

摘要

在我们所处的这个客观世界，所有的实体对象除了尺寸、形状、颜色、纹理，还有空间位置这些基本属性。而随着社会的发展，在很多领域仅仅用二维图像来描述实体对象的颜色、纹理以及二维平面上的空间位置已经不能满足实际需求，此时就需要对实体对象进行三维重建，来更好地描述其在客观世界的基本属性。

本文的研究根据 SFM (Structure from Motion) ^[1] 算法原理产生稀疏点云，在此过程中使用 SIFT (Scale-invariant feature transform) ^[2] 算法进行特征点的查找和匹配，根据三角原理进行位置标定；然后根据 CMVS (Clustering Views for Multi-view Stereo) 和 PMVS (Patch-based Multi-view Stereo Software) ^[3] 产生稠密点云，并进行颜色和纹理的重建；而在对大型建筑的重建中，往往会存在一次的图片样本采集不完全的情况，这就需要使用 ICP (Iterative Closest Point) ^{[4][5]} 算法对不同的点云进行拼接。在实现功能时，研究使用了 openCV, openMVG 以及 PCL (Point Cloud Library) 三个库，使用 VS2012 进行程序功能的开发，使用 Qt 进行图形界面的开发。

研究进行了 SIFT、SURF (Speeded Up Robust Features)、Harris 角点以及 FAST (Features from Accelerated Segment Test) ^[6] 四种特征点检测算法的测试；在进行稀疏点云重建时，对 Incremental SFM 和 Global SFM 两种 SFM 算法进行了测试；在进行稠密重建时，将 CMVS 与 PMVS 结合使用；实现点云拼接功能时，对经典 ICP、Levenberg-Marquardt 优化的 ICP 以及经典 ICP 的变种 Generalized ICP 进行了性能和结果的测试。最后使用 Qt 将比较好的功能模块进行组合，实现一个具有图形操作界面、功能完整的程序。

关键词： 三维重建，点云拼接，特征点检测，sfm, icp

Abstract

In the real world we live in, all the physical objects have size, shape, color, texture, and spatial location these basic attributes. With the development of society, in many areas we only use two-dimensional image to describe the physical objects' color, texture and the spatial position in two-dimensional plane which can not meet the actual demand. Then we need a three-dimensional reconstruction of the entity object to describe the basic properties of the objective world better.

This study produced a sparse point cloud according to SFM (Structure from Motion) algorithm principle, using SIFT (Scale-invariant feature transform) algorithm in the process to find and match feature points position, and calibration based on the principle of triangulation. Then generate dense point cloud according to CMVS (Clustering Views for Multi-view Stereo) and PMVS (Patch-based Multi-view Stereo Software), and rebuild the color and texture. In the reconstruction of large buildings, often there will be the case that the collection of sample is not entirely at one time, which requires the use of ICP (Iterative Closest Point) algorithm for splicing different point cloud. When carrying out the function, using openCV, openMVG and PCL (Point Cloud Library) three libraries, using VS2012 development program functions and using Qt to develop a graphical interface.

In the study we tested SIFT, SURF ((Speeded Up Robust Features), Harris corner and FAST (Features from Accelerated Segment Test) four kinds of feature detection algorithm. When making sparse point cloud reconstruction, we tested two kinds of SFM algorithms -- Incremental SFM and Global SFM. During dense reconstruction, PMVS used in conjunction with CMVS. When stitching point cloud, we tested classic ICP, Levenberg-Marquardt optimization of ICP and Generalized ICP. Finally, using Qt to combine function modules to form a full-featured software.

Keywords: 3d reconstruction, merge point could, collection of features, sfm, icp

目录

摘要.....	1
ABSTRACT	2
第一章 绪论.....	6
1.1 引言	6
1.2 研究现状	7
1.3 研究内容和意义	10
1.4 本文组织结构	11
1.5 本文小结	12
第二章 重建算法概要	13
2.1 特征点检测匹配	13
2.1.1 Harris 角点	13
2.1.2 FAST 算法.....	15
2.1.3 SIFT 算法.....	17
2.1.4 SURF 算法.....	19
2.2 三维重建	21
2.2.1 SFM 稀疏重建	21
2.2.2 基于面片稠密重建	22
2.3 本章小结	23
第三章 多视图稠密三维重建.....	24
3.1 稀疏重建	24
3.1.1 增量式 SFM.....	24
3.1.2 全局式 SFM.....	27
3.2 稠密重建	30
3.2.1 pmvs 算法.....	30
3.2.2 cmvs 算法.....	32
3.3 本章小结	34
第四章 图像拼接.....	35
4.1 二维图像拼接	35
4.2 三维拼接算法.....	37
4.2.1 经典 ICP 算法	37
4.2.2 Generalized ICP 算法	39
4.2.3 Levenberg-Marquardt icp 优化算法.....	40
4.3 本章小结	41
第五章 总结与展望.....	43
5.1 总结	43

5.2 展望	44
参考文献.....	45

图目录

图 2-1 检测原图.....	14
图 2-2 HARRIS 角点结果图	15
图 2-3 FAST-N 比较示意图 ^[1]	16
图 2-4 FAST 算法结果图	16
图 2-5 图像金字塔 ^[2]	17
图 2-6 极值点查找 ^[2]	18
图 2-7 SIFT 算法结果.....	19
图 2-8 图像金字塔	20
图 2-9 SURF 描述子构建	21
图 2-10 SURF 算法结果图	21
图 2-11 SFM 原理 ^[1]	22
图 3-1 位置关系示意图 ^[1]	24
图 3-2 重建原图	26
图 3-3 左侧观测点云	27
图 3-4 右侧观测点云	27
图 3-5 位置关系示意图 ^[1]	28
图 3-6 左侧观测点云	30
图 3-7 右侧观测点云	30
图 3-8 连通性聚类效果图	32
图 3-9 质心聚类效果图.....	33
图 3-10 分布聚类效果图	33
图 3-11 左侧观测图.....	34
图 3-12 右侧观测图	34
图 4-1 原图一.....	36
图 4-2 原图二.....	36
图 4-3 拼接结果.....	37
图 4-4 左侧观测图	38
图 4-5 右侧观测图	39
图 4-6 左侧观测图	41
图 4-7 右侧观测图	41
图 5-1 软件界面	43
图 5-2 重建流程	44
图 5-3 拼接流程	44

第一章 绪论

1.1 引言

真实的客观世界是三维的，一个客观物体除了有其相应的形状、尺寸、颜色及纹理信息^[7]外，还有物体内部各部分之间的相对位置关系。如果我们希望更具体地描述一个物体，除了描述二维图像上的形状、尺寸、颜色及纹理，我们还需要描述一个物体的三维坐标信息，即一个场景下各个物体的深度信息，这就需要根据一定的视觉几何原理和算法从我们已拥有的所有二维图像中的提取有用的三维坐标或深度信息。

在自然界中人类及其他有视觉系统的生物观测到的客观世界只是一个二维图像，但通过视觉系统的转换可以恢复所观测对象的三维信息，例如许多生物都会通过头部和眼球的转动使观测的对象产生一个视觉差，根据这个视觉差恢复观测对象的深度信息^[8]。计算机视觉作为模拟人类视觉系统，使计算机拥有类似人类感知客观世界的学科，实现计算机从二维图像中恢复观测对象的三维信息是该学科研究的重要组成部分；而作为计算机视觉的研究部分，基于视觉的三维重建技术最大的特点在于采用非接触式的测量方式来获取三维信息：利用普通的单目数码相机获取目标对象的二维图像，通过计算机对所获得的图像进行处理以及基于视觉几何原理的算法对图像数据的计算，来获得目标对象比较准确的三维信息。

因为基于视觉的三维重建不用对观测对象进行接触式地测量，也不用按比例模型建造，因此重建过程不会受重建对象的形状、大小以及所处空间位置的限制，对于大型观测对象（比如大型建筑、山峰、瀑布）的重建也将更简单、更方便。

随着社会进步与生产力的发展，越来越多的领域需要三维重建这一技术的辅助，且这一技术也正在进入更多的领域：文化艺术数字化保存、医学研究中的全膝置换手术、破损头骨的复原、重建下颌骨以及假肢的扫描重构等方面；在逆向工程中，对全体物件的实物模型或零件进行数字化测量并在此基础上建立CAD(Computer Aided Design)模型等；另外在航空航天、建筑测绘、人工关节模

型的建立等领域也有广泛的应用。而基于计算机视觉的三维重建因为其快速、方便以及较低的重建成本，将在上述领域发挥越来越重要的作用。

1.2 研究现状

20 世纪的 60 年代，已经有相关的专家和学者对视觉系统中的立体视觉的各种原理属性开始了各类研究。而在 20 世纪 80 年代初，David Marr 提出了提出的计算机视觉理论将视觉系统大致分为三个过程：在第一个过程为获取图像中包括角点、边缘等特征信息；第二个过程则是在在获取特征信息的基础上根据几何原理进行计算，获得图像中观测对象的三维立体信息；而第三个过程则是在第二个过程的基础上进行补充和优化，对图像中的场景建立一个相对的坐标系，较为准确地描述出观测对象相对于场景中的三维立体信息，并对观测物体的表面进行颜色、纹理的填充与优化。这个理论框架对人们认识和研究计算机视觉系统提供了优秀的理论指导，在很大程度上减少了后来的专家、学者在研究过程中的错误。

在 Marr 的计算机视觉框架下，三维重建的过程主要可分为六部分：图像获取，相机标定，特征提取，立体匹配，确定观测对象相对三维坐标以及后处理阶段^[9]。

- 1、图像获取。这一步骤是三维重建中基础的工作，后续工作都需要以此为前提。除去激光扫描等专业的获取观测对象深度信息的仪器，获得观测对象图像的工具一般为单目相机和多目相机，而以单目相机更为普遍。虽然在三维重建中图像获取是最为简单的步骤，但仍需要根据使用的算法，获取光照、角度、前背景、纹理等符合算法要求的图像。
- 2、相机标定。相机标定即确定某一图像对应的相机位置以及相机的光圈、感光元件大小等相机属性，以此来确定立体空间中观测对象点同它的像点之间的对应关系。
- 3、特征提取。特征是指图像中的像素、像素集合或者前两者的抽象表达。作为计算机视觉的重要概念，特征在观测对象的分割和跟踪、观测对象的匹配等方面具有重要作用。而在多视图的三维重建和点云的拼接中特征的提取更是极其重要的步骤。
- 4、立体匹配。立体匹配即对提取的特征点在三维空间上进行匹配，相对与

二维平面的特征点匹配，立体匹配存在空间位置的匹配。在多视图的三维重建中立体匹配是恢复观测对象三维信息的关键步骤，匹配的结果将直接影响观测对象重建结果的准确程度；而因为图像与相机的关系，此过程将会存在基础矩阵估计的步骤，用来表示相机间的旋转平移关系。而且立体匹配也是点云拼接的关键步骤，匹配过程将影响两个点云拼接程度。

- 5、确定相对坐标。完成立体匹配之后，就能确定相机与特征点之间、特征点与特征点之间的相对位置关系，以此来建立坐标系就可以相对准确的回复观测对象的立体视觉信息。
- 6、后处理。最后这一步骤主要着重于观测对象的表面优化，对点云进行稠密重建，添加观测物体的颜色和纹理信息等。

以上只是三维重建的一般步骤，最为适用于多视图三位重建，但对于像激光扫描等专业的三维重建仪器一样适用，只是专业仪器在获取图像时直接获取了观测物体的深度信息，并直接完成了特征点的查找与立体匹配没有如此细分。

依托于上述的六个步骤，众多专家与学者根据人类视觉特征提出了在特定环境下具有良好效果过的三维重建算法：

- 1、明暗度(shape from shading SFS): 由 Horn^[10]在 1970 年首次提出，而 Vogel 在 2008 年提出了非朗伯特的 SFS 模型，对原有的模型进行了改进。SFS 算法基于：a、反射模型为朗伯特模型，即从各个角度观察，同一点的明暗度都相同的；b、光源为无限远处点光源；c、成像关系为正交投影，这三个假设；通过分析图像中的明暗度信息，运用反射光照模型，恢复出物体表面法向量信息进行三维重建。
- 2、光度立体视觉 (photometric stereo): 由 Woodham^[11]在 1980 年对 SFS 进行改进后提出，并且由 Noakes、Sun、Shi 等人分别对算法进行了改进和优化。这一算法的主要原理是通过多个不共线的光源获得物体的多幅图像，再将不同图像的亮度方程联立，求解出物体表面法向量的方向，最终实现三维重建。
- 3、纹理法 (shape from texture SFT): 由 WIKTIN 在 1987 年首次提出，之后 Warren 和 Liboy 等学者分别在这一算法基础上进行了发展和创新。这

种算法根据当物体被投影在二维图像上时，物体上的纹理单元会发生投影变形和透视收缩变形这一原理，分析变形后的纹理单元，反向求取物体表面法向和深度信息，进行三维重建。

- 4、轮廓法(shape from silhouettes): 由 Martin 在 1983 年提出, 并且由 Szeliski、Snow 和 Landabso 等研究者分别对原算法进行了改良, 提高了建模速度。这种方法将物体所在的三维空间离散化为体素, 再将图像分割为前景(物体) 和背景, 通过投影测试判定一个体素点的投影是在前景还是背景中。如果一个体素点在不同视角都投影到了前景内, 那么就认为它是物体的一部分, 将所有这样的体素点集合起来就可以重构出物体的三维形状。
- 5、调焦法(shape from focus): 由 Rajagopalan 于 1997 年率先提出, Pradeep、Sahay 和 Thelen 等研究者在此算法的基础上分别进行了优化。这种算法的原理是: 物体只有位于摄像机焦距处时才会投影出清晰的图像, 因此通过建立物体到投影中心的距离与图像清晰度之间的关系, 就可以恢复出深度信息, 之后别刻意通过深度信息进行三维重建。
- 6、运动法(structure from motion SFM): 运动法基于特征点的企图于查找。由 Harris 在 1988 年提出 Harris 角点, 之后在特征点匹配上分别发展出了 SIFT、PCA-SIFT、GIOH 和 SURF 等一系列优秀的匹配算法。SFM 的主要原理通过多幅未标定图像中检测匹配特征点集, 使用数值方法恢复摄像机参数与三维信息, 然后通过相机参数和三维信息来实现三维重建。

在三维重建的应用方面, 人脸三维模型的重建因为人脸含有大量的低纹理区和相似纹理区成为一个比较困难的问题, 因此常采用多种方法综合使用的策略^{[12][13]}的。而这种多种方法综合使用的策略, 也被更多的使用: 在用未标定图像进行三维匹配时就可以先引入限制因子来消除 Harris 角点聚簇的现象, 使用高斯曲面拟合内插使 Harris 角点达到亚像素级; 接着采用特征点的 Sift 特征描述符进行初始匹配, 利用随机抽样算法估计基础矩阵的同时剔除误匹配点对; 最后在估计的基础矩阵的引导下进行双向匹配^[14]。将多种算法进行组合能弥补各自算法的不足, 产生更好的结果。

而对于多种方法综合使用的集成者, 对应各种语言代码库进一步促进了三维重建的研究和应用。作为计算机视觉中一个重要的开源库 openCV 对三维重建方

面也提供了相应的方法；通过对 openCV 中对应代码和相应文档的研究^[15]，发现 openCV 采用的方法为介于标定和自标定的一种方法：它通过在至少 3 个不同的位置获取标定物的图像，计算出摄像机所有的内外参数^[16]。而另一个开源库 PCL 测试完全用于三维图像的重建和处理。

1.3 研究内容和意义

在上一小节已经概述了三维重建的步骤为图像获取、相机标定、特征提取、立体匹配、确定观测对象相对三维坐标以及后处理，而特征提取、立体匹配以及相对坐标的确定作为关键步骤，是众多三维重建算法的研究重点；而为了直观呈现重建的结果，稠密重建和重建表面的颜色、纹理优化也必不可少。

特征点的检测与匹配作为计算机视觉的基础算法，很早就开始了相关研究。在 20 世纪 80 年代，Harris^[17]等人提出了角点的概念，并且提出了基于图像灰度的方法通过计算点的曲率及梯度来检测角点的 Harris 算法；而目前使用比较广泛的特征点提取和匹配方法是由 Lowe 提出的 SIFT (Scale-invariant feature transform) 算法；借鉴 SIFT 算法的思想，Bay 提出了 SURF ((Speeded Up Robust Features) 算法^[18]。多种优秀的特征提取匹配算法为三维重建这样复杂的计算机视觉技术提供了基础算法理论。

有了以上特征提取匹配算法的支撑，无需专业的图像获取设备，并省去了多目相机对各个相机的之间位置、焦距的调整测试，SFM 作为多视角三维重建中的重要算法在图像获取阶段显示出了巨大优势。SFM 算法在进行特征点匹配时使用 SIFT 算法，保证特征具有尺度不变性，在改变旋转角度、图像亮度和拍摄角度的情况下仍然具有很好的特征点提取效果，而且可以消除特征点对图像灰度非线性变换敏感的问题^[19]，由此可以保证在室外复杂光照条件下对大型观测对象（例如建筑物等）特征点提取的准确性。在进行立体匹配时对用于计算相机平移和旋转的基础矩阵进行稳定性检测，以此来保证立体匹配的准确。

然而 SFM 算法一般只产生稀疏的点云，如要更直观的展现三维重建的结果，就需要对 SFM 产生的点云进行稠密重建，此时就需要用到 CMVS/PMVS 算法。PMVS 是基于 patch（补丁）的点云重建，对于 SFM 输出的稀疏点云通过匹配、扩张、过滤的过程完成稠密重建，输出稠密点云。而 CMVS 是对 PMVS 的一个

优化算法。

以上的三个算法已经可以解决从特征点的提取到观测对象表面重建优化的一系列问题，但对于室外大型观测对象的重建仍然存在许多实际问题，由于观测对象体积大小和几何形状的影响，以及观测方法的限制，无法一次性获得观测对象的全貌便是其中一个。在二维图像中，由于相机成像的限制，在只能获得大型观测对象一部分的图像的情况下，首先想到的就是对多张图像进行拼接以此来获取观测对象的全貌，为此边有学者使用 Harris 算法提取角点，再通过特征点匹配的图像自动拼接算法^[20]。图像配准是影响图像拼接结果的主要因素，当前有根据拼接图像灰度值来判断图像重叠部分并进行图像配准的区域相关拼接算法，以及查找图像特征点根据特征点匹配进行图像配准的特征相关拼接算法。

类比二维图像，在三维重建中如果要得到观测对象的全貌就需要对多个三维重建模型进行拼接。对于三维图像的拼接主要为对同一观测对象的不同三维点云进行拼接，将所有的点云数据集统一到全局坐标系下，以此尽可能还原观测对象的三维图像的全貌，解决观测对象的三维图像部分缺失的问题，达到对三维重建的优化的目的。点云的拼接究其根本仍然是特征的匹配与合并，而其本质便是将不同角度观测所得到的点云进行空间坐标的平移与旋转变换，在统一到全局坐标系后进行重叠坐标的合并。当前主流的精确点云拼接算法为 ICP 算法，经过多年的发展，ICP 算法产生了多种变种和优化算法，本研究将对比一些 ICP 算法，进行一些分析，选取合适的算法整合到三维重建的系统中。

研究的最终目标是整合 SFM 算法和 ICP 算法，形成一个重图像输入到三维重建结果输出、从点云输入到点云拼接结果输出并且带有图形操作界面的一个小系统，使三维重建和点云拼接更系统、更方便。

1.4 本文组织结构

本文共分为五章，以计算机视觉中的重要分支多视角的三维重建为主题，了解了三维重建的发展现状，理清了三维重建和点云拼接的基本原理和相关步骤，阅读了基本算法的相关资料并对这些算法进行了试验分析，各章的详细内容描述如下：

第一章：介绍课题研究的背景，国内外相关领域的研究现状及应用场景，课

题研究的主要任务和本文的主要工作。

第二章：测试了作为计算机视觉基础算法的各类特征点检测匹配算法，之后对比了增量 SFM 算法和全局 SFM 算法。

第三章：对稠密重建进行原理描述，在第二章所得结果的基础上，应用 PMVS 算法进行稠密重建，对重建表面进行颜色、纹理的重建，优化直观显示结果。

第四章：叙述三维点云重建的 ICP 算法的原理，测试三种 ICP 算法的拼接效果。

第五章：对研究产生的结果进行分析，并根据可优化的步骤进行下一步的工作计划。

1.5 本文小结

本章简要介绍项目的研究背景、在国内外相关领域的研究现状和应用场景以及项目的研究的任务和意义。最后，给出了本文的主要工作及本文的组织结构。

第二章 重建算法概要

2.1 特征点检测匹配

2.1.1 Harris 角点

角点是指二维图像中亮度变化剧烈的点或图像边缘曲线上曲率极大值的点，一般由图像中一些几何结构的关节点构成，很多都是线条之间产生的交点。角点的检测主要有基于图像边缘的方法和基于图像灰度的方法两类。而由于基于图像边缘的检测方法对于图像的分割和边缘的提取有很大的依赖，如果观测对象发生局部变化很容易导致检测失败，所以很多角点检测会采用图像灰度检测，Harris 角点^[20]便是基于图像灰度检测的一种方法。

Harris 角点是一类比较经典的角点类型，它的基本原理是计算图像中每点与周围点变化率的平均值。对于给定的检测点 (x, y) 以及位移 $(\Delta x, \Delta y)$ ，则其自相关的表达式为：

$$c(x, y) = \sum_w [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \quad (2-1)$$

其中 I 为图片的数学表达， (x_i, y_i) 表示以 (x, y) 中心的检测窗口 W 中的点。

而变换后的图片可以用泰勒一次展开近似表示为：

$$I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i) I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2-2)$$

其中 $I_x(x_i, y_i)$ 和 $I_y(x_i, y_i)$ 分别表示对 x 和 y 的偏导，最后将(2-2)式代入(2-1)式可得结果：

$$c(x, y) = \sum_w \left([I_x(x_i, y_i) I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 = [\Delta x, \Delta y] C(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2-3)$$

其中矩阵 $C(x, y)$ 表示检测区域中特征比较强烈的区域。

假设 α_1 和 α_2 为矩阵 $C(x, y)$ 的特征值，如果 α_1 和 α_2 都比较小则表示矩阵 $C(x, y)$

所表示的区域像素平滑过渡；如果特征值一个大一个小表示检测区域存在一个明显的边缘；如果两个特征值都比较大，则表示检测区域存在一个明显的折点。根据上述的数学模型以及相关原理，实现 Harris 角点检测结果如图 2-1 和 2-2 所示。



图 2 -1 检测原图



图 2 -2 Harris 角点结果图

可以看到图像中物体的边缘与折点都有很好的识别，将建筑物作为观测对象时，所有角点的集合可以组成建筑物的轮廓，但丢失了很多细节特征。

2.1.2 FAST 算法

很多传统的算法都很耗时，而且特征点检测算法只是很多复杂图像处理里的第一步，得不偿失。FAST 特征点检测是公认的比较快速的特征点检测方法，只利用周围像素比较的信息就可以得到特征点，简单，有效。

FAST 特征检测算法来源于 **corner** 的定义，这个定义基于特征点周围的图像灰度值，检测候选特征点周围一圈的像素值，如果候选点周围领域内有足够多的像素点与该候选点的灰度值差别够大，则认为该候选点为一个特征点。

为了获得更快的结果，还采用了额外的加速办法。如果测试了候选点周围每隔 90 度角的 4 个点，应该至少有 3 个和候选点的灰度值差足够大，否则就不用再计算其他点，而直接认为该候选点不是特征点。候选点周围的圆的选取半径是一个很重要的参数，为了简单高效，可以采用半径为 3，共有 16 个周边像素需要比较。为了提高比较的效率，通常只使用 N 个周边像素来比较，也就是大家

经常说的 FAST-N。如图 2-3 所示：

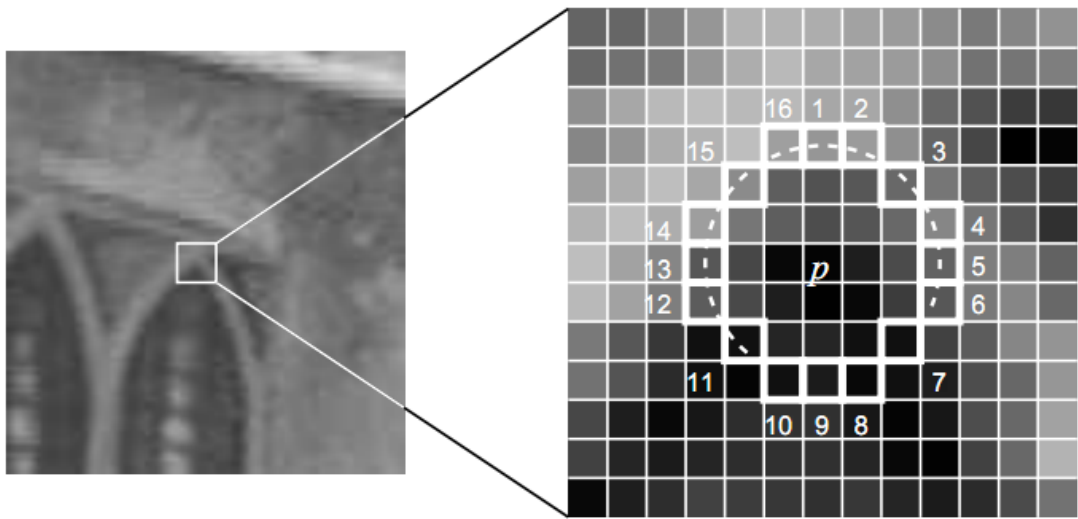


图 2 -3 FAST-N 比较示意图^[1]

根据上述原理描述，实现 fast 算法的结果如图 2-4 所示，可对比图 2-1：



图 2 -4 fast 算法结果图

由结果图可知 fast 算法效果良好，检测出了很多的特征点。

2.1.3 SIFT 算法

SIFT 是一种局部特征检测算法，所提取的特征具有尺度和旋转不变性，也不会受光照和拍摄角度的影响。SIFT 算法首先需要构造一个尺度空间，以此来模拟图像数据的多尺度特征。利用高斯卷积核将二维图像的尺度空间定义为：

$$L(x,y,\sigma) = G(x,y,\sigma) \times I(x,y) \quad (2-4)$$

其中 $G(x,y,\sigma)$ 为尺度可变函数 $G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$ ， (x,y) 为空间坐标，

σ 决定图像的平滑程度， σ 值越大对应的图像越粗糙，即分辨率越低。

而为了在尺度空间提取稳定的关键点，又提出了高斯差分尺度空间（DOG scale-space）。

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (2-5)$$

并由此可建立图像金字塔如图 2-5 所示。

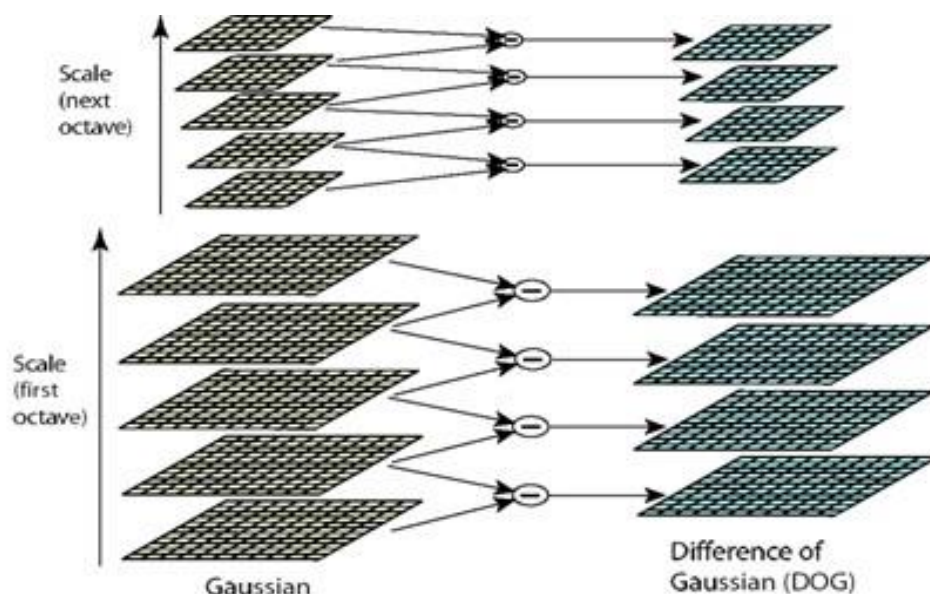


图 2-5 图像金字塔^[2]

在完成尺度空间的构建后，就进行关键点的查找，而各尺度关键点即为尺度空间的极值点，每一个采样点要和它所有的相邻点比较，看其是否比它的图像域和尺度域的相邻点大或者小。如图 2-6 所示，中间的检测点和它同尺度的 8 个相邻点和上下相邻尺度对应的 9×2 个点共 26 个点比较，以确保在尺度空间和二维

图像空间都检测到极值点。

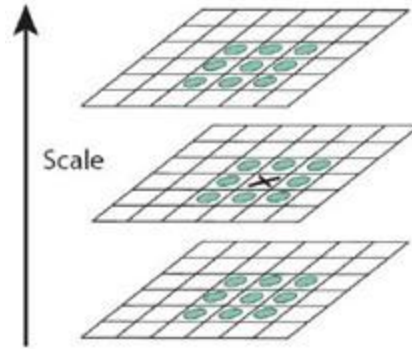


图 2 -6 极值点查找^[2]

至此得到了二维图像的特征点，但因为噪音的存在，某些特征点并不能使用，这就需要对错误特征点的剔除。之后为每个剩余的特征点计算一个方向，依照这个方向做进一步的计算，利用关键点邻域像素的梯度方向分布特性为每个关键点指定方向参数，使算子具备旋转不变性。

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2-6)$$

$$\theta(x,y) = a \tan 2 \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \quad (2-7)$$

由 $m(x,y)$ 和 $\theta(x,y)$ 可以计算出点 (x,y) 的梯度的模长和方向，其中 L 表示关键点各自所在的尺度空间。至此便可以确定关键点的位置、方向以及所处尺度。

在确定特征点的主方向后，并根据特征点周围领域的方向产生 128 维的特征点的描述子用于特征点的匹配等计算。运用 SIFT 算法产生的特征点匹配结果如图 2-7 所示，可对于比图 2-1：



图 2 -7 sift 算法结果

对比之前的两个算法，sift 算法所获得的特征更多，相应的观测物体的细节描述也更为丰富，但测试时所用的时间也明显多余其余两种算法。

2.1.4 SURF 算法

SURF 特征是类似于 SIFT 特征的一种尺度不变的特征点，而为了达到尺度不变性，SURF 算法需要构造一个尺度空间，形成一个图像金字塔（见图 2-8）。不同于 SIFT 算法，SURF 采用的是 Hessian 矩阵行列式近似值图像，由此可以形成一个图像金字塔。图像金字塔可以分为很多层，每一层被称为 octave，每个 octave 中又有几张尺度不同的图片。在 SURF 中，图片的大小是一直不变的，不同的 octave 得到的待检测图片是改变高斯模糊尺寸大小得到的，而同一个 octave 中个的图片用到的高斯模板尺度也不同。SURF 采用这种方法节省降采样过程，以此提高了算法的处理速度。

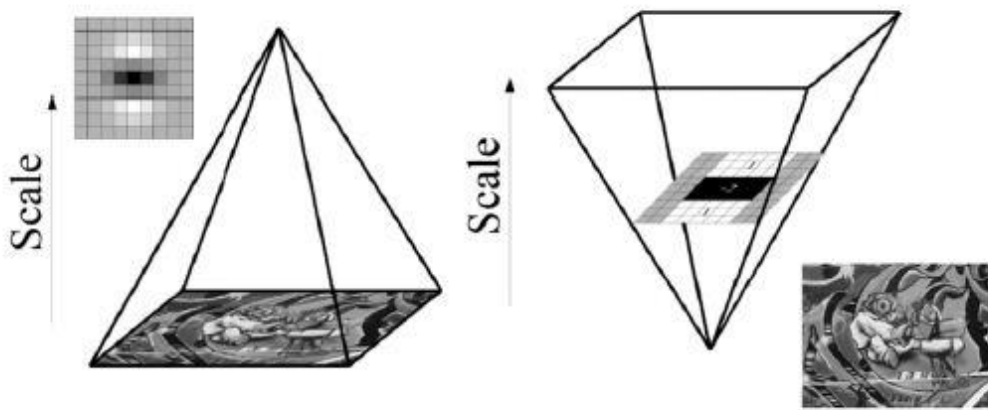


图 2 -8 图像金字塔

在得到图像的尺度空间后，与 SIFT 类似将中间的检测点和它同尺度的 8 个相邻点和上下相邻尺度对应的 9×2 个点共 26 个点比较，将极值点作为特征点保存下来。之后统计特征点领域内的 Harr 小波特征，来确定特征点的主方向，以保证特征点的旋转不变性，这与 SIFT 算法有很大不同。

在确定主方向后，以特征点为中心选定一个方形领域，以特征点主方向为方形领域方向；然后把该领域分为 16 个子区域，每个子区域统计 25 个像素的水平方向和垂直方向的 Haar 小波特征（这里的水平和垂直方向都是相对主方向而言）。该 Haar 小波特征为水平方向值之和，水平方向绝对值之和，垂直方向之和，垂直方向绝对值之和。该过程如图 2-9 所示。这样每个小区域就有 4 个值，所以每个特征点描述子就是 $16 \times 4 = 64$ 维的向量，相比 sift 而言，少了一半，而这大大加快了特征匹配速度。

SURF 的测试结果如图 2-10 所示，特征点的密集程度超过了 FAST 算法和 Harris 角点的检测，而相对 SIFT 的特征点比较少，但其运算速度要快于 SIFT 算法。

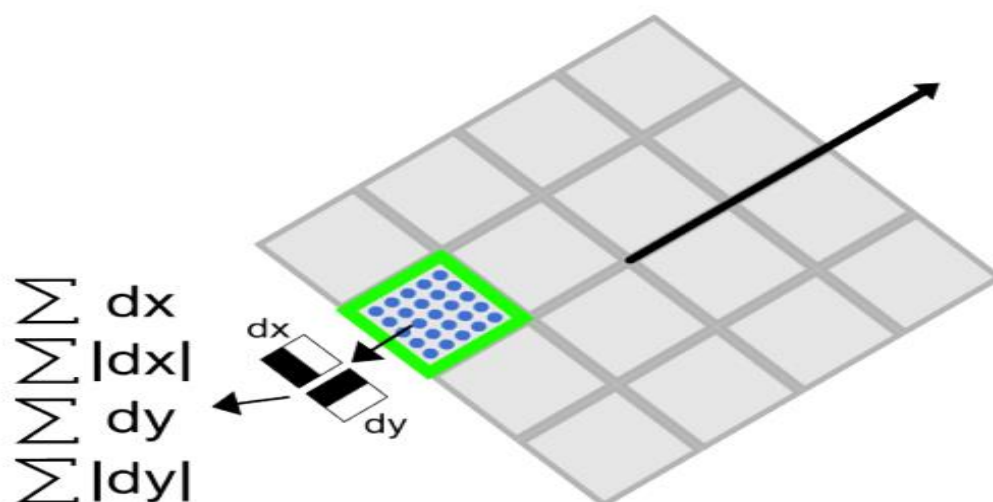


图 2 -9 SURF 描述子构建



图 2 -10 surf 算法结果图

2.2 三维重建

2.2.1 SFM 稀疏重建

SFM 即基于运动的构建，是多视图三维重建中的重要算法。SFM 的基本思想为从一组无序的图集中或者一个视频文件中提取特征点集，通过特征点之间的

匹配，跟踪特征点的运动轨迹，在获得相机内部参数的条件下根据特征点匹配关系和三角原理可以恢复相机的位置信息和观测对象的三维信息，进而得到场景的稀疏重建，如图 2-11 所示。

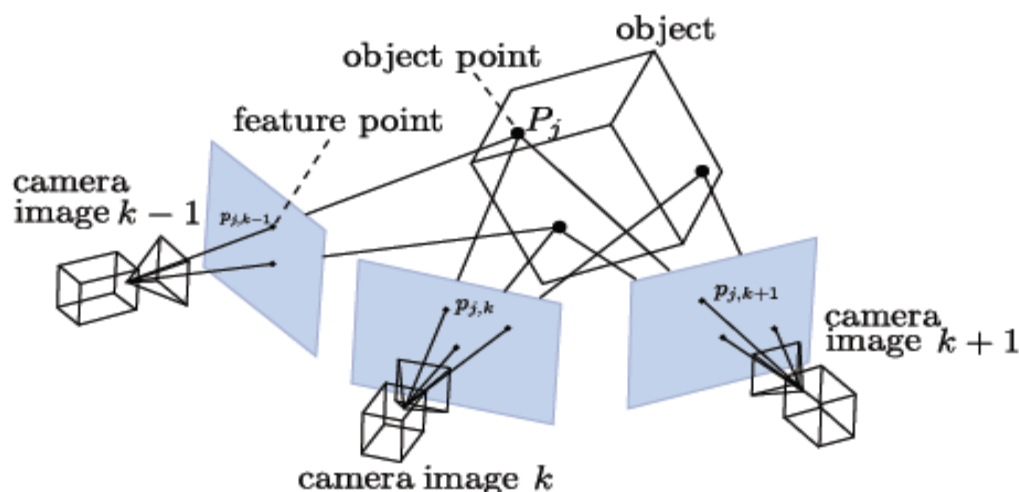


图 2-11 SFM 原理^[1]

SFM 算法最先需要使用特征点的提取，上述的四种特征点提取算法是当前常用的算法，而在非实时的重建系统中，对于重建的算法并没有很苛刻的效率要求，所以一般会使用准确性比较高的 SIFT 算法。

在确定特征点提取算法后，SFM 所需解决的问题是特征点的匹配和跟踪，这使得现实世界中的一个点或多幅图像中的呈现保持一致性，进而确定图像与图像之间的位置角度关系。常用的匹配算法有 NNS（Nearest neighbor search，即最近临近查找）和 K-NN（K-Nearest Neighbor，即 K 临近查找），NNS 适合两两成对地进行匹配，而 K-NN 则适合多个特征点进行匹配。

在完成匹配跟踪后，便需要标定相机位置和观测对象的三维信息，当前通常采用 BA（Bundle Adjustment，即集束约束法）算法，根据所有点在图像中的投影作为标准，同时提炼出描述场景结构的 3D 点坐标、相对运动参数和相机的光学参数。之后便能完成稀疏重建。

2.2.2 基于面片稠密重建

完成稀疏重建后可以使用稠密重建的相关算法对之前的结果进行优化，Y. Furukawa 提出了一种基于面片的多视角立体视觉算法，即 PMVS 算法以及进行

计算优化后的 CMVS 算法。

基于片面的算法首先查找二维图像中观测对象表面上灰度急剧变化的边缘和顶点作为特征点，之后使用匹配跟踪算法确定这些特征点在不同的二维图像上的对应关系，然后应用计算机视觉理论和三角理论恢复各个点的三维信息，完成稀疏重建，即上述 SFM 所完成的步骤。稀疏重建的模型通常用稀疏点云来呈现，点云中的“点”并非只是一个点而是矩形面片，并用面片的法向量来表示该“点”的方向，稠密重建的最后步骤就是将稀疏的面片进行合理扩展，扩展为稠密的面片，使形成的稠密点云能比较完全的覆盖观测对象。

PMVS 算法测试在上述的稠密重建策略中会根据灰度一致性和几何可见性进行多次的扩展和剔除，以此来优化重建的效果。

2.3 本章小结

本章从 SFM 所依赖的特征点检测算法为切入点，介绍了 Harris 角点、FAST 算法、SIFT 算法以及 SURF 算法并展现了各个算法的结果；然后介绍了 SFM 算法的基本原理以及相关步骤的常用算法，之后介绍基于面片的稠密重建和当前常用的 PMVS 算法。

第三章 多视图稠密三维重建

3.1 稀疏重建

3.1.1 增量式 SFM

增量式的 SFM 算法从作为重建基础的初始的两张二维图像开始重建，暂时将初始的两张图构成的重建基础称为重建框架，通过位置估算和三角化将新的三维视点迭代地添加到重建框架中，对重建框架不断地进行扩展与优化。

使用 SFM 算法进行三维重建会涉及到基础矩阵（Fundamental matrix），基础矩阵表示现实世界中的一个点，在两幅图像中所显示的位置的变化关系。如图 3-1 所示，对于两幅图中给定的对应的两个点 (x_l, x_r) ，它们间的转化关系如下

$$x_l^T F x_r = 0 \tag{3-1}$$

其中 F 为 3×3 的基础矩阵，基础矩阵将立体空间中的点 x 在某一图像上的投影约束在一条投影线上，当获取点 x 在另一平面上的投影时便可知道点的空间位置。

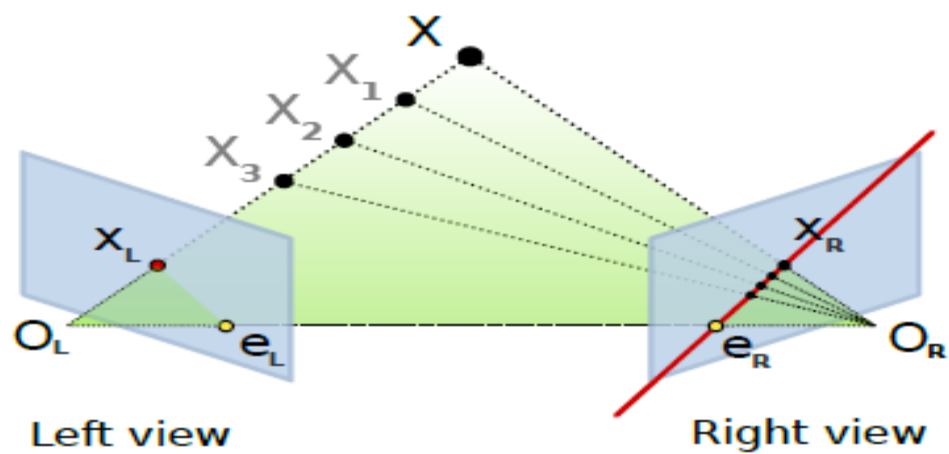


图 3-1 位置关系示意图^[1]

对于基础矩阵 F 计算可以用两幅图中以完成匹配的 N 对特征点进行估算，例

如给定一组特征点 $x_l(u, v, 1)$ 和 $x_r(u', v', 1)$ ，根据公式 3-1 可得

$$u'uf_{11} + u'vf_{12} + u'f_{13} + v'uf_{21} + v'vf_{22} + v'f_{23} + uf_{31} + vf_{32} + f_{33} = 0 \quad (3-2)$$

若记 $f = (f_{11}, f_{12}, f_{13}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}, f_{33})^T$ ，即 f 为有基础矩阵三个行向量组成的 9 维向量。而因为基础矩阵非零，所以 f 为非零向量，由此可知所需特征点对数 $N \geq 8$ ，由此求得 f 即可获取基础矩阵 F 。但实际计算中，特征点之间的匹配会存在误差，所以会在一定的约束条件下利用最小二乘法对基础矩阵进行逼近。

在求得基础矩阵 F 后，为方便表示和计算可引入投影矩阵。设立体空间中的点 $B(X, Y, Z, 1)$ 在两幅图像上的投影点分别为 $b_l(x, y, 1)$ 和 $b_r(x', y', 1)$ ，则对于投影矩阵 P 和 P' 有 $b_l = PB$ ， $b_r = P'B$ ，结合公式 2-8 便可求取各个图像所对应的投影矩阵。

基础矩阵 F 的获取解决了增量式 SFM 算法中确定特征点三维坐标的问题，但这只能确定两两图像之间的点的坐标，无法选取最合适的两张图像来构建重建框架，也不能将所有图像中的点整合在一起，形成观测对象的完整点云。而一般会使用以下策略来完成增量式 SFM 的三维重建：

- 1、从所有可以完成匹配的特征点对中选取关键跟踪点（即标定在图像集的大部分图像中都存在的特征点），并采用随机采样一致算法（RANSAC）剔除匹配错误的特征点。
- 2、从图像集中选取特征点足够多、焦距尽可能接近、图像空间间隔适中的两幅关键图像用于 SFM 的初始计算。
- 3、根据两幅关键图像可求得基础矩阵和每幅图像对应的投影矩阵，之后根据三角原理便可以获取关键图中特征点的三维信息，完成重建框架的构建。
- 4、在重建框架上逐步加入新的图片，以此来获取其他特征点的三维信息，并优化已获取的特征点的三维信息，直至遍历所有图片，完成重建。

对应于 openMVG 库中，有相应的类 `IncrementalReconstructionEngine`，调用

其成员函数即可完成重建；其中 `process()` 为核心函数，调用其它函数来完成重建。
具体的代码流程如下：

```
ReadInputData(): 读入图片、特征描述子等数据；
InitialPairChoice(): 初始化重建所需最佳的两张图；
MakeInitialPair3D(): 初始化本质矩阵  $E$ ，计算  $[R|t]$ ；
BundleAdjustment(): 初始调整三维点和相机及参数；
while 仍存在未计算图片 do
|   FindImagesWithPossibleResection()列出对当前模型拥有最多匹配点的图
|   Resection()在当前三维模型中根据三角模型添加新图片
|   while 精度未符合要求 do
|   |   BundleAdjustment()使用 BundleAdjustment 算法调整精度
|   end
end
```

根据以上策略，对图书馆进行增量式 SFM 重建，结果如图 3-2，3-3 和 3-4 所示：



图 3 -2 重建原图

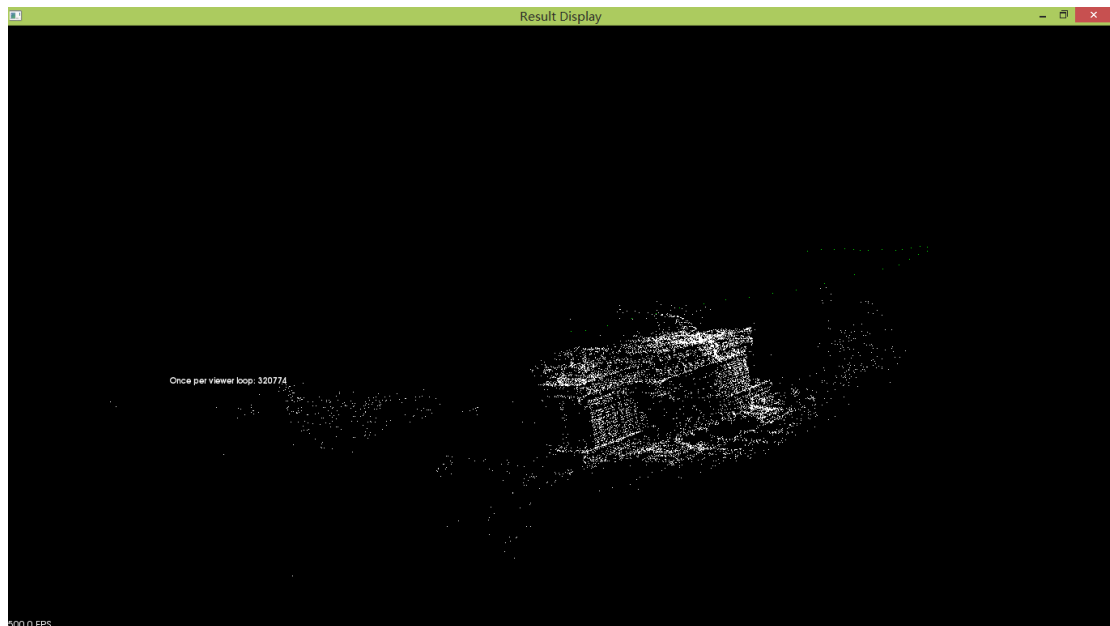


图 3 -3 左侧观测点云

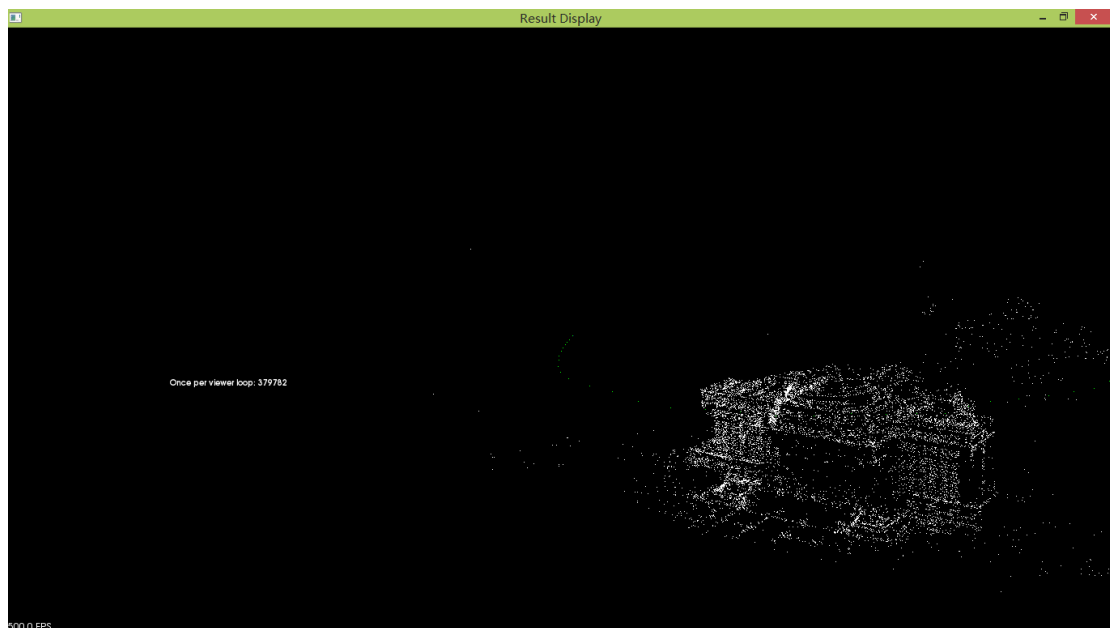


图 3 -4 右侧观测点云

虽然比较稀疏，但是已经能观测出建筑物的大致轮廓。

3.1.2 全局式 SFM

全局式 SFM 不同于增量式 SFM 选取两幅关键图像来构成，它会对整个图像集做一个估算，此时会用到本质矩阵（Essential matrix）。本质矩阵的计算公式如下：

$$E = [R|t]x = K'^T FK \quad (3-3)$$

其中 E 为本质矩阵， R 为相机相对于世界坐标系的旋转变换矩阵， t 表示在原始世界坐标系下的位置对应与相机为中心的坐标系的表达，而 $[R|t]$ 可以表示相机不同时刻间的位置变换， K 表示相机的固有参数矩阵， F 为基础矩阵。如图 3-5 所示，本质矩阵能表示出相机间的平移与旋转变换，并且将相机位置与基本矩阵 F 关联起来。

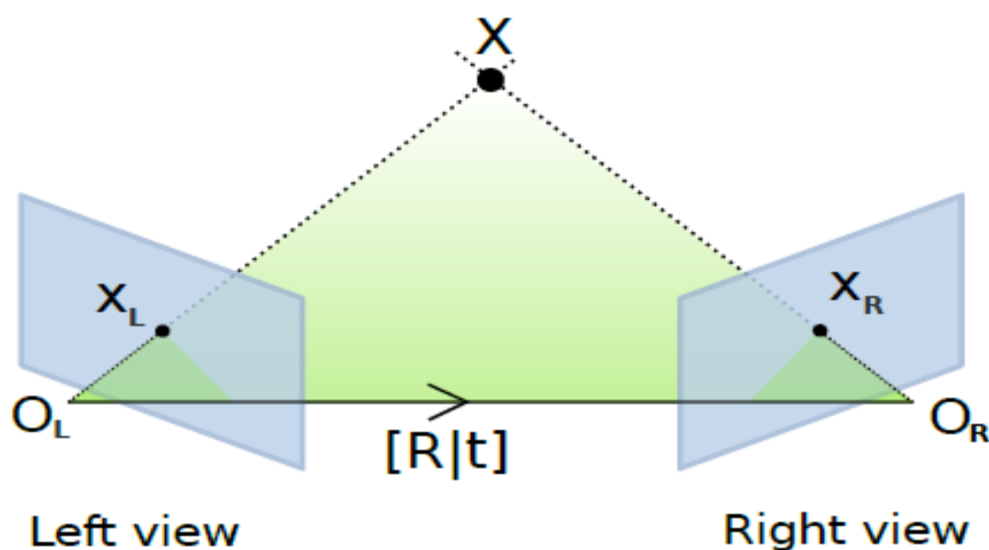


图 3-5 位置关系示意图^[1]

在已计算出基础矩阵并得到相机焦距、感光元件尺寸的前提下，根据公式便可获取本质矩阵。但获取本质矩阵只是全局式 SFM 中的一个步骤，其具体步骤如下：

- 1、特征匹配。类似增量式 SFM，首先使用 SIFT 进行二维图像集合的特征点提取，然后根据对极几何将所有相互有重叠的幅图像两两进行特征点的匹配，计算出本质矩阵，确定点的位置以及观测对象的结构，最后将位置和结构进行合并。
- 2、一致性检测。使用贝叶斯推断（Bayesian inferences）检测特征点是不是不同图像上是否具有有一致性，剔除有噪音引起的异常点。
- 3、计算转换关系。根据每幅图像的边界计算对应图像的相对转换估计，以此来确定该图像的投影方向，从而确定每个对应相机的全局方向。

4、全局坐标。建立全局坐标系

5、完成重建。通过三角原理计算每个点的三维空间位置，并对进行转换以应到全局坐标中，再使用集束约束（Bundle Adjustment）进行优化。

对应于 openMVG 库中有 GlobalReconstructionEngine 类，调用其成员函数 process()来实现各个功能的执行，从而完成重建，具体代码执行流程如下：

```
ReadInputData(): 读入图片、特征描述子等数据；
exportToGraphvizData(): 保存图片重建所需图片；
CleanGraph(): 清除不需要的图片；
ComputeRelativeRt(): 计算[R|t]转换关系；
rotationInference(): 检测并排除旋转变换的异常数据；
tripletListing(): 列出对于三维点上相关联的图片；
列出相机坐标点，建立一致性表；
computeGlobalRotations(): 通过相对旋转关系计算全局旋转关系；
computePutativeTranslation_EdgesCoverage(): 以边缘覆盖算法计算转换关系；
检测相对转换，建立相机坐标和相对顺序 map，建立针孔相机数据；
通过相关联的图片建立轨迹（track），对所有轨迹进行三角分割；
while 仍存在未遍历轨迹（track） do
| 查找三角化所需的特征点，建立投影矩阵 P；
| 计算三维点及其深度；
end
移除无用的轨迹和三维点；
bundleAdjustment_t_Xi()及 bundleAdjustment_Rt_Xi(): 对模型进行优化；
```

根据上述的代码流程所得的程序完成全局式 SFM 的重建结果如图 3-6 和 3-7 所示，相比增量式 SFM，全局式 SFM 需要更多的特征点，因此重建时的效率更低；虽然点云相对稠密，但因为过多的特征点也会产生对于观测对象无用的点云。

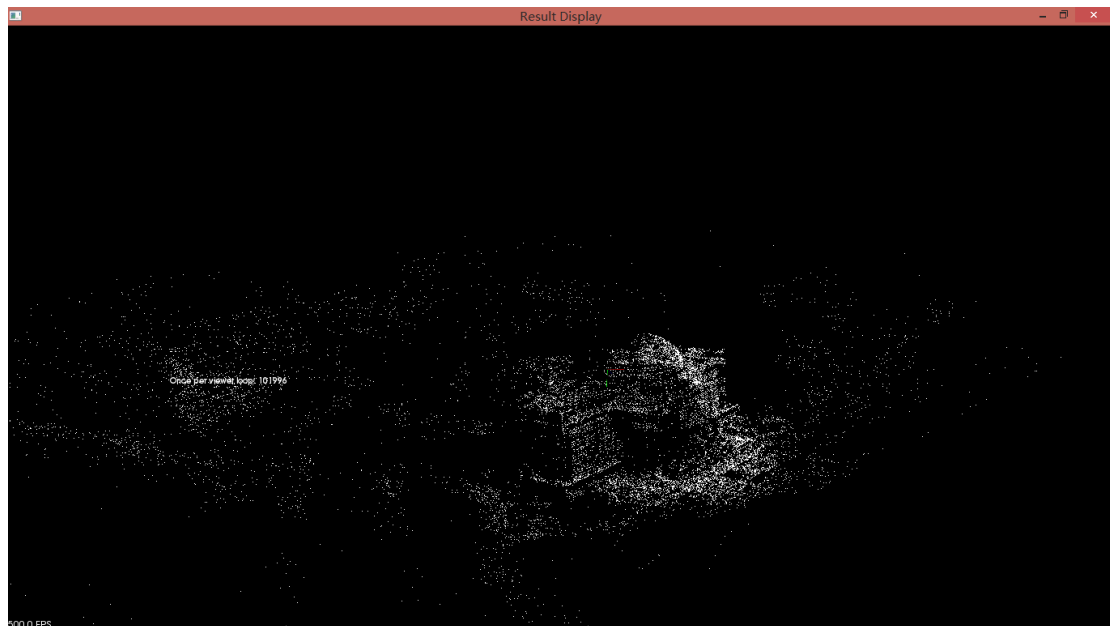


图 3 -6 左侧观测点云

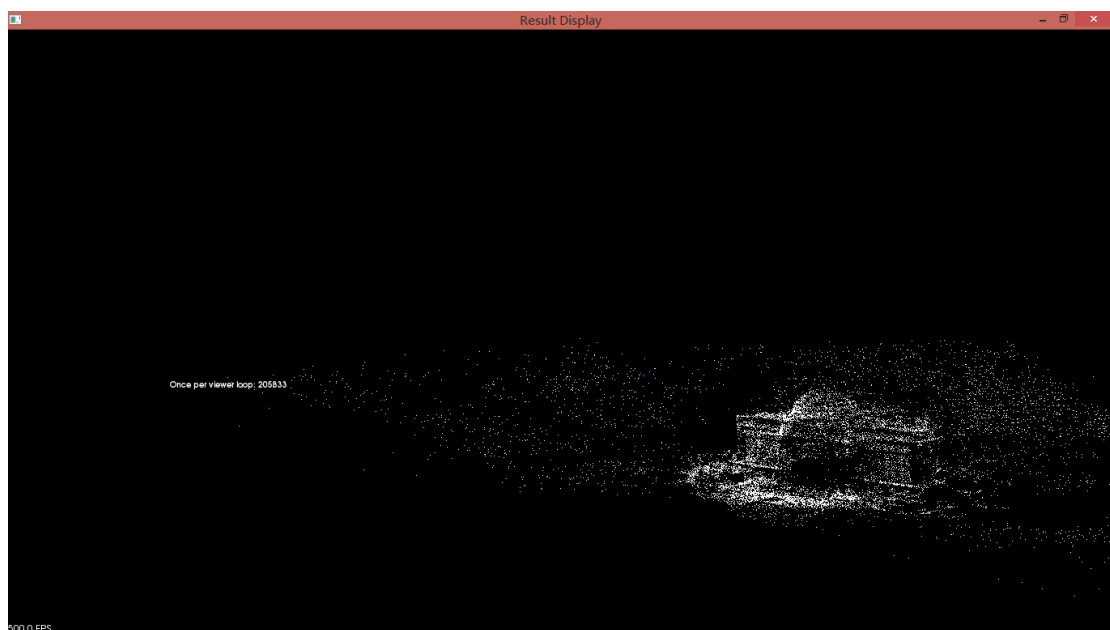


图 3 -7 右侧观测点云

3.2 稠密重建

3.2.1 pmvs 算法

PMVS 算法所重建而产生的点云用方形面片表示，用面片的法向量来表示该三维点的方向。算法在执行时会由关键面片进行扩展，来生成新的面片已达到稠密重建的目的，而完成稠密重建的标志是面片对观测对象达到了完全覆盖。为进

行完全覆盖的定量判断，会将每幅图像描述成由一定数量的图像细胞（即一块像素集合）组成的集合，每次面片扩展会进行一次扫描，如果所有图像细胞都至少对应一个重建面片的投影，则认为物体表面全部被覆盖，结束算法执行，完成 PMVS 重建。

PMVS 算法具体分为三个步骤：特征匹配，关键面片扩展，剔除异常点。其中特征匹配与 SFM 中的类似，不再赘述。

关键面片扩展就是使面片集合能够完全观测对象可见表面，从而获得对象表面的完整信息；扩展的具体过程为：对于一个已存在的面片，从其相邻的图像细胞集合中找到还未被扩展且在现实世界中所处位置深度连续的图像细胞，用来扩展新面片，然后用原面片的法向量、所属图像和满足视角约束的可见图片集来初始化新面片，并用之前得到的图像细胞中心视线和原面片所在平面的角点作为新面片中心点，之后用共轭梯度法来优化新面片的法向量与中心点，最后将新面片的根据深度映射得到的可见图片并入到视觉约束可见图片集中，如果灰度一致的可见约束图片集达到某一阈值，则表示完成一个新面片的扩展。

由于噪音的存在，扩展后所得到的稠密点云会包含错误的面片，所以会对异常点进行剔除。为确保重建的准确性，会进行三步的剔除工作。首先剔除对应同一图像细胞中与关键面片不满足可见一致性的扩展面片，然后剔除灰度一致的可见约束图片集小于某一阈值的面片，最后选取某一指定面片所在图像细胞和相邻图像细胞对应的所有面片，当指定面片的相邻面片总数小于所有选取面片的 1/4，则剔除指定面片。

对应于 CMVS-PMVS 库中，有类 CpatchOrganizerS 用来完成面片的存储，用类 Cseed 来产生种子面片，用类 Cexpand 对面片进行扩展，用类 CfindMatch 执行对稠密重建的执行，具体执行代码流程如下：

```
m_seed.run(): 产生种子面片;  
m_pos.collectPatches(): 遍历并存储面片;  
while 小于迭代上限 do  
|   m_expand.run(): 对面片进行扩展;  
|   m_filter.run(): 剔除错误面片;  
|   updateThreshold(): 更新阈值;  
end
```

依照以上代码流程所得的程序可完成对 SFM 所得到的稀疏重建的稠密重建。

3.2.2cmvs 算法

PMVS 虽然很好地解决了稠密重建的问题，但由于需要对每个面片进行扩展和剔除的计算，计算量非常的庞大，往往导致稠密重建速度低下、效率不高的情况。而 CMVS 的提出就是为了解决 PMVS 计算效率低下的缺点，具体方法则是用在进行面片的扩展和剔除时采用聚类分析的方法来简化运算，提高运算效率。

聚类分析的主要目的是将相似的对象进行合并，组成一个新的集合，一般用于统计数据的分析与挖掘。聚类分析并没有明确的概念定义，因此存在多种聚类分析算法，不同领域的研究人员会根据不动的需要选择不同的聚类模型。

常用的聚类分析有基于连通性的聚类、基于质心的聚类和基于分布的聚类。基于连通性的聚类的基本思想是距离近的对象间的相关程度要大于距离较远的对象间的相关程度，因此该聚类模型需要确定的便是聚类的最大距离，即在怎样的距离范围内可作为对象的一个聚类，不同的距离设定会产生不同的聚类效果，基于连通性聚类的效果如图 3-8 所示。基于质心的聚类模型需要先寻找若干个中心向量，而这些中心向量不一定是数据集中的对象，每个对象聚类到离自己最近的中心向量，最后使所有对象到中心向量的距离最小，完成聚类，基于质心聚类的效果如图 3-9 所示。基于分布的聚类模型会根据对象集的分布特点，与统计学中的数据分布模型进行类比，然后使用最相近的分布模型来对对象集进行聚类，基于分布聚类的效果如图 3-10 所示。

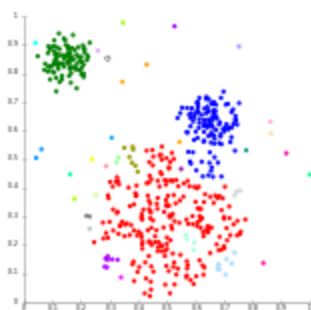


图 3-8 连通性聚类效果图

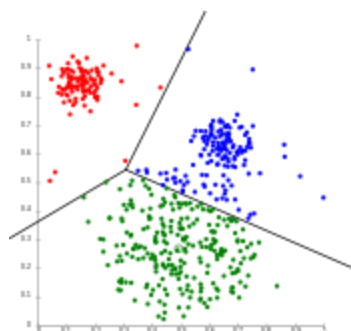


图 3-9 质心聚类效果图

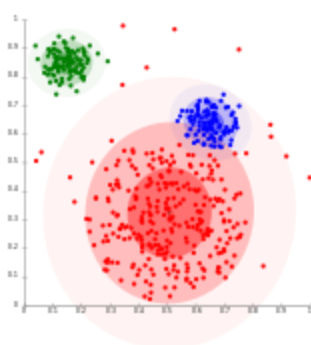


图 3-10 分布聚类效果图

CMVS 即在重建前，根据视角对图像序列进行聚类，而在聚类之后对每一类进行并行的重建，以加快重建速度。对应于 CMVS-PMVS 库中，有类 Cbundle 来完成聚类操作，具体的代码流程如下：

prep(): 对图片系统进行初始化；

while 未达到聚类数量 **do**

| **addImagesP()**: 添加图片；

| **while** 未遍历完图片 **do**

| | **divideImages()**: 将图片分配至所属类中；

| **end**

end

最后完成稠密重建的结果如图 3-11 和 3-12，在完成纹理的映射后基本恢复观测对象的概貌。

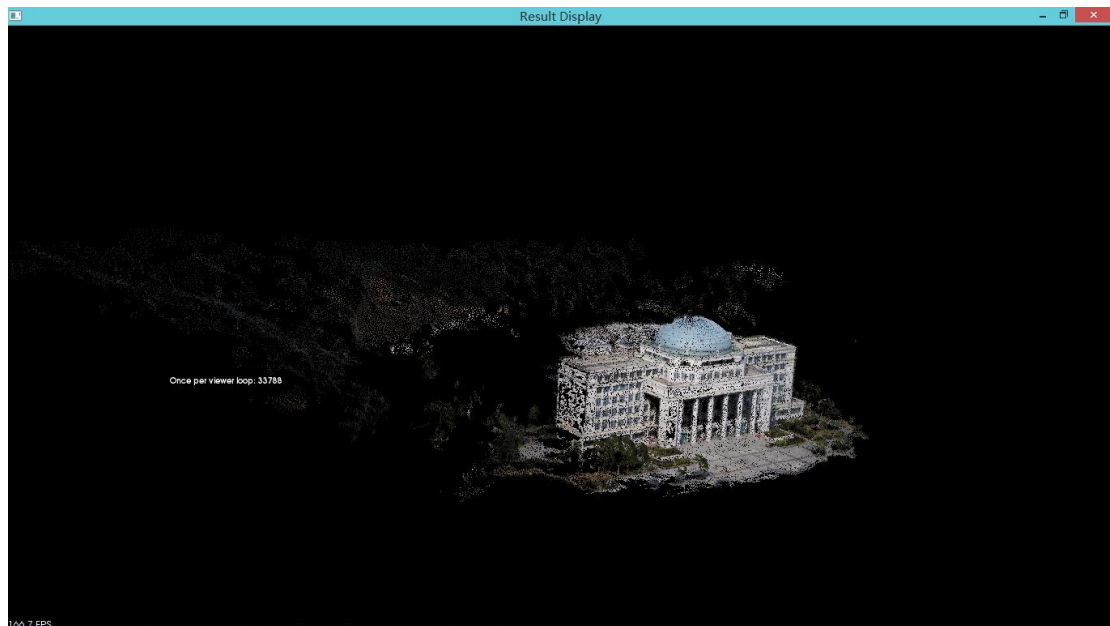


图 3 -11 左侧观测图

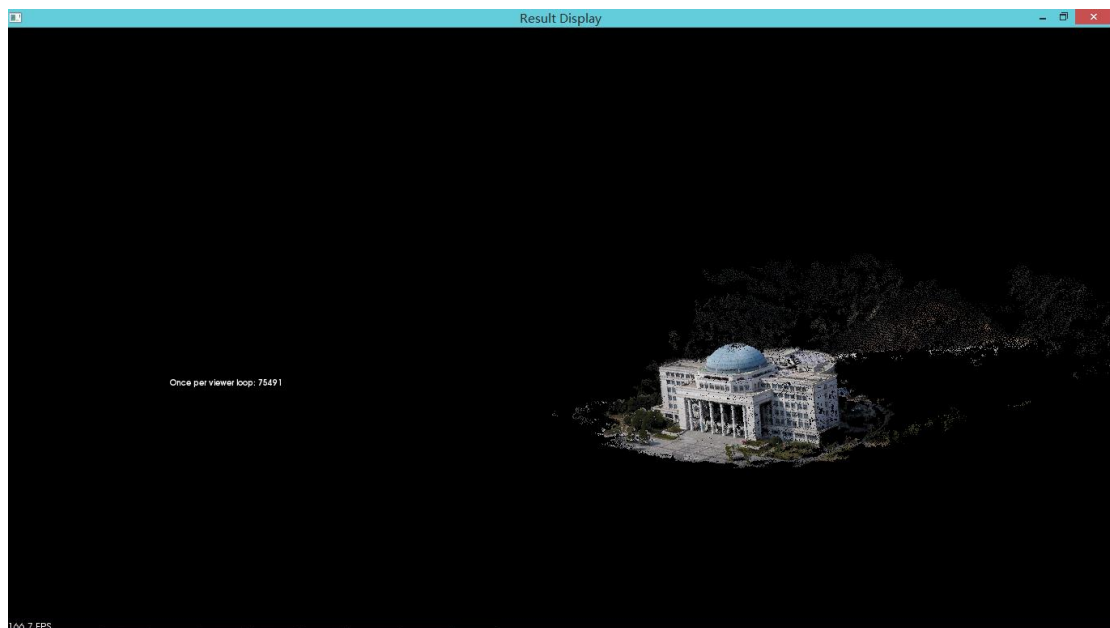


图 3 -12 右侧观测图

3.3 本章小结

本章开始介绍了 SFM 算法的增量式 SFM 和全局式 SFM 两个具体算法的实现步骤，并分别展现两种 SFM 算法的实现结果；在 SFM 稀疏重建的基础上介绍基于面片的稠密重建的重建算法 PMVS，以及 PMVS 对计算的改进 CMVS，最后展现观测对象的稠密重建结果。

第四章 图像拼接

4.1 二维图像拼接

无论是区域相关拼接还是特征相关拼接，图像拼接算法的步骤一般由图像处理、图像配准、图像融合组成。

图像的处理是指对获取图像的预处理，以减少之后处理的误差。图像在获取时可能会因为相机本身属性、拍摄环境、拍摄角度等影响，造成图像中观测对象的几何畸变等情况，而这样的几何畸变将导致图像配准时误差。此时就需要对失真对象进行合理矫正，从失真图像中得到真实值得估计。除了失真，获取的图片往往存在随机噪音，最终导致实际获得的图像与理想环境下所获得的图像存在误差，而噪音的存在也将导致图片处理的各个步骤出现其他误差。一般会对图像进行滤波的预处理，矫正图像的失真、剔除图像的噪音。

图像的配准根据所选算法的不同，其步骤也不同。区域相关拼接会选定配准模板，用配准模板对图像进行遍历式的配准，然后选取与配准模板最相似的部分，如此重复来确定拼接图像间的重叠部分，完成图像的配准；而某些情况下为了提高配准的计算效率，减小亮度变化对配准结果的影响，会把图像从空域转换到频域进行图像配准。而特征相关配准则是先提取图像的特征点，然后进行图像间特征点的匹配，以此来确定拼接图像间的重叠部分。

图像的融合即完成拼接的最后一步，但图像的融合并不是对图像的重叠部分后简单的将重叠部分的像素进行相加平均处理，因为不同的图像的重叠部分会存在差异，这样会出现明显的拼接裂缝。经过多年的研究，图像的融合有多种融合准则，一般认为在不同分辨率图像中，具有较大值的像素会包含更多的图像信息，所以在融合时会取较大的像素作为融合后图像的值。图 4-1，4-2 为原图，图 4-3 为拼接结果图。



图 4 -1 原图一



图 4 -2 原图二



图 4 -3 拼接结果

4. 2 三维拼接算法

4. 2. 1 经典 ICP 算法

类似于二维图像拼接中的特征相关算法，三维图像拼接也需要进行重合部分的叠加；但不同于二维图像拼接中只需要在二维坐标上的图像配准，三维图像的配准需要对点云进行旋转与平移这样的空间变换，使两个点云统一在同一三维坐标系下，但空间变换的规则是未知的，而且还存在两个点云中点的数目不一定相同的情况。而 ICP 算法需要解决的问题便是估计出空间变换规则，并用迭代最近点来解决点云中点数不一致的问题。

对于空间变换可以分解成旋转变换和平移变换，设旋转变换矩阵为 \mathbf{R} ，平移变换矩阵为 \mathbf{T} ，则对于三维空间中的点 \vec{p}_i 和点 \vec{q}_i 有以下关系：

$$\vec{p}_i = \mathbf{R} \vec{q}_i + \mathbf{T} \quad (4-1)$$

用数据点对 \mathbf{R} 和 \mathbf{T} 进行估算时，可以用最小二乘法求最优解，即使

$$E = \sum_i^N [(\mathbf{R} \vec{q}_i + \mathbf{T}) - \vec{p}_i]^2 \quad (4-2)$$

值最小，对应的 \mathbf{R} 和 \mathbf{T} 即为所需的空間变换规则。

根据上述的数学计算方法，先将问题分解为旋转变换矩阵 \mathbf{R} 的求取和平移变换矩阵 \mathbf{T} 的求取。首先对 \mathbf{T} 进行估算， \mathbf{T} 的估算值即为将两点云中心点进行重合

所进行的平移变换；从点云中选取控制点（Control Points）即已经确定的空间位置相近的点来确定旋转变换矩阵 R ；之后根据关系式：

$$T = R \rightarrow \begin{matrix} \rightarrow \\ q_i \end{matrix} \rightarrow \begin{matrix} \rightarrow \\ p_i \end{matrix} \quad (4-3)$$

来优化之前 T 的估值。由此得到旋转变换和旋转变换的初始值。

ICP 的具体算法描述如下：

- 1、对于源点云中的每一个点，在目标点云中寻找与其距离最近的点；
- 2、根据上一步的匹配结果，使用均方误差函数对平移矩阵和旋转矩阵进行修正；
- 3、用修正后的变换矩阵对源点云进行转换；
- 4、迭代，直到完成对所有点的匹配。

根据 ICP 算法可得到如图 4-4，4-5 的结果：

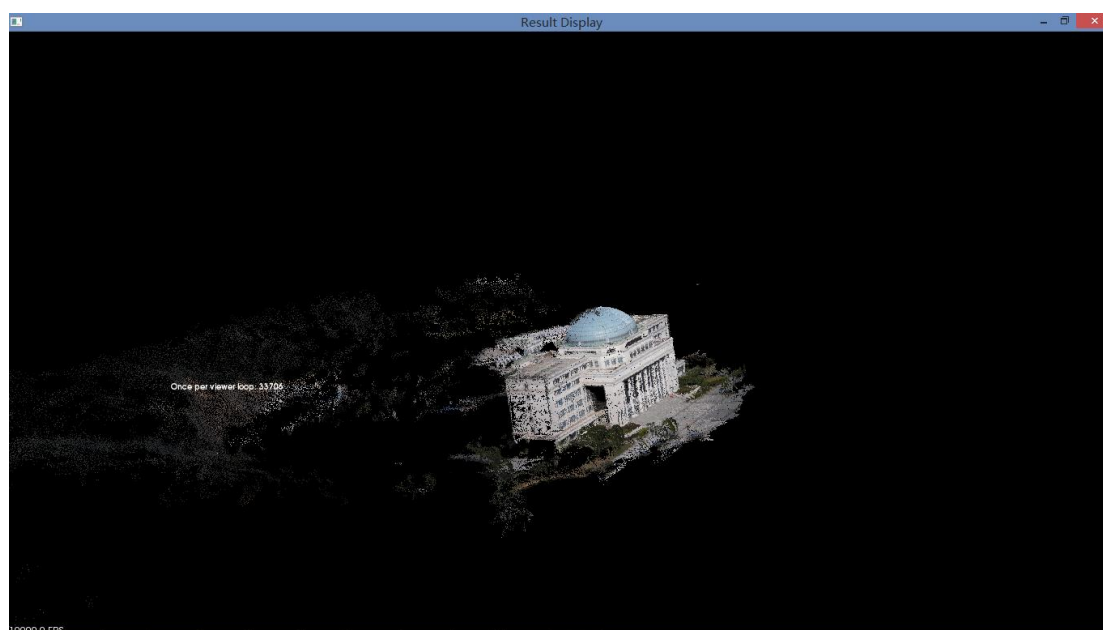


图 4 -4 左侧观测图

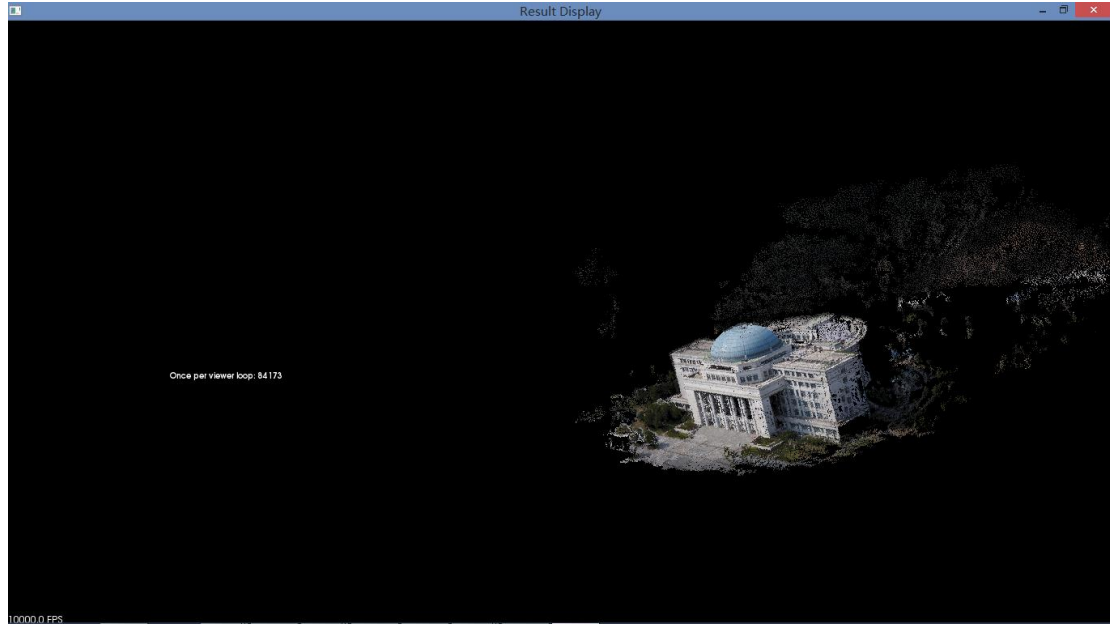


图 4 -5 右侧观测图

4. 2. 2Generalized ICP 算法

Generalized ICP 以标准 ICP 算法为基础，使用概率模型来计算点云配准时变换规则，并保持 ICP 算法的其他步骤不变以减少计算复杂度且保持运算速度。并且使用标准欧式距离来计算一致性，以确保在寻找最近点时可以使用 kd-树算法，这使 ICP 算法相比其他概率方法更快速简便。

在使用概率模型前，对于两个点云存在相应的两个基础点集 $\hat{A} = \{\hat{a}_i\}$ 和 $\hat{B} = \{\hat{b}_i\}$ ，有这两个点集根据正态分布可以产生 \hat{A} 所在的点集 A 以及 \hat{B} 所在的点集 B。假设基础点集具有完美的一致性，并且有准确的转换规则 T^* ，则会有以下的关系：

$$\hat{b}_i = T^* \hat{a}_i \quad (4-4)$$

对于任意的转换 T，定义 $d_i^{(T)} = b_i - T a_i$ ，则可知 $d_i^{(T^*)}$ 是其特例，且 $d_i^{(T)}$ 是 $d_i^{(T^*)}$ 的正态分布：

$$d_i^{(T^*)} \sim N(0, C_i^B + (T^*) C_i^A (T^*)^T) \quad (4-5)$$

其中 C_i^A 和 C_i^B 表示与被检测点相关的协方差矩阵，将等式 4-4 与 4-5 进行合并

便可得到转换规则 T 的计算公式。

4.2.3 Levenberg-Marquardt icp 优化算法

最优化算法是寻找使得函数值最小的参数向量，而 Levenberg-Marquardt 是最优化算法中的一种。最优化算法在经济学、网络分析、最优设计等众多领域有着非常广泛的应用。根据求导方法的不同，最优化算法可分为两类：知道模型函数解析式可直接求导与使用数值差分求导；而根据使用模型不同，分为非约束最优化、约束最优化以及最小二乘最优化。

LM 是广泛使用的非线性最小二乘法，介于牛顿法和梯度下降法之间，对于过参数化问题不敏感，能有效处理冗余参数问题，使代价函数陷入局部极小值的机会大大减小，这些特性使得 LM 算法在计算机视觉等领域得到广泛应用。LM 算法的关键是用模型函数 f 对待估参数向量 p 在其领域内做线性近似，剔除二阶以上的导数项，从而转化为线性最小二乘问题，它具有收敛速度快等优点。LM 算法属于一种“信赖域法”，所谓的信赖域法，即在最优化算法中，都是要求一个函数的极小值，每一步迭代中，都要求目标函数值是下降的；从初始点开始，先假设一个可以信赖的最大位移 s ，然后在以当前点为中心，以 s 为半径的区域内，通过寻找目标函数的一个近似函数的最优点，来求解得到真正的位移。在得到了位移之后，再计算目标函数值，如果其使目标函数值的下降满足了一定条件，那么就说明这个位移是可靠的，则继续按此规则迭代计算下去；如果其不能使目标函数值的下降满足一定的条件，则应减小信赖域的范围，再重新求解。

用 LM 算法优化 ICP 算法的结果如图 4-6 和图 4-7 所示：

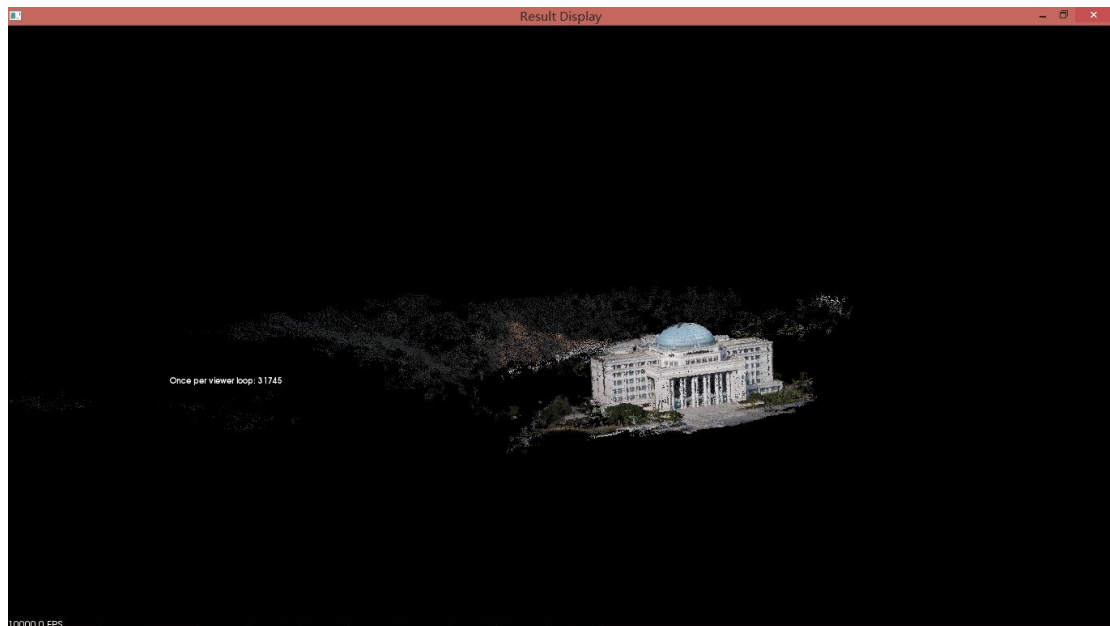


图 4 -6 左侧观测图

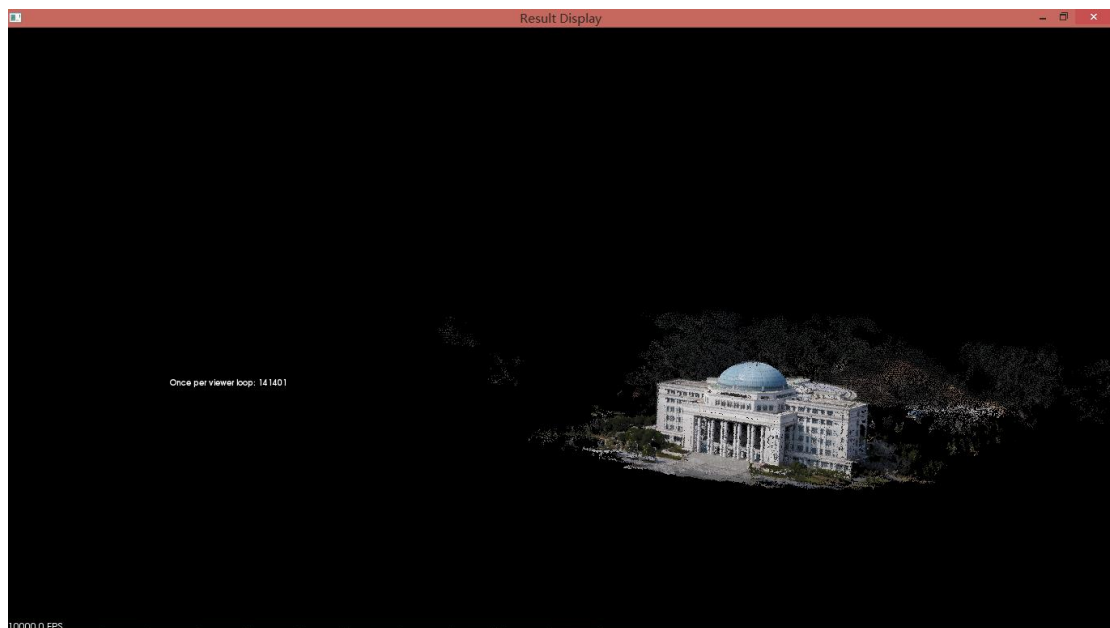


图 4 -7 右侧观测图

4.3 本章小结

本章先介绍了二维图像拼接的一般原理和步骤，然后扩展到三维图像的拼接算法；对于三维图像拼接算法介绍了当前常用的 ICP（迭代最近点）算法，并描述了 Generalized ICP 和 LM-ICP 等 ICP 算法的优化算法，展示了对应算法的结果图。

PCL 库中分别有模板类 `IterativeClosestPoint`, `GeneralizedIterativeClosestPoint`,

IterativeClosestPointNonLinear 来完成上述三个 ICP 算法，用模板类 NormalEstimation 来完成点云法向量的计算，用模板类 KdTree 完成 kd-树搜索。在代码实现上，均采用以下代码流程：

```
loadData(): 导入点云数据;  
模板类 NormalEstimation 的方法 compute()计算法向量;  
建立对应的 icp 算法模板类;  
setTransformationEpsilon(): 设置变换阈值;  
setMaxCorrespondenceDistance(): 设置一致性距离;  
while 小于迭代次数 do  
|   align(): 进行点云对齐计算;  
|   getFinalTransformation(): 对每次迭代变换进行累积;  
|   改良一致性距离;  
end
```

第五章 总结与展望

5.1 总结

本文完成的主要工作：测试了 Harris 角点、FAST 算法、SIFT 算法和 SURF 算法；了解 SFM 的算法原理，并阅读了 openMVG 中 SFM 算法的相关代码，为了整合进自己的软件进行了一些修改；了解 CMVS/PMVS 的算法原理，并阅读了 CMVS-PMVS 库中的相关代码；了解 ICP 算法的原理。阅读 PCL 库的相关接口，并调用相关接口编码完成点云的拼接功能、ply 格式转 pcd 格式功能、ply 和 pcd 文件显示功能；学习 Qt 图形编程的相关操作，完成一个具有重建和拼接功能、拥有图像操作界面的软件。

软件的最终界面如图 5-1，软件主要功能模块的操作如流程图 5-2 和图 5-3：

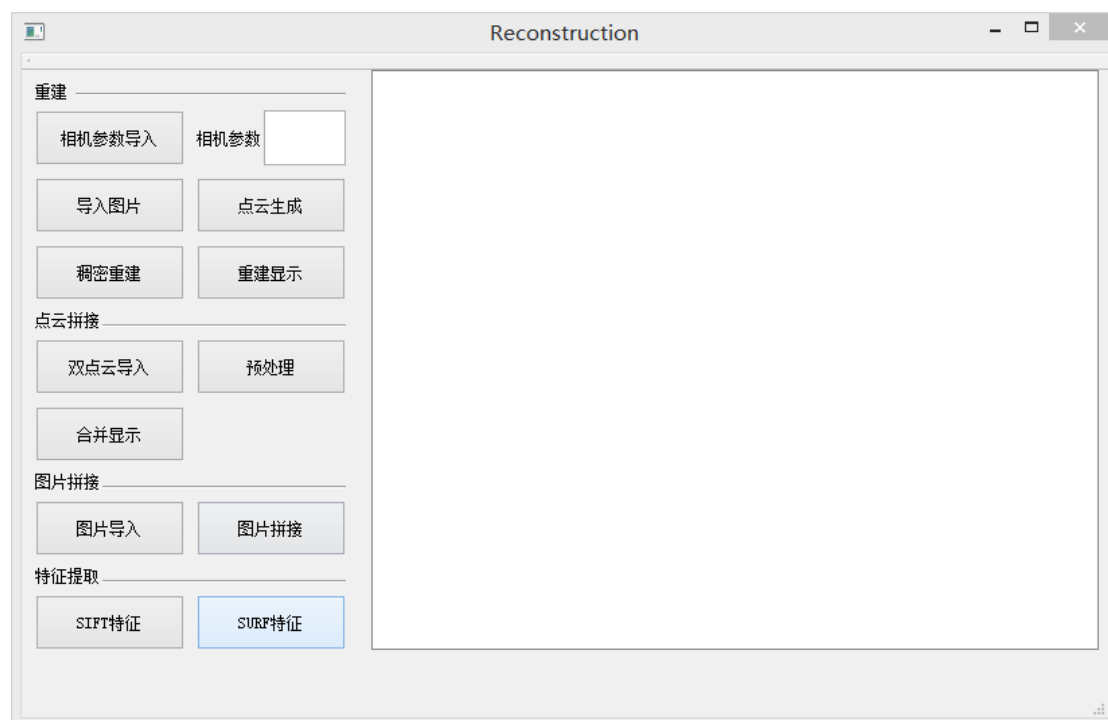


图 5 -1 软件界面



图 5-2 重建流程

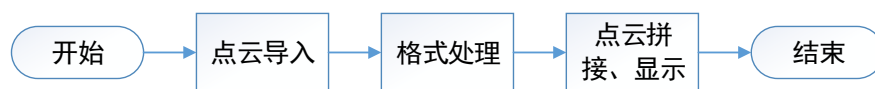


图 5-3 拼接流程

5.2 展望

对于 openMVG, CMVS-PMVS 等开源库的解读与使用还只停留在最基本的层面，以后可以对这些开源库进行更深入的理解和使用。对于 PCL 和 openCV 这些成熟的图形处理库，可以更多的去了解各种功能接口，理解某些处理的算法逻辑。而对 Qt 的界面编程引用也只是停留在初级阶段，还需要更好的去理解信号和信号槽的实现原理和交互逻辑。

参考文献

- [1] Pierre Moulon, Pascal Monasse, Renaud Marlet. Adaptive Structure from Motion with a contrario model estimation [A]. ACCV 2012[C]
- [2] Lowe D G. Distinctive image features from scale-Invariant keypoints[J]. International Journal of Computer Vision, 2004.
- [3] Yasutaka Furukawa, Jean Ponce; Accurate, Dense, and Robust Multi-View Stereopsis[J].
- [4] 谢则晓, 徐尚, 三维点云数据拼接中 ICP 及其改进算法综述[J] 中国海洋大学学报 2010 40 (1) 099-103
- [5] Aleksandr V . Segal, Dirk Haehnel, Sebastian Thrun, Generalized-ICP
- [6] S. M. Simth, J. M. Brady, Susan - a new approach to low level image processing [J]. International Journal of Computer Vision 23, 1997
- [7] Akca, D.: Full automatic registration of laser scanner point clouds [J]. In: Optical 3-D Measurement Techniques VI, pp. 330 - 337. Zurich (2003)
- [8] Xutao Kuang, Mark Gibson, Bertram E. Shi, and Michele Rucci. Active Vision During Coordinated Head/Eye Movements in a Humanoid Robot [J]. IEEE TRANSACTIONS ON ROBOTICS.
- [9] Park J S. Interactive 3D reconstruction from multiple images: A primitive-based approach.[J]. Pattern Recognition Letters, 2005, 26 (16):2558-2571
- [10] BELHUMEUR P, KRIEGMAN D, YUILLE A. The bas-relief ambiguity [J]. International Journal of Computer Vision, 1999, 35 (1): 33-44
- [11] WOODHAM R J. Photometric method for determining surface orientation from multiple images [J]. Optical Engineering, 1980 19(1):139-144
- [12] Zhang Zheng-You. A new multistage approach to motion and structure estimation by gradually enforcing geometric constraints [A]. In: Proceedings of the ACCV'98[C], Hong Kong, 1998, 567-564
- [13] 王琨 郑南宁. 基于 SFM 算法的三维人脸模型重建[J]. 计算机学报, 2005,

28(6), 1048-1053

- [14] 呼艳 耿国华 王小凤 王璐璐. 一种用于未标定图像三维重建的立体匹配算法[J]. 计算机应用研究. 2010, 27(10), 3964-3967
- [15] 李健 史进. 基于 openCV 的三维重建研究[J]. 微电子学与计算机. 2008, 25(12), 29-32
- [16] 毛剑飞 邹细勇 诸静. 改进的平面模板两步法标定相机[J]. 中国图像图形学报. 2004, 9(7), 846-852
- [17] C. Harris and M.J. Stephens, A combined corner and edge detector [A]Alvey Vision Conference, pp[C]. 147 - 152, 1988.
- [18] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, SURF: Speeded Up Robust Features [J]. Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346--359, 2008.
- [19] 宰小涛 赵宇明. 基于 SIFT 特征描述子的立体匹配算法[J]. 微计算机信息, 2007, 23(8—3)
- [20] 赵向阳, 杜利民. 一种全自动稳健的图像拼接融合算法[J]. 中国图象图形学报, 2004, 9(4): 417—42