

摄影测量学

——点特征提取及相关系数法匹配

学 院：遥感信息工程学院

班 级：2006班（20F10）

姓 名：马文卓

学 号：2020302131249

**目录**

[一、ReadMe 3](#_Toc21530)

[1. 实习任务 3](#_Toc5902)

[2. 实验数据 3](#_Toc4629)

[3. 编程环境 3](#_Toc13895)

[4. 文件组织结构 3](#_Toc10001)

[二、原理及算法 4](#_Toc9476)

[1. 点特征提取 4](#_Toc16613)

[1.1Moravec算法 4](#_Toc22533)

[1.2Forstner算法 5](#_Toc3728)

[2. 相关系数法匹配 6](#_Toc667)

[三、编程思路及流程图 7](#_Toc32038)

[1. 点特征提取 7](#_Toc28783)

[1.1Moravec算法 7](#_Toc30985)

[1.2Forstner算法 10](#_Toc22093)

[1.3点特征提取主程序 14](#_Toc15869)

[2. 相关系数法匹配 15](#_Toc6761)

[四、试验结果及分析 21](#_Toc21726)

[1. 点特征提取 21](#_Toc22465)

[1.1结果分析 21](#_Toc5119)

[1.2阈值分析 25](#_Toc15745)

[1.3随机/均匀提取实施 26](#_Toc26656)

[2. 相关系数法匹配 27](#_Toc2148)

[2.1特征点随机分布 27](#_Toc27325)

[2.2特征点均匀分布 28](#_Toc1972)

# 一、ReadMe

## 实习任务

本次实习的任务主要包含两大模块：点特征提取、相关系数法匹配。

·点特征提取：利用Moravec或Forstner算法提取影像的特征点，并在影像上显示出来。同时研究兴趣阈值、抑制窗口对结果的影响，以及随机提取和均匀提取的实施分析。

·相关系数法匹配：利用相关系数匹配算法，对两张影像进行匹配，得到同名点，并进行可视化。同时研究匹配窗口、相关系数阈值等对结果的影响。

## 实验数据

本次实习当中所用到的数据均存放于data文件夹中。包含三张图片：

·panLeft.bmp：灰度影像，942x1023

·panRight.bmp：灰度影像，887x805

·chess.bmp：彩色影像，467x638，主要用于检测点特征提取算法的效果

## 编程环境

本次实习当中编程环境如下：

·编程语言：Python

·环境：Python 3.9.7

·第三方库：opencv、numpy、math（其余Moravec、Forstner、匹配算法均为自己编写）

## 文件组织结构

·点特征提取及相关系数匹配：项目总文件夹

·data：实验数据文件夹

·PointFeatureExtraction：点特征提取文件夹

·Moravec(py)：Moravec算法

·Forstner(py)：Forstner算法

·main(py)：点特征提取主程序

·CorrelationCoefficientMatching：相关系数匹配文件夹

·Moravec(py)：匹配算法中所用到的Moravec特征提取算法

·main(py)：相关系数法匹配的主程序

·Result：实验结果文件夹

·PointFeatureExtraction：点特征提取结果文件夹

·Moravec：Moravec算法结果文件夹

·chess：棋盘数据结果文件夹

·tenniscourt：网球场数据结果文件夹

·Forstner：Forstner算法结果文件夹

·chess：棋盘数据结果文件夹

·tenniscourt：网球场数据结果文件夹

·CorrelationCoefficientMatching：相关系数匹配结果文件夹

·实验报告(docx)：实验报告

·ReadMe(txt)：项目概述

·illustration：流程图文件夹

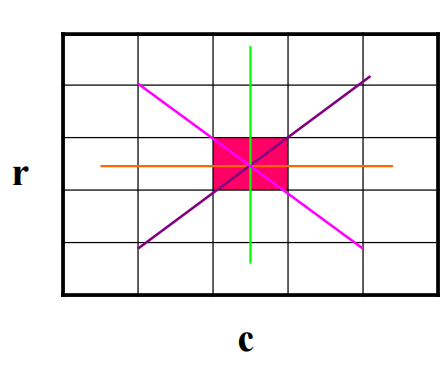
# 二、原理及算法

## 点特征提取

点特征提取是指运用某种算法从影像中提取我们感兴趣的或者有利用价值的点的过程。其中特征点主要指明显点、具有某种特征的点（如灰度特征），提取特征点所用的算子则称为兴趣算子或有利算子。主要的算子有Moravec、Forstner、Harris算子等，具体介绍见下文。

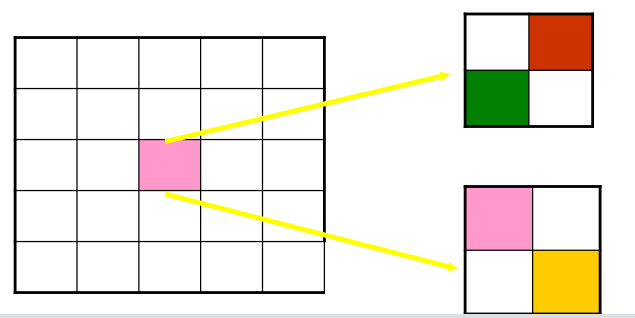
### 1.1Moravec算法

1. **概述**：一种基于灰度方差的角点检测算子，该算子计算图像中每个像素点沿着水平、垂直、对角线及反对角线的四个方向的灰度方差，其中的最小值选作该像素点的兴趣值 IV，再通过局部非极大值抑制来检测其是否为特征点（角点）
2. **算法步骤**：
3. 计算各像元的兴趣值IV：以像素（c,r）为中心的wxw影像窗口中，计算四个方向（垂直、水平、正对角线、反对角线）相邻像素灰度差的平方和，去四个方向的最小值为（c，r）像点的IV



1. 得到候选点：给定经验阈值，将兴趣值大于阈值的点作为候选点（阈值的选择应该以候选点中包括所需要的特征点而又不含过多非特征点为原则）
2. 非极大值抑制：在一定的抑制窗口内，将候选点中兴趣值最大的点保留作为特征点，其他点去掉
3. **算法特点：**
4. 简单快速
5. 对强边缘比较敏感，这是由于响应值是自相关的最小值而不是差值
6. 不是旋转不变的，当图像发生旋转时，检测出的特征点具有较低的重复率
7. 对噪声也比较敏感，可以考虑用比较平滑的滤波来消除噪声

### 1.2Forstner算法

1. **概述**：通过计算各像素的Robert’s梯度和像素（c，r）为中心的窗口内的灰度协方差矩阵，在影像中寻找具有尽可能小而接近圆的误差椭圆的点作为特征点
2. **算法步骤：**
3. 计算个像素的Robert’s梯度



1. 计算IxI窗口中的灰度协方差矩阵



1. 计算兴趣值q、w



1. 确定候选点：当q>Tq同时w>Tw时，该像元为候选点



1. 非极大值抑制：在一个窗口内，选取权值w最大的候选点作为特征点
2. **算法特点：**
3. 精度高
4. 计算量大

## 相关系数法匹配

1. **概述：**基于相关系数来判断搜寻两张影像上的同名点
2. **算法步骤：**
3. 读取左右影像
4. 确定目标点位置：使用Moravec算子或Forstner算子对左影像进行特征点提取，将提取的特征点作为目标点（具体提取步骤见上）
5. 预测右影像的搜索范围：使用ps等工具得到左右影像的大致视差，然后根据搜索窗口的大小确定搜索范围（搜索窗口不能太小，否则可能找不到匹配同名点）
6. 逐窗口计算相关系数并保存：计算窗口内左影像个右影像灰度的相关系数，并且保存起来。公式如下：



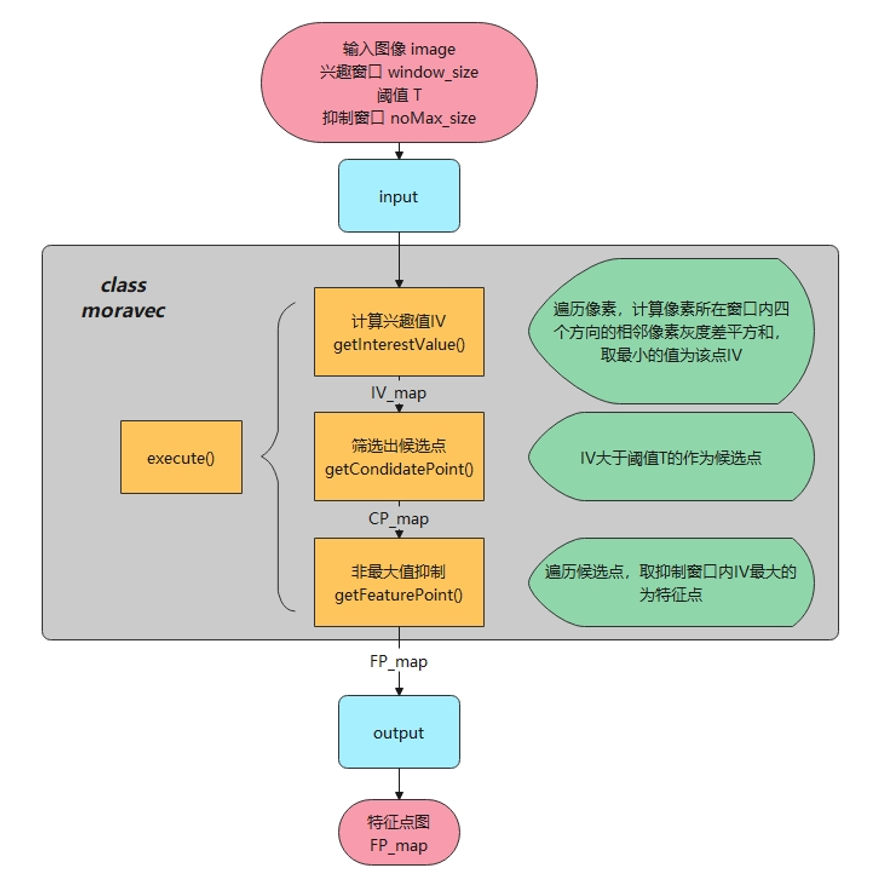
5>比较一个搜索窗口内的相关系数，如果最大值大于相关系数阈值，则该最大值对应的像点则为对应目标点在右影像上的同名点

# 三、编程思路及流程图

## 点特征提取

### 1.1Moravec算法

1. **流程图**



1. **编程思路**

总体编程架构为：编写moravec类，其包含了getInterestValue、getCondidatePoint、getFeaturePoint、execute等方法来实现特征点的提取。

1. init

初始化函数。

主要初始化参数为：输入图像（彩色）、兴趣窗口大小、兴趣阈值、抑制窗口大小。详细信息见下图：



1. getInterestValue

计算兴趣值函数。

遍历像素（边界像素可忽略），计算垂直、水平、正反对角线对应相邻像素灰度差平方和，取最小值为该点IV，并记录在IV\_map中，最后输出IV\_map图像。

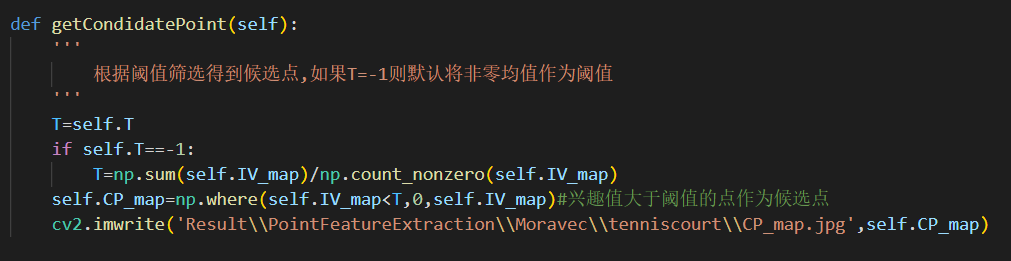


1. getCondidatePoint

筛选出候选点。

根据之前提供的阈值，将IV大于阈值T的点作为特征点候选点，记录在CP\_map中，并输出CP\_map图像。

如果T为默认值-1，则阈值为图像IV非零均值。



1. getFeaturePoint

非极大值抑制，得到特征点。

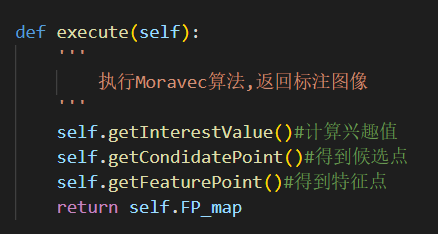
以抑制窗口遍历图像，每个抑制窗口内选取IV最大的候选点作为特征点，记录在FP\_map中，并输出FP\_map图像。值得注意的是，在此步骤中一定要防止抑制窗口越界。



1. Execute

运行函数，执行Moravec算法。

使用上述方法，依次进行兴趣值计算、阈值筛选、非极大值抑制，返回特征点图FP\_map。



### 1.2Forstner算法

1. **流程图**



1. **编程思路**

总体编程架构为：编写forstner类，其包含了getPrimaryPoint、getGraycovarianceMatrix、getQandW、getCondidatePoint、getFeaturePoint、execute等方法来实现特征点的提取。

1. init

初始化。

主要初始化参数为：输入图像（彩色）、计算窗口大小、抑制窗口大小、初选差分阈值、q值阈值、w阈值系数等。

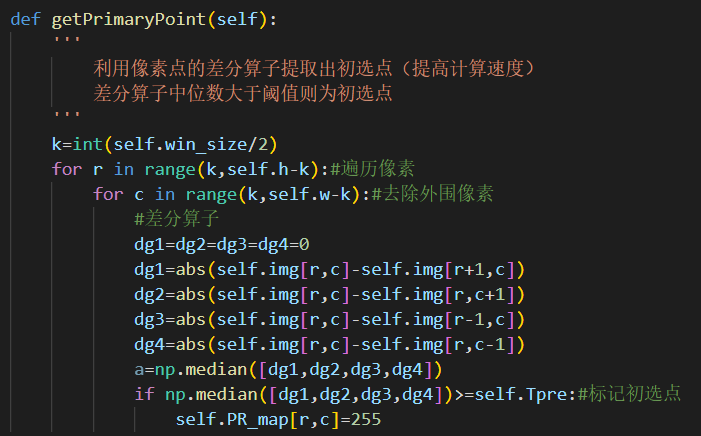
具体见下图：



1. getPrimaryPoint

筛选初选点函数。

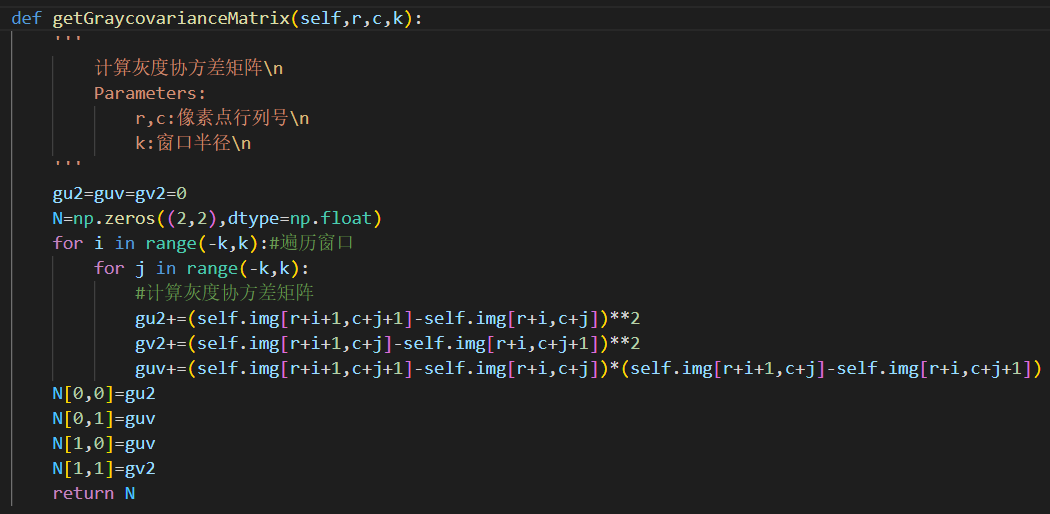
为了减少不必要的计算，加快算法速度，通过计算每个像素的差分算子，如果差分算子的中位数大于初选阈值，则该点标记为初选点，记录在PR\_map中。（这样一些没有价值的点就被排除，不需要经历后续负责的运算，节约时间成本）。



1. getGraycovarianceMatrix

计算灰度协方差矩阵。

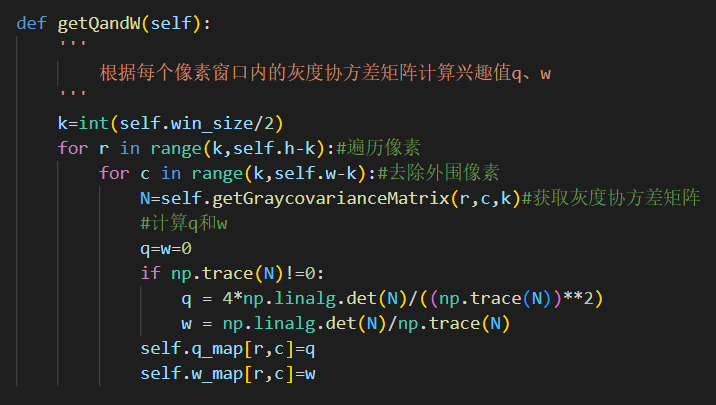
根据输入的范围以及上述公式，计算出该点的灰度协方差矩阵N。



1. getQandW

计算兴趣值q、w。

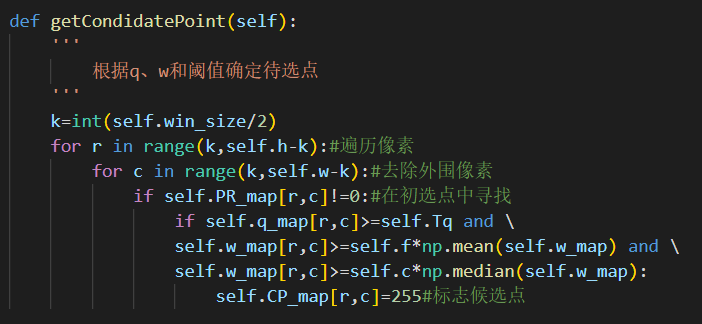
根据灰度协方差矩阵和上述公式计算该点兴趣值q、w，并存储在q\_map和w\_map中。



1. getCondidatePoint

筛选出候选点。

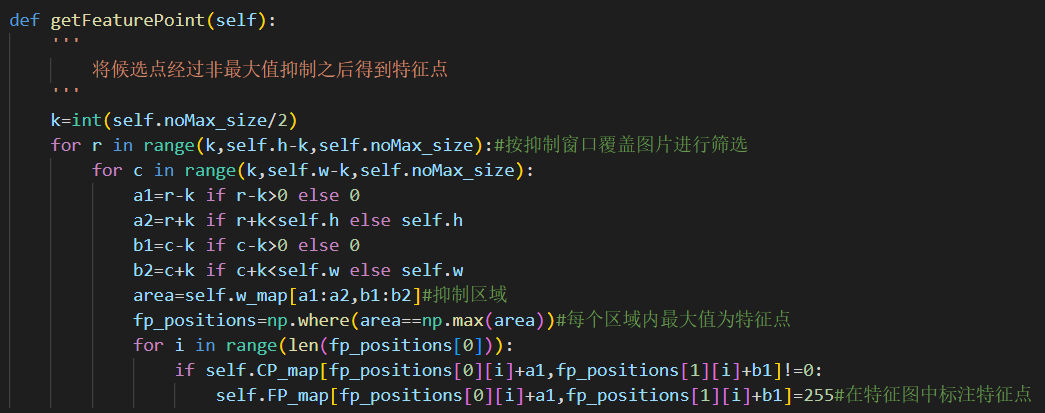
根据w阈值系数确定w阈值，将初选点中同时满足q阈值和w阈值的点作为特征点。



1. getFeaturePoint

非极大值抑制，得到特征点。

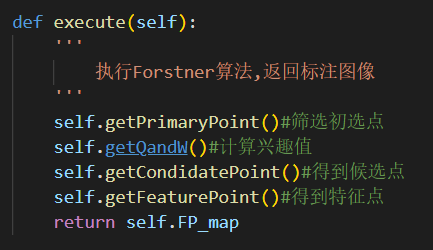
取每个抑制窗口中w值最大的候选点作为特征点。



1. Execute

运行函数，执行forstner算法。

使用上述方法，依次进行初选、兴趣值计算、阈值筛选、非极大值抑制，返回特征点图FP\_map。



### 1.3点特征提取主程序

本程序主要是进行点特征提取算法主框架编写，进行图片读取、算法选择、特征点标注、结果保存等工作。

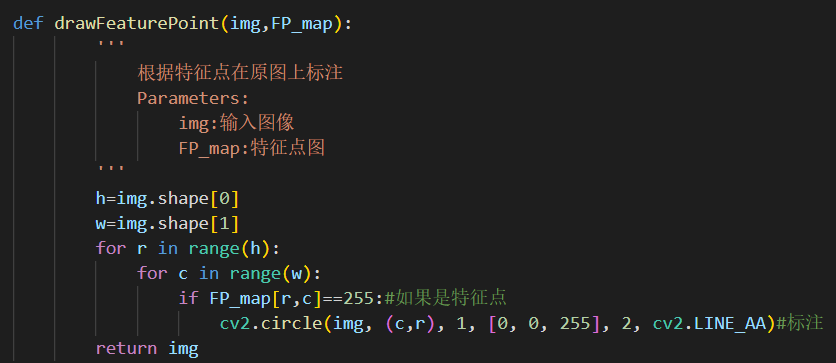
1. 图片读取

使用opencv库的imread函数进行图片读取并展示。



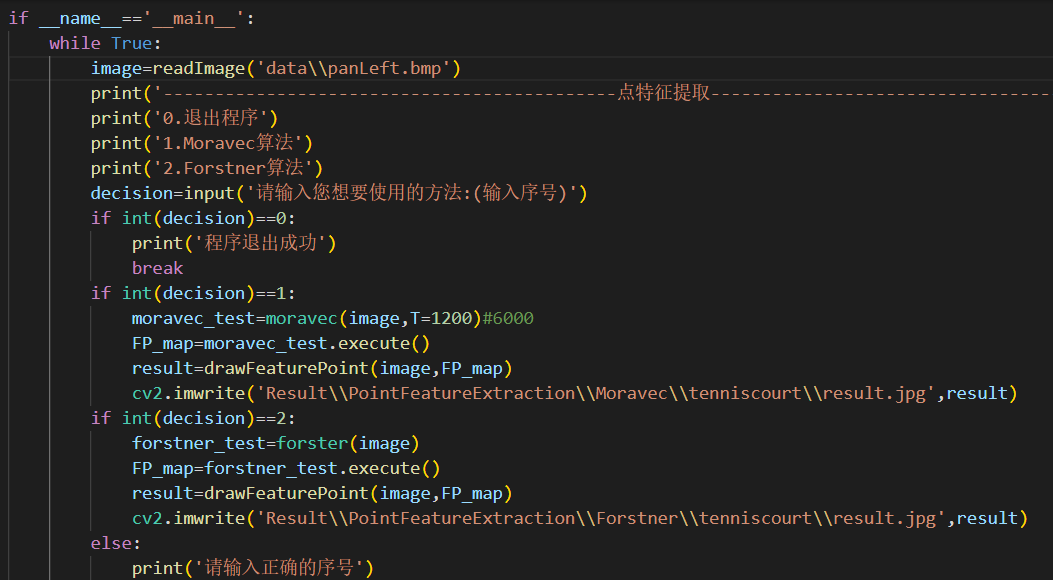
1. 特征点标注

使用opencv库的circle函数进行特征点的标注，值得注意的是这个函数中的坐标参数采用的是（列号，行号）。



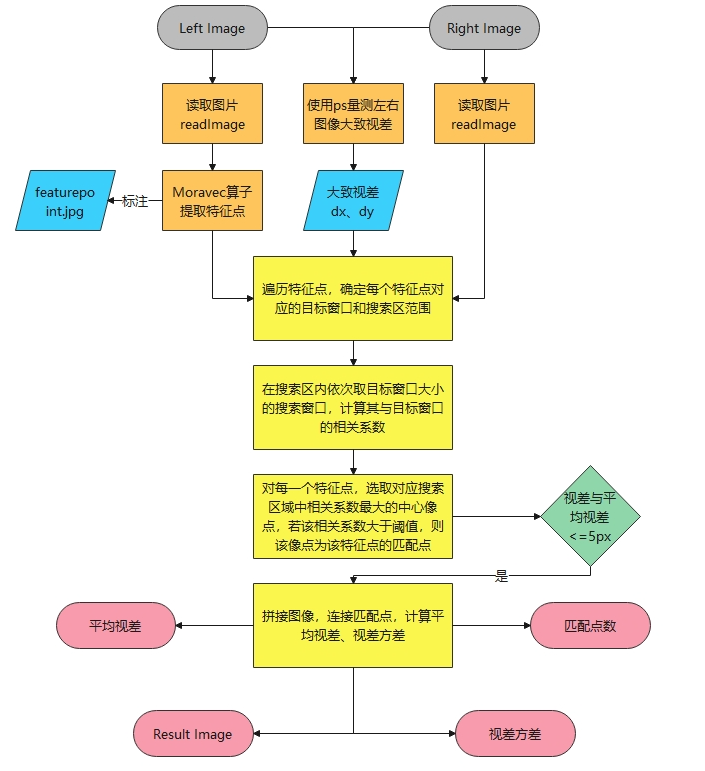
1. 算法选择和结果保存

这里编写了一个简单的循环选择界面，可给用户提供简洁的使用需求。



## 相关系数法匹配

1. **流程图**



1. **编程思路**

相关系数匹配的实现主要是两个模块完成，一个是引用了之前点特征提取中的Moravec类完成算法中的特征提取部分，其与部分（图片读取、匹配、结果保存等）均在main.py中。

1. 量测大致视差

为了防止遗太多误匹配（如果不量测大致视差，正确的匹配点有时可能会在搜索区域外，从而导致很多误匹配），我使用ps对左右图像的进行大致视差的量测。具体步骤如下：

①将两张图片在ps中打开，在左图中取一明显点，记录坐标(x1,y1)

②在右图中取相同明显点，记录坐标(x2,y2)

③计算视差：水平视差=x2-x1，上下视差=y2-y1

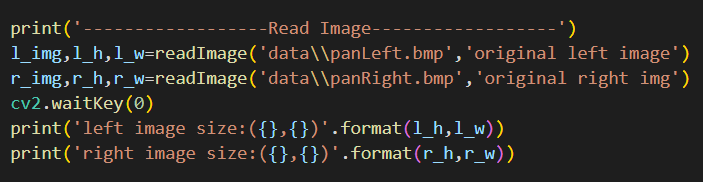
④重复①-③步骤4-5次，取视差平均值为大致视差

经过量测，大致视差为：水平视差=176，垂直视差=8

1. 读取图片

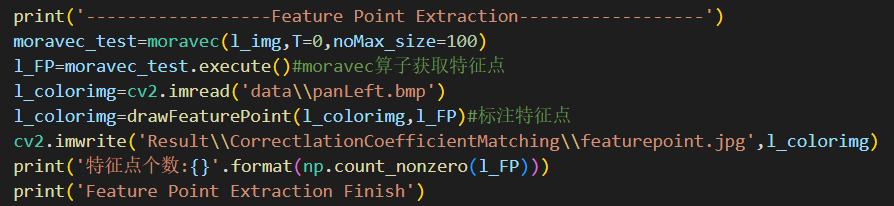
和点特征提取中的readImage一致。

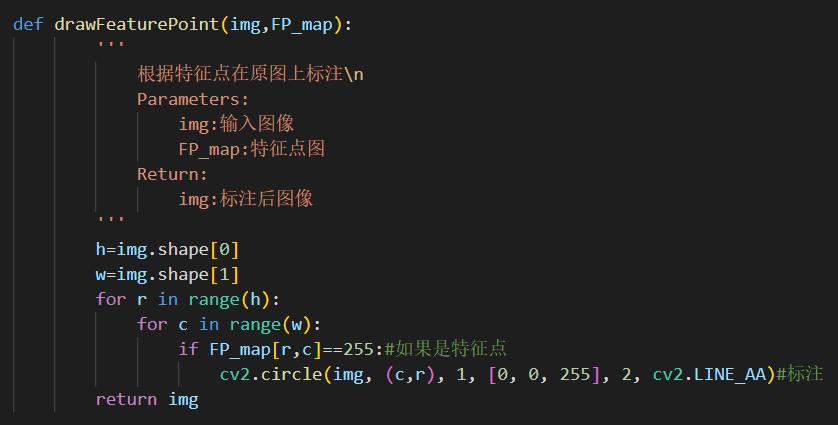




1. 提取目标点并绘制目标点

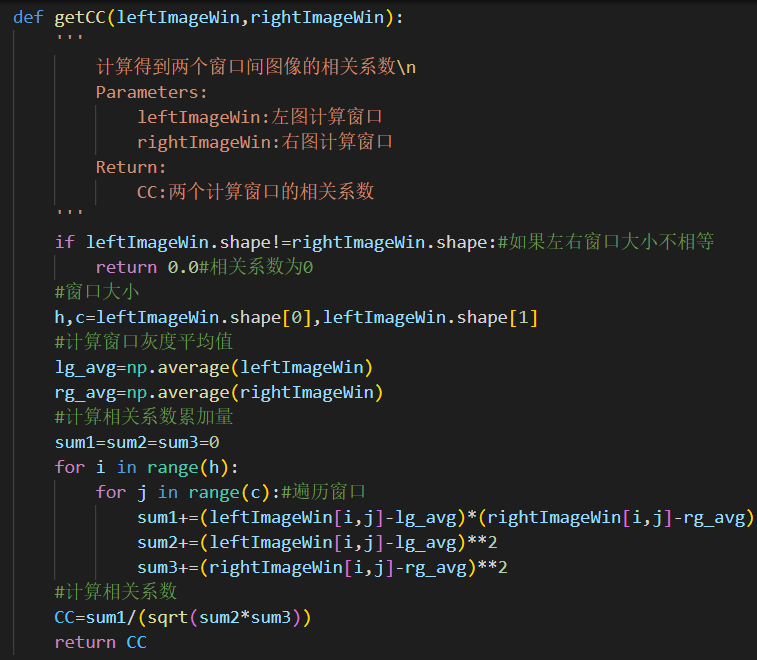
这里引用的是点特征提取中的Moravec算法，特征点绘制也不过多赘述，和上述模块一致，可见【第三节1.1】。





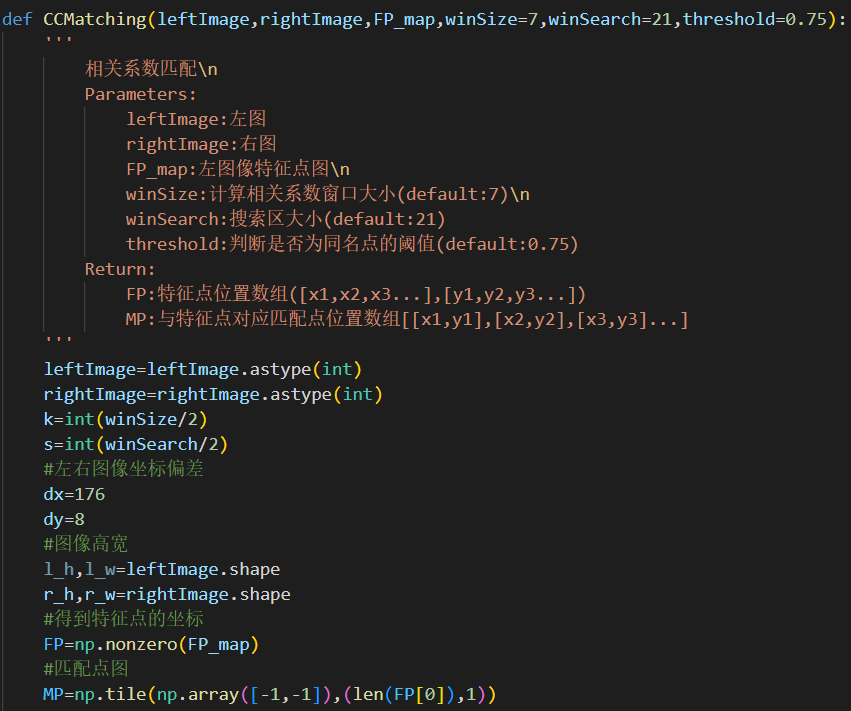
1. 计算相关系数

相关系数的计算封装在getCC函数中，其输入输出均可见下图。基本思想就是根据输入的两个窗口进行遍历，然后根据公式求相关系数。

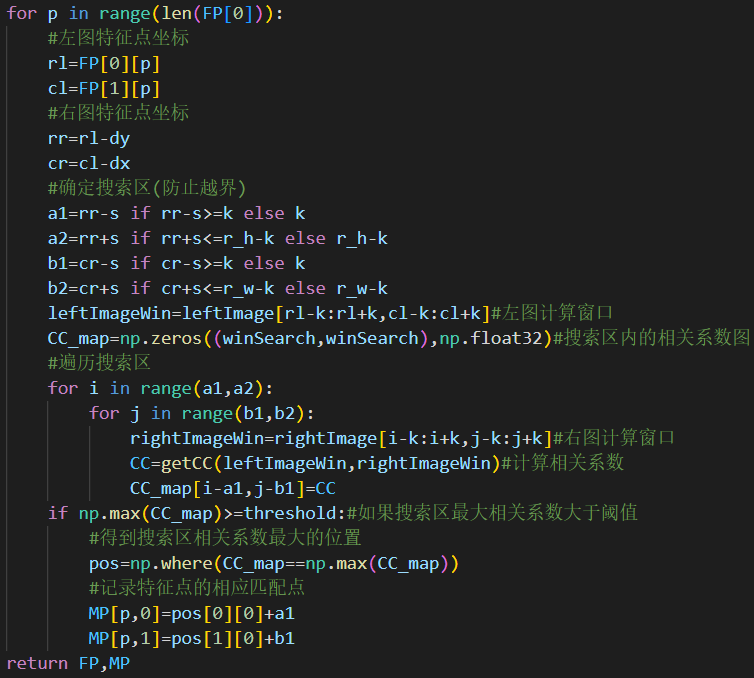


1. 确定匹配点

确定匹配点的功能封装于CCMatching函数，函数的输入和输出可见下图。

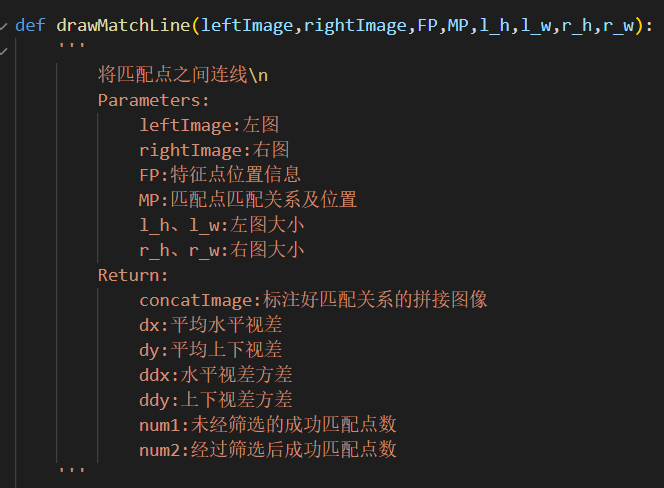


首先遍历每个特征点，根据大致视差和搜索区域大小确定其对应的搜索区域；然后对搜索区域内的每个像素的计算窗口都将其和目标窗口进行相关系数计算；一个搜索区域内最大相关系数对应的像点，若其相关系数大于给定阈值，则该点确定为该特征点的匹配点。记录下特征点信息FP和对应匹配点信息MP，其组织结构可见函数信息。

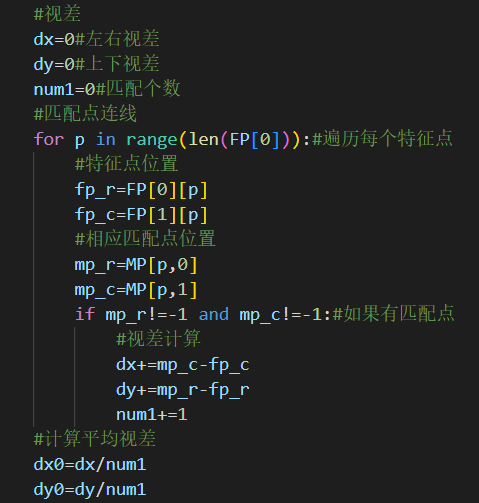


1. 筛选匹配点、结果图绘制、结果输出

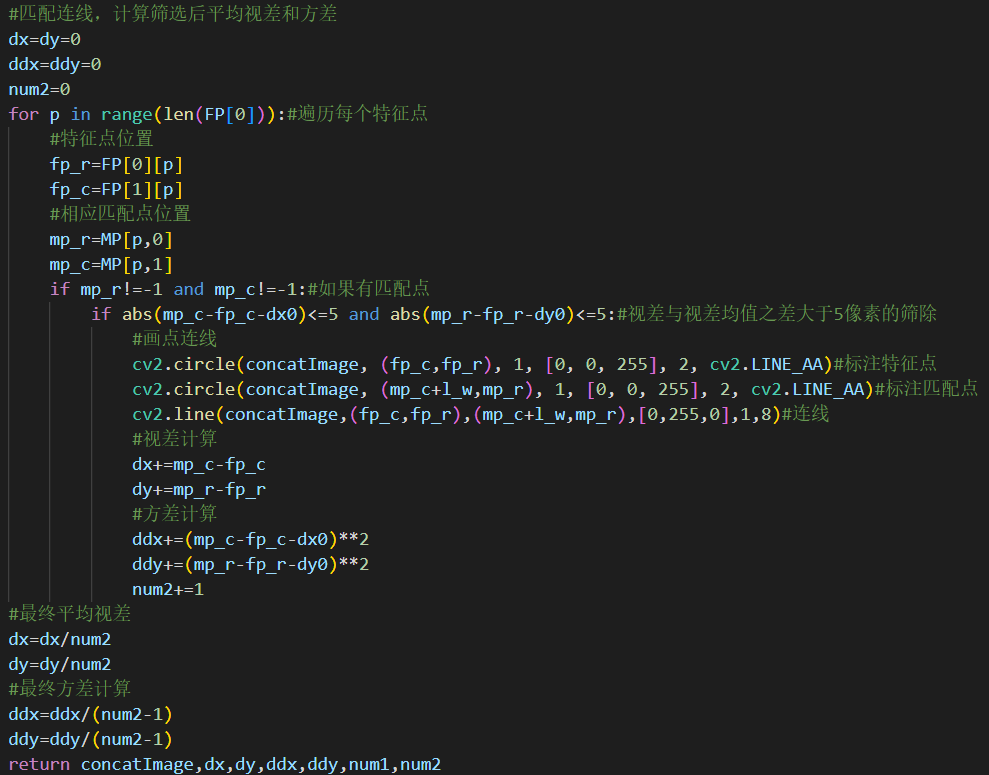
这部分内容封装于drawMatchLine函数中，函数详细信息可见下图。



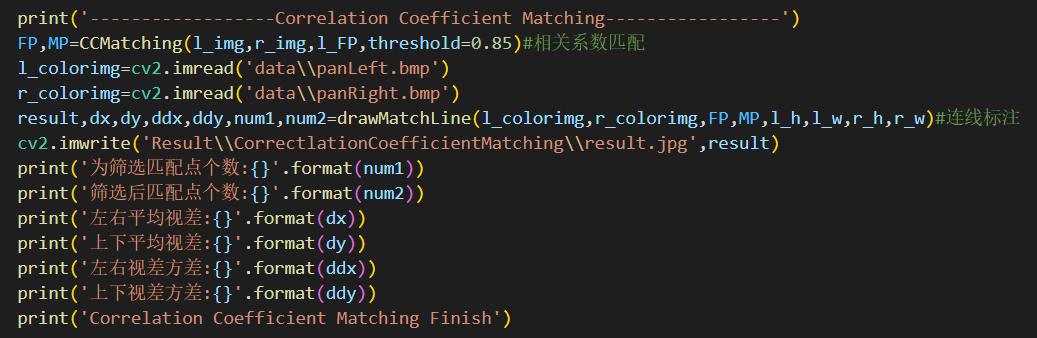
首先对之前匹配成功的点进行平均视差的计算，将平均视差作为视差真值（由于偶然误差的随机性，所以平均误差可近似看做真值）。



在次遍历匹配点，筛选出视差与真值之差的绝对值在5px内的点作为最终匹配点。并且在图中连线标注，并计算此时的平均视差和视差方差。



最后进行结果保存和输出。



# 四、试验结果及分析

## 点特征提取

### 1.1结果分析

点特征提取部分我分别使用chess.bmp图像和panLeft.bmp图像进行了实验，具体结果如下。

1. **Moravec算法**
2. chess图像结果：

经过试验，最终结果参数如下：

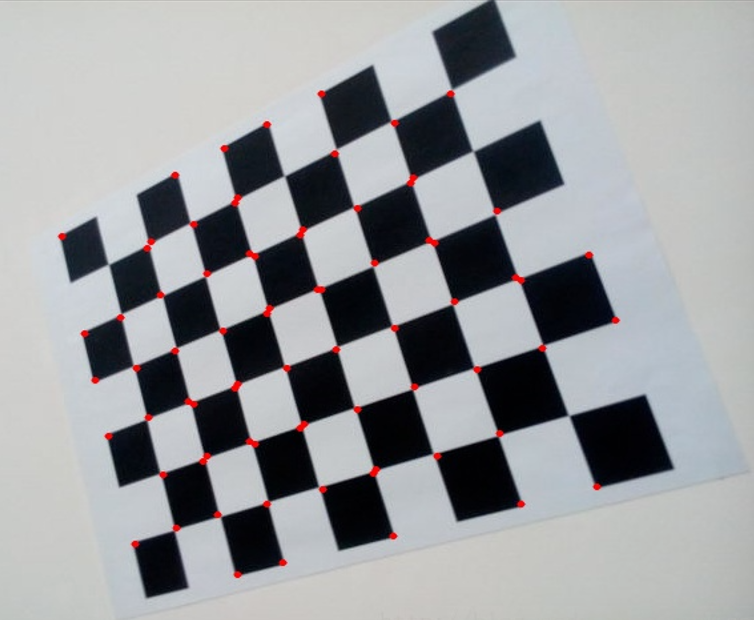
·阈值：6000

·兴趣窗口大小：5x5

·抑制窗口大小：5x5

由结果图可以知道，Moravec算法完成了大部分角点的提取，但是仍然有些许角点没有提取出来。值得一提的是这些未被提取出来的点大多数靠近图像边界。就运行速度而言，Moravec算法比较快，因为其计算量并不大。总体而言，在小计算量的前提下，比较好的完成了特征点的提取工作。

本项目还保存了Moravec算法运行过程中的兴趣图、候选点图、特征点图，均可在Result->PointFeatureExtraction->Moravec文件夹中查看，在此就不过多引用。



1. panLeft图像结果

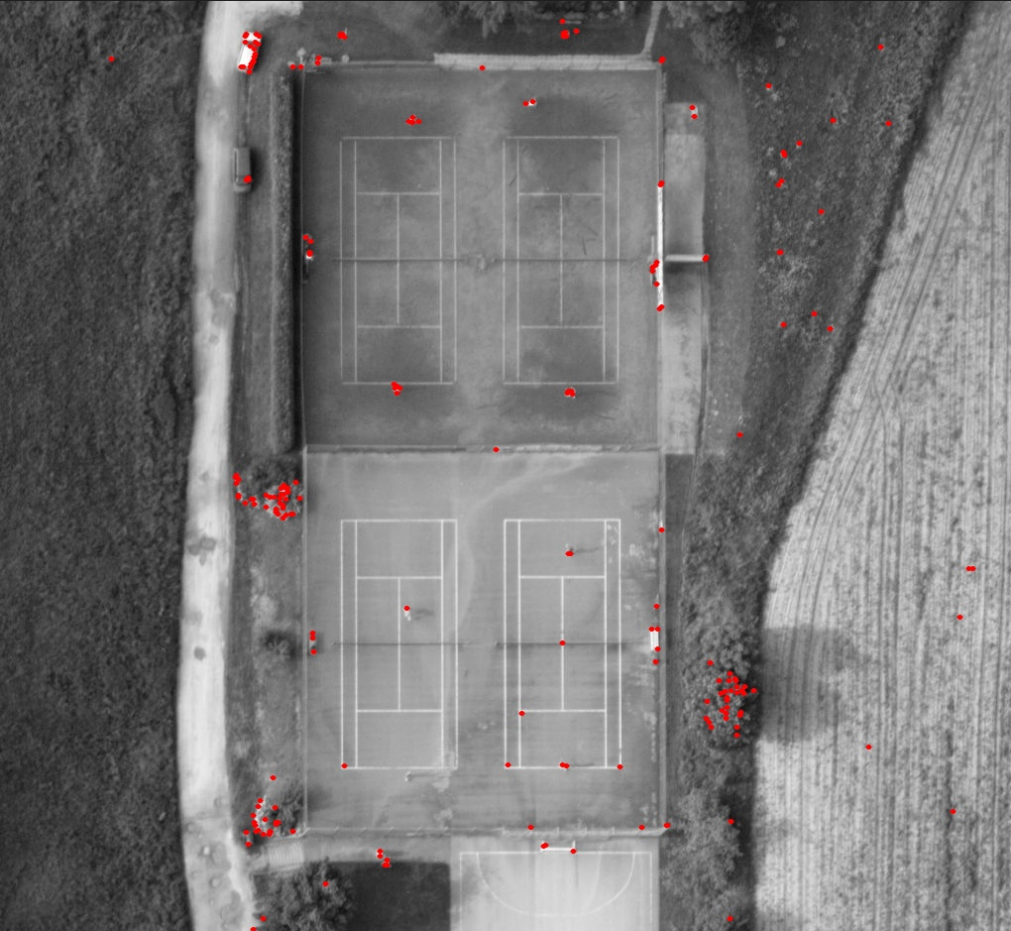
经过试验，最终结果参数如下：

·阈值：1200

·兴趣窗口大小：5x5

·抑制窗口大小：5x5

从结果图可以看到Moravec算法将网球场上的人、角点都提取了出来。球场旁边的两棵树上特征点比较多，这应该是由于其灰度变化比较频繁。



1. **Forstner算法**
2. chess图像结果

经过试验，最终结果参数如下：

·兴趣窗口大小：5x5

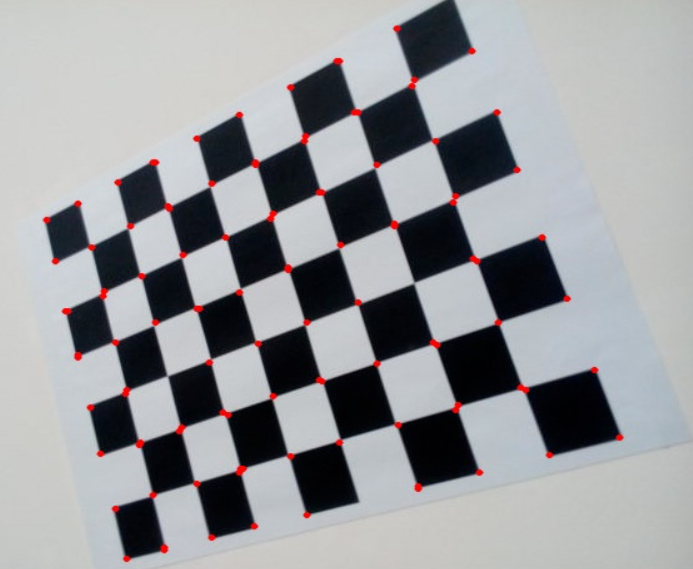
·抑制窗口大小：7x7

·初选阈值：20

·q阈值：0.5

·f、c：0.5、5

由结果图可以知道，Forstner算法完成了所有特征点的提取，效果上比Moravec算法更好，这是因为其计算的灰度协方差矩阵计算量更大，考虑的信息跟充分，但这导致了该算法在速度上较慢一些。



1. panLeft图像结果

经过试验，最终结果参数如下：

·兴趣窗口大小：5x5

·抑制窗口大小：7x7

·初选阈值：20

·q阈值：0.5

·f、c：0.5、5

由结果图可以看到，使用forstner算法提取出的特征点较Moravec算法提取的特征点要少一些，经过分析，认为可能对于这种灰度变化较为细密的图像而言，forstner的初始差分筛选会筛选掉一些不明显的特征点。



### 1.2阈值分析

点特征提取中阈值会直接影响到特征点的数量，我做了多次试验（部分试验结果如下表）。

就Moravec算法而言，根据计算的兴趣值IV，进行统计分析可以大致判断出较为合理的阈值在1000左右。然后设置多组不同阈值的实验，得到的特征点数目如下，根据结果图像分析，我认为阈值T=1200较为合适。当阈值过大时，一些特征点无法提取出来；当阈值过小时，很多非特征点会被提取出来。

|  |  |
| --- | --- |
| 阈值T | 特征点数目 |
| 1000 | 306 |
| 1100 | 253 |
| 1200 | 215 |
| 1400 | 152 |

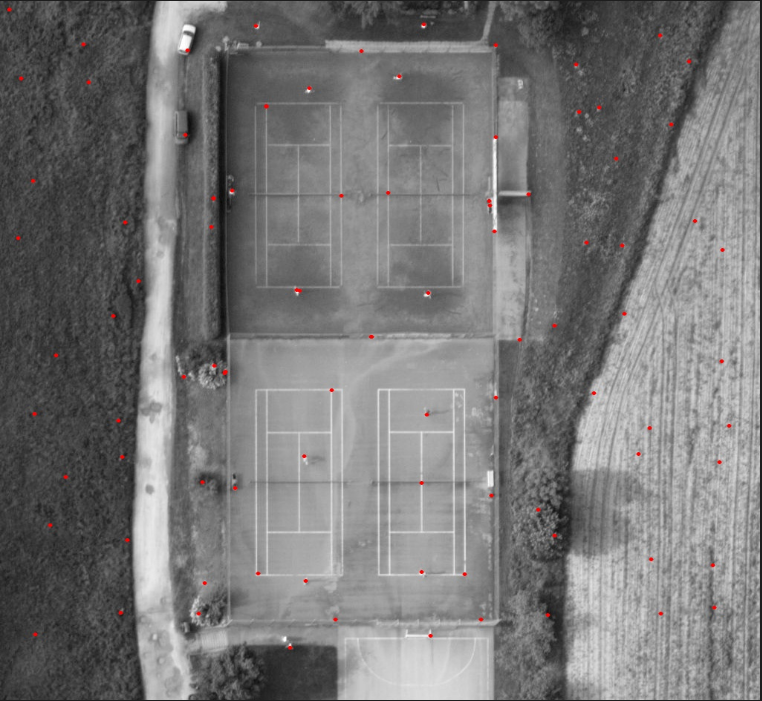
### 1.3随机/均匀提取实施

特征点提取有两种模式，一是随机提取，即全图使用统一的阈值，只要满足条件则为特征点（上述结果图即为随机提取），但这样也可能会导致一些特征点过于集中（如上图中的树叶有很多特征点），而一些区域又没有特征点。

基于此第二种方法就是均匀提取，即在每个区域内使用一个阈值，以求达到特征点尽量均匀分布的目的，不过这样的操作也有可能导致提取出非特征点。

具体实施方法为：通过降低阈值（比如将T=0），增大抑制窗口大小，来实现随机提取。降低阈值让全图大多数点成为候选点，增大抑制窗口，让一个抑制窗口内只能挑选出一个兴趣值最大的点作为特征点，这样就完成了均匀提取。操作和结果如下图：





## 相关系数法匹配

### 2.1特征点随机分布

下面是随机目标点的匹配效果图，所用参数如下所示：

·Moravec兴趣阈值：1000

·Moravec兴趣窗口：5x5

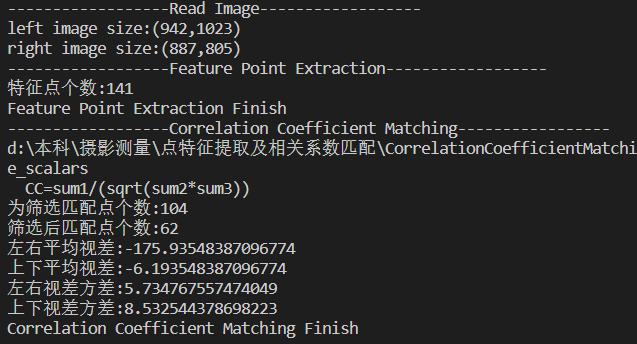
·Moravec抑制窗口：5x5

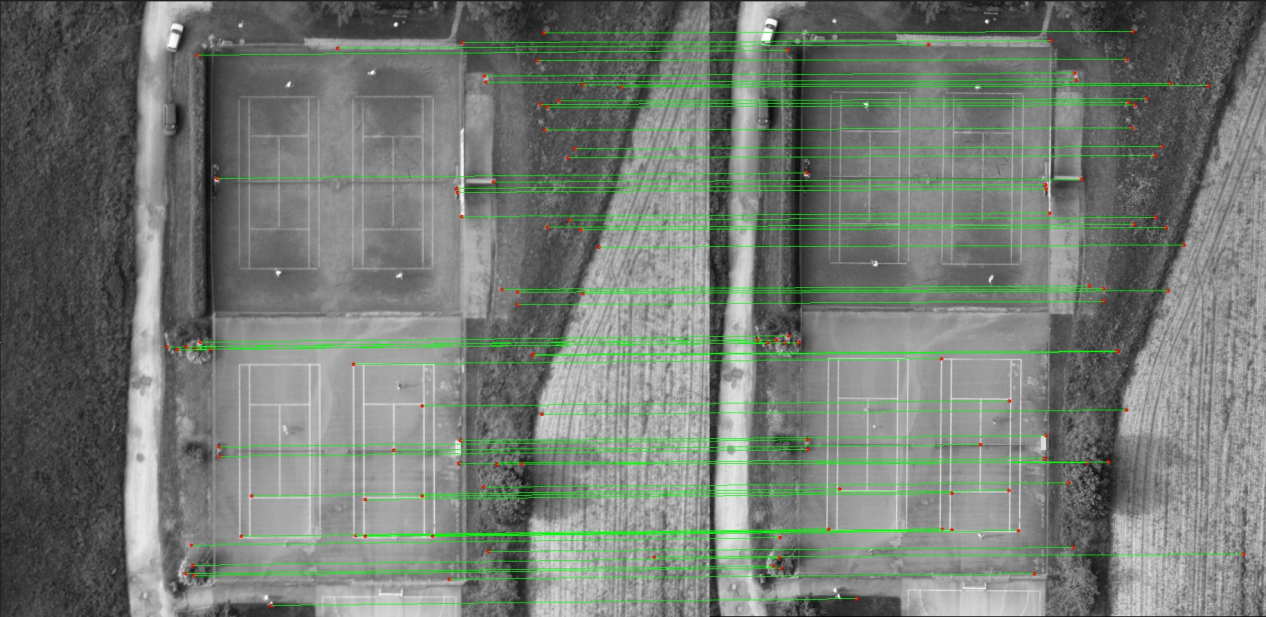
·相关系数匹配计算窗口：7x7

·相关系数匹配搜索区域：21x21

·相关系数阈值：0.85

由计算结果可知，左右视差约为-175.94px，上下视差约为-6.19px。





### 2.2特征点均匀分布

下面是均匀目标点的匹配效果图，所用参数如下所示：

·Moravec兴趣阈值：0

·Moravec兴趣窗口：5x5

·Moravec抑制窗口：100x100

·相关系数匹配计算窗口：7x7

·相关系数匹配搜索区域：21x21

·相关系数阈值：0.85

由计算结果可知，左右视差约为-176.46px，上下视差约为-6.46px。

