基于surf特征点的低重叠度图像拼接

报告人：李志宇2018202082

1. 题目说明

针对于重叠度低的图像研究基于图像特征匹配的图像拼接，实现多组图像对的拼接。

1. 问题分析

图像拼接是指将通过相机获取的具有一定重叠区域的两幅或多幅图像拼接成一张宽视角、高分辨率的全景图像。

本文的课题背景是横向分布的左右红外相机的实时拼接，对拼接的实时性提出了更高的要求。现阶段应用广泛的基于特征点图像拼接算法有基于SIFT（Scale Invariant Feature Transform）特征、SURF（Speeded Up Robust Features）特征。传统的SIFT算法能够提取更多的特征点，匹配精度高，但是特征提取和匹配是运算复杂，时间空间复杂度高。经测试，SURF特征检测虽然速度快但对于课题的低重叠度图像并不适用，综上所述，本文拟采用基于SIFT特征检测算法的进行图像拼接。虽然课题背景是左右相机的图像拼接，但在实际测试中涉及的是一组离线的静态图片，所有的算法设计路线均是依据静态图像拼接。对于具有固定结构的左右相机图像的拼接，只需要在首帧计算一次变换矩阵即可，大大缩短图像拼接时间，整个过程与静态图像拼接类似。

1. 算法说明

3.1 图像拼接流程

图像拼接过程由图像特征点提取、特征点匹配、图像配准、图像融合这四部分组成。图像配准和图像融合算法是比较成熟的技术，图像拼接算法特点重点集中特征点提取以及特征点匹配。



图2-1 图像拼接流程

若图像成像质量较差或者两张图片有过大的旋转倾斜，前期还有图像预处理阶段，主要是对输入图像进行去噪、平滑和旋转变换处理，以保证后续拼接的可靠性。特征点提取是图像拼接的核心前提，后续的图像配准使用的单应性矩阵（Homography Matrix）是依据重叠区域匹配后的特征点计算得到的，能否进行准确匹配取决于特征点提取精度。特征点匹配，是将左右图像提取的重叠区域的特征点进行一一匹配。图像配准是根据建立的数学模型，即二维图像的刚体变换—旋转和平移，将待拼接的图像转换到被匹配图像的坐标系中，一次构建完整的拼接图像。图像配准时，经常遇到坐标值非整数的情况，图像坐标插值理论应用到图像拼接算法中。图像融合是为了消除左右图像亮度变化在拼接处产生的拼接痕迹，一般在拼接处进行平滑处理。

3.2 特征点检测与匹配

计算机视觉中，引入尺度不变的特征，主要的思想是每个检测到的特征点都伴随着对应的尺寸因子。当匹配不同图像时，经常会遇到图像尺度不同的问题，不同图像中特征点的距离变得不同，物体变成不同的尺寸，若修正特征点的大小，就会造成强度不匹配。



图2-2 特征检测流程

SIFT算法流程如图2-2，利用图像金字塔生成尺度空间确定候选点，然后进行非极大值抑制，筛选出真正的特征点，降低运算复杂度，同时精确定位特征点坐标，选取特征点的主方向，最后利用特征点方向构造特征点的描述子。以下分部分介绍SIFT算法流程。

3.2.1 尺度空间生成及极值检测

在一定的范围内，无论物体是大还是小，人眼都可以分辨出来。然而计算机要有相同的能力却不是那么的容易，在未知的场景中，计算机视觉并不能提供物体的尺度大小，其中的一种方法是把物体不同尺度下的图像都提供给机器，让机器能够对物体在不同的尺度下有一个统一的认知。在建立统一认知的过程中，要考虑的就是在图像在不同的尺度下都存在的特征点。

1. **高斯金字塔**

在早期图像的多尺度通常使用图像金字塔表示形式。图像金字塔是同一图像在不同的分辨率下得到的一组结果，其生成过程一般包括两个步骤：

1. 对原始图像进行平滑

2) 后的图像进行降采样（通常是水平、垂直方向的1/2）降采样后得到一系列不断尺寸缩小的图像。显然，一个传统的金字塔中，每一层的图像是其上一层图像长、高的各一半。多分辨率的图像金字塔虽然生成简单，但其本质是降采样，图像的局部特征则难以保持，也就是无法保持特征的尺度不变性。

我们还可以通过图像的模糊程度来模拟人在距离物体由远到近时物体在视网膜上成像过程，距离物体越近其尺寸越大图像也越模糊，这就是高斯尺度空间，使用不同的参数模糊图像（分辨率不变），是尺度空间的另一种表现形式。

一副图像其高斯尺度空间可由其和不同的高斯卷积得到:



其中，G(x,y,σ)是高斯核函数 , σ称为尺度空间因子，它是高斯正态分布的标准差，反映了图像被模糊的程度，其值越大图像越模糊，对应的尺度也就越大。L(x,y,σ)代表着图像的高斯尺度空间。

构建尺度空间的目的是为了检测出在不同的尺度下都存在的特征点，而检测特征点较好的算子是(高斯拉普拉斯,LoG):



使用LoG虽然能较好的检测到图像中的特征点，但是其运算量过大，通常可使用DoG（差分高斯，Difference of Gaussina）来近似计算LoG[Marr and Hidreth]。

设k为相邻两个高斯尺度空间的比例因子，则DoG的定义：



其中，L(x,y,σ)是图像的高斯尺度空间。

从上式可以知道，将相邻的两个高斯空间的图像相减就得到了DoG的响应图像。为了得到DoG图像，先要构建高斯尺度空间，而高斯的尺度空间可以在图像金字塔降采样的基础上加上高斯滤波得到，也就是对图像金字塔的每层图像使用不同的参数σ进行高斯模糊，使每层金字塔有多张高斯模糊过的图像。降采样时，金字塔上边一组图像的第一张是由其下面一组图像倒数第三张降采样得到。

1. DoG空间极值检测及定位

为了寻找尺度空间的极值点，每个像素点要和其图像域（同一尺度空间）和尺度域（相邻的尺度空间）的所有相邻点进行比较，当其大于（或者小于）所有相邻点时，该点就是极值点。如图所示，中间的检测点要和其所在图像的3×3邻域8个像素点，以及其相邻的上下两层的3×3领域18个像素点，共26个像素点进行比较。

从上面的描述中可以知道，每组图像的第一层和最后一层是无法进行比较取得极值的。为了满足尺度变换的连续性，在每一组图像的顶层继续使用高斯模糊生成3幅图像，高斯金字塔每组有S+3层图像，DoG金字塔的每组有S+2组图像。

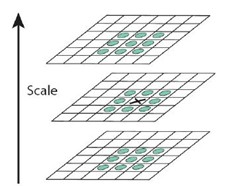


图2-2 极值检测

1. 非极大值抑制

通过比较检测得到的DoG的局部极值点实在离散的空间搜索得到的，由于离散空间是对连续空间采样得到的结果，因此在离散空间找到的极值点不一定是真正意义上的极值点，因此要设法将不满足条件的点剔除掉。可以通过尺度空间DoG函数进行曲线拟合寻找极值点，这一步的本质是去掉DoG局部曲率非常不对称的点。

要剔除掉的不符合要求的点主要有两种：低对比度的特征点；稳定的边缘响应点。

3.2.2 特征点主方向定位

为了实现图像旋转不变性，需要给特征点的方向进行赋值。利用特征点邻域像素的梯度分布特性来确定其方向参数，再利用图像的梯度直方图求取关键点局部结构的稳定方向。

找到了特征点，也就可以得到该特征点的尺度σ，也就可以得到特征点所在的尺度图像 计算以特征点为中心、以3×1.5σ为半径的区域图像的幅角和幅值，每个点L(x,y)的梯度的模m(x,y)以及方向θ(x,y)可通过下面公式求得





计算得到梯度方向后，就要使用直方图统计特征点邻域内像素对应的梯度方向和幅值。梯度方向的直方图的横轴是梯度方向的角度（梯度方向的范围是0到360度，直方图每36度一个柱共10个柱，或者没45度一个柱共8个柱），纵轴是梯度方向对应梯度幅值的累加，在直方图的峰值就是特征点的主方向。Lowe在论文中指出15%的关键点具有多方向，而且这些点对匹配的稳定性很关键。得到特征点的主方向后，对于每个特征点可以得到三个信息(x,y,σ,θ)，即位置、尺度和方向。由此可以确定一个SIFT特征区域，一个SIFT特征区域由三个值表示，中心表示特征点位置，半径表示关键点的尺度，箭头表示主方向。具有多个方向的关键点可以被复制成多份，然后将方向值分别赋给复制后的特征点，一个特征点就产生了多个坐标、尺度相等，但是方向不同的特征点。

3.2.3 生成描述子

通过以上的步骤已经找到了SIFT特征点位置、尺度和方向信息，下面就需要使用一组向量来描述关键点也就是生成特征点描述子，这个描述符不只包含特征点，也含有特征点周围对其有贡献的像素点。描述子应具有较高的独立性，以保证匹配率。

特征描述符的生成大致有三个步骤：

校正旋转主方向，确保旋转不变性。

生成描述子，最终形成一个128维的特征向量

归一化处理，将特征向量长度进行归一化处理，进一步去除光照的影响。

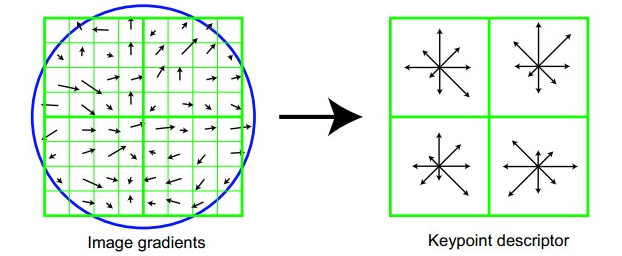
旋转后以主方向为中心取 8×8的窗口。下图所示，左图的中央为当前关键点的位置，每个小格代表为关键点邻域所在尺度空间的一个像素，求取每个像素的梯度幅值与梯度方向，箭头方向代表该像素的梯度方向，长度代表梯度幅值，然后利用高斯窗口对其进行加权运算。最后在每个4×4的小块上绘制8个方向的梯度直方图，计算每个梯度方向的累加值，即可形成一个种子点，如右图所示。每个特征点由4个种子点组成，每个种子点有8个方向的向量信息。

图2-3 种子点生成描述子

对每个关键点使用4×4共16个种子点来描述，这样一个关键点就可以产生128维的SIFT特征向量。

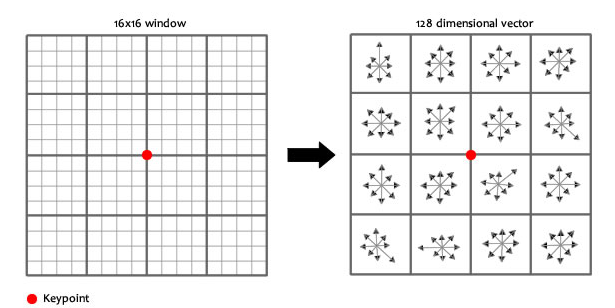


图2-4 128维特征点描述子

2.2.6 特征点匹配

与Sift特征点匹配类似，Surf也是通过计算两个特征点间的欧式距离来确定匹配度，欧氏距离越短，代表两个特征点的匹配度越好。不同的是Surf还加入了Hessian矩阵迹的判断，如果两个特征点的矩阵迹正负号相同，代表这两个特征具有相同方向上的对比度变化，如果不同，说明这两个特征点的对比度变化方向是相反的，即使欧氏距离为0，直接予以排除。

2.3 图像配准

两幅平面图像之间的转换关系右一个单应性矩阵（Homography Matrix）H决定，单应性矩阵H形式如下：



假设左右图像上对应匹配后的齐次坐标为和，则有：



矩阵展开后有等式：





由于是齐次坐标系，也就是说可以进行任意尺度的缩放，单应矩阵H只有8个自由度，求解H至少需要四对点，通常会用多于四对点计算H矩阵。

得到H矩阵后，就能使用H矩阵利用将右图像的点转换到左图像坐标系下，完成配准。

2.4 图像融合

两图直接拼接并不自然，原因就在于拼接图的交界处，两图因为光照色泽的原因使得两图交界处的过渡很糟糕，所以需要特定的处理解决这种不自然。这里的处理思路是加权融合，在重叠部分由前一幅图像慢慢过渡到第二幅图像，即将图像的重叠区域的像素值按一定的权值相加合成新的图像，本实验采用的是亮度调整后按距离比例融合，这样融合比较自然。

1. 程序说明

计算机视觉市场巨大而且持续增长，且这方面没有标准API，OpenCV致力于真实世界的实时应用，通过优化的C代码的编写对其执行速度带来了可观的提升，OpenCV拥有大量比较成熟的视觉应用案例。本程序开发基于Visual Studio2015，OpenCV3.4.0和对应的Contrib3.4.0。

4.1 程序总体流程



图4-1 程序总体流程

如图所示，为程序的总体流程。首先根据图像路径载入待拼接的图像，同时检查图像是是否载入正确。若正常载入进行特征点检测，若图像载入为空，则结束程序，检查路径是否错误。找到图像上的特征点后，根据最邻近算法将左右图像的对应点匹配，然后判断匹配后的点对是否足够求解单应性矩阵(至少四个，一般需要更多)。若数据点足够多，那么使用点对求解单应性矩阵；若点数过少，则使用预先存储的单应性矩阵。求的单应性矩阵，以左图像坐标系为标准，将右图像按照单应性矩阵转换到左图像坐标下，完整直接拼接。最后在拼接处进行图像优化融合，使得拼接更加自然。

4.2程序关键函数

1. int ImageStitch::FindFeaturesPoints(void)

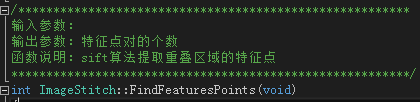


图4-2 检测特征点函数

本函数利用的关键是SURF类函数Ptr<Feature2D> Detector = xfeatures2d::SIFT::create(2000);

1. void ImageStitch::ImageRegistration(void)

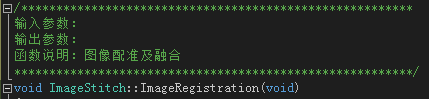


图4-3 图像配准函数

本函数是核心函数，其中会调用关键函数 Mat findHomography( InputArray srcPoints, InputArray dstPoints, OutputArray mask, int method = 0, double ransacReprojThreshold = 3 )计算单应性矩阵，然后完成图像直接拼接；最调用图像融合优化函数OptimizeSeam(void)使得拼接自然。

1. void ImageStitch::OptimizeSeam(void)



图4-4 图像融合优化函数

图像融合优化函数，按距离分配权重值。

4.3 运行结果

第一组图片：

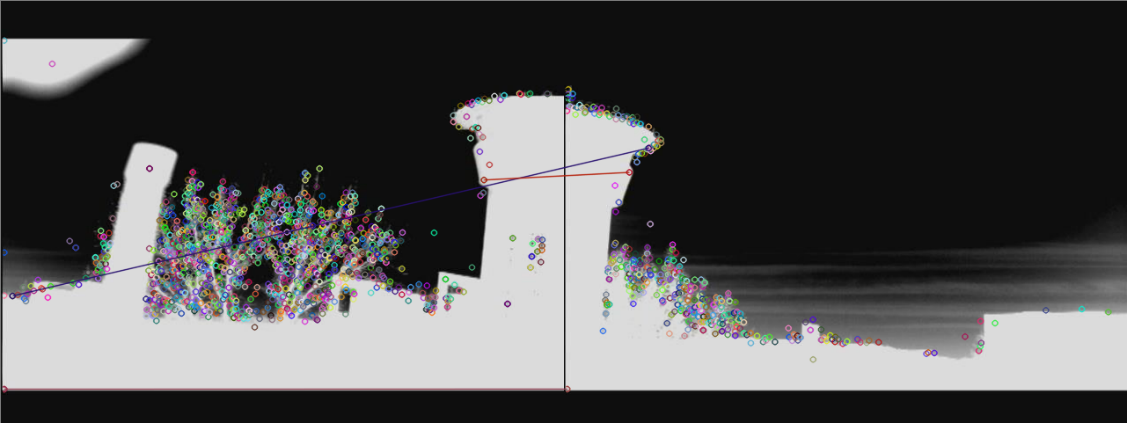
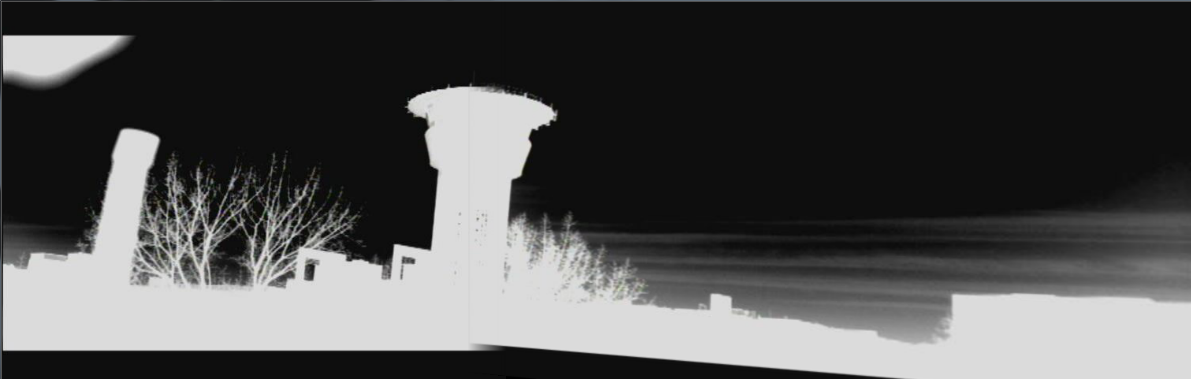
图4-5匹配后的特征点

图4-6最终拼接图

第一组图片由于检测的左右图像重叠区域的特征过少，无法求出单应性矩阵H，使用后面计算的并存储的单应性矩阵拼接。

第二组图片：



图4-7 匹配后的特征点

图4-8 最终拼接图

第三组图片：

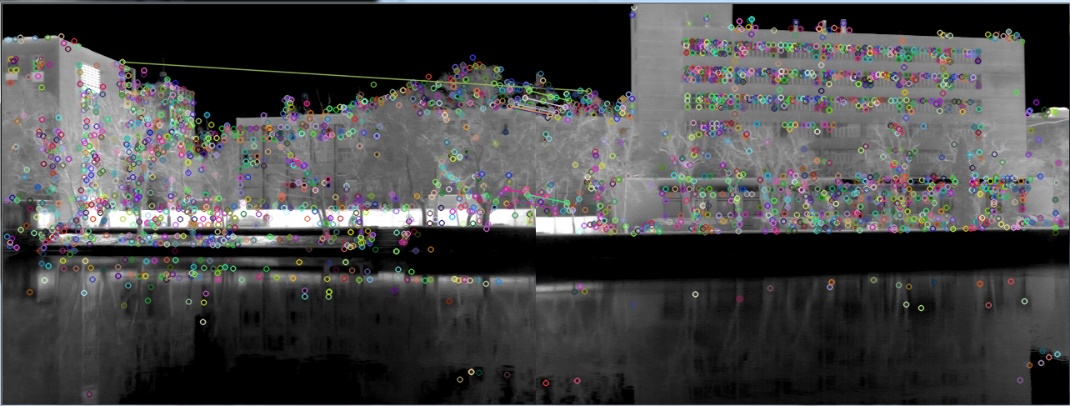
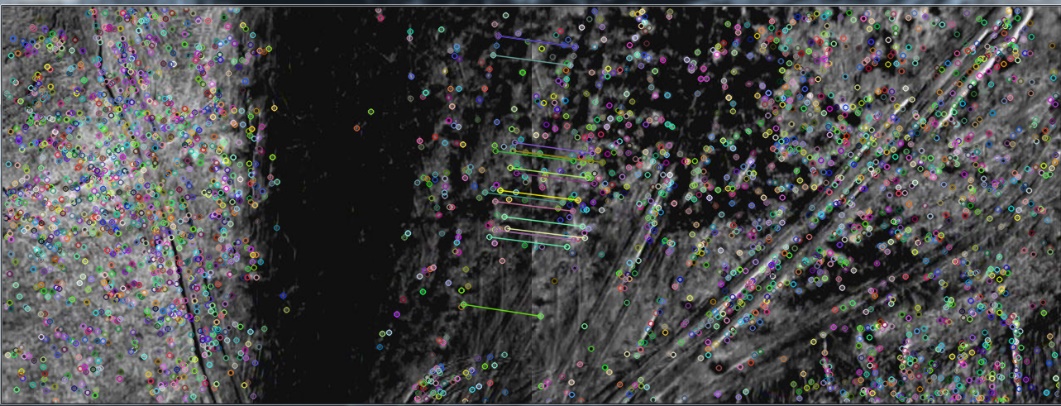


图4-9 匹配后的特征点

图4-10 最终拼接图

第四组图片：



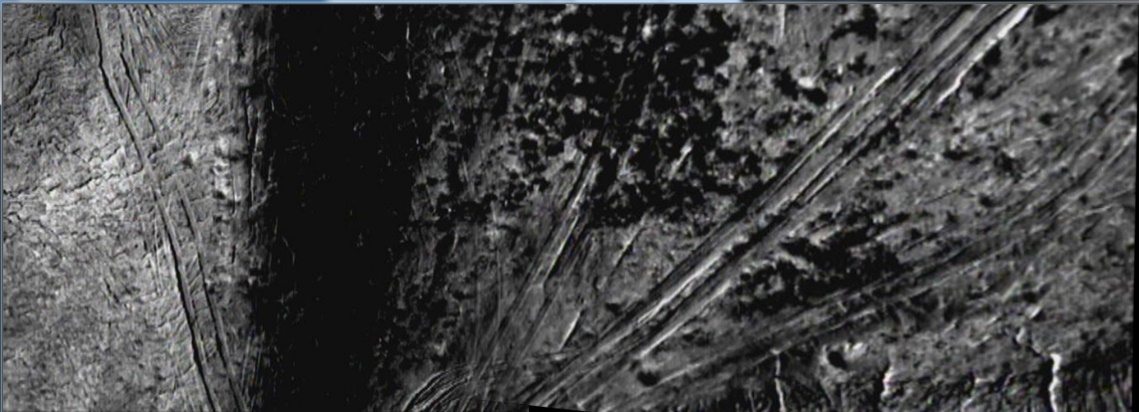
图4-11 匹配后的特征点

图4-12 最终拼接图

除了第一组图片由于检测的左右图像重叠区域的特征过少，无法求出单应性矩阵H，后面三张图片均能正常计算。由于课题背景是固定左右相机的图像拼接，所以采用后面计算得到的单应性矩阵做第一组的图像拼接是合理的，拼接效果也在接受范围内。

1. 附录

本程序开发基于Visual Studio2015，OpenCV3.4.0和对应的Contrib3.4.0。可以直接运行压缩包里的DEBUG里的exe文件。

1. 结论说明

在课题开始之前，自己专注于使用surf特征检测的快速，但最后发现对于特征点不明显的红外图像，surf算法检测的特征点不够精确，计算的单应性矩阵在做图像拼接时误差较大，改用sift精度明显提高。通过本课题，在尝试中提高自己分析问题和实践操作的能力。