**第3章 面向堆体的三维重建方法改进**

3.1 引言

在计算机视觉中，三维重建是指基于对环境或者物体的一系列不同视角的照片，通过特定流程的处理，获得环境或者物体的三维模型，模型表达方法众多，常见形式包括点云格式，网格格式，深度地图模式等。三维重建的整个操作流程简易，只需要将采集到的2D图像，或者截取视频中的图像作为输入，传递给三维重建系统即可，通过一系列的处理即可得到所拍摄场景的三维维点云模型，每一个元素具备三维位置信息和RGB颜色信息。三维重建的应用广泛，在自动驾驶，VR，AR等众多领域都有涉及，在未来也会进一步的和计算机视觉中的各种识别方法相互结合，更好的服务于现代科学。

基于视觉的三维重建流程已经相对成熟，在一般情况下，也能得到一个较为满意的结果，但是传统三维重建方法还是存在很多的问题，三维重建大多数都应用于离线环境，一旦输入数据量较大时，流程本身会十分耗时，且对处理设备的要求也会提升；此外三维重建对场景的要求也比较高，在光照条件不良，或者场景重复度较高的环境中，三维重建最终输出的点云会存在无法闭合，噪音点对以及场景歧义等现象，这些都限制了三维重建的应用范围。

基于以上问题，本章将提出一种结合SLAM结果的优化三维重建方法，以解决三维重建耗时和精度不高的问题。三维重建的输入为无序的图像序列，因此在匹配和解算位姿时都会耗费较大的算力和内存，并且匹配结果因为确实时序信息无法保证匹配的准确性，从而导致后续的位姿解算也会发生错误。但在SLAM系统中，由于考虑到图像的时序信息，在匹配时不需要进行完全匹配，可以从相邻帧或者回环中检测出匹配对从而节省算力，此外更加精确的匹配关系能得到更加准确的位姿信息，以为后续三维重建中的稀疏建模和稠密建模获取更高的精度和鲁棒性。

3.2 三维重建方法研究

3.2.1 三维重建流程概述

三维重建是指从三维图像中复原三维场景或者物体的过程，整个流程的输入为无序的图片即可，输出可以得到三维重建后的稀疏点云和稠密点，大致流程如图~\ref{fig:3Dconstr\_pipiline\_sfm}所示。

\begin{figure}[H] % use float package if you want it here

\centering

\includegraphics[width=12cm]{3Dconstr\_pipiline\_sfm.png}

\caption{三维重建流程图}

\label{fig:3Dconstr\_pipiline\_sfm}

\end{figure}

对于以上流程进一步进行细化，整个三维重建的过程可以划分为以下几个主要的步骤：

1. 2D图像采集：多角度拍摄或者从视频中提取到一组图像序列，将图像序列作为整个系统的输入;

2. 特征点提取和匹配：根据拍摄到的图像，提取每张图像之间的特征点，并进行特征点的匹配;

3. 稀疏点云：根据匹配结果估计特征点的深度，提取出稀疏点云，并估计相机的位姿和参数;

4. 稠密点云：根据优化后的相机参数和匹配结果，获得稠密点云;

5. 纹理映射：根据以上点重建物体表面，进行纹理映射。

三维重建的一般步骤可以简化为如图~\ref{fig:3Dconstr\_pipiline}所示。

\begin{figure}[H] % use float package if you want it here

\centering

\includegraphics[width=12cm]{3Dconstr\_pipiline.png}

\caption{三维重建一般步骤流程图}

\label{fig:3Dconstr\_pipiline}

\end{figure}

三维重建主要分为多个步骤，目前很多开源的系统都可以完成其中的部分环节，对于完整的三维重建流程还需要多个系统相互连接实现，表~\ref{tab:3D\_compare}是当前三维重建系统的简要对比，其中绿色框图代表该重建系统包括该流程，灰色则不包括。

\begin{table}[h]

\centering

\caption{常见三维重建系统对比表}

\label{tab:3D\_compare}

\begin{tabular}{C{3.6cm}C{2.4cm}C{2.4cm}C{2.4cm}C{2.4cm}}

\toprule

\textbf{系统名称} & \textbf{稀疏点云} &\textbf{稠密点云} & \textbf{重建表面} &\textbf{纹理映射} \\

\midrule

Bundler &\cellcolor{green}是&\cellcolor{gray}否&\cellcolor{gray}否&\cellcolor{gray}否\\

CMVS &\cellcolor{gray}否&\cellcolor{green}是&\cellcolor{gray}否&\cellcolor{gray}否\\

Colmap &\cellcolor{green}是&\cellcolor{green}是&\cellcolor{green}是&\cellcolor{gray}否\\

Meshlab &\cellcolor{gray}否&\cellcolor{gray}否&\cellcolor{green}是&\cellcolor{green}是\\

MVE &\cellcolor{green}是&\cellcolor{green}是&\cellcolor{green}是&\cellcolor{gray}否\\

MVS-texturing &\cellcolor{green}是&\cellcolor{green}是&\cellcolor{green}是&\cellcolor{gray}否\\

openMVG &\cellcolor{green}是&\cellcolor{green}是&否\cellcolor{gray}&\cellcolor{gray}否\\

openMVS &\cellcolor{gray}否&\cellcolor{gray}否&\cellcolor{green}是&\cellcolor{green}是\\

Theia &\cellcolor{green}是&\cellcolor{gray}否&\cellcolor{gray}否&\cellcolor{gray}否\\

VisualFSM &\cellcolor{green}是&\cellcolor{green}是&\cellcolor{gray}否&\cellcolor{gray}否\\

\bottomrule

\end{tabular}

\end{table}

3.2.2三维重建详细过程解析

3.2.2.1图像获取

目前三维重建仅需要输入无序图片，在构图的过程中，可以极大地降低操作的复杂度，并且对于相机的内参和外参也无需提前提供给整个三维重建的系统，在重建过程中，这些参数都可以计算得到。对于输入的图像，可以通过随着时间流单帧拍摄的方式获取或者通过截取视频流的方式得到，且图像之间不能仅有纯旋转，这样无法估计深度。对于堆体场景的图形采集，如图所示，需要注意保证每连续两帧图像之间尽可能保证30$\%$的重叠区域，相邻两帧之间的旋转角度为30度到45度之间，且物体中的每一个点至少能被三帧图像观测到。

3.2.2.2特征点提取与检测

特征是图像信息的另外一种数字表现形式，良好的特征应该是不受光线，噪音，几何变形影响的。目前，在三维重建领域中图像的特征匹配就是以特征点为基础而进行的，所以，如何定义和找出一幅图像中的特征点就非常重要，常见的特定点检测和匹配主要包括FAST，SIFT，ORB，Harris角点等。特征提取的目的是为了后续能够尽可能准确、稳定地估计出相机的运动，且特征点应该具备可重复性，高效率，可区别性以及本地性的特点，本文列举了在图像处理领域几种常用的特征检测和提取的方法。

\textbf{Harris角点：}当从不同的方向去移动一个视觉窗口，假设该区域内的灰度发生了很大的的变化，则认定存在角点。对于图像I(x,y)，当在点(x,y)处平移(Δx,Δy)后的对应窗口的像素点灰度变化描述为：

\begin{equation}

c(x, y ; \Delta x, \Delta y)=\sum\_{(u, v) \in W(x, y)} (I(u, v)-I(u+\Delta x, v+\Delta y))^{2}

\label{equ:Harris}

\end{equation}

结合泰勒分解公式，上式可以化简为：

\begin{equation}

c(x, y ; \Delta x, \Delta y) \approx \sum\_{w}\left(I\_{x}(u, v) \Delta x+I\_{y}(u, v) \Delta y\right)^{2}=[\Delta x, \Delta y] M(x, y)\left[\begin{array}{c}{\Delta x} \\ {\Delta y}\end{array}\right]

\end{equation}

其中

\begin{equation}

\begin{split}

& M(x, y)=\sum\_{w}\left[\begin{array}{cc}{I\_{x}(x, y)^{2}} & {I\_{x}(x, y) I\_{y}(x, y)} \\ {I\_{x}(x, y) I\_{y}(x, y)} & {I\_{y}(x, y)^{2}}\end{array}\right]\\

& =\left[\begin{array}{cc}{\sum\_{w} I\_{x}(x, y)^{2}} & {\sum\_{w} I\_{x}(x, y) I\_{y}(x, y)} \\ {\sum\_{w} I\_{x}(x, y) I\_{y}(x, y)} & {\sum\_{w} I\_{y}(x, y)^{2}}\end{array}\right] \\

& =\left[\begin{array}{cc}{A} & {C} \\ {C} & {B}\end{array}\right]

\end{split}

\end{equation}

公式~\ref{equ:Harris}即可转化为

\begin{equation}

c(x, y ; \Delta x, \Delta y) \approx A \Delta x^{2}+2 C \Delta x \Delta y+B \Delta y^{2}

\end{equation}

又提出角度响应值R来判断该点是否为角点：

\begin{equation}

R=\operatorname{det} \boldsymbol{M}-\alpha(\operatorname{trace} \boldsymbol{M})^{2}

\end{equation}

通过以上公式可以看出，Harris角点具备以下特点：

1. 对亮度和对比度的变化不敏感：因为微分运算对图像密度的变化不敏感，即亮度或者对比度的变化对Harris的检出影响较小;

2. 具有旋转不变性：Harris角点检测算子的本质可以表示为一个椭圆，但椭圆旋转时，并不会影响R值的大小;

3. 不具有尺度不变性。

\textbf{FAST特征点：}对图像中的中的一个像素p，假设其亮度值为$I\_p$，阈值为t，如图所示，取以其为中心，半径为是三个像素的圆，圆上共有16个像素点，假设这16个点中有连续N个点都比$I\_p+t$大或者比$I\_p-t$小，则认定其为Fast角点，为了加快检测过程，会直接检测第1,5,9,13这4个点，如果有三个满足要求，也会认为是角点。

\begin{figure}[H] % use float package if you want it here

\centering

\includegraphics[height=6cm]{2VSLAM\_Fast.png}

\caption{FAST特征点}

\label{fig:2VSLAM\_Fast}

\end{figure}

\textbf{ORB特征点：}结合了FAST特征点的检测和BRIEF描述子，其中RIEF特征描述符的计算过程：首先平滑图像，在某个像素点的周围选择一个领域，根据特定的点对选择方法挑选出$n\_d$个点对，比较每个点对之间亮度值的大小，即可得到一个长度为$n\_d$的二进制串。由于FAST特征点不具有方向，ORB在此基础上进行了改良，即利用灰度质心法求解出灰度额质心之间的偏移方向，首先定义特征点p的邻域像素的矩：

\begin{equation}

m\_{p q}=\sum\_{x, y} x^{p} y^{q} I(x, y)

\end{equation}

图像的质心为：

\begin{equation}

C=\left(\frac{m\_{10}}{m\_{00}}, \frac{m\_{01}}{m\_{00}}\right)

\end{equation}

那么偏移方向可以定义为FAST的特征点方向：

\begin{equation}

\theta=\arctan \left(m\_{01}, m\_{10}\right)

\end{equation}

以上说明了图像处理中常用的特征点提取方式，接下来将以此为基础说明特征点的匹配过程：在获取到每张图像上的特征点后，需要对图像之间建立匹配关系，常用的方式可以采用计算欧式距离的办法：

1）完全匹配，对所有的特征点都进行穷举，计算其对应距离。

2）邻近搜索,建立KD树,缩小搜索范围,能提高效率,但也有可能不是最优,所以邻域取值是关键,越大越准确,越大计算量越大。由图~\ref{fig:3Dconstr\_match}~所示，两帧之间大多数特征点都可以正确匹配，但依然存在存在部分匹配是错误的，在本文中，选择了RANSAC（随机抽样一致性）的方式来剔除错误的匹配对，以更加准确的估计相机位姿，RANSAC是指可以从一组包括局外点（错误匹配）的观测数据中，通过迭代的方式估计数学模式中的参数,通过RANSAC处理后的匹配结果如图~\ref{fig:3Dconstr\_matchAfterRansac}所示,误检的匹配已经被明显的降低。

\begin{figure}[H]

\centering

\subcaptionbox{第1帧图像提取SIFT特征}{\label{fig::3Dconstr\_a}

\includegraphics[width=6cm]{3Dconstr\_a.png}\hskip2cm}

\subcaptionbox{第2帧图像提取SIFT特征}{\label{fig:3Dconstr\_b}

\includegraphics[width=6cm]{3Dconstr\_b.png}}

\vskip0.5cm

\subcaptionbox{根据特征点进行匹配\label{fig:chap03:3Dconstr\_match}}{

\includegraphics[width=12cm]{3Dconstr\_match.png}\hskip2cm}

\caption{特征提取与匹配结果示意图}\label{fig:3Dconstrmatchresult}

\end{figure}

3.2.2.3 SfM

在~\ref{sec:3.2.2.2}小节中，可以获得初始的匹配关系，但是这种匹配关系不完全可靠，需要添加几何约束进行检测，该集合约束完全依赖于场景中的客观事实。可以通过基本矩阵F将匹配好的两帧图像之间的像素坐标(x,y),(x',y')进行关联，假设一个符合条件的匹配对像素坐标需要满足以下公式：

\begin{equation}

\begin{bmatrix}x'&y'&z'\end{bmatrix}\mathrm F\begin{bmatrix}\mathrm x\\\mathrm y\\\mathrm z\end{bmatrix}=0

\end{equation}

找到相机基线最大的像对，根据该像对，通过RANSC八点法计算本征矩阵，再通过对本征矩阵SVD分解得到第二个图像的R、T，在这一步需要进行畸变校正，然后根据R、T和矫正后的像点坐标三角计算出三维点。

当所有的两两匹配图像对被确定以后，可以开始计算相机的位姿（3\*3的旋转矩阵R，1\*3的平移向量t），摄像机的内参（焦距f，畸变参数$k\_1$，$k\_2$）。几何场景提供轨迹中的每个3D点$X\_j$，通过投影方程，将3D点投影到摄像机的2D成像平面上，投影误差的定义为投影点和图像上真实点之间的欧式距离,如图~\ref{fig:3Dconstr\_reprojection-error}所示。

\begin{figure}[H] % use float package if you want it here

\centering

\includegraphics[width=8cm]{3Dconstr\_reprojection-error.png}

\caption{误差投影示意图}

\label{fig:3Dconstr\_reprojection-error}

\end{figure}

对于n个视角和m个轨迹，投影误差的目标优化方程为:

\begin{equation}

\mathfrak g(C,X)=\sum\_{i=1}^n\sum\_{j=1}^m\omega\_{ij}\left|\left|q\_{ij}-P(C\_i,X\_j)\right|\right|^2

\end{equation}

其中$\left|\left|q\_{ij}-P(C\_i,X\_j)\right|\right|^2$就是摄像机i中的轨迹j的投影误差累积和,SFM算法的目标就是找到合适的相机和场景参数去优化这个目标函数，g是采用一个非线性最小二乘的优化方法求解，常用BA（光束平差法）来优化上述过程。最后，不断添加新的摄像机和3D点进行BA。这个过程直到剩下的摄像机观察到的点不超过20为止，说明剩下的摄像机没有足够的点可以添加，BA结束。得到相机估计参数和场景几何信息，即稀疏的3D点云。

3.2.2.4 MVS

SfM是指从运动中恢复结构的过程，而MVS则是多视角立体视觉生成统，SfM生成的是稀疏点云，恢复相机之间的几何关系，MVS生成的是稠密点云，由SfM获得的一些相机参数和相机之间的几何关系，来进行MVS。在SfM中，重建的点都是由特征匹配的点，这些点集本身就就不稠密，因此无法获取到稠密的点云，而在三维重建的过程就需要通过MVS的方式获取稠密的点集，MVS利用图像中的像素点来实现点云的重建，将图像中的每一个像素点估计其三维坐标，构成稠密点云。

在稠密点云的估计过程中，无法将每一个像素点按照特征点的方式计算其描述子，因此提出了极线搜索和快匹配技术来匹配图像中的某一像素点在其他图像中的对应的点，在找到每一个像素点在其他图像中出现的位置之后，就可以利用三角测量的发放确定其深度，但是因为用一个像素点会出现在多个图像中，所以就期望通过多次三角化让该点的深度值收敛。

对于极线搜索和块匹配技术，如图~\ref{fig:3Dconstr\_jixiansousuo}所示。

\begin{figure}[H] % use float package if you want it here

\centering

\includegraphics[width=8cm]{3Dconstr\_jixiansousuo.png}

\caption{极线搜索示意图}

\label{fig:3Dconstr\_jixiansousuo}

\end{figure}

首先对于极线搜索和块匹配技术，如图所示，左侧的相机观测到了像素点$p\_1$，由于该相机为单目相机，没法确定深度，可以先假设该点的实际位置在0到正无穷的区间中,因此该像素对应的空间点就分布在某条线段上，对于右侧相机，上述线段可以在成像平面上生成一段投影，在已知两帧图像之间的相对位姿时，是可以确定该极线的位置的，接下来就需要在极线上寻找与像素点$p\_1$所对应的$p\_2$点的位置。

因为单个像素没有对应的特征值，无法进行匹配，此外依靠单个特征点的亮度来进行匹配也并不可靠，可以基于图像块在前后变化得到过程中的灰度不变性特征，在P1周围选育一块w\*w得到小块，然后在右侧的极线上选择多个这样的小块来提高区分度。对于计算两个小块之间的差异，一般可以通过以下方法来计算：

1.SAD取两个小块的差的绝对值之和

\begin{equation}

S{(A,B)}\_{SAD}=\sum\_{i,j}\left|A(i,j)-B(i,j)\right|

\end{equation}

2.SSD取两个小块比较求平方和

\begin{equation}

S{(A,B)}\_{SSD}=\sum\_{i,j}(A(i,j)-B{(i,j))}^2

\end{equation}

3.NCC取两个小块计算相似性

\begin{equation}

S{(A,B)}\_{NCC}=\frac{{\displaystyle\sum\_{i,j}}A(i,j)B(i,j)}{\sqrt{{\displaystyle\sum\_{i,j}}A{(i,j)}^2\underset{i,j}\sum B{(i,j)}^2}}

\end{equation}

在极线上，计算了 A 与每一个 Bi 的相似性yo度量，那么将得到一个沿着极线的数值分布，该分布的形状取决于图像本身信息，在搜索距离较长的情况下，通常会得到一个非凸函数：该分布存在许多峰值，但是真实值只有一个，接下来需要使用深度滤波器使用概率分布来描述深度值从而找到对应的$p\_2$点。

根据以上流程，可以对图所示的连续堆体图像进行三维重建，点云结果如图所示。

2.3 联合二维码的SLAM系统描述

2.3.1引入二维码对SLAM优化研究

SLAM是一个在导航的过程中同时进行建图的工作，在使用单目相机时，即使目前表现较好的ORB-SLAM或者LDSO等算法也很难达到预期想要的效果。第一个问题：所构成的地图的尺度未知，这样就造成无法在实际的导航任务中发挥作用；

2.3.2包含二维码的地图描述

在SLAM运行的过程中，可以生成一套地图系统，对于同一场景，该地图可以作为先验复用于后续的建图和定位，通过这样的方式可以添加对系统的约束，提

2.3.3联合二维码的SLAM系统描述

本文所提出

2.4 面向堆体无人机定位系统设计

2.4.1 无人机相机坐标变换数学描述

人机上携带

2.4.2 获取真实世界坐标系方法

在方法1。

2.4.3 面向堆体的无人机自主定位设计

运行前设定,具体

2.5 本章小结

本章阐述