

2022~2023 学年下学期
近景摄影测量课程设计

实
习
报
告

学生姓名：殷雨薇

学 号：2020302131119

指导教师：李欣

专业及方向：遥感科学与技术（摄影测量）

时间：校历第 16 周（2023. 05. 29-2023. 06. 03）

地点：V-318 机房

2023 年 6 月 6 日

目 录

1 绪论

1.1 实习目的.....	1
1.2 实习内容.....	1
1.2.1 控制场影像获取.....	1
1.2.2 像点坐标量测.....	2
1.2.3 空间后方交会-前方交会	2
1.2.4 直接线性变换.....	2

2 控制场影像获取

2.1 过程及结果.....	3
2.1.1 相机参数选择.....	3
2.1.2 拍摄影像.....	3
2.1.3 实习成果.....	3
2.2 本章小结.....	5

3 像点坐标量测

3.1 原理.....	6
3.1.1 连通域灰度重心算法.....	6
3.1.2 Hough 圆检测算法.....	7
3.2 过程及结果.....	8
3.2.1 连通域灰度重心算法.....	8
3.2.2 Hough 圆检测算法.....	11
3.3 结果分析.....	12
3.4 本章小结.....	14

4 空间后方交会-前方交会

4.1 原理.....	15
4.1.1 坐标系转换.....	15
4.1.2 畸变差模型选择与参数初值估计.....	15
4.1.3 空间后方交会.....	16
4.1.4 精度计算.....	19
4.1.5 前方交会.....	19
4.2 过程及结果.....	20

4.2.1 坐标与参数初始化.....	20
4.2.2 空间后方交会计算内外方位元素与畸变系数.....	21
4.2.3 精度评定.....	22
4.2.4 空间后方交会计算结果.....	22
4.2.5 空间前方交会计算结果.....	22
4.3 结果分析.....	24
4.4 本章小结.....	24
5 直接线性变换	
5.1 原理.....	25
5.1.1 DLT 基本关系式	25
5.1.2 内外方位元素及 ds 和 $d\beta$ 求解	26
5.1.3 DLT 解算过程	28
5.1.4 DLT 精度检验	30
5.2 过程及结果.....	30
5.2.1 l 系数与畸变系数解算	31
5.2.2 内外方位元素解算.....	32
5.2.3 待定点物方坐标解算.....	34
5.2.4 物方坐标精度评定.....	34
5.2.5 活动控制架距离计算.....	35
5.3 结果分析.....	35
5.4 本章小结.....	36
结论	37
参考文献	38
附录 A 像点坐标量测结果	39

1 绪论

1.1 实习目的

- 1) 熟悉近景摄影测量控制场的组成，学习控制场布设的方法。
- 2) 熟悉近景摄影测量人工标志，照明条件等相关摄影测量原理和影像获取流程。
- 3) 理解光圈、快门、ISO 等相机相关参数及主距、焦距、超焦距、景深等重要概念的实际意义及其对影像质量的影响。
- 4) 使用单反相机、三脚架等摄影设备，拍摄清晰且利于摄影测量处理的控制场影像，为后续像点坐标量测程序提供影像数据。
- 5) 了解近景摄影测量中人工标志特点，并结合控制场中标志点分布确定最佳量测方案。
- 6) 量测所拍摄的室内三维控制场中的像点坐标，体会影像质量，影像参数等对像点坐标量测精度的影响。
- 7) 掌握 Hough 变换、椭圆拟合、连通域灰度重心算法等相关计算机视觉技术，将其应用于控制场影像实现自动像点量测。
- 8) 加深理解近景摄影测量单像空间后方交会的理论，结合像点量测程序输出结果编写代码实现单像空间后方交会。
- 9) 加深理解近景摄影测量直接线性变换（DLT）的理论，结合像点量测程序输出结果编写代码实现直接线性变化。

1.2 实习内容

1.2.1 控制场影像获取

- 1) 听李老师讲解解控制场布设要求及布设原理，单反相机摄取控制场影像的基本流程及相机使用细节。
- 2) 对单反相机进行探索，熟悉各个部件的功能，为后续拍摄影像做准备。
- 3) 采用三脚架固定并调整相机参数，获取两张控制场影像，选取左右影像构成立体像对。
- 4) 记录相机编号、影像编号、相机参数及近似摄影中心位置。

1.2.2 像点坐标量测

- 1) 基于 Matlab R2021a 实现控制点坐标精确提取。
- 2) 使用示例影像进行算法测试，再根据自行拍摄的控制场影像调整参数，修改代码，进一步筛选出最佳算法。
- 3) 通过控场点编号分布规律手动为提取的像点实现编号，最后将量测的子像素级像点坐标与其对应的点号保存在 txt 文件中。

1.2.3 空间后方交会-前方交会

基于摄取的控制场影像及像点量测程序获得左右影像中标志点对应像点坐标，将像点坐标及控制点空间坐标作为输入，编写代码实现空间后方交会，获取影像内外方位元素及畸变系数。并通过前方交会求解待定点空间坐标，与 DLT 计算结果做对比。

1.2.4 直接线性变换

以控制点像点坐标及控制点空间坐标作为输入，编写代码实现直接线性变换 (DLT)，获取影像内外方位元素、L 系数、畸变系数，根据所得系数求解检查点空间坐标及活动控制场待定点空间坐标。最后，将检查点计算所得空间坐标作为观测值与真值求差统计精度及残差，利用所得活动控制场待定点求解每个点到 52 号点的空间距离。

2 控制场影像获取

2.1 过程及结果

2.1.1 相机参数选择

1) 设置光圈。在近景摄影测量的影像获取过程中, 光圈对超焦距、景深具有关键影响。旋转相机右上部的旋钮, 转动到 **Av** 档, 即光圈优先模式。选择较小的光圈能够获得景深更大的影像, 光圈太大则景深小, 将造成部分像点模糊不清。因此, 在保证成像清晰的前提下调整为 **F7.1**。

2) 设置焦距。焦距的选择决定了最终摄取的控制场影像的视场范围大小, 具体而言, 如果焦距较大, 则照片视场角窄无法包含足够数量的控制点, 不利于后续选点进行像点坐标量测; 焦距过小则视场角过大, 在影片边缘将出现明显的几何畸变, 对其进行几何纠正将降低数据处理效率而且会对像点量测精度产生负面影响。除此之外, 单反相机在变焦即对焦过程中焦距都会发生变化。对于近景摄影测量而言, 需要保证焦距不变, 因此在摄影过程中首先需要将相机对焦模式切换到 **MF** 即手动对焦, 同时在摄影时要避免触碰变焦环导致焦距改变。

3) 设置 **ISO**。为了使摄取的控制场影像更加清晰, 减少噪声影响, 将 **ISO** 调整为 **400**。

2.1.2 拍摄影像

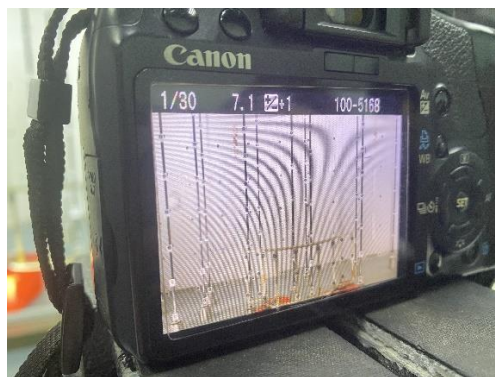
首先, 在距离控制场坐标原点具有一定距离的地方架设三脚架, 将单反相机固定在三脚架上, 检查确保三脚架与相机固定无松动后, 轻按快门即可完成自动测光摄取控制场影像。使用米尺粗略测量摄站位置作为后续摄影测量处理的初值。将相机及三脚架调整到另一摄影位置实现交向摄影。移动过程中保证相机焦距和相机姿态不变。移动完成后旋转三脚架, 使两摄站主光轴产生一定夹角。使用米尺粗略测量摄站位置作为后续摄影测量处理的初值。最后, 我们将相机参数包括相机编号、影像编号、光圈、焦距、**ISO** 进行详细记录。左、右影像拍摄记录如图 1 所示。

2.1.3 实习成果

最终使用相机 **Canon EOS 450D** 获取的控制场立体像对左右影像如图 2 所示, 具体参数信息可在文件属性中查看。其中, 曝光时间均为 **1/30** 秒, 焦距均为 **25mm**。

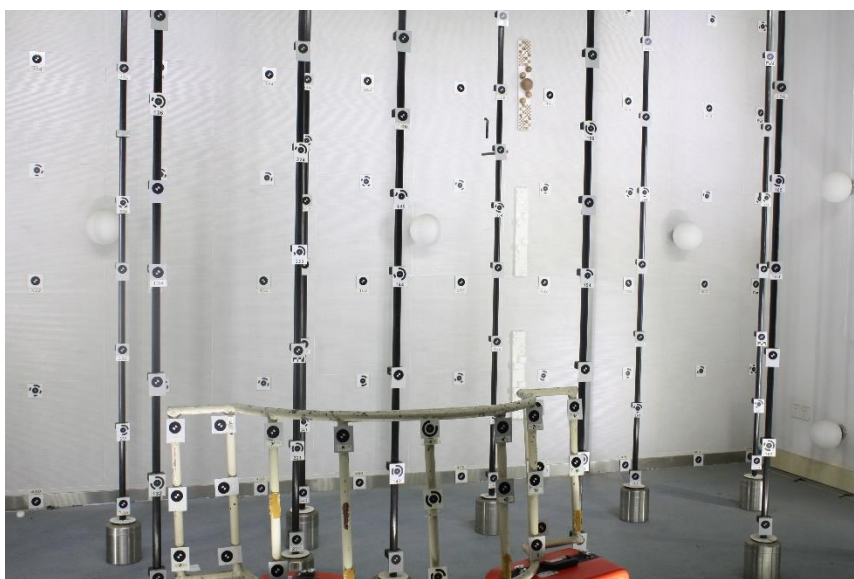


(a) 左

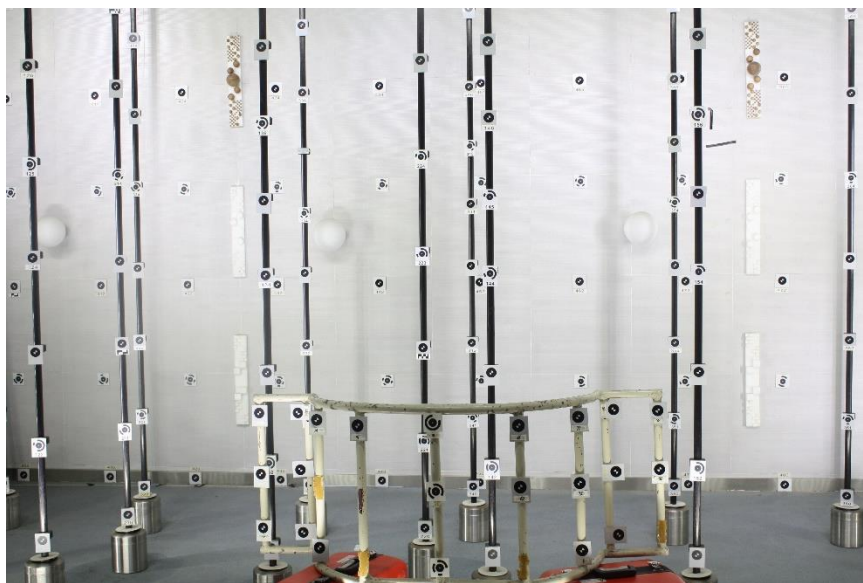


(b) 右

图 1 影像拍摄记录



(a) 左影像 IMG_5167.JPG



(b) 右影像 IMG_5168.JPG

图 2 获取的立体像对

2.2 本章小结

本阶段实习采用单反相机 Canon EOS 450D 完成了控制场影像获取，使我对近景摄影测量控制场布设规则和近景摄影测量影像获取流程有了更加深刻的理解。最终获得的立体像对曝光合适，景深较大，包含足量的控制点且像点清晰度高，影像间重叠度较大符合后续处理要求。由于场地限制，摄影基线可能不够长，应该进一步增大摄影基线。

3 像点坐标量测

3.1 原理

在本实验中，像点坐标量测的关键在于标志点中心定位环节，可以使用的算法包括椭圆拟合算法，连通域灰度重心算法，霍夫圆检测算法等。本实验采用了后两种算法。最终实验结果表明，在本实验场景中，连通域灰度重心算法较为鲁棒。本节将详细介绍连通域灰度重心算法和 Hough 圆检测算法的具体原理。

3.1.1 连通域灰度重心算法

连通域灰度重心算法是一种基于图像处理的物体测量技术，它可以用于对物体的几何特征进行测量和分析。该算法主要基于图像二值化和灰度重心的原理，可以计算出物体的位置、面积和形状等信息，适用于对简单或复杂形状的物体进行测量和分析。

该算法首先对图像进行二值化处理，将物体与背景区分出来，然后进行连通域分析，将相邻像素区域内的像素分为同一连通域。接着，对每个连通域计算其灰度重心，即将连通域内每个像素的位置与像素的灰度值加权平均，得到连通域的灰度重心。最后，通过连通域的灰度重心可以计算出物体的位置和面积等信息。

在计算灰度重心时，每个像素的位置和灰度值被视为其空间和灰度空间中的一个向量，可以将其表示为一个 n 维向量，其中 n 为像素的数量。通过对每个向量进行加权平均，即可得到连通域的灰度重心。灰度重心的计算方法具有较高的计算精度和鲁棒性，可以有效地应用于不同形状和大小的物体测量中。

连通域灰度重心算法适用于对图像中单个物体或多个物体进行测量和分析，可以用于实现自动化检测、分类和识别等应用。该算法具有较高的计算精度和鲁棒性，但在处理复杂或重叠的物体时可能存在误差，需要进一步优化和改进。

本实验中的基本步骤如下：

- 1) 灰度影像高斯滤波：影像转灰度图并应用高斯滤波去噪；
- 2) 二值化：OSTU 方法（最大类间方差法）选择合适阈值进行二值化；
- 3) 开操作：用于消除图像噪点；
- 4) 连通区域提取：提取影像所有连通区域，通过限制区域面积、离心率与最小正外接矩形的形状，把控制点所在的圆/椭圆作为连通区域提取出来；

5)重心提取:提取出来连通区域的重心,采用 Matlab 的内置函数进行求取。

3.1.2 Hough 圆检测算法

Hough 圆检测算法是 Hough 变换在圆检测中的应用。与 Hough 直线检测类似, Hough 圆检测也是将图像中的圆转换为在参数空间(霍夫空间)中的点,通过检测在霍夫空间中的峰值来确定圆。该算法的基本原理是将图像中的每个边缘点,都假设为一个圆心,然后找出以该点为圆心的所有可能的圆。如果多个边缘点检测到同一个圆,则该圆的圆心坐标和半径会被统计累加,从而得到最终的圆心坐标和半径。

对直线来说,一条直线能由极径极角 (r, θ) 表示,而对于圆来说,我们需要三个参数:圆心 (a, b) , 半径 r 。笛卡尔坐标系中圆的方程如公式(1)。

$$(x - a)^2 + (y - b)^2 = r^2 \quad (1)$$

将其转化到极坐标系简化可得到如公式(2)。

$$\begin{aligned} a &= x - r \cos \theta \\ b &= y - r \sin \theta \end{aligned} \quad (2)$$

对于 (x_0, y_0) , 我们可以将通过这一点的所有圆统一定义如式(3):

$$\begin{aligned} a &= x_0 - r \cos \theta \\ b &= y_0 - r \sin \theta \end{aligned} \quad (3)$$

这就意味着每一组 (a, b, r) 代表一个通过点 (x_0, y_0) 的圆。对于一个给定点 (x_0, y_0) , 我们可以在三维直角坐标系中, 绘出所有通过它的圆。最终我们将得到一条三维的曲线。我们可以对图像中所有的点进行上述操作, 如果两个不同点进行上述操作后得到的曲线在空间 $a-b-r$ 相交, 即它们有一组公共的 (a, b, r) , 这就意味着它们在同一个圆上。越多曲线交于一点, 也就意味着这个交点表示的圆由更多的点组成。我们可以设置一个阈值, 来决定多少条曲线交于一点我们才认为检测到了一个圆。这就是霍夫圆变换要做的, 它追踪图像中每个点对应曲线间的交点, 如果交于一点的曲线的数量超过了阈值, 那么可以认为这个交点所代表的参数 (a, b, r) 在原图像中为一个圆。

在本实验中的基本步骤如下:

- 1) 灰度影像高斯滤波: 影像转灰度图并应用高斯滤波去噪;
- 2) 边缘检测: 使用 Canny 算子检测边缘点;

3) Hough 变换：对边缘点进行 Hough 变换，在极坐标空间中建立累加器，统计每个圆心点的可能半径范围；

4) 初次筛选：对累加器进行高斯平滑处理和非极大值抑制，找出可能的圆心点和半径。

本实验中，Hough 圆检测算法主要基于 MATLAB 中的 `imfindcircles` 函数来实现，该函数可以检测出图像中的多个圆形，并返回其圆心坐标和半径等信息。经典 Hough 变换需要三维数组来存储多个半径的投票，这需要很大的存储空间和很长的处理时间。`imfindcircles` 提供了两种计算累加器数组的计算方法，相位编码方法和两阶段方法，都通过对所有半径使用同一个二维累加器数组来解决此问题。虽然这样做需要额外的半径估计步骤，但总体计算负载通常较低，尤其是在处理大半径范围时。这是目前基于圆形 Hough 变换实现中广泛采用的做法。

3.2 过程及结果

本节以左影像为例叙述像点坐标量测的过程，并展示左右像片的像点坐标量测结果。

3.2.1 连通域灰度重心算法

1) 读取原始影像并显示，如图 2 (a)。

2) 将原始影像转换为灰度图，进行高斯滤波去除噪声，结果如图 3。



图 3 灰度高斯滤波影像

3) 采用 OSTU 方法（最大类间方差法），对影像进行二值化。主要使用到的函数包括 `graythresh`（计算二值化阈值）、`im2bw`(计算二值化图像)，结果如图 4。

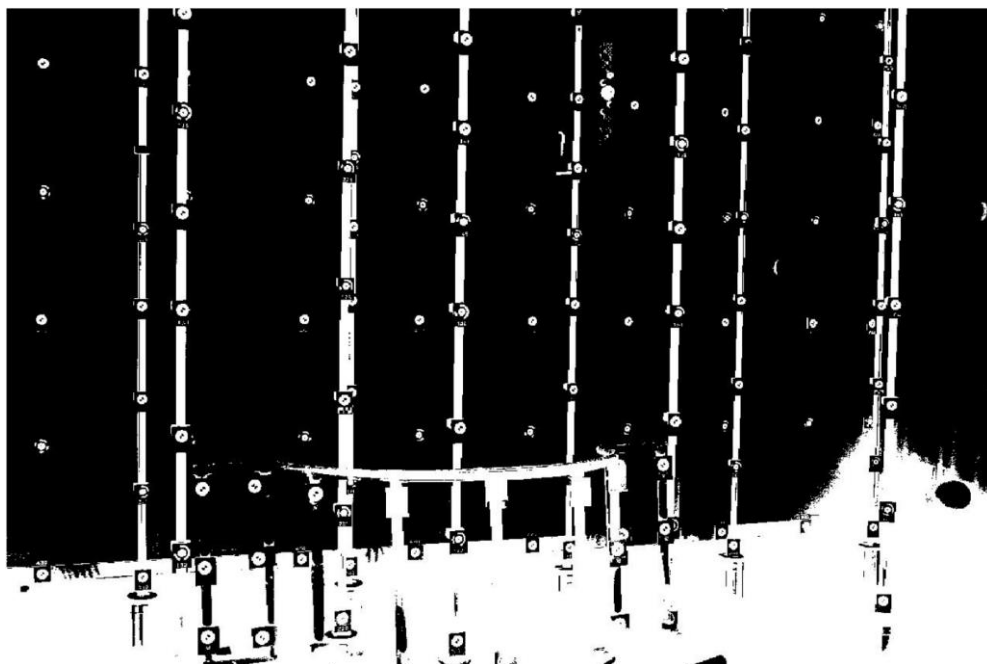


图 4 OSTU 二值化结果

4) 开操作消除图像中的噪点。调用 `strel` 函数进行创建 `disk` 圆盘结构元素，使用 `imopen` 函数把开元素构造元素指定形状圆盘对二值化后的图像进行处理，结果如图 5。

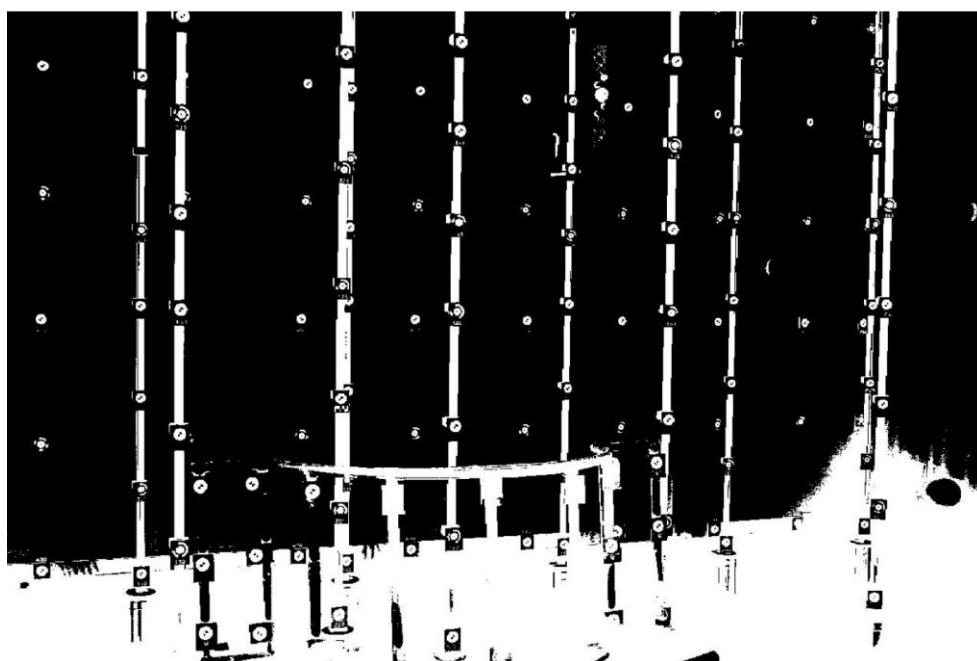


图 5 开操作结果

5) 获取连通区域，并显示彩色化标定点图像。首先调用函数 `imclearborder` (边界对象抑制函数)，删除和图像边界相连的对象。再调用 `bwareaopen` 函数，删除二值图像 `bw` 中面积小于 300 的对象，默认情况下使用 8 邻域。调用 `bwlabel` 函数，获取 `bw` 中的连通区域储存在 `L` 中，`num` 代表连通区域个数。最后调用 `label2rgb` 函数，将连通域进行彩色可视化。具体代码如图 6，可视化结果如图 7。

```
bw=imclearborder(bw, 4);
bw=bwareaopen(bw, 300);
[L, num]=bwlabel(bw, 8);
RGB = label2rgb(L);
figure;
imshow(RGB);
title('connected components rgb');
```

图 6 获取连通区域相关代码

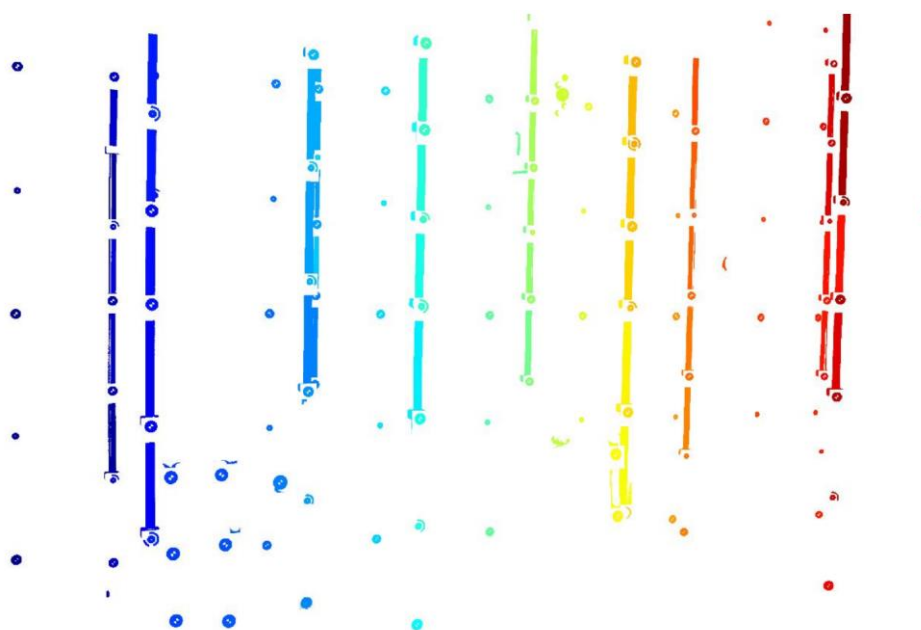


图 7 连通区域提取结果

6) 获取连通区域的 `basic` 属性，限定条件进行重心的寻找与标定。调用函数 `regionprops` 获取连通区域的 `Area/Centroid/BoundingBox/PixelList/Eccentricity` 属性，其中 `Area` 是图像各个区域中像素总个数，`BoundingBox` 是包含相应区域的最小矩形，`Centroid` 是每个区域的质心（重心），`PixelList` 存储上述索引对应的像素坐标，`Eccentricity` 是每个连通区域的偏心率。对于本实验中使用的拍摄设备与影像，限定像素面积 20~3000，偏心率<0.8 可获得较为理想的像点对应连通区域提取结果，并用 `rectangle` 标定符合要求的连通域的位置，结果如图 8。

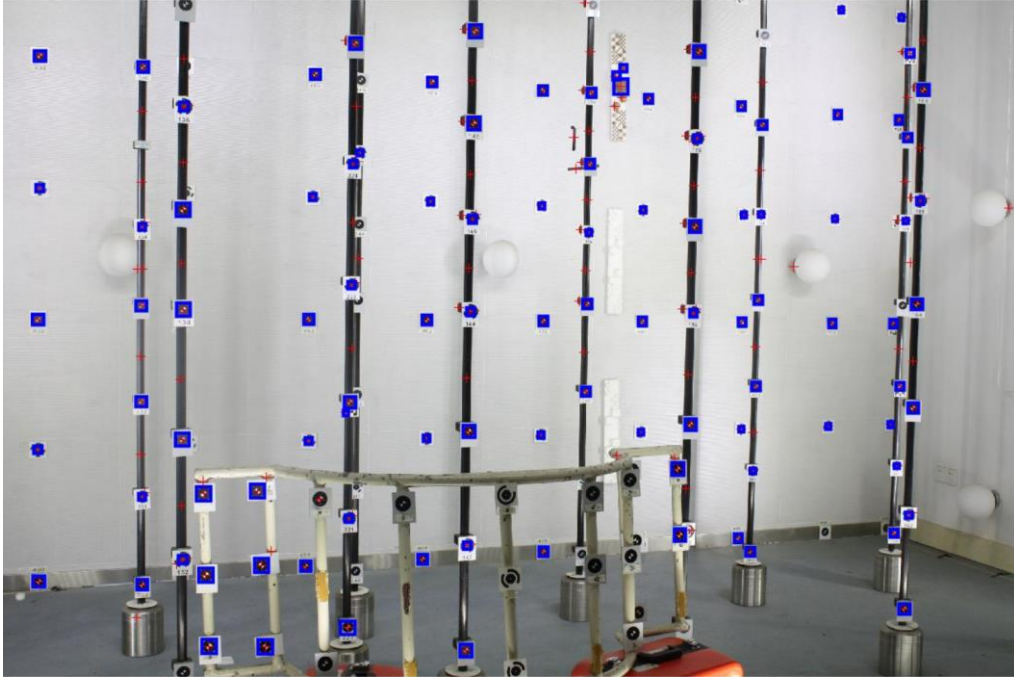


图 8 像点提取结果

3.2.2 Hough 圆检测算法

- 1) 与上一个算法相同，读取原始影像并显示。
- 2) 将原始影像转换为灰度图，进行高斯滤波去除噪声，结果如图 3。
- 3) 使用 Canny 算子对图像进行边缘检测，阈值参数设置为 0.3，结果如图 9。

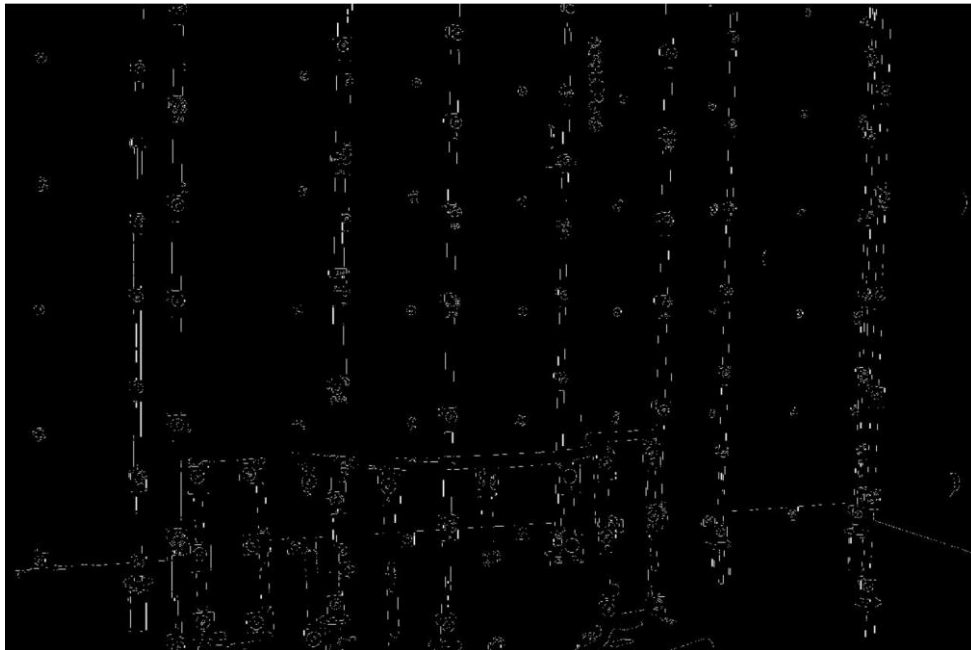


图 9 Canny 边缘检测结果

- 4) 对边缘点进行 Hough 变换，在极坐标空间中建立累加器，统计每个圆心

点的可能半径范围。调用 `imfindcircles` 函数，半径范围设置为 (10, 600)，为了检测出更多的圆，这里将 `Sensitivity` 设置为 1，得到圆的圆心坐标 `centers`，半径大小 `radii` 和累加器数组 `metric`，如图 10。

```
[centers, radii, metric] = imfindcircles(BW, [10 600], 'Sensitivity', 1);
```

图 10 Hough 变换函数的输入输出

5) 对累加器数组进行高斯平滑处理和非极大值抑制，找出可能的圆心点和半径。由于是初次筛选，可以将累加器数组中大于阈值的对应圆心点和半径都保留，经过多次实验，这里设置阈值为累加器数组最大值的一半。在图像中用蓝色曲线绘制出符合条件的圆，并用红色十字标志出圆心位置，圆心检测结果如图 11。

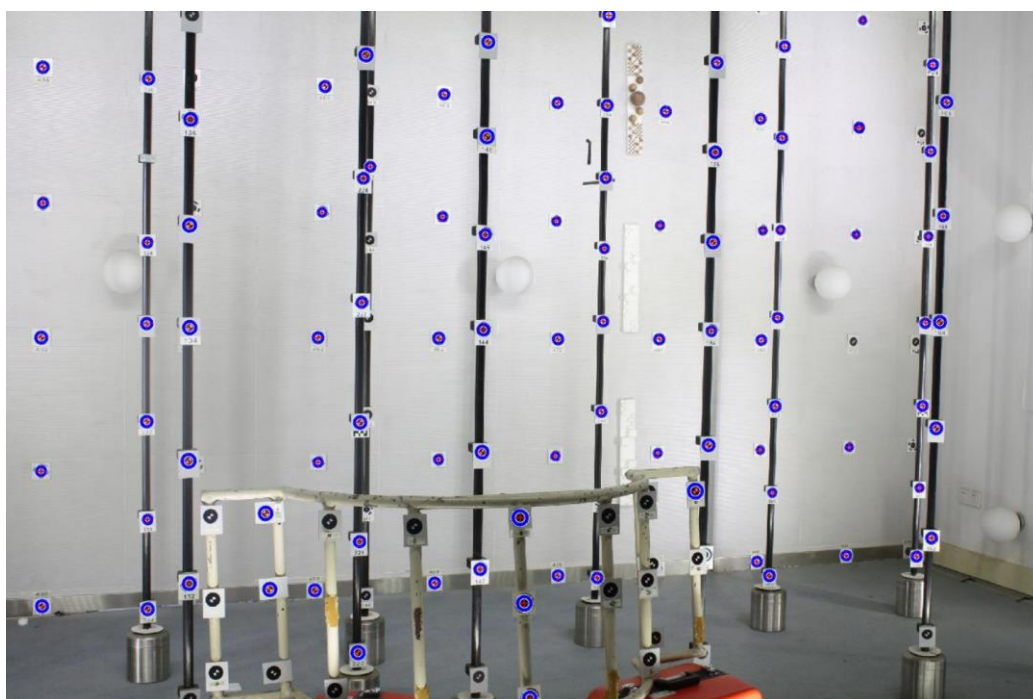


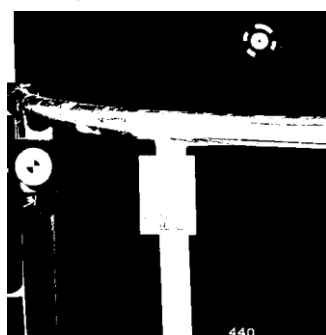
图 11 圆心检测结果

6) 通过对实验结果的分析比较，发现连通域灰度重心算法得到的子像素级像点坐标精度更高。因此对该方法得到的像点，按照控制场像点编号规则，按编号由小到大的顺序，将像点编号和像素坐标依次填入 txt 文件中，见附录 A。左、右影像分别量测出 98 和 115 个像点坐标。

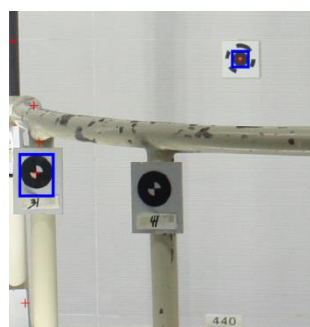
3.3 结果分析

1) 影像拍摄光线条件对提取结果有一定影响。由于控制场光线条件不是很理想，造成部分标志点在影像中过暗或者过亮，导致影像二值化出现问题，最终

未识别到该部分点，如图 12。



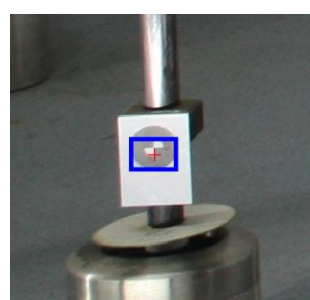
(a) 二值化影像



(b) 提取结果



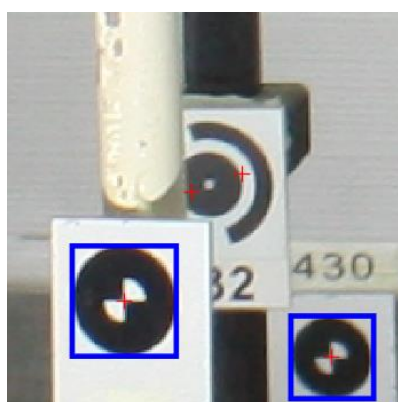
(c) 二值化影像



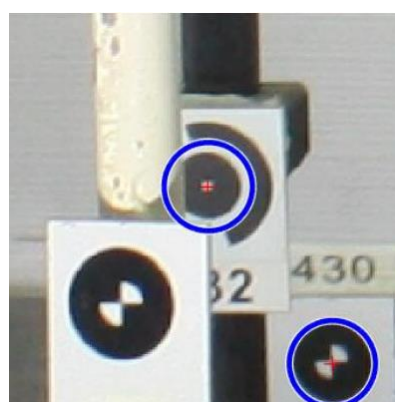
(d) 提取结果

图 12 光线过暗或过亮导致提取失败

2) 影像拍摄角度对提取结果有一定影响。角度不同造成部分标志点有不同程度的遮挡，在灰度重心算法中，连通区域连接关系受到影响，从而导致部分点未识别出来。但在 Hough 圆检测算法中，由于圆心点提取是按照累加器数组的大小来判断的，累加器数组值越大，其对应的点为圆心点的概率就越大。因此，将阈值调到一个较低的值即可检测出部分遮挡的圆形标志点的圆心坐标。在此情况下，两种算法的对比如图 13。角度不同还会造成圆形控制点在影像中的形状发生变形，在这种情况下，灰度重心算法量测像点的精度更高，如图 14。

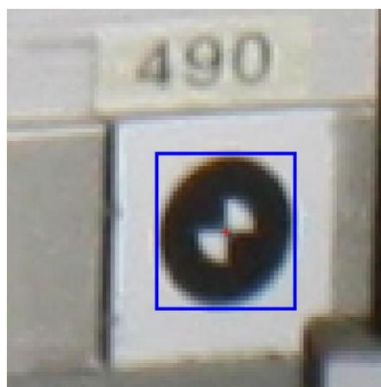


(a) 连通域灰度重心算法



(b) Hough 圆检测算法

图 13 遮挡对提取结果的影响



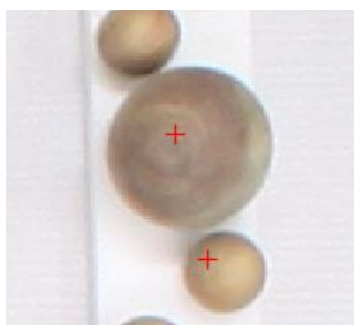
(a) 连通域灰度重心算法



(b) Hough 圆检测算法

图 14 像点量测精度比较

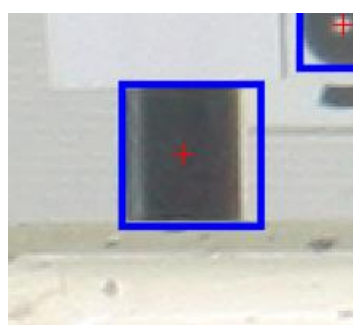
3) 对比两种算法的像点提取结果发现，灰度重心算法提取到的像点更多，两张影像均能达到 200 个左右，但错误率也较高，不是待量测像点的区域也提取出了重心，甚至标志为感兴趣区，如图 15。而对于 Hough 圆检测算法，由于控制场标志点均为圆形，在场景中较为特殊，使用该算法更具有针对性，能更为准确地识别出标志点。



(a)



(b)



(c)

图 15 灰度重心算法错误检测

3.4 本章小结

本阶段实习使用 Matlab 自行编制了像点坐标量测程序，让我对像点坐标量测的原理有了更加深刻的理解。通过查阅资料，我了解到除了课上老师所讲的椭圆拟合算法，还有连通域灰度重心算法、Hough 圆检测算法等。我对这些算法进行了尝试，最终通过对比实验结果，选出了效果较好的连通域灰度重心算法，进行最终的像点量测工作。连通域灰度中心算法的量测结果精度较高，符合人眼检测结果，可用于后续近景摄影测量处理。

4 空间后方交会-前方交会

4.1 原理

4.1.1 坐标系转换

根据提供的点号与物方三维坐标的变换规律,可以知道控制场坐标系定义是一个以门口测墩为原点、两测墩基线为 Y 轴,控制场前后方向为 Z 轴的左手坐标系,如图 16 (a)。若为正直摄影,主光轴平行于 X 轴。而对于一般基于共线方程推导的空间后方交会关系式均是右手系,且正直摄影时,主光轴一般朝向 Z 轴负向,如图 16 (b)。为了便于使用解析摄影测量的相关公式,在控制场三维空间坐标读入和输出处理时均需进行一定的转换,转换后的坐标系如图 17 所示。

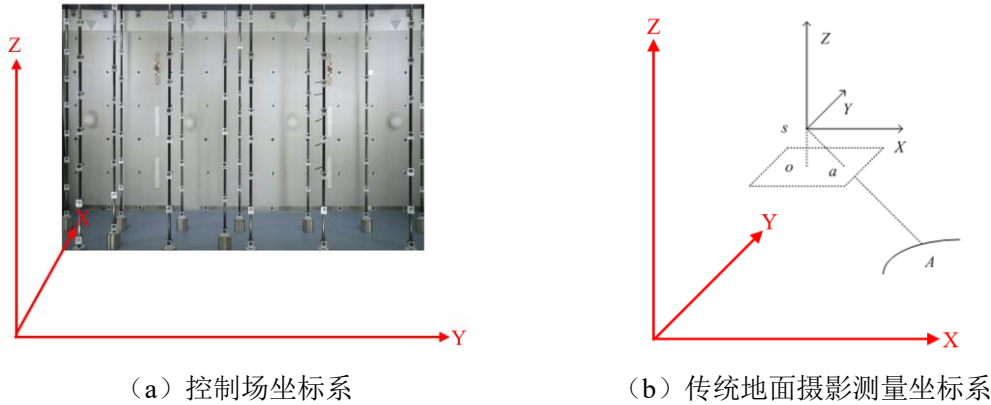


图 16 两种三维空间坐标系

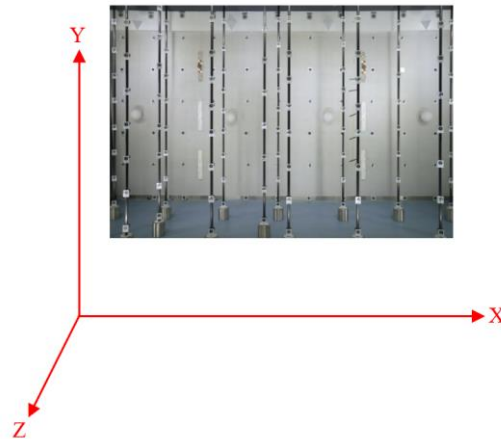


图 17 转换后的控制场坐标系

4.1.2 畸变差模型选择与参数初值估计

按照 Moffitt 和 Mikhail 的定义,空间后方交会是根据照片中出现的地面控制点图像的摄影测量结果来确定照片的空间位置和方向的过程。传统上,一张照片

的空间位置和方向由测站的三个坐标和三个旋转参数表示。除此之外，本阶段实习还需要拓展误差方程，统一解算内方位元素（像主点中心偏移 x_0, y_0 ，相机主距 f ）、外方位元素（ $X_S, Y_S, Z_S, \varphi, \omega, \kappa$ ）及 4 个畸变系数 k_1, k_2, p_1, p_2 。因此，所选畸变差模型如式 4。

$$\begin{cases} \Delta x = (x - x_0)(k_1 r^2 + k_2 r^4) + p_1[r^2 + 2(x - x_0)^2] + 2p_2(x - x_0)(y - y_0) \\ \Delta y = (y - y_0)(k_1 r^2 + k_2 r^4) + p_1[r^2 + 2(y - y_0)^2] + 2p_2(x - x_0)(y - y_0) \end{cases} \quad (4)$$

其中， (x, y) 为像主点为原点的像点量测坐标， k_1, k_2 表示待定对称径向畸变系数， p_1, p_2 表示待定切向畸变系数， r 表示像点向径，满足 $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ 。

对于内方位元素初值， f 可取实际拍摄时的相机焦距参数 25mm 。因为计算时使用的像平面坐标是以像主点为原点的，因此像主点中心偏移 x_0, y_0 取为 0。外方位元素初值均是由像片拍摄时实地量测记录的数据估计而来的。因为拍摄时使用的是 Canon EOS 450D 相机，畸变系数一般很小，故这里均设置为 0 即可。

4.1.3 空间后方交会

单像空间后方交会为基于共线条件方程的近景像片解析处理方法，解析其具体原理应该从共线条件方程出发。一般地，共线条件方程具有如式（5）的形式：

$$\begin{aligned} x - x_0 + \Delta x &= -f \frac{a_1(X - X_S) + b_1(Y - Y_S) + c_1(Z - Z_S)}{a_3(X - X_S) + b_3(Y - Y_S) + c_3(Z - Z_S)} = -f \frac{\bar{X}}{\bar{Z}} \\ y - y_0 + \Delta y &= -f \frac{a_2(X - X_S) + b_2(Y - Y_S) + c_2(Z - Z_S)}{a_3(X - X_S) + b_3(Y - Y_S) + c_3(Z - Z_S)} = -f \frac{\bar{Y}}{\bar{Z}} \end{aligned} \quad (5)$$

为了进行最小二乘平差处理，需要将共线条件方程式转换为共线方程式像点坐标误差方程式一般式。此时，公式（5）可改化如式（6）：

$$\begin{aligned} x &= \left(x_0 - f \frac{\bar{X}}{\bar{Z}} - \Delta x \right) \\ y &= \left(y_0 - f \frac{\bar{Y}}{\bar{Z}} - \Delta y \right) \end{aligned} \quad (6)$$

平差处理过程中，因存在多余观测值（即存在观测值的改正数 v_x, v_y ），而且计算过程是一个迭代计算过程（即存在近似值差值 d_x, d_y ），故式（6）应该写式（7）：

$$\begin{aligned}
x + v_x &= \left(x_0 - f \frac{\bar{X}}{\bar{Z}} - \Delta x \right) + d_x = (x) + d_x \\
y + v_y &= \left(y_0 - f \frac{\bar{Y}}{\bar{Z}} - \Delta y \right) + d_y = (y) + d_y
\end{aligned} \tag{7}$$

其中， (x) ， (y) 是前一次迭代运算结果的近似值，由此有像点坐标改正数方程式如式 (8)：

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} d_x \\ d_y \end{bmatrix} - \begin{bmatrix} x - (x) \\ y - (y) \end{bmatrix} \tag{8}$$

考虑畸变系数并将物方点空间坐标未知数分作控制点未知数 X_c 与待定点未知数 X_u 两组，对误差方程取泰勒级数展开的一次项式，最终得到像点坐标误差方程一般式 (9)：

$$\mathbf{V} = \mathbf{A}\mathbf{t} + \mathbf{B}_c \mathbf{X}_c + \mathbf{B}_u \mathbf{X}_u + \mathbf{C}\mathbf{X}_2 + \mathbf{D}\mathbf{X}_{ad} - \mathbf{L} \tag{9}$$

对于空间后方交会，已知物方控制点，不要求解求待定点物方坐标，则 $\mathbf{X}_c = \mathbf{X}_u = 0$ ，将控制点像点坐标量测结果作为观测值，解求的未知数为内方位元素改正数 \mathbf{X}_2 、外方位元素改正数 \mathbf{t} 与畸变系数改正数 \mathbf{X}_{ad} 共 13 个参数。

将量测的像点坐标同时列出误差方程，其矩阵形式如式 (10)：

$$\mathbf{V} = \mathbf{A}\mathbf{t} + \mathbf{C}\mathbf{X}_2 + \mathbf{D}\mathbf{X}_{ad} - \mathbf{L} \tag{10}$$

对于单点，系数矩阵与常数矩阵的维度为： $\mathbf{A}_{2 \times 6}$ 、 $\mathbf{C}_{2 \times 3}$ 、 $\mathbf{D}_{2 \times 4}$ 、 $\mathbf{L}_{2 \times 1}$ ，像片旋转矩阵定义如式 (11)：

$$\mathbf{R} = \mathbf{R}_\varphi \mathbf{R}_\omega \mathbf{R}_\kappa = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \tag{11}$$

其中， $a_1 = \cos \varphi \cos \kappa - \sin \varphi \sin \omega \sin \kappa$

$$a_2 = -\cos \varphi \sin \kappa - \sin \varphi \sin \omega \cos \kappa$$

$$a_3 = -\sin \varphi \cos \omega$$

$$b_1 = \cos \omega \sin \kappa$$

$$b_2 = \cos \omega \cos \kappa$$

$$b_3 = -\sin \omega$$

$$c_1 = \sin \varphi \cos \kappa + \cos \varphi \sin \omega \sin \kappa$$

$$c_2 = -\sin \varphi \sin \kappa + \cos \varphi \sin \omega \cos \kappa$$

$$c_3 = \cos \varphi \cos \omega$$

$$\text{外方位元素系数矩阵: } \mathbf{A}_i = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \end{bmatrix},$$

$$\text{若记 } \bar{X}_i = a_1(X_i - X_S) + b_1(Y_i - Y_S) + c_1(Z_i - Z_S)$$

$$\bar{Y}_i = a_2(X_i - X_S) + b_2(Y_i - Y_S) + c_2(Z_i - Z_S)$$

$$\bar{Z}_i = a_3(X_i - X_S) + b_3(Y_i - Y_S) + c_3(Z_i - Z_S),$$

$$\text{则 } a_{11} = \frac{1}{Z_i} [a_1 f + a_3(x_i - x_0)]$$

$$a_{12} = \frac{1}{Z_i} [b_1 f + b_3(x_i - x_0)]$$

$$a_{13} = \frac{1}{Z_i} [c_1 f + c_3(x_i - x_0)]$$

$$a_{21} = \frac{1}{Z_i} [a_2 f + a_3(x_i - x_0)]$$

$$a_{22} = \frac{1}{Z_i} [b_2 f + b_3(x_i - x_0)]$$

$$a_{23} = \frac{1}{Z_i} [c_2 f + c_3(x_i - x_0)]$$

$$a_{14} = (y_i - y_0) \sin \omega - \left\{ \frac{x_i - x_0}{f} [(x_i - x_0) \cos \kappa - (y_i - y_0) \sin \kappa] + f \cos \kappa \right\} \cos \omega$$

$$a_{15} = -f \sin \kappa - \frac{x_i - x_0}{f} [(x_i - x_0) \sin \kappa + (y_i - y_0) \cos \kappa]$$

$$a_{16} = y_i - y_0$$

$$a_{24} = -(x_i - x_0) \sin \omega - \left\{ \frac{y_i - y_0}{f} [(x_i - x_0) \cos \kappa - (y_i - y_0) \sin \kappa] - f \sin \kappa \right\} \cos \omega$$

$$a_{25} = -f \cos \kappa - \frac{y_i - y_0}{f} [(x_i - x_0) \sin \kappa + (y_i - y_0) \cos \kappa]$$

$$a_{26} = -(x_i - x_0).$$

$$\text{内方位元素系数矩阵: } \mathbf{C}_i = \begin{bmatrix} \frac{x_i - x_0}{f} & 1 & 0 \\ \frac{y_i - y_0}{f} & 0 & 1 \end{bmatrix}.$$

畸变系数系数矩阵:

$$\mathbf{D}_i = \begin{bmatrix} -(x_i - x_0)r^2 & -(x_i - x_0)r^4 & -r^2 - 2(x_i - x_0)^2 & -2(x_i - x_0)(y_i - y_0) \\ -(y_i - y_0)r^2 & -(y_i - y_0)r^4 & -r^2 - 2(y_i - y_0)^2 & -2(x_i - x_0)(y_i - y_0) \end{bmatrix}.$$

$$\text{常数矩阵: } \mathbf{L}_i = \begin{bmatrix} x_i - x_0 + f \cdot \frac{\bar{X}_i}{\bar{Z}_i} + \Delta x_i \\ y_i - y_0 + f \cdot \frac{\bar{Y}_i}{\bar{Z}_i} + \Delta y_i \end{bmatrix}。$$

则将方程组简记为: $\mathbf{V} = \mathbf{KX} - \mathbf{L}$,

其中,

$$\mathbf{V}_{2n \times 1} = [v_{x_1} \ v_{y_1} \ v_{x_2} \ v_{y_2} \ \cdots \ v_{x_n} \ v_{y_n}]^T$$

$$\mathbf{K}_{2n \times 13} = \begin{bmatrix} \mathbf{A}_{2 \times 6}^{(1)} & \mathbf{C}_{2 \times 3}^{(1)} & \mathbf{D}_{2 \times 4}^{(1)} \\ \vdots & \vdots & \vdots \\ \mathbf{A}_{2 \times 6}^{(n)} & \mathbf{C}_{2 \times 3}^{(n)} & \mathbf{D}_{2 \times 4}^{(n)} \end{bmatrix}$$

$$\mathbf{X}_{2n \times 1} = [\mathbf{t}_{6 \times 1} \ \mathbf{X}_{2 \times 3 \times 1} \ \mathbf{X}_{ad \ 4 \times 1}]^T$$

$$\mathbf{L}_{2n \times 1} = [\mathbf{L}_1^T \ \cdots \ \mathbf{L}_n^T]^T$$

4.1.4 精度计算

根据式 (10) 对待求参数进行最小二乘平差解算后, 保存残差并计算单位权中误差及各参数的中误差进行精度评定, 输出观测值改正数并进行分析, 思考控制点的分布与数量、控制点量测精度对计算结果的影响。

具体地, 单位权中误差计算公式如式 (12):

$$m_0 = \sqrt{\frac{\mathbf{V}^T \mathbf{V}}{2n - 13}} \quad (12)$$

对于各参数中误差, 需要计算协因数阵 \mathbf{Q}_{vv} 对角线元素与单位权中误差 m_0 的乘积, 具体计算公式如式 (13):

$$\begin{aligned} \mathbf{Q}_{vv} &= (\mathbf{K}^T \mathbf{K})^{-1} \\ m_i &= m_0 \sqrt{(\mathbf{Q}_{vv})_{ii}} \end{aligned} \quad (13)$$

4.1.5 前方交会

之前已经通过后方交会计算出了两张像片的内外方位元素和畸变系数, 故可以建立前方交会模型计算控制点和待定点的坐标。通过比较控制点坐标的计算值与真实值, 可以对之前计算得到的后方交会模型进行精度评价; 前方交会模型计算的待定点坐标还可以与直接线性变换计算的待定点坐标进行对比。

本次使用的前方交会的主要原理是: 利用后方交会计算的左右像片的内外方

位元素和畸变系数，对像点坐标进行畸变差改正后利用点投影系数法计算物方坐标初值，再利用共线方程严密方法迭代求解各点结果，并输出三维空间坐标和控制点坐标计算值与真值的差与中误差。

4.2 过程及结果

本阶段实习使用 C++ 将空间后方交会处理包装在空间后方交会类 CResection 中。如图 18 所示，在 Re_Intersection 函数中调用 cal_resection 函数完成后方交会，并通过点投影和共线方程严密解法完成空间前方交会。最终在 main 函数中调用 Re_Intersection 函数即可完成解算过程。所有计算步骤分装为不同函数，下面将详细介绍算法实现流程。

```
cout << endl << "/****** Resection *****/" << endl << endl;
cout << endl << "*****" << endl;
cout << "Left Image processing..." << endl;
CResection rs_l(iops1, eops1), rs_r(iops2, eops2);
rs_l.readfile_gcps(path_gcps);
rs_l.readfile_photos(path_left);
rs_l.cal_resection("left");
//rs_l.outter_precision_check();

cout << endl << "*****" << endl;
cout << "Right Image processing..." << endl;
rs_r.readfile_gcps(path_gcps);
rs_r.readfile_photos(path_right);
rs_r.cal_resection("right");
//rs_r.outter_precision_check();

//calculate rotation matrix
Mat R1 = rs_l.Rotation(rs_l.omega, rs_l.phi, rs_l.kappa);
R2 = rs_r.Rotation(rs_r.omega, rs_r.phi, rs_r.kappa);

//calculate photogrammetric baseline components
double Bx = rs_r.Xs - rs_l.Xs,
By = rs_r.Ys - rs_l.Ys,
Bz = rs_r.Zs - rs_l.Zs;

double M1, N2, Xtp, Ytp, Ztp;
ofstream f_output("result/Intersection.txt");
rs_l.readfile_unknown(path_unknown);
f_output << rs_l.id_unknown.size() << endl;
```

图 18 Re_Intersection 函数部分主要代码

4.2.1 坐标与参数初始化

根据 2.1.2 小节所述，为左右像片的内外方位元素赋予初值。具体地，在原始控制场坐标中， $X_{S左} = X_{S右} = 1000\text{mm}$ ， $Y_{S左} = 1500\text{mm}$ ， $Y_{S右} = 3000\text{mm}$ ， $Z_{S左} = Z_{S右} = 0\text{mm}$ 。对于两张像片的畸变系数，均赋为 0。在读取已知点物方坐标文件时，按照“-Z-X-Y”的顺序赋值，实现测墩物方空间坐标系到参与共线方程计算的物方空间坐标系的转换，如图 19 所示。在读取像方坐标时，采用以像主点为原点的像平面坐标系，单位先由像素转换为 mm，在解算完成后再将像方坐标转换为像素单位比较各参数量级。像点坐标对应点可分为三位数编号的控制点、检查点（物方坐标已知）与二位数编号的待定点（物方坐标未知），可设置不同的控制点个数与控制点选取方式，如随机或按顺序，选取方式会影响控制点在影像上的

分布。检查点则选取控制点外的三位数编号点。具体地，左像片有物方坐标已知点 81 个，右像片有物方坐标已知点 97 个，左右像片均选取前 50 个已知点作为控制点，保证了控制点在像片上均匀分布且数量合理。检查点则选取为左右像片同名像点中除了控制点外的已知点。

```
//read ground coordinates of GCP (transform)
void CResection::readfile_gcps(string filename) {
    ifstream fin(filename);
    int num_gcps, id, flag;
    PointGCP tmp;

    fin >> num_gcps;
    for (int i = 0; i < num_gcps; i++) {
        fin >> id >> tmp.Z >> tmp.X >> tmp.Y >> flag; //transform to photogrammetric coordinate system
        tmp.Z = -tmp.Z;

        gcps.push_back(tmp);
        id_gcps.push_back(id);
    }
    fin.close();
    cout << "GCPs Num:" << num_gcps << endl;
};
```

图 18 已知点空间坐标读取与转换

4.2.2 空间后方交会计算内外方位元素与畸变系数

首先逐点对应控制点像方编号查询物方坐标，由 4.1.3 小节所述原理计算该点对应的误差方程系数矩阵元素，然后迭代法化求解计算参数改正数至满足限差 $1e^{-15}mm$ ，可得到内外方位元素与畸变系数改正后的结果、像点观测值的改正数，具体代码如图 19。

```
cout << endl << "———— interior & exterior orientation elements & distortion coefficient ————" << endl << endl;

do {
    //calculate rotation matrix
    double a1, a2, a3, b1, b2, b3, c1, c2, c3;
    a1 = cos(phi) * cos(kappa) - sin(phi) * sin(omega) * sin(kappa);
    a2 = -cos(phi) * sin(kappa) - sin(phi) * sin(omega) * cos(kappa);
    a3 = -sin(phi) * cos(kappa);
    b1 = cos(omega) * sin(kappa);
    b2 = cos(omega) * cos(kappa);
    b3 = -sin(omega);
    c1 = sin(phi) * cos(kappa) + cos(phi) * sin(omega) * sin(kappa);
    c2 = -sin(phi) * sin(kappa) + cos(phi) * sin(omega) * cos(kappa);
    c3 = cos(phi) * cos(kappa);

    //establish error equations point-by-point and normalize: V=A*x-L
    for (int i = 0; i < num; i++) { ... }

    //d=(A'A)^-1-A'L
    delta = (A.t() * A).inv() * (A.t() * L); //改正数

    //correction result
    Xs += delta.at<double>(0, 0);
    Ys += delta.at<double>(1, 0);
    Zs += delta.at<double>(2, 0);
    phi += delta.at<double>(3, 0);
    omega += delta.at<double>(4, 0);
    kappa += delta.at<double>(5, 0);
    f += delta.at<double>(6, 0);
    x0 += delta.at<double>(7, 0);
    y0 += delta.at<double>(8, 0);
    k1 += delta.at<double>(9, 0);
    k2 += delta.at<double>(10, 0);
    p1 += delta.at<double>(11, 0);
    p2 += delta.at<double>(12, 0);

    flag++;
} while (flag <= iter && abs(delta.at<double>(0, 0)) > 1e-15 && abs(delta.at<double>(1, 0)) > 1e-15 && abs(delta.at<double>(2, 0)) > 1e-15
&& abs(delta.at<double>(3, 0)) > 1e-15 && abs(delta.at<double>(4, 0)) > 1e-15 && abs(delta.at<double>(5, 0)) > 1e-15
&& abs(delta.at<double>(6, 0)) > 1e-15 && abs(delta.at<double>(7, 0)) > 1e-15 && abs(delta.at<double>(8, 0)) > 1e-15
&& abs(delta.at<double>(9, 0)) > 1e-15 && abs(delta.at<double>(10, 0)) > 1e-15 && abs(delta.at<double>(11, 0)) > 1e-15 && abs(delta.at<double>(12, 0)) > 1e-15);
```

图 18 空间后方交会迭代求解

4.2.3 精度评定

根据 4.1.4 小节原理计算单位权中误差与各未知数的中误差，长度单位与输入坐标采用的单位系统相同，也可自行转换以评估精度，具体代码如图 19。

```
//accuracy assessment
Mat V = A * delta - L;
Mat tmp = V.t() * V;
Qvv = (A.t() * A).inv();
double m0 = sqrt(tmp.at<double>(0, 0) / (2 * num - 13)); //6+3+4=13

//m0*Qxx[i]
double m_Xs = m0 * sqrt(Qvv.at<double>(0, 0)), //Xs
m_Ys = m0 * sqrt(Qvv.at<double>(1, 1)), //Ys
m_Zs = m0 * sqrt(Qvv.at<double>(2, 2)), //Zs
m_phi = m0 * sqrt(Qvv.at<double>(3, 3)), //phi
m_omega = m0 * sqrt(Qvv.at<double>(4, 4)), //omega
m_kappa = m0 * sqrt(Qvv.at<double>(5, 5)), //kappa
m_f = m0 * sqrt(Qvv.at<double>(6, 6)), //f
m_x0 = m0 * sqrt(Qvv.at<double>(7, 7)), //x0
m_y0 = m0 * sqrt(Qvv.at<double>(8, 8)), //y0
m_k1 = m0 * sqrt(Qvv.at<double>(9, 9)), //k1
m_k2 = m0 * sqrt(Qvv.at<double>(10, 10)), //k2
m_p1 = m0 * sqrt(Qvv.at<double>(11, 11)), //p1
m_p2 = m0 * sqrt(Qvv.at<double>(12, 12)), //p2
```

图 18 空间后方交会精度评定

4.2.4 空间后方交会计算结果

输出像点改正数，保存在 points_left/pixel(pixel).txt 文件中，部分结果如表 1 所示。

表 1 像点改正数部分结果

left			right		
像点编号	V _x (pixel)	V _y (pixel)	像点编号	V _x (pixel)	V _y (pixel)
370	0.066528	-0.01497	333	0.097886	-0.18835
371	0.140992	-0.09588	334	0.025259	0.052073
372	-0.03418	-0.05009	336	0.153243	-0.09728
374	-0.17335	0.1671	337	0.292896	-0.16692
375	-0.01876	0.026357	340	0.026627	-0.09373
376	0.036106	0.049407	341	-0.07438	-0.02982
.....

由协因数矩阵和中误差计算得到各参数的中误差，将各参数计算值与中误差一同输出，如表 2。

4.2.5 空间前方交会计算结果

利用上述步骤得到的像片参数和待定点像点对量测坐标文件 pair_unknown.txt 对两张像片进行空间前方交会，得到待定点对的三维空间坐标，部分结果如表 3 所示。其中，对于像点编号为三位数的已知点，计算其前方交会

计算值与所给真实值的差值，并进一步求得其点位中误差，部分结果如表 4 所示。

表 2 左右影像后方交会结果

left		right	
Iteration times: 8		iteration times: 9	
m0 = 0.000867005	(mm)	m0 = 0.000917325	(mm)
Xs = 1253.09 (±) 0.315758	(mm)	Xs = 999.554 (±) 0.344633	(mm)
Ys = 1754.12 (±) 0.642917	(mm)	Ys = 3061.37 (±) 1.15308	(mm)
Zs = -6.96012 (±) 0.168839	(mm)	Zs = -14.2568 (±) 0.226366	(mm)
phi = 0.338996 (±) 0.000407352	(rad)	phi = -0.0972593 (±) 0.000404652	(rad)
omega = -0.0545265 (±) 0.000271609	(rad)	omega = -0.053882 (±) 0.000287443	(rad)
kappa = 0.0184761 (±) 2.80789e-05	(rad)	kappa = -0.0103541 (±) 3.1258e-05	(rad)
f = 25.6083 (±) 0.00493982	(mm)	f = 25.6019 (±) 0.00775709	(mm)
x0 = 0.28849 (±) 0.0102184	(mm)	x0 = 0.257856 (±) 0.0100116	(mm)
y0 = -0.103832 (±) 0.00672772	(mm)	y0 = -0.116076 (±) 0.00767138	(mm)
k1 = 0.000182088(±) 3.02478e-06		k1 = 0.000179847 (±) 2.12772e-06	
k2 = -4.0756e-07 (±) 2.31998e-08		k2 = -4.0387e-07 (±) 1.32565e-08	
p1 = -2.29154e-05 (±) 5.86082e-06		p1 = -1.76385e-05 (±) 5.69508e-06	
p2 = 4.70601e-05 (±) 4.13675e-06		p2 = 4.85356e-05 (±) 4.24042e-06	

表 3 前方交会结果（部分）

像点编号	X(mm)	Y(mm)	Z(mm)
11	4610.72	1990.42	-631.151
13	4585.35	2001.84	-1063.89
22	4588.95	2150.02	-839.982
52	4053.29	2718.32	-780.474
91	4590.34	3462.18	-610.162
430	7016.29	1404.56	-1461.21
431	7024.64	1405.69	-842.377
432	7022.89	1410.16	-238.288
433	7019.51	1418.25	364.469
451	7019.46	2669.78	-832.522
.....

表 4 前方交会精度评定（部分）

像点编号	$\Delta X(\text{mm})$	$\Delta Y(\text{mm})$	$\Delta Z(\text{mm})$
430	0.158873	5.47822	0.94932
431	-0.29856	4.0913	-0.0938
432	-0.89341	3.95197	-1.77466
433	-1.72755	4.9084	-4.93332
451	-0.19671	2.79247	-3.14734
453	-0.20205	3.89507	-4.91561
461	-0.22972	3.02833	-1.44134
462	0.072211	3.14235	-3.40104
463	0.451817	3.9288	-4.72706
464	0.803697	5.6928	-5.90753
.....

4.3 结果分析

结合像点坐标量测的实际情况与控制场控制点的分布情况，最终左右像片均选择了 50 个控制点参与后方交会解算，像点坐标改正数均在 0.5 个像素以内，精度较高。当选择不同数量的控制点参与解算时，像片外方位元素线元素尤其是 X_s 变化较大，如表 5 所示。经分析，因为本实习控制点选择规则是按照点号从小到大选取的，且控制点编号规则是按照坐标轴顺序而来，所以当选择的控制点数量减少时，其在控制场中沿 X 轴空间分布较集中，当点少于一定数值时甚至近似为一个平面，导致模型精度和可靠性下降。

表 5 不同控制点数量计算的外方位线元素中误差结果（左影像）

控制点数量	$m_{Xs}(\text{mm})$	$m_{Ys}(\text{mm})$	$m_{Zs}(\text{mm})$
50	0.315758	0.642917	0.168839
40	1.00453	0.374913	0.228073
30	2.34828	0.446045	0.399393
.....

4.4 本章小结

本阶段实习采用 VS2019+OpenCV4.5.0 实现了单像空间后方交会，采用上章的像点量测程序结果结合控制场坐标作为输入实现单张影像的外方位元素、内方位元素、畸变系数解算。最终解算结果精度较高，运算速度快，程序执行简单，能够实现高效相机检校。前方交会计算得到的三维空间坐标不仅能有效检验后方交会模型的精度，还可以为下章的直接线性变换模型计算的待定点三维空间坐标结果提供参考值。

5 直接线性变换

5.1 原理

5.1.1 DLT 基本关系式

直接线性变换（DirectLinearTransform, DLT）解法，原则上也是从共线条件方程式演绎而来。按共线条件方程式（5），当把非量测相机所摄像片安置在某坐标仪上，共线方程中的系统误差改正数假设暂时仅包含坐标轴不垂直性误差 $d\beta$ 和比例尺不一误差 ds 引起的线性误差改正数部分。假设 x 方向无比例尺误差，而 y 方向比例尺规范化系数为 $1+ds$ 。此时 x 方向像片主句若为 f_x ，则 y 方向像片主距离 f_y 满足公式如（14）。

$$f_y = \frac{f_x}{1 + ds} \quad (14)$$

这里，比例尺不一误差 ds 可以认为是所用坐标仪 x 轴和 y 轴的单位长度不一以及摄影材料的不均匀变形等因素引起的；而不正交性误差 $d\beta$ 可以认为是所用坐标仪 x 轴和 y 轴的不垂直性等因素引起的。由此，结合具体几何关系可以推得线性误差改正数 Δx 与 Δy 应满足公式（15）。

$$\begin{aligned} \Delta x &= (1 + ds)(y - y_0) \sin d\beta \\ \Delta y &= [(1 + ds) \cos d\beta - 1](y - y_0) \end{aligned} \quad (15)$$

将式（15）带入式（5）中，可以得到仅含 11 个独立参数的共线条件方程式（16），经过改化最终形成线性表达式如公式（17）也即 DLT 解法的基本关系式。

$$\begin{cases} (x - x_0) + (1 + ds)(y - y_0) \sin d\beta + f_x \frac{a_1(X - X_s) + b_1(Y - Y_s) + c_1(Z - Z_s)}{a_3(X - X_s) + b_3(Y - Y_s) + c_3(Z - Z_s)} = 0 \\ (y - y_0) + [(1 + ds) \cos d\beta - 1](y - y_0) + f_x \frac{a_2(X - X_s) + b_2(Y - Y_s) + c_2(Z - Z_s)}{a_3(X - X_s) + b_3(Y - Y_s) + c_3(Z - Z_s)} = 0 \end{cases} \quad (16)$$

$$\begin{aligned} x + \frac{l_1 X + l_2 Y + l_3 Z + l_4}{l_9 X + l_{10} Y + l_{11} Z + 1} &= 0 \\ y + \frac{l_5 X + l_6 Y + l_7 Z + l_8}{l_9 X + l_{10} Y + l_{11} Z + 1} &= 0 \end{aligned} \quad (17)$$

综合共线条件方程式及各有关关系式，我们获取各 l 系数的严格表达式，可以知道这些 l 系数是内、外方位元素以及 ds 和 $d\beta$ 的函数：

$$\begin{aligned}
\gamma_1 &= -(a_1 X_S + b_1 Y_S + c_1 Z_S) \\
\gamma_2 &= -(a_2 X_S + b_2 Y_S + c_2 Z_S) \\
\gamma_3 &= -(a_3 X_S + b_3 Y_S + c_3 Z_S) \\
l_1 &= \frac{1}{\gamma_3} (a_x - a_2 f_x \tan d\beta - a_3 x_0) \\
l_2 &= \frac{1}{\gamma_3} (b_1 f_x - b_2 f_x \tan d\beta - b_3 x_0) \\
l_3 &= \frac{1}{\gamma_3} (c_1 f_x - c_2 f_x \tan d\beta - c_3 x_0) \\
l_4 &= -(l_1 X_S + l_2 Y_S + l_3 Z_S) \\
l_5 &= \frac{1}{\gamma_3} \left[\frac{a_2 f_x}{(1 + ds) \cos d\beta} - a_3 y_0 \right] \\
l_6 &= \frac{1}{\gamma_3} \left[\frac{b_2 f_x}{(1 + ds) \cos d\beta} - b_3 y_0 \right] \\
l_7 &= \frac{1}{\gamma_3} \left[\frac{c_2 f_x}{(1 + ds) \cos d\beta} - c_3 y_0 \right] \\
l_8 &= -(l_5 X_S + l_6 Y_S + l_7 Z_S) \\
l_9 &= \frac{a_3}{r_3} \\
l_{10} &= \frac{b_3}{r_3} \\
l_{11} &= \frac{c_3}{r_3}
\end{aligned} \tag{18}$$

5.1.2 内外方位元素及 ds 和 $d\beta$ 求解

5.1.2.1 l 系数的矩阵表达式 \mathbf{L}

观察公式 (18) 不难注意到某些 l 系数表达式在外观上的相似性, 从而可以将各 l 系数排列成如 (18) 的一个矩阵 \mathbf{L} :

$$\mathbf{L} = \begin{bmatrix} l_1 & l_2 & l_3 & l_4 \\ l_5 & l_6 & l_7 & l_8 \\ l_9 & l_{10} & l_{11} & 1 \end{bmatrix} \tag{19}$$

已知 (X_S, Y_S, Z_S) 是摄站点 S 在物方空间坐标系 $D-XYZ$ 内的坐标, 而 $(\gamma_1, \gamma_2, \gamma_3)$ 是物方空间坐标系 $D-XYZ$ 原点 D 在像方空间坐标系内的坐标, 故有公式 (20):

$$\mathbf{L} = \frac{1}{\gamma_3} \begin{bmatrix} f_x & -f_x \tan d\beta & -x_0 \\ 0 & \frac{f_x}{(1+ds)\cos d\beta} & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_1 & b_1 & c_1 & \gamma_1 \\ a_2 & b_2 & c_2 & \gamma_2 \\ a_3 & b_3 & c_3 & \gamma_3 \end{bmatrix} \quad (20)$$

根据 \mathbf{L} 矩阵的结构相似性，现在就可以容易地推导出由 l 系数解算 11 个独立参数的关系式，包括求解内外方位元素的关系式。

5.1.2.2 内方位元素及 ds 和 $d\beta$ 的解算关系式

根据前述 \mathbf{L} 系数矩阵的结构，并考虑到旋转矩阵的正交性，进行具体推证，得到广义内方位元素参数的表达式如 (21)：

$$\begin{aligned} x_0 &= -\frac{l_1 l_9 + l_2 l_{10} + l_3 l_{11}}{l_9^2 + l_{10}^2 + l_{11}^2} \\ y_0 &= -\frac{l_5 l_9 + l_6 l_{10} + l_7 l_{11}}{l_9^2 + l_{10}^2 + l_{11}^2} \\ d\beta &= \arcsin \sqrt{\frac{C^2}{AB}} \\ ds &= \sqrt{\frac{A}{B}} - 1 \\ f_x &= \sqrt{A} \cos d\beta \\ f_y &= \frac{f_x}{(1+ds)} \\ A &= \gamma_3^2 (l_1^2 + l_2^2 + l_3^2) - x_0^2 \\ B &= \gamma_3^2 (l_5^2 + l_6^2 + l_7^2) - y_0^2 \\ C &= \gamma_3^2 (l_1 l_5 + l_2 l_6 + l_3 l_7) - x_0 y_0 \end{aligned} \quad (21)$$

利用以上关系式，通过已知的 11 个 l 系数，解算像主点在坐标仪坐标系内的坐标 (x_0, y_0) ，解算所摄像片的 x 方向主距 f_x 和 y 方向主距 f_y ，解算比例尺不一系数以及不正交性系数。

5.1.2.3 外方位元素的解算关系式

1) 外方位线元素的解算

根据 l 系数关系式中的关系联立可解得 3 个外方位线元素 (X_S, Y_S, Z_S) 满足公式 (22)：

$$\begin{aligned}
l_1 X_S + l_2 Y_S + l_3 Z_S &= l_4 \\
l_5 X_S + l_6 Y_S + l_7 Z_S &= -l_8 \\
l_9 X_S + l_{10} Y_S + l_{11} Z_S &= -1
\end{aligned} \tag{22}$$

2) 外方位角元素的解算

各外方位角元素满足式 (23):

$$\begin{aligned}
\tan \varphi &= -\frac{a_3}{c_3} \\
\sin \omega &= -b_3 \\
\tan \kappa &= \frac{b_1}{b_2}
\end{aligned} \tag{23}$$

5.1.3 DLT 解算过程

直接线性变换解法的具体解算过程, 包括 l 系数的解算以及空间坐标 (X, Y, Z) 的解算共两个步骤, 通过该过程建立了像点的坐标仪坐标 (x, y) 与物方空间坐标的直接线性关系式。

5.1.3.1 l 系数的近似值计算与实际解算

1) l 系数近似值解算

为解得其中 11 个 l 系数, 选择 6 个控制点 (已知它们的空间坐标), 列出 12 个方程, 求解超定方程以解得各 l 系数的近似值。

$$\begin{aligned}
\mathbf{A}\mathbf{L}^0 - \mathbf{U} = 0 &\rightarrow \mathbf{L}^0 = \mathbf{A}^T \mathbf{U} \\
\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & x_1 X_1 & x_1 Y & x_1 Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & y_1 X_1 & y_1 Y_1 & y_1 Z_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & x_2 X_2 & x_2 Y_2 & x_2 Z_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & y_2 X_2 & y_2 Y_2 & y_2 Z_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_6 & Y_6 & Z_6 & 1 & 0 & 0 & 0 & 0 & x_6 X_6 & x_6 Y_6 & x_6 Z_6 \end{bmatrix} & \begin{bmatrix} l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \\ l_6 \\ l_7 \\ l_8 \\ l_9 \\ l_{10} \\ l_{11} \end{bmatrix} - \begin{bmatrix} -x_1 \\ -y_1 \\ -x_2 \\ -y_2 \\ \vdots \\ -x_6 \end{bmatrix} &= 0
\end{aligned} \tag{24}$$

2) l 系数的解算过程

当考虑像点的非线性改正时, $\Delta x, \Delta y$ 模型见式 (15), 故可列出像点坐标观测值的误差方程如式 (25):

$$\begin{cases} v_x = -\frac{1}{A}(l_1 X + l_2 Y + l_3 Z + l_4 + x l_9 + x Y l_{10} + x Z l_{11} + A \cdot \Delta x + x) \\ v_y = -\frac{1}{A}(l_5 X + l_6 Y + l_7 Z + l_8 + y l_9 + y Y l_{10} + y Z l_{11} + A \cdot \Delta y + y) \end{cases} \quad (25)$$

该误差方程式与相应法方程式矩阵可记作如式 (26)，进行迭代计算，可由式 (21) 计算 f_x 作为迭代判据，即满足相邻两次运算差值小于等于 0.01mm 时结束迭代。

$$\begin{aligned} \mathbf{V} &= \mathbf{ML} - \mathbf{W} \\ \mathbf{L} &= (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T \mathbf{W} \end{aligned} \quad (26)$$

其中， $\mathbf{V} = [v_x \ v_y]^T$ ， $\mathbf{W} = \begin{bmatrix} \frac{x}{A} & \frac{y}{A} \end{bmatrix}^T$

$$\begin{aligned} \mathbf{L} &= [l_1 \ l_2 \ l_3 \ l_4 \ l_5 \ l_6 \ l_7 \ l_8 \ l_9 \ l_{10} \ l_{11} \ k_1 \ k_2 \ P_1 \ P_2]^T \\ \mathbf{M} &= - \begin{bmatrix} \frac{X}{A} & \frac{Y}{A} & \frac{Z}{A} & \frac{1}{A} & 0 & 0 & 0 & 0 & \frac{xX}{A} & \frac{XY}{A} & \frac{xZ}{A} & (x-x_0)r^2 & (x-x_0)r^4 & 2(x-x_0)^2 + r^2 & 2(x-x_0)(y-y_0) \\ 0 & 0 & 0 & 0 & \frac{X}{A} & \frac{Y}{A} & \frac{Z}{A} & \frac{1}{A} & \frac{yX}{A} & \frac{yY}{A} & \frac{yZ}{A} & (y-y_0)r^2 & (y-y_0)r^4 & 2(y-y_0)^2 + r^2 & 2(x-x_0)(y-y_0) \end{bmatrix} \end{aligned}$$

5.1.3.1 待定点和检查点物方坐标计算

1) 物方坐标近似值解算

对拍摄的左右像对，为解求 X 、 Y 、 Z 三个未知数，至少应列出三个方程，故由两张像片的 l 系数与各同名待定点像点坐标 (x_i, y_i) ， (x'_i, y'_i) ，根据关系式 (27) 列出 4 个方程，略去其中一个可得式 (28) 用于求解第 i 个待定点的坐标初值 (X^0_i, Y^0_i, Z^0_i) 。

$$\begin{cases} (l_1 + x l_9)X + (l_2 + x l_{10})Y + (l_3 + x l_{11})Z + (l_4 + x) = 0 \\ (l_5 + y l_9)X + (l_6 + y l_{10})Y + (l_7 + y l_{11})Z + (l_8 + y) = 0 \end{cases} \quad (27)$$

$$\begin{bmatrix} l_1 + x l_9 & l_2 + x l_{10} & l_3 + x l_{11} \\ l_5 + y l_9 & l_6 + y l_{10} & l_7 + y l_{11} \\ l'_1 + x' l'_9 & l'_2 + x' l'_{10} & l'_3 + x' l'_{11} \end{bmatrix} \begin{bmatrix} X^0 \\ Y^0 \\ Z^0 \end{bmatrix} + \begin{bmatrix} l_4 + x \\ l_8 + y \\ l'_4 + x' \end{bmatrix} = 0 \quad (28)$$

2) 物方坐标 (X, Y, Z) 的解算过程

把非线性误差改正后的坐标 $(x + \Delta x, y + \Delta y)$ 直接当作直接线性变换关系式 (17) 中的 (x, y) ，其中畸变差模型见式 (15)，可列出像点对应物方坐标关于像点坐标观测值的误差方程如式 (29)：

$$\begin{cases} v_x = -\frac{1}{A}[(l_1 + xl_9)X + (l_2 + xl_{10})Y + (l_3 + xl_{11})Z + l_4 + x] \\ v_y = -\frac{1}{A}[(l_5 + yl_9)X + (l_6 + yl_{10})Y + (l_7 + yl_{11})Z + l_8 + y] \end{cases} \quad (29)$$

该误差方程式与相应法方程式矩阵可记作如式 (30)，对于各待定点应逐点进行迭代计算，可根据物方空间坐标精度要求选择迭代判据，这里选择满足相邻两次运算各坐标差值小于 0.1mm 时结束迭代。

$$\begin{aligned} \mathbf{V} &= \mathbf{NS} - \mathbf{Q} \\ \mathbf{S} &= (\mathbf{N}^T \mathbf{N})^{-1} \mathbf{N}^T \mathbf{Q} \end{aligned} \quad (30)$$

其中， $\mathbf{V} = [v_x \ v_y \ v_x' \ v_y']^T$ ， $\mathbf{S} = [X \ Y \ Z]^T$

$$\mathbf{N} = \begin{bmatrix} -\frac{1}{A}(l_1 + xl_9) & -\frac{1}{A}(l_2 + xl_{10}) & -\frac{1}{A}(l_3 + xl_{11}) \\ -\frac{1}{A}(l_5 + yl_9) & -\frac{1}{A}(l_6 + yl_{10}) & -\frac{1}{A}(l_7 + yl_{11}) \\ -\frac{1}{A}(l'_1 + x'l'_9) & -\frac{1}{A}(l'_2 + x'l'_{10}) & -\frac{1}{A}(l'_3 + x'l'_{11}) \\ -\frac{1}{A}(l'_5 + y'l'_9) & -\frac{1}{A}(l'_6 + y'l'_{10}) & -\frac{1}{A}(l'_7 + y'l'_{11}) \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} \frac{l_4 + x}{A} & \frac{l_8 + x}{A} & \frac{l'_4 + x'}{A} & \frac{l'_8 + x'}{A} \end{bmatrix}^T$$

5.1.4 DLT 精度检验

1) 中误差和像点观测值残差

同 4.1.4，对待求参数进行最小二乘平差解算后，保存残差并计算单位权中误差及各参数的中误差，进行精度评定，输出观测值改正数并进行分析。

2) 检查点物方坐标精度评定

将 5.1.3.1 中解算的检查点坐标与已有参考数据进行对比，进行统计比较与精度评估。

5.2 过程及结果

本阶段实习采用 C++ 将直接线性变换包装在类 CDLT 中。在具体编程实现过程中，我将 DLT 算法大致拆分成类似后交和类似前交的两个部分最终实现。前一个部分通过 GetElement 函数实现，得到 l 系数、畸变系数和内外方位元素，并进行精度评定；后一部分通过 init_XYZ 和 cal_XYZ 函数实现，得到检查点和待定点的物方空间坐标，并计算活动控制架中待定点与 52 号点的距离，最后对检

查点进行精度评定。所有步骤通过在 main 函数中调用 DLT 函数即可完成。

5.2.1 l 系数与畸变系数解算

与空间后方交会一致,左右像片均选取前 50 个作为控制点,解算 DLT 系数。根据 5.1.3.1 小节中的算法,如图 19 所示,首先利用 6 个控制点计算 l 系数的初始值。

```
//initialize L1~L11 and distortion coefficients (using first 6 control points)
Mat CDLT::init_L() {

    Mat A = Mat::zeros(2 * 6, 11, CV_64F), U = Mat::zeros(2 * 6, 1, CV_64F);
    Mat LL = Mat::zeros(15, 1, CV_64F);

    //establish error equations point-by-point and normalize
    // A*L=U
    for (int i = 0; i < 6; i++) {
        A.at<double>(2 * i, 0) = gcps_ctrl[i].X;
        A.at<double>(2 * i, 1) = gcps_ctrl[i].Y;
        A.at<double>(2 * i, 2) = gcps_ctrl[i].Z;
        A.at<double>(2 * i, 3) = 1;

        A.at<double>(2 * i, 8) = gcps_ctrl[i].X * pho_ctrl[i].x;
        A.at<double>(2 * i, 9) = gcps_ctrl[i].Y * pho_ctrl[i].x;
        A.at<double>(2 * i, 10) = gcps_ctrl[i].Z * pho_ctrl[i].x;

        A.at<double>(2 * i + 1, 4) = gcps_ctrl[i].X;
        A.at<double>(2 * i + 1, 5) = gcps_ctrl[i].Y;
        A.at<double>(2 * i + 1, 6) = gcps_ctrl[i].Z;
        A.at<double>(2 * i + 1, 7) = 1;

        A.at<double>(2 * i + 1, 8) = gcps_ctrl[i].X * pho_ctrl[i].y;
        A.at<double>(2 * i + 1, 9) = gcps_ctrl[i].Y * pho_ctrl[i].y;
        A.at<double>(2 * i + 1, 10) = gcps_ctrl[i].Z * pho_ctrl[i].y;

        U.at<double>(2 * i, 0) = -pho_ctrl[i].x;
        U.at<double>(2 * i + 1, 0) = -pho_ctrl[i].y;
    }
    Mat l = (A.t() * A).inv() * A.t() * U;
    for (int j = 0; j < 11; j++) {
        LL.at<double>(j, 0) = l.at<double>(j, 0);
    }
    for (int j = 0; j < 4; j++) {
        LL.at<double>(j + 11, 0) = 0;
    }
    return LL;
}
```

图 19 l 系数初始值解算

将 l 系数的初始值与畸变系数共同构成 L 矩阵后进行逐点建立误差方程并迭代求解,每次迭代均计算判据 f_x 是否满足两次差值小于等于 0.01mm,输出迭代次数与 L 矩阵结果如表 6,并输出像点坐标观测值残差如表 7,单位权中误差与各未知数的中误差如表 8。

表 6 / 系数与畸变系数解算结果

left	right
Control points num: 50	Control points num: 50
Iteration times: 4	Iteration times: 5
L1~L11:	L1~L11:
0.0137581	0.0364306
0.000260566	-0.000395982
0.00466236	-0.00389811
-18.2893	-115.403
-8.94103e-06	0.000205796
0.0145072	0.0366008
-0.000779467	-0.00184079
-0.857134	-1.9682
-0.000188328	0.000139395
3.0873e-05	7.6517e-05
0.000534251	0.00142319
Distortion Coefficients:	Distortion Coefficients:
0.000181928	0.000179803
-4.05181e-07	-4.03614e-07
-2.22406e-05	-9.58958e-06
4.70147e-05	4.29279e-05

表 7 像点坐标观测值残差部分结果

left			right		
像点编号	V _x (pixel)	V _y (pixel)	像点编号	V _x (pixel)	V _y (pixel)
133	-0.381528	-0.02767	122	0.101474	-0.06797
134	-0.307262	0.154528	123	0.170291	0.00945
135	-0.111558	-0.02393	124	0.176215	-0.12132
141	0.14699	0.341348	125	-0.0316887	-0.01927
142	0.0401087	-0.06973	126	-0.0381011	0.024044
143	-0.00841886	0.068464	133	-0.332346	-0.00801
.....

5.2.2 内外方位元素解算

按 5.1.2 小节中公式求解内外方位元素，畸变系数直接取矩阵的四个参数。左右片结果见表 9。与空间后方交会的结果进行对比，数据基本拟合且差值在各数值的 10%以内。

表 8 左右影像像点坐标观测值单位权中误差与各未知数中误差

left		right	
Iteration times: 4		iteration times: 5	
m0 = 0.000808489	(mm)	m0 = 0.000746414	(mm)
L1~L11:		L1~L11:	
	7.37293e-06		7.37293e-06
	5.00188e-07		5.00188e-07
	2.73077e-06		2.73077e-06
	0.00958932		0.00958932
	2.75148e-07		2.75148e-07
	7.78955e-06		7.78955e-06
	7.02804e-07		7.02804e-07
	0.00321321		0.00321321
	1.81572e-07		1.81572e-07
	1.43957e-07		1.43957e-07
	2.56309e-07		2.56309e-07
Distortion Coefficients:		Distortion Coefficients:	
	2.89992e-06		2.89992e-06
	2.18929e-08		2.18929e-08
	5.5263e-06		5.5263e-06
	3.86873e-06		3.86873e-06

表 9 左右影像 DLT 参数计算结果

left		right	
Iteration times: 4		iteration times: 5	
m0 = 0.000808489	(mm)	m0 = 0.000746414	(mm)
Xs = 1253.03	(mm)	Xs = 1001.66	(mm)
Ys = 1754.11	(mm)	Ys = 3060.41	(mm)
Zs = -7.16043	(mm)	Zs = -13.8101	(mm)
phi = 0.338907	(rad)	phi = -0.0976339	(rad)
omega = -0.0544466	(rad)	omega = -0.0534573	(rad)
kappa = 0.0184187	(rad)	kappa = -0.0103757	(rad)
x0 = 0.286216	(mm)	x0 = 0.243727	(mm)
y0 = -0.102946	(mm)	y0 = -0.102145	(mm)
f _x = 25.6086	(mm)	f _x = 25.585	(mm)
f _y = 25.6084	(mm)	f _y = 25.5909	(mm)
dβ = -0.000153415	(mm)	dβ = -0.000171793	(mm)
ds = 7.02244e-06	(mm)	ds = -0.000229131	(mm)
k1 = 0.000181928		k1 = 0.000181928	
k2 = -4.05181e-07		k2 = -4.05181e-07	
p1 = -2.22406e-05		p1 = -2.22406e-05	
p2 = 4.70147e-05		p2 = 4.70147e-05	

5.2.3 待定点物方坐标解算

与后方交会-前方交会使用的待定点相同，包括左右影像上有对应同名像点的 9 个未知点以及挑选已知同名点 18 个。利用 5.1.3 小节中叙述原理首先计算各待定点物方坐标初值，再迭代计算其精确值，得到的部分结果如表 10 所示。

表 10 待定点物方坐标解算结果（部分）

像点编号	X(mm)	Y(mm)	Z(mm)
11	4609.48	1990.37	-627.553
12	4595.46	1994.93	-857.154
22	4587.8	2149.91	-836.321
473	7018.43	3874.66	369.49
483	7016.75	4455.81	370.137
484	7015.88	4473.54	967.353
.....

5.2.4 物方坐标精度评定

对于已知点（检查点），计算与所给物方坐标的坐标残差，结果如表 11，中误差为 2.37635mm。

表 11 检查点物方坐标残差计算结果

像点编号	V _x (mm)	V _y (mm)	V _z (mm)
430	1.04962	-0.03437	1.84791
431	1.62652	-0.44872	0.599417
432	3.02453	-1.00203	0.565282
433	6.06675	-1.7952	1.57445
451	2.13384	-0.4195	-0.32891
453	3.26914	-0.48698	0.516999
461	0.118878	-0.61843	-0.01671
462	1.68283	-0.39864	-0.01471
463	2.65401	-0.09987	0.582991
464	3.51777	0.178989	2.03959
470	-1.32319	-0.95784	1.56166
471	0.225309	-0.53609	0.123198
472	1.45194	-0.02356	0.036756
473	1.69837	0.259439	0.60341
481	0.00235621	-0.48485	0.584185
482	1.62518	0.389332	0.226141
483	2.05206	0.819516	0.884781
484	2.43585	1.47461	1.95726
Mean Error:	2.37635 (mm)		

对于未知待定点，计算与前方交会结果的坐标残差，结果如表 12：

表 12 待定点物方坐标残差计算结果

像点编号	V _x (mm)	V _y (mm)	V _z (mm)
11	-1.24	-0.05	3.598
12	-1.39	-0.07	3.718
13	-1.54	-0.09	3.84
21	-1.02	-0.08	3.547
22	-1.15	-0.11	3.661
23	-1.32	-0.14	3.81
52	-0.8	-0.31	3.695
91	-0.78	-0.45	3.501
92	-0.9	-0.52	3.601

5.2.5 活动控制架距离计算

由 DLT 解算物方空间坐标后即可根据每个标志点与 52 号点的坐标差值直接计算距离，最终距离计算结果如下表 13。

表 13 活动控制场距离计算结果

像点编号	距离(mm)
11	928.419
12	907.804
13	935.72
21	813.878
22	782.839
23	819.046
52	0
91	933.011
92	913.797

5.3 结果分析

在直接线性变换计算待定点时，检查点的分布与其计算残差也体现出一定的相关性。结合拍摄的影像和表 11 可以发现，分布在影像四周的的点误差明显较大，由其集中性猜测是控制点分布对该区域控制效果不强导致。检查控制点的选择发现可能是由于沿摄影方向较近且摄影并不是严格的正直摄影造成的。而关于未知待定点坐标的对比，前方交会的计算结果只能作为参考，因为前方交会也是利用后方交会计算的内外方位元素进一步进行计算的，而后方交会也有误差。从表 12 的残差结果来看，竖直方向坐标残差统一均约为 3mm，摄影方向残差为 0.78mm~1.54mm，水平 Y 轴方向残差稳定小于 0.6mm，两种方法计算得到的结

果基本一致，位置相对准确。

5.4 本章小结

本阶段实习采用 VS2019+OpenCV4.5.0 实现了直接线性变换（DLT），采用第三章的像点量测程序结果结合控制场坐标、检查点坐标作为输入实现了影像的内外方位元素、畸变系数、线性变换系数解算及物方空间坐标计算，基于计算所得物方空间坐标进行了检查点精度评定及活动控制场距离计算。最终计算结果与空间后方交会接近，证明算法复现正确，且精度合理、代码逻辑层次清晰，能够实现高效近景摄影测量处理。

结论

本次实习阶段 1 采用单反相机完成了控制场影像获取，使我对控制场布设，使用摄影设备获取摄影测量数据有了更加深刻的理解。最终摄取的立体像对曝光合适，景深较大，包含足量的控制点且像点清晰度高，影像间重叠度较大符合后续处理要求。实习 1 帮助我为后续摄影测量工作打下了良好基础。

实习阶段 2 自行编制了像点坐标量测程序，连通域灰度重心算法、Hough 圆检测算法等相关计算机视觉技术有了更加深刻的理解，并将其应用于控制场影像实现自动像点量测。最终对比多种算法选取出效果最好的两种进行最终的像点量测工作，量测结果精度较高符合人眼检测结果，可用于后续近景摄影测量处理。让我对于摄影测量成像原理、成像特点及最小二乘平差原理有了新的理解。

实习阶段 3 自行编制单像空间后方交会及 DLT 算法，让我在课堂学习之后能够通过实践验证并深入理解相关理论。通过本次实习我也明白了书上得来终觉浅，例如 DLT 算法书上有数十个公式在理论学习过程中我只觉繁杂不知如何将它们联系起来。但在实际编程实现时，按照逻辑按部就班就能够联系起所有公式，串联其基本原理最终复现算法。我还认识到，控制点的数量与分布对结算结果的影响之大，像点坐标量测结果精度几乎决定了空间后方交会模型的精度。

本次实习过程历时一周，我对课上所学的知识有了基于实践的认知，编程能力与解决问题的能力都得到了有效训练，也让我从实践中体会并理解了近景摄影测量原理，收获颇多。非常感谢老师的帮助，祝老师工作顺利！

参考文献

- [1] 潘励, 段延松, 刘亚文, 等. 摄影测量学[M], 武汉大学出版社, 2023.
- [2] 张恒, 胡文龙, 丁赤飙. 基于快速连通域分析的目标特征提取算法 [J]. 计算机工程与应用, 2009, 45 (29): 230 - 232.
- [3] 张会章, 张利霞, 郭雷. 用霍夫变换来提取目标边界 [J]. 计算机应用, 2003 (z1): 117 - 119.
- [4] ZENG Z, WANG X. A general solution of a closed-form space resection[J]. Photogrammetric Engineering and Remote Sensing, 1992, 58: 327 - 327.
- [5] ABDEL-AZIZ Y I, KARARA H M, HAUCK M. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry[J]. Photogrammetric engineering & remote sensing, 2015, 81 (2): 103 - 107.

附录 A 像点坐标量测结果

左影像(98 个)

单位: 像素(pixel)

像点编号	X	Y
11	847.645	2079.59
12	857.425	2420.5
13	870.738	2719.35
21	1073.85	2068.26
22	1091.18	2379.91
23	1107.16	2721.22
91	2839.47	1976.08
92	2847.77	2252.43
132	759.739	2356.06
133	758.334	1852.43
134	762.708	1307.57
135	761.86	889.016
136	765.357	456
141	1949.52	2736.59
142	1955.111	2297.18
143	1962.57	1817.86
144	1968.82	1316.25
145	1976.92	928.095
146	1983.09	527.173
147	1988.87	140.416
153	2893.31	1789.35
154	2904.7	1322.79
155	2911.99	959.157
156	2921.05	590.847
157	2929.07	225.133
161	3791.33	2562.67
162	3809.15	2169.38
163	3827.85	1719.86
164	3843.87	1285.6
165	3857.1	851.865
166	3870.93	386.618
221	1458.04	2184.04
222	1462.42	1696.42
223	1468.93	1205.21

224	1475.54	697.805
225	1486.1	192.079
330	589.926	2460.82
331	589.062	2092.2
332	586.256	1693.13
333	585.891	1291.46
334	589.252	960.189
336	594.154	290.715
345	1504.41	650.375
352	2452.63	1651.64
353	2459.54	1283.95
354	2465.79	984.581
355	2471.16	696.577
356	2476.63	398.702
360	3143.3	2323.32
361	3155.01	1984.1
362	3167.68	1629.86
363	3177.29	1267.81
364	3189.07	908.726
365	3195.86	533.247
370	3747.94	2245.17
371	3758.84	1963.69
372	3772.18	1628.61
374	3796.46	935.901
375	3805.11	586.548
376	3815	232.927
430	155.857	2447.64
431	151.601	1895.13
432	152.506	1349.48
433	159.7	798.421
434	160.216	244.641
450	1276.75	2383
451	1289.3	1858.73
452	1289.28	1348.64
453	1306.38	836.095
454	1317.77	321.925
460	1767.85	2353.89
461	1784.02	1846.92

462	1785.64	1350.79
463	1800.98	854.357
464	1808.27	353.722
470	2275.38	2322.49
471	2264.75	1833.61
472	2274.71	1355.56
473	2268.68	873.149
474	2272.65	388.983
481	2684.69	1821.36
482	2690.38	1357.43
483	2694.16	890.117
484	2716.72	423.998
490	3093.12	2266.21
491	3105.85	1809.07
492	3109.24	1360.56
493	3117.02	910.362
494	3107.06	454.387
501	3470.78	1797.46
502	3488.38	1365.27
503	3499.58	927.178
504	3511.27	489.842
505	3525.97	51.42
511	3733.09	1789.25
512	3744.19	1366.24
514	3767.8	512.457
515	3778.27	82.8668

右影像(115 个)

单位: 像素 (pixel)

像点编号	X	Y
11	1250.71	1999.72
12	1259.09	2301.26
13	1270.73	2566.45
21	1441.63	1998.75
22	1454.65	2280.02
23	1468.63	2588.71
31	1534.1	2035.85
72	2833.68	2344.21
81	3000.89	1981.95

82	3013.3	2289.08
91	3219.27	1987.45
92	3227.98	2277.77
122	162.799	2159.89
123	145.712	1699.76
124	131.703	1234.12
125	117.841	772.189
126	106.031	287.354
133	1288.88	1801.26
134	1281.81	1320.03
135	1272.48	950.942
136	1265.6	569.671
137	1261.31	192.352
141	2403.76	2697.33
142	2400.61	2267.96
143	2397.44	1800.15
144	2391.18	1310.91
145	2389.45	932.479
146	2384.82	541.707
147	2379.62	165.03
151	3412.62	2701
152	3417.39	2272.75
153	3421.94	1797.18
154	3421.36	1302.19
155	3419.64	916.694
156	3419.13	525.813
157	3417.38	137.338
210	602.332	2501.83
211	586.868	2082.99
212	573.195	1656.97
213	562.195	1250.53
214	553.187	824.92
215	542.843	386.282
220	2069.38	2567.16
221	2065.2	2138.08
222	2059.82	1677.43
223	2055.23	1214
224	2049.19	735.396

225	2046.21	258.619
320	682.425	2363.99
321	670.607	2008.32
322	659.81	1640.96
323	650.874	1285.31
324	641.929	884.429
325	634.848	525.549
326	627.599	150.736
332	1482.93	1667.04
333	1475.58	1308.75
334	1472.3	1013.38
336	1463.13	417.962
337	1460.08	86.124
340	2307.24	2362.07
341	2304.75	2023.45
342	2301.25	1652.82
343	2296.49	1272.51
344	2294.87	967.679
345	2291.01	681.789
346	2287.39	389.396
350	3299.48	2359.63
351	3302.04	2029.26
352	3304.3	1652.76
353	3304.12	1270.95
354	3303.38	960.095
355	3301.86	661.06
356	3299.62	351.446
360	4152.49	2408.57
361	4158.6	2032.33
362	4165.4	1639.58
363	4168.65	1238.12
364	4173.91	839.321
365	4173.18	421.498
402	40.3092	1374.68
403	22.5626	912.561
411	481.291	1833.27
412	468.814	1370.33
413	448.475	906.434

414	438.853	433.087
421	910.921	1834.62
422	897.117	1366.57
423	879.359	894.725
424	868.376	417.759
430	1356.72	2328.19
431	1347.73	1844.28
432	1340.98	1365.79
433	1338.37	883.219
434	1330.57	399.849
440	1857.36	2332.81
441	1883.95	1839.67
442	1844.23	1358.53
443	1857.6	872.536
444	1843.84	384.472
451	2347.81	1841.3
453	2346.33	863.193
461	2839.51	1843.24
462	2832.9	1350.05
463	2838.08	856.335
464	2832.82	358.242
470	3370.2	2351.16
471	3353.56	1843.64
472	3355.98	1349.13
473	3338.5	850.655
480	3846.17	2353.47
481	3836.97	1843.53
482	3834.3	1345.84
483	3828.12	844.757
484	3843.43	342.268
