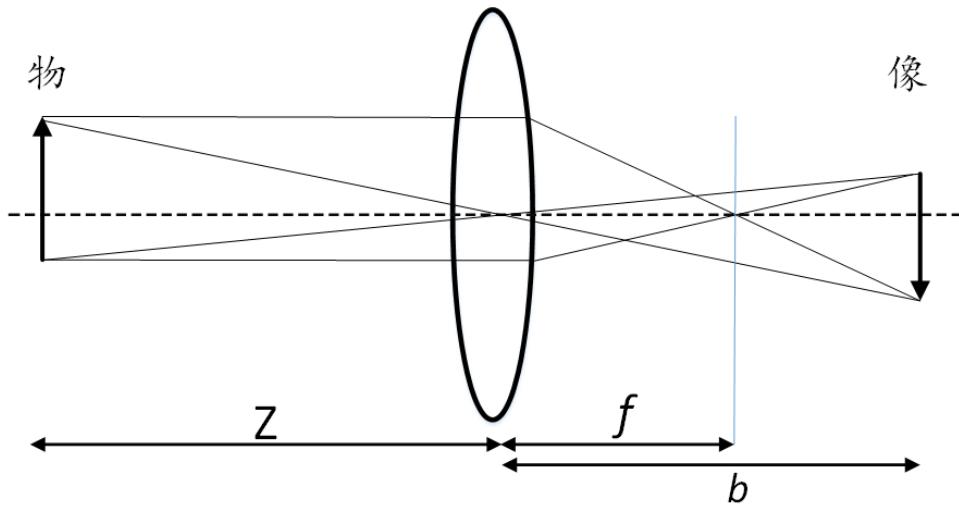


图 3.1 透镜成像原理

接简化为绕光心的旋转。但是常见的相机使用的模型还是小孔成像模型，如图 3.2 所示。数码相机的镜头相当于一个凸透镜，感光元件就处在这个凸透镜的焦点附近，将焦距近似为凸透镜中心到感光元件的距离时就成为小孔成像模型。这可以类比为艺术家画一幅画的过程，将眼睛放在图 3.2 光心 C 处，在物体与眼睛之间放置一个画布（图 3.2 中虚像位置），按照光线直线传播的原则，则物体反射的光线与眼睛的连线交画布一点，如此物体可以在画布上成像。只是相机的成像在光心后面，所以说存在一个虚拟成像平面。

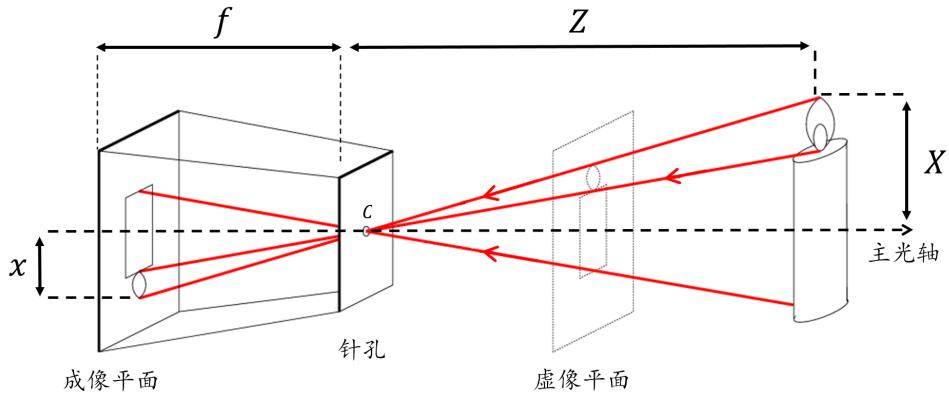


图 3.2 小孔成像模型

在三维空间中，物体反射的光线经过小孔形成倒立二维图像，根据其数学模型，可以得到

$$\frac{-x}{X} = \frac{f}{Z} \quad (3.2)$$

其中, f 是小孔到成像平面的距离, 即焦距; Z 是物体距光心的距离, 简称物距; x 是物体在成像平面的投影长度; X 是物体实际长度。为了简化以上模型, 用图 3.3 表示小孔成像的等价模型, 其中 C 是相机中心, $P(X, Y, Z)$ 为空间中的 3D 点, $p(x, y, 1)$ 表示 3D 点成像在感光元件上的点, 小孔成像近似后, 成像平面与相机中心的距离为焦距 f 。该图也表示了两个坐标系, 分别为图像坐标系和像素坐标系, 在下节中详细介绍。

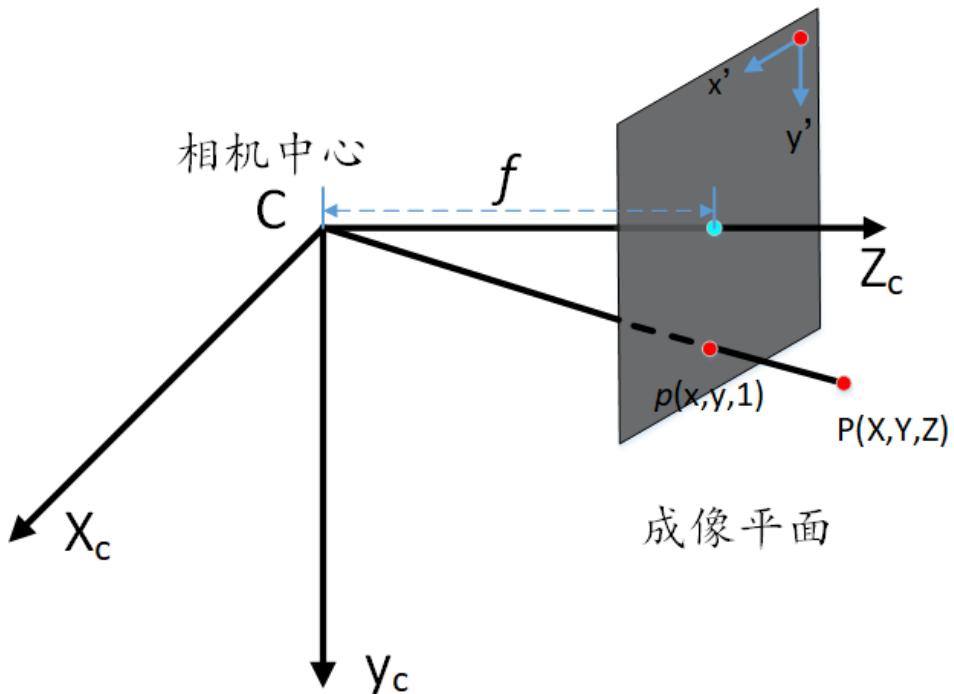


图 3.3 小孔成像等价模型

3.3 相关坐标系

从上一节知道, 仅仅相机内部就存在多个坐标系, 那么摄像机在空间中涉及到的坐标系转换问题将在这节详细探讨。相机成像的过程是一个 3D 物体显示在 2D 成像平面的过程, 在此涉及物体, 真实空间与相机三者的位置关。物体在真实空间的位置即是物体在世界坐标系中的位置坐标表达, 在图 3.3 中, (X, Y, Z) 为 3D 点 P 在相机坐标系 (X_c, y_c, Z_c) 下的坐标表示, 这里就存在两个三维坐标的转换, 即同一个点在世界坐标系与相机坐标系下表达方式的转换关系, 如式 3.3, 该变换可以看做三维坐标系的刚体变换。

$${}^cP = {}^wR_w^cP + {}^cT_w \quad (3.3)$$

其中 c 表示相机坐标系, w 表示世界坐标系, wP 表示世界坐标系下 P 点3D坐标, cP 是相机坐标系下 P 点3D坐标, cR_w 和 cT_w 分别表示由世界坐标系到相机坐标系的旋转矩阵和平移矩阵。

将世界坐标系下的3D点表示为齐次坐标 $(X, Y, Z, 1)$, 式3.3可以改写为:

$${}^cP = \begin{bmatrix} {}^cR_w & {}^cT_w \\ 0^{1 \times 3} & 1 \end{bmatrix}_{4 \times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.4)$$

其中 cR_w 是 3×3 矩阵, 三个列向量俩俩正交; cT_w 是 3×1 矩阵, 代表两个坐标系原点连线的向量, cP 坐标表示为 (x_c, y_c, z_c) 。

从图3.3中可以看到成像平面内的二维坐标系 x', y' , 该二维坐标系称为图像坐标系, 可以推导相机坐标系与图像坐标系(3D到2D)变换的公式表示:

$$x' = f \frac{x}{z} \quad (3.5)$$

$$y' = f \frac{y}{z} \quad (3.6)$$

改为矩阵表达形式为

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (3.7)$$

相机在生成图片的过程中, 由于成像平面由许多CCD感光元件组成, 所以成像的横纵坐标是不连续的, 引入像素的概念, 同时也存在从图像坐标系到像素坐标系(2D到2D)的转换关系:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{s_x} & s & p_x \\ 0 & \frac{1}{s_y} & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (3.8)$$

其中, dx 是u轴方向单像素的宽度, dy 是v轴方向单像素的宽度, 当前大部分ccd的像素宽度, $dx = dy$, 只有以前老式电视机使用的是矩形ccd, 导致 $dx \neq dy$ 。 p_x, p_y 是主点(图3.3)与图像左边界的距离与 p_y 是主点与图像上边界的距离。 s 是衡量主光轴与成像平面的倾斜程度, 如果主光轴垂直于成像平面, 则 $s = 0$ 。这些参数被称作相机内参, 一般可以通过查询相机手册得到包括焦距在内的相机内参。

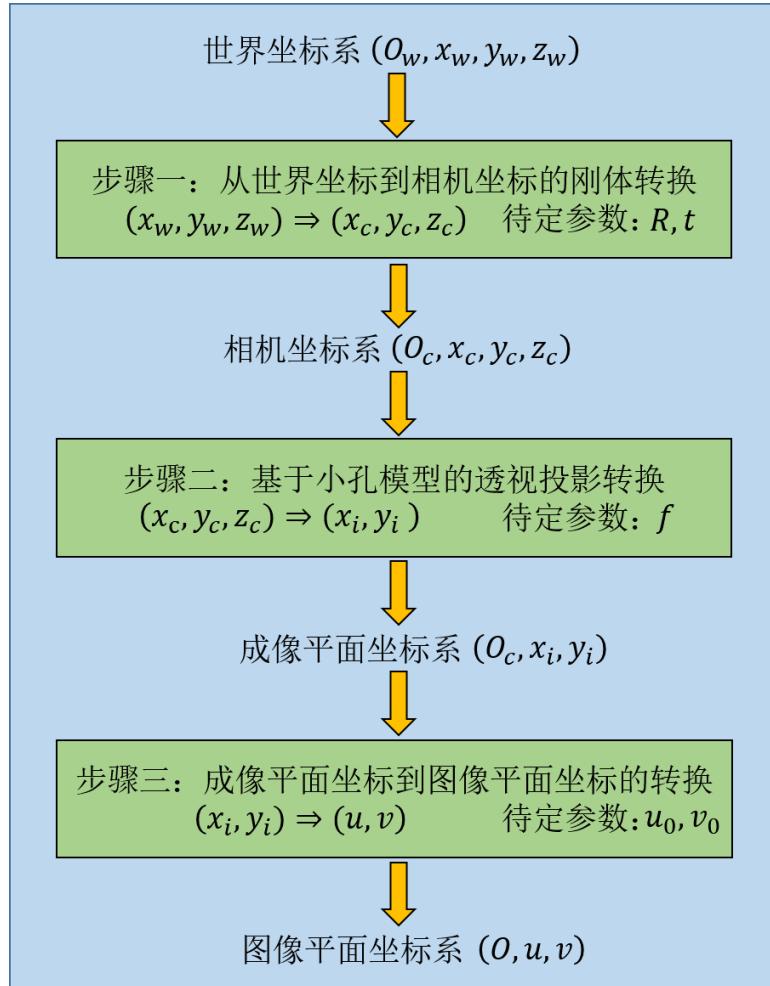


图3.4 坐标转换流程图

至此我们可以得到从世界坐标系到像素坐标系变化过程, 如图3.4和3.5。图3.4过程用公式表达为3.9

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{s_x} & s & p_x \\ 0 & \frac{1}{s_y} & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cR_w & {}^cT_w \\ 0^{1 \times 3} & 1 \end{bmatrix}_{4 \times 4} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.9)$$

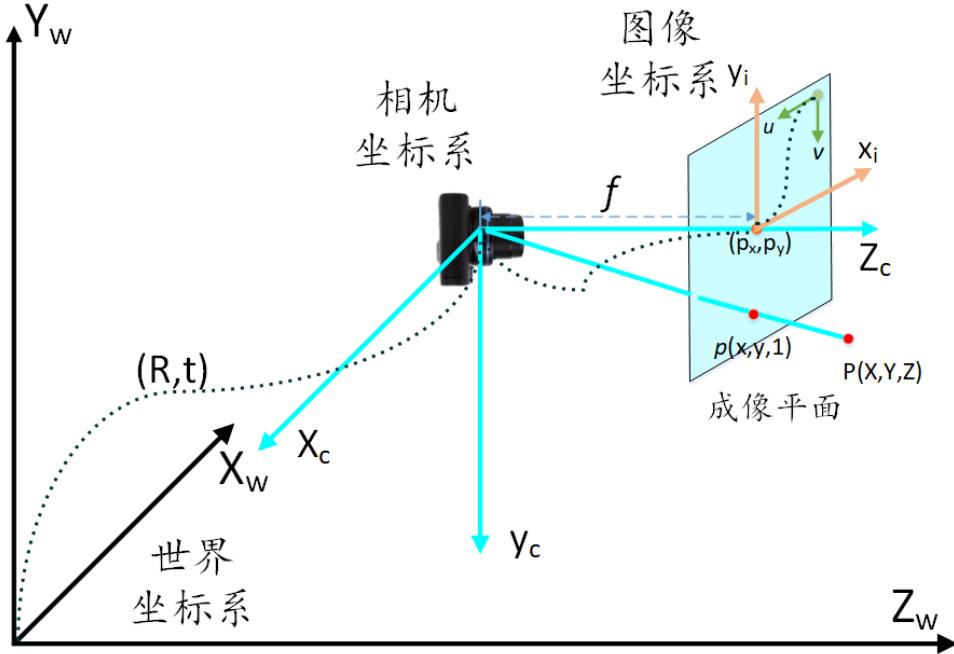


图 3.5 坐标转换关系图

至此我们知道了相机成像过程中某点如何在不同坐标系下转换，从而完成摄像过程。式 3.9 涉及到相机参数 f, p_x, p_y 以及空间变换相关的参数（外参 R, t ），实际上在相机成像过程中除了这些参数，我们还会遇到无法线性建模的参数：像 Gopro 等鱼眼镜头中，我们可以看到世界中的直线不再是直线，如图。。。这些弯曲的线有一些特别的特性：他们都是被关于中心扭曲的我们称这种畸变为径向畸变。这意味着像素点是沿着从中心放射的径向成比例扭曲的。为了去除相机的畸变，需要对畸变用多项式进行建模，如式 3.10。

$$u^{dist} = u(1 + k_1 r + k_2 r^2 + k_3 r^3 + \dots) \quad (3.10a)$$

$$v^{dist} = v(1 + k_1 r + k_2 r^2 + k_3 r^3 + \dots) \quad (3.10b)$$

$$\text{where } r^2 = u^2 + v^2 \quad (3.10c)$$

其中， k_1, k_2, k_3 是未知参数，需要标定才可以得到，而实际校正径向畸变时一般也只用到 2 个或 3 个参数，如图 3.6 所示，左图为存在径向畸变的图片，右图为经过校正后的图片。



图 3.6 径向畸变的校正

在相机校正的过程中，实际上还需要校正图像的切向失真，这是由于摄像机安装时成像平面与透镜没有平行导致的，如果说径向畸变是坐标点距离中心点的长度放生了变化，那么切向畸变就是坐标点与中心点的水平夹角发生了变化。对切向畸变的建模如式 3.11，一般使用 p_1, p_2 两个参数。

$$u^{dist} = u + 2p_1uv + p_2(r^2 + 2u^2) \quad (3.11a)$$

$$v^{dist} = v + 2p_2uv + p_1(r^2 + 2v^2) \quad (3.11b)$$

$$\text{where } r^2 = u^2 + v^2 \quad (3.11c)$$

综上所述，本节详细介绍了相机模型与相机相关坐标系转换关系，以及相机成像的校正等相关内容，这部分属于图像处理的基础，熟悉该部分内容将为下面几节多视图几何相关内容作铺垫。

3.4 特征点提取

SfM 首先需要找到图片集中俩俩图片间的几何关系，这样才能估算相机间的运动与成像的三维点坐标。为了找到图片之间的几何关系，现今有许多方法，例如光流法，特征点法等，光流法适用于匀速运送的视频流中，要求帧间运动比较小，从而找出帧间运动关系；特征点法指对图像特征的提取与存储，是图像中比较有代表性的特征。由于本文使用的数据集为无序的航拍图像，特征点法较为适宜使用。特征点提取更通俗的理解为将二维矩阵描述的图像降维为一维特征点描述的图像，突出图像的重点，去除冗余信息的过程。

最简单的图像特征为图像的角点，角点就是图像中线的交点，例如大厦最高处的顶点。两幅图像的角点不会随着图像移动而改变，但是角点会随着离相机的距离太近

而变得无法识别，所以许多研究者设计了精巧而互有优劣的特征检测与描述方法，比较著名的有SIFT(Scale Invariant Feature Transform)^[6]，SURF^[7]，ORG^[8]等。相对于简单的角点，这些特征检测具有以下优点^[9]：

- 可重复：相同的特征可以在不同的图片中找到
- 可区别：不同的特征有不同的表示
- 高效率：特征点数远远小于像素数

实际上，特征点可以分为局部特征点和全局特征点，像上面举例的三个特征点实际上是局部特征点，局部特征点顾名思义就是基于图像的突出区域明显的区分图像的特征，一般而言，局部特征需要具有旋转不变性，光线明暗的鲁棒性等。因此图像可以被提取到的一系列称为感兴趣区域的特征描述。相反全局特征是将图像表示为一个向量，向量的值为图像的各个方面，如颜色，纹理或形状。例如像区分图像，一些是海洋，一些是森林，一个全局颜色描述子可以很好的对其进行归类。在这种情况下，全局描述子涉及图像所有像素的特定属性，这个属性可以是颜色直方图，纹理，边缘等。如图 3.7 所示，使用哪种特征描述子取决于图像处理的情景。由于地理空间的航拍图像在颜色或者纹理等全局特征上，图片的相似度极高，不易辨识，难以找到图片之间的匹配关系，所以此文选用局部特征描述子：SIFT。

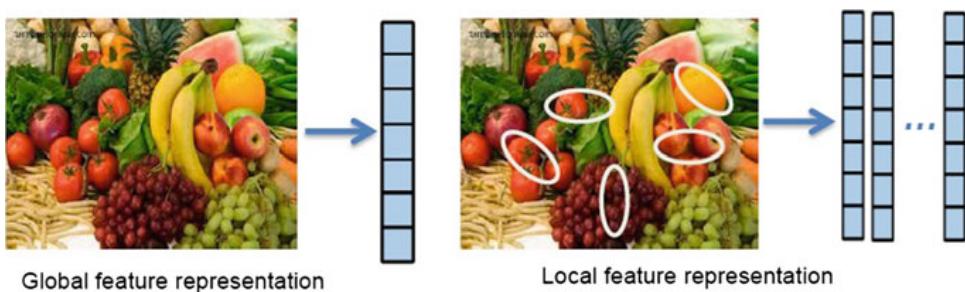


图 3.7 全局特征与局部特征描述

特征点提取包括特征点检测与特征点描述两部分，在此以SIFT特征点为例，简单介绍一下检测子与描述子的使用方法。检测子有很多，使用较多的有Harris Detector^[11]，FAST Detector^[10]。Harris检测子结合边缘与角点检测方法得到图像不同方向的自相关系数变化率，例如强度变化率等，该方法将局部特征描述为对称的自相关系数矩阵。Fast检测子考虑某像素周围一定大小的圆，院上像素强度与该像素比较，如果大于或小于某一阈值则定义该像素为角点，这两种检测子对尺度变化都不具有不

变性，不适合直接应用在拍摄高度不同的航拍图形的特征检测中。而SIFT作为一种具有尺度、放缩不变，对不同光线鲁棒的特定，很适合应用于环境复杂的航拍图形的特征特取与描述中。SIFT算法由四个步骤组成：尺度空间极值检测，兴趣点（关键点）定位，方向计算以及兴趣点描述。第一个阶段是使用Difference of Gaussian(DoG)确认潜在的兴趣点，这里使用DoG代替Laplacian of Gaussian (LoG)算子以提高计算速度。

Laplacian of Gaussian (LoG)是二阶导数的线性组合，也是检测斑点(blob)的检测子，给出一个图片 $I(x,y)$,尺度空间 $L(x,y,\sigma)$ 是由图片 $I(x,y)$ 与不同大小的高斯核 $G(x,y,\sigma)$ 卷积得到的，表示为

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (3.12)$$

其中，

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.13)$$

据此可计算Laplacian算子为

$$\nabla^2 L(x,y,\sigma) = L_{xx}(x,y,\sigma) + L_{yy}(x,y,\sigma) \quad (3.14)$$

这对于大小为 $\sqrt{2\sigma}$ 的斑点有最强的响应，然而该算子严重依赖图像中斑点大小与用来平滑图像的高斯核大小，即大小不同的高斯核函数被用来寻找对应大小的blob。为了在一幅图像中自动检测不同大小的斑点，一种多尺度的方法被提出来^[12]，其通过尺度不同的归一化的Laplacian算子对图片进行平滑从而在不同尺度空间找到大小不同的blob，尺度归一化的高斯核函数为式??。

$$\nabla^2_{norm} L(x,y,\sigma) = \sigma^2 L_{xx}(x,y,\sigma) + L_{yy}(x,y,\sigma) \quad (3.15)$$

大的 σ 对应图片的粗略特征，小 $sigma$ 对应图片的精细特征。同时LoG算子是对称的，所以它具有blob旋转对其没有影响，但是LoG算子耗费计算资源，为了加速计算并在尺度空间检测到稳定的兴趣点，Lowe^[6]提出了高斯差分尺度空间，简称DoG

scale-space。如式 3.16所示，利用不同尺度的高斯差分核函数与图像卷积生成。

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (3.16)$$

计算DoG的关键步骤是构建图像金字塔：对于某个图像，使用大小不同的高斯核与该图像卷积，可以得到模糊程度不同的但是图像长宽一致的图像，这些图像构成了一个八度(octave)，而后对这一系列图像进行降采样，降采样就是对octave中的图像隔行隔列采集像素，最后图像尺度(scale)变为原来大小的 $\frac{1}{4}$ ，构成下一个octave。这样就避免了对不同尺度空间下的图像进行卷积，减少了计算。为了寻找尺度空间的极值点，在DoG尺度空间中，每个采样点要和其周围所有相邻点（8邻域中，一个像素点周围共有26个像素点）比较，若该点是极大值或者极小值则定位为兴趣点。整个过程可以用图 3.8描述。

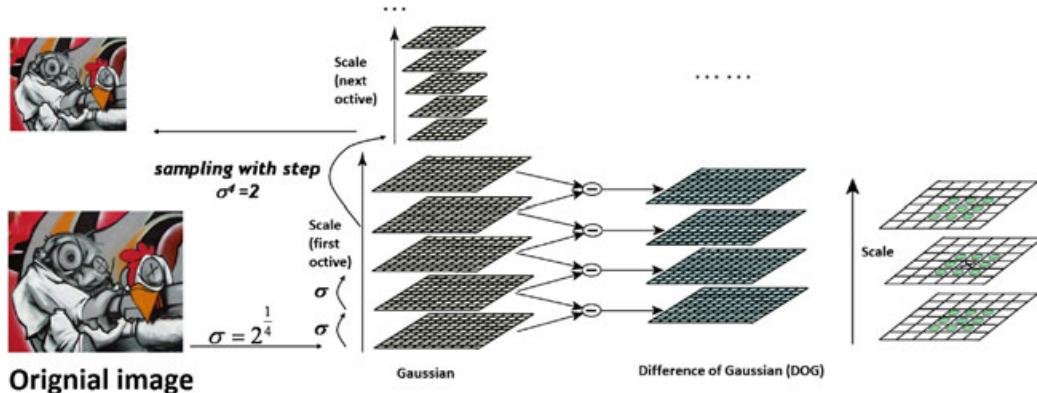


图 3.8 图像金字塔的构建

检测到特征点后，需要定量的表示各个特征点，以为特征点匹配做准备。Lowe在其论文中提出用一个128维度的向量表示每个特征点，具体做法为：如图 3.9，首先对每个关键点周围的 16×16 邻域内所有像素计算其梯度的幅值与方向，用直方图统计邻域像素的梯度方向，梯度直方图将 $0 \sim 360^\circ$ 分为8个区间，对得到的直方图进行高斯平滑，以减少突变带来的影响。选取直方图的峰值作为该关键点处邻域梯度的主要方向。所以在每个 4×4 的象限内，将每个像素的主方向加权到直方图的8个方向区间中的一个，计算一个梯度方向直方图，这样对于每个特征点可以形成 $4 \times 4 \times 8 = 128$ 维的描述子。

至此基本介绍了本文使用的SIFT特征点提取的方法，详细介绍了特征的提取与描述方法，以上是使用SIFT特征点检测多张图像的结果，其中特征点主方向以及特征点的强度分别表示在图中的。。。

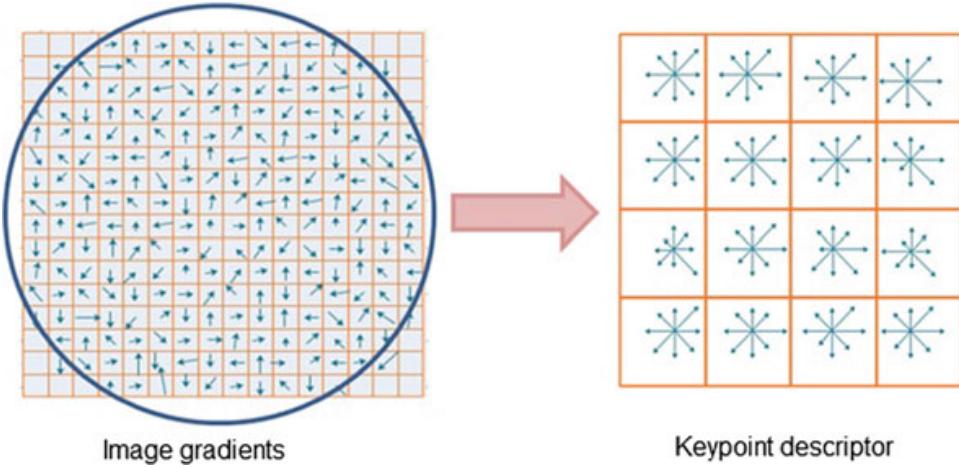


图 3.9 SIFT描述子的构建



图 3.10 SIFT特征点检测结果

3.5 特征点匹配

上面提到本文使用SIFT特征点作为关键点检测的方法，特征点提取并描述后，就应该找出各个图像中以特征点为依据的关系，即每张图像中的特征点如何与其他图像的特征点进行匹配，找到相近的特征点并去除错误的匹配的特征点。特征点匹配解决航拍图像之间的数据关联问题，将无序的图片数据集建立关联关系，为下文通过图像之间的关系，计算相机位姿与还原特征点的三维坐标提供基础。很好理解，一对匹配点就是实际空间的一个点，如果将所有图片对应同一个真实点的特征点连在一起，将使数据集中所有无序图片“有序化”。相同的，如果匹配点中有大量的错误匹配，那

对后续的位姿估计与还原三维坐标都产生不利影响，所以如果去除错误匹配更是至关重要的问题。下面就这两方面展开讨论。

假设在图片集 $I = \{I_i | i = 1 \dots N_I\}$ 中找到的局部特征点为 $F_i = \{(x_j, f_j) | j = 1 \dots N_{F_i}\}$ ，其中 x_j 表示特征点位置坐标， f_j 表示特征点描述子，如果是SIFT特征点，则 f_j 是128维向量。对于 F_i 与 F_j 的特征点匹配的方法，最简单的是使用暴力匹配，即每个 $(x_j, f_j) | j = 1 \dots N_{F_i}$ 与 $(x_j, f_j) | j = 1 \dots N_{F_j}$ 中特征点测量两个向量之间的距离，需要进行 $N_{F_i} \times N_{F_j}$ 次距离比较。对于像SIFT这样的浮点描述子，一般采用欧拉距离作为衡量依据，而对于像BRIEF（ORB特征使用的描述子）二进制描述子，则一般使用汉明距离作为度量依据。

但是当图片长宽较大，分辨率较高时，图片特征点会很多，这时采用暴力匹配会导致效率很低，一般实际使用中采用近似最近邻(Approximate Nearest Neighbor, ANN)匹配方法。事实上，无论SIFT特征匹配还是数据库检索本质上是相同的，都是使用距离函数在高维矢量空间中寻找相近对象的过程。常用的方法试K近邻查询，设置查询点与正整数K，从另一个数据集中根据距离公式找到距离最近的K个数据，如果K=1，改方法变为最近邻查询算法。K近邻查询算法是通过构建KD-Tree或者R-Tree等实现的。在此不做详细介绍。

在此简单介绍近似最近邻的快速库(Fast Library for Approximate Nearest Neighbors, FLANN)，FLANN具有一种内部机制，该机制可以根据数据本身选择合适的算法来处理数据集，基于FLANN的匹配非常准确、快速。FLANN在使用的时候需要配置两个参数：indexParams和searchParams。使用FLANN时，indexParams可以选择为LinearIndex、KTreeIndex、KMeansIndex、CompositeIndex和AutotuneIndex，其中KTreeIndex灵活且可被并行处理。对于searchParams，用来指定索引树被遍历的次数，即check值，该值越大则匹配的时间越长，当然也越准确。

当然即使用上面的方法，也无法保证所有的匹配都是正确的，Lowe^[13]在1999提出一种简单的方法可以剔除大部分的错误匹配。对于某个特征点向量，如果查询数据集中与其距离最近的特征点向量与次近的特征点向量的距离之比大于0.7，可以避免接近90%的错误匹配，但是正确的匹配也因此变少。使用FLANN并采用Lowe提出的方法可以去掉大部分的错误匹配，如图 ??所示。

以上提到的方法，匹配后仍然存在错误的匹配点，由于刚刚的匹配是单纯基于特征描述子的向量值，无法保证匹配的两个特征点对应相同的场景点。因此SfM使用投

影几何的知识估计两张图像的特征点变换。依赖图像对的空间配置，不同投影描述图片间不同的几何关系。例如，单应性变换（又称射影变换^[14]）描述相机拍摄的二维图片之间纯旋转和平移变换。对极几何通过基本矩阵（Essential matrix） E 和基础矩阵（Fundamental matrix） F 描述移动相机的关系，并可以通过三焦张量扩展到三视图的关系。无论哪种变换关系，有效的变换可以满足大部分匹配点的几何关系，那么这种变换或者匹配点被认为是有效的。本文在删除错误匹配时，使用的是单应性变换，一下简单介绍一下单应性变换的详细内容。

单应性又称射影变换、保线变换，与坐标系无关。映射 h 是射影变换的充要条件是：存在一个 3×3 非奇异矩阵 H ，使得任意一个矢量 X 表示的坐标点都满足 $h(x) = HX$ 。即

$$\begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.17)$$

矩阵 H 是一个齐次矩阵，在 H 的九个元素中有八个独立比率，即交比(cross ratio) $h_{11} : h_{12} : h_{13} : h_{21} : h_{22} : h_{23} : h_{31} : h_{32} : h_{33}$ 不会因为 H 因为乘以非零因子而变化，所以单应性变化有八个自由度，而每个二维特征点可提供2个方程，故需要解算 H 需要至少4对匹配点。已知4对匹配点，使用直接线性法解算 H ，示意图如图 3.11，已知两个视图四对匹配的特征点坐标 X_1, X_2 ，求 H 的过程。线性变换表示为 $X_2 = HX_1$ ，

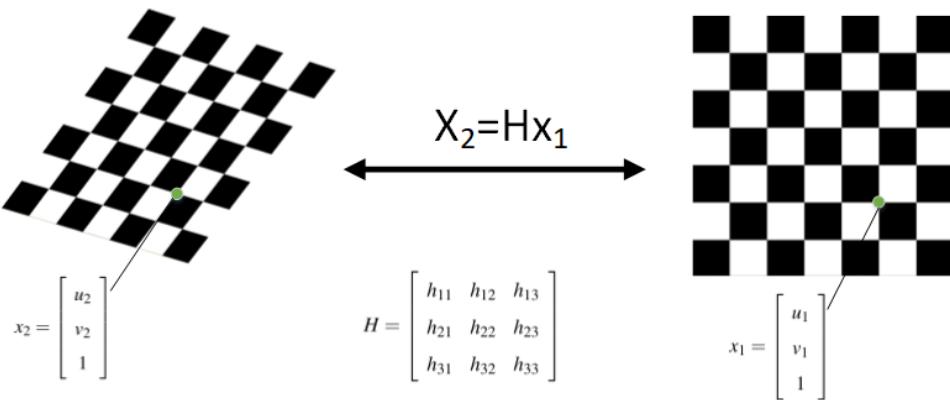


图 3.11 单应性变换

这是齐次矢量方程，三维矢量 X_2 和 HX_1 不相等，实际可以写为 $\lambda X_2 = HX_1$ ，线性变换可以变形为 $X_2 \times HX_1 = 0$ （矢量与自身外积为零），记为 $[X_2] \times HX_1 = 0$ ，设 h_i 为 H 的第*i*行，

$$\begin{aligned}
& \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \times \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} X_1 = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \times \begin{bmatrix} h_1 X_1 \\ h_2 X_1 \\ h_3 X_1 \end{bmatrix} = \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} \times \begin{bmatrix} X_1^T h_1^T \\ X_1^T h_2^T \\ X_1^T h_3^T \end{bmatrix} \\
& = \begin{bmatrix} 0 & -1 & v_2 \\ 1 & 0 & -u_2 \\ -v_2 & u_2 & 0 \end{bmatrix} \begin{bmatrix} X_1^T & 0_{1 \times 3} & 0_{1 \times 3} \\ 0_{1 \times 3} & X_1^T & 0_{1 \times 3} \\ 0_{1 \times 3} & 0_{1 \times 3} & X_1^T \end{bmatrix} \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} \tag{3.18}
\end{aligned}$$

最后一步是由线性代数的知识得到，即对3维矢量 $x = (x_1, x_2, x_3)$ 的 3×3 的反对称矩阵为式 3.20

$$[a]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{3.19}$$

合并 3.18 最后一步的前两项得到下式为

$$= \begin{bmatrix} 0_{1 \times 3} & -X_1^T & v_2 X_1^T \\ X_1^T & 0_{1 \times 3} & -u_2 X_1^T \\ -v_2 X_1^T & u_2 X_1^T & 0_{1 \times 3} \end{bmatrix} \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} = 0 \tag{3.20}$$

上式简写为 $Ab = 0$ ，其中 A 为 3×9 矩阵， $\text{rank}(A) = 2$ ，这是因为每队匹配点只提供两个量： u 和 v ，也可以从 A 矩阵的值看出，将 A 的第一行乘以 u_2 与第二行乘以 v_2 相加得到第三行。 b 为 9×1 矩阵，由于只需要维持 b 各项的交比不变即可，所以有8个自由度，故需要4对匹配点即可在这个线性系统上获得足够的条件计算得到单应性矩阵 H 的各项。由以上分析我们可以知道当给定的匹配点为4个时，并将 $\|H\| = 1$ ，方程有精确解。但是大部分的匹配问题匹配点的数目远不止4对，如果多于4对，那么 $Ab = 0$ 是超定的，如果所有的匹配点的位置是精确的，那么 $Ab = 0$ 的解仍然是精确的，但是通常情况匹配点中有噪声，存在错误匹配等问题，所以这时候方程存在最小二乘解（超定解）。除了零解可以忽略外，方程解可以使用SVD找到其近似解。

SVD又称奇异值分解，在此不详细介绍，SVD可以将一个矩阵，无论是否满秩，均可以分解为 $A = UDV^T$ 形式，其中 $UU^T = I$ 和 $VV^T = I$ ，即 U 和 V 是正交矩阵， D 是对

角阵，对角元素非负。实际上SVD可以看做特征值分解的推广，SVD在主成分分析，求解方程的最小二乘解中广泛应用。回到求解单应性变换的问题来，根据SVD求解最小二乘解的原理，可以得到：最小二乘解是 $A^T A$ 的最小特征值的特征向量。具体讲，假设有n个匹配点，则A为 $2n \times 9$ 的矩阵，且 $A = UDV^T$ ，且D对角阵元素按降序排列，那么b是V的最后一列。这是由于 $Ab = 0$ 的解，即A的零空间，又由于 $\text{rank}(A) = n - 1$ ，所以A的零空间只有一个基向量，即SVD中最小特征值对应的特征向量V的最后一列。

奇异值分解是广泛用于图像处理中的一种算法，不仅在求单应性矩阵时使用，下文亦有涉及。但是这里要介绍一种本文使用到的另一种方法——RANSAC(Random Sample Consensus)，该方法在图像领域中也广泛应用，其根本原理是使用统计原理，可以通过多次迭代估计含有噪声的数据模型的参数。RANSAC中文译为随机抽样一致性，其一般过程为从一组含有外点（噪声）的数据中随机选取数个数据，估计模型参数，然后将估计得到的模型应用于剩余的其他数据，从而得到该模型的误差，重复以上过程，直到误差达到某设定阈值或迭代次数达到设定最大次数，停止迭代，选取最优的一组模型作为最终结果。根据最终结果与事先设定的阈值可以过滤一部分外点并得到较为精确的模型。由以上内容知，估计单应性变换需要4对匹配点即可计算一个单应性变换，所以RANSAC每次选取4对匹配点迭代计算得到单应性矩阵，而后将单应性矩阵应用于剩余的匹配点，计算误差，过滤外点。如图 3.12所示，其中a,b是仅根据SIFT描述子通过最近邻匹配过滤的结果。c,d是使用RANSAC方法通过单应性变换过滤的结果，从图中可以直观看出使用RANSAC可以过滤掉一部分错误的匹配，同样从表 3.1中也可以看出，其中对角数字表示图 3.12中三张图片提取到的SIFT特征点数量，其他表格数目，例如55/32表示使用RANSAC方法前得到的匹配点数目，以及使用RANSAC后过滤剩下的匹配点。

表 3.1 匹配点过滤

图片名		400	415	424
图片名	特征点			
400		861	55/32	75/46
415			370	19/10
424				599

经过各种方法，得到的图片匹配点基本上是正确的，基于此我们可以对数据集中无序图片构建它们的关联结构，对于多张图片对应的同一个匹配点，设置统一标



图 3.12 RANSAC筛选的单应性变换结果

号，称为track。例如图片一中第125个特征点，图片二中第259个特征点，图片三中第6个特征点...匹配，实际上它们对应实际地理空间的同一个点，所以可以使用同一个track序号标记这些匹配点，这样做使整个数据集通过特征点匹配构成关联关系，为下面的多视图重建打下基础。

3.6 多视图重建

本节内容主要介绍如何通过多视图几何的只是，恢复上节匹配的特征点的三维空间结构，以及摄像机位姿等相关信息，本节内容采用增量式SfM的方法，首次从track点最多的两张图片入手，恢复两视图匹配特征点的三维空间位置与两视图对应的摄像机的相关关系，而后加入第三张图片，计算第三张图片与前两张图片得到的3D点的关系，使用三角测量方法还原第三张图片的特征点3D位置坐标，如此循环。其中每加入一张图片使用光束平差法（Bundle Adjustment，简称BA）优化3D点与相机位姿。最终得到较为理想的点云信息。

3.6.1 两视图重建

在式 3.9中， z_c 为每个像素的深度，在单目成像时， z_c 是未知的。所以当知道某点的世界坐标系时，可以很容易得到该点的像素坐标，但是反之无法已知像素坐标得到世界坐标。所以从单张图像中无法还原场景的三维结构，所以下文介绍从两张存在特征点匹配的图片中恢复场景的三维结构。两视图几何最核心的原理是对极约束，在阐述堆积约束相关知识之前，我们先复习一下3.3节中关于摄像机成像过程中涉及到的相关坐标系。如果三维空间的一个点 X ，其在相机所成的像坐标

为 x , 则有 $\lambda x = PX = k[R \ t]X$ 。其中 λ 表示每个像素的深度; $[R \ t]$, 表示摄像机坐标系与世界坐标系刚体变换。了解这些后下面将详细阐述对极约束的相关知识。如图 3.13 所示, 其中 C_1, C_2 表示两个摄像头同时看到三维世界中同一点, 该点在 C_1 为原点的坐标系的三维坐标为 X_1 , 在 C_2 为原点的坐标系的三维坐标为 X_2 。在该对极几何中, 世界坐标原点选为 C_1 , 坐标轴与该相机坐标系相同, 所以对于 C_1 表示的相机成像矩阵 $P = k[I \ 0]$, 而对于 C_2 表示的相机成像矩阵 $P = k[R \ t]$, 所以 $X_2 = RX_1 + t$, 表示 C_1 为原点的坐标系的点经过 R, t 可以转换到 C_2 为原点的坐标系表示的过程。在 C_2 为原点的坐标系中, 我们可以得到如图 3.13 中红色三角形三边对应的矢量为 $t, X_2, X_2 - t$, 其中 $X_2 - t = RX_1$, 由于三边共面有向量的混合积如式 3.21 所示, 当然式中 $[t] \times X_2 = t \times X_2$ 表示垂直于图 3.13 红色平面的向量, 由绿色箭头显示。最后我们可以得到简洁的形式 $0 = -X_2^T EX_1$, 其中 $E = [t] \times R$, E 被Longuet Higgins称为本质矩阵, 他首先发现的这一关系。本质矩阵是 3×3 的矩阵, 它将旋转与平移的复杂关系转换为简单的形式, 下文将介绍如何计算 E 与如何从 E 中恢复旋转、平移信息。

这里还有几个概念需要介绍一下, 所有由3D点, 相机中心 C_1, C_2 组成的平面交成像平面于两条直线, 如图 3.13 中蓝色直线所示, C_1 成像平面中一点 λX_1 对应的极线为红色平面与 C_2 成像平面的交线, 即蓝色线 Fx_1 。如果改变3D点的位置形成另外一个红色平面, 其与成像平面的交线为另外一条极线, 所有形成的极线交于一点, 该点称为极点, 如图中 e_1, e_2 , 所示。从极线约束的关系看 $0 = -X_2^T EX_1$, 结合齐次坐标的性质点 x 在直线上的充要条件是 $x^T L = 0$, 所以 EX_1 是过 X_2 的直线, 也就是与 C_2 成像平面的极线平行的直线, 由于所有的极线 Ex_1 一直通过同一个点(极点 e_2), 所以 $e_2^T E = 0, Ee_1 = 0$ 。极点与极线的关系在图 3.14 中展示出来, 图a中粉色点表示选取的两个特征点, 而图b中对应的图a两点的两条极线标注为红色, 这两条线的交点为极点。

$$0 = (X_2 - t)^T \cdot [t] \times X_2 = (RX_1)^T \cdot [t] \times X_2 = X_1^T R^T [t] \times X_2 = -X_2^T [t] \times RX_1 = -X_2^T EX_1 \quad (3.21)$$

$$0 = -X_2^T EX_1$$

已知 $KX_1 = x_1$, 其中 K 为内参矩阵, X_1 表示 C_1 为原点的坐标系中三维点坐标, x_1 表示 C_1 的像素坐标系中二维点坐标。这里和上文相同都是其次坐标, 所以是在齐次意义上相等。带入 $-X_2^T EX_1 = 0$, 中有式 3.22 所示, 其中 F 称为基本矩阵, $rank(F) = 2$,

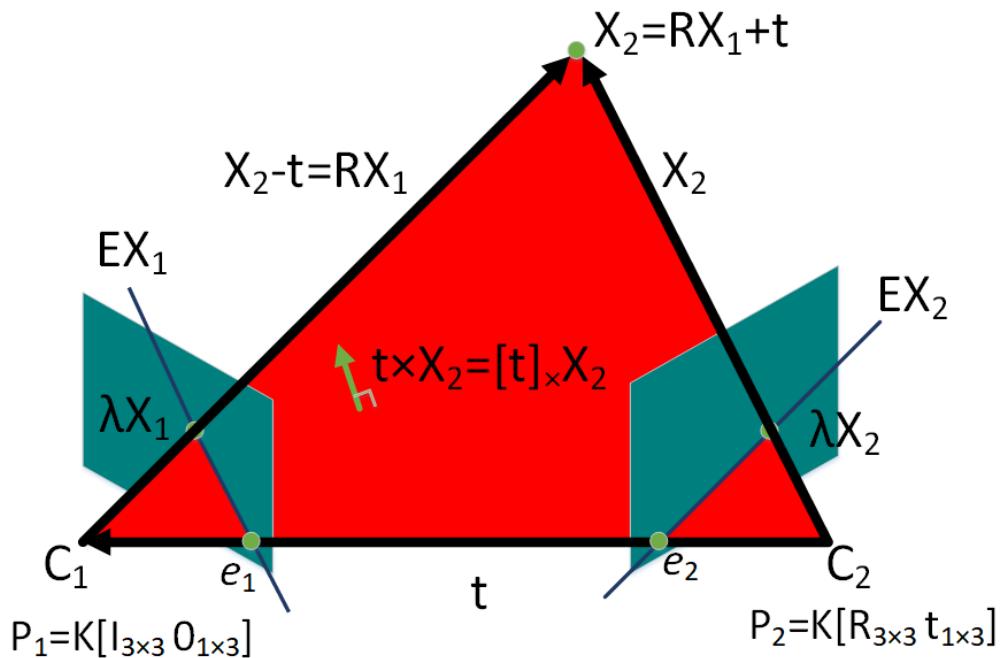


图 3.13 对极几何约束



(a) a



(b) b

图 3.14 极点与极线的关系

F 的自由度为8。根据式 3.22，一对匹配点可以得到一个方程，所以至少需要8对匹配点才能计算得到矩阵 F 。如式 3.24所示，可以简写为 $AX = 0$ ，与上文计算单应性矩阵的方法相同，由于匹配点远不止8对，所以此时方程存在最小二乘解，使用SVD对左边矩阵分解得 UDV^T ，则基本矩阵即为 V 的最后一列重新排列成 3×3 的形式，当然最后还需要保证 $\text{rank}(F) = 2$ ，这可以通过将 F 进行SVD分解并使最小奇异值为零得到秩为2的 F 。至此我们可以根据至少8对匹配点解算出基本矩阵 F ，该方法称为八点法，在两视图重建中被广泛使用。另外一种方法是使用RANSAC代替SVD分解，避免噪声带来的影响。

$$\begin{aligned} X_2^T E X_1 &= 0 \\ x_2^T k^{-T} E k^{-1} x)) 1 &= 0 \\ x_2^T F x_1 &= 0 \end{aligned} \quad (3.22)$$

基于式 3.22可以得到下式

$$\begin{bmatrix} u_i^1 & v_i^1 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} u_i^1 \\ v_i^1 \\ 1 \end{bmatrix} = 0 \quad (3.23)$$

$$\begin{bmatrix} u_1^1 u_1^2 & u_1^1 v_1^2 & u_1^1 & v_1^1 u_1^2 & v_1^1 & u_1^2 & v_1^2 & 1 \\ u_2^1 u_2^2 & u_2^1 v_2^2 & u_2^1 & v_2^1 u_2^2 & v_2^1 & u_2^2 & v_2^2 & 1 \\ \vdots & \vdots \\ u_8^1 u_8^2 & u_8^1 v_8^2 & u_8^1 & v_8^1 u_8^2 & v_8^1 & u_8^2 & v_8^2 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \quad (3.24)$$

在得到基本矩阵 F 后，根据式 3.22以及已知相机内参矩阵 K 的情况下，可以得到本质矩阵 E ，从本质矩阵可以恢复相机之间的旋转平移关系。

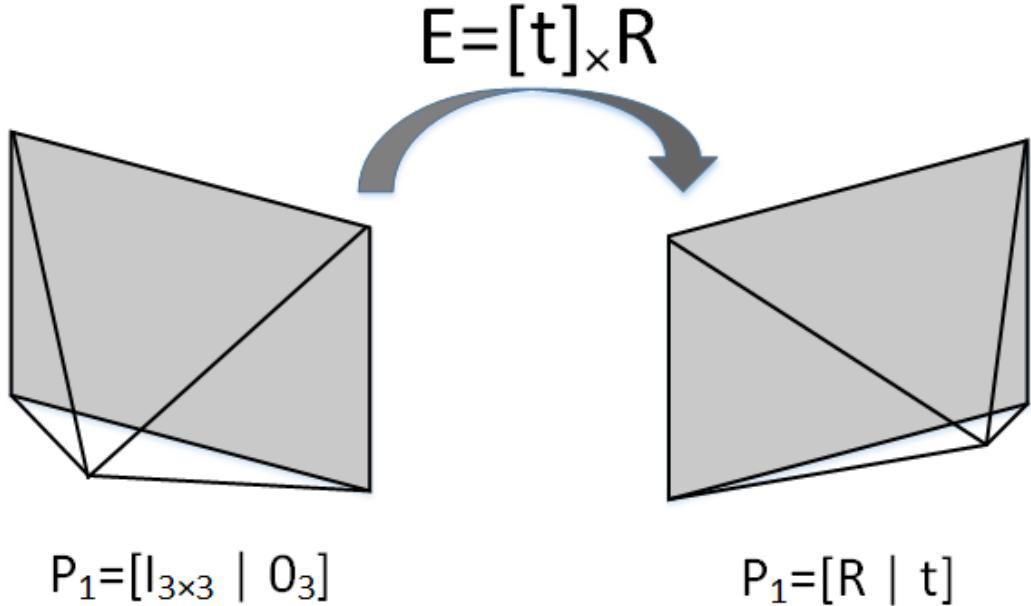


图 3.15 本质矩阵恢复旋转平移关系

由于 $P_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = [R|t] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = t$, 所以 $t^T E = 0$, 由SVD知 t 是 E 的左零空间, 也是第二视图极点 $e_2^T E = 0$, 所以 $t = U[:, -1]$, 即 E svd 分解后 U 的最后一列, 即 $t = -u_3, u_3$

$$E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T = [t]_x | R = U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} U^T | U Y V^T \quad (3.25)$$

where $U = \begin{bmatrix} u_1 & u_2 & t \end{bmatrix}$

据此我们可以得到

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} Y \quad (3.26)$$

所以可以得到矩阵 Y 的值， $Y/Y^T = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ 所以 R 有两种可能值，这是因为 $R = UYV^T$ 。

综上通过 E 恢复旋转矩阵 R 和平移矩阵 t 有四种可能的解，图 3.16 形象地显示了分解本质矩阵得到四个解的过程。保持成像平面的点（红点）不变的情况下，可以画出四种可能的情况，但是只有(a)是正确的，因为其有正深度，所以恢复旋转与平移后需要将3D点带入图中四个解中，检测该点在两个相机中的深度，即可确定 R 和 t 。实际上利用 E 的内在性质，其只有五个自由度：旋转(3)，平移(1)，相机中心(1)。但是使用五点法形式复杂，所以本文还是采用上文介绍的八点法计算旋转与平移关系。

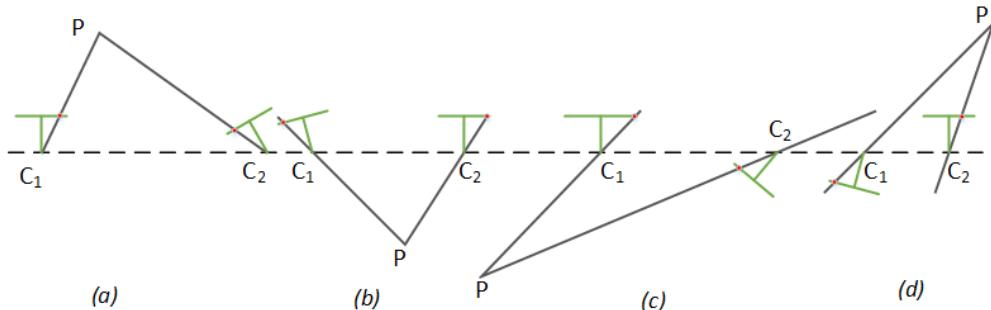


图 3.16 分解本质矩阵得到的四个解

3.6.2 三角定位法

在得到相机的位姿后，需要恢复匹配点的三维坐标信息，该方法称为三角测量 (triangulation)，三角测量时指不同视角观察同一个点的夹角，确定该点的位置。三角测量最早由高斯提出并应用于天文学中。在SfM中，我们使用三角测量估计匹配点的像素深度。

小孔相机成像模型简化为 $\lambda \begin{bmatrix} x_1 \\ 1 \end{bmatrix} = P_1 \begin{bmatrix} X_1 \\ 1 \end{bmatrix}$ ，即 $\begin{bmatrix} x_1 \\ 1 \end{bmatrix} \times P_1 \begin{bmatrix} X_1 \\ 1 \end{bmatrix} = 0$ ，由于一个视图可以提供两个方程，所以只需要两个视图即可计算得到3D点坐标，但是由于噪声的影响，空间中的两视图射线不一定交于一点，所以使用两视图恢复出来的三维点坐标存在不准确的问题，所以一般情况下使用多视图进行三角测量恢复多视图匹配的3D点坐标，如图 3.17 所示，多个视图匹配的同一个3D点在不同视图成像平面的投影为 x_1, x_2, \dots, x_n ，这些投影点具有相同的track序号（第三章第二节末提到的定义），

当从多视图中恢复3D位置坐标，该问题又归为对式 3.27 进行奇异值分解求最小二乘解的问题。由于 $\text{rank}([\])_{3n \times 4} = 2$ 所以对 $[]$ 进行SVD分解， V 的最后一列就是 $[]$ 的右零空间，也就是3D点的位置坐标。

三角测量的多视图必须存在平移关系，否则单纯的旋转无法使用三角测量，因为此时对极约束永远满足。在平移存在的情况下，三角测量存在不确定性，当平移很小时，射线夹角很小，计算得到的像素深度不确定性很大；但是平移太大后，图像变换会较大，导致匹配失效。所以三角测量时，需要考虑多视图夹角，也就是多视图平移大小的影响。

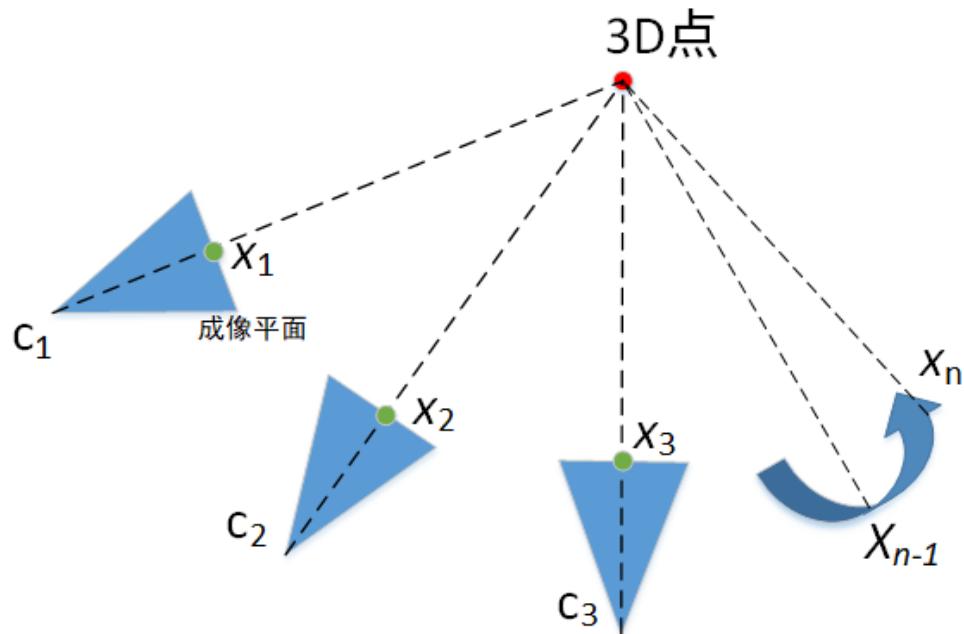


图 3.17 三角测量示意图

$$\left[\begin{array}{c|cc} \begin{bmatrix} x_1 \\ 1 \\ x_2 \\ 1 \\ \vdots \end{bmatrix} & P_1 \\ \hline & \times \\ \begin{bmatrix} x_3 \\ 1 \\ \vdots \end{bmatrix} & P_2 \\ & \times \\ & \vdots \end{array} \right]_{3n \times 4} = \begin{bmatrix} X \\ 1 \end{bmatrix}_{4 \times 1} = 0 \quad (3.27)$$

3.6.3 2D-3D位姿求解

经过以上的计算，我们可以得到初始的两视图匹配点的3D位置坐标，也就是我

们通过两视图匹配关系得到场景的特征点3D位置坐标，或者称为稀疏点云，以及两视图的空间几何关系，至此我们加入第三张匹配的图片，这时就存在如何计算第三张图片对应的相机与刚刚三角测量得到的点云的空间位置关系，完成这一步才能继续增加其他图片，完成对数据集的增量式重建过程。

PnP(Perspective-n-Point)是求解3D到2D点对运动的方法。新图片能被通过解决PnP问题与当前的点云模型（两视图或者多视图得到的）配准。PnP问题被用来估计相机位姿，包括未矫正相机的内参矩阵。我们知道对极几何估计2D-2D位置关系，使用八点法，但也存在纯旋转等问题。然而，如果两张图像其中一张特征点的3D位置已知，那么最小只需要三对匹配点就可以估计相机运动。特征点的3D位置可以有上文的三角测量或者深度传感器确定。3D-2D方法不需要使用对极约束，其有多种解法，直接线性方法，RANSAC或者非线性优化构建最小二乘问题并接待求解的Bundle Adjustment。

$$\begin{aligned}
 & \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} X = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times \begin{bmatrix} P_1 X \\ P_2 X \\ P_3 X \end{bmatrix} \\
 &= \begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} X^T & 0_{1 \times 4} & 0_{1 \times 4} \\ 0_{1 \times 4} & X^T & 0_{1 \times 4} \\ 0_{1 \times 4} & 0_{1 \times 4} & X^T \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} \\
 &= \begin{bmatrix} 0_{1 \times 4} & -X^T & vX^T \\ X^T & 0_{1 \times 4} & -uX^T \\ -vX^T & uX^T & 0_{1 \times 4} \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} = 0
 \end{aligned} \tag{3.28}$$

Which is $A_1 b = 0$

下面仅介绍使用直接线性方法结合SVD分解，其他的方法与此原理相似不一一展开，首先回忆最经典的小孔成像模型 $\lambda \begin{bmatrix} x_1 \\ 1 \end{bmatrix} = P_1 \begin{bmatrix} X_1 \\ 1 \end{bmatrix}$ ，即 $\begin{bmatrix} x_1 \\ 1 \end{bmatrix} \times P_1 \begin{bmatrix} X_1 \\ 1 \end{bmatrix} = 0$ ，其中 P 为未知量，2D坐标 x 与3D坐标 X 都是已知的，参照求单应性矩阵的方法，我们可以把推导写成如式 3.28 所示，其中 $\text{rank}(A) = 2$ ，这很容易理解，因为每对匹配点仅能提供两个方程， A 为 3×12 矩阵， b 为 12×1 的矩阵，所以需要至少 6 对匹配点才能计

算得到位置关系矩阵 b 。此时等式简写为 ??的组合矩阵，其中每对3D-2D匹配点可以写成类似 A_1 的形式。此时可以对式 ??中最左边的组合矩阵运用SVD求解。对于实际中匹配点大于6对时，SVD求得的是最小二乘解。

$$\begin{bmatrix} A_1 \\ A_2 \\ \dots \\ A_6 \end{bmatrix} b = 0 \quad (3.29)$$

通过以上方法，我们可以得到矩阵 P ，由于 $P = K[R|t]$ ，所以旋转矩阵 $R = K^{-1}P_{[1:3]}$ ， $P_{[1:3]}$ 表示 P 的前三列，由于旋转阵 R 是正交矩阵，为了保证这一性质，所以旋转阵取式 3.30 中 R_+

$$R_+ = UV^T \quad (3.30)$$

where $UDV^T = R$

那么该如何恢复平移阵 t 呢？如式 3.31 所示，最后 $P = K[R_+t]$

$$t = K^{-1}P_4/\sigma_1 \quad (3.31)$$

where $D = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$

and $\sigma_1 > \sigma_2 > \sigma_3$

3.7 三维重建中的优化

上面介绍了如何使用多视图几何的知识计算相机位姿与特征点的3D坐标，一般是先计算相机位姿，后计算特征点3D坐标，而本章要介绍的非线性优化的问题，将相机位姿与特征点3D位置放在一起优化，实际上SfM中structure就是指特征点3D坐标，而motion指相机位姿，但是以上内容将图像配准（image registration）和三角测量作为独立的步骤，实际上它们是紧密关联的：相机姿态的不确定将导致三角测量的不确定性，反之亦然。没有优化操作，SfM将很快离散到难以恢复的状态。本文一下介绍同时优化相机位姿与3D点坐标的常用方法，称为光束平差法(Bundle Adjustment, BA)，主要手段是最小化重投影误差。

3.7.1 光束平差法

BA是对相机位姿矩阵 P_c 与3D点 X_k 的非线性优化方法，其最小化重投影误差如式3.32，其中函数 $f(x)$ 表示3D点的重投影误差，函数 π 是3D点在成像平面的投影的像坐标， ρ_j 是损失函数，用来降低外点带来的影响。

$$E = \arg \min_x \sum_{j=1}^k \|f_j(x) - x_j\| = \sum_j \rho_j(\|\pi(P_c, X_k) - x_j\|_2^2) \quad (3.32)$$

为解决该问题，使用最多的方法是梯度下降法，该算法是解决非线性最小二乘问题的通用方法，而BA实际上就是非线性最小二乘问题。实际操作中，我们一般使用Levenberg-Marquardt^[15]，简称LM，LM将非线性问题转变为一系列正则线性问题。设 $J(x)$ 是 $f(x)$ 的雅各比矩阵，最小化重投影误差 E 如式3.33

$$\min_x \|f(x) - b\|^2 = \min_x (f(x) - b)^T (f(x) - b) = \min_x f(x)^T f(x) - 2b^T f(x) \quad (3.33)$$

对式3.33求导数有

$$\frac{\partial E}{\partial x}|_{x^*} = 2 \frac{\partial f(x)}{\partial x}^T f(x) - 2 \frac{\partial f(x)}{\partial x} b = 0 \quad (3.34)$$

其中

$$J = \frac{\partial f(x)}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \quad (3.35)$$

雅各比矩阵的 m 一般不等于 n ， m 表示组成 $f(x)$ 的等式个数， n 表示与 $f(x)$ 相关的变量的个数，下文会详细解释在光束平差法中各个变量代表的含义。雅各比矩阵反映存在 m 个约束条件的误差函数的变化率。对式中的 $f(x)$ 泰勒展开如式3.36，最终结果可简写为 $H\Delta x = -J^T \nabla$ ，其中 H 称为Hessian矩阵。

$$\begin{aligned} \frac{\partial E}{\partial x}|_{x^*} &= 2 \frac{\partial f(x)}{\partial x}^T (f(x) + \frac{\partial f(x)}{\partial x} \Delta x) - 2 \frac{\partial f(x)}{\partial x} b = 0 \\ \frac{\partial f(x)}{\partial x}^T \frac{\partial f(x)}{\partial x} \Delta x &= \frac{\partial f(x)}{\partial x}^T (b - f(x)) \\ \Delta x &= (J^T J)^{-1} J^T (b - f(x)) \end{aligned} \quad (3.36)$$

式 3.36 的推导结果称为高斯牛顿法，LM 是基于高斯牛顿法，只是在迭代步长时不使用 $\Delta x = (J^T J)^{-1} J^T (b - f(x))$ ，而改为 $\Delta x = (J^T J + \lambda D(x)^T D(x))^{-1} J^T (b - f(x))$ ，其中 $D(x)$ 是非负对角矩阵，是 $J^T J$ 的对角元素的平方根，也可以简写为 $H_\lambda \Delta x = -J^T \nabla$ ，式中 H_λ 被称为增广 Hessian 矩阵。相较高斯牛顿法，LM 的优点是可以动态调节 Δx ，当下降太快时，使用较小的 λ ，反之亦然。经过上面的计算我们知道实际上光束平差法是一种非线性最小二乘法，关键是如何计算 $f(x)$ 的雅各比矩阵 J ，即 $f(x)$ 的梯度，并最终计算步长 Δx ，直至误差 E 收敛至最小。下面介绍一下，BA 中针对多个相机多个 3D 点使用的 LM 算法计算雅各比矩阵的形式。最小化误差函数存在 $6(F - 1)$ 个运动约束和 $3N - 1$ 个结构约束， F 指相机个数， N 指 3D 点的个数，这可以理解为第一个相机为基准，所以不计入考虑，而每个相机有 6 个约束条件，即旋转 (3)、相机中心 (3)，每个 3D 点都是 3 个约束条件，但是缺失一个尺度，所以减去 1。Hessian 矩阵 $J^T J$ 的维数为 $(6F + 3N - 7) \times (6F + 3N - 7)$ 。既然如此，由于 3D 点与相机非常多，所以 BA 得到的雅各比矩阵与 Hessian 矩阵将非常大。如图 3.19 所示，两视图、同一个 3D 点对应的矩阵为式 3.18 中的 J

$$J_{\text{part}} = \begin{bmatrix} \frac{\partial f(R(q), C, X)}{\partial R} & \frac{\partial R}{\partial q} & \frac{\partial f(R(q), C, X)}{\partial C} & \frac{\partial f(R(q), C, X)}{\partial X} \end{bmatrix}$$

$$J = \begin{bmatrix} \text{Left View} & 0_{2 \times 7} & 3D \text{ Point} & 0_{2 \times 3} \\ 0_{2 \times 7} & \text{Right View} & 3D \text{ Point} & 0_{2 \times 3} \\ \text{Left View} & 0_{2 \times 7} & 0_{2 \times 3} & 3D \text{ Point} \\ 0_{2 \times 7} & \text{Right View} & 0_{2 \times 3} & 3D \text{ Point} \end{bmatrix}$$

图 3.18 雅各比矩阵

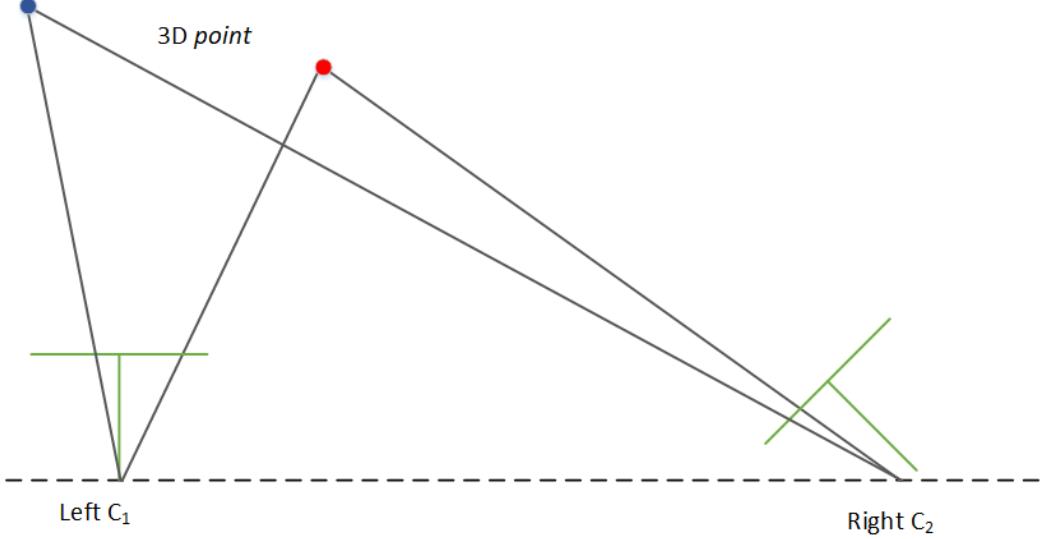


图 3.19 雅各比矩阵示意图

$$\begin{bmatrix} U_\lambda & W \\ W^T & V_\lambda \end{bmatrix} \begin{bmatrix} \Delta x_c \\ \Delta x_p \end{bmatrix} = - \begin{bmatrix} J_c^T(b-f) \\ J_p^T(b-f) \end{bmatrix} \quad (3.37)$$

为了下文分析方便，我们设 $U = J_c^T J_c$, $V = J_p^T J_p$, $U_\lambda = U + \lambda D_c^T D_c$, $V_\lambda = V + \lambda D_p^T D_p$, $W = J_c^T J_p$, 下标 c 表示与相机参数有关的向量, 下标 p 表示与 3D 坐标点参数有关的向量, 可以将式 $\Delta x = (J^T J + \lambda D(x)^T D(x))^{-1} J^T (b - f(x))$ 写为分块矩阵形式, 如式 3.37 所示。 U_λ 和 V_λ 是分块对角矩阵, 对此可以采用 Schur Complement 高效求解该方程。考虑求解线性系统 $M \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$, 式 3.38 表示分块矩阵 M 的分解

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ CA^{-1} & 1 \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & \bar{D} \end{bmatrix} \begin{bmatrix} 1 & A^{-1}B \\ 0 & 1 \end{bmatrix} \quad (3.38)$$

where $\bar{D} = D - CA^{-1}B$

其中, A 必须是非奇异方阵, D 称为 A 在 M 中的 Schur Complement。所以使用矩阵 $\begin{bmatrix} 1 & 0 \\ -CA^{-1} & 1 \end{bmatrix}$ 左乘该线性系统, 有 $\begin{bmatrix} A & B \\ 0 & \bar{D} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ \bar{b}_2 \end{bmatrix}$, 其中 $\bar{b}_2 = b_2 - CA^{-1}b_1$, 至此我们可以得到一个降阶系统 $\bar{D}x_2 = \bar{b}_2$, 求解 x_2 后, 回带解算 x_1 。

回到光束平差法问题中，可以得到降阶系统 3.39 和 3.40，矩阵 $S = (V_\lambda - WU_\lambda^{-1}W^T)\Delta x_p$ 是 Schur Complement。

$$(U_\lambda - WV_\lambda^{-1}W^T)\Delta x_c = -J_c(b-f) + W^T V_\lambda^{-1} J_p^T (b-f) \quad (3.39)$$

$$\Delta x_p = -V_\lambda^{-1}(J_p^T(b-f) + W^T \Delta x_c) \quad (3.40)$$

S 是对称正定矩阵，使用 Cholesky 分解可以求解式 3.39。以上求解 BA 问题的方法之所以有效，是由于相机数量要远小于 3D 点数量，所以可以先求解 Δx_c ，再求解 Δx_p 。

3.7.2 最小化相机中心位置误差

上节介绍了如何使用 Schur Complement 简化线性系统，求解使重投影误差最小的 $\Delta x_c, \Delta x_p$ ，其中涉及到的变量有旋转、相机中心（平移）、3D 点坐标，而本节将介绍一种优化方法：最小化相机中心位置误差。由于从图片中，我们可以提取相机的 EXIF (Exchangeable image file format) 信息，从而得到相机的经纬度坐标。EXIF 是由数码相机制造商在图片、声音上标记的相机、图片、声音等相关内容的标准格式文件。EXIF 包含的信息丰富，可以从中找到经纬高信息，也可以从中提取相机模型，包括焦距等，也包括相机制造厂商，甚至相机位姿等。

对于同一个 3D 点在两视图中的左视图的相机坐标系中坐标为 X_1 ，在右视图的相机坐标系中坐标为 X_2 ，已知两视图相机位姿关系 R, t ，则坐标可以表示为 $RX_1 + t = X_2$ 。对于右视图相机中心这一 3D 点有 $X_2 = 0$ ，则该点在左视图的相机坐标系中的坐标为 $X_1 = -R^T t$ ，旋转矩阵是正交矩阵有 $R^T = R^{-1}$ 。所以在多视图中每个视图与第一个视图的位姿关系已知为 R_i, t_i ，则这些视图对应的相机中心坐标在第一视图的相机坐标系中的坐标为 $-R_i^T t_i$ 。这样我们就建立了每个视图的经纬高 $pos = (Lon, Lat, High)$ 与相对于第一视图的位姿关系 $-R_i^T t_i$ ，基于此我们可以建立最小二乘问题如式 3.41。

$$E_{position} = \arg \min_{R, t} \sum_{i=1} \|f_j(x) - x_j\| = \sum_i (\|-R_i^T t_i - pos\|_2^2) \quad (3.41)$$

在此不详细介绍如何解算上式，相对于最小化重投影误差的过程，该过程只调整相机相关参数，没有涉及 3D 点相关的调整，所以只能作为 BA 的辅助步骤，也可以认为这属于光束平差法的一部分，下文不再过多强调。

本节介绍了基本的稀疏点云生成的过程，包括多视图几何知识从匹配点恢复结构，而后增量式的重建其他图片；在每加入一张图片时，使用BA优化相机位姿与3D点的位置，在此我加上最小化相机中心位置误差，作为BA的辅助步骤。图 3.20(b)是使用CeresSolver库迭代三次优化重投影误差的结果，初始代价函数结果为 $3.483778e + 01$ ，BA优化后的代价为 $1.732320e + 01$ 。从 3.20两张图对比可以看出BA最小化重投影误差的作用：(a) 仅使用多视图几何计算相机位姿并使用三角测量恢复3D点位置，而(b)在(a)的基础上增加BA环节，图中红色点是提取的2D特征点坐标，绿色点是3D根据相机 R, t 关系投影回成像平面的点，从中可以明显看出BA对重投影误差有改善作用，可以将误差平摊到所有点上。

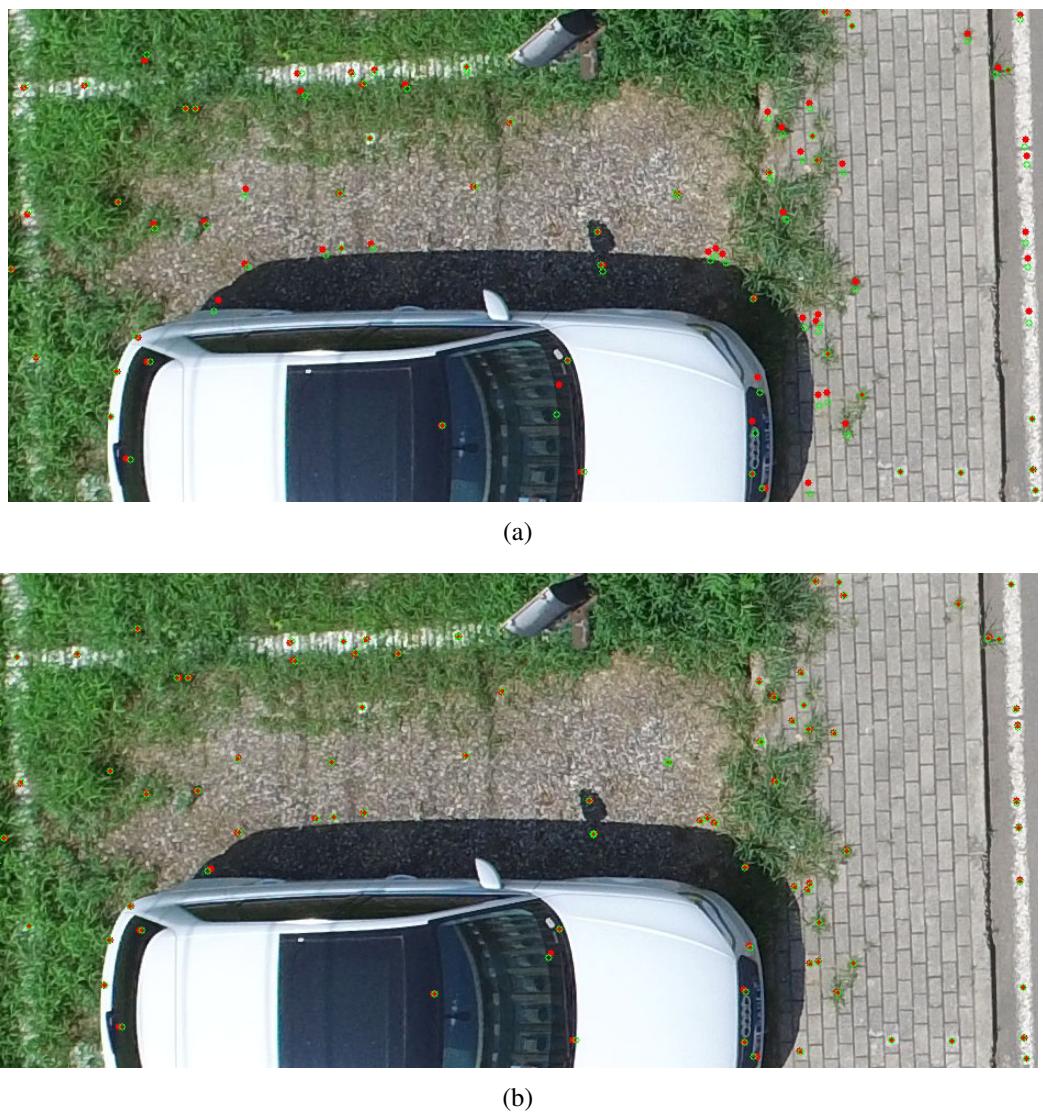


图 3.20 BA 对比结果图

3.8 稀疏点云的渲染

3.8.1 点云稠密化

3.8.2 点云网格渲染（三角剖分）

3.9 本章小结

第4章 二维图像中的道路提取

4.1 引言

最近数十年来许多方法被应用于无人地面车辆自主导航，使用地图提高定位、场景理解的方法是非常成功的。大部分无人机依靠详细路网地图导航，然后这些路网地图基本是靠手工完成的，如此限制了当前无人车辆的大范围普及。提取道路的方法被广泛的研究用于车辆导航与区域分割，这些算法对于UGV有很好的效果，然而对于UAV这些算法基本都失效。这是因为UGV的相机朝前安装而UAV相机俯视地面，UGV视野中道路在视野尽头一般收敛为消失点，针对消失点的提取道路的方法在UAV上效果较差，为了解决这一问题本文提出了使用图像二值化并结合道路拓扑关系相关内容对航拍图像进行道路提取。

4.2 图像二值化

本文采用最小二乘法提取道路区域得到二值化的图像。在实际工程中我们总是希望找到数据的线性关系，不幸的是对于实际数据找到这样完美的线性关系不容易，然而最小二乘法提供了一种可信赖的最小化残差方法，这展示了观测与拟合之间的差别。考虑到某区域的道路颜色特征基本一致的特点，本文采用线性最小二乘的方法拟合图像道路区域的RGB颜色特征，得到道路的二值化图。

首先从UAV图片中找到一部分道路区域作为训练的样本集，如图4.1中非黑色部分，再次训练集中的图片分为RGB三个通道展示在三维空间坐标系中，如图4.2中蓝色点所示，而后最小二乘法被用于拟合这些观测值（蓝色点）为一条直线 $\frac{x}{a} + \frac{y}{b} = \frac{z}{c}$ ，如图4.2中红色直线所示，最后可以计算所有观测点与拟合直线之间的平均欧式距离 D 。对于需要处理的图片，也就是测试集数据，遍历图片的每个像素，如果测试集像素与拟合直线的欧式距离大于 $3D$ 则将该像素赋值为零，否则赋值为一，进而得到和原图大小相同的二值图，按照这种方法进而每张UAV图片转化为二值图。

4.3 非道路区域移除

通过最小二乘法我们初始的道路区域，然而这有很多的噪声和错误的道路检测，为了减少噪声和误差，本文采用数学形态学精简结果。数学形态学是关于基于拓扑的



图 4.1 训练集

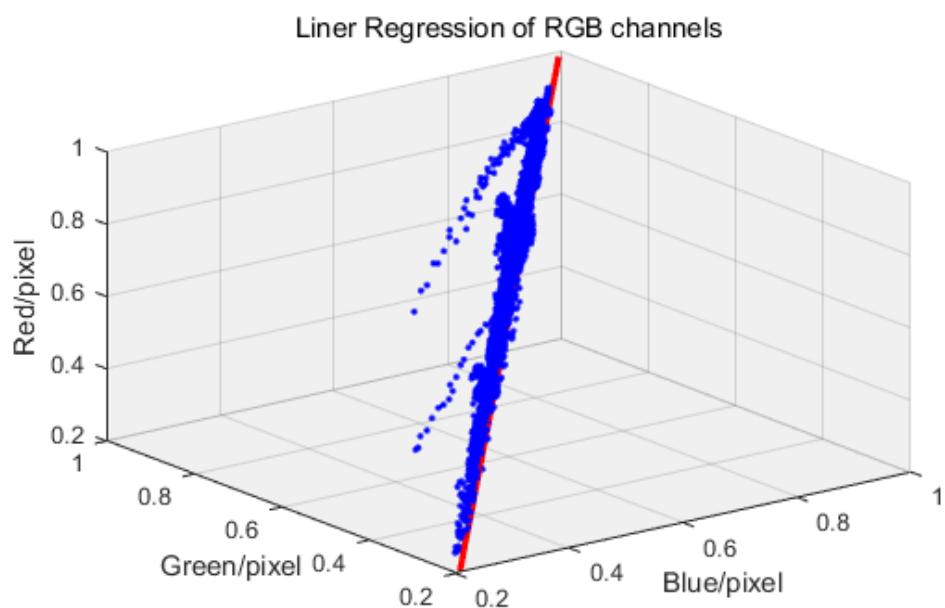


图 4.2 线性回归

图像分析学科，基本的操作有开闭运算、俯视膨胀运算和形态学梯度等。道路有显而易见的不同于图片周围物体的形态学特征——瘦长。这些拓扑特征可以被用来消除二值图中的非道路区域，精简二值图。

4.3.1 开闭运算

在最小二乘法得到的二值图中存在许多噪声，例如小孔等，这影响了道路的完整性，本文中，开闭运算被用来移除孔洞，消除噪声，平滑道路区域。开运算是先腐蚀图片后膨胀图片的运算，而闭运算正好相反，是先膨胀后腐蚀图片，式 4.2 和 ?? 所示。其中二值图像的腐蚀是指选择大小一定的核函数 K ， K 一般是方阵，则二值图像 I 被 K 腐蚀可以理解为 K 在 I 内部移动时， K 的中心经过的区域即为腐蚀后的 I 的值。二值图像的膨胀是指选择大小一定的核函数 K ， K 一般为方阵，则二值图像 I 被 K 膨胀可以理解为 K 的中心在 I 内部移动时， K 所经过的区域。腐蚀膨胀的示意图如 4.3 中左图深蓝色为腐蚀掉的像素，右图浅蓝色为膨胀后的像素。

$$\text{Open}(I, K) = \text{dilate}(\text{erode}(I, k), K) \quad (4.1)$$

$$\text{Close}(I, K) = \text{erode}(\text{dilate}(I, k), K) \quad (4.2)$$

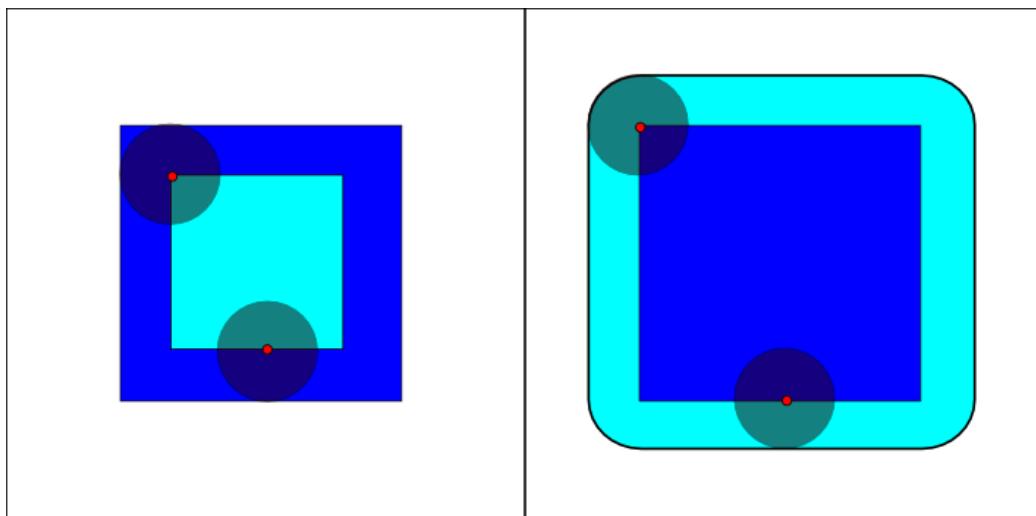


图 4.3 腐蚀膨胀示意图

基于开闭运算的方法，道路区域被平滑，大量干扰（噪声）被移除，但是仍然存在非道路区域，像房子边缘，被错误识别为道路，由于这些区域面积较大，所以基于开闭运算的方法无法移除这些区域，所以采用以下方法去除非道路道路的干扰。

4.3.2 轮廓提取

Ramer Douglas Peucker算法（简称RDP）是由Ramer、David Douglas和Thomas Peucker提出的^[16]，通过物体轮廓估计物体形状。需要移除的房子边缘被检测为满足式 4.3、4.4与 4.5的轮廓，而这些被需要移除。

$$\|C - C'\| \leq 10\%C \quad (4.3)$$

$$60^\circ \leq \theta \leq 100^\circ \quad (4.4)$$

$$\frac{L}{L'} \leq 6 \quad or \quad \frac{L}{L'} \geq \frac{1}{6} \quad (4.5)$$

其中， C 、 C' 代表原始轮廓与近似轮廓的周长， θ 代表 C' 中每两个相邻顶点连线所成的角度。 L 和 L' 代表 C' 的相邻两边。按照该式可以检测到几个两个非道路区域，如 4.4 中红色矩形所示。至此，我们完成了形态学滤波得到较为理想的道路区域，然而

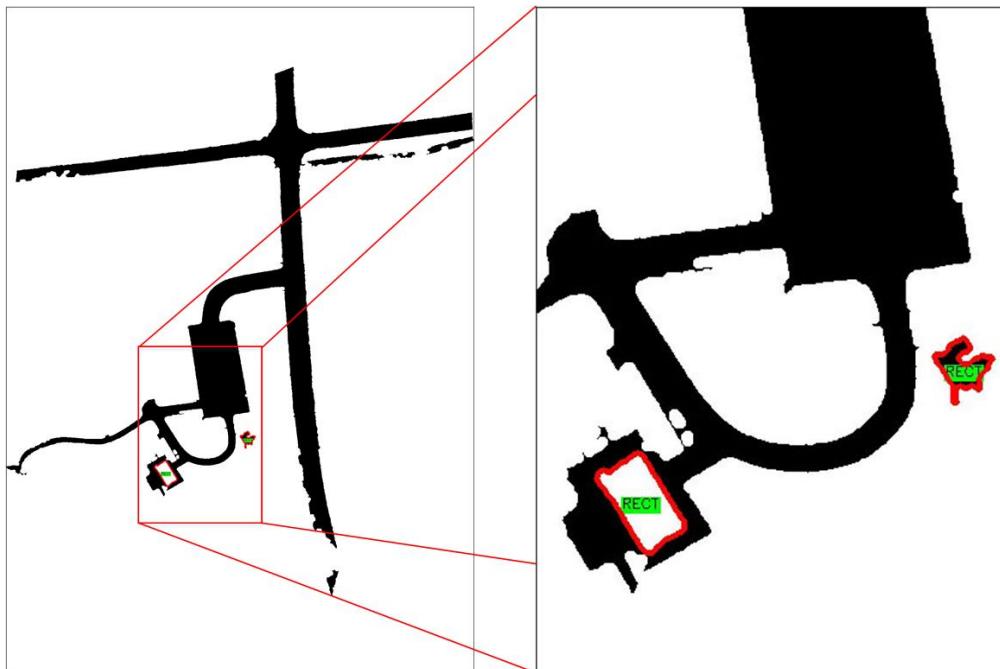


图 4.4 轮廓检测

这样的道路并不适合直接应用于无人车路径规划与导航，因为UGV路径规划需要拓扑路网，下面将介绍如何从栅格道路提取道路骨架，得到拓扑路网的过程。

4.4 拓扑处理

4.4.1 拓扑细化

栅格地图可以被简化为无人车路径规划中广泛使用的拓扑地图，这种地图有基本几何元素组成，例如节点和弧线、直线等。Choi^[17]使用拓扑细化的方法提取道路骨架，道路的宽度不会影响算法的有效性，图 ?? 显示了拓扑细化算法的过程，图中左边类似形态学处理后的道路区域，右边类似拓扑处理后的道路骨架，可以看出线段宽度不影响最后得到的道路骨架结果。拓扑细化的目的就是剥离像素，知道无法剥离

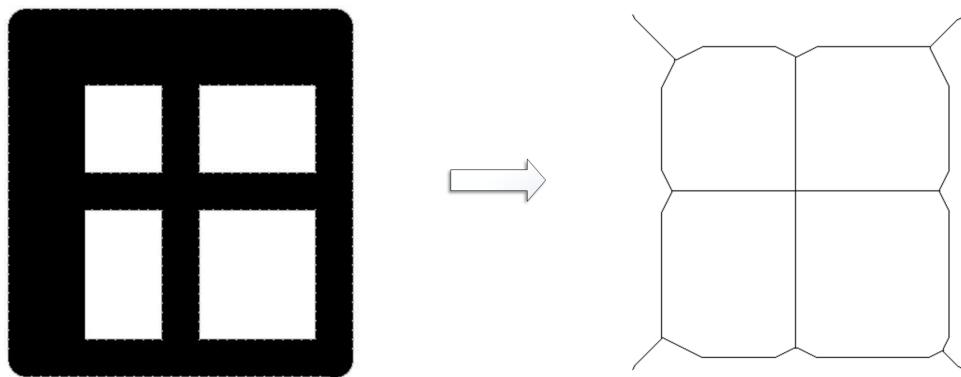


图 4.5 拓扑细化

为止，从而得到单像素构成的拓扑骨架。对于构成道路的像素是否可以剥离取决于像素的连通数，连通数的概念由式 4.6 和图 4.6 定义。

$$C = \sum(N_k - N_k N_{k+1} N_{k+2}), k = 1, 3, 5, 7, N_9 = N_1 \quad (4.6)$$

N_1	N_2	N_3				
N_8	N_0	N_4				
N_7	N_6	N_5				

(a) 8-region connectivity (b) Connectivity Example (c) Branch Example

图 4.6 Connectivity

在 4.6(a)中 N_i 表示以 N_0 为中心的8邻域中第*i*个像素大小，例如 $N_1 = 1, N_2 = 0$ ，二值图像中像素值非0即1；根据式 4.6可以计算某像素的8邻域中连通数的大小。如果一个像素的连通数等于1并且该像素不是端点（拓扑的起点或者终点），那么我们可以简化删除该像素简化拓扑关系。按照这一原则对于图 4.6(b)，其中黑色圆形对应的像素值为1，白色为0， N_0 的连通数 $C = 2$ ，所以 N_0 无法删除；此时如果将图 4.6(b)的 N_6 设为1， N_0 的连通数变为1，且 N_0 不是端点，所以 N_0 可以被移除；同样如果 $N_4 = N_5 = N_7 = N_8 = 0$ ， N_0 不可移除，尽管 $C_{N_0} = 1$ ，但 N_0 为端点。按照拓扑细化的规则，遍历图片中所有像素，对每个像素采用拓扑细化的方法就可以得到基本的道路骨架。

4.4.2 去除小枝丫

拓扑细化的规则虽然能够得到保持道路的连通、完整，但是同时产生冗余的小枝丫，如图 4.5所示所示，本文提出了一种从道路骨架中移除小枝丫的方法。在给出介绍之前，先定义一下连通度的概念，不同于应用于拓扑细化的连通数概念，连通度的定义如下：

- (1) 一个像素的连通度：在8邻域中，一个像素直接与之相邻的像素个数，图 4.6(b)中心像素的连通度是5。
- (2) 道路骨架的端点：在8邻域中，一个像素与之相邻的像素仅一个，即连通度为1。
- (3) 道路骨架的节点：一个像素与之相邻的像素个数大于2，例如 4.6(b)中心像素即为拓扑的节点，换句话说像素连通度大于2。
- (4) 枝丫的长度：从起点到最近节点的像素个数。

基于以上定义，我们可以推断图 4.6(c)中 $P(1,2), P(2,5), P(5,6), P(8,4)$ 是端点， $P(3,4), P(5,4)$ 是道路节点，枝丫长度对于端点 $P(1,2), P(2,5), P(5,6), P(8,4)$ 分别为3, 1, 3, 4。实际道路枝丫通常较长，在 4.5中短的枝丫很显然并不属于真实道路骨架，设计逐渐增大的阈值，如果枝丫长度小于初始较小阈值，则该枝丫被移除。阈值逐渐增大，小于该阈值的枝丫被删除，直到阈值到达设定最大值，该过程结束。在图 4.6(c)中，黑色点将被移除，剩下的绿点就是道路骨架。图 4.7展示了阈值逐渐增大过程中，拓扑路网的简化过程。

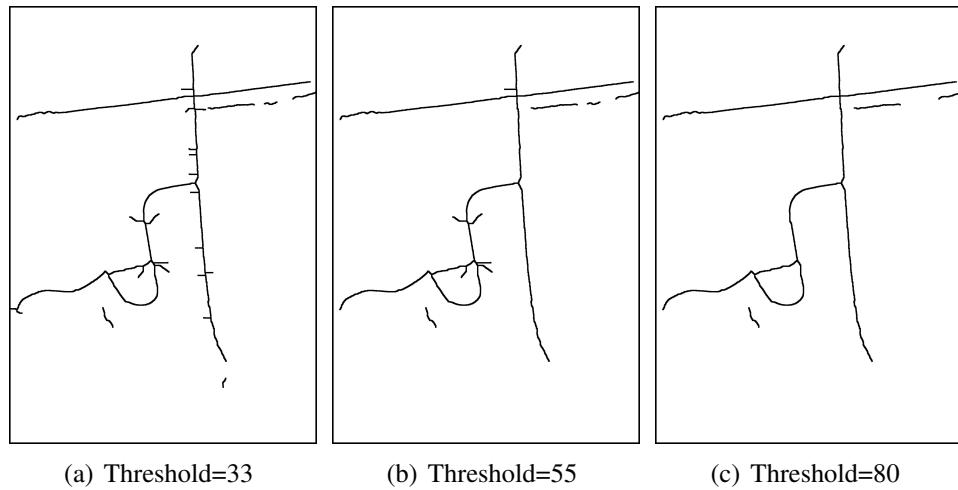


图 4.7 Branches Elimination

4.5 本章小结

本章提出使用最小二乘法拟合道路区域得到二值图而后使用开闭运算等方法平滑道路区域，之后采用RDP算法提取道路骨架得到适合无人车辆路径规划的拓扑网络，经过拓扑细化的道路区域虽然保持连通性，但也存在过多的冗余小枝丫，本文设计了通过计数枝丫长度的方法去除小枝丫的方法。本章的方法简单实用，但对于复杂的道路仍然存在少量错误提取的道路网络，而这些需要手动加以修改。

第 5 章 实验结果分析

5.1 引言

5.2 全景图像构建地图实验结果

5.2.1 全景图像地图实验平台搭建

5.2.2 全景图像地图构建效果及精度分析

5.3 地理空间三维重建实验结果

5.3.1 地理空间三维重建实验平台搭建

5.3.2 三维重建效果及精度分析

5.4 道路提取精度分析

5.5 本章小结

总结与展望

工作总结

这里是总结。

未来工作展望

这里是展望。

参考文献

- [1] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [2] Engel J, Schops T, Cremers D. LSD-SLAM: Large-scale direct monocular SLAM[C] European Conference on Computer Vision. Springer, Cham, 2014: 834-849.
- [3] Zhang J, Singh S. LOAM: Lidar Odometry and Mapping in Real-time[C]//Robotics: Science and Systems. 2014, 2.
- [4] Carrivick J L, Smith M W, Quincey D J. Structure from Motion in the Geosciences[M]. John Wiley & Sons, 2016.
- [5] Mattyus G, Wang S, Fidler S, et al. Enhancing Road Maps by Parsing Aerial Images Around the World[C] IEEE International Conference on Computer Vision. IEEE, 2015:1689-1697.
- [6] Lowe D G, Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110.
- [7] Bay H, Ess A, Tuytelaars T, et al. Speeded-Up Robust Features (SURF)[J]. Computer Vision & Image Understanding, 2008, 110(3):346-359.
- [8] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[C] IEEE International Conference on Computer Vision. IEEE, 2012:2564-2571.
- [9] 高翔, 张涛等, 视觉SLAM十四讲: 从理论到实践[B] 电子工业出版社, 2017-3:138
- [10] Rosten E, Drummond T. Machine learning for high-speed corner detection[J]. Computer Vision – ECCV 2006, 2006: 430-443.
- [11] Harris C, Stephens M. A combined corner and edge detector[C] Alvey vision conference. 1988, 15(50): 10.5244.
- [12] Lindeberg, T.: Feature detection with automatic scale selection. Int. J. Comput. Vis. 30(2), 79 – 116 (1998)
- [13] Lowe D G. Object Recognition from Local Scale-Invariant Features[C] iccv. IEEE Computer Society, 1999:1150.
- [14] Multiple View Geometry in Computer Vision Second Edition, Richard Hartley and Andrew Zisserman, Cambridge University Press, March 2004:105

- [15] Hartley R, Zisserman A. Multiple view geometry in computer vision[J]. *Kybernetes*, 2003, 30(9/10):1865 - 1872.
- [16] P. S. Heckbert and M. Garl, “Survey of polygonal surface simplification algorithms,” 1997.
- [17] C.-H. Choi, J.-B. Song, W. Chung, and M. Kim, “Topological map building based on thinning and its application to localization,” in *Intelligent Robots and Systems*, 2002. IEEE/RSJ International Conference on, vol. 1. IEEE, 2002, pp. 552 – 557.

攻读硕士学位期间发表论文与研究成果清单

项目成果

- [1] 航天八院目标位姿视觉测量项目

致谢