



单位代码 10006

学 号 16031144

分 类 号 \_\_\_\_\_

密 级 公开

# 北京航空航天大學

B E I H A N G   U N I V E R S I T Y

## 毕业设计(论文)

### 大视差多摄像机图像拼接算法研究

院（系）名称 自动化科学与电气工程学院

专业名称 自动化

学生姓名 吴炳坤

学院指导教师 张海

工业部门导师 \_\_\_\_\_

2020 年 6 月

# 北京航空航天大学

## 本科生毕业设计（论文）任务书

I、毕业设计（论文）题目：

大视差多摄像机图像拼接算法研究

II、毕业设计（论文）使用的原始资料（数据）及设计技术要求：

原始材料：典型大视差图像样本及实验验证实地采集的图像。

技术要求：

1. 学习图像拼接相关的射影几何知识；

2. 完成图像特征匹配方法学习，并在程序设计基础上对比、分析面向图像拼接的图像特征的适用性；

3. 实现基于 APAP 架构的大视差图像拼接与融合处理，并利用线约束、先验信息等不同手段改进处理效果；

4. 完成拼接软件设计与调试验证。

III、毕业设计（论文）工作内容：

研究对比基于不同方法的图像拼接算法，在充分理解原理的基础上，设计完整图像拼接算法并自主实现算法各环节的主要程序。将本设计所实现的方法与以往方法进行对比，并从原理及实验结果的角度对各功能组件的实

际作用加以分析，以体现大视差问题对图像拼接的影响及针对性的解决方法。

---

#### IV、主要参考资料：

##### 1. 图像拼接相关学位论文

---

##### 2. 关键英文文献

---

(1) Brown, Lowe. Recognising panoramas[A]. Proceedings Ninth IEEE International Conference on Computer Vision[C]. Nice, France: IEEE, 2003: 1218–1225 V2.

---

(2) Zaragoza J, Chin T-J, Brown M S, et al. As-Projective-As-Possible Image Stitching with Moving DLT[A]. 2013 IEEE Conference on Computer Vision and Pattern Recognition[C]. Portland, OR, USA: IEEE, 2013: 2339–2346.

---

##### 3. 课题组研究文档

---

自动化科学与电气工程 学院（系） 自动化 专业类 160323 班  
学生 吴炳坤

毕业设计（论文）时间： 2020 年 12 月 23 日至 2020 年 6 月 4 日

答辩时间： 2020 年 6 月 3 日

成 绩： \_\_\_\_\_

指导教师： 张海

兼职教师或答疑教师（并指出所负责部分）：

---

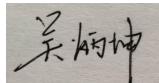
---

\_\_\_\_\_ 系（教研室）主任（签字）： \_\_\_\_\_

## 关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中做出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

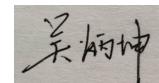
学位论文作者签名： 

日期： 2020 年 6 月 7 日

## 学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

本论文不涉及任何保密内容。若有涉密内容，本人愿意承担相关法律责任。

学位论文作者签名： 

日期： 2020 年 6 月 7 日

学院指导教师签名： \_\_\_\_\_

日期： 2020 年 6 月 7 日



# 大视差多摄像机图像拼接算法研究

学 生：吴炳坤

指导教师：张 海

## 摘 要

图像拼接是计算机视觉中的一个基础任务，在监控系统、虚拟现实、遥感技术等领域有着广泛应用。传统的图像拼接算法虽然已经被成熟地集成在商业软件中，但其基于全局单应性的配准变形方法对于解决拼接过程中的视差问题有着固有缺陷。本文第一章结合拼接算法的应用背景，介绍了其发展历史沿革中不同方法的优点与缺陷。基于这些分析并旨在解决大视差问题对拼接效果带来的不利影响，实现了一套完整的图像拼接处理流程。第二章主要介绍所实现方法的特征提取模块，第三章介绍算法核心的图像配准与网格优化部分，第四章介绍终末处理并介绍了整个算法的一些软件实现细节。所设计的图像拼接方法在多个环节均提出了针对性的改进，以解决拼接全景图中的投影变形、图形结构破坏等问题。在第五章中，所实现的算法与以往方法及商业软件进行了充分的对比，结合主观感受与失真、结构破坏的定量分析，说明了所实现方法的优越性。实验的最后，算法各环节的组件被逐一对比测试，以说明其在拼接效果中的具体功能，与理论部分相互印证。研究设计的图像拼接算法为高层次计算机视觉任务提供了良好基础，其中的许多组件也为进一步优化图像拼接效果提供了思路。

**关键词：**图像拼接，计算机视觉，图形学，全景图像，多目视觉



# Research on Image Stitching Algorithm of Multi-Camera with Large Parallax

Author : Wu Bingkun

Tutor : Zhang Hai

## Abstract

Image Stitching is a fundamental task in Computer Vision, it is an important link in Surveillance System, Virtual Reality, Remote Sensing and so many other complicated applications. Although traditional stitching algorithm has already been integrated into commercial software nicely, its alignment and deformation based on global homography has inherent defects in solving the parallax problem in the process of stitching. In the first chapter of this paper, the advantages and disadvantages of different methods in the development history are introduced. Based on these analysis and aimed to solve the adverse effect of large parallax, a complete set of image stitching processing flow is realized. The second chapter mainly introduces the feature extraction module, the third chapter introduces the image alignment and mesh optimization part of the algorithm core, the fourth chapter introduces the final processing and some software implementation details of the whole algorithm. In order to solve the problems of projection deformation and graphic structure damage, the improved image stitching method is proposed in many aspects. In the last chapter, the algorithm realized in this design is fully compared with the previous methods and commercial software. At the end of the experiment, the components are tested one by one to show their specific functions, which is verified with the theoretical part. The image stitching algorithm proposed in this study provides a good foundation for high-level Computer Vision tasks, and many of its components also inspire ideas for further optimization.

**Key words :** Image Stitching, Computer Vision, Graphics, panoramic image, multivision



## 目 录

<b>第一章 绪论</b>	1
1.1 课题背景及目的	1
1.2 国内外研究现状	2
1.3 课题研究方法	5
1.4 论文构成及研究内容	6
<b>第二章 图像拼接过程中的图像特征匹配</b>	8
2.1 图像读取与文件解析	8
2.2 图像点特征匹配	9
2.2.1 SIFT 特征点匹配	10
2.2.2 ORB 特征点匹配	11
2.2.3 SIFT 与 ORB 的对比及本设计采取的方案	12
2.3 图像线特征匹配	13
2.4 多平面 RANSAC 筛选特征匹配对	16
2.5 本章小结	21
<b>第三章 拼接图像配准与网格优化变形</b>	22
3.1 图像配准 LAPAP	22
3.2 网格优化变形	25
3.2.1 区域配准项	26
3.2.2 局部相似项	28
3.2.3 全局相似项	30
3.2.4 线性对齐项	32
3.2.5 线性保持项	34
3.2.6 待拼接子图的旋转与尺度计算	36
3.3 网格优化问题的解算	38
3.4 本章小结	38
<b>第四章 拼接终末处理及全流程软件设计</b>	39
4.1 曝光补偿	39
4.2 图像融合	39



4.2.1 平均融合与线性融合.....	39
4.2.2 最优缝合线.....	40
4.3 先验信息辅助.....	41
4.4 全流程软件实现.....	43
4.4.1 项目架构概览.....	43
4.4.2 图像线特征提取与匹配程序实现.....	43
4.4.3 图像配准及网格优化程序实现.....	44
4.5 本章小结.....	46
<b>第五章 拼接算法对比与评价.....</b>	<b>47</b>
5.1 大视差问题.....	47
5.2 算法效果对比与分析.....	50
5.2.1 直观评价.....	50
5.2.2 投影变形评价.....	54
5.2.3 线性结构保持评价.....	57
5.3 各组件作用分析.....	60
5.3.1 LAPAP 优化配准.....	60
5.3.2 线特征相关网格优化约束项.....	61
5.3.3 图割优化.....	62
5.3.4 曝光平衡.....	63
5.3.5 先验信息.....	64
5.4 本章小结.....	65
<b>结 论.....</b>	<b>66</b>
<b>致 谢.....</b>	<b>67</b>
<b>参考文献.....</b>	<b>68</b>
<b>附录 A 图像读取与文件解析程序片段.....</b>	<b>72</b>
1 无法以字典序读入文件的原始程序.....	72
2 以字典序读入文件.....	72
<b>附录 B 线特征及多平面 RANSAC 相关说明程序.....</b>	<b>73</b>
1 线特征信息转换.....	73



2 Sequential RANSAC 实现程序..... 73



# 第一章 绪论

图像拼接算法在许多计算机视觉任务中充当基础性环节，随着生活中视觉传感器的增多，图像拼接所生成的全景图成为了获取大场景深层次视觉信息的重要手段。从上世纪70年代开始，对于图像拼接效果的改进从未停止，算法的复杂性也逐渐攀升。

## 1.1 课题背景及目的

图像拼接技术就是把数张具有重叠区域的图像拼接成一幅包含各图像序列信息的宽视角场景的新图像的技术，这些待拼接图像有可能是在不同视角，不同时间或由不同相机拍摄的。随着视觉传感器在越来越多的领域被采用，图像拼接作为一个计算机视觉的基础任务在多种应用场景中发挥着重要的作用<sup>[1]</sup>。

基于安防等考虑，公共场所的摄像监控设备逐渐增多<sup>[2]</sup>。目前的大部分监控系统以单个摄像机覆盖的视场作为信息输入，由人为浏览获取信息以识别特定目标。而对于大范围视场的监控，往往需要多个不同视角的监控摄像机同时获取信号并反映在多个终端上，这样的离散图像为浏览分析监控的人员带来了不便与低效。显然，对于不同视角的泛在摄像机，如果能将具有重合区域的图像拼接为效果良好的全景图像，将对整个监控场景的分析起到极大的辅助作用。

在如自动化仓储机器人等移动机器人上，单目视觉视野范围较小不能满足机器人导航的需要。为提升机器人视觉导航的能力，可以将多目摄像机采集的图像利用图像拼接技术进行合成，以达到更大的视野。在虚拟现实领域中为了虚拟实际场景，图像拼接技术常常被用于构建360° 全景图像。这种基于全景图的虚拟现实系统，通过提取全景图深度信息，恢复场景的三维信息，进而建立三维模型。

在医学图像处理方面或遥感技术领域中，不完整的图像被拼成整体才可以使远程医疗或地图建模等任务得到充分准备。

由此可见，图像拼接的效果直接影响众多计算机视觉任务的后续环节与最终表现，只有得到良好的拼接图像，才可以继续结合具体应用场景提出有应用价值的方案与分析。

图像拼接算法从上世纪 70 年代开始得到了不断的发展，单应性矩阵变换、局部单应性、网格优化以及图割优化等不同组件或思路被广泛尝试于图像拼接流程中，用以应



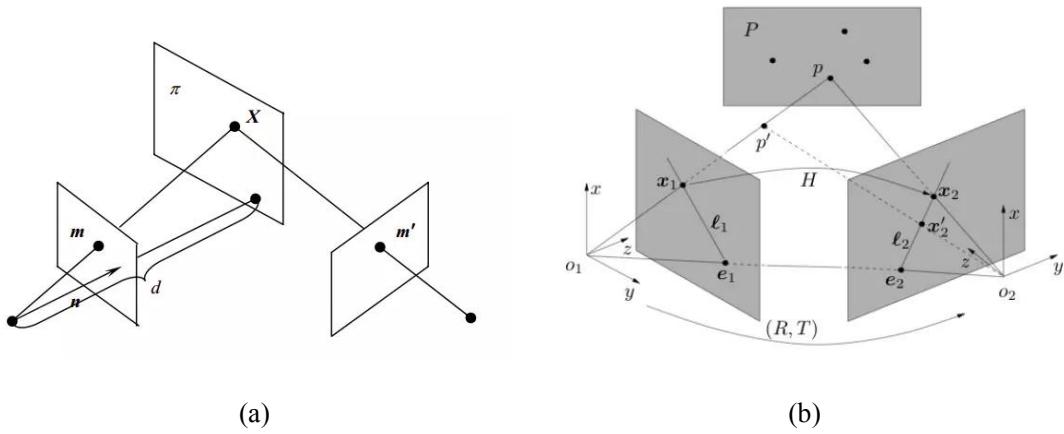
对拼接过程中出现的诸多挑战，包括配准、失真等。学术界深入的研究使得商业化软件中的图像拼接效果及观感自然度有了长足的进步，但在图像间大视差、重叠区域低纹理，前景物体较多，拍摄设备旋转或平移等极端条件下，现有方法的表现仍然差强人意。

本研究将大视场多视角视觉传感器信息结合，以图像拼接作为主要攻克对象，与传统图像拼接问题相比，需要解决大视差、多视角、多张待拼接图片等问题，这些问题的优化对于图像拼接问题将有极大的推动，从而促进虚拟现实，室内建图等技术的发展。本研究将首先研究比对静态图像拼接算法，选取表现良好的多幅图片拼接算法并参考、改进，使得拼接出的全景图片失真小，过渡平滑自然。在大视差，重叠区域少的多张图片上，产出一种效果优良的端到端（End-to-End）拼接算法。实验部分除了与以往方法进行直观的观感自然度对比，还需要针对影响观感的失真等内容进行分析，说明本研究引入的多个组件在优化图像拼接中所发挥的作用。

## 1.2 国内外研究现状

最早的图像拼接算法是相位相关法，于1975年由Kuglin和Hines提出，该算法将原始图像变换到频域进行配准，这种算法仅在无视差的条件下可以工作<sup>[8]</sup>。随着21世纪计算机性能与技术的进步，传统的拼接算法以全局单应性矩阵、柱面或球面投影以及多频带融合为核心得到了成熟的发展。Brown在2003年与2007年提出著名的AutoStitch是这一系列方法的里程碑，这是一种基于尺度不变特征转换(SIFT)的图像拼接算法，该算法以全局单应性矩阵对齐图像、柱面或者球面投影、光束法平差（Bundle Adjustment）和多频带融合（Multi Band Blending）核心，较好地实现了单视点条件下多幅图像的无缝合理拼接<sup>[13]</sup>。各种基于AutoStitch的拼接软件与应用纷纷落地，例如基于OpenCV实现的Stitcher，微软的ICE和Photoshop中的拼接工具。

虽然传统以AutoStitch为代表的算法效果良好，但仍存在一些问题与限制。首先，这类算法要求图像之间的重叠区域最好处于同一个视觉平面，想要达到最佳效果则要求整个图像处于同一平面。其次，是要求各次拍摄时相机光心近乎重合。这两个条件可以合并为重叠区域场景中要求没有深度变化，否则就会产生明显的伪影和重影。这就是视差问题带来的图像拼接最大难点：一个单应性矩阵实现了同一视觉平面的像素配准，而位于其他平面的像素点则无法根据这个单应性矩阵实现对齐。图1.1示意了全局单应性的配准。



(a)图表示一个单应性矩阵将空间中一点X在目标图像上对应的点 $m$ 映射到参考平面上一点 $m'$ , (b)图表示当空间中存在另一点 $p'$ 与 $p$ 点不在同一平面时, 全局单应性矩阵会将 $p'$ 与 $p$ 映射至参考平面同一位置, 对于空间点 $p'$ 这显然是错误配准。这就是单应性变换的视差问题。

图1.1 全局单应性矩阵配准模型

对于上述问题, Gao J与Kim S J等人于2011年提出的Dual-HomographyWarping(DHW)用两个单应性矩阵将背景平面和前景平面分别配准变形<sup>[16]</sup>。而Lin W Y等人在同年提出的Smoothly Varying Affine (SVA), 使用了更多的仿射变换矩阵, 更多的自由度使为局部对齐与变形带来了灵活性, 视差问题得到一定的解决<sup>[17]</sup>。可以看出, 使用更多更强的矩阵变换为处理视差问题带来了可能性。Zaragoza J与Chin T J等人在2013年与2014年提出的As-Projective-As-Possible (APAP) 将这一思想发展到了极致, 该算法利用网格将图像完整覆盖, 每个网格都用一个单应性矩阵对齐, 称为局部单应性<sup>[20]</sup>。其Moving DLT的计算方法也使得多个单应性矩阵的计算效率得到保证。

在相邻平面视差变化较大时, APAP虽然能达到不错的重叠区域配准效果(Alignment Quality), 但存在多视角情况下的投影变形(Projective Distortion)与透视失真(Perspective Distortion), 造成其可拼接的视角有限, 非重叠区域失真形变也较严重, 从而使多张图像的拼接存在困难。2014年由Zhang F和Liu F提出的Parallax-tolerant Image Stitching, 借鉴经典视频去抖方法Content-preserving Warps (CPW) 的优化项和缝合线主导的Seam-driven(13’Eurographics), 大幅提高了大视差场景的拼接性能<sup>[22]</sup>。为了进一步减少透视变换对非特征区域带来的变形, 2014年的Shape-Preserving Half-Projective (SPHP)借鉴形状矫正的思想, 追求拼接图像形状的矫正<sup>[21]</sup>。该方法引入图像缩放的Shape-Preserving类方法, 在图像的非重叠区域增加相似变换约束, 使其可以过渡为全局相似变换, 减少了透视变换带来的失真<sup>[7]</sup>。



可以看到，约束项对网格形式的局部单应性拼接方法带来了进一步的优化。由Lin C等人于2015年提出的Adaptive As-Natural-As-Possible（AANAP）依然在网格优化的基础上采用自适应确定旋转角度以及全局相似变换矫正形状<sup>[23]</sup>，拼接结果的观感自然度比SPHP提升了一个档次。2016年，Chen Y S等人提出的Global Similarity Prior（NISwGSP）使用APAP初始化网格，引入局部与全局相似约束项，并使用直线对齐约束确定全局相似矩阵的角度选择，多图拼接性能与观感自然度进一步提升<sup>[25]</sup>。同年，Lin K等人在算法的基础上，改进了缝合线处理、轮廓检测和直线检测，加入了曲线和直线结构保持约束项的结构保持绘制，提出的Seam-guided Local Alignment（SEAGULL）也有不俗的大视差处理效果<sup>[26]</sup>。

总的来说，APAP这一里程碑式的算法以网格局部单应性为核心，已经很好地解决了图像与图像间重叠区域配准的问题（Alignment in Overlapping Area），之后的研究在APAP的基础上，主要致力于解决拼接图片的观感自然度提升与失真变形的消除。在失真变形中主要存在两种，一是投影变形（Projective Distortion），体现为非重叠区域的物体相比原始图片在拼接后被不恰当的放大；二是透视失真（Perspective Distortion），体现为同一物体在两张图片中的透视关系（视角）不一致，视觉上会导致同一物体在拼接接缝处两侧的比例不一致<sup>[30]</sup>。

Zhang G等人于2016年提出一种生成宽基线场景的正交投影，在变换中保留了提取出来的线段。2015年Li和Yuan等人提出的Dual-Feature Warping（DFW）将线特征引入拼接算法，点线结合的特征提取增强了低特征区域的配准精度并避免了一些失真<sup>[24]</sup>。Xiang T等人通过提取高层特征的方式，在重叠区域的纹理稀疏区域加入了线特征匹配以补充点特征的不足，类似的更为先进的特征提取为图像配准带来了利好<sup>[31]</sup>。为了保证场景中特点目标的观感自然，Herrmann等人采取深度学习的方法（如：Mask R-CNN）建立了以目标检测为核心的缝合线生成方法，在图像拼接中优先保障了特点语义目标的观感自然度<sup>[29]</sup>。2018年，Nan Li等人提出了评估拼接图片观感自然度的方法（Naturalness Quality）并给出了Quasi-homography Warps用以平衡上文介绍的两种失真的影响。

将上述介绍的方法再次进行细致分类，目前比较成熟的图像拼接算法分为以下几类：

### （1）全局对齐方法（Global Alignment）

以AutoStitch为代表，利用一个单应性矩阵对齐图像，适合视差极小即重叠区域大的场景。



### (2) 空域变化变换方法 (Spatially-Varying Warping)

以局部单应性及网格优化为主，代表算法较多。

### (3) 缝合线主导方法 (Seam-Driven)

以缝合线为主导，不需要严格对齐整个重叠区域，而是仅对齐缝合线附近的区域，以Parallax-Tolerant和SEAGULL为代表，对大视差的处理效果也较好。

针对目前图像拼接的研究现状，主要挑战依然是大视差场景下的图像配准与失真消除（观感自然度提升）。这其中图像配准与重叠区域特征匹配等因素有关，而失真消除则要依靠拼接前投影变形时的约束项。

## 1.3 课题研究方法

本研究通过调研相关文献，选取当下较为先进的基于网格优化的拼接算法NISwGSP 作为基本框架。深入分析该算法及其他经典算法的优劣后，进一步调研相关文献并针对性选取一系列对拼接效果有所增强的组件，在原始框架的基础上改进、融合。选取的组件包括 LAPAP、线性约束项、最优缝合线与先验信息提取等，同时也对原框架的文件解析、特征匹配及筛选等模块进行了改进，提升程序的健壮性与拼接效果。结合这些功能模块的开发，本研究从理论角度描述了拼接流程中各组件的功能，并进行了一系列的消融实验说明各组件在拼接效果中的实际功用。开发环境选用 Ubuntu16.04 + C++11，最终开发得到一个可执行程序用于对多图对象进行通用且操作简单的拼接，本设计产出的拼接算法效果优于近年来一些文献中提出的方法，且显著优于成熟商业软件中的图像拼接功能。最后，本研究设计了充分的对比实验，从主观观感自然度与图形拼接前后的失真程度分析了所设计算法的优越性。

### (1) 基础框架——基于网格优化的 NISwGSP

NISwGSP 是近几年提出的较为典型的基于网格优化的图像拼接算法，其大致流程框架如图 1.2：



图 1.2 基于网格优化的拼接算法基本流程



本研究按照文献及开源代码复现了此算法,由于其利用了局部单应性的配准思路及网格优化变形的精细化处理,在拼接时具有相当高的自由度,故基础效果已经较为优秀。通过对其配准表现一般、线性结构丢失、存在伪影重影以及多传感器曝光不一致等问题的分析,引入多种组件加以改善。

### (2) 引入的组件

在众多不同思路的拼接算法以及其他计算机视觉研究中,有许多处理环节与组件对解决拼接过程中的问题有所帮助。本设计首先优化了开源代码中文件读取接口,使得程序的通用性及健壮性有所提升。针对图片重合部分两两配准不够准确的问题,本设计引入线特征匹配对点特征进行补足,并设计配准算法与约束项,利用线特征增强图像配准与网格变形效果,这也使得图像中的线结构得到了更好的保留。与线特征有关的约束项与配准及相似性约束项一起构成了网格优化问题的约束项,网格优化是整个算法的核心,其本质是  $AX = B$  的线性方程组求解问题,也是保证拼接图像观感自然度的关键组件。后续处理中为了消除拼接图像中的重影与伪影,引入了最优缝合线代替平均融合或线性融合完成各网格变形后拼接子图的融合。针对不同传感器或同一传感器在不同拍摄参数、拍摄时间下曝光有所差异的情况,引入拼接子图的增益补偿环节。对于视差较大或低纹理重叠区域的情况,开发了手动选取匹配特征的先验信息提取环节,对于极端情况的配准及网格优化又有了进一步的提升。

### (3) 实验对比及分析

在完成本设计的拼接算法后,将所完成算法与基于全局单应性的经典方法 AutoStitch、基于网格优化的 NISwGSP、应用商店中排名靠前的图像拼接商业软件以及智能手机自带的全景图拍摄进行对比。除了对拼接结果的主观观感评价,本研究提取出原始待拼接图像与拼接结果中的特定前景目标进行失真衡量,对拼接结果的投影变形与透视失真进行量化分析与比较。为了验证拼接流程中各组件的实际作用,本研究亦进行了消融实验(Ablation Study)进行说明。

## 1.4 论文构成及研究内容

论文主体共包含 5 部分,除第一章绪论外,还包含第二章图像拼接过程中的图像特征匹配,第三章拼接图像配准与网格优化变形,第四章拼接终末处理,第五章拼接算法对比与评价。其中,第二章至第四章主要按照程序工作顺序介绍本研究所设计实现的图

像拼接算法各环节。第五章是实验对比分析。

所设计算法的流程如图 1.3 所示：

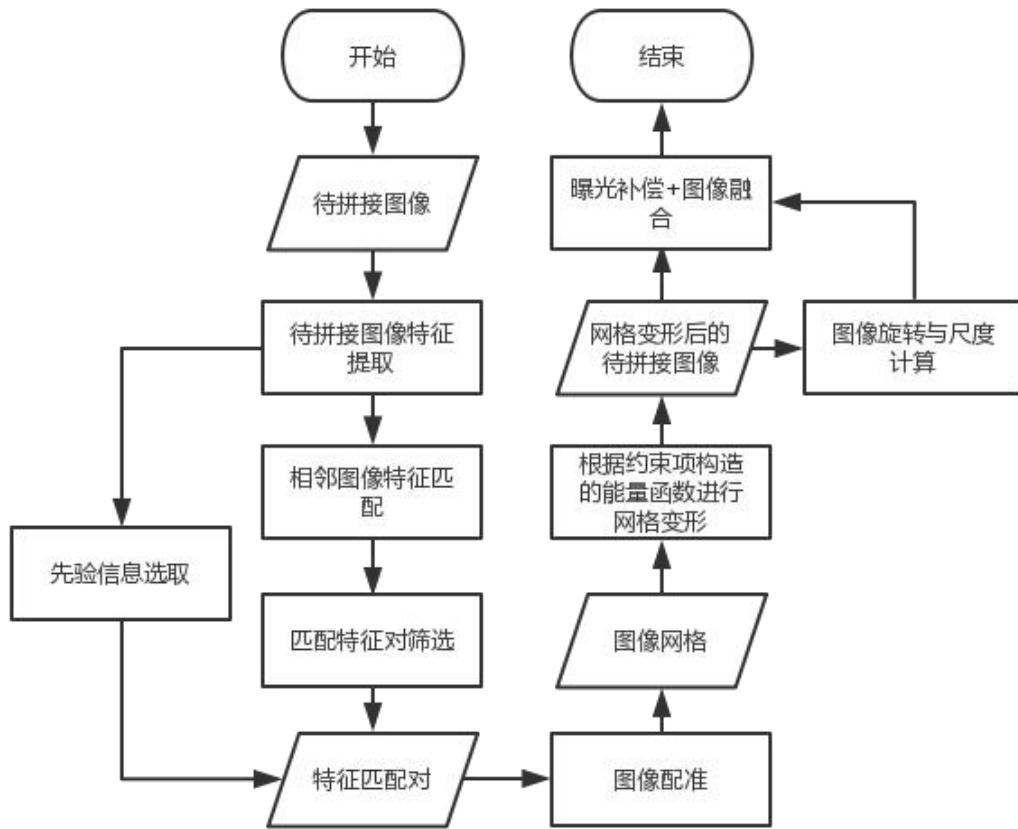


图 1.3 本设计所实现图像拼接算法的流程

第一章绪论介绍了主要研究对象拼接算法的应用背景与研究现状，概括了本设计采取的主要流程方法与本研究所包含的内容。

第二章对应拼接算法的预处理、点线特征提取与匹配和特征匹配对筛选。

第三章介绍拼接算法核心部分图像配准及网格优化变形的原理，提出本研究改进后的设计。

第四章介绍优化变形后各待拼接子图的融合方法，曝光补偿及先验信息辅助拼接的原理，并说明了本设计的完整拼接算法软件实现框架及主要程序细节。

第五章包含本设计算法与以往方法的对比与分析，由拼接结果分析及消融实验组成，对算法中不同组件的作用加以分析。



## 第二章 图像拼接过程中的图像特征匹配

本章首先介绍了读取图像的接口设计,如何储存图像间的相邻关系对图像拼接有重要意义。之后,介绍了两种点特征提取匹配的方法并对比优劣,选择本设计采用的点特征匹配方法。紧接着介绍了本设计采用的线特征匹配方法。特征匹配对的筛选采用多平面 RANSAC,在本章最后进行了说明。

### 2.1 图像读取与文件解析

对于图像拼接任务,当拼接两幅图像时只需简单读入。对于可以拼接多幅图像的算法,可以简单按空间顺序读入,确保拼接时相邻两幅图像在空间上也具有重叠部分。这种情况下,拼接算法在待拼接图像间两两运行,同一幅图像只与一幅相邻图像进行配准,这在拼接一些宽场景时较为适用。然而在拼接一些集中区域场景时,同一幅待拼接图像可能与多张其他待拼接图像相邻,拼接算法就需要在这多组相邻关系中运行以取得更好的拼接效果。图像读取对拼接的影响如图 2.1 所示。



(a) 宽场景只需依次读取, 两两拼接



(b) 同一幅图像只与一幅相邻图像配准



(c) 拼接算法在多组相邻关系中运行



图 2.1 图像读取对拼接的影响

如图 2.1，集中区域场景若同一幅图像只与一幅相邻图像配准，则红框中出现重合区域未配准形成的伪影，全景图像部分区域也没有得到良好优化与旋转。

对于一组待拼接图像，可以使用一个配置文件记录其图像间的相邻信息等。图 2.2 是一个配置文件内容示例：

```
{center_image_index | 2 | center image index}  
{center_image_rotation_angle | 0 | center image rotation angle}  
{images_count | 6 | images count}  
{matching_graph_image_edges-0 | 1,2,3 | matching graph image edge 0}  
{matching_graph_image_edges-1 | 2,3 | matching graph image edge 1}  
{matching_graph_image_edges-2 | 3,4,5 | matching graph image edge 2}  
{matching_graph_image_edges-3 | 4,5 | matching graph image edge 3}  
{matching_graph_image_edges-4 | 5 | matching graph image edge 4}
```

图 2.2 配置文件示例

第一行表示中心图像的序号为 2，第二行表示中心图像的旋转角度，第三行记录了本组图片的数量。之后的几行即表示图像的相邻信息，如序号为 0 的图像与 1,2,3 号图像均相邻。

在原 NISwGSP 项目中，图像文件用如附录 A 1 程序读入，虽然图像文件是按字典序命名，但 readdir 函数读入文件并无特定顺序，导致读入图片顺序与配置文件无法对应，直接造成拼接失败。

使用字典序读取函数 scandir 修改此部分程序，使得读入文件顺序正确，程序片段如附录 A 2 所示。

对于读入的图片文件，采取降采样手段使得各幅待拼接图像具有接近的大小，这样既使得拼接过程具有良好一致性，也加快了计算机处理速度。在本设计中，所有待拼接图像降采样至 64 万像素。

## 2.2 图像点特征匹配

图像特征点匹配是计算机视觉中的一个基础任务，在许多任务中都有所涉及。图像特征点匹配主要包括以下三个步骤：

- (1) 特征点提取
- (2) 特征点描述



### (3) 特征点匹配

特征点提取用来从图片中提取出关键点（或特征点、角点）等，关键点信息一般而言会包括：

- (1) 关键点的位置（坐标）
- (2) 关键点的尺度，对应特征的尺度不变性
- (3) 关键点的方向，对应特征的旋转不变性

有了关键点的信息，之后将对关键点进行描述，从而可以根据不同关键点的不同描述来判断关键点之间的匹配关系。

特征点描述是用一组数学向量对特征点进行描述，其主要保证不同的向量和不同的特征点之间是一种一一对应的关系，同时相似的关键点之间的向量的差异尽可能小<sup>[5]</sup>。在进行特征匹配的时候，将通过计算不同特征描述向量之间距离的远近来判断特征点之间是否相似。

特征点匹配其实就是指特征向量之间的距离计算。

SIFT 特征匹配算法于 2004 年提出，而 ORB 算法则提出于 2011 年，虽然 ORB 算法更加新颖，但具体在图像拼接任务中的应用还要结合两种算法的特性而决定。

#### 2.2.1 SIFT 特征点匹配

SIFT (Scale-Invariant Feature Transform) 算法围绕图像特征的尺度不变性展开，相同的局部特征在不同尺寸的图片下其特征的描述向量基本相同。

在特征提取阶段，首先将图像经过一系列的降采样，构造高斯尺度空间。高斯尺度空间是由图像金字塔降采样结合高斯滤波得到的。高斯尺度空间分为多个组（每组图像具有相同分辨率），使用不同的  $\sigma$  对同一组的图像进行高斯模糊将每个组进一步分为不同的层。图 2.3 表示高斯尺度空间的构造。

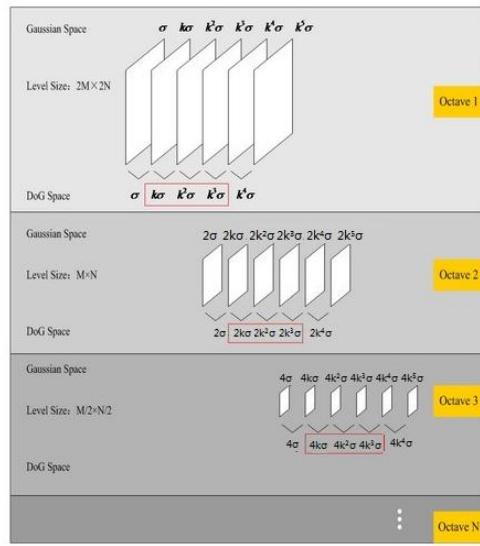


图 2.3 高斯尺度空间

特征点提取的具体步骤如下：

(1) 将同一组中的相邻高斯尺度  $\sigma$  之间的灰度图像进行减法运算，得到高斯差分图像<sup>[4]</sup>。这一步的目的在于获得清晰的边缘特征。

(2) 然后在高斯差分图像上检测特征点。

选择好特征点之后，还需要提取出特征点的尺度信息及方向信息。在特征描述过程中，尺度信息体现在特征点的描述区域范围，是金字塔系数  $\times$  高斯尺度  $\sigma$ 。并以像素周围的灰度变化描述特征点的方向信息。最终 SIFT 的特征点将用 128 维向量表示。之后进行特征匹配，计算特征描述算子之间的欧式距离，如果距离小于阈值，则匹配成功。公式(2.1)至(2.3)中， $x$ ， $y$  分别表示两个 SIFT 特征点的描述向量元素， $des$  表示描述向量，而  $d(des_1, des_2)$  表示向量间的欧氏距离。

$$des_1 = (x_1, \dots, x_{128}) \quad (2.1)$$

$$des_2 = (y_1, \dots, y_{128}) \quad (2.2)$$

$$d(des_1, des_2) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_{128} - y_{128})^2} \quad (2.3)$$

### 2.2.2 ORB 特征点匹配

ORB (Oriented FAST and Rotated BRIEF) 如其名称所述，为 FAST 特征点赋予了方向，且根据特征点的方向，采用改进自 BRIEF 算法的 Rotated BRIEF 算法对特征进行描述。FAST 特征点的核心优势是其提取的快速性，但重大缺陷是其完全不具有旋转不

变性，故进行如 ORB 的改进。

### 2.2.3 SIFT 与 ORB 的对比及本设计采取的方案

ORB 相比 SIFT 主要的优势就是其采用了速度更快的 FAST 特征点以及使用二进制串为特征描述子，使得特征描述的速度也极大提升。ORB 的计算速度是 SIFT 的 100 倍。

但对于本设计的图像拼接任务，图像之间的配准极大依赖于特征匹配对的数量与分布，理想的情况是特征匹配点对密集而均匀地分布在两幅图像的重叠区域。在这样的背景下，SIFT 能在更多区域得到匹配对，尤其是在图像中视差差异较大的不同平面。且对于 ORB 算法更为致命的问题是其完全没有考虑尺度不变性的问题，其旋转不变性效果也不如 SIFT 算法，这在配准、拼接一些不同视角或存在旋转的图像时会有极其不利的影响。图 2.4 中 SIFT 与 ORB 算法的效果对比很好地说明了这些问题，相比 ORB，SIFT 提取到了不同视觉平面（大视差范围，包括前景与背景）的匹配特征点。故 ORB 更适合需要实时性的视频任务而不适用于追求大视差图像的精确拼接。这也说明了为什么 OpenCV 中集成的 Stitcher 图像拼接模块使用 SIFT (SURF) 算法作为默认特征匹配算法。图 2.5 对比了 SIFT 与 ORB 的尺度及旋转不变性，即对比图像间发生旋转及尺度变化时对特征匹配的搜寻能力。

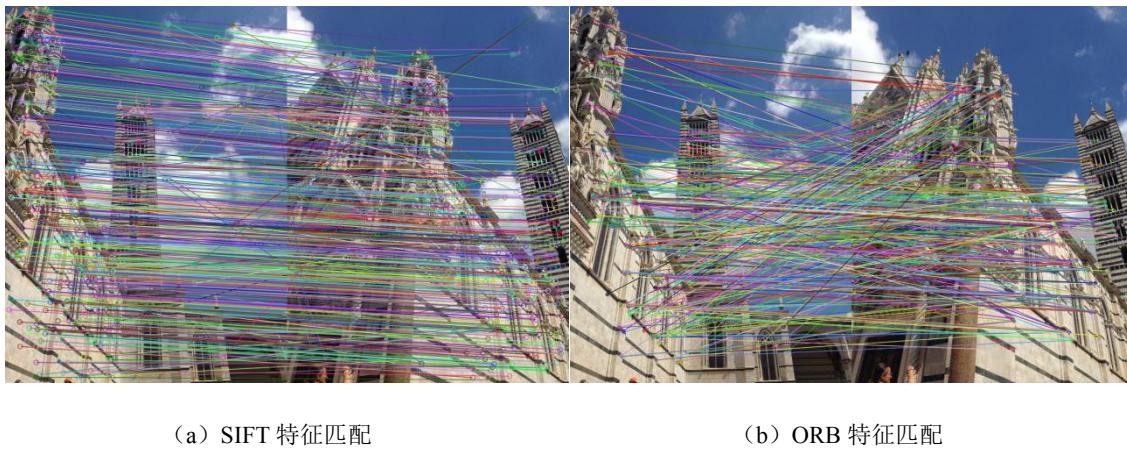


(a) SIFT 特征匹配



(b) ORB 特征匹配

图 2.4 SIFT 与 ORB 对比 (一)



(a) SIFT 特征匹配

(b) ORB 特征匹配

图 2.5 SIFT 与 ORB 对比 (二)

如图 2.5, SIFT 不仅提取到了更多匹配对, 其正确匹配对也显著多于 ORB, 这是由于左右图中场景存在的旋转与尺度变化, 显然 ORB 并不具有尺度不变性。

本设计最终采用 SIFT 算法, 而其实现则使用 VLFeat 开源库。由于之后还要对特征匹配对进行 RANSAC 筛选, 故特征点匹配程序中的 `vl_sift_set_peak_thresh()` 与 `vl_sift_set_edge_thresh()` 设置可将边缘阈值与对比度阈值设置的较小, 使得更多匹配点对得以保留。

### 2.3 图像线特征匹配

近年来许多讨论图像拼接的文章将着眼点放在改善图像特征匹配上面, 意图用更多样化的特征指导图像配准以及之后的网格变形。线特征作为一种相对点特征更高维度的信息在此时被引入。由点至线, 特征匹配的流程与点特征匹配的流程是一致的:

- (1) 线特征提取
- (2) 线特征描述
- (3) 线特征匹配

针对线特征提取在这里采用 LSD (Line Segment Detector) 算法。其流程如图 2.6:

---

**Algorithm 1: LSD: Line Segment Detector**

---

```
1  $I_S \leftarrow \text{ScaleImage}(I, S, \sigma = \frac{\Sigma}{S})$ 
2  $(\text{LLA}, |\nabla I_S|, \text{OrderedListPixels}) \leftarrow \text{Gradient}(I_S)$ 
3 Status  $\leftarrow \begin{cases} \text{USED}, & \text{pixels with } |\nabla I_S| \leq \rho \\ \text{NOT USED}, & \text{otherwise} \end{cases}$ 
4 foreach pixel  $P \in \text{OrderedListPixels}$  do
5   if Status( $P$ ) = NOT USED then
6     region  $\leftarrow \text{RegionGrow}(P, \tau)$ 
7     rect  $\leftarrow \text{Rectangle}(region)$ 
8     while AlignedPixelDensity(rect,  $\tau) < D$  do
9       region  $\leftarrow \text{CutRegion}(region)$ 
10      rect  $\leftarrow \text{Rectangle}(region)$ 
11    end
12    rect  $\leftarrow \text{ImproveRectangle}(rect)$ 
13    nfa  $\leftarrow \text{NFA}(rect)$ 
14    if nfa  $\leq \varepsilon$  then
15      Add rect  $\rightarrow out$ 
16    end
17  end
18 end
```

---

图 2.6 LSD 算法流程

提取到的线段以 OpenCV 中的 KeyLine 对象储存，该类中的类方法可以很方便地获取检测到的线段的信息（起、始点坐标，线段长度，线段旋转角度等）。

下一步就是对提取到的线特征进行描述，本设计采用 LBD（Line Band Descriptor）描述子。LBD 算法本身也具备类似 SIFT 算法那样从尺度空间中提取线特征的部分（采用 EDLines 算法，如图 2.7），但这一部分已经使用效果更好的 LSD 算法代替。

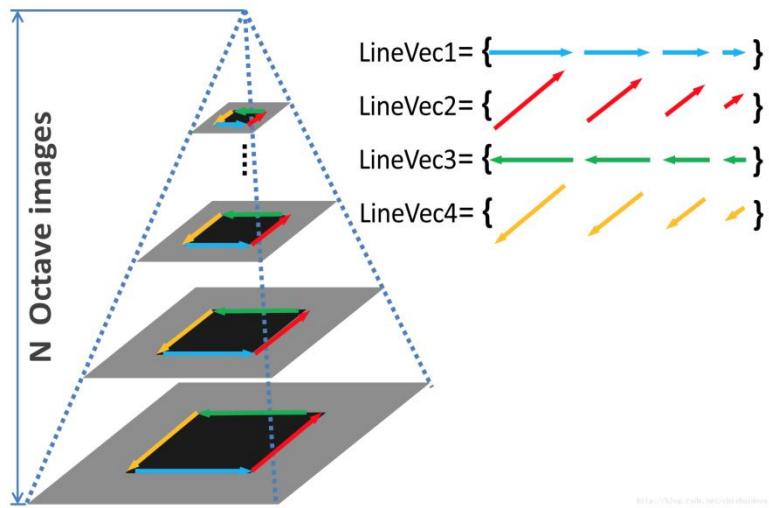


图 2.7 LBD 算法中的线特征提取——EDLines 算法在尺度空间提取线段

LBD 描述符将线段周围局部区域划分为条带(Band)，确定线段条带和方向，通过一系列权重计算构成条带描述符矩阵，并通过均值与均方差计算最终构成描述子。图 2.8 显示了不同线特征匹配方法的效果对比。



(a) 原始图像

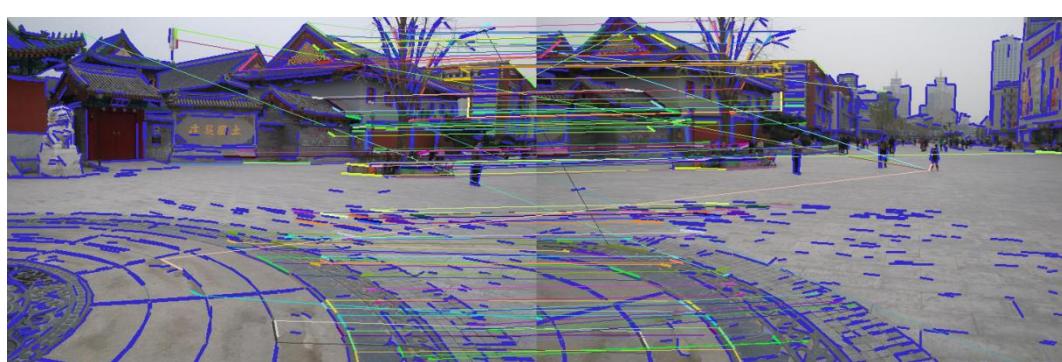


(b) EDLines

(c) LSD



(d) 原始 LBD 提取的线特征



(e) LSD 提取+LBD 描述

图 2.8 线特征匹配效果



由图 2.8 不难看出, LSD 算法可以提取到更多的线特征, 这也进一步保障了特征描述后的匹配对数量。

LSD 算法与 LBD 均在 `opencv_contrib` 包中进行了封装, 这一部分程序在整合时将 OpenCV KeyLine 类型中的信息进行转换以存入自定义的类型中, 方便后续环节的调用。部分转换程序如附录 B 1。

## 2.4 多平面 RANSAC 筛选特征匹配对

同一特征在两幅图像中处于不同的空间平面, 前述步骤提取到的点、线特征匹配对主要用于估计两幅图像间不同空间平面的透视关系, 坐标点的转换关系在齐次坐标系中可以由单应性矩阵表示:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2.4)$$

公式(2.4)中,  $(x', y', 1)$  表示目标图像中点的齐次坐标,  $(x, y, 1)$  表示参考图像中与之配对的点的齐次坐标。由齐次坐标的特性可知, 单应性矩阵实际具有 8 个自由度, 至少需要 4 对匹配点进行估计, 当匹配对多于 4 个时, 需要对匹配对进行筛选, 剔除噪点及不属于某一所需透视关系模型的外点。

考虑使用 RANSAC(RANdom SAmple Consensus)方法来筛选之前得到的点特征匹配对与线特征匹配对。经典 RANSAC 从一个较大的数据空间中随机选取数据估计模型, 并使用未被选中的数据为该模型评分。反复进行这一过程, 评分较低的模型被舍弃而评分最高的模型被当做正确估计保留。在本设计中这个模型就指代将一个视觉平面投影到其相邻图像中视觉平面的单应性矩阵, 一个单应性矩阵可以由相互独立的至少四对匹配特征值估计得出。故在本设计中, 用以产生模型估计的数据集就指代所有特征匹配对的集合。

在经典的 RANSAC 流程中, 目标函数  $C$  可以被看作: 某次迭代过程中, 符合当前模型的数据集  $\Omega$  中类内点的个数, 而 RANSAC 就是最大化  $C$  的过程。在置信度为  $\eta_0$  的条件下, 在循环过程中, 至少有一次采样, 使得采样出的  $m$  个点均为类内点, 这样才能保证在循环的过程中, 至少有一次采样能取得目标函数的最大值。因此, 采样次数  $k$

应该满足以下条件:

$$k \geq \frac{\log(1-\eta_0)}{\log(1-\varepsilon^m)} \quad (2.5)$$

这里除了置信度  $\eta_0$  外,  $m$  为子集大小,  $\varepsilon$  为类内点在  $\Omega$  中的比例, 其中置信度一般设置为[0.95, 0.99]的范围内。 $\varepsilon$  可以取初始状态下最坏条件下的比例, 然后随着迭代次数, 不断更新为当前最大的类内点比例。图 2.9 为经典 RANSAC 算法的伪代码, 图 2.10 说明了 RANSAC 相比最小二乘法在模型估计上的优势。

**Algorithm 1.** Standard RANSAC algorithm

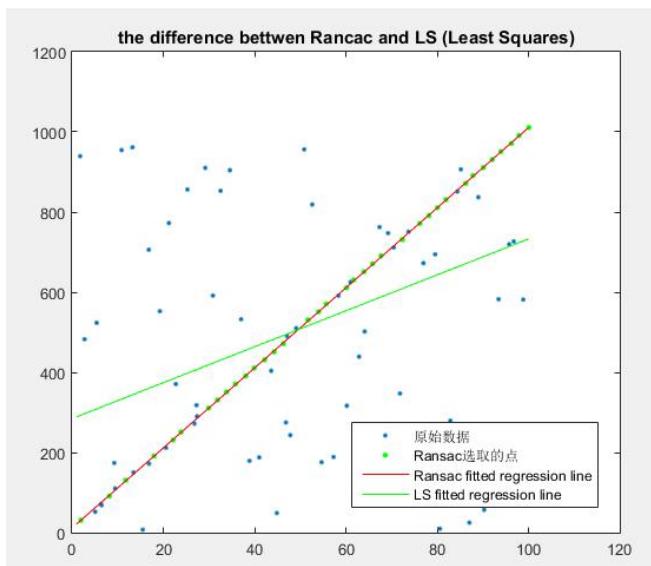
```

Input:  $\mathcal{U}, \eta_0, k_{max}, t$ 
Output:  $\theta^*, \mathcal{I}^*$ 
 $k = 0, I_{max} = 0$ 
while  $k < k_{max}$  do
    1. Hypothesis generation
    Randomly sample minimal subset of  $m$  points
    Estimate model parameters  $\theta_k$ 
    2. Verification
    Calculate the support set  $\mathcal{I}_k$ 
    if  $|\mathcal{I}_k| > I_{max}$  then
         $\theta^* = \theta_k, \mathcal{I}^* = \mathcal{I}_k$ 
        Recompute  $k_{max}$  from eqn. (3) using  $\varepsilon = |\mathcal{I}^*|/N$ 
    end if
     $k = k + 1$ 
end while

```

<http://blog.csdn.net/>

图 2.9 经典 RANSAC 算法流程



图中数据包含内点、外点以及噪点, 红色直线是使用 RANSAC 拟合出的直线, 而绿色直线则是使用最小二乘法拟合出的直线。最小二乘法更易受到噪点影响。

图 2.10 RANSAC 与最小二乘法的比较

由于本设计采用的是基于局部单应性矩阵的拼接算法，而经典 RANSAC 只能估计出一个全局单应性矩阵而将不属于该变换的特征匹配对全部舍弃，故本设计应该采用多模型 RANSAC，从多个单应性矩阵模型中估计内点并保留。

本设计采用的多平面 RANSAC 为 Sequential RANSAC，图 2.11 是其伪代码流程。

```
1. Models = {}  
2. While(!终止条件)  
    BestCS = RANSAC(Data, M, ε);  
    If SatisfactoryModel(BestCS) then  
        Data = Data - BestCS  
        Models = Models ∪ {BestCS}  
3. Return Models
```

图 2.11 Sequential RANSAC 流程

Sequential RANSAC 的本质是重复进行多次 RANSAC，每一次都将前一次 RANSAC 选取的内点剔除再进行下一次 RANSAC，当产生的模型被拒绝时终止整个算法。图 2.12 表示该算法估计多个模型的效果。

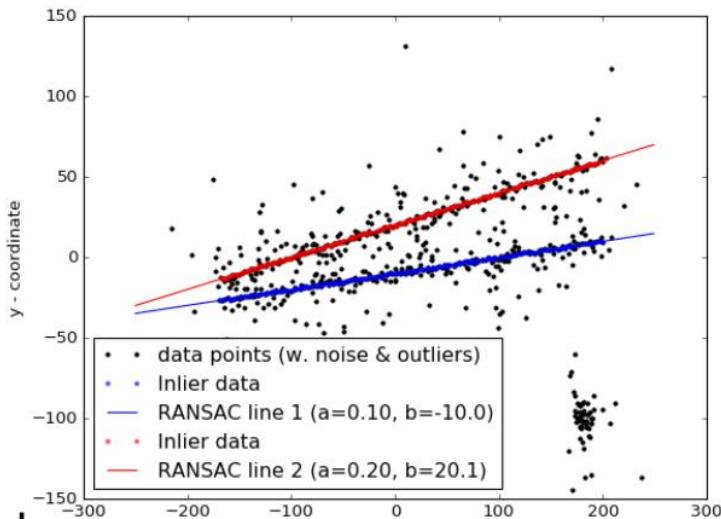


图 2.12 Sequential RANSAC 进行多条直线的拟合

由于线特征匹配对不能直接用来计算单应性矩阵，对于线特征匹配对的 RANSAC，本设计采用线段起始点与终止点估计单应性矩阵，且由于同一组图像中线特征匹配对的数量远少于点特征匹配对，故其在筛选模型时的阈值需要做出一定调整，其他流程与点特征匹配的 RANSAC 相似。

附录 B 2 是结合 OpenCV 中 findHomography() 函数的实现的 Sequential RANSAC 程序片段，local\_homography\_min\_features\_count 是一个模型最少需要包含的内点数量阈值，若某次估计出的模型所包含的内点数量小于这一阈值，则该模型被拒绝，整个算法

终止。

图 2.13 是对点特征匹配对的多平面 RANSAC 筛选结果，并与传统 RANSAC 算法的效果进行对比。



(a) 未经筛选的 SIFT 点特征匹配对, 345 对



(b) 经典 RANSAC 筛选后的匹配对, 164 对



(c) 多平面 RANSAC 筛选后的匹配对, 273 对

图 2.13 多平面 RANSAC (Sequential RANSAC) 与经典 RANSAC 效果对比 (点特征)

由图 2.13 可以看出多平面 RANSAC 在保证匹配对正确性的前提下保留了更多的匹配对，尤其是包含了不同视差平面（前景，背景）的匹配对。

图 2.14 是对线特征匹配对的多平面 RANSAC 筛选结果，并与传统 RANSAC 算法的效果进行对比。

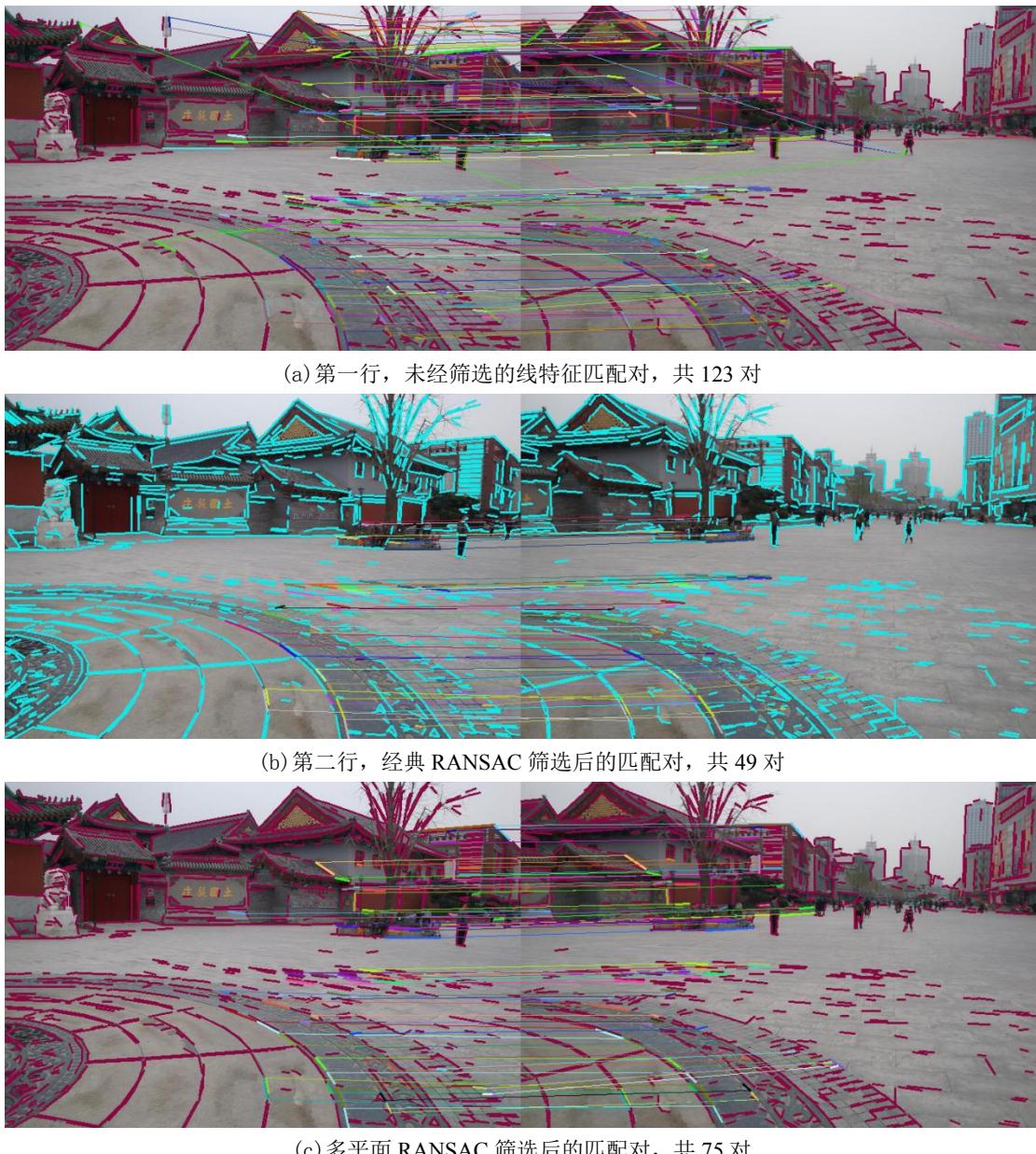


图 2.14 多平面 RANSAC(Sequential RANSAC)与经典 RANSAC 效果对比 (线特征)

从图 2.14 可以看出, 线特征筛选结论与筛选点特征匹配对类似, 多平面 RANSAC 在对误匹配的剔除及对正确匹配对的保留都显著优于经典 RANSAC, 线特征匹配时的匹配错误率相比点特征更高, 且由于其后续进行配准时的权重较大, 对局部区域的对齐有更强指导作用, 其合理筛选显得尤为重要。对点、线特征匹配对的多平面 RANSAC 程序被整合入图片拼接项目程序整体中。

图 2.15 是本节方法在纹理重复区域的运行结果, 其中仍然包含错误匹配, 体现了本筛选方法对于纹理高度重复的特征匹配对效果有待提升, 错误的匹配对可能直接造成拼接失败。



图 2.15 纹理重复区域存在误匹配对，造成拼接失败

## 2.5 本章小结

本章首先介绍了图像读取的接口设计。之后对于拼接算法中的基础环节特征提取与匹配进行了详细的介绍。结合拼接算法对于特征的使用特点，SIFT 与 ORB 算法的优劣被说明并阐述了最终所用方法的适用性。线特征的提取与匹配也在不同方法之间进行了充分的比对，最终选取的方法可以保质保量地实现线特征匹配对的提取。由于图像拼接算法对特征匹配的要求十分高，一种基于多平面 RANSAC 的筛选方法被引入，以承接原始匹配对提取，对点、线特征匹配对做精细化的筛选。本章给出了筛选的结果说明，可见效果显著。



### 第三章 拼接图像配准与网格优化变形

本章按照图像拼接算法整体处理流程，顺接上一章内容，介绍特征提取后如何利用特征匹配进行图像配准。并在 3.2 节详细介绍了本设计所提出的拼接算法核心部分——网格优化变形。对于网格优化变形中的各约束项，进行了准确的原理表述与效果说明。

#### 3.1 图像配准 LAPAP

相邻图像间的对齐或配准是保障重叠区域最终拼接融合效果的基础，良好的配准可以使得最终拼接全景图中的重叠区域无明显重影或透视变形，其思想是利用提取的特征匹配对获取两幅图重叠区域之间准确的透视投影关系，即以单应性矩阵表示。令  $\mathbf{J}$  表示相邻图像对的集合，对于  $\mathbf{J}$  中的一对相邻图像  $I_i$  和  $I_j$ ，采用结合线特征的 LAPAP (Line As-Projective-As-Possible) 算法进行配准<sup>[31]</sup>，而使用第二章中得到的点、线特征匹配对作为指导。LAPAP 在经典 APAP 的基础上加入了线特征匹配对，是一种基于网格的局部单应性配准方法，在处理过程中每幅图片都被划分为均匀的网格。点线特征匹配对一般都密集存在于纹理复杂的区域，而弱纹理区域就缺少配准信息，基于网格的配准方法则是使覆盖整幅图像的网格均匀地得到配准信息，故具有更好的配准效果。首先， $I_i$  中  $I_i$  与  $I_j$  重叠部分的网格顶点构成配对点集合  $\mathbf{M}^{ij}$ 。对于  $\mathbf{M}^{ij}$  中的每个网格顶点，可以通过 LAPAP 算法获得  $I_j$  中的对应点。同理，可以获得配对点集合  $\mathbf{M}^{ji}$ 。下面叙述 LAPAP 的具体处理原理。

$l = [a, b, c]^T$ ,  $l' = [a', b', c']^T$  代表目标图像  $I_j$  与参考图像  $I_i$  中的一组线特征匹配对。令  $p_j^{0,1} = [x^{0,1}, y^{0,1}, 1]^T$  代表参考图像  $I_i$  中线段  $l$  的两个端点，而  $p_j'^{0,1} = [x'^{0,1}, y'^{0,1}, 1]^T$  则是目标图像  $I_j$  中线段  $l'$  的两端点，它们满足  $p_j'^{0,1T} H p_j^{0,1} = 0$ ，这代表线段  $l$  的端点经过变换  $H$  应该落在  $l'$  的两端点处。因此，变换矩阵  $H$  可以用一系列线特征匹配对通过最小化几何距离  $\sum_j \|p_j'^{0,1T} \times H p_j^{0,1}\|^2$  估计得到。

然后给出结合线特征与点特征的单应性矩阵联合估计公式:

$$\begin{aligned}\hat{h} &= \arg \min_h (\sum_i \left\| p_i' \times H p_i \right\|^2 + \sum_j \left\| p_j'^{0,1T} \times H p_j^{0,1} \right\|^2) \\ &= \arg \min_h (\sum_i \|A_i h\|^2 + \sum_j \|B_j h\|^2), s.t. \|h\|=1\end{aligned}\quad (3.1)$$

其中  $\hat{h}$  为估计的单应性矩阵,  $p_i$  与  $p_i'$  为点特征匹配点对。类似于 APAP, 局部单应性矩阵相比全局单应性矩阵有更多自由度, 故下面继续扩展至结合线特征匹配对的局部单应性矩阵估计。位于  $p_* = [x_*, y_*]$  的第  $k$  个网格的局部单应性矩阵  $h_k$  由公式(3.2)进行估计:

$$h_k = \arg \min_h \|W_k C h\|^2, s.t. \|h\|=1 \quad (3.2)$$

矩阵  $C$  是将全体点特征系数矩阵  $A_i$  与全体线特征系数矩阵  $B_j$  垂直叠加构成的统一矩阵, 即  $C = [A; B]$ 。 $W_k$  是所有点特征权重系数  $w^p$  与所有线特征权重系数  $w^l$  构成的权重向量。

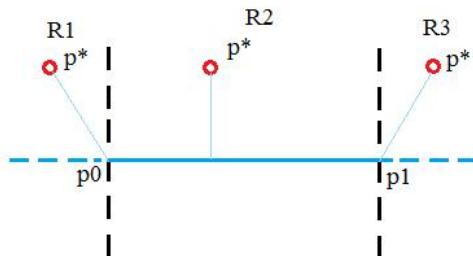
点特征匹配对的权重计算与 APAP 算法一致, 而线特征的权重系数计算如(3.3):

$$w^{l_j} = \max(\exp(-d_l(p_*, l_j)^2 / \sigma^2), \eta) \quad (3.3)$$

其中  $\sigma$  是一个常值比例因子,  $d_l(p_*, l_j)$  代表网格中心  $p_*$  到线段  $l_j$  的最短距离, 其计算方式如(3.4):

$$d_l(p_*, l_j) = \begin{cases} \min(\|p_* - p_j^0\|, \|p_* - p_j^1\|) \\ |a_j x_* + b_j y_* + c_j| / \sqrt{a_j^2 + b_j^2} \end{cases} \quad (3.4)$$

该距离的图形表示图 3.1 所示:



$p_0, p_1$  为线段两端点, 网格中心点  $p^*$  与线段相对位置在不同区域时, 其距离计算方式为到线段两端点的最短距离或与线段的垂直距离。

图 3.1 网格中心  $p_*$  到线段  $l_j$  的距离示意

显然，离局部网格较近的特征匹配对在估计该局部网格的局部单应性矩阵时被赋予了更大的权重。综上，解线性方程组以获得单应性矩阵  $\mathbf{h}$ ，即为 WC 的最小有效右奇异向量（smallest significant right singular vector）。

这样大体框架思想与 APAP 一致而加入了线特征匹配对的配准方式就是 LAPAP，此环节后，第二章得到的特征匹配对被转化为网格配准关系。基于网格的 LAPAP 配准如图 3.2 所示。

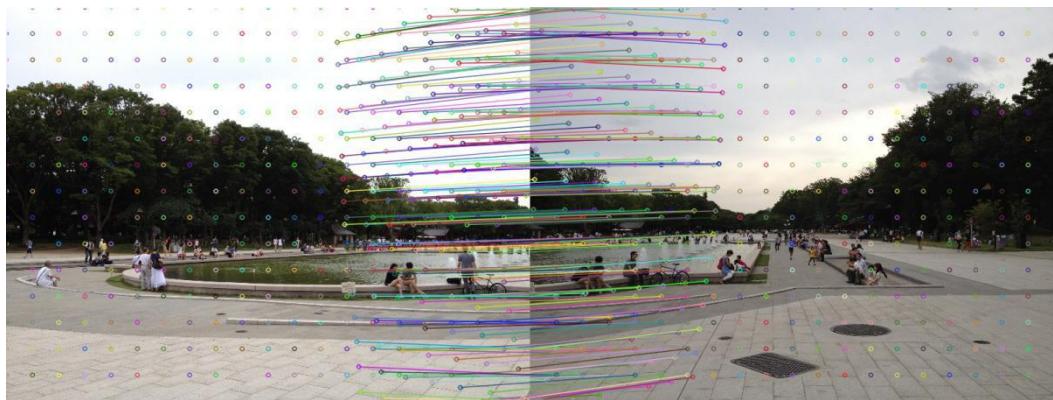


图 3.2 基于网格的 LAPAP 配准

重叠区域的网格点按照前述计算的单应性矩阵进行变换，获得其在相邻图像中的配准点（非在网格处），这样的每一对新匹配对不仅均匀分布在重叠区域，且包含的配准信息综合考虑了整幅图像的点、线特征匹配，使其配准更加精确。

与 APAP 相比，在计算局部单应性矩阵时由于线特征的加入，网格配对点获得了相应的更加丰富的配准信息，在线特征丰富的建筑物或其他人造场景对网格配准有显著的提升效果。如图 3.3 所示，本样例本身点特征较丰富，配准效果优良，而线特征的加入使得整体配准精度提升，在一些线特征丰富的区域附近尤其效果显著。



(a) 采用 APAP 的配准效果



(b) 采用 LAPAP 的配准效果

图 3.3 APAP 与 LAPAP 配准效果对比

### 3.2 网格优化变形

完成图像配准后，理论上已经可以按照得到的一系列局部单应性矩阵将待拼接图像进行透视变形并投影至一个画布上进行拼接。但图像配准过程仅考虑到了重叠区域的对齐，按照这一结果进行网格变形很可能造成拼接结果中只注重追求配准效果而过渡扭曲图形结构，也即观感自然度较差。引入网格优化变形可以从配准、相似性及图形结构（如：直线）保留等多方面因素约束图像变形，使得最终的拼接结果具有更好的观感自然度，并在拼接全景图中保留如直线这样的高维结构供后续分析使用。这里的观感自然度主要是以拼接前的各待拼接子图为基准，即认为各原始子图中的物体形状较为自然。

所谓网格优化变形，与上一节网格配准一样采用了覆盖整张图像的网格，并以网格为基础进行图像变形（Image Deformation）。使  $V_i$  与  $E_i$  分别代表图像  $I_i$  网格的顶点集与边集， $V$  代表全部图像中的网格构成的网格顶点集。网格优化变形算法的目的是找到一个变形后的网格顶点集  $\tilde{V}$ ，使得能量函数  $\Psi(\tilde{V})$  最小。本设计的能量函数包含 5 项约束项，分别是保持图像配准的区域配准项  $\Psi_a(V)$ 、保留局部透视关系的局部相似项  $\Psi_l(V)$ 、约束每幅待拼接子图变形保持整体性的全局相似项  $\Psi_g(V)$ 、加强直线（线段）配准效果的线性对齐项  $\Psi_m(V)$  以及保留整幅图像全部直线特征的线性保持项  $\Psi_c(V)$ 。其中线性对齐项与线性保持项共同作用，用以保留待拼接图像中的直线特征，减少线性结构扭曲，使得拼接全景图中的一些高维特征结构（如：建筑物轮廓，交通道路等）得到更真实的体现。图 3.4 展现了不具备该两项约束的 NISwGSP 算法效果。

最终整个网格优化问题由一个  $AX=b$  的稀疏矩阵线性方程表示，稀疏矩阵  $A$  中填充各约束项转化为最小二乘问题后的系数计算值乘以该约束项对应权重  $\lambda$ 。

$$\tilde{V} = \arg \min_{\tilde{V}} (\lambda_a \Psi_a(V) + \lambda_l \Psi_l(V) + \Psi_g(V) + \lambda_m \Psi_m(V) + \lambda_c \Psi_c(V)) \quad (3.5)$$

公式(3.5)为能量函数表达式，其中权重  $\lambda$  用来调节该约束项在整体能量函数中的占比，如配准项权重高代表图像在网格变形时更多考虑配准的影响。权重的设置依据算法运行的结果及对配准、相似性、直线保持的要求人工选择，在不同场景中权重的设置也会有所区别。在权重调节过程中，配准项权重较为灵活，而相似项权重比较敏感，线性相关的两项权重不宜过大，在 3.2.5 中有所介绍。



NISwGSP 拼接结果，虽然其结果观感自然度较好，但仍存在一些不必要的变形，比如建筑边缘线结构的扭曲。使用直线特征相关的约束项可以减轻这一问题。

图 3.4 NISwGSP 拼接效果中的高维特征破坏

### 3.2.1 区域配准项

区域配准项通过约束图像配准环节得到的配对点保持对准，使得图像配准关系在网格变形中得到保留。其形式可以表示为：

$$\Psi_a(V) = \sum_{i=1}^N \sum_{(i,j) \in \mathbf{J}} \sum_{p_k^{ij} \in \mathbf{M}^{ij}} \left\| \tilde{v}(p_k^{ij}) - \tilde{v}(\Phi(p_k^{ij})) \right\|^2 \quad (3.6)$$

该式表示区域配准项通过最小化配对点之间的坐标差来达到配准的目的， $\Phi(p)$  返回的是配对点  $p$  的匹配点， $\tilde{v}(p)$  表示点  $p$  的位置由其所在网格的四个网格顶点位置的线性组合表示，也即  $\sum_{i=1}^4 \alpha_i \tilde{v}_i$ ， $\tilde{v}_i$  表示四个网格顶点的坐标， $\alpha_i$  则是双线性插值系数。因此，(3.6)式可转化为线性方程组的形式：

$$\Psi_a(V) = \sum_{i=1}^N \sum_{(i,j) \in \mathbf{J}} \sum_{p_k^{ij} \in \mathbf{M}^{ij}} \| A_a V' - b_r \|^2 \quad (3.7)$$

其中  $V'$  为待求解的经网格优化后的网格顶点坐标(x,y)纵向叠加形成的向量,  $A_a$  是双线性插值系数构成的系数矩阵:

$$\left\{ \begin{array}{l} A_a = \begin{bmatrix} \alpha_1 & 0 & \dots & \alpha_4 & 0 & \dots & -\alpha'_1 & 0 & \dots & -\alpha'_4 & 0 \\ 0 & \alpha_1 & \dots & 0 & \alpha_4 & \dots & 0 & -\alpha'_1 & \dots & 0 & -\alpha'_4 \\ \vdots & \vdots \end{bmatrix} \\ b_a = 0 \end{array} \right. \quad (3.8)$$

配准是驱动重叠区域待拼接图像变形的主要因素, 在这一区域, 网格倾向于形成透视变换以使得处于不同视觉平面上的相同物体在拼接结果中处于同一平面。在前一环节 LAPAP 中, 由特征匹配得到的匹配信息已经被合理分配于各网格顶点上, 同一网格内的所有像素根据所在网格的四个网格顶点的配准信息而进行对齐。显然, 网格划分的越小, 重叠区域在进行对齐时的精确度越高, 当网格较大且其中存在多个视觉平面的物体时, 这些处于不同空间平面的物体也就无法简单按照四个网格顶点进行的配准而对齐。为了说明区域配准项对于重叠区域拼接效果的实际影响, 在能量函数中减小配准项权重以观察效果, 如图 3.5 所示。



(a) 减小区域配准项在能量函数中权重占比



(b) 区域配准项正常

图 3.5 区域配准项作用说明 (左侧为全景效果, 右侧为局部放大)

图 3.5 (a) 中将区域配准项在能量函数中的权重减小 100 倍, 可以看见图中的树叶部分相比使用配准项出现了大面积的重影, 这说明相邻图像中的同一物体在网格变形时

并没有被投影至空间的同一视觉平面，也即在二维全景图上的像素没有对齐。当图像融合中同时采集两幅图像重叠区域的像素时，重影即产生。

### 3.2.2 局部相似项

局部图像往往包含一些辨识度较高的目标，如行人，标识等，图像在局部区域变形过大或不一致会导致局部目标和特征的严重扭曲，使得拼接全景图中的一些所需的形状失真严重，即如图 3.6 所示。

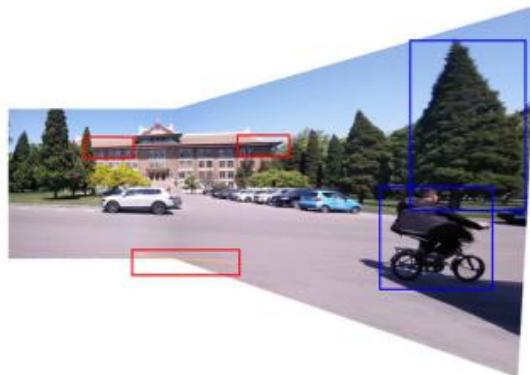


图 3.6 非重叠区域局部变形不一致导致形状严重失真

局部相似项通过局部相似变换的方式最小化局部变形，使得网格尽量进行相同的相似变换 (Similarity Transform)。由于待拼接图像的非重叠区域没有配准信息的约束，局部相似项也可以使得重叠区域较为复杂的网格变形合理过渡到非重叠区域的相似变换。局部相似项可以由式(3.9)表示：

$$\Psi_l(V) = \sum_{i=1}^N \sum_{(j,k) \in E_i} \|(\tilde{v}_k^i - \tilde{v}_j^i) - S_{jk}^i (v_k^i - v_j^i)\|^2 \quad (3.9)$$

其中， $v_k^i$  代表原始网格顶点， $\tilde{v}_k^i$  代表变形后的网格顶点， $S_{jk}^i$  是顶点  $v_j^i$  与  $v_k^i$  构成的边  $e_{jk}$  的相似变换：

$$S_{jk}^i = \begin{bmatrix} c_j & s_j \\ -s_j & c_j \end{bmatrix} = \arg \min_{S_{jk}^i} \sum_{v \in N(e_{jk})} \|S_{jk}^i v - \tilde{v}\|^2 \quad (3.10)$$

$N(e_{jk})$  表示边  $e_{jk}$  附近的四个网格顶点构成的集合，即  $N(e_{jk}) = \{v_j, v_k, v_l, v_r\}$ ，图 3.7 展示的是网格与边  $e_{jk}$  的关系。

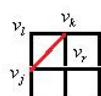


图 3.7 边附近的网格顶点示意图



进一步将相似变换矩阵(3.10)展开:

$$\{c_j, s_j\} = \arg \min_{c_j, s_j} \sum_{v \in N(e_{jk})} \left\| \begin{bmatrix} c_j & s_j \\ -s_j & c_j \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} - \begin{bmatrix} \tilde{v}_x \\ \tilde{v}_y \end{bmatrix} \right\|^2 \quad (3.11)$$

$$\begin{aligned} &= \arg \min_{c_j, s_j} \left\| \begin{bmatrix} v_{jx} & v_{jy} \\ v_{jy} & -v_{jx} \\ v_{kx} & v_{ky} \\ v_{ky} & -v_{kx} \\ v_{lx} & v_{ly} \\ v_{ly} & -v_{lx} \\ v_{rx} & v_{ry} \\ v_{ry} & -v_{rx} \end{bmatrix} \begin{bmatrix} c_k \\ s_k \end{bmatrix} - \begin{bmatrix} \tilde{v}_{jx} \\ \tilde{v}_{jy} \\ \tilde{v}_{kx} \\ \tilde{v}_{ky} \\ \tilde{v}_{lx} \\ \tilde{v}_{ly} \\ \tilde{v}_{rx} \\ \tilde{v}_{ky} \end{bmatrix} \right\| \\ &= \arg \min_{c_j, s_j} \left\| G_k \begin{bmatrix} c_k \\ s_k \end{bmatrix} - \tilde{V} \right\| \end{aligned}$$

对以上线性方程组进行求解即可得到  $\begin{bmatrix} c_j \\ s_j \end{bmatrix} = M_j \tilde{V}$ ，将其代入局部相似项(3.9)可得:

$$\begin{aligned} \Psi_l(V) &= \sum_{i=1}^N \sum_{(j,k) \in E_i} \left\| (\tilde{v}_k^i - \tilde{v}_j^i) - S_{jk}^i (v_k^i - v_j^i) \right\|^2 \quad (3.12) \\ &= \sum_{i=1}^N \sum_{(j,k) \in E_i} \left\| (\tilde{v}_k^i - \tilde{v}_j^i) - \begin{bmatrix} x_{jk} & y_{jk} \\ y_{jk} & -x_{jk} \end{bmatrix} \begin{bmatrix} c_j \\ s_j \end{bmatrix} \right\|^2 \\ &= \sum_{i=1}^N \sum_{(j,k) \in E_i} \left\| (\tilde{v}_k^i - \tilde{v}_j^i) - \begin{bmatrix} x_{jk} & y_{jk} \\ y_{jk} & -x_{jk} \end{bmatrix} M_j \tilde{V} \right\|^2 \\ &= \sum_{i=1}^N \sum_{(j,k) \in E_i} \| Q \tilde{V} - b_l \|^2 \end{aligned}$$

$b_l$  与在区域配准项中一样是零向量。

由以上推导过程不难发现,局部相似项利用相似变换矩阵将同一个网格的四个顶点在变形过程中关联了起来,使局部网格在变形时保持了一致性。

同样,在图 3.8 中,为了说明局部相似项的作用,将局部相似项在能量函数中删除,以观察其带来的影响。



(a) 取消局部相似项



(b) 使用局部相似项

图 3.8 局部相似项效果说明 (左侧为全景效果, 右侧为局部放大)

如图 3.8, 不使用局部相似项的拼接效果虽然在整体上没有大的观感不足, 即每张拼接子图的变形保持了一定的一致性 (这主要受益于 3.2.3 介绍的约束项), 但放大观察局部区域, 一些前景物体发生了局部扭曲 (如红色框所选中区域)。思考这一约束项的原理可知, 取消局部相似项, 每一个网格顶点进行的相似变换只与其自身有关, 而加入该项, 一个网格顶点的相似变换矩阵由其相邻顶点共同估计而来。由于相邻网格间共用一条边即具有两个公共顶点, 一个局部网格的相似变换矩阵在局部地区的几个网格间就具有一定程度的一致性, 也就保证了局部地区变形的相似性。而由于相邻网格估计相似变换矩阵的点只有一半重复, 这种一致性会随着区域的变化逐渐减少, 也就保证了相似变换的合理过渡, 不会使得某一前景物体的一部分突然发生与周边不和谐的变形。

### 3.2.3 全局相似项

全局相似项使得每幅待拼接子图尽量进行相同的相似变换, 如果一幅待拼接图像缺少全局相似的约束, 那么即使它在局部区域表现的变形较为一致, 在范围较大的区域之间却有可能出现整体性的扭曲, AANAP 算法即具有这一问题。因此全局约束项在更大范围内约束了变形的一致性。其表达式如下所示:

$$\Psi_g(V) = \sum_{i=1}^N \sum_{e_j^i \in E_i} w(e_j^i)^2 [(c(e_j^i) - s_i \cos \theta_i)^2 + (s(e_j^i) - s_i \sin \theta_i)^2] \quad (3.13)$$

其中  $s_i$  与  $\theta_i$  分别表示待拼接图像  $I_i$  的尺度与旋转角度, 在 3.2.6 中介绍其获取方法。

而  $c(e_j^i)$  与  $s(e_j^i)$  来自于 3.2.2 中局部相似项利用的边  $e_j^i$  的相似变换矩阵。由于相似变换即为旋转与尺度缩放的组合, 该表达式的思想就是将每个局部网格的相似性变形与整幅待拼接子图  $I_i$  的整体相似变换向关联, 也即约束所有局部变换与整体变换保持一致。权重  $w(e_j^i)$  的表达式如(3.14):

$$w(e_j^i) = \beta + \frac{\gamma}{|Q(e_j^i)|} \sum_{q_k \in Q(e_j^i)} \frac{d(q_k, M^i)}{\sqrt{R_i^2 + C_i^2}} \quad (3.14)$$

$\beta$  与  $\gamma$  是人为设置的权重, 决定了全局相似项在所有约束项中的重要性程度占比,  $Q(e_j^i)$  是边  $e_j^i$  所在网格的集合 (边  $e_j^i$  可能同时属于两个网格),  $M^i$  代表两幅待拼接子图间重叠区域所有的网格集合。 $d(q_k, M^i)$  返回网格  $q_k$  距重叠区域所有网格的相对距离, 其分母  $R_i$  与  $C_i$  是该幅待拼接图像的网格行数与列数。不难看出当一个局部网格离重叠区域较远时, 权重  $w(e_j^i)$  较大, 该局部网格就主要受全局相似的约束。而处于重叠区域的局部网格, 权重  $w(e_j^i)$  较小, 其主要就受全局配准项的约束, 其变形主要考虑与相邻图像的对齐。这样的设置体现了优良图像拼接算法的一个重要思想, 即重叠区域要尽量追求配准, 变形幅度可以较大, 而非重叠区域由于没有配准信息指导, 应逐渐过渡为整体性的相似变换, 拼接结果也就不会相对原始待拼接子图有较大变形失真。图 3.9 中绘出实际网格变形, 较好地显示了这一设计思想。

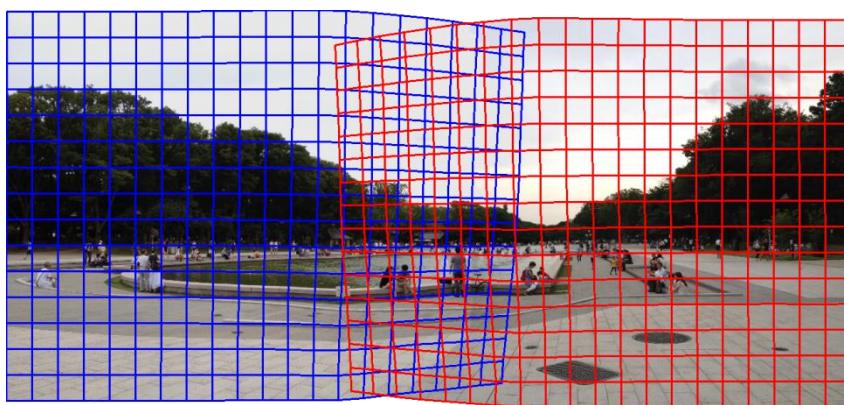


图 3.9 重叠区域受配准约束而非重叠区域过渡至相似变换

为了说明全局相似项在网格优化中的作用, 调整  $\beta$  与  $\gamma$  的值并观察影响, 如图 3.10。



图 3.10  $\beta$  与  $\gamma$  的值分别缩小 10 倍后的拼接效果

显然，没有全局相似的约束对整体形变的影响是致命的，图像中的每个局部区域进行的相似变换虽然考虑到了配准及局部一致的要求，但大范围的区域之间的比例完全不符合自然。也即是说，图像中不同的物体虽然将自己本身的形状结构保持的较为理想，但其完全没有考虑到场景中其他物体与自己的尺寸、旋转之比。使用估计出的整幅图的尺度与旋转作为基准的意义，在于告诉局部物体变换时要有所参照。这与 3.2.2 中分析一致，仅依赖局部相似项则局部相似变换的一致性随着某一小区域网格的向外延伸而逐渐削弱，为了保持整幅图中全部物体的形变在一个合理的范围内，全局相似项对每个网格的相似变换矩阵给予同一约束，是一个强有力的基准调节。这样的调节也是一把双刃剑，如同该项的构建方法所示，不合理的待拼接子图尺度与旋转估计  $s_i$  与  $\theta_i$  会使得整个子图的相似变换参照错误的基准，造成更严重的全局扭曲。

### 3.2.4 线性对齐项

区域对准项利用配对点对 (matching point pairs) 约束重叠区域的图像配准，如本节开头分析所述，大部分拼接算法并没有考虑到对高维特征变形的约束。在 LAPAP 配准的基础上，为了进一步在网格约束变形中对齐重叠区域的直线特征，本设计引入线性对齐项对 2.3 中提取到的重叠区域直线匹配对进行约束，这样做既能够提升重叠区域的配准效果，又可以减少重叠区域的线性结构弯曲，对于保持具有直线轮廓的特定前景目标的观感自然度有重要辅助作用。其表达式如(3.15):

$$\Psi_m(V) = \sum_{i=1}^N \sum_{(i,j) \in J} \sum_{l \in L^i} \sum_{p_k^l \in P^l} \left\| \tilde{v}(p_k^l) - \tilde{v}(\Phi(p_k^l)) \right\|^2 \quad (3.15)$$

其中， $L^i$  代表一幅待拼接子图重叠区域的全部配对直线构成的集合， $P^l$  代表直线  $l$  的 2 个端点及 3 个四等分点构成的点集，图 3.11 对应展示了顺承之前采集的线特征获



取采样点的程序段。3.14 中其他符号定义与区域配准项类似。与区域配准项的思想类似，线性配准项通过约束线段上的点配准来达到线段配准的目的。

```
const vector<pair<int, int>> & images_match_graph_pair_list = parameter.  
getImagesMatchGraphPairList();  
vector<vector<vector<pair<pair<cv::Point, cv::Point>, pair<cv::Point, cv::Point>>>>>  
match_lines = get_line_match_result();  
for(int i = 0; i < images_match_graph_pair_list.size(); i++){  
    const pair<int, int> & match_pair = images_match_graph_pair_list[i];  
    for(int j = 0; j < match_lines[match_pair.first][match_pair.second].size(); j++){  
        Point pt1=match_lines[match_pair.first][match_pair.second][j].first.first;  
        Point pt5=match_lines[match_pair.first][match_pair.second][j].first.second;  
        Point pt3=0.5*(pt1+pt5);  
        Point pt2=0.5*(pt1+pt3);  
        Point pt4=0.5*(pt3+pt5);  
  
        Point ptt1=match_lines[match_pair.first][match_pair.second][j].second.first;  
        Point ptt5=match_lines[match_pair.first][match_pair.second][j].second.second;  
        Point ptt3=0.5*(ptt1+ptt5);  
        Point ptt2=0.5*(ptt1+ptt3);  
        Point ptt4=0.5*(ptt3+ptt5);  
  
        line_match_sample_points[match_pair.first][match_pair.second].  
            emplace_back(make_pair(pt1,ptt1));  
        line_match_sample_points[match_pair.first][match_pair.second].  
            emplace_back(make_pair(pt2,ptt2));  
        line_match_sample_points[match_pair.first][match_pair.second].  
            emplace_back(make_pair(pt3,ptt3));  
        line_match_sample_points[match_pair.first][match_pair.second].  
            emplace_back(make_pair(pt4,ptt4));  
        line_match_sample_points[match_pair.first][match_pair.second].  
            emplace_back(make_pair(pt5,ptt5));  
    }  
}
```

图 3.11 线对齐点集构造

图 3.11 中程序说明了构造线对齐所用点集的流程，线特征匹配信息在之前被转换为端点之间对应的 pair，依据一对相邻图像的索引调用该相邻图像间所有匹配线段的首尾结点，并连续折半获得所需采样点集。具体线特征匹配对如何转化为端点匹配对亦在 get\_line\_match\_result( )方法中有所实现。

### 3.2.5 线性保持项

线性保持项与线性对齐项共同作用以保留图像中的线性结构。相比线性对齐项，线性保持项作用于待拼接子图中提取到的全部线特征，即使用无匹配的单一线性特征构造约束项，确保了非重叠区域的线性结构在变形中得到一定的约束。

对于跨越多个网格的单一线段，我们希望这些网格在变形时保持一定的整体性，使得变形后这些网格截取的线段片段与变形之前具有相同的尺度，而这些网格在变换时受相似约束项的约束，进行的相似变换不会有太大差别而造成线段片段之间的旋转。这些网格在进行变形时的整体相似变换虽然会造成通过其中的线性结构的整体性旋转，但这种旋转并没有破坏到整体的线特征结构。线性保持项通过约束线段上采样点间距的不变达到上述效果，其具体形式为：

$$\Psi_c(V) = \sum_{i=1}^N \sum_{l_i^j \in L_i} \sum_{k=1}^{m-2} \|(\tilde{v}(\tilde{d}_{k+2}^j) - \tilde{v}(\tilde{d}_{k+1}^j)) - (\tilde{v}(\tilde{d}_{k+1}^j) - \tilde{v}(\tilde{d}_k^j))\|^2 \quad (3.16)$$

$L_i$  为待拼接图像  $I_i$  上所有直线的集合， $\tilde{d}_1^j, \tilde{d}_2^j, \tilde{d}_3^j \dots \tilde{d}_m^j$  表示线段上的采样点，如图 3.12 所示。

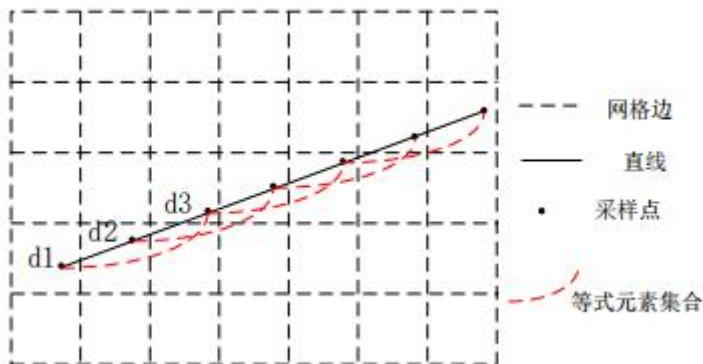


图 3.12 直线采样示意

与 3.2.1 同理，将采样点按照所在网格的四个顶点进行双线性插值表示，可以将表达式改写为线性方程组的形式。图 3.13 表示对线特征进行采样的程序。

```
for(int j = 0; j < res[k].size(); j++){
    double x = image_lines[j][0];
    double y = image_lines[j][1];
    int sample_points_num = 0;
    vector<Point2d> sample_points;
    for (int i=0; x <= image_lines[j][2] && x>=image_lines[j][0]; i++) {
        sample_points.push_back(Point2d(x, y));
        x += (sqrt(2)*lw*cos(atan(image_lines_slope[j])));
        y += (sqrt(2)*lw*sin(atan(image_lines_slope[j])));
        float min_y = (image_lines[j][3] > image_lines[j][1]) ? image_lines[j][1] : image_lines[j][3];
        float max_y= (image_lines[j][3] > image_lines[j][1]) ? image_lines[j][3] : image_lines[j][1];
        if(y>max_y||y<min_y)
            y -= (sqrt(2)*lw * sin(atan(image_lines_slope[j])));
        sample_points_num++;
    }

    res[k][j].reserve(sample_points_num);
    for (int ii = 0; ii < sample_points.size(); ii++)
        res[k][j].emplace_back(images_data[k].mesh_2d->getInterpolateVertex(sample_points[ii]));
}
```

图 3.13 直线采样程序示意

图 3.13 中程序说明了所实现的直线采样功能，可以看见直线被等间隔采点并存储，而每个采样点通过 `getInterpolateVertex()`方法被转化为网格顶点的线性插值。

为说明线性相关约束项对拼接效果中直线保持的作用及其对应权重调节的方法，在网格变形中取消该约束项作为对比，并将线性保持项权重手动调节至正常值（正常权重下具有线性保持效果且整体观感自然）的 10 倍以说明权重选择不宜过大，其他约束项权重保持在合理区间不变，如图 3.14。



(a) 取消线性相关约束项，道路弯曲



(b) 使用正常权重的线性保持项



(c) 使用十倍权重的线性保持项

图 3.14 线性保持项不同权重的直线保持效果对比（左侧为全景效果，右侧为局部放大）

图 3.14 (a) 中图像不提取图像线特征并进行相关约束，车道线部分弯曲明显，而图 3.14 (b) 中正常权重的线特征相关约束项对图像中直线结构的保持效果有所体现。图 3.14 (c) 调节线性保持项权重为合理数值（手动选取）的十倍，其他约束项基本失效，由于线性相关的约束项实际上采用的是线段上的采样点参与计算，故在这种严重扭曲下已经看不出直线保持的效果。其实与直线相关的约束项在设计时的思路十分直接，线性对齐项保证了变形后的全景图上，相邻图像重叠区域的线段仍然保持对齐，这对于人造场景十分有效；而线性保持项则依靠线段的片段长度不变性保证提取到的线特征结构在变形时保持一定的一致性，由于线段穿过的网格进行了一致的相似变换，而相似变换前的线段在变换后依然具有线性，故整个线段结构在相似变换中得到了保留。

### 3.2.6 待拼接子图的旋转与尺度计算

各待拼接图像在拍摄时由于各种原因可能无法保持旋转角度的一致性。在这种情况下，虽然可以使用具有旋转不变性的 SIFT 算法提取重叠区域的点特征匹配信息并指导图像配准，但由于缺少一个旋转基准的参考，网格变形会趋向于弯曲重叠区域图形结构来追求配准，而非重叠区域则整体变换至一个错误的旋转角度，最后的拼接结果既没有自然度也损失了真实场景的正确性。图 3.15 展示了待拼接图像间存在的旋转。



图 3.15 待拼接图像间存在的旋转

而由于拍摄时与被摄物体相对距离的变化，各待拼接图像中的相同场景也可能具有不同尺度，需要在网格变形相似变换时得到矫正。图 3.16 展示了这一尺度变化。



图 3.16 待拼接图像间相同场景具有不同尺度

### (1) 旋转角度计算

NISwGSP 中采用了 MLDR (Minimum Line Distortion Rotation) 选取各幅待拼接图像的旋转角度。单幅图像中提取的线特征经过 APAP 配准后可在相邻图像中找到其投影结果线段（线段两端点经 APAP 估计的单应性矩阵进行变换后相连得到的线段）。相邻图像中两两组成的每一对线段对都可以获得一个相对旋转角度，按照 RANSAC 筛选这些旋转角度并进一步利用所有相邻图像之间的相对旋转角度构造能量函数，使用最小二乘法得到每幅待拼接图像最优的旋转角度  $\theta_i$ 。

### (2) 尺度计算

不同图像间相同物体的尺度变化与相机透镜光心到被摄物体的距离成正比，对同一



物体的拍摄距离与相机焦距  $f$  亦成正比。图像的拍摄焦距  $f$  可以依靠 APAP 过程中得到的单应性矩阵估计得到，当前待拼接图像  $I_i$  的焦距估计值  $f^i$  与选定的参考图像  $I_c$  焦距估计值  $f^c$  作比就可以得到该子图相比参考图像的尺度缩放因子  $s_i$ 。

每幅待拼接子图的旋转角度  $\theta_i$  与尺度缩放因子  $s_i$  作用于 3.2.3 中全局相似项权重的计算，作为整幅图像相似变换的参考，使网格在变形过程中考虑到合理的旋转与尺度变化，尤其可以使得重叠区域的复杂变形自然过渡至非重叠区域的整体相似变换。

### 3.3 网格优化问题的解算

由上述介绍的 5 个网格优化约束项可以构造整体的能量函数，并由能量函数最小化表示最终希望得到的网格顶点集。

$$\tilde{V} = \arg \min_{\tilde{V}} (\lambda_a \Psi_a(V) + \lambda_l \Psi_l(V) + \Psi_g(V) + \lambda_m \Psi_m(V) + \lambda_c \Psi_c(V)) \quad (3.17)$$

$\lambda$  表示人工选取的各约束项权重系数。由 3.2.1 至 3.2.5 的推导可知，所有约束项都可转化为系数矩阵与网格坐标向量相乘的形式。因此，网格优化问题实际上是由 5 个约束项构造的最小二乘问题组成的，这些最小二乘问题的求解项都是网格顶点的位置，故可将问题合并。将各约束项对应的系数矩阵存储为一个稀疏矩阵，使用共轭梯度最小二乘法（Least-Square Conjugate Gradient）求解最小能量函数，最终就可以得到所有约束项同时作用的全局最优网格顶点坐标。利用变形前后的网格顶点坐标，可以求解一个网格区域内所对应的二维坐标仿射变换矩阵，利用该仿射变换矩阵即可将这一网格内的全部像素映射到网格变形后的区域。

### 3.4 本章小结

本章顺承第二章，介绍了利用特征匹配进行配准的 LAPAP 算法。并详细论述了网格优化问题中的各约束项及最终问题构建与解算的方法。对于不同的约束项，本章除了从原理角度叙述，亦给出了一些比较结果以深入说明各约束项的实际作用与作用机理。



## 第四章 拼接终末处理及全流程软件设计

本章介绍了拼接算法的最后几个环节，包含曝光补偿、图像融合及先验信息选取。而在 4.4 节中，对应所设计的完整拼接算法，介绍了程序设计中的大体框架及重要环节的软件实现细节。

### 4.1 曝光补偿

如 1.1 中定义所述，用以拼接的图像可以拍摄于不同时间，且自动选定曝光参数的拍摄设备也可能受环境影响在拍摄待拼接子图的过程中发生曝光水平的剧烈变化。曝光水平差异较明显的相邻图像除了造成观感上的不自然，对于后续图像融合步骤也有技术上的影响，尤其是最优缝合线这样的非线性图像融合方法，相邻图像重叠区域像素值的差异过大使得基于相似度的最优缝合线查找表现欠佳，且缝合线两侧分界对比明显。

曝光补偿针对每一幅网格变形完成但还未融合的待拼接图像进行，本设计采用增益补偿的方法<sup>[42]</sup>，为每幅图像赋予一个增益系数以达到图像间的曝光平衡。

定义误差函数：

$$e = \frac{1}{2} \sum_{(i,j) \in J} N_{ij} \left[ \frac{(g_i I_{ij} - g_j I_{ji})^2}{\sigma_N^2} + \frac{(1-g_i)^2}{\sigma_g^2} \right] \quad (4.1)$$

式(4.1)中  $N_{ij}$  表示两幅图像重叠部分的像素数量， $g$  代表希望求解的增益系数， $I_{ij}$  表示图像  $I_i$  在重叠区域的平均强度（像素值）。 $\sigma_N$  与  $\sigma_g$  分别表示误差和增益的标准差，为定值。想要求解误差函数最小时的增益系数，可对误差函数  $e$  分别求关于每幅图像增益系数  $g$  的导数并使导数为零，就可以解出各个待融合子图对应的增益系数。显然，这里采取的曝光补偿虽作用在单幅待融合图像上，但主要是为一致化相邻图像重叠区域的曝光水平而设计的。

### 4.2 图像融合

#### 4.2.1 平均融合与线性融合

平均融合或线性融合是较为简单的图像融合方式。平均融合即在两幅待融合图像的重叠区域将重合像素点的像素值各取一半再相加从而得到拼接结果的像素值。对于重叠

区域的任一点像素值，线性融合则是如(4.2)计算形式：

$$I(x) = (1 - \alpha)f_i(x) + \alpha f_j(x) \quad (4.2)$$

$I(x)$  表示所求的融合结果点  $x$  处像素值， $f_i(x)$  表示图像  $I_i$  中点  $x$  处像素值。不难发现，平均融合即为线性融合  $\alpha = 0.5$  时的特殊情况，而线性融合可以在重叠区域对来自某一幅图像的像素值赋予更大的比重，这可以在一定程度上淡化重影，使图像中的前景物体更加清晰。

#### 4.2.2 最优缝合线

最优缝合线查找算法是基于最大流最小割原理实现的一种图像切割算法<sup>[41]</sup>，最初是来源于图像分割问题的一种工具，亦可用作一种非线性地图像融合（缝合）算法。在最优缝合线查找算法中，图像间的缝合线搜索被转化为网络流的最小割问题。在本研究的图像拼接问题中，合理的缝合线可以穿过重叠区域中相似度较高的区域，使得图像拼接的接缝处看起来的差异极小。

网络流的最小割问题首先需要建立网络加权图，该加权图的节点是重叠区域的像素点坐标，而连接节点的边表示基于像素值相似程度构建的权值，其定义如下：

$$M(m, n, I_i, I_j) = \frac{\|I_i(m) - I_j(m)\| + \|I_i(n) - I_j(n)\|}{\|G_{I_i}^d(m)\| + \|G_{I_i}^d(n)\| + \|G_{I_j}^d(m)\| + \|G_{I_j}^d(n)\|} \quad (4.3)$$

$M(m, n, I_i, I_j)$  表示图像  $I_i$  和  $I_j$  的重叠区域中相邻两像素点  $m$  和  $n$  所构成的边的权值， $I_i(m)$  与  $I_j(m)$  分别表示图像  $I_i$  与  $I_j$  在点  $m$  的强度（像素值）。 $G_{I_i}^d(m)$  表示图像  $I_i$  在点  $m$  处的梯度。若相邻两点及其相互关系在两幅具有重叠区域的相邻图像间具有较强的相似性，该权值较小，这两点间可能被潜在的图割线穿过。图 4.1 示意最大流最小割原理。

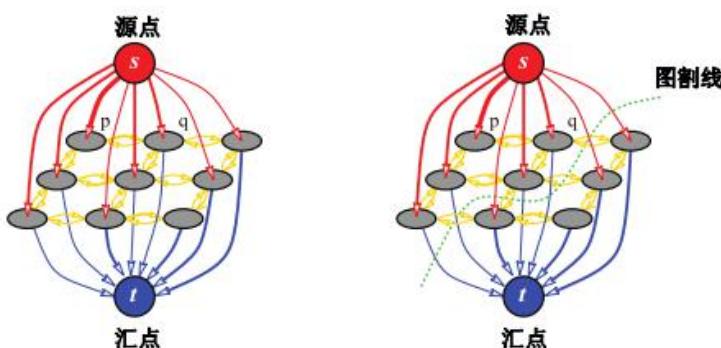


图 4.1 基于最大流最小割原理的图割线分割

如图所示，最小割问题找到权值最小的边并依次穿过形成图割线。在图像融合时，图割线切割后，依然与汇点连通的节点代表该位置采用汇点所代表图像的像素值，与源点连通的节点代表该位置采用源点所代表图像的像素值。图割线像是缝合线一般将两幅区域共视区域的待融合子图缝合在一起。这种非线性的融合方法相比 4.2.1 最大的优势在于重影与伪影的消除，由于对应共视区域同一目标物体的像素点只会从两幅待融合图像像素点二选一而非线性叠加，同一目标物体的影像就不会由于未能完全配准对齐而出现在拼接结果的不同区域，造成伪影或重影。

### 4.3 先验信息辅助

从上述环节不难看出，图像特征提取对整个图像拼接算法有极其重要的作用，待拼接图像重合区域的配准依赖特征匹配对，重叠区域的配准质量又影响网格变形的自然程度，依据配准产生的重叠区域变形继续影响着融合算法的表现。图像特征亦参与尺度与旋转计算，对全局相似性有所影响。相比以往拼接算法，本设计虽然采用了分布较为均匀密集的 SIFT 特征并辅以线特征，但在一些纹理稀疏区域还是无法采集到足够的特征以指导图像拼接，尤其是线特征匹配对的算法自动提取非常困难，精度也有限，往往会造成一些重要的匹配对使得一些视觉上较明显的特征在拼接后没有得到理想的效果。

观察拼接结果，针对一些没有得到良好配准或图形结构保持的区域，本设计提供手动选取线特征匹配对的先验信息提取环节。手动选取的特征匹配作为算法选取的补充，可以对选取的特定区域拼接效果产生显著的影响。图 4.2 展示了先验信息选取的提出背景。



图 4.2 线特征匹配算法在纹理稀疏区域的表现并不理想，错误匹配较多  
先验信息提取流程如图 4.3 所示：

```
Foreach image ∈ ImagePair do
    line_start_point ← point_selected1
    line_end_point ← point_selected2
    If confirm do
        line_selected ← make_line(line_start_point, line_end_point)
        line_feature ← Add line_selected
        line_pair ← Add line_selected
    End
    Clear line_start_point, line_end_point, line_selected
End
If line_pair.size = 2 do
    match_pair(line_pair[1], line_pair[2])
    Clear line_pair
End
...
LAPAP
...
Mesh Optimization
```

图 4.3 先验信息提取流程

先验信息虽然没有从原理上提升拼接流程，但其对拼接效果具有重要的实践意义。观察拼接全景图中具有明显伪影及重叠部分扭曲严重的区域，手动选取该区域线特征匹配对，对局部区域的效果产生修补作用而不破坏整体拼接算法在全景图其他部分的表现。在工程上，这是一种反馈的思想。



(a) 全景图中有明显伪影的区域

(b) 手动选取先验信息增强后



(c) 相邻图像手动选取先验信息示意图

图 4.4 先验信息提取说明

图 4.4 说明了在全景图存在伪影的区域选取显眼信息增强的效果，手动选取地面停车场杆上的直线匹配信息后，局部区域的配准效果显著提升，伪影被消除，而周边区域效果没有受到明显影响。使用者可以通过先验信息针对性提升拼接结果中的局部效果，对算法的工程实践有着重要意义。

## 4.4 全流程软件实现

本节将介绍本设计提出的拼接算法的各个环节软件实现方法，本算法在实现时主要基于 NISwGSP 开源项目框架。本研究在文件解析、特征提取、网格优化直到拼接终末处理等环节都进行了不同程度的改进与工具设计，这些新设计的程序都被有机的融入到了原框架程序中。

### 4.4.1 项目架构概览

图 4.5 是整个软件设计的项目构成示意。

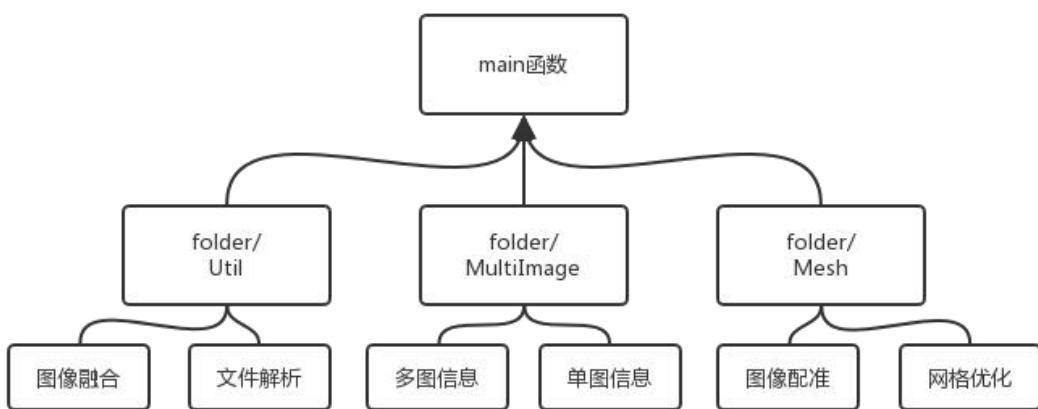


图 4.5 项目构成

本设计的 C++ 项目主要由三大部分构成，Util 部分包含工具性组件，如图像融合与文件解析相关的程序与头文件。MultiImage 文件夹包含表述多图信息与单图信息的程序与头文件，这部分主要有单张图对象提取到的特征、多图相邻关系及多图间特征匹配对信息。Mesh 文件夹主要处理与网格相关的问题，包含 LAPAP 图像配准及网格优化的相关定义与处理。

多图信息的处理是算法的重点，所以 MultiImage 类型包含大量数据成员与方法。而网格优化相关的处理主要依靠 MeshOptimization 类完成，该类包含了对各约束项的定义与网格优化问题中最小二乘问题的构建。文件解析部分的设计已在 2.1 节中介绍，后续小节将进一步介绍软件实现的其他细节。

### 4.4.2 图像线特征提取与匹配程序实现

单张图像的线特征储存在单张图像类型 ImageData 中的 keylines 成员中，以 OpenCV 中 KeyLines 类型构成的数组描述。提取单张图像线特征的方法定义为 getKeyLines()。在该方法中，首先将图像转换为单通道灰度图像，然后定义一个 LSD 方法的智能指针



以调用 OpenCV 中的 LSD 线特征提取算法。之后，LBD 描述符同样以智能指针形式被定义，并对 LSD 算法提取的线特征进行描述，描述子信息被储存。每幅图像提取到的线特征及描述信息都存储在该幅图像对应的 ImageData 对象的数据成员中，以方便多图对象进行后续使用。

在 MultiImage 类型中，getLineFeaturePairs()方法用来对相邻图像进行线特征匹配，其返回类型是一个三维数组，前两维对应一对相邻图像，最后一维用来储存匹配对信息，该信息以 DMatch 类型记录，DMatch 类型存储了一对线特征在原各自图像全部线特征中的索引，根据 DMatch 信息可找到相邻图像中的具体线特征。要想获得线特征匹配对，首先要根据 ImageData 类型中存储的描述子信息进行特征匹配，这里使用 BinaryDescriptorMatcher 智能指针实现。获取到的匹配对之后借由 2.4 中介绍的 Sequential RANSAC 程序进行筛选，得到了最终的线特征匹配对。在程序设计的过程中发现，线特征 RANSAC 较点特征 RANSAC 误差较大，故在设计中对线特征关于首尾端点进行两轮 RANSAC 以提升筛选的效果。在后续使用中，还定义了 getLineFeatureMatches()方法用来按照 DMatch 索引获得线特征 KeyLine 对象，这样才能直接使用所需线特征匹配对。

#### 4.4.3 图像配准及网格优化程序实现

LAPAP 配准部分可以使用一个函数完成，其输入参数包括参考图像、目标图像分别的匹配点 Point2 数组与匹配线 KeyLine 数组，以及参考图像的网格顶点 Point2 数组。输出参数则含有目标图像中的配准点 Point2 数组及用 Mat 类型存储的局部单应性矩阵。

首先使用 KeyLine 类型的固有方法获得匹配线的两端点并计算得到三等分点，将这些点与匹配点存入同一个向量。由于这些点的坐标较为分散，首先通过变换矩阵将所有点的坐标标准化（Normalization），使得坐标数值较为集中方便后续计算。之后进一步将这些点线性变换，使坐标位于一个方便计算的数值区间。按照 3.1 定义，运算过程中的参数被引入，而系数矩阵则使用科学计算库 Eigen3 提供的 MatrixXd 类型初始化，其大小为  $2 \times (N+M \times 5) \times 9$ , N 代表匹配点的数量而 M 代表匹配线的数量，也即一个系数矩阵在垂直方向叠加了一个待配准网格点所需的全部系数。之后的计算是一个双重循环，对于每一个待配准网格点，所有匹配点及匹配线所对应的权重都要被计算并按位置存入前面定义的系数矩阵中。调用 Eigen3 的 JacobiSVD 方法对系数矩阵的右奇异矩阵进行求解，逆标准化即可得到该待配准网格点对应的局部单应性矩阵。图 4.6 为该部分程序的

流程图。



图 4.6 LAPAP 程序流程

3.3 中已经阐明，网格优化问题的本质是关于所有网格顶点变形后坐标的  $AX=B$  型最小二乘问题求解问题，其中  $X$  代表网格变形后坐标叠加组成的向量，相邻元素为同一网格顶点的  $x, y$  坐标值， $A$  代表以能量函数构建的稀疏矩阵。在程序实现过程中，首先要按照约束项的个数初始化稀疏矩阵，在程序中用 `reserveData_line()` 方法实现。在该方法中，使用三元组 `triplets` 存储稀疏矩阵元素坐标及该矩阵位置对应数值。稀疏矩阵中存储的具体数值计算按照 3.2 中介绍进行。在这一环节中，任意点以其所在网格的四个顶点的插值表示，所以区域配准项及线性对齐项在稀疏矩阵中的实际位置数量要乘以 4 倍。稀疏矩阵数值填充完成后，以 `triplets` 作为参数可直接构造 Eigen3 中的 `SparseMatrix` 类型对象，再用 `LeastSquaresConjugateGradient` 方法求解最小二乘问题，可得变形后网格的全局坐标。图 4.7 表示利用三元组 `triplets` 存储稀疏矩阵的程序示意。



```
//image1
    _triplets.emplace_back(equation + eq_count + k, images_vertices_start_index[m1] + DIMENSION_2D *
        poly_vertices_index_m1[polate_point1.polygon].indices[0] + k,
        polate_point1.weights[0] * line_match_weight);
    _triplets.emplace_back(equation + eq_count + k, images_vertices_start_index[m1] + DIMENSION_2D *
        poly_vertices_index_m1[polate_point1.polygon].indices[1] + k,
        polate_point1.weights[1] * line_match_weight);
    _triplets.emplace_back(equation + eq_count + k, images_vertices_start_index[m1] + DIMENSION_2D *
        poly_vertices_index_m1[polate_point1.polygon].indices[2] + k,
        polate_point1.weights[2] * line_match_weight);
    _triplets.emplace_back(equation + eq_count + k, images_vertices_start_index[m1] + DIMENSION_2D *
        poly_vertices_index_m1[polate_point1.polygon].indices[3] + k,
        polate_point1.weights[3] * line_match_weight);

//image2
    _triplets.emplace_back(equation + eq_count + k, images_vertices_start_index[m2] + DIMENSION_2D *
        poly_vertices_index_m2[polate_point2.polygon].indices[0] + k,
        -polate_point2.weights[0] * line_match_weight);
    _triplets.emplace_back(equation + eq_count + k, images_vertices_start_index[m2] + DIMENSION_2D *
        poly_vertices_index_m2[polate_point2.polygon].indices[1] + k,
        -polate_point2.weights[1] * line_match_weight);
    _triplets.emplace_back(equation + eq_count + k, images_vertices_start_index[m2] + DIMENSION_2D *
        poly_vertices_index_m2[polate_point2.polygon].indices[2] + k,
        -polate_point2.weights[2] * line_match_weight);
    _triplets.emplace_back(equation + eq_count + k, images_vertices_start_index[m2] + DIMENSION_2D *
        poly_vertices_index_m2[polate_point2.polygon].indices[3] + k,
        -polate_point2.weights[3] * line_match_weight);
```

图 4.7 稀疏矩阵存储示意

图 4.7 中程序设置的是线性对齐项的约束项系数计算值，可见三元组增添元素时的前两个参数表示稀疏矩阵中的坐标位置，而最后一个参数对应约束项的系数计算值乘以相应约束项权重。此处说明的是线性对齐项，程序中后半部分所用元素值相比前半部分增加负号，这与 3.2.4 中介绍的原理相印证，代表约束项转换为最小二乘问题时，配准点坐标插值相互之间的作差关系。

#### 4.5 本章小结

本章介绍了待拼接图像经过网格变形后，进行图像融合、曝光补偿的终末处理方法。在图像融合环节，线性融合与非线性的最优缝合线方法均得到介绍。而本章的后半部分从软件设计的角度介绍了整个图像拼接算法的实现，包含了程序开发上的结构与细节。

## 第五章 拼接算法对比与评价

本章是实验章节，按照第二章至第四章所介绍的流程实现的本设计最终拼接算法与以往方法进行了整体性的比较。5.2 首先定义了图像拼接效果中的大视差问题，围绕大视差问题带来的挑战，本章在 5.3 节中对各方法拼接结果进行了主观评价与定量分析，旨在说明以往方法应对大视差问题时的固有缺陷以及本设计方法的优越性。5.4 节介绍了本设计方法中引入的一些有效组件的实际作用。

### 5.1 大视差问题

视差 (disparity) 是立体视觉里的一个基础概念，指在处于一定距离的两个点观察同一物体时，该物体在两个成像平面的位置差异。人的双眼就是典型的平行视差系统，以双眼连线所在水平方向为 x 轴方向，同一物体在左眼成像与右眼成像的水平 x 坐标具有差异，这一差异就是视差。视差以像素值为单位，具体公式如(5.1)所示。

$$Z = \frac{b \times f}{d} \quad (5.1)$$

视差  $d$  与两眼间距  $b$  (基线, baseline) 及所观察物体深度  $Z$  (拍摄距离) 有关，其中，视差大小与深度成反比，即被摄物体离相机越近，视差越大， $f$  表示相机焦距。图 5.1 表示同一棵树在平行拍摄的相邻图像间存在视差。



图 5.1 同一棵树在相邻图像存在视差

在图像拼接问题中，相邻待拼接图像同样具有视差。当图像中包含处于三维空间不同深度的物体时，离相机距离较近的物体视差较大，而远景物体视差较小，这种在同一组图像中由于拍摄物体分布远近不同造成视差变化较大的问题是拼接问题中的第一种“大视差”问题。图 5.2 中，离摄像机较近的石雕与距离较远的建筑物具有不同的视差。



图 5.2 第一种“大视差”问题

第二种“大视差”问题则指待拼接图像的拍摄基线  $b$  较大, 对于空间中同一被摄物体  $P$ , 其与相邻图像拍摄位置  $P_1$  及  $P_2$  连线的夹角称为视差角  $\beta$ , 如图 5.3 所示。

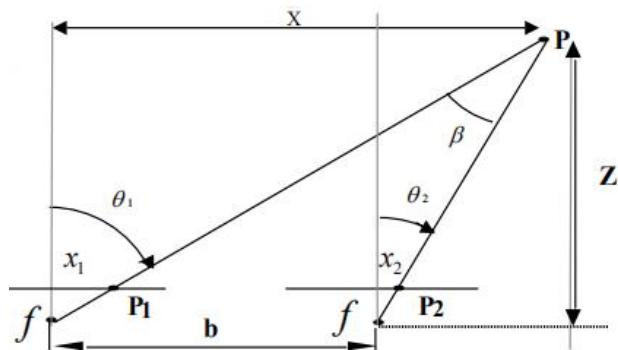


图 5.3 视差角

当被摄物体深度一定时, 视差角与基线成正比, 如公式(5.2)。

$$Z \approx \frac{b}{\beta} \quad (5.2)$$

对于空间中深度一定的物体, 当相邻待拼接图像拍摄间距变大, 视差角变大, 而距离较近的被摄物体显然视差角也较大, 这就是第二种“大视差”问题, 图 5.4 中电脑即具有较大视差角。

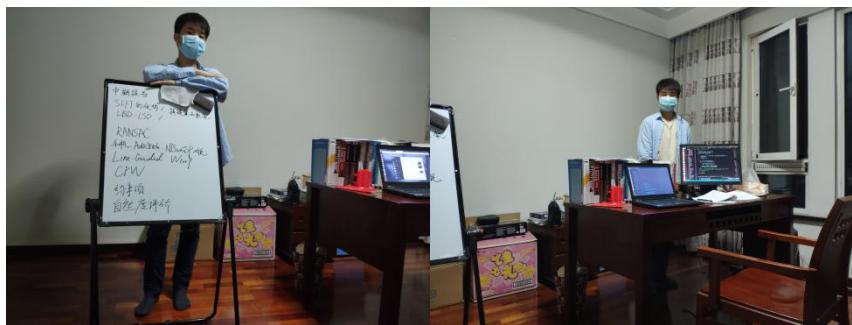


图 5.4 第二类“大视差”问题, 待拼接图像重叠区域的电脑具有较大视差角

对于第一种“大视差”问题, 距离较近的物体与较远的物体需要用不同的单应性矩阵进行配准才能使重叠区域具有良好的配准效果。对于以 AutoStitch 为代表的传统基于全

局单应性矩阵的拼接算法，整幅图像使用估计出的唯一单应性矩阵进行配准、变换，这就会造成远景区域得到较好配准变形而距离较近景物的合理配准变形被忽略，可能产生严重的投影变形，图 5.5 展示了局部单应性方法中没有得到良好配准（产生伪影）的空间距离较近景物。本设计采用基于局部单应性的方法，可以同时考虑到不同深度景物的透视形变，并进一步使他们都获得合理的变形。



图 5.5 第一种“大视差”问题造成的伪影

对于第二种“大视差”问题，视差角较大造成观察同一物体时潜在的视角变化，也就难以提取匹配特征并达到配准，视角的变化意为着同一物体在不同图像中的空间旋转与尺度发生变化，重叠区域就要进行更为复杂的透视投影，连带非重叠区域变形的复杂度提升。视差角较大的场景重叠区域也较小，亦造成特征匹配对减少。即使是基于局部单应性的拼接方法，较少的特征匹配对也难以指导重叠区域的图像配准，且无法估计尺度与旋转，得到的网格变形也就不十分合理。想要解决这一问题，首先可以想办法增加特征匹配对，如本设计加入直线匹配及先验信息，其次可以在配准过程中利用 APAP 这样的方法将特征匹配对的配准信息均匀分摊到整个重叠区域以缓解配准信息的不足。图 5.6 中由于视差角较大，传统拼接方法在局部区域出现了不正确的尺度与旋转，地面完全弯曲。

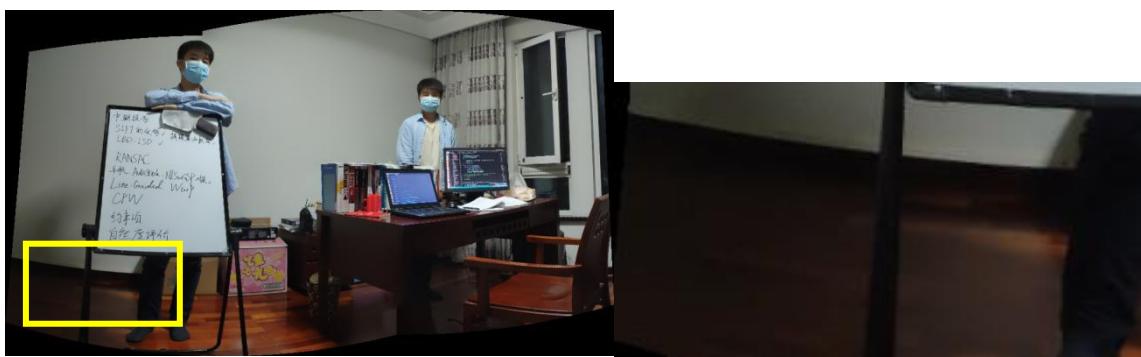


图 5.6 第二种“大视差”问题造成的不合理变形（左侧为全景效果，右侧为局部放大）

为了避免大视差，传统的图像拼接算法一般要求整个图像处于同一空间视觉平面以避免图像中深度变化，且各次拍摄时相机光心近乎重合以避免较大视差角。本设计对于两种“大视差”问题都加入了针对性的组件，其效果在本章后续分析中得到体现。



## 5.2 算法效果对比与分析

本设计的主要框架来源于 NISwGSP 开源项目，但在程序设计的实现上进行了大量调整与修改，且融入了许多其他计算机视觉及图形学的组件以提升拼接效果，在程序可读性、健壮性上做了大量工作。整个项目的开发在 Ubuntu16.04 64 位操作系统下进行，开发语言为 C++11，编译环境采用 CMake 3.5 + GCC 5.7.0。算法运行主机配置为 2.60GHz Intel Core CPU。

所有图像文件在读取时被高斯降采样至  $800 \times 600$  像素（可在项目配置文件中调整，更大分辨率图像虽拼接效果更优，但处理时间较长），点特征提取及描述使用轻量级计算机视觉库 VLFeat 中的 SIFT 算法。在配准及网格优化时用到的网格大小为  $40 \times 40$  像素（可在项目配置文件中调整）。网格优化能量函数各系数  $\lambda_a = 1$ ,  $\lambda_l = 0.75$ ,  $\lambda_m \in [1,2]$ ,  $\lambda_c \in [1,2]$ ,  $\beta = 6$ ,  $\gamma = 20$ ，可以按照对配准、变换相似性、线性结构保留的需求进行参数调节。曝光补偿采用 OpenCV 库中的 ExposureCompensator 类型实现，最优缝合线则采用 detail::GraphCutSeamFinder 类实现。先验信息的手动选取依靠图上画点作线实现，可以重复选择，使用较为方便。

本设计亦复现了 NISwGSP 算法与 AutoStitch 算法作为实验对比，其中 AutoStitch 算法利用 OpenCV 中的 Stitcher 模块实现。在商业图像拼接软件方面，本研究选择谷歌应用商店中下载量最高的 Bimostitch 软件及小米 8 手机自带的全景拍摄功能作为对比。这些算法无需调节参数。实验所用图片来自自主采集及前人研究公开资料。

### 5.2.1 直观评价

直观评价主要是主观的评价拼接结果的观感自然度，包括对景物的形变（包含尺度与旋转相对真实观感的偏差）、曝光的平衡、重叠区域的对齐（重影）、线结构的保持等特性的观察。图 5.7 包含手机拍摄、商业软件 BimoStitch 拼接、采用平均融合的 NISwGSP 算法及本设计实现算法的拼接结果。



(a) 小米 8 手机全景拍摄



(b) BimoStitch+软件自带曝光平衡



(c) NISwGSP (Average Blending)



(d) 本设计方法 (线性融合)



(e) 本设计方法 (最优缝合线)

图 5.7 各拼接方法最终效果 I

图 5.7 (a) 中, 手机直接拍摄的全景图在前景与背景都具有较大的扭曲变形, 尤其是中间距离拍摄设备较近的建筑物部分, 由于其视差大于建筑物其余部分, 受前述第一种“大视差”问题的影响, 其投影变形 (Projective Distortion) 相比其他拼接结果最为显著。前景的地面也扭曲严重, 观感自然度较差。

图 5.7 (b) 中较远处建筑物的整体形状较为流畅, 但较近处的石雕受第一种“大视差”问题的影响没有得到良好配准, 且在进行线性融合时产生了明显的重影, 受第二种“大视差”问题影响, 背景建筑物的招牌文字部分由于视角变化没有得到合理的投影变形, 如图 5.8 所示。利用 BimoStitch 自带的曝光补偿功能进行曝光平衡后, 图像左半部分曝光水平虽然得到统一, 但整体过曝影响观感。



图 5.8 BimoStitch 中受大视差问题影响带来的重影及投影失败

如图 5.7 (c) 采用局部单应性+网格优化变形的 NISwGSP 算法相比前两种算法对整体失真抑制上有了进一步的提升, 前景停车场与背景建筑都得到了正确的尺度与合理的变形。采用平均融合的融合效果较差, 伪影及重影显著多于线性融合, 且重叠部分过渡生硬, 如图 5.9。



图 5.9 NISwGSP 采用平均融合, 各拼接图像交界处过渡生硬

如图 5.7 (d) (e), 本设计的最终拼接效果保留了原始局部单应性+网格优化变形的

优点，变形中的尺度正确且观感自然。如图 5.10，得益于直线特征相关组件的作用，前景地面的停车位划线更直，整体弯曲更少，即线性结构得到了保留。图 5.11 可参考图 5.8 作为对比，由两类大视差问题造成的近处物体无法配准及视角变化引起的错误透视投影得到了解决，建筑物上的招牌文字变得更加清晰，且空间距离较近的石雕处消除了重影。整幅拼接全景图的曝光正常，曝光过渡也较为自然，达到了曝光平衡。



(a) 图 5.7 (a) 中手机拍摄的局部区域地面停车线



(b) 图 5.7 (e) 中本设计方法的局部效果

图 5.10 全景图中线结构保留效果对比



图 5.11 图 5.7 (e) 针对性解决了大视差问题带来的影响

可以看到，本设计的方法在投影变形及透视失真上都有较好抑制，景物整体比例协调且同一物体的视角过渡连贯，也即符合正常的透视关系。在线特征保持即重影消除上的表现也有独到之处。相比商业软件、经典方法及近年来提出的一些算法，本设计的最终效果处于前列。图 5.12 在另一场景体现了本设计相比全局单应性方法的优良效果，建筑物作为场景中深度变化较大的物体，受第一种“大视差”问题的影响，建筑物的许

多部分没有得到正确的单应性矩阵变换，在拼接结果中的尺度及旋转都不正确。

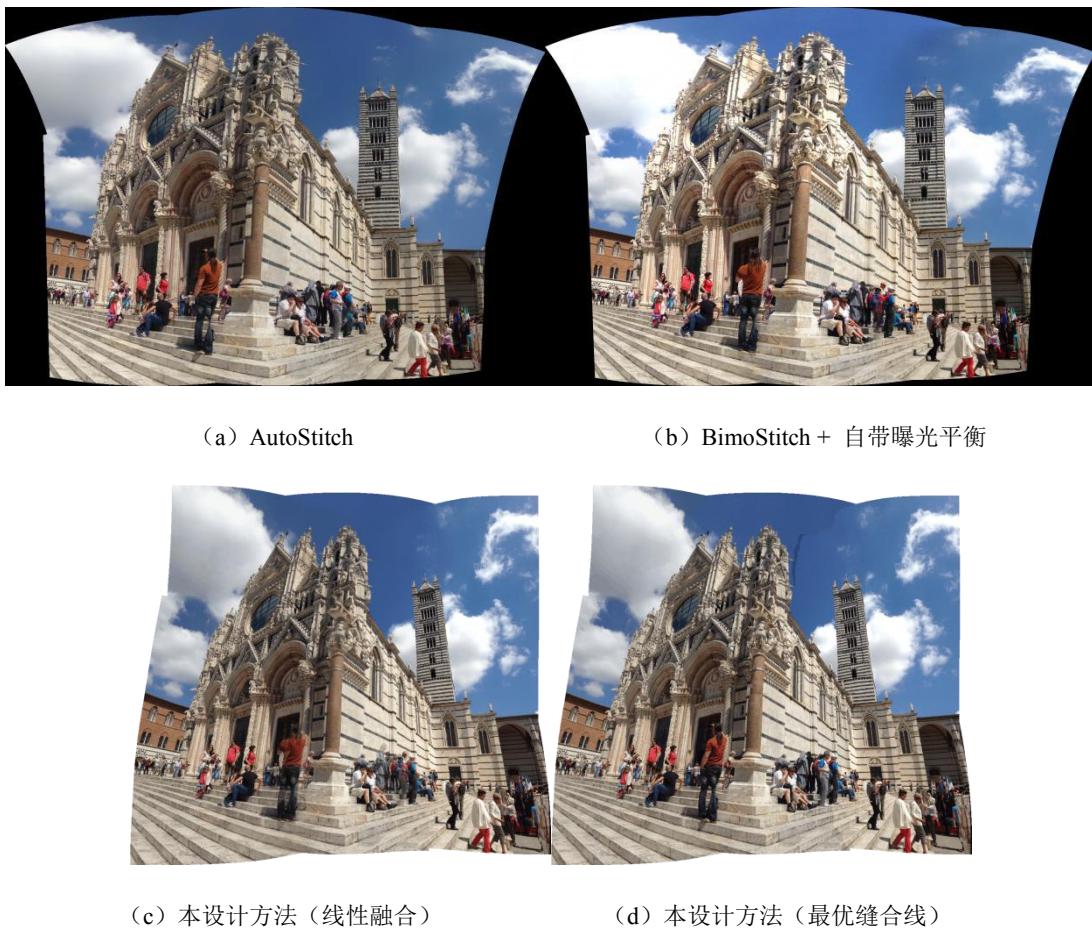


图 5.12 各拼接方法最终效果 II

### 5.2.2 投影变形评价

投影变形 (Projective Distortion) 是图像拼接中较为常见且明显的一类失真，主要体现为局部图像的不正常尺度伸缩及旋转。其产生原因主要是局部的变换没有保持相似性导致目标物体发生扭曲，对观感自然度有较大影响。

这里选取的作为对比的例子，为了同时配准背景的建筑物与前景的地面及人群，对拼接算法的尺度变换与扭曲程度有较大挑战。选取前景中的两个行人目标，测量其从头顶至脚部连线在不同拼接效果中的纵向尺度变化与旋转，代表对拼接前后投影变形的定量描述。尺度与旋转的基准参考原始待拼接图像，连线由人工选取，尺度以连线长度的像素为单位，旋转则是以连线与水平面夹角衡量 ( $<90^\circ$ )。

尺度失真  $n_s$  如(5.1)定义：

$$n_s = \left| \frac{n_i - n_0}{n_0} \right| \times 100\% \quad (5.1)$$

$n_0$  表示待拼接图像中目标长度的基准值， $n_i$  表示拼接效果中同一目标的长度。

同理，定义旋转失真  $r_s$ ：

$$r_s = \left| \frac{r_i - r_0}{r_0} \right| \times 100\% \quad (5.2)$$

$r_0$  表示待拼接图像中目标相比水平轴的旋转基准值， $r_i$  表示拼接效果中同一目标相对水平轴的旋转角度。图 5.13 是不同算法结果及其中用以对比的前景目标。表 5.1 及 5.2 记录两个目标在不同算法结果中的尺度失真与旋转失真结果。目标 1 是图像中左侧站立行人，目标 2 是图像中右侧骑车行人。

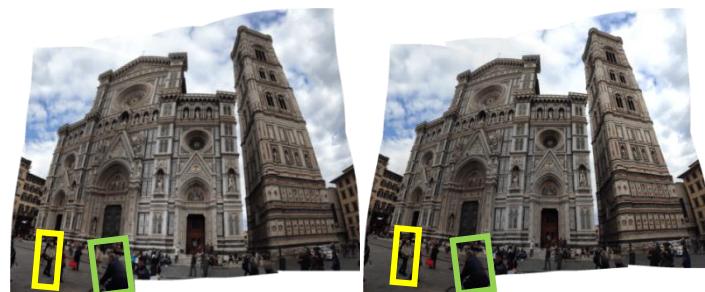


(a) 原始待拼接图像



(a) AutoStitch

(c) BimoStitch



(d) NISwGSP

(e) 本设计方法（线性融合）



(a) 原始待拼接图像 (b) AutoStitch (c) BimoStitch (d) NISwGSP (e) 本设计方法 (线性融合)

图 5.13 投影变形样例

表 5.1 目标 1 投影变形记录

图像	目标纵向尺寸 (像素)	目标旋转角度 (°)	尺度失真	旋转失真
原始待拼接	154.055	83.2926		
AutoStitch	137.463	81.6365	10.77%	1.99%
BimoStitch	139.514	82.2711	9.44%	1.23%
NISwGSP	147.895	80.9713	4.0%	2.79%
本设计方法 (线性融合)	151.753	81.2849	1.49%	2.41%
本设计方法 (最优缝合)	151.753	81.2849	1.49%	2.41%

表 5.2 目标 2 投影变形记录

图像	目标纵向尺寸 (像素)	目标旋转角度 (°)	尺度失真	旋转失真
原始待拼接	180.76	59.038		
AutoStitch	148.111	56.8481	18.06%	3.71%
BimoStitch	150.712	56.7126	16.62%	3.94%
NISwGSP	153.42	61.3336	15.13%	3.89%
本设计方法 (线性融合)	157.156	60.6639	13.06%	2.75%
本设计方法 (最优缝合)	156.78	60.1662	13.27%	1.91%

由实验结果不难看出, AutoStitch 与 BimoStitch 由于采用自由度较低的全局单应性矩阵进行配准变换, 前景具有不同视差的目标无法满足变换相似性的需求, 产生了较严重的尺度失真。而采用网格优化方法的拼接算法在变形过程中约束了局部地区的变换相似性, 使得目标尺寸得到了较好保留。由于配准的需求, 一些区域的原始尺度会被一定程度地牺牲, 而待拼接子图也会自主选择一个合适的旋转角度使目标发生旋转。由于这

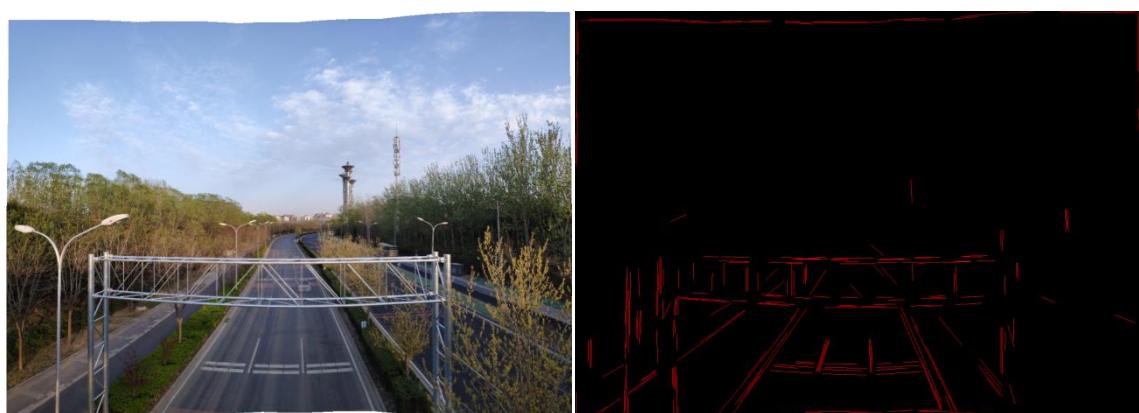
个样例的拼接前各子图间相互旋转不大，所有算法在前景人物的旋转保持上都有不错的表现。总的来看，本设计得益于良好配准及网格优化约束的设计，在投影变形上有较好抑制作用，最终的拼接结果在观感自然的基础上保证了局部目标图形结构的正确性。

### 5.2.3 线性结构保持评价

本设计依靠线特征提取、LAPAP、线性相关网格优化约束项及先验信息等一系列新增环节约束线特征配准及线性结构保持，效果显著，充分利用了拼接结果中的高维信息。本小节利用 LSD 算法提取各算法拼接结果全景图中的线特征。筛选长度较长（大于 35 像素）的线特征作为有效线特征并统计每种拼接算法结果中的线特征数目，作为评价线特征保持的一种指标。此外，对于前景中明显的直线结构，计算线结构中点距两端点连线的垂直距离作为线结构弯曲程度的评价指标。图 5.14 是一个典型的线结构丰富场景，左侧一列为各拼接方法得到的全景拼接效果，右侧一列为对应的 LSD 线特征提取结果。



(a) AutoStitch



(b) NISwGSP

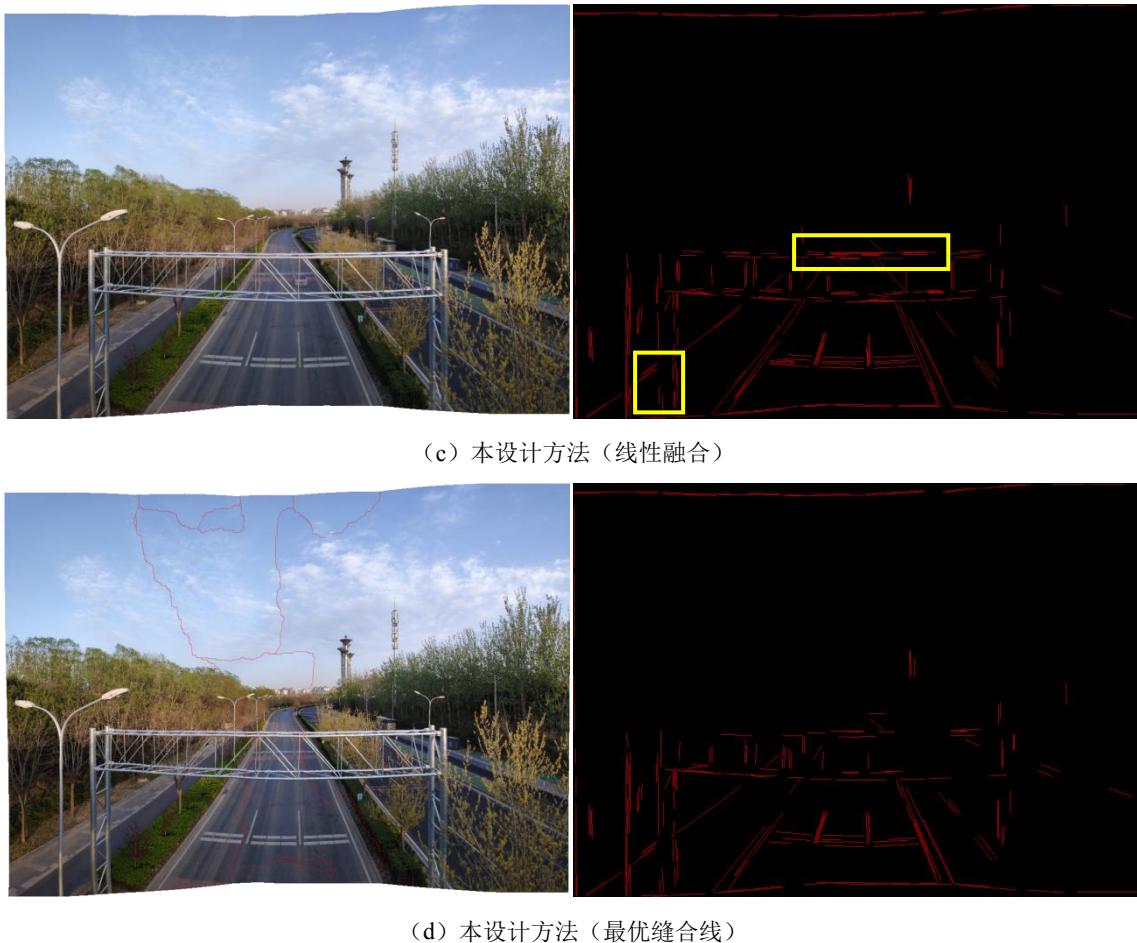


图 5.14 线结构保持评价样例 (左侧为拼接全景图, 右侧为线特征提取结果)

图 5.14 中, 左侧一列为不同方法拼接全景图, 右侧一列为对应线特征提取效果。可见, 该场景中主要的线结构存在于前景的车道线及横跨图像的桁架结构, 保持横跨多张子图的桁架结构对算法的直线保持能力提出了挑战。在之后的对比中, 以该桁架的结构中心点到桁架两端点所连线的垂直距离作为对该场景线结构弯曲程度的定量衡量。

图 5.15 与图 5.16 是各算法结果中线特征数量以及线结构弯曲的统计结果。图 5.15 纵轴表示全景图中提取到的有效直线数量, 图 5.16 纵轴表示桁架的结构中心点到桁架两端点所连线的垂直距离, 横轴对应各算法。图 5.15 中可以看到本方法结合线性融合在最终全景图中相比其他方法保留了最多的直线, 而使用最优缝合线进行图像融合, 保留的直线数量有所下降。图 5.16 中体现了本设计方法对桁架整体弯曲的较好抑制, 桁架的结构中心点到桁架两端点所连线的垂直距离最小。

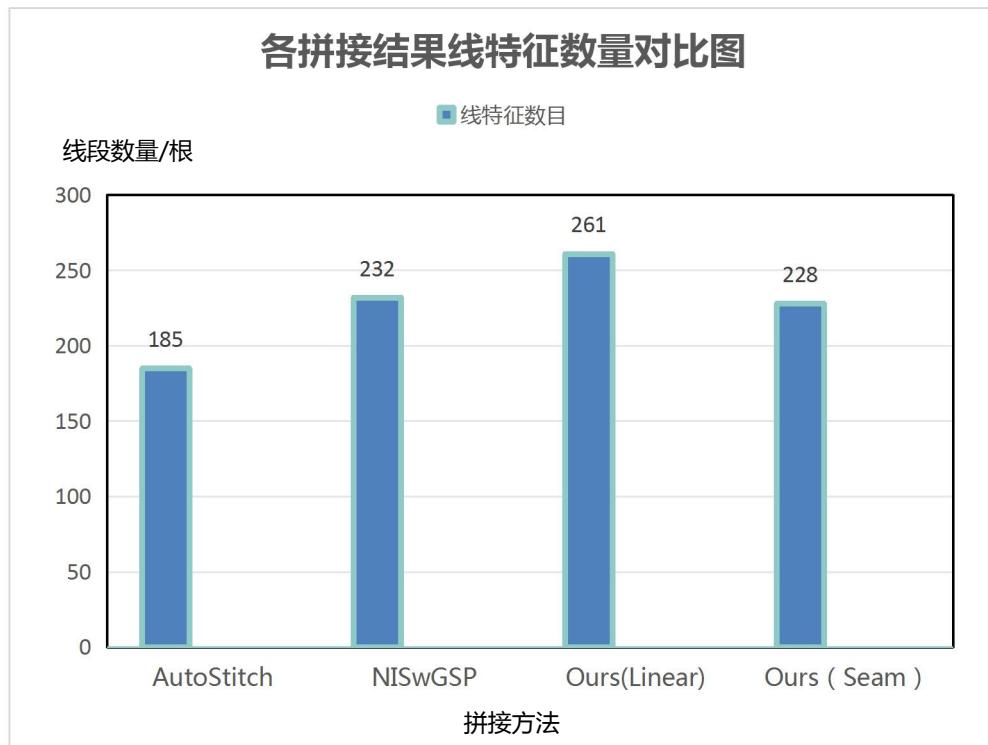


图 5.15 提取到的线特征数量结果柱状图

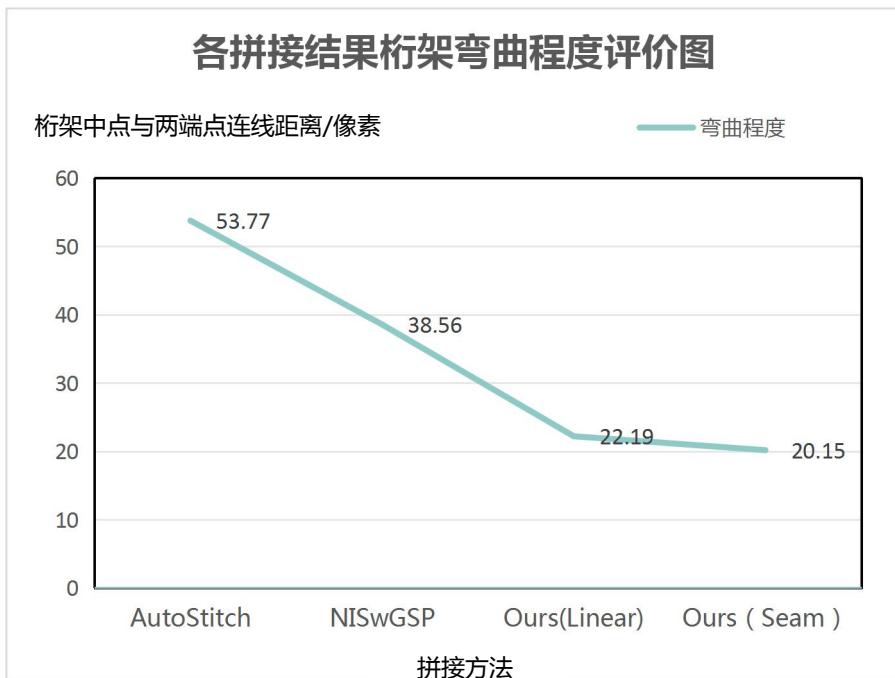


图 5.16 弯曲程度评价折线图

可以直观地看到，本设计方法极大程度保留了前景桁架及道路的直线结构，图 5.15 最终结果中提取到的线特征最多，而传统 AutoStitch 拼接结果中桁架线结构损失严重，整体弯曲大，道路边线也发生了明显的弯曲。由于最优缝合线在图像融合时的非线性特

性，各子图之间的线性结构有一定不连贯，也造成了桁架线结构的损失。

线结构的保留对于最终拼接全景图中的进一步信息提取有重要的作用，本设计算法已经基本达到了保留明显高维特征的目的，并极大抑制了高维特征的结构破坏。

### 5.3 各组件作用分析

为了说明并分析本设计在图像拼接过程中加入的各环节及工具组件对拼接效果的实际影响，本节通过消融实验的方式控制变量，控制特定模块的有无以对比该模块对最终拼接效果的作用。

#### 5.3.1 LAPAP 优化配准

LAPAP 是基于原始 APAP 思想加入线特征匹配进行配准的一种图像配准方法，图 5.17 中样例分别采用原始 APAP 配准及 LAPAP 配准，其他处理环节保持一致，可以发现 LAPAP 对局部地区的配准效果提升明显。



(a) APAP 配准拼接效果（左侧为全景效果，右侧为局部放大）



(b) LAPAP 配准拼接效果（左侧为全景效果，右侧为局部放大）



(c) 第一行代表使用 APAP 配准的拼接结果，第二行以 LAPAP 配准替换，第三行为局部放大

图 5.17 LAPAP 效果对比样例

图 5.17 可以看到，在第一组示例中，采用 LAPAP 配准的拼接全景在局部广告牌上的文字相比采用 APAP 配准的结果有极大的可视度提升，文字内容变得清晰可见。而第二组示例中，电脑具有丰富的线性结构特征，利用 LAPAP 使其在过渡区域得到了良好对齐。

原始的 APAP 虽然利用局部单应性在图像配准中相比全局单应性已经有了极大的提升，但利用线特征匹配进行补充的 LAPAP 在一些线特征纹理较为丰富的区域进一步提升了配准效果。这对于解决第二类“大视差”问题有着至关重要的帮助，而良好的配准效果也使得线性融合后的图像具有更高的清晰度，对于后续局部信息的处理有着直接影响。

### 5.3.2 线特征相关网格优化约束项

3.2.4 与 3.2.5 中提出的线性对齐项与线性保持项使得线性结构在网格变形中得到进一步配准与保留。也就是说，重叠区域在进行透视变换及非重叠区域进行相似变换时，除了要考虑变换的相似性及与相邻图像的对齐，还要尽可能减少对直线结构的破坏。

图 5.18 中样例展现了网格优化变形中有无这两个约束项的最终拼接效果对比。线结构的配准与保留除了如 5.2.3 中介绍的全景图中直线数量增多及直线弯曲度减小，还可以对全景图中物体的尺寸有所纠正。



(a) 不使用线特征约束



(b) 使用线特征约束

图 5.18 线特征相关约束项效果对比样例 (左侧为全景效果, 右侧为局部放大)

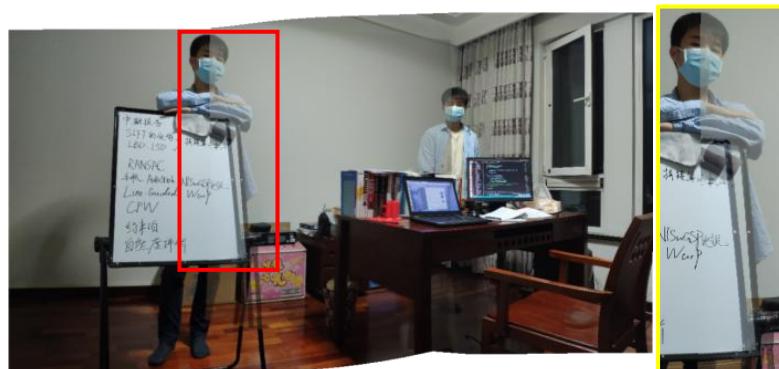
图 5.18 样例中, 框中区域的线性结构在线性约束项加入后表现的更加自然, 物体配准得到了改善, 多余重影被消除, 而图中右侧被框中放大的物体在全景图中的横向尺寸也得到了恢复。表 5.3 表示框中物体在线特征约束项作用下的尺寸与原始尺寸更加接近。

表 5.3 线特征约束项对物体尺寸的影响

原始尺寸/像素	无约束项尺寸/像素	有约束项尺寸/像素
34	48	39

### 5.3.3 图割优化

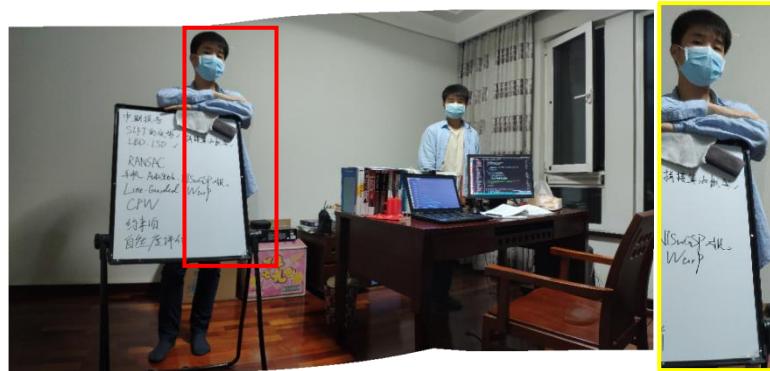
图 5.19 中样例对比体现了不同图像融合方法对拼接效果的影响。



(a) 平均融合, 过渡不自然



(b) 线性融合，白板上仍存在伪影



(c) 图割优化（最优缝合线）

图 5.19 图割优化效果对比样例 (左侧为全景效果, 右侧为局部放大)

对于室内场景这样视差较大的拼接任务, 图像中全部物体得到良好配准十分难以实现, 且室内光线明暗变化强烈, 在融合时对来自两幅图像中像素的配比要求较高才能实现最终像素强度的正常。在这种情况下, 图 5.19 (a) 中平均融合造成了明显的伪影、重影, 重叠区域的过渡也十分不自然。图 5.19 (b) 中线性融合虽然改善了来自两幅图像像素的配比, 但仍然有区域存在伪影。图 5.19 (c) 中最优缝合线作为非线性融合的方法, 不存在两幅图像像素叠加的问题, 从根本上解决了重影及伪影的问题。但由于前景存在桌子这样的物体而最优缝合线算法在搜寻分割线时没有考虑到避开前景目标, 穿过前景的分割线会显得较为明显, 对前景目标的切割也对观感有所影响。

#### 5.3.4 曝光平衡

图 5.20 中的样例对比了曝光平衡算法对拼接结果观感的影响。曝光平衡主要用以解决各待拼接子图之间的明暗差异以增强观感。当两幅相邻图像曝光水平差异明显时, 最优缝合线的分界也极为明显。



图 5.20 曝光平衡效果对比样例（左侧图像未使用曝光平衡）

图 5.20 左图中，曝光水平对观感有着强烈的影响，尤其是采用最优缝合线作为图像融合方法的拼接算法，各子图间差异较大的曝光会使得缝合线两侧变化剧烈，缝合线明显。本设计采用的曝光平衡算法从相邻图像中采样进行曝光增益计算以补偿各子图曝光。这样计算曝光增益的好处在于同时考虑了过曝区域与欠曝区域，使得最终拼接全景图中的过亮区域被压暗，而亮度不足区域被提亮，整幅图像的曝光水平适中，缝合线处的过渡也就十分自然，如图 5.20 右图所示。

### 5.3.5 先验信息

手动选取先验特征匹配信息在拼接实操中有重要意义，从重叠区域手动选择算法难以识别的特征匹配对可以补足纹理稀疏区域的配准信息，提升拼接效果。而在拼接完成后，对于拼接效果不满意的区域，包括未能正确对齐或缝合线穿越破坏前景结构的区域，可以针对性地手动选取这些区域的线特征匹配对并重新拼接，对拼接效果有立竿见影的效果。先验信息另一个重要的应用方法是，对固定安装的大视差摄像机，摄像机的抖动、焦距变化、扫描视场变化，会造成拼接方案的变化，这时利用先验匹配点，在左右摄像机中分别进行特征点匹配，然后形成共视特征点匹配关系，可以大幅度提高扰动下拼接质量。图 5.21 展示了先验信息对于拼接结果局部的影响。



(a) 全景图像



(b) 局部放大，左侧桌子腿被缝合线截断

图 5.21 先验信息效果对比样例（左侧为全景效果，右侧为局部放大）

对于本小节样例中的桌子腿处，原始的拼接效果造成最优缝合线穿过桌子腿使得此处过渡十分不自然，如图 5.21 (b) 中左图所示，桌子腿在一个地方被截断。手动选取桌子腿的线特征匹配对再重新拼接，可见缝合线避开了桌子腿，没有造成局部结构的破坏，如图 5.21 (b) 右图。深入思考其原因，这是因为手动选取的匹配对区域在后续配准时得到增强，网格变形后桌子腿轮廓更加显著且准确对齐，该轮廓在相邻图像间像素相似度明显低于背景中像素互相之间的相似度，故最优缝合线查找时轮廓处节点间权重较大，缝合线无法穿过。

#### 5.4 本章小结

目前学界对于拼接效果的定量分析描述较少，本研究用较为简单的方式衡量了拼接过程的失真及结构破坏，但仍然缺少足够的理论分析及对应的数据支撑。特别的，对于最优缝合线在图像拼接中的应用，由于该工具原本用于图像语义的分割，故其在图像拼接中的权重构造应该得到特定改进，以使分割线避开前景目标。在面临第二种“大视差”问题的场景，本方法也仍然缺少相应的解决手段使得配准信息缺失，有时会造成不正确的整体旋转及大面积拉伸变形。本设计采用的线特征匹配及筛选算法在面对纹理重复时也会形成误匹配对造成拼接失败。希望在将来的研究中可以提出进一步的分析与策略。



## 结 论

图像拼接是计算机视觉的基础任务，可以应用在场景监控、虚拟现实等多个复杂领域中。传统基于全局单应性的图像拼接方法及近年来基于网格优化的拼接算法具有不同的优势与缺陷，结合大视差问题在图像拼接过程中带来的影响，此次设计基于开源框架实现了一个完整的图像拼接软件。线特征配准及约束、图割优化、先验信息等针对性的组件被加入在端到端的图像拼接软件中，并从原理及实验结果的角度分析了这些组件对拼接效果的意义。

对于图像拼接算法中的特征提取及匹配部分，研究对比了不同点特征算法在图像拼接算法应用中的优劣，并最终选定 SIFT 特征点作为点特征匹配。而线特征作为一种高维特征被引入以增强特征匹配。点、线特征匹配通过多平面 RANSAC 得到筛选。

顺承特征提取与匹配，研究过程中结合线特征改进了 APAP 配准方法，在得到网格配准关系后，利用配准、相似性、线特征等约束解算网格优化问题，将待拼接图像的变形转化为网格顶点坐标的求解，使得各待拼接图像依据网格优化变形达到良好的拼接前效果。各待拼接图像经过网格变形后，经过曝光平衡及图像融合处理，最终得到拼接全景图。针对拼接效果中有缺陷的区域，开发实现了先验信息提取工具。整个拼接算法的软件架构及实现细节亦有所介绍。

在最终的实验对比中，首先分析了图像拼接中的大视差问题。应用所实现方法得到的拼接全景图具有良好的观感自然度，在失真抑制、图形结构保留与伪影消除等多方面相比 AutoStitch、NISwGSP 算法及商业软件有更好的效果。最后，设计了多个对比实验与分析环节。从拼接结果上定性、定量地分析说明所设计组件的实际作用，与设计时的原理相对应。

不过，所实现的图像拼接算法仍具有许多待改进之处，直线作为一种二维结构仍然无法涵盖许多更加复杂的图形结构，对待拼接图像中一些有价值的目标（如：行人）没有针对性的特征保留手段。最优缝合线的权重设计没有考虑到避开显著的前景目标，时常导致对前景结构的破坏，造成图像过渡不自然。回到图像中的特征，现有特征提取及匹配算法在提取弱纹理区域特征时无法提取到足够的特征，虽然本设计加入了线特征及先验信息，但在一些极端场景仍显乏力。特征匹配对的筛选虽实现简单但筛选效果一般。这些环节的改进都会使得目前得到的软件在拼接效果上有进一步的提升。



## 致 谢

首先，我要感谢张海老师在几个月的毕业设计过程中提供的指导。从选题到最终撰写论文，老师在专业知识上的理解以及对待工作一丝不苟的态度极大促进了我在整个毕设流程中的进步，也使得我在完成毕设工作的过程中收获颇丰。也要感谢实验室曲旭中师兄对我问题的耐心解答，得益于他在这一课题上的铺垫，我才能攻克程序编写的难题并真正理解复杂艰深的原理。

在大学四年学习生活中，有太多热爱奉献的师长与温暖贴心的同学支撑着我走到今天。感谢我的同班同学李闻达，三年同校，七年同班，十年的同窗之情使得我们之间的默契难以用言语表达。感谢 Python 病友群里两位相距万里但无话不谈的挚友，虽然我们从未有一天在群里学习过 Python，但感谢你们的陪伴让我度过孤独。很幸运，我的大学同寝是三位性格各异但勤劳质朴的好兄弟，你们的细心体贴与聪明刻苦时刻提醒着我正在身处一个何其优秀的集体。

突如其来的疫情让我无法与以上的任何一人相见，但可以想见我们正在共享怎样一种奇妙而略带酸楚的毕业情感。虽前路还尚显模糊，但祝我们各自安好。



## 参考文献

- [1] 贾彤. 基于改进RANSAC算法的室内视觉定位方法研究[D]: 哈尔滨工业大学, 2017
- [2] 关凯. 基于标识的室内视觉定位算法研究[D]: 哈尔滨工业大学, 2016
- [3] 李爽. 基于视觉的室内定位算法研究[D]: 哈尔滨工程大学, 2017
- [4] 谷雨, 周阳, 任刚, 等. 结合最佳缝合线和多分辨率融合的图像拼接[J]. 中国图象图形学报, 2017, 022(006):842-851.
- [5] 谭康. 图像拼接技术研究与实现[D]. 南京理工大学.
- [6] 蔡静. 粒计算在图像拼接中的应用研究[J]. 湖北电力, 2009, 33 (S1) :97-99.
- [7] 魏阳阳. 基于改进 AKAZE 算法的手机全景图像拼接[D].
- [8] C.D.Kuglin, D.C.Hines. The Phase Correlation Image alignment method[M]. New York: ,1975:163-165
- [9] Trajković M, Hedley M. Fast corner detection[J]. Image and Vision Computing, 1998, 16(2): 75–87.
- [10] Brown, Lowe. Recognising panoramas[A]. Proceedings Ninth IEEE International Conference on Computer Vision[C]. Nice, France: IEEE, 2003: 1218–1225 V2.
- [11] Lowe D G. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91–110.
- [12] Rosten E, Drummond T. Machine Learning for High-Speed Corner Detection[A]. A. Leonardis, H. Bischof, A. Pinz. Computer Vision – ECCV 2006[M]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, 3951: 430–443.
- [13] Brown M, Lowe D G. Automatic Panoramic Image Stitching using Invariant Features[J]. International Journal of Computer Vision, 2007, 74(1): 59–73.
- [14] Liu F, Gleicher M, Jin H, et al. Content-preserving warps for 3D video stabilization[J]. ACM Transactions on Graphics, 2009, 28(3): 1.
- [15] Hutchison D, Kanade T, Kittler J, et al. Accelerated Hypothesis Generation for Multi-structure Robust Fitting[A]. K. Daniilidis, P. Maragos, N. Paragios. Computer Vision – ECCV 2010[M]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, 6315: 533–546.



- [16] Gao J, Kim S J, Brown M S. Constructing image panoramas using dual-homography warping[A]. CVPR 2011[C]. Colorado Springs, CO, USA: IEEE, 2011: 49–56.
- [17] Lin W-Y, Liu S, Matsushita Y, et al. Smoothly varying affine stitching[A]. CVPR 2011[C]. Colorado Springs, CO, USA: IEEE, 2011: 345–352.
- [18] Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[A]. 2011 International Conference on Computer Vision[C]. Barcelona, Spain: IEEE, 2011: 2564–2571.
- [19] Wang X. Intelligent multi-camera video surveillance: A review[J]. Pattern Recognition Letters, 2013, 34(1): 3–19.
- [20] Zaragoza J, Chin T-J, Brown M S, et al. As-Projective-As-Possible Image Stitching with Moving DLT[A]. 2013 IEEE Conference on Computer Vision and Pattern Recognition[C]. Portland, OR, USA: IEEE, 2013: 2339–2346.
- [21] Chang C-H, Sato Y, Chuang Y-Y. Shape-Preserving Half-Projective Warps for Image Stitching[A]. 2014 IEEE Conference on Computer Vision and Pattern Recognition[C]. Columbus, OH, USA: IEEE, 2014: 3254–3261.
- [22] Zhang F, Liu F. Parallax-Tolerant Image Stitching[A]. 2014 IEEE Conference on Computer Vision and Pattern Recognition[C]. Columbus, OH, USA: IEEE, 2014: 3262–3269.
- [23] Lin C-C, Pankanti S U, Ramamurthy K N, et al. Adaptive as-natural-as-possible image stitching[A]. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)[C]. Boston, MA, USA: IEEE, 2015: 1155–1163.
- [24] Li S, Yuan L, Sun J, et al. Dual-Feature Warping-Based Motion Model Estimation[A]. 2015 IEEE International Conference on Computer Vision (ICCV)[C]. Santiago, Chile: IEEE, 2015: 4283–4291.
- [25] Chen Y-S, Chuang Y-Y. Natural Image Stitching with the Global Similarity Prior[A]. B. Leibe, J. Matas, N. Sebe, et al. Computer Vision – ECCV 2016[M]. Cham: Springer International Publishing, 2016, 9909: 186–201.
- [26] Lin K, Jiang N, Cheong L-F, et al. SEAGULL: Seam-Guided Local Alignment for Parallax-Tolerant Image Stitching[A]. B. Leibe, J. Matas, N. Sebe, et al. Computer



- Vision – ECCV 2016[M]. Cham: Springer International Publishing, 2016, 9907: 370–385.
- [27] Zhang G, He Y, Chen W, et al. Multi-Viewpoint Panorama Construction With Wide-Baseline Images[J]. IEEE Transactions on Image Processing, 2016, 25(7): 3099–3111.
- [28] Liu W X, Chin T-J. Correspondence Insertion for As-Projective-As-Possible Image Stitching[J]. arXiv:1608.07997 [cs], 2016.
- [29] Herrmann C, Wang C, Bowen R S, et al. Object-Centered Image Stitching[A]. V. Ferrari, M. Hebert, C. Sminchisescu, et al. Computer Vision – ECCV 2018[M]. Cham: Springer International Publishing, 2018, 11207: 846–861.
- [30] Li N, Xu Y, Wang C. Quasi-homography warps in image stitching[J]. arXiv:1701.08006 [cs], 2018.
- [31] Xiang T-Z, Xia G-S, Bai X, et al. Image Stitching by Line-guided Local Warping with Global Similarity Constraint[J]. Pattern Recognition, 2018, 83: 481–497.
- [32] Li J, Jiang P, Song S, et al. As-Aligned-As-Possible Image Stitching Based on Deviation-Corrected Warping With Global Similarity Constraints[J]. IEEE Access, 2019, 7: 156603–156611.
- [33] Wei L, Zhong Z, Lang C, et al. A survey on image and video stitching[J]. Virtual Reality & Intelligent Hardware, 2019, 1(1): 55–83.
- [34] Saha A, Maity S, Bhowmick B. Multi-modal Image Stitching with Nonlinear Optimization[A]. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)[C]. Brighton, United Kingdom: IEEE, 2019: 1987–1991.
- [35] Zheng J, Wang Y, Wang H, et al. A Novel Projective-Consistent Plane Based Image Stitching Method[J]. IEEE Transactions on Multimedia, 2019, 21(10): 2561–2575.
- [36] Liao T, Li N. Single-Perspective Warps in Natural Image Stitching[J]. IEEE Transactions on Image Processing, 2020, 29: 724–735.



- [37] Zhang Y, Lai Y-K, Zhang F-L. Content-Preserving Image Stitching with Piecewise Rectangular Boundary Constraints[J]. IEEE Transactions on Visualization and Computer Graphics, 2020: 1–1.
- [38] Fan X, Lei J, Fang Y, et al. Stereoscopic Image Stitching via Disparity-Constrained Warping and Blending[J]. IEEE Transactions on Multimedia, 2020, 22(3): 655–665.
- [39] Bian J, Lin W-Y, Matsushita Y, et al. GMS: Grid-based Motion Statistics for Fast, Ultra-Robust Feature Correspondence[J]. : 10.
- [40] Gao J, Li Y, Chin T-J, et al. Seam-Driven Image Stitching[J]. : 4.
- [41] Kwatra V, Schodl A, Essa I. Graphcut Textures: Image and Video Synthesis Using Graph Cuts[J]. : 10.
- [42] Xu W, Mulligan J. Performance evaluation of color correction approaches for automatic multi-view image and video stitching[A]. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition[C]. San Francisco, CA, USA: IEEE, 2010: 263–270.



## 附录 A 图像读取与文件解析程序片段

### 1 无法以字典序读入文件的原始程序

```
if((dir = opendir (dir_name.c_str())) != NULL) {  
    while((ent = readdir (dir)) != NULL) {  
        string file = string(ent->d_name);  
        for(int i = 0; i < image_formats.size(); ++i) {  
            if(file.length() > image_formats[i].length() &&  
                image_formats[i].compare(file.substr(file.length() - image_formats[i].length(),  
                image_formats[i].length()) == 0) {  
                result.emplace_back(file);  
            }  
        }  
    }  
    closedir(dir);  
}
```

### 2 以字典序读入文件

```
struct dirent **namelist;  
vector<string> result;  
int n = scandir(dir_name.c_str(), &namelist, fileNameFilter, alphasort);  
if(n < 0){  
    printError("F(getImageFileFullNamesInDir) could not open directory");  
    return result;  
}  
for(int i = 0; i < n; i++){  
    string filePath(namelist[i]->d_name);  
    result.emplace_back(filePath);  
    free(namelist[i]);  
}  
free(namelist);
```



## 附录 B 线特征及多平面 RANSAC 相关说明程序

### 1 线特征信息转换

```
for(int i = 0; i < line_count; ++i) {
    lines_points[0].emplace_back(keylines[i].getStartPoint().x, keylines[i].getStartPoint().y);
    lines_points[1].emplace_back(keylines[i].getEndPoint().x, keylines[i].getEndPoint().y);
    //lines_length.emplace_back(norm(lines_points[1][i] - lines_points[0][i]));
    lines_length.emplace_back(double(keylines[i].lineLength));
    lines_width.emplace_back(double(keylines[i].size));
}
```

### 2 Sequential RANSAC 实现程序

```
while(tmp_X.size() >= HOMOGRAPHY_MODEL_MIN_POINTS &&
      parameter.local_homogrpahy_max_inliers_dist <
      parameter.global_homography_max_inliers_dist) {
    const int LOCAL_MAX_ITERATION = log(1 - OPENCV_DEFAULT_CONFIDENCE)
        / log(1 - pow(LOCAL_TRUE_PROBABILITY,
        HOMOGRAPHY_MODEL_MIN_POINTS));
    vector<Point2> next_X, next_Y;
    vector<char> mask(tmp_X.size(), 0);
    findHomography(tmp_X, tmp_Y, CV_RANSAC,
    parameter.local_homogrpahy_max_inliers_dist, mask, LOCAL_MAX_ITERATION);

    int inliers_count = 0;
    for(int i = 0; i < mask.size(); ++i) {
        if(mask[i]) { ++inliers_count; }
    }
    if(inliers_count < parameter.local_homography_min_features_count) {
        break;
    }
    for(int i = 0, shift = -1; i < mask.size(); ++i) {
        if(mask[i]) {
            final_mask[mask_indices[i]] = 1;
        } else {
            next_X.emplace_back(tmp_X[i]);
            next_Y.emplace_back(tmp_Y[i]);
            mask_indices[++shift] = mask_indices[i];
        }
    }
#ifndef NDEBUG
    cout << "Local true Probabiltiy = " << next_X.size() / (float)tmp_X.size() << endl;

```



```
#endif  
tmp_X = next_X;  
tmp_Y = next_Y;  
}
```

---