

基于Colmap的稀疏重建

1 简介

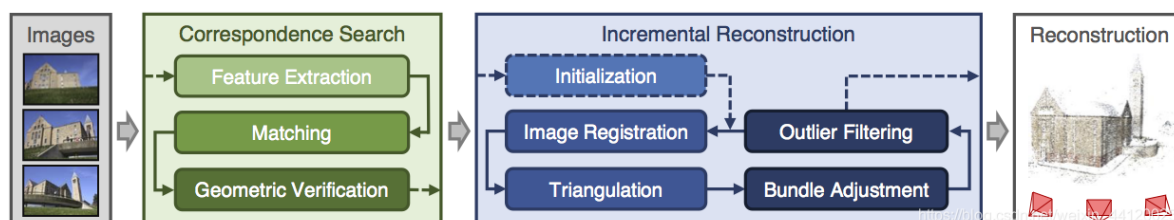
我们主要利用SfM进行稀疏重建。SfM, (Structure From Motion, 从运动中恢复结构), 是一种从一组不同视角下拍摄的无序或有序影像中, 同时恢复场景三维结构和相机姿态的技术。colmap主要采用增量式SfM, 目的是得到场景中的相机姿态和表示场景结构的稀疏点云。

2 增量式SfM

选择无序影像进行特征提取和特征匹配, 并进行几何纠正、三角测量恢复稀疏点云结构, 通过已有点云重新估计相对姿态, 再进行局部和全局的BA优化。之后逐步向已有的结构中增加视角或影像, 进行三角测量和姿态估计, 再进行BA优化修正结构数据, 最后输出全部的相机参数和稀疏三维点云。

• 具体介绍 (主要分为两个模块) :

1. 对于correspondence search, 先从图片中提取焦距信息(之后初始化BA(Bundle adjust)需要), 然后利用SIFT等特征提取算法去提取图像特征, 用kd-tree模型去计算两张图片特征点之间的欧式距离进行特征点的匹配, 从而找到特征点匹配个数达到要求的图像对。对于每一个图像匹配对, 计算对极几何, 估计F矩阵并通过ransac算法优化改善匹配对。这样子如果有特征点可以在这样的匹配对中链式地传递下去, 一直被检测到, 那么就可以形成轨迹。
2. 对于structure-from-motion (sfm) 部分, 关键的第一步就是选择好的图像对去初始化整个BA过程。首先对初始化选择的两幅图片进行第一次BA, 然后循环添加新的图片进行新的BA, 最后直到没有可以继续添加的合适的图片, 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。其中两幅图片之间的bundle adjust用的是稀疏光束平差法sba软件包, 这是一种非线性最小二乘的优化目标函数算法。



2.1 关于增量式sfm和全局式sfm

目前主流的SfM (Structure from Motion, 运动结构恢复) 可以分为两大类型, 一种是全局式的, 一种是增量式的。全局式sfm是先对所有的图像计算匹配关系、进行三角化生成三维点、通过pnp估计出位姿, 然后用BA (Bundle Adjustment) 进行一个整体的优化。这类方法效率较高, 但是鲁棒性差, 很容易重建失败。增量式sfm则是一边三角化和pnp, 一边进行局部ba。这类方法效率较低, 但是鲁棒性较高。

3 具体步骤

3.1 模块一: Correspondence search

3.1.1 特征检测 (feature extraction)

使用的是具有尺度和旋转不变性的SIFT描述子,其鲁棒性较强,适合用来提取尺度变换和旋转角度的各种图片特征点信息,其准确性强,在这种离线算法不需要考虑时间成本的情况下也较有优势。SIFT算法通过不同尺寸的高斯滤波器(DOG)计算得到特征点的位置信息(x,y),同时还提供一个描述子descriptor信息,在一个特征点周围4×4的方格直方图中包含8个梯度方向,即得到一个4×4×8=128维的特征向量。

3.1.2 特征匹配 (matching)

1. 计算特征点距离:

一般采用欧式距离.有两种方法:

- 粗暴匹配,对所有特征点都穷举计算距离
- 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

一旦每个图片的特征点被提出来以后,就需要进行图片两两之间的特征点匹配,用 $F(I)$ 表示图像 I 周围的特征点。对于每一个图像对 I 和 J ,考虑每一个特征 $f \in F(I)$ 找到最近邻的特征向量 $f_{nn} \in F(J)$:

$$f_{nn} = \arg \min_{f' \in F(J)} \|f_d - f'_d\|_2.$$

2. 确定匹配特征点

事实上算法中用到一个kd-tree的数据结构去计算最近邻匹配。然后令最近邻的距离为 d_1 ,再找到第二近的匹配对点之间距离为 d_2 ,如果两个距离 d_1 和 d_2 之比小于一个阈值如0.6,就可以判定为可接受的匹配对。

3.1.3 几何验证 (geometric verification)

1. 去除重复特征点匹配

但是图像 J 中可能匹配图像 I 中多个特征点,就会出现多对一的情况,实际上特征点之间应该一一对应。所以还需要一个**去除重复特征点匹配对的算法**去解决这种多对一的情况。

当距离小于一定阈值的时候就认为匹配成功,但是误匹配也比较多,需要采取多种手段剔除:

- 如果最近距离与次近距离的比值大于某个阈值,应该剔除
- 对匹配点采用采样一致性算法RANSAC八点法计算基础矩阵,剔除不满足基础矩阵的匹配对。
- 几何约束

然而初选的匹配对可能还是不可靠,需要用几何约束去检测。这个测试是基于事实的,假设一个静止场景,不是所有的匹配特征点在实际场景中是符合物理规律的。那么就需要计算对极几何。

一般采用的方法是利用E矩阵(已知相机内参 K 的情况下)或者F矩阵(相机内参 K 未知的情况)等,通过RANSAC来去除错误匹配。

2. 确定匹配图像

每张图像有多个特征点,最后如果两个图片之间的特征点匹配数不少于16个即为初选图像对。

3. 生成轨迹 (track)

当所有的两两匹配图像对被确定以后,就可以考虑把多个图像中都出现的共同特征匹配点连接起来,就能形成轨迹了。

输出为经过验证的图像对,及其对应的内点,也可包含图像间几何关系。该阶段输出也被称为scene graph(作为增量重建sfm的输入),图像为顶点,关联关系为边。

3.2 模块二：incremental construction

输入为scene graph，输出为注册的图像位姿和地图点

3.2.1 初始化 (initialization)

首先选择合适的初始化图像对，这十分重要，一旦错误的初始化，将会陷入局部最优而使得之后的BA陷入死循环，无法正确求解得到全局最优。具体有两点要求：第一，要有足够多的匹配点；第二，要有足够远的相机中心。

3.2.2 图像注册 (image registration)

求解PnP将新的图像添加进来。在这里用到两个图像变换之间的单应性模型来找初始化图像对。如果不能很好的符合单应性模型，说明相机中心还是有一定距离的。由于外点的影响，同样采用RANSAC方法来降噪，改善匹配的可靠性，尽量选取低的内点百分比，但是至少保证100个匹配内点。

3.2.3 三角化 (triangulation)

系统采用5点法来估计初始化匹配对的外参，然后轨迹三角化后可以提供初始化的3D点。

3.2.4 光束平差法 (Bundle Adjustment)

初始化的两帧图片就可以开始进行第一次bundle adjustment了。在这里用的是稀疏光束平差法sparse bundle adjustment(SBA)。Bundle Adjustment是一个迭代的过程，在一次迭代过后，将所有三维点反向投影到各自相片的像素坐标并分别与初始坐标比对，如果大于某个阈值，则应将其从track中去掉，如果track中已经小于2个了，则整个track也去掉，一直优化到没有点可去为止。

3.2.5 小结

初始化主要是指选取两张匹配的图像，设定其中一张图像的位姿为单位阵，然后通过它们之间的匹配点对估计出E矩阵，将E矩阵分解获得另一张图像的位姿，最后在估计出两张图像的位姿后，就可以通过三角化来生成三维点，通过BA优化误差累积和函数。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。。

初始化后是一个循环，每次新加入一张图像

每次循环包括如下四步：

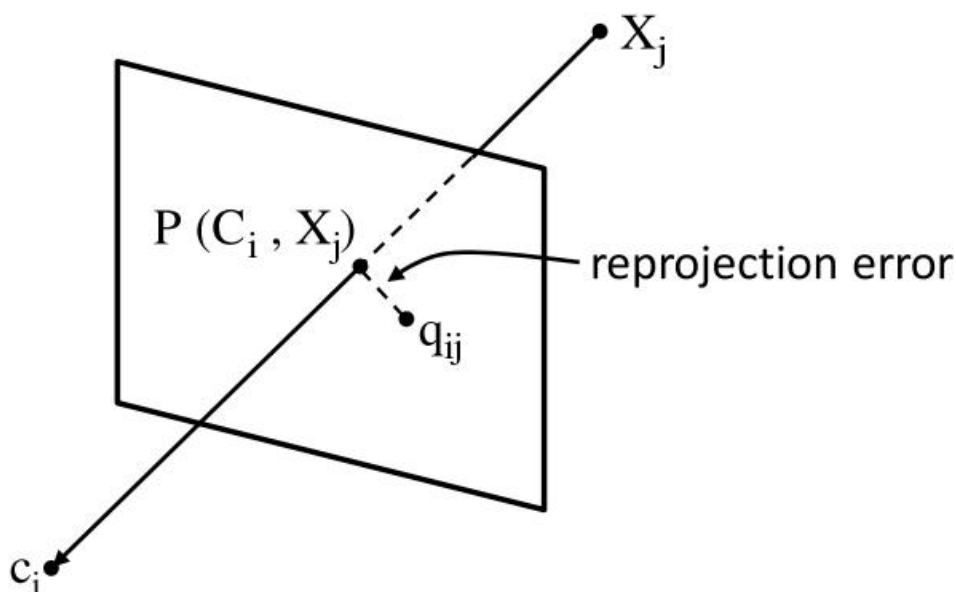
1. 获取下一最佳匹配图像 (correspondence search里给出)
2. 对新加入的最佳匹配图像进行pnp估计该位姿
3. 对新加入的图像进行三角化
4. 三角化后对所有已生成的三维点和已估计出的位姿进行ba优化 (ceres)

其实第一次投影指的就是相机在拍照的时候三维空间点投影到图像上的过程。然后我们利用这些图像对一些特征点进行三角定位 (triangulation，很多地方翻译为三角化或者三角剖分等等，当然笔者最喜欢的还是三角定位，显然是利用几何信息构建三角形来确定三维空间点的位置嘛，相关内容请参考对极几何)

最后利用我们计算得到的三维点的坐标 (注意不是真实的) 和我们计算得到的相机矩阵 (当然也不是真实的) 进行第二次投影，也就是重投影

3.2.7 关于光束平差算法的作用和投影误差的概念

描述摄像机的外参数用到33的旋转矩阵 R 和13的平移向量(或者摄像机中心坐标向量), 摄像机的内参数用一个焦距 f 和两个径向畸变参数 k_1 和 k_2 描述。几何场景提供轨迹中的每个3D点 X_j , 通过投影方程, 一个3D点 X_j 被投影到摄像机的2D图像平面上。投影误差就是投影点和图像上真实点之间的距离。如下图:



对于 n 个视角和 m 个轨迹, 投影误差的目标优化方程可以写为:

$$g(C, X) = \sum_{i=1}^n \sum_{j=1}^m w_{ij} \|q_{ij} - P(C_i, X_j)\|^2$$

当摄像机 i 观察到轨迹 j 的时候 w_{ij} 取1, 反之取0, $\|q_{ij} - P(C_i, X_j)\|$ 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, 著名的有光束平差法bundle adjustment.

4 Colmap实现稀疏重建

4.1 软件介绍

COLMAP是一款结合SfM(Structure-from-Motion)和MVS(Multi-View Stereo)的三维重建Pipeline, 编译成功后我们能获取带有图形界面的软件(Graphic Interface)和不需要图形界面的二进制可执行文件(Command-Line Interface), 可进行稀疏重建和稠密重建。源码链接及教程为: <https://github.com/colmap/colmap>。

以下稀疏重建过程均以colmap的图形界面软件为例。

- COLMAP的安装和简单使用:

<https://my.oschina.net/vision3d/blog/4493412>

(感谢以上博客作者整理的安装帮助)

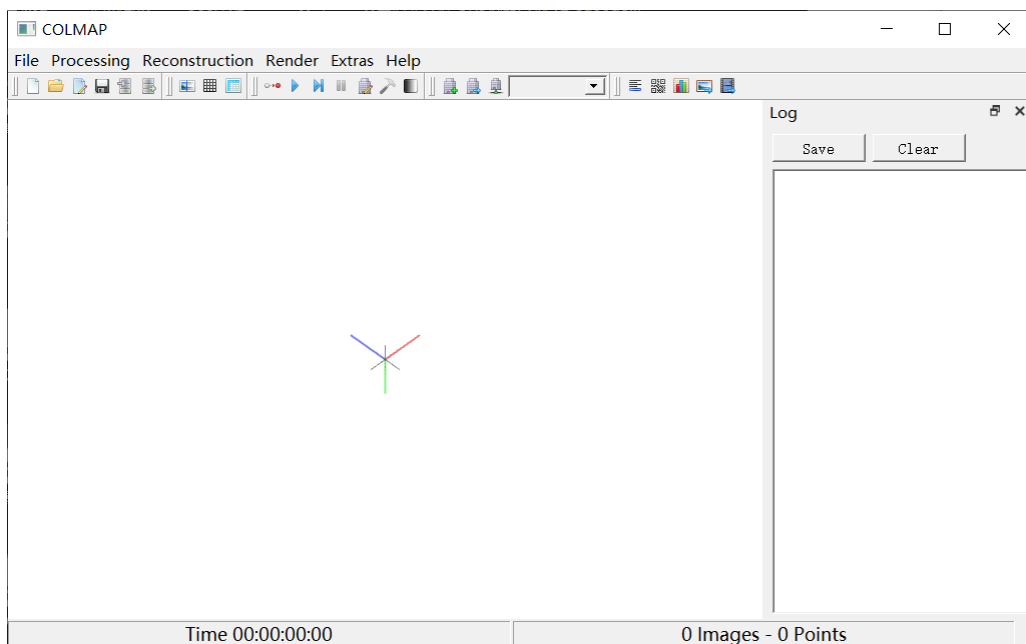
4.2 自采数据

- 尽量使用单反相机或专业数码相机进行数据采集，如果要用手机进行采集，请使用单摄像头的手机进行数据采集。
- 尽量选择纹理丰富的外界环境进行数据采集，避免玻璃围墙、瓷砖和打蜡地板等强反光材料环境
- 尽量选择光照明亮，且光照条件变化不剧烈的环境，最好选择室内环境。如室内客厅，开启客厅大灯进行灯光补偿。
- 尽量围绕重建物体或环境采集较多的影像，且在采集过程中控制快门速度，避免模糊。

4.3 稀疏重建

4.3.1 准备工作

1. 首先，我们需要准备几个目录：
 - 包含原始图像的目录（自行命名，如：images）
 - COLMAP工程的目录（自行命名，如：Project）
2. 通过命令行Linux：`~$ colmap gui` 或 直接打开COLMAP的图形界面，如图：



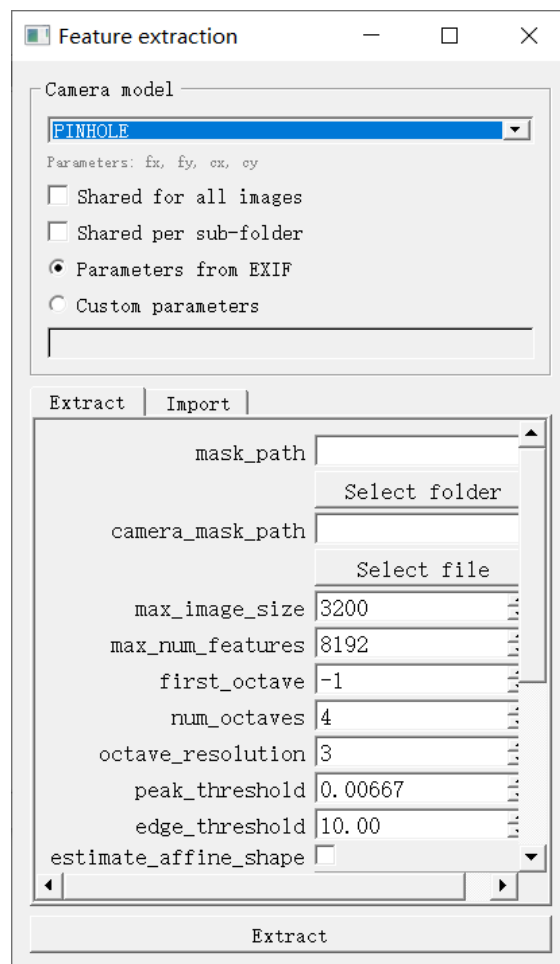
3. 新建项目：点击“file”，再点击“New Project”，弹出“project”窗口，点击“New”新建工程文件，将该工程文件保存在Project目录下，并点击“Select”选择场景原始图片所在的目录。最后点击“save”保存。

保存结束后，数据库将会显示.db文件，内部保存原始图片地址、之后特征提取匹配等数据。

4.3.2 特征提取

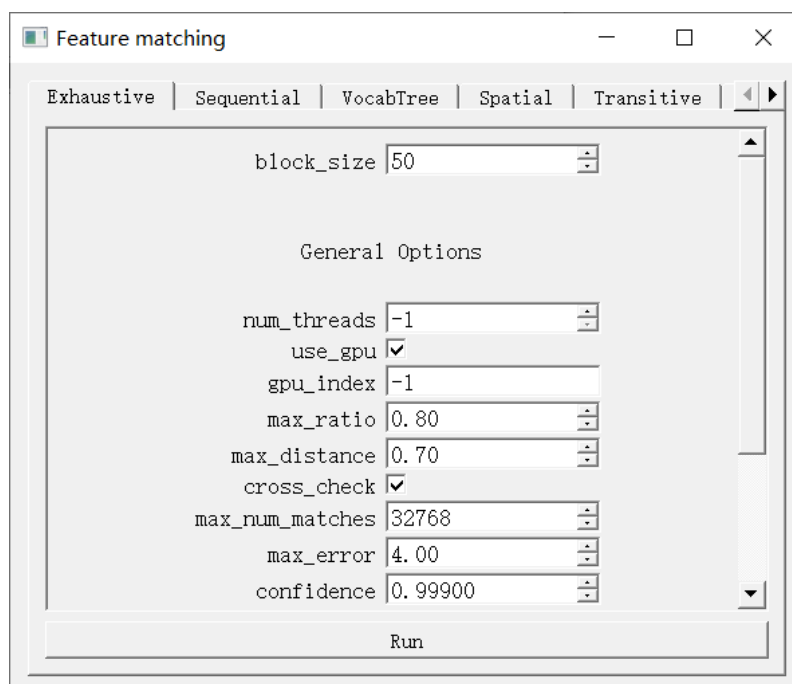
按照稀疏重建的步骤，需要我们进行对应点搜索，可以理解为全局的特征匹配。

1. 进行特征提取，点击“processing”中的“Feature Extraction”，弹出选择窗体，这里面，只需要将相机模型选择为“Pinhole”模型即可，其他参数默认可以不变。
2. 点击“Extract”即可进行特征提取。



4.3.3 特征匹配

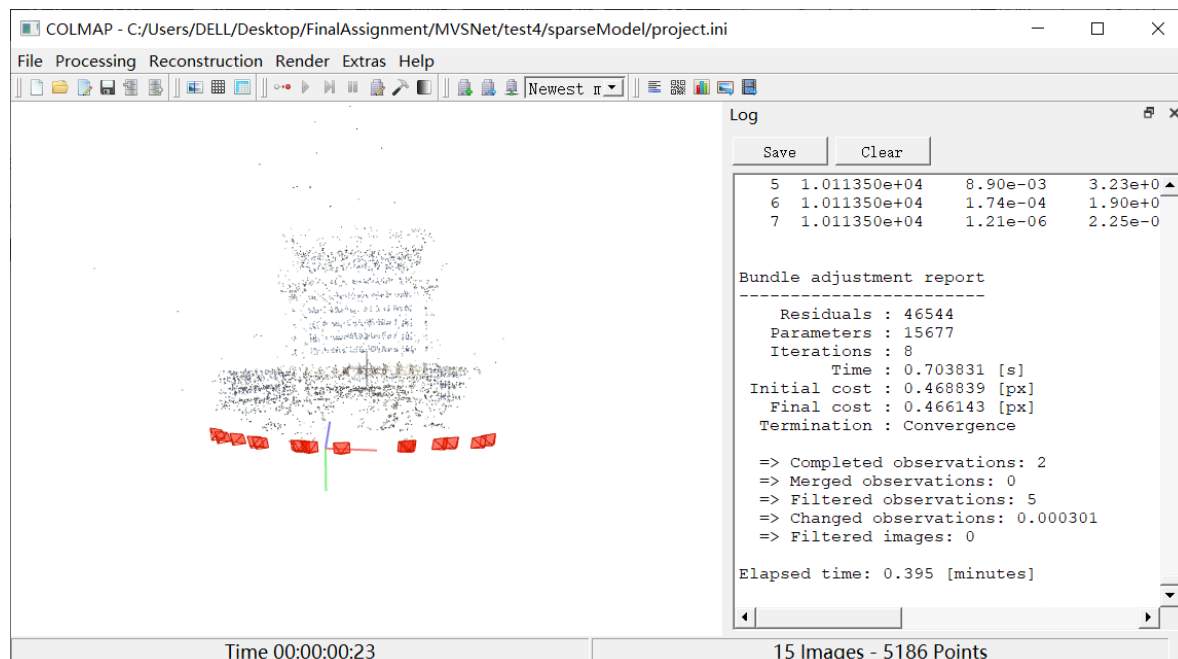
1. 点击“processing”中的“Feature Matching”，弹出选择窗体。里面的参数都可以选择默认的参数，然后点击“Run”，即可进行特征匹配



4.3.4 增量式sfm

增量式重建是个逐渐增加视角，并进行迭代优化重投影误差的过程。目的是计算不同视图的相机参数、得到场景的稀疏点云和确定不同视图与点云之间的可视关系。

1. 点击“reconstruction”中的“start reconstruction”进行一键式重建，整个过程将会自动进行增量式重建，我们可以从旁边的log框内查询当前状态。



如图，当前正在新增第2个视角，当前影像可以看到已有点云的914个，进行姿态估计(Pose Refinement Report)，再进行BA优化，最后进行三角测量(Retriangulation)新增观测点。上述过程结束后，进行迭代全局的BA优化，优化已有相机的姿态和三维稀疏点云坐标。

```
=====
Registering image #2 (8)
=====

=> Image sees 914 / 2290 points

Pose refinement report
-----
Residuals : 1720
Parameters : 8
Iterations : 20
Time : 0.0187179 [s]
Initial cost : 0.829007 [px]
Final cost : 0.602535 [px]
Termination : Convergence

=> Continued observations: 859
=> Added observations: 623

Bundle adjustment report
-----
Residuals : 21074
Parameters : 5417
Iterations : 17
Time : 0.450932 [s]
Initial cost : 0.433131 [px]
Final cost : 0.384505 [px]
Termination : Convergence

=> Merged observations: 0
=> Completed observations: 25
=> Filtered observations: 78
=> Changed observations: 0.009775

Bundle adjustment report
-----
Residuals : 20966
Parameters : 5378
Iterations : 3
Time : 0.0742637 [s]
Initial cost : 0.416612 [px]
Final cost : 0.406775 [px]
Termination : Convergence
```



```

Bundle adjustment report
-----
    Residuals : 20966
    Parameters : 5378
    Iterations : 3
        Time : 0.0742637 [s]
Initial cost : 0.416612 [px]
    Final cost : 0.406775 [px]
    Termination : Convergence

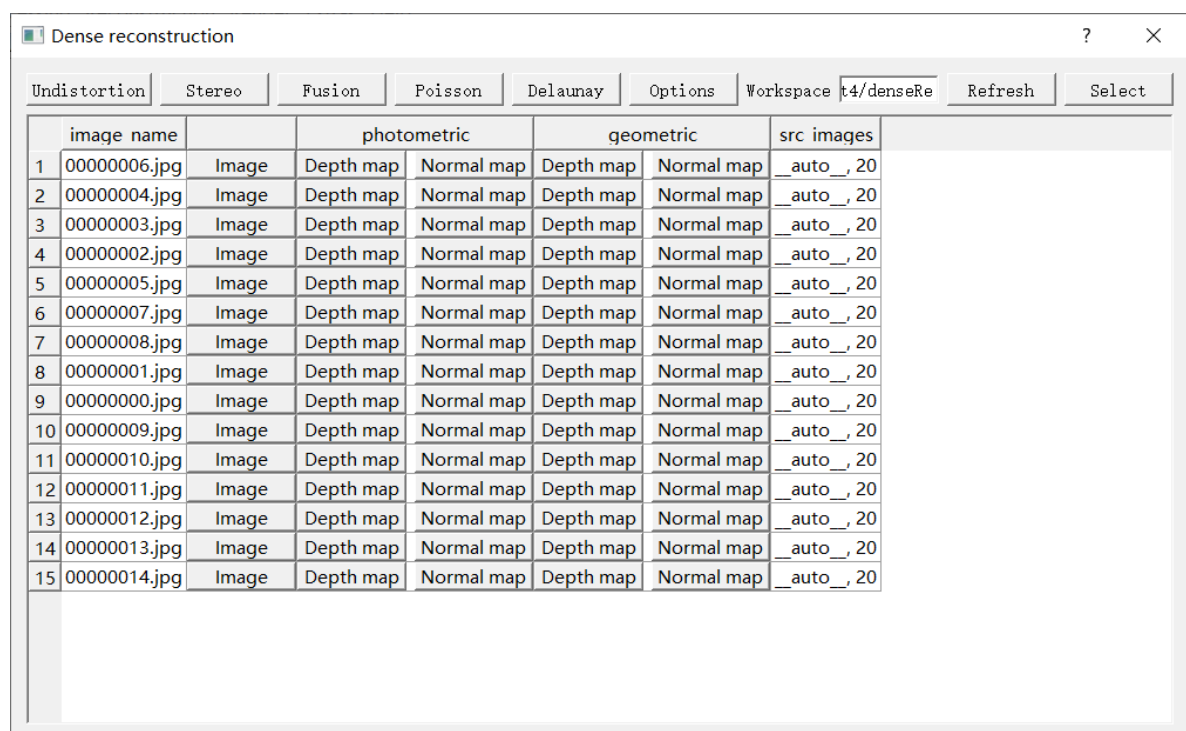
=> Merged observations: 0
=> Completed observations: 5
=> Filtered observations: 1
=> Changed observations: 0.000572

=====
Retriangulation
=====

```

4.3.5 图像去畸变

1. 点击“reconstruction”中的“dense reconstruction”. 打开稠密重建窗口，进行图像畸变纠正。
2. 在Project目录下新建子文件夹为denseRe，以Project/denseRe 为workspace地址，点击“undistortion”进行图像去畸变。



再

返回到Project下，可以看到当前的文件夹结构。这里的sparse是自动创建的保存相机数据和稀疏场景结构的文件夹。

	images	2021/6/19 9:59	文件夹	
	sparse	2021/6/19 9:59	文件夹	
	stereo	2021/6/19 9:59	文件夹	
	run-colmap-geometric.sh	2021/6/19 9:59	SH 文件	1 KB
	run-colmap-photometric.sh	2021/6/19 9:59	SH 文件	1 KB

4.3.6 导出相机参数和中间数据

1. 点击COLMAP中“file”的“export data as TXT”
2. 以上一步中得到的sparse/文件夹作为导出地址，导出三个文件：camera.txt, images.txt, points3D.txt.
 - o camera.txt是保存相机内参数的文件

- images.txt是保存图片外参数和图片二维特征与三维空间点对应信息的文件
- points3D.txt是保存三维空间点在世界坐标系下坐标、RGB值以及在各个影像上的轨迹(track)

4.3.7 导出稀疏重建模型

1. 点击COLMAP中“file”的“export model”，如想保存为点云格式（.ply）可以点击“export model as...”
2. 选择导出地址或导出格式，导出文件包含camera.txt, images.txt, points3D.txt,稀疏重建模型及其配置文件。