分类号：TP3　　密级：公开

编号：212015449

桂林理工大学

硕士研究生学位论文

（专业学位）

鱼眼镜头下基于ORB算法的图像拼接技术研究

Research on Image Mosaic Technology Based on ORB Algorithm under Fisheye Lens

学位类别 工程硕士

研 究 生 李东辉　**学　　号** 212015449

工程领域 计算机技术

研究方向 数字图像处理

导　　师 李 新　**职　　称** 副教授

2016 年 4 月至 2017 年 4 月

中国知网 https://www.cnki.net

Research on Image Mosaic Technology Based on ORB Algorithm under Fisheye Lens

Major：Technology of Computer

Direction of Study：Digital Image Processing

Graduate Student：Li Donghui

Supervisor：A.P.Li Xin

College of Information Science and Engineering

Guilin University of Technology

April, 2016 to April, 2017

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net

摘　要

鱼眼图像有着视场大范围广的优点，但其采集的图像往往会有很大的变形，这就会对鱼眼图像的拼接造成阻碍。因此，若要有效利用鱼眼镜头采集的宽视场图像，就需要把畸变的图像经过透视变换转化成无形变的图像，然后再进行图像的拼接，这样才能使鱼眼图像的拼接获取最大的利用价值。本文针对鱼眼镜头采集的图像，展开了对图像拼接技术的研究工作。

鱼眼图像的拼接有以下的几个组成部分：鱼眼图像的校正部分，介绍了两种不同的校正法：基于球面透视投影约束的校正法和基于经度定位的校正法，基于经度定位的校正算法丢失像素较少，能够满足实验的需要；在特征点的检测部分，采用改进的 ORB（Oriented BRIEF）算法，该算法在传统 ORB 的基础上，借助 SURF（Speed-Up Robust Features）算法的思想，使 ORB 算法具有了尺度不变性，提高了特征点检测的正确率。ORB 相对于 SIFT 和 SURF 不但有着高的匹配率，而且在匹配的速度上也有着很大的提升，改进的 ORB 算法虽然在时间上相对于传统的 ORB 算法有所增加，但是改进算法具有尺度不变性，可以检测出更多的特征点，进而增加了匹配的精确度；特征点的提纯部分，运用了 PROSAC（Progressive Sample Consensus）剔除误匹配的特征点，PROSAC 在消除错误匹配的速度上要快于 RANSAC（Random Sample Consensus）；图像融合部分，由于在采集图像过程中的视角、光照变化等因素，会导致融合后的图像会出现重影、拼接线、图像像素缺失等现象，本文采用渐入渐出以及双线性插值的方法消除融合后图像产生这些现象，进而使拼接后的图像更加自然。

本文使用 Matlab R2015b 和 Opencv+Visual Studio2015 相结合编程实现以上实验算法。在校正实验中，对比基于球面透视投影约束的校正法和基于经度定位的校正法，并选取后者作为本文的校正算法；在图像特征点提取实验中，与传统的 ORB 算法进行比较。实验表明：采用基于经度定位的校正算法可以很好地保留鱼眼图像的原有信息，当图像的尺度发生变化时，改进的 ORB 算法能够检测到更多的特征点，更有利于图像拼接的进行。

关键词：图像拼接，鱼眼图像校正，ORB，PROSAC，图像融合

中国知网　https://www.cnki.net

Abstract

Fisheye image has the advantages of wide field of view, but it will be very large deformation, which will hinder the image stitching. Therefore, if we want to make full use of fisheye image, that requires the distortion after image perspective transform into images without distortion, and then conduct the image stitching, so as to make mosaic fisheye images obtained by using the maximum value. This paper focuses on the research of image stitching technology based on fisheye lens.

The stitching of fisheye image has the following components. In the image calibration, two different calibrations are introduced. The calibration algorithm based on spherical perspective projection constraint and the calibration method based on latitude and longitude are used in this paper. The latter is used to correct the distorted fisheye images. This algorithm has fewer pixels lost and can meet the needs of the experiment. In the aspect of feature points detection, improved ORB(Oriented BRIEF) algorithm is adopted. On the basis of SURF(Speed-Up Robust Features), the improved algorithm ORB has the dimensionality invariance and improves the correct rate of the feature point detection. Compared with the algorithms SIFT and SURF, the speed of matching has a big improvement while ensuring matching accuracy. Although the improved ORB algorithm has increased in time compared with the traditional ORB algorithm, it has scale invariance and detect more feature points and increase the matching accuracy. In the aspect of the purification of feature points, PROSAC (Progressive Sample Consensus) was used to remove the mismatched feature points. PROSAC is faster than RANSAC(Random Sample Consensus) in eliminating error matching. In the image fusion, because of the angle of view and change of illumination or other factors in the process of collecting images, it will lead to the phenomenon of ghosting, splicing and other phenomena. This paper adopts the gradual elimination algorithm to eliminate these phenomena, which in turn makes the spliced image more natural.

This paper use Matlab R2015b and Opencv，Visual Studio2015 for the above process experiments. In the calibration experiment, compare calibration algorithm which based on spherical perspective projection and algorithm which based on the latitude and longitude location, choose the latter as calibration algorithm in this paper. In the feature extration experimrnt, comparing the traditional ORB algorithm. The results of experiment show that the calibration algorithm based on the latitude and longitude location can retain the original

中国知网 https://www.cnki.net

fisheye image information when the image scale changes, the improved ORB algorithm can detect more feature points, and this is conducive to image mosaic.

Keywords: Image stitching, Fisheye image calibration, ORB, PROSAC, Image fusion

中国知网 https://www.cnki.net

目 录

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net

第 1 章 绪论

本章主要是对论文研究的背景、意义以及国内外的现状等进行了概括性的总结，并在本章的最后对本文的结构布局进行了叙述。

1.1 课题背景及研究意义

因为图像采集设备对于视场的限制，导致没法获得一幅全景的图像，在某些特殊的应用场景中，需要将有着公共区域的多张图之间拼接成视野宽阔的图像进而使人们观察到的图像视野更广阔。

鱼眼图像拼接分成三个主要的过程：第一是图像的校正，主要是对圆形的鱼眼图像进行处理，以更有利于图像融合的进行。第二是图像配准，主要内容是进行特征的抽取和匹配。第三是图像融合，寻求图像间的映射关系。在图像的拼接中，所使用的镜头包含两类：第一类是普通的镜头采集到的无变形的图像和由鱼眼镜头采集到的有变形的图像。普通镜头采集到的图像范围有限，这是因为该镜头的视角比较小的原因，因此要组成一幅全景图，就需要有多幅具有公共区域的图像进行拼接而成。那么，处理多张图像的计算量都会大大的增加。

鱼眼镜头是一种特殊的成像设备，它有视野广，焦距短等的特点，因而拍摄出的范围更大。由于镜头的镜片凸起类似于鱼的双眼，所以称为鱼眼镜头[1]。由于鱼眼镜头的不同于普通镜头的结构，鱼眼图像所容纳的信息要多余普通的镜头，但鱼眼镜头拍摄的鱼眼图像的缺点是，拍摄的图像会发生变形。在鱼眼图像中，信息量最大的是在图像的圆心附近，这里的形变是最小的，但是随着圆形区域的半径的增加，拍摄到的信息量就会减少，形变量就越来越大。

图像拼接技术的主要应用场景有：

（1）遥感图像处理

在遥感图像的摄影中，由于要进行观测的目标的尺寸一般很大、并且该遍及的范围广。因此就需要把从各个位置，以及同一地方不同时期的图像进行合并，形成一个宽幅的图像。例如，在全景图像下，我们可以更加便捷地对大气污染，土地资源等进行全面的检测。

（2）智能监控技术

在监控系统中，往往是由多个设备获取的影像独立地显示在显示器上面，因此若是将多路影像合成一幅大视角的视频，这将会对监控领域产生极大的便捷性。对于全面地了解一个场景的情况，很有实用的必要性。

中国知网 https://www.cnki.net

（3）虚拟现实技术

虚拟现实技术指的是在计算机的协助下，对采集到的大量数据进行操作，并使数据可以与人交互，这种技术可以把数据变成可视化的数据，从而方便处理[2]。利用该技术，可以创建一个类似于现实的场景，从而以图像的方式展现在人们的面前。图像拼接技术在全景图的生成中扮演着重要的角色，先把生成的虚拟影像拼接起来，然后再投射到空间内。

1.2 国内外研究现状

1.2.1 国外研究和发展现状

2004 年，SIFT（Scale Invariable Feature Transformation）算法被提出，当前国内外最流行的是改进 SIFT 算法。2004 年，CVPR 国际顶级计算机视觉会议上，CMU 的研究人员提出了改进的尺度不变特征变换算法，改进后的 SIFT 算法在性能上相对于经典 SIFT 有着很大的提升。2006 年，英国 University of Cambridge 的研究人员提出了 FAST 算法，该算法是对图像中的角点进行检测，因为该算法独特的角点检测功能，所以它对角点特别敏感，运算速度也相对较高，在角点检测上具有很大的优势[3]。2010 年的 CVPR 计算机视觉国际会议上 Ondrej Chum 提出了 BRIEF 算子，该算子的所占用的空间是 SURF 的 1/4，因此可以进一步提高算法的匹配速度[4]。2012 年的计算机视觉国际会议上，Alahi 等人提出了 FAST Retina Keypoint 特征描述方法，其原理是仿照视网膜上的细胞，但由于在实际的应用中会有大量误匹配的出现，因此在实际的运用过程中还需要对该算法作进一步的改进。2016 年 Emilio Garcia-Fidalgo 等人提出的一种基于二进制字符串的视觉检索的算法，该算法是在线处理数据，从而有效地避免了需要训练后才能进行视频检索的先决条件，实验表明了在不同的环境和成像设备参数下该算法依然可以保持很好的计算效果，并且在不同的场景下对同一事物的检测有着一样的结果，算法具有很强的鲁棒性[5]。

1.2.2 国内研究和发展现状

国内的研究者们的工作主要是在国外已有算法上面进行算法的改进。2000 年王建琦等，改进了 SUSAN 算法，主要是增强了该算法的抗噪能力，因为经典 SUSAN 算法的对于噪声的影响很敏感，不能够满足实际中的需要，因此改进后的算法在抵抗噪声的能力上具有很大的提升[6]。2008 年解放军理工大学的张春美等人对 SIFT 算法进行改进，主要是提高了该算法的运行速度，优化了该算法的计算性能，使运算量减小，从而加快算法的执行速度。2000 年王建琦等，改进了 SUSAN 算法，主要是增强了该

中国知网 https://www.cnki.net

算法的抗噪能力，因为经典 SUSAN 算法的对于噪声的影响很敏感，不能够满足实际中的需要，因此改进后的算法在抵抗噪声的能力上具有很大的提升。2011 年东南大学包加桐等，把特征点检测算法 SURF 迁移到对人的手势进行识别的算法上来，并把该算法应用到机器视觉上面来。2013 年王俊峰等，把 BRIEF 应用实际的生产中，把该技术应用到 AR 中去，获得了良好的生产效果[7]。2015 年哈尔滨工业大学王昌盛等人，将 ORB 算法和 PROSAC 算法应用到双目视觉测量上，提高了原始算法的精度和目标位置确定的准确性。2016 年中科院张敏等人，采用改进的 SIFT 算法对视频拼接技术进行了研究，实现了三路视频流相关试验。

国内研究学者于 2009 年提出适用于非动态大视场路视频的拼接技术，该技术的基础是多幅图像的拼接，在拼接的过程中可以将单张的图像与已经进行拼接好的图像直接进行拼接，计算过程得到一个变换矩阵，使用该拼接的顺序可以避免过程中误差的累加，从而使拼接过程更加精准。熊鹏等提出生成有预测图像块的拼接方法，该种方法不同于其它的拼接方式，首先计算拼接图像之间的公共区域的百分比，然后使用该比例形成一个区域，当进行两幅图像的特征点提取时，不需要对整幅的图像进行运算，直接对计算出的区域内的特征点进行检测，从而大大地提高了计算的速度，同时也避免了一些不必要的计算[8]。

拼接的图像会产生图像重影，颜色过渡不自然的问题。解决办法有 Brown M，Lowe D 使用的 Multi-ban，平均加权法，双线性插值等，以及国内的有李亮等提出的泊松融合方法[9]。

1.3 论文主要研究内容及章节部署

本文主要对鱼眼图像的拼接方法进行了研究，鱼眼图像的拼接流程主要分为四步：图像的校正、特征点提取、图像配准、图像融合。在四个步骤中进行了传统算法之间，以及改进算法与传统之间的对比，并结合实验结果对算法的优劣进行分析，最后得出了一种更为有效，合理的方法和结论，论文的流程框架如图 1.1 所示：

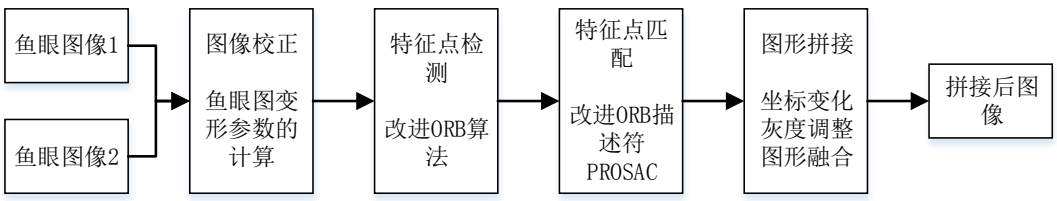


图 1.1 鱼眼图像拼接框架图

中国知网 https://www.cnki.net

第一章：介绍了鱼眼成像原理以及图像拼接的研究背景、意义、图像拼接技术的现状，国内外在该研究领域内的典型应用等。

第二章：分析鱼眼图像的校正方法，并进行了实验和对比。主要分析了基于球面透视投影约束和经度定位的校正方法。球面透视投影约束的方法是先依据空间中的直线形成椭圆的原理，来求得其中椭圆的半径和圆心参数大小。而后者是根据球面分布的经线模型，从而对畸变进行复原。经试验的对比，本文最后采用基于经度定位的校正方法进行鱼眼图像的校正。

第三章：主要对图像特征点的匹配算法进行了详细的分析。首先对常用的特征点检测的算法如：SIFT 尺度不变特征变化以及该算法的改进 SURF 等进行了介绍，然后又对它们的描述子的生成算法进行了解析。最后介绍了传统的 ORB 特征检测，同时对该算法进行了改进，使算法具有了尺度不变性，对比实验中与其它同类算法进行了比较，结果显示改进算法有利于检测更多的特征点。

第四章：介绍了随机抽样一致性算法 RANSAC 以及其改进的 PROSAC，并结合前面章节的特征点提取算法进行实验。实验结果证明，改进的具有尺度变化的 ORB 与 PROSAC 算法结合，可以有效的除去误匹配的点对并增加匹配的精确性。

第五章：分析了插值和融合的技术，并给出了鱼眼图像的校正和融合的完整实现过程。先使用改进的 ORB 算法和 PROSAC 算法对校正后的鱼眼图像进行拼接，接下使用渐入渐出法对图像进行融合，并采用双线性插值的方法调整拼接后图像的像素值，从而消除图像某些区域亮度的不一或者空洞像素的存在。

第六章：对本文所做的工作进行了概括和总结，并指出了基于改进 ORB 算法的鱼眼图像拼接技术所存在的一些不足之处，展望了基于该算法的图像拼接需要采取的改进措施。

中国知网 https://www.cnki.net

第 2 章 鱼眼图像校正研究

2.1 鱼眼图像校正

图 2.1 鱼眼图像图

图 2.2 经过校正后的图像

图 2.1 是由鱼眼镜头生成的具有畸变的图像，从图像中我们可以看到，处在圆形区域边缘弯曲的楼层变得笔直，边缘区域的校正将会有利于拼接的进行。因此，为了使鱼眼图像可以更好地进行融合，有必要对鱼眼图像进行校正，如图 2.2，是经过校正后的图像。

2.2 鱼眼图像投影模型

对于空间中有一个点，得到投影的方法是，先对该点进行球面的投影，然后再把这些球面的所有点进行透视投影，然后再转换为平面图像[10]。

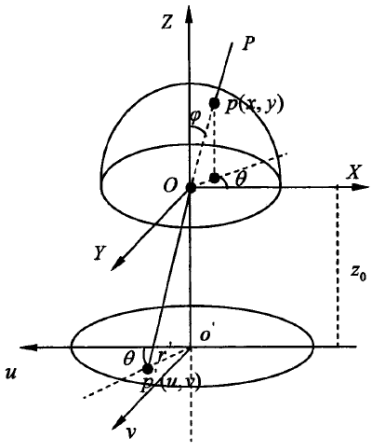


图 2.3 鱼眼图像成像校正模型

中国知网 https://www.cnki.net

如图 2.3 所示，设 Z_0 为一个平面到点的距离，该平面是经过转换后形成的平面与鱼眼镜头光心之间的距离。如图所示，空间中的点与光轴 OZ 之间的夹角大小记为 φ，则可以得到小孔成像各项参数如下：

小孔成像模型：$r'=z_0*\tan\varphi$；球面投影模型：$r'=2z_0*\tan(\frac{\varphi}{2})$；正交投影模型 $r'=z_0*\sin\varphi$；等距投影模型：$r'=z_0*\varphi$；等立体角投影模型 $r'=2z_0*\sin(\frac{\varphi}{2})$。

2.3 基于球面透视投影约束的校正

鱼眼图像在获取的过程中由于成像效果的不相同，可有两种类型：一是呈现圆形的鱼眼图像；另一种是不呈现圆形的，称为对角图像。

如图 2.4，图 2.5，表示的是在空间里面的某个点和某条直线，分别在两种不同的投影方法下所呈现的状态。

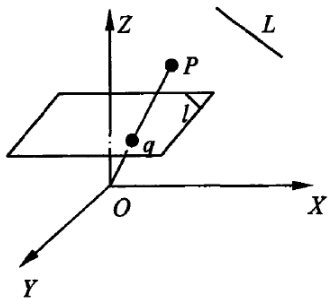


图 2.4 平面方法

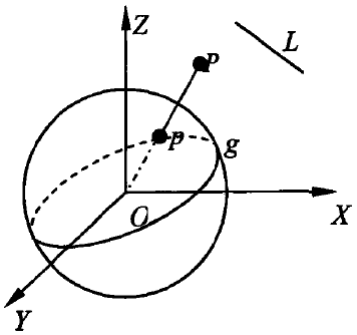


图 2.5 球面透视法

位于空间中的点 P，我们要获取成像的点，可以将该点与 O 连线，连线所形成的射线 OP 与该投影面有一个角点，则该交点就是成像的点[11]。空间有一条直线 L，可以投影为圆球上的一个圆 g 以及平面上都可以找到与之对应的一点。在平面投影和球面透视投影两种投影下，一种是形成的一个平面，而后者的投影结果是具有中心点的一个球面。如图 2.6，图 2.7 所示，投影 $p(x,y,z)$ 显示在图像上为 $m(u,v)$。

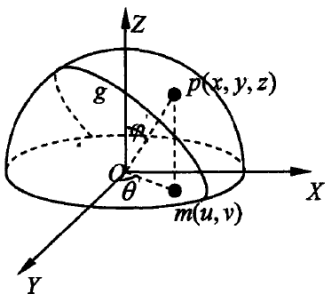


图 2.6 空间中的点投射到投影面

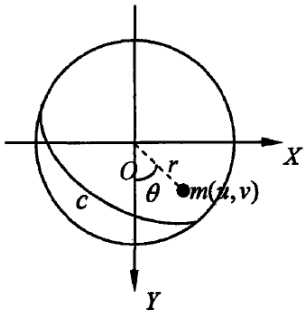


图 2.7 球面上的点垂直投射到投影面

中国知网 https://www.cnki.net

设 $p(x,y,z)$ 是球的面上任意的一个点，R 为该面到中心的长度，即球的半径的大小。该点在在 XOZ 坐标面上的坐标值变成 $m(u,v)$，如图 2.8：

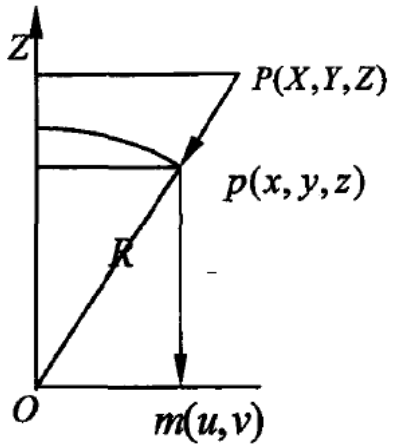


图 2.8 鱼眼图像的球面变换模型

则有下面的变换关系：

$$\frac{x}{u}=\frac{y}{v}=\frac{\sqrt{x^2+y^2}}{\sqrt{u^2+v^2}}=\frac{\sqrt{x^2+y^2+z^2}}{R}$$

$$x=\frac{zu}{\sqrt{R^2-u^2-v^2}},y=\frac{zv}{\sqrt{R^2-u^2-v^2}} \quad \text{公式（2.1）}$$

$$u=\frac{Rx}{\sqrt{x^2+y^2+z^2}},v=\frac{Ry}{\sqrt{x^2+y^2+z^2}}$$

选取空间中的一点 (x,y,z)，并把选取的该点当作是透视投影形成的平面 $z=z_0$ 上的点，然后再执行下一步的球面透视的投影。先考虑平面 $z=z_0$ 上平行于 x 轴的直线 $y=y_0$，有

$$u=\frac{Rx}{\sqrt{x^2+y_0^2+z_0^2}},v=\frac{Ry_0}{\sqrt{x^2+y_0^2+z_0^2}}$$

$$u^2=\frac{R^2x^2}{x^2+y_0^2+z_0^2}=R^2-\frac{R^2y_0^2}{x^2+y_0^2+z_0^2}\bullet\frac{y_0^2+z_0^2}{y_0^2}=R^2-v^2\frac{y_0^2+z_0^2}{y_0^2} \quad \text{公式（2.2）}$$

$$\frac{u^2}{R^2}+\frac{v^2(y_0^2+z_0^2)}{R^2y_0^2}=1$$

公式（2.3）表示的是一个椭圆的通式。它表示一个半长轴为 R，半短轴为

中国知网 https://www.cnki.net

$\frac{R|y_0|}{\sqrt{{y_0}^2+{z_0}^2}}$，中心坐标的原点的一个圆[12]。对于平面 z_0 上的直线进行转动，该直线的特点是不与 x 轴平行，我们把该直线绕着 z 轴进行转动，转到一个合适的位置，该位置总可以使直线与 x 轴平行。说明存在于空间中的一个点 (x,y,z)，把该点绕着不用于 x， y 的 z 轴进行转动。那么同理可得，该点的在球面上所形成的投影的旋转角度，和该点的旋转角度是相同的。

设椭圆的方程为:

$$Au^2+2Buv+Cv^2+Du+Ev+1=0 \qquad \text{公式（2.3）}$$

其中 $B^2<AC, A, C>0$,参数归一化后 $F=1$。

根据上面的等式的变形，我们可以得到椭圆的圆心的坐标，设为该点为 (u_0,v_0) 该方程的半长轴的值为 a：

$$u_0=\frac{CD-BE}{2(B^2-AC)}, v_0=\frac{AE-BD}{2(B^2-AC)} \qquad \text{公式（2.4）}$$

$$a=\sqrt{\left(\frac{CD^2-2BE+AE^2}{4(B^2-AC)}+F\right)\frac{A+C+\sqrt{(A-C)^2+4B^2}}{2(B^2-AC)}} \qquad \text{公式（2.5）}$$

计算半长轴 R 以及圆心的坐标值的大小的步骤如下：

1)先在鱼眼图像中找到一条在正常情况下为直线的变形曲线，该直线在鱼眼图像中所呈现的形态是一条变形的弧。我们在这条弧上找到不同位置的 10 个点，并记下它们所有的坐标值 (x_i,y_i)。

2)采用最小二乘法拟合这些点所在椭圆方程

$$Ax_i^2+2Bx_iy_i+Cy_i^2+Dx_i+Ey_i+1=0 \qquad \text{公式（2.6）}$$

的系数 A， B， C， D， E。

3)得出上面椭圆方程系数 A， B， C， D， E，再由公式（2.5）和（2.6）计算出图像的中心坐标值 (u_0,v_0)、半长轴的大小 R。

4)由公式（2.1）和（2.2），由空间的一点 (x,y,z) 以及图像上 (u,v) 映射关系可以将鱼眼图像转化为透视的图像[13]。

图 2.9 是对鱼眼图像进行标定获取参数的过程，采用一块有固定间距图案的棋盘标定板来进行鱼眼摄像头的标定。图 2.9 为本文在实验中使用到的标定板。

中国知网 https://www.cnki.net

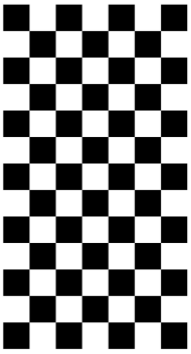
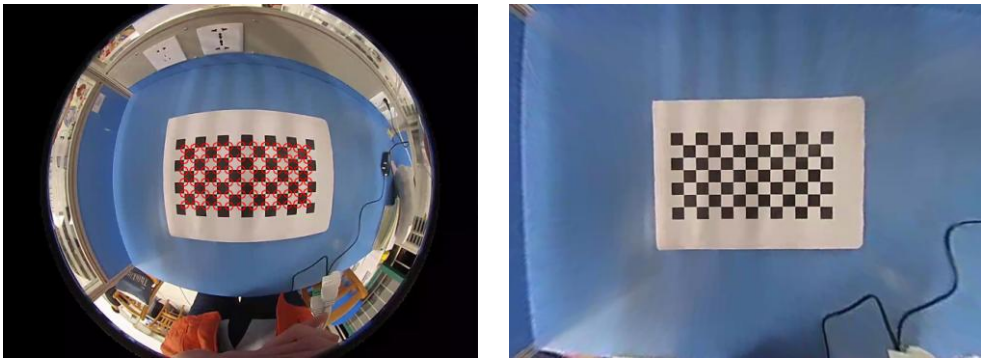


图 2.9 棋盘标定板样式

如图 2.10（a）是一个鱼眼的标定图像，大小为 400*600。在一条代表直线的弧上进行椭圆的拟合，然后计算出拟合后的结果，包括圆心的坐标以及椭圆的半径。多次实验以后，把最佳效果时的参数作为校正的参数，得到的中心点为（144.54,205.40），半长轴为 R=197.92，实验过程中使用的鱼眼摄像头的型号为海康威视 DS-2CD3942F-I，鱼眼图像的大小为 400*600，获取的鱼眼图像的像素为 400W 像素。



(a)棋盘标定图像　　(b)棋盘校正结果

图 2.10 椭圆拟合校正结果

2.4 基于经度法的校正

将经度图引进鱼眼图像中。如图 2.11 所示：

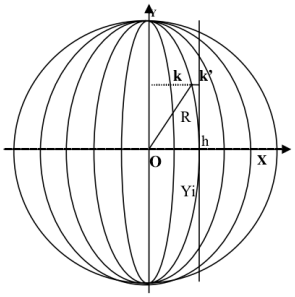


图 2.11 鱼眼图像校正经度图

中国知网 https://www.cnki.net

在确定的一条经度的椭圆上，方程:

$$\frac{y^2}{a^2}+\frac{x^2}{b^2}=1 \qquad \text{公式（2.7）}$$

式中 a 为长轴的半径。$b=1,2,3,\ldots,H$，H 是鱼眼图像宽的半径。当 $Y=Y_k$ 由上述方程可知：

$$x_k=b\cdot\sqrt{1-y_k{}^2/a^2} \qquad \text{公式（2.8）}$$

k 点坐标 (x_k,y_k)，校正后 k' 点坐标 (b,y_k)，h 点坐标 (x_h,y_h)，校正后的 h' 点 (b,y_h)。点 $P(x,y)$ 是为原始图上的任意选取的某一个点，设与之相应的经度算法计算后点的坐标值的大小为 $P1(x_1,y)$，则有：

$$\frac{x}{x_1}=\sqrt{1-y^2/a^2} \qquad \text{公式（2.9）}$$

即

$$x_1=\frac{1}{\sqrt{1-y^2/a^2}}\cdot x \qquad \text{公式（2.10）}$$

在进行校正算法以后，上图中的点 k 和 h 有着相同的横、纵坐标[14]。可以对同一条经线上的像素点的值进行确定，即这些像素点的值得纵坐标与原来的保持一致，不发生变化，但 x 坐标的值是相等的。

2.5 实验结果与分析

本章对鱼眼图像校正的方法进行了分析，由于各种校正方法的效果不一样，因此根据实验的要求，选择不同的方法来进行。利用上面标定后得到的鱼眼图像的校正参数，分别对鱼眼图像使用球面透视投影和经度法进行校正的对比实验。

本文在鱼眼图像的校正实验以及拼接实验中，所选取的实验图像的尺寸大小均为400*600。因为这些图像由同一鱼眼镜头采集，所以在对畸变的图像进行校正时，所使用的校正参数是相同的。实验结果如下：

中国知网 https://www.cnki.net

图 2.12 球面透视投影约束校正

图 2.12 可以看出，该校正方法基本还原出原来景物的原貌，弯曲的部分可以很好地还原成直线。但图像从球面到平面的时，由于投影的面大于成像面，所以不可以将鱼眼图像整个的信息都完全被容纳到平面里面，因此就会导致最后显示的图像的细节要少于原始图像。

图 2.13 右图表示的是使用经度坐标定位法校正后的图像。

图 2.13 经度校正法

图 2.13 中可以看出，图像中垂直方向上已经校正的很好了，能在一定程度上解决景象弯曲的问题，右侧弯曲的教学楼已经接近直线，基于经度校正的方法是可行的。

由实验图像可看出，球面透视投影方法利用前文经过标定后的鱼眼参数对上面图像进行畸变的校正。但这种投影的方法往往会丢失部分的边缘细节信息。使用经度的方法对图像进行校正以后，图像的上下边缘会有较大的形变没被还原，算法的好处是不会造成图像的缺失，图像的细节内容和原图像保持一致。根据实验的结果分析，本文选取在校正过程中信息保持完整的经度法来进行鱼眼的校正。

中国知网 https://www.cnki.net

2.6 本章小结

本章对常用的鱼眼图像的校正方法进行了分析，并根据实验的结果对比了两种不同的鱼眼图像的校正方法：第一种是基于球面透视投影的校正方法。第二种是基于经度坐标定位的校正方法。从实验中我们可以看出，基于球面透视投影约束法可以准确地计算出图像参数，对图像的形变也有很好的还原，但该方法会造成边缘信息的缺失，使校正后图像的视场变小；虽然基于经度坐标定位的校正方法在图像的下边缘产生了少量形变，但是整体而言，该算法在校正的过程中没有像素的丢失，图像信息保存完整。因此，本文选取基于经度坐标定位的校正方法来进行鱼眼图像的校正。

中国知网 https://www.cnki.net

第 3 章 图像的特征提取与匹配

特征提取是图像拼接的核心部分，当图像的分辨率比较高时，那么表示这张图像所用到的矩阵的维数就会很大，因此产生运算量是很大，需要考虑使用降维度的方法来减少矩阵的运算，并把能代表整体矩阵特点的矩阵提取出来，作为特征来进行描述。本章首先介绍了常用的局部特征算法，并对广泛使用的特征提取算法进行了分析，最后通过实验对这些算法的性能进行了比较。实验证明，改进 ORB 与 PROSAC 算法的结合能有效的缩短特征匹配的时间和匹配的准确性。

3.1 图像局部特征

图像局部特征主要包括：边缘、线、角点。其中被广泛应用的是对于图像中的角点进行检测，许多研究者们都致力于对角点的研究。在图像中的一块区域中，先取得该区域的灰度值图像，然后使用一定的算法对该区域上灰度值的变化情况进行识别，若在区域的某个地方，灰度值的变化非常的大，则该区域的这个地方就称为角点或者边缘 [15]。检测角点的方法可以应用到很多的领域，比如在视频中对特定物体的追踪，识别其中是否有特定的物体。

特征提取有两个过程：第一，特征检测：对图像的局部进行检测，检测到特征点以后对其位置进行定位；第二特征描述：使用一种方法，例如梯度直方图统计方向等方法来生成特征的描述子。

3.2 特征点检测算法

常用的特征点检测的算法有：(1)角点检测：如 Kitchen-Rosenfeld、Harris、SUSAN 和 FAST 算法等；(2)按邻域性质来分的检测算法：如 SIFT，改进的 SURF 算法等。

3.2.1 Harris 角点检测

Harris 把灰度方差使用 Taylor 公式展开，可以得到算法的相关系数矩阵，如下：

中国知网 https://www.cnki.net

$$V=\sum_{u,v}\left[I(x+u,y+v)-I(x,y)\right]^2$$
$$=\sum_{u,v}\left[uI_x+vI_y+o(u^2,v^2)\right]\approx\sum_{u,v}(uI_x+vI_y)^2 \qquad \text{公式（3.1）}$$
$$=\sum_{u,v}(u^2I_x{}^2+2uvI_xI_y+v^2I_y{}^2)=\sum_{u,v}((u,v)\begin{bmatrix}I_x{}^2 & I_xI_y\\ I_xI_y & I_y{}^2\end{bmatrix}(u,v)^T)$$

$I(x,y)$ 为把图像进行灰度处理以后，横纵坐标为 x 和 y 的点处的灰度值，$I_x=\frac{\partial f(x,y)}{\partial x}, I_y=\frac{\partial f(x,y)}{\partial y}$ 指的是分别对 x 和对 y 求偏导后的表达式。

在 Harris 角点检测的过程中，使用以下矩阵的表达式来替换上式的 Hessian 矩阵进行更简便的计算：

$$M(x,y)=w_{u,v}\otimes\begin{bmatrix}I_x{}^2 & I_xI_y\\ I_xI_y & I_y{}^2\end{bmatrix} \qquad \text{公式（3.2）}$$

$w_{u,v}=\frac{1}{2\pi\sigma^2}\exp(-\frac{(u^2+v^2)}{2\sigma})$ 是高斯窗口函数，它用来对将要处理的区域进行滤波。其中替换矩阵为 Hessian 矩阵的 $M(x,y)$，$4.5*\sigma$，λ_2 是它的两个特征值，可以根据特征值的大小把一幅图像划分为三个不同的区域，如图 3.1 所示：

(1) 若 λ_1,λ_2 较小，则代表普通区域，即 x,y 两坐标轴上的值变化比较缓慢。

(2)若 $\lambda_1\gg\lambda_2$ 或者 $\lambda_1\ll\lambda_2$,表示处在边缘区域上，两坐标轴方向上灰度变化情况为：其中的一个值改变大，而在另外一个的改变缓慢。

(3) 若 λ_1,λ_2 两者取很大值时，则此时代表检测的位置为角点。

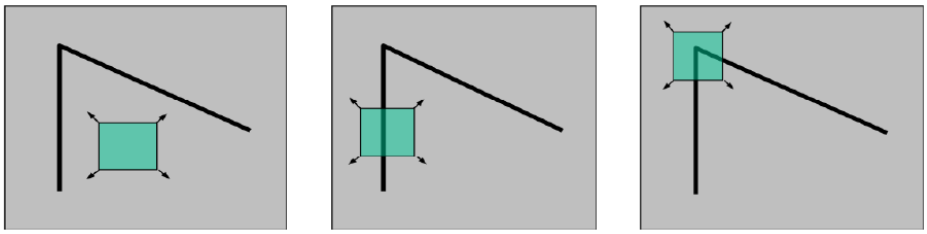


图 3.1 上图分别为：非角点、检测到的为边缘、检测到角点的存在

定义一种角点变化的值，以减小算法的计算量：

$$R=\det(M(x,y))-trace(M(x,y))^2=\lambda_1\lambda_2-k(\lambda_1+\lambda_2)^2 \qquad \text{公式（3.3）}$$

中国知网 https://www.cnki.net

其中敏感因子为k，当求得的角点响应值R大于设定的值时，那么该点就符合选取为角点的条件，就可以被称作角点。

3.2.2 SIFT 特征点检测

SIFT 在图像发生旋转时，也能检测出很多的特征点，具有很强的抵抗旋转特性。

SIFT 生成特征点的过程如下：

1.生成图像的多尺度空间

Gaussian 卷积核的表达形式为：

$$G(x,y)=\frac{1}{2\pi\sigma^2}\exp(-(x^2+y^2)/2\sigma^2) \qquad \text{公式（3.4）}$$

其中σ为高斯尺度因子。当高斯尺度因子σ的值越小时，经过变换后的图像细节就会越分明、丰富，得到的图像内部信息也就越多；相反，σ值越大时，图像就会越模糊[10]。尺度空间的定义如下：

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y) \qquad \text{公式（3.5）}$$

$I(x,y)$是整幅图像上各个像素点的灰度值的表达式，即代表的是(x,y)点处值的大小。为了简化运算，在 LOG 的基础上，Lowe 等提出了 DoG(Difference of Gaussian)：

$$D(x,y,\sigma)=(G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y) \qquad \text{公式（3.6）}$$

在相邻的两层尺度空间的生成上，我们使用尺度因子进行每一层的尺度空间大小的控制，其中k的值代表的是层与层之间两者的大小的比例。生成高斯金字塔和生成差分金字塔的过程如下：

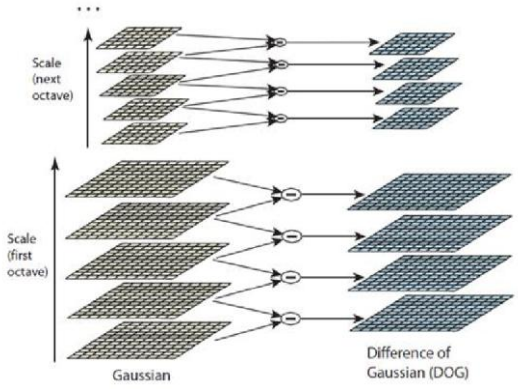


图 3.2　LOG 金字塔与 DOG 金字塔

中国知网　https://www.cnki.net

如图 3.2 所示我们把高斯金字塔以分组的形式呈现，在这里处理后被分为o组。每组又分层，分层的数量为s。Lowe 的取值为$o=4$，$s=5$。σ_o^s表示的是尺度因子的值，上下标分别代表的是该值所在的位置，即第o组的第s层。在生成的金字塔中，我们分为多组，每组里面我们又分为多层，各层之间的关系为成一个比例，该比例σ_o^s的比值是k，此时k的取值为$k=2^{\frac{1}{s}}$，在相邻的两组中，每组中都有相同的层数，在每组相同层号之间的比例σ_o^s比例为 2。在生成第二层的尺度空间时，该层的图像的尺寸为第一层图片大小的一半，第三层为第二层的一般，以此类推，则可以推算出当组数的大小为o时，可以根据上面的组数求得o组的高斯尺度通式为：

$$2^{o-1}(\sigma,k\sigma,k^2\sigma,...,k^{s-1}\sigma) \qquad \text{公式（3.7）}$$

2.使用高斯差分进行局部的极值点检测

在尺度不变特征变换算法 SIFT 中，使用 DOG 对图像进行局部的极值进行检测，把生成的极值组成一个集合，则图像中确定出的特征点就为该集合的一个子集。要确定某个点是否为该区域的极值点，不但需要将该点与同层的点进行比较，而且还需要将该点与相邻的上下每层对应的周围像素进行对比，比较的方法为：在同一层中，比较以该点为中心形成方形区域，该区域的边长为 3 个像素点，则就需要与较该点周围共 8 个像素进行比较；在上下两层中，先找到该中心点的对应点，然后查找以该点为中心的方形区域，该方形区域的边长大小也为 3 个像素，，然后拿中间层的中心与上下两层方形区域的点进行比较，此时需要比较共 18 个点，因此需要比较中间层中心周围的 26 个像素点，从而确定该点是否为极值点。通过比较获得极值点的过程如下图所示：

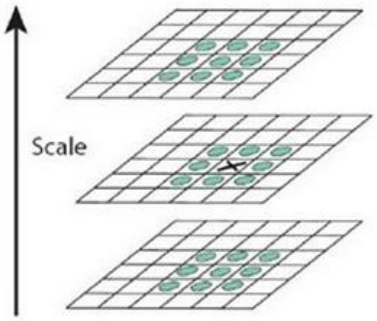


图 3.3 通过 DOG 获得局部的极值点

3.获取检测极值点的位置信息

通过以上的比较过程获得的极值点，往往分布在图像的不同区域，即呈散状分布，因此在获得一个极值点的位置等信息以后，无法直接取得下一个极值点的值，因此需要使用一个函数来求取这些散列分布的极值点。在求取这些分散极值点的过程中，我们还可以除去一些对比性比较弱的点，如：灰度值不是很具有区分性的点、响应值比

中国知网 https://www.cnki.net

较弱的点等，通过该过程同时也除去一些不必要参与计算的点，进而减少计算量。DOG 函数泰勒展开式：

$$D(X)=D+\frac{\partial D^T}{\partial X}+\frac{1}{2}X^T\frac{\partial^2 D}{\partial X^2}X \qquad \text{公式（3.8）}$$

求导，令$D(X)=0$，得：

$$\hat{X}=-\frac{\partial D^T}{\partial X}(\frac{\partial^2 D}{\partial X^2})^{-1} \qquad \text{公式（3.9）}$$

将公式（3.9）带入（3.8），得

$$D(\hat{X})=D+\frac{1}{2}\frac{\partial D^T}{\partial X}X \qquad \text{公式（3.10）}$$

当有$|D(\hat{X})|\le 0.03$时，如果不满足该条件，那么丢弃这个点。

4.除去 DOG 的边缘的响应点

在使用 DOG 求出区域内的极值点以后，先根据极值点求出该点曲率的大小，该曲率的计算方法为使用 Hassian 矩阵来求出该值的大小：

$$H=\begin{bmatrix}D_{xx} & D_{xy}\\ D_{xy} & D_{yy}\end{bmatrix} \qquad \text{公式（3.11）}$$

其中$\alpha=\gamma\beta$，α，β为公式（3.11）两个特征值的大小，γ为常数，那么可以得到以下的关系：

$$\begin{aligned}&\mathrm{tr}(H)=D_{xx}+D_{yy}=\alpha+\beta\\ &\det(H)=D_{xx}D_{yy}-(D_{xy})^2=\alpha\beta,\alpha=\gamma\beta\\ &\frac{tr(H)^2}{Det(H)}=\frac{(\alpha+\beta)^2}{\alpha\beta}=\frac{(\gamma\beta+\beta)^2}{\gamma\beta^2}=\frac{(\gamma+1)^2}{\gamma}\end{aligned} \qquad \text{公式（3.12）}$$

如果满足条件$\frac{tr(H)^2}{Det(H)}>\frac{(\gamma+1)^2}{\gamma}$，那么就可以对极值点进行筛选，此时我们需要去掉边缘响应函数值大于设定值的极值点。

5.求出特征点的主方向

(1)求出特征点的灰度值

中国知网 https://www.cnki.net

使用下面的公式来获取该点的梯度值：

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y)+(L(x,y+1)-L(x,y-1))^2}$$
$$\theta(x,y)=\tan^{-1}(L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y)) \quad \text{公式（3.13）}$$

$L(x,y)$是选取的特征点在转换成灰度图像以后的值的大小。

(2)依据梯度方向生成图像的直方图

把该特征点作为一个圆的圆心，4.5*σ是该圆的半径的长度，以此来画圆，使用直方图来表示圆内点的方向。把一个点的一周平均分为 36 个扇形，每一个扇形的区域赋予一个方向，该方向代表的是这个扇形内的点的方向，然后把方向相同的扇形放到直方图的同一个柱状图内。统计扇形区域的方向后，生成的统计图如下：

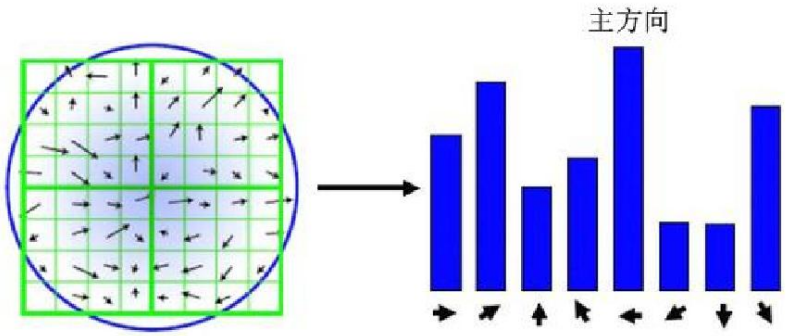


图 3.4 生成并选取特征点的主方向

由直方图可以的到该点的主方向，即柱状图中最大值即为该点的主方向，把达到主方向直方图值五分之四以上的另一个最大的直方图的梯度方向为辅方向。若没有次峰值，则该点没有辅方向。在获得了主方向以后，就实现了对 SIFT 特征点获取的整个过程。

3.2.3 SURF 特征点检测

SIFT 检测出的特征点的信息量丰富且彼此之间易于区分，缺点是大的信息量引起的高计算量，导致算法执行得很慢。基于以上 SIFT 算法的不足之处，SITF 算法的改进算法 SURF 被提出，该算法相对于 SIFT 具有较强的鲁棒性的特点。SURF 算法更加快速的主要原因是在求积分的过程中，SURF 使用的是积分图像的算法，并且使用加强版的变换矩阵来提高算法的性能，而在求取极值的上所不相同的是，SURF 使用 Harr 小波的方法来进行极值的求取。

1.积分获取灰度值

在积分求灰度值的过程中，SURF 为了使积分求灰度的过程更加的快速，所以应

中国知网 https://www.cnki.net

用积分图像的积分法来实现本步骤的运算。(x_I, y_I)表示的是位于积分图像中的某一个点[16]。$I(x_I, y_I)$代表的是一个范围内的灰度值的总和，该范围包括的是从图像的坐标零点到像素点所形成的方形区域内，包含的点的灰度值总和 $\sum_{i<x_I, j<y_I} L(x,y)$。积分区域如图 3.5：

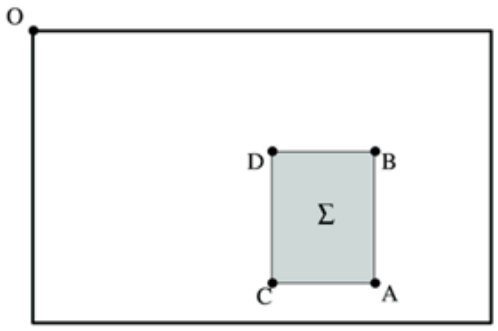


图 3.5 SURF 积分求取灰度值

灰色方框 S 所代表的积分的大小为 $S = A - B - C + D$ 来表示。

2.SURF 特征点检测

该算法是使用计算量更小的 Hessian-Laplace 来求取图像局部的最大值，然后选取这个最大值的点近似地表示所要表示的特征点[17]。

$$H(x,y,\sigma) = \begin{bmatrix} L_{xx}(x,y,\sigma) & L_{xy}(x,y,\sigma) \\ L_{xy}(x,y,\sigma) & L_{yy}(x,y,\sigma) \end{bmatrix} \qquad \text{公式（3.14）}$$

SURF 使用的是高斯差分算子 $D(x,y,\sigma)$ 来进行金字塔的生成。

对矩阵 Hessian 进行求解时，我们需要对图像进行前期的处理，处理方法是适用高斯平滑滤波来进行的，在这里使用的是其中的一种，即采用箱式滤波器(Box Filers)来处理分散的像素点的二阶导数的求取：

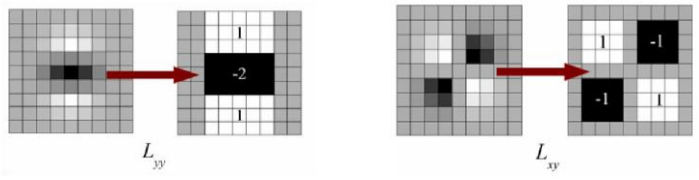


图 3.6 SURF 箱式滤波器

图 3.6 中左侧的结果为线性平滑滤波后的高斯滤波后生成的，其为只在 y 方向上进行求二阶导数运算的一个模板；右侧为坐标轴的两个方向上求二阶偏导的模板[18]。

中国知网 https://www.cnki.net

(1)构建尺度空间

SURF 建立尺度空间时需要进行尺度的变换，在这里采用的策略是调节箱式滤波器的大小来实现该算法的尺度变换，最终构建成 SURF 的尺度空间。该法的好处是没有 SIFT 中的降采样的步骤，降低了运算量。

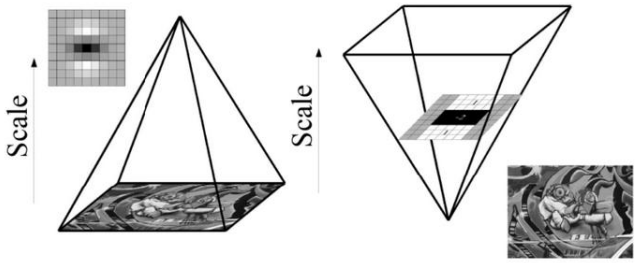


图 3.7 SIFT 和 SURF 生成尺度空间的对比

(2)求取 SURF 特征点的主方向

SURF 特征点主方向的统计方法与 SIFT 的统计方法是不相同的，SURF 所不一样的是在特征点主方向的选择方式上，SURF 选取的是 Harr 小波，然后再生成小波特征，最后统计小波的特征并作为该点的主方向 [19]。对于极值点的检测，我们要设定一个检测的范围，即检测区域。设一个圆以某一个特征点的所在的像素位置为中心，以6σ为圆的半径大小，作一个圆。然后我们对该圆所包含的像素点进行积分的运算，以 60 度的大小在圆上截取区域，该区域为扇形，然后使用 Harr 小波来生成并计算该截取区域内的响应值。最后把一定的弧度大小作为步长的值，对该扇形进行转动一周，每次转动的角度为一个扇形的大小，从而转动一周共获取了 6 个扇形区域的小波值，这样就获得了特征点一周的小波响应值，对获取的几个小波值进行比较，把最大小波值的扇形区域选定为主方向所在的区域，求扇形区域的中线的方向，则该方向代表该特征点的主方向，求取主方向的过程如下图所示：

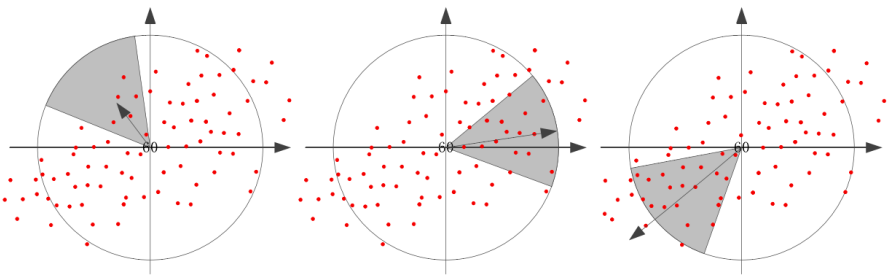


图 3.8 确定 SURF 主方向

中国知网 https://www.cnki.net

3.2.4 检测 FAST 角点

上文介绍了与自相关矩阵响应值相关的算子。在求得两幅图像的灰度值以后，对两者的灰度值进行求差值的运算，进而得到差值的平方。该过程中包括卷积的计算，所以会导致很大的计算量。

FAST 算法属于角点检测的范畴，适用于需要进行角点检测的应用领域[20]。该算法的定义为：在某像素点的周围选取一个邻域，该邻域为圆形区域，把该点与圆形区域内的点进行灰度值的比较，若符合比较的条件，则该点就可作为特征点进行选取。

FAST 算子的描述过程如下：

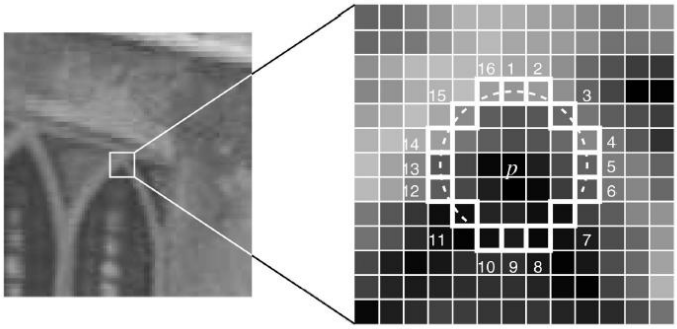


图 3.9 FAST 角点检测过程

图 3.9 中右侧图像，虚线的圆形区域内，P 为该圆形邻域的中心，它的邻域是以 r 为半径的一个圆，该圆的半径值使用像素个数来表示，其中 $r=3$。若要确定该圆心是不是为要求的角点时，我们可以求出该点处的灰度值以及该圆形邻域范围内点的灰度值的大小，然后分别与周围的点的值作差求绝对值，若该值大于设定值 ε_d [21]。可以使用一个函数来表示响应值的大小，此时的函数 N 如下：

$$N=\sum_{i=1}^{16}\left|I(i)-I(p)\right|>\varepsilon_d \qquad \text{公式（3.15）}$$

P 点所处的圆形区域上第 i 个点的值为 $I(i)$，经过实验的验证，当 N_0 的值为 9 或 12 能达到最好的效果。当 $N>N_0$ 时，则称 P 点为角点。

3.3 特征描述算子

特征描述算子是这样的一类向量，该类向量描述了和特征点有关的方向、大小尺度、位置信息等。而且存储特征点所需空间小、生成描述子的速度快，不但可以节约运行时的内存空间，而且还可以提升生成描述子的速度。但性能好的特征点描述算子，

中国知网 https://www.cnki.net

在存储空间上占用也相对较大，所耗费的时间也相对较大。

进行特征描述的算法有两类：第一类是根据梯度直方图来进行生成算子的方法，如上文的 SIFT 和 SURF 都属于该类的描述算子[22]。这些算子根据特征点获取过程中描述特征点信息的直方图，直接用来附加到生成的描述子上。第二种是基于生成的二进制串，然后对该生成的二进制串进行汉明距离的比较[23]。

3.3.1 基于梯度直方图的特征描述子

由上文的介绍可知，在描述特征点的过程中用到的直方图，主要是用来说明特征点及其邻域特征点的梯度方向的。该描述算子的典型代表是 SIFT 和 SURF。

1.SIFT 特征描述算子

在获取到组成特征点所需的位置、尺度、主方向等以后，然后根据以上的信息生成特征描述：

(1)如下图所示，原点代表一个像素点，红色的箭头代表的是该点的直方图描述的主方向，为了降低生成的描述子对旋转的敏感性，我们把坐标轴的 x 方向旋转到与该点的主方向重合的位置，这样描述出来的特征点便具备了抗旋转的特性[24]。

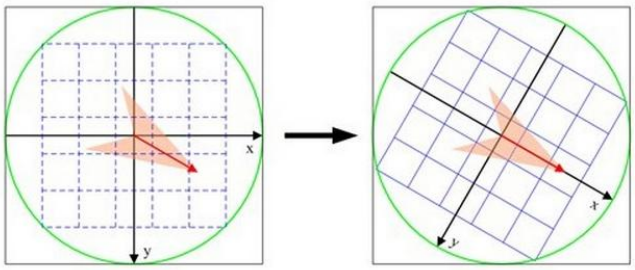


图 3.10 特征点主方向旋转示意图

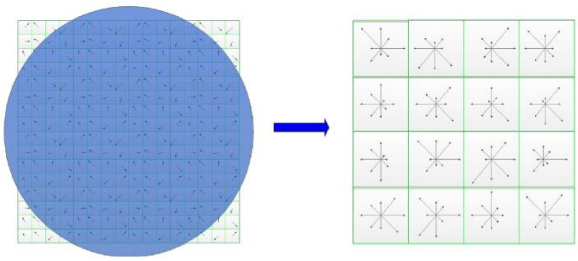


图 3.11 生成 SIFT 描述子

(2)在该特征点的周围选择一个方形的区域，该区域是以特征点为中心，边长为 16 个像素的区域为邻域，此时需要求方形区域内像素点的梯度方向。我们称这个区域的点为采样点，然后以每四个像素为单位，将大的区域划分为一共 16 个小的区域，通过高斯加权的方法，使每个小区域都具有 8 个不同方向，每个生成的方向都使用一个向

中国知网 https://www.cnki.net

量来代替，各个方向所代表的向量是不相同的，因此此时的向量维度为 8。如图 3.11，各个点上附加一个向量，向量的方向与该点的梯度方向是一致的 [25]。一个特征点的描述向量的维数越大，就说明它能拥有越多的附加信息，这就会使每个特征点就更具有区分性，在匹配的过程中也就降低误匹配情况的出现频率[26]。

2.生成 SURF 描述子

该算子进行描述的时和 SIFT 相近，同样在一个特征点的附近选取区域，该区域为方形并且对角线的中点与该特征点重合，该区域的边长为 20σ（σ 为尺度因子），旋转该特征点所在的坐标轴的方向，使 x 轴的正方向与之重合，这样就使特征点不受旋转的干扰，因而就具有了旋转不变性[27]。然后把这个区域分为边长为 4 的方形区域，则共可以分为 16 个小的子区域。然后在每个的子区域上分别统计两坐标轴上的 Harris 小波的特征。并分别求出 Harr 小波特征在 x 轴方向上特征值的和 $\sum dx$ 和该方向特征绝对值的和 $\sum|dx|$；垂直方向的和 $\sum dy$ 以及绝对值的和 $\sum|dy|$ [28]。

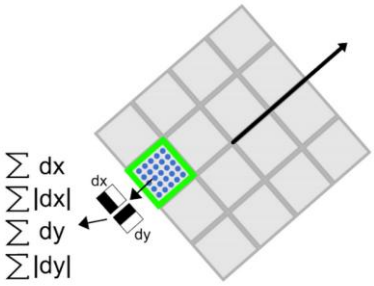


图 3.12 生成 SURF 算子的描述子

以上分析我们就可以得知，在划分的每个小区域里，我们得到了四个方向。则整个特征点的向量维度为 4*16 一共 64 维。由于获得维度更低的描述子会减少花费的时间，因此 SURF 生成描述子的速度要快于 SIFT，因此 SURF 在描述子的生成上，更有优势[29]。

3.3.2 基于比较的二进制特征描述算子

因用梯度直方图的方法进行描述的描述子具有多方面良好的性能。2010 年出现的根据二进制串生成特征描述子的算法，相对于 SURF，在速度又提升了一个数量级[30]。第一个出现的二进制特征描述子是 BRIEF，之后出现新的基于二进制串的描述子。BRIEF 的采样模板是由许多有公共圆心的圆的点，而 FREAK 的是依据视神经细胞的原理，模仿而成的模板。下面是生成 BRIEF 描述子的过程：

首先我们先在图像上选择一部分，该部分为 P，在 P 范围内选取一个测试的点组成的集合，若该集合为 τ，那么：

中国知网 https://www.cnki.net

$$\tau(p;x,y):=\begin{cases}1 & if\ \ p(x)<p(y)\\ 0 & otherwise\end{cases} \qquad \text{公式（3.16）}$$

选出用来测试的集合$n_d\left(x,y\right)$。该集合里面的点对有着确定的位置关系。在进行特征点的配对时，先求出描述子之间的 Hamming 距离，然后把该距离与阈值相比较，当距离的大小小于我们所规定的阈值的范围时，我们可称该点对为匹配的点对，若不满足阈值的条件则不是匹配点对[31]。BRIEF 描述子的构成是用n_d维串来表示：

$$f_{n_d}(p)=\sum_{1\le i\le n_d}2^{i-1}\tau(p;x_i,y_i) \qquad \text{公式（3.17）}$$

BRIEF 的取值为 128、 256 或 512。由于表示描述子二进制串越长，表达信息的能力就越大，因此我们可以根据实际情况的需要，选择合适的二进制串进行描述，越短的串，其生成描述子时需要的时间，存储空间等资源也越少。这三种的描述子的名称为 BRIEF-16、BRIEF-32、BRIEF-64，其中的数字 16、3、64 分别表示该描述子的位数[32]。而 SURF 的每一个描述子所占的内存空间为 256 字节，而一个 BRIEF 的描述子占用空间最大的也只占 SURF 的 1/4 的字节数。

BRIEF 算子的核心部分是点对集$n_d\left(x,y\right)$的生成，设点对$\left(x,y\right)$服从高斯分布：

$$(x,y)\sim Gaussian\left(0,\frac{1}{25}S^2\right) \qquad \text{公式（3.18）}$$

其中σ表示方差。当σ取值为$\frac{1}{5}S$时，BRIEF 的性能最优。同时，该算子也有很强的抗视角变化以及光照等的鲁棒性。但是该算法也有一定的缺点，它并不具备抗旋转的特性[33]。图 3.13 为边长为 32 的方形区域内，当维数为 512 时，匹配特征点的布局图：

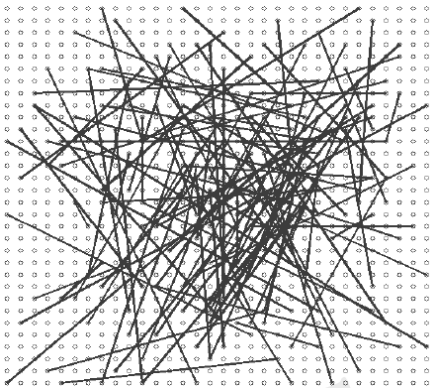


图 3.13 64 位的 BRIEF 特征点对分布情况

中国知网 https://www.cnki.net

3.4 ORB 提取特征和生成描述子

ORB 是由两部分的算法融合而成的，第一部分是提取特征点，其采用的是改进的 FAST 算法；第二部分是结合了特征描述的算法 R-BRIEF[34]。FAST 本身缺少方向的特性，在 ORB 中，采用灰度质心法给使其具有了方向的特性，同时使用 BRIEF 描述子对提取的特征点进行描述。

3.4.1 O-FAST 角点检测

ORB 算法中，FAST 算子的阈值 $N_0=9$，即使用的算子为 FAST-9。在设定的 $N_0=9$ 下，可以获得较多的特征点。由于 FAST 并没有提供描述角点性质的解决办法，为此使用 Harris 来进行排序：先减小阈值的大小，以保证可以得到 N 个以上的点；其次我们要按照一定的规则对它们进行排序，Harris 算法可以实现这个排序的过程，然后从排序的结果中选取位于最前面的 N 个点[35]。但是，由于 FAST 并没有提供解决尺度变化的方法，因此借鉴其它的算法，通过建立图像金字塔的方式，使用 FAST 算子在金字塔的每一层进行特征的提取，进而可以形成 ORB 算法尺度空间。

oFAST 相对于最主要的改进就是描述的特征具有了方向性，从而使描述子具有了抗旋转的特性[36]。其中使用的是 Rosin 的灰度质心法[37]。

矩定义为：

$$m_{pq}=\sum_{x,y}x^p y^q I(x,y) \qquad \text{公式（3.19）}$$

依据公式（3.19）计算该区域的区域质心坐标为：

$$C=\left(\frac{m_{10}}{m_{00}},\frac{m_{01}}{m_{00}}\right) \qquad \text{公式（3.20）}$$

ORB 以点 O 为圆心，以 3 个单位的像素作为半径的圆形范围为 FAST 关键点的邻域，即 x,y 的取值属于 $[-3,3]$。连接圆心得到 $\overrightarrow{OC}$。把 $\overrightarrow{OC}$ 的方向 θ 当作 FAST 获取特征点的方向：

$$\theta=a\tan 2(m_{01},m_{10}) \qquad \text{公式（3.21）}$$

$atan2$ 是值的范围在 $(-\pi,\pi]$ 的函数[38]。

中国知网 https://www.cnki.net

3.4.2 R-BRIEF 生成描述子

ORB 算法使用 R-BRIEF 生成描述子。因为 BRIEF 算法并不具备抗旋转的特性，ORB 中改进的 R-BRIEF 使生成的描述子具有了旋转不变的特性。

ORB 选取 32 位的 BRIEF 算子，即生成 256bits 的描述子。我们先对图像进行预处理，以便后面使图像符合要求，在这里是对图像进行平滑处理，使用的算法为积分图像法[39]。为了提高 ORB 算法对于方向的鲁棒性，需要使该算法具有抗旋转的特性，为了便于与旋转矩阵进行运算，我们需要把描述子的点集转用矩阵的形式来进行表示。首先需要将点集转化成矩阵表示。给定一个有 n 字节的测试点的集合 (x_i, y_i)，经过矩阵的转化后，我们可以获取一个 $2\times n$ 的矩阵：

$$S=\begin{pmatrix} x_1 x_2 \dots x_n \\ y_1 y_2 \dots y_n \end{pmatrix} \qquad \text{公式（3.22）}$$

由特征点检测时获得的方向角 θ，然后结合方向角并根据矩阵 S，求出旋转矩阵为 R_θ。这样就生成了具有方向的点的集合 $S_\theta = R_\theta S$。在 ORB 算子中的名称为 steered BRIEF 则：

$$g_n(p,\theta) := f_{n_d}(p) \mid (x_i, y_i) \in S_\theta \qquad \text{公式（3.23）}$$

由上文可知 steered BRIEF 具备了方向特性，因此描述子就具有了相对大的方差，从而用来进行特征点描述的二进制串的均值大约为 0.5。与 BRIEF 选择测试点集的大小不一样，ORB 的采用的点集是边长为31×31方形块。图像块上的子窗口为边长 5 个像素的方形块，使用子窗口函数的作用就是减少了单个的周围像素对中心像素的值的影响，从而使得到的像素的值更加的精确[40]。

如图 3.14 所示，边长是 31 个像素的飞方形块，绿色的子窗口在这个图像块中移动，这样就有 $N=(Wp-Wt)2$ 个候选子窗口。从这些子窗口中任选两个作为 BRIEF 的点对，由于在边缘处会产生重复计算的子窗口，因此需要把这些重复的去掉。

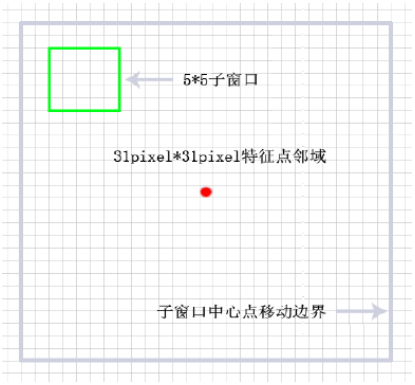


图 3.14 R-BRIEF 算子进行采样

中国知网 https://www.cnki.net

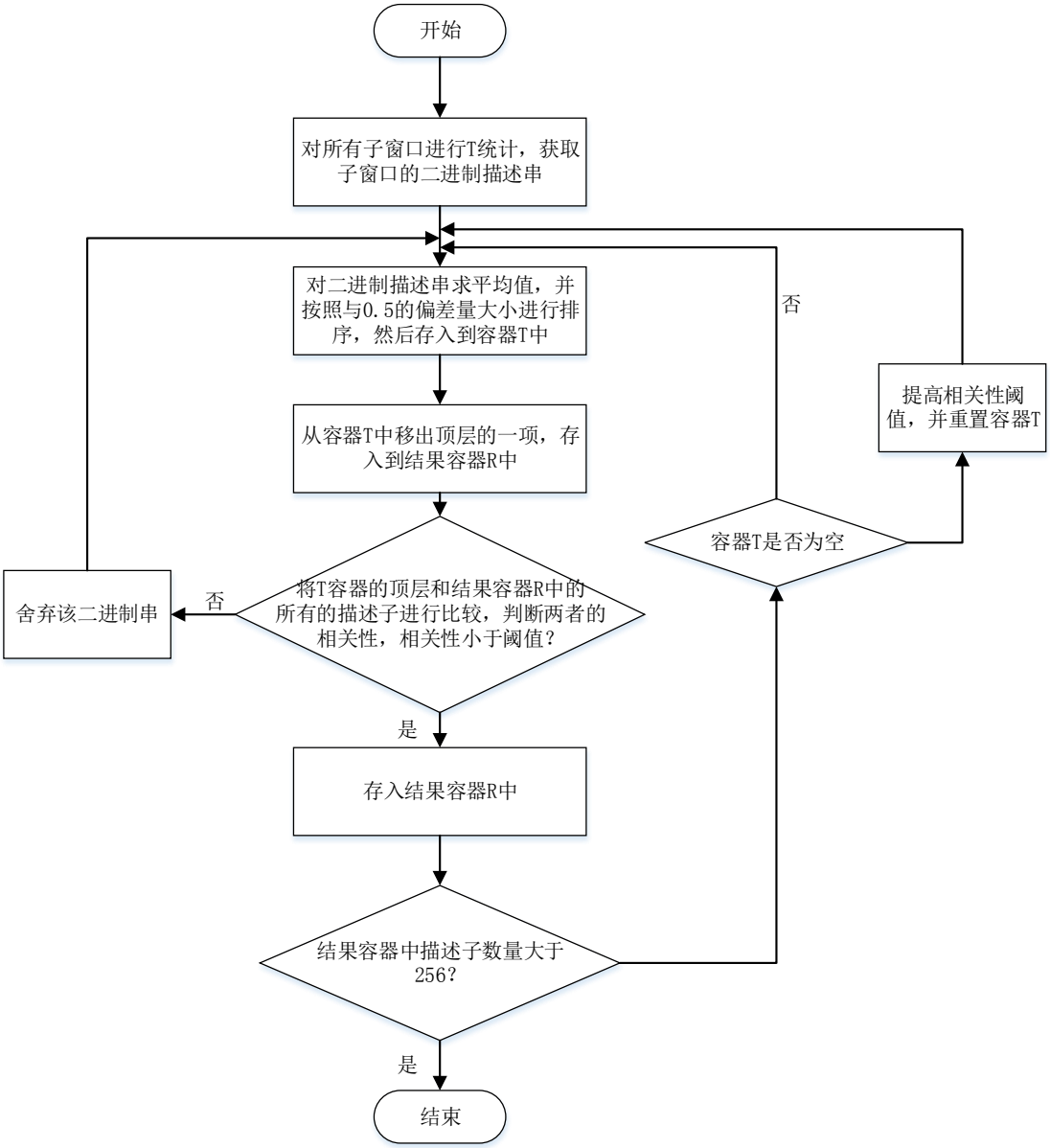


图 3.15 生成 R-BRIEF 描述子

ORB 从采样集合中选出 256 个点，具体如下：

（1）根据上文的描述，使用 o-FAST 算子对区域内的子窗口的数量进行检测，设检测到的数量为τ。

（2）记录由 o-FAST 算法得到的二进制串，为了方便比较，先求出生成字符串的平均值，然后和 0.5 进行比较，统计每个字符串与 0.5 差量的值。最后按照均值与 0.5 偏差值的大小进行排列顺序，然后把得到的这些二级制串放到T中。

（3）进行贪婪算法的搜索：

（a）开始取出容器T中存放的二级制串，由于该容器的特点是从顶部取出最后放入的数据，因此我们把取出来的第一项转存到存放结果的容器R中。

（b）然后我们要对两个容器中的二进制串的相关性进行比较，我们继续拿出存

中国知网 https://www.cnki.net

放二进制串的容器顶部的串，与结果容器的串进行相关性的比较，最后将两者相关性值的大小与设定的阈值进行对比，若是符合设定阈值的条件，那我们就保留这个二进制串到结果容器中，否则舍弃该二进制串，继续进行下个串的比较：

（c）重复（a）（b），直到 R 中有 256 个二进制字符串。若算法结束后字符串的个数后仍小于 256，则把阈值增大，直到个数达到 256 个。

经过筛选后的二进制描述子，其二进制串的平均值满足一定的范围，才能有比较大的方差出现，这个比较的二进制的均值一般为 0.5，这样做的目的就是为了让描述子之间具有更大的区分性[41]。

上述的过程完成了 ORB 的特征点的生成和描述，在进行配对时，该算法采用的是 LSH(Locality Sensitive Hashing)法，比 SIFT 和 SURF 采取的 KD-Trees 要快。

3.5 改进的 ORB 算法

传统 ORB 并没有尺度的不变性，其原因是使用的 FAST 得到的特征点没有尺度不变的特性。如（3.2.3）章节所介绍，借助 SURF 的方法，利用不一样尺寸大小的盒状滤波模板和积分图像生成的多尺度空间来进行稳定极值点的检测，从而进一步确定特征点的方向等信息，使改进的 ORB 算法具有了尺度不变性，生成具有尺度不变性的特征点。并用 ORB 对这些特征点描述，获取旋转不变的二进制描述子，改进算法不但保留了传统算法速度快的优点，还解决了其传统算法没有尺度不变性的缺点，提高了匹配的准确度[42]。

首先使用 Intensity Centroid 来求取特征点的主方向；然后使用 ORB 生成具有尺度不变性和旋转不变性的描述子，这样不但解决了 ORB 算法不具备尺度不变性的缺陷，又保留了其在速度和抗旋转的性能。

给定图像 I 中的某一点 $x=(x,y)$，在该点 x 处，尺度为 σ 的 *Hessian* 矩阵定义为：

$$H(x,\sigma)=\begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \qquad \text{公式（3.24）}$$

其中，$L_{xx}(x,\sigma)$ 是高斯二阶微分 $\frac{\partial^2 g(\sigma)}{\partial x^2}$ 在点 $x=(x,y)$ 处于图像 I 的卷积，$L_{xy}(x,\sigma)$ 和 $L_{yy}(x,\sigma)$ 有类似的含义。

高斯函数虽然是最佳的尺度空间分析工具，但在实际应用时总要对高斯函数进行离散化和剪裁的处理，从而导致损失了一些特性。这就为我们使用其他工具代替高斯函数对尺度空间进行分析提供了可能，应为既然高斯函数会带来误差，那么其他工具所带来的误差也是可以被忽略的，只要产生的误差足够小就可以。

中国知网 https://www.cnki.net

Bay 等人提出了盒状滤波来近似高斯滤波，章节 3.2.3 中的图 3.6 所示：x，y 和 xy 方向的加权盒状滤波器近似用 D_{xx}、D_{xy} 和 D_{yy} 表示。

其中 D_{xx} 和 D_{yy} 中的白色部分的像素的值为 1，黑色部分为-2；D_{xy} 中的白色部分为 1，黑色部分为-1；其它部分区域权值为 0，然后利用积分图，盒状滤波器的运算完全不取决于它的尺寸大小，这是 *SURF* 算法快速的一个重要原因。

用盒状滤波器近似代替高斯滤波器得到的 *Hessian* 矩阵行列式比较精确的估计式子为：

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \qquad \text{公式（3.25）}$$

为使特征点具有尺度不变性，首先去掉行列式值低的像素，然后进行尺度空间和邻域空间的非极大值抑制，最后所得的极值点即为特征点。

为增强特征点匹配时的稳定性和抗噪能力，首先使用泰勒插值滤掉低对比度的极值点，然后通过主曲率比值去掉边缘上的极值点。

由于 *Hessian* 矩阵的特征值和极值点 B 的主曲率对应成正比，所以只需关心特征值的比值。设 γ 是最大特征值 α、最小特征值 β 的比值，即 $\alpha = \gamma\beta$。则可求得：

$$\frac{Trace(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(\gamma+1)^2}{\gamma} \qquad \text{公式（3.26）}$$

上式的右边随着 γ 的增加而增加，所以要想检查主曲率的比值小于某一阈值 γ，只要检查下式是否成立即可：

$$\frac{Trace(H)^2}{Det(H)} < \frac{(\gamma+1)^2}{\gamma} \qquad \text{公式（3.27）}$$

改进 ORB 算法的步骤如下：

1)使用 SURF 中多尺度空间的思想，获取出具有尺度不变的特征点。

2)求出特征点的质心方向。

3)求取 ORB 算法的特征点描述子。

4)采用汉明距离实现特征点的匹配。

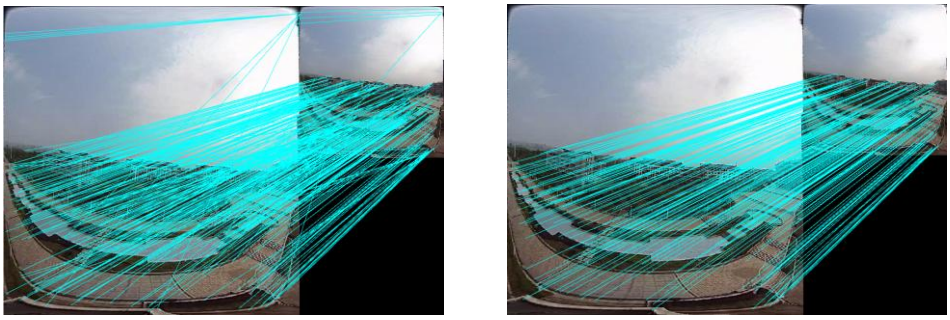
中国知网 https://www.cnki.net

3.6 实验结果与分析

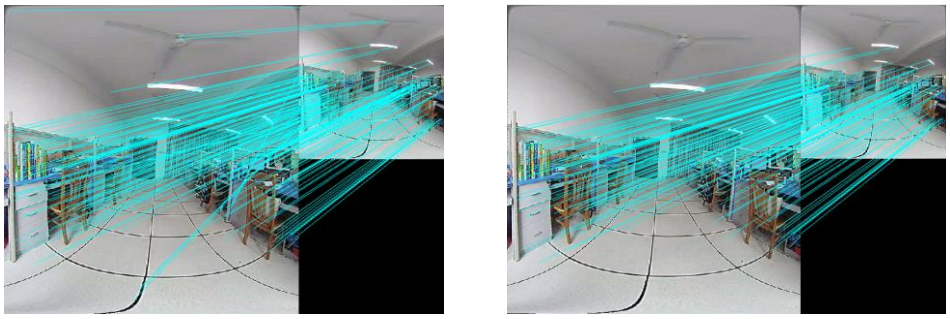
本节从多个方面对改进的 ORB、SIFT 和 SURF 进行了对比。在实验中，三者的最近邻域值的大小分别取 0.6，0.8，0.6，进行算法性能的比较，在此比较中不考虑此过程生成的误匹配点对。实验的环境是 32 位 Windows7，Intel 酷睿 i5，4G 内存，Matlab2015b。

3.6.1 尺度不变性对比试验

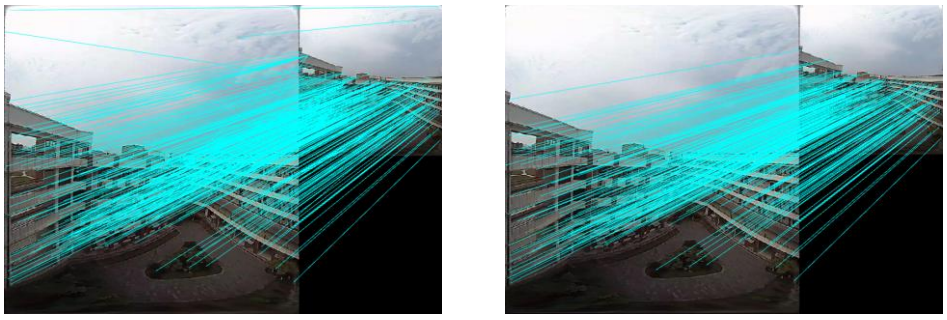
为了验证改进的 ORB 解决了传统 ORB 没有尺度不变性的这一缺点，做了如下的对比实验，将未发生尺度变化的图像与发生尺度变化的图像进行匹配，从而验证改进 ORB 在尺度不变性方面的性能：



(a)

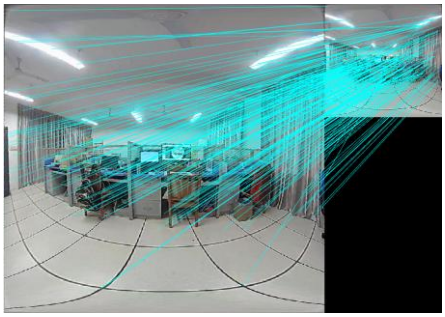


(b)

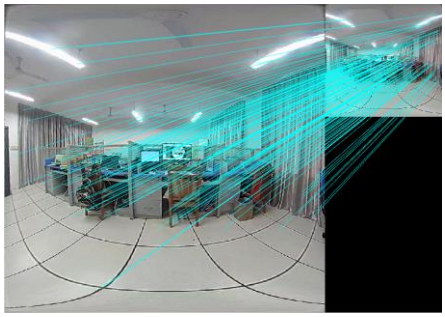


(c)

中国知网 https://www.cnki.net



ORB 特征点匹配结果图



(d) 改进 ORB 匹配结果图

图 3.16 (a)(b)(c)(d)图像尺寸变化的匹配对比

通过对比图 3.16 两侧匹配的图像看出，若匹配的物体的尺度的变化较大，改进的算法可以很好地对该物体进行匹配，对比其它算法，有着良好的性能。

以下对比了传统算法和改进 ORB 的配准度和耗费时间，并统计了以下 4 组数据。如表 3.1 所示：

表 3.1 各个算法准确度的对比

组别	算法	检测数目(对)	匹配数目(对)	匹配率(%)
1	ORB	325	67	20.6
	改进 ORB	215	192	89.5
2	ORB	312	66	21.3
	改进 ORB	280	252	90.1
3	ORB	226	53	23.4
	改进 ORB	201	183	91.2
4	ORB	214	48	22.4
	改进 ORB	198	185	93.2

表 3.2 平均匹配准确度(%)

组别	ORB	改进 ORB
1	20.6	89.5
2	21.3	90.1
3	23.4	91.2
4	22.4	93.2
平均	21.9	91.0

中国知网 https://www.cnki.net

表 3.3 ORB 和改进 ORB 时间对比

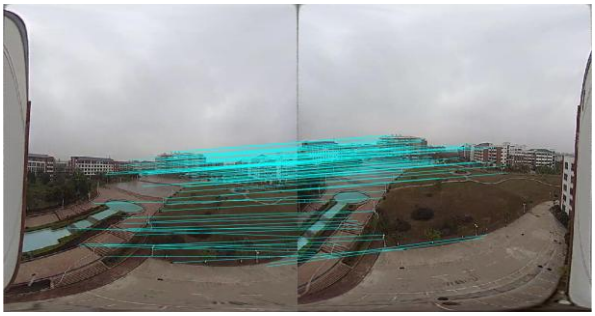
组别	ORB(ms)	改进 ORB(ms)
1	20.43	48.75
2	38.35	89.21
3	35.12	73.01
4	33.98	64.37

从表 3.2 和表 3.3 中可以看到，在 4 组数据中，改进 ORB 法的特征点均匹配正确率为 91.0%，比传统 ORB 算法的 21.9%要高，准确率是传统 ORB 的 4 倍，说明了改进 ORB 在尺度不变性上的优越性。从表 3.3 中看出，虽然改进的 ORB 算法在时间上有所增加，约为原始算法的 2 倍，但是特征点匹配准确率的提高，将有利于后面 PROSAC 算法去除误匹配的进行，所以虽然时间上有所上升，但不足以影响整个过程的算法性能。

3.6.2 改进 ORB 与其它算法的对比

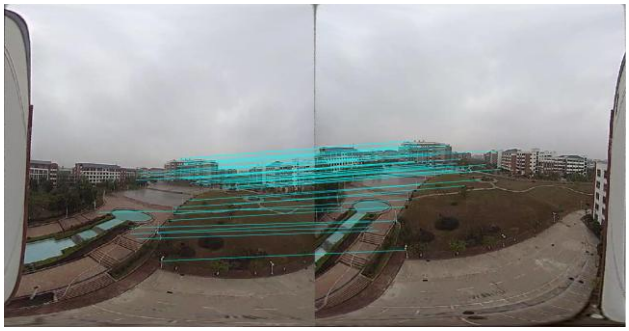
1.平移变化

图 3.17 平移测试原始图像

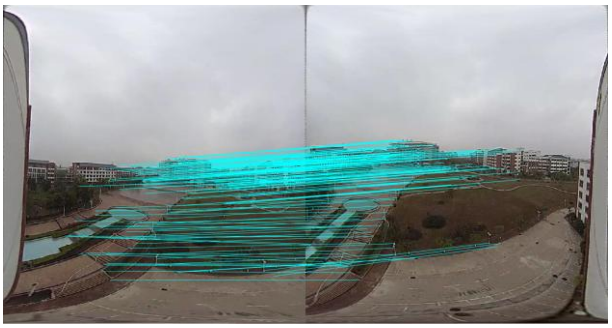


(a)SIFT 平移测试

中国知网 https://www.cnki.net



(b)SURF 平移测试



(c)改进 ORB 平移测试

图 3.18 (a)(b)(c)平移匹配结果

平移测试数据如下表 3.4：

表 3.4 平移测试数据

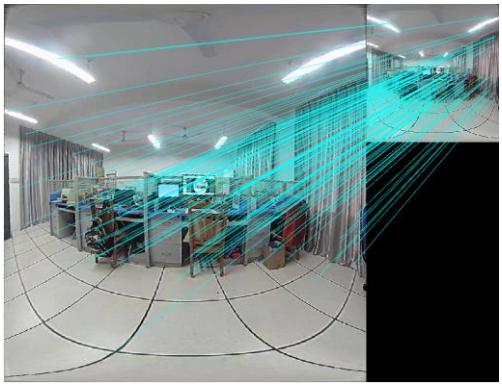
算法	特征点（对）	匹配的特征点（对）	匹配时间（s）	千特征点耗时（ms）
SIFT	410	275	0.216452s	527.9ms
SURF	213	102	0.153574s	721.0ms
改进 ORB	489	435	0.146703s	300.4ms

由平移实验下的数据可以看出：相对于 SURF 和改进的 ORB，SIFT 获得的特征点的数量是最多的，但同时从匹配的数量上来看，其误匹配的产生的概率还是很大；虽然 SURF 获得了较少的特征点的数目，但该算法正确匹配的特征点的数量要更高一些；改进的 ORB 算法由于是有角点检测算法组合而成的，所以它对角点的信息特别敏感，从而检测出了更多的特征点数目，这就更利于下一步特征点的匹配。

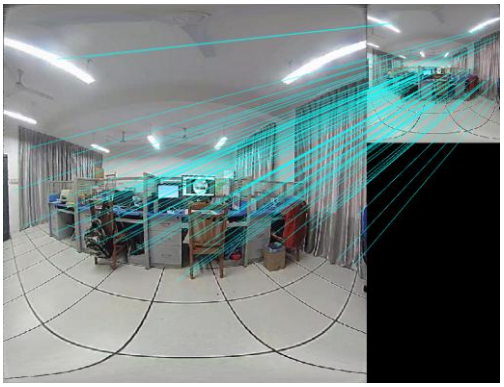
2.尺寸变化

中国知网 https://www.cnki.net

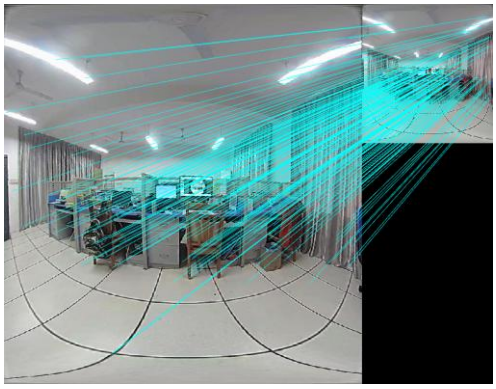
图 3.19 尺寸变化原始图像



(a) SIFT 尺寸变换



(b) SURF 尺寸变化



(c) 改进 ORB 尺寸变化

图 3.20 尺寸变化匹配效果

中国知网 https://www.cnki.net

结果如下表所示：

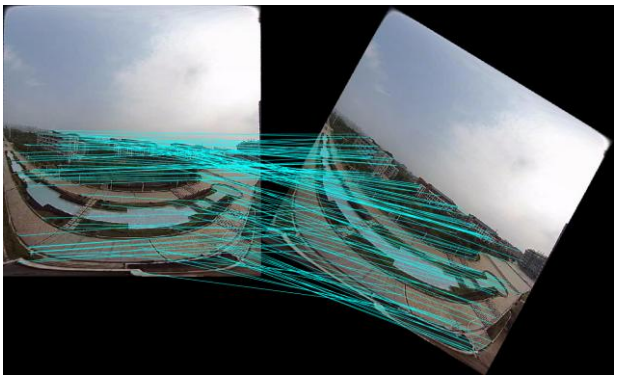
表 3.5 尺寸变化数据

算法	特征点数（对）	匹配的特征点数（对）	匹配时间（s）	千特征点耗时（ms）
SIFT	53	34	0.076541s	1444.1ms
SURF	46	24	0.035479s	771.2ms
改进 ORB	325	289	0.073691s	227.2ms

从尺寸变化的实验的数据中可以看出，SIFT 算法检测出的特征点在图像上的分布比较的均匀，对整个图像的信息都有一个很好的检测。SURF 和 SIFT 的匹配效果在尺度变化的测试上比较相近。而改进 ORB 算法在尺度变化时检测出了更多的特征点，说明了改进 ORB 对尺度变化的适应能力要比 SIFT 和 SURF 算法的高。

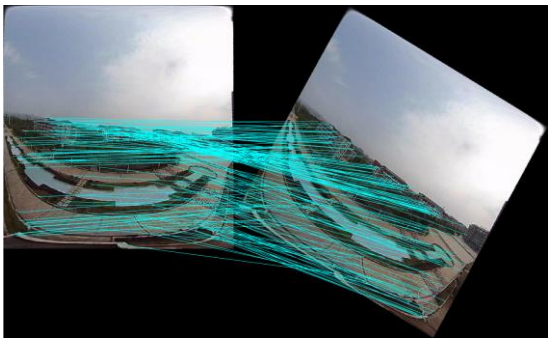
3.旋转

图 3.21 旋转测试原图

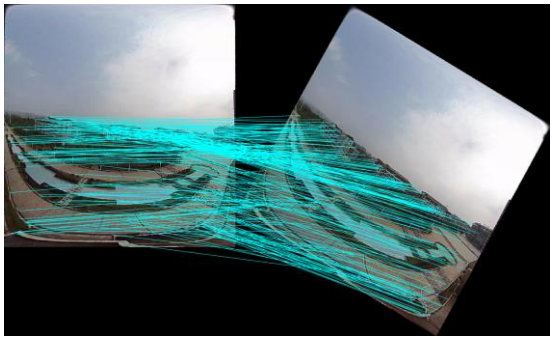


(a) SIFT 旋转变化

中国知网 https://www.cnki.net



(b) SURF 旋转变化



(c) 改进 ORB 旋转变化

图 3.22 旋转变化匹配效果

数据如下表所示：

表 3.6 旋转变化数据

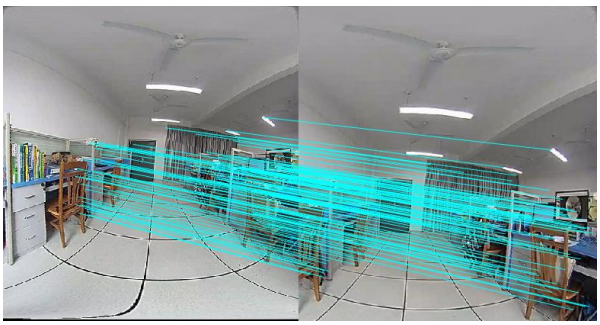
算法	特征点数（对）	匹配的特征点数（对）	匹配时间（s）	千特征点耗时（ms）
SIFT	375	235	0.176324s	470.1ms
SURF	302	168	0.158859s	526.0ms
改进 ORB	489	436	0.155781s	319.4ms

旋转测试中，效果最好的是 SIFT，其次是 SURF。虽然改进的 ORB 检测到较多的特征点，但是这些特征点主要分布在中部角点较为多的地方。

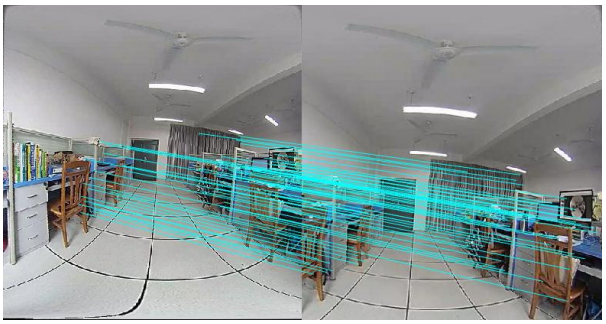
4.视角变化

图 3.23 视角变化测试原图

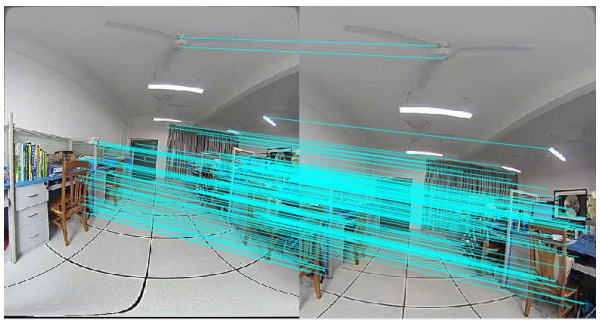
中国知网 https://www.cnki.net



(a) SIFT 匹配效果图



(b) SURF 匹配效果图



(c) 改进 ORB 匹配效果图

图 3.24 视角变化匹配

数据如下表所示：

表 3.7 视角变化测试数据

算法	特征点数（对）	匹配的特征点数（对）	匹配时间（s）	千特征点耗时（ms）
SIFT	474	102	0.271892s	573.6ms
SURF	358	86	0.237526s	663.4ms
改进 ORB	502	451	0.182343s	363.1ms

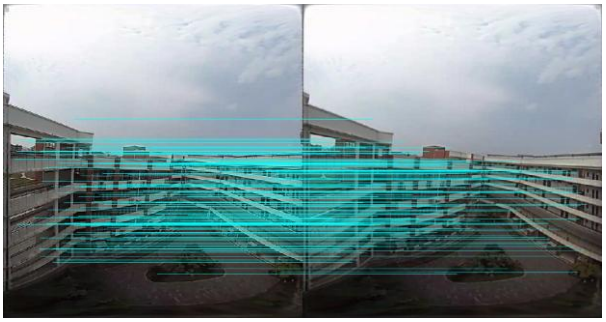
从视角变化得到的实验数据看出，SIFT 检测的效果要稍微好于 SURF。改进 ORB 检测出了最多的特征点，同时匹配的特征点的数目量也是最多的，说明了改进 ORB

中国知网 https://www.cnki.net

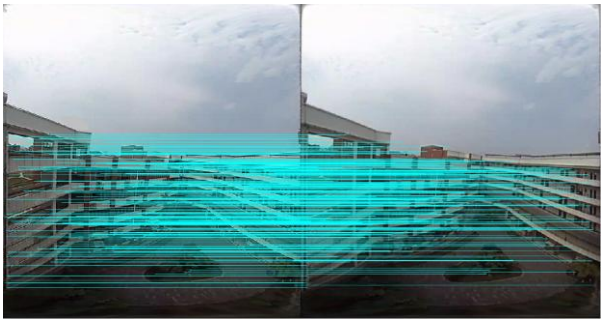
在视角变化情况下的性能，要好于 SIFT 和 SURF 算法。

5.尺度变化

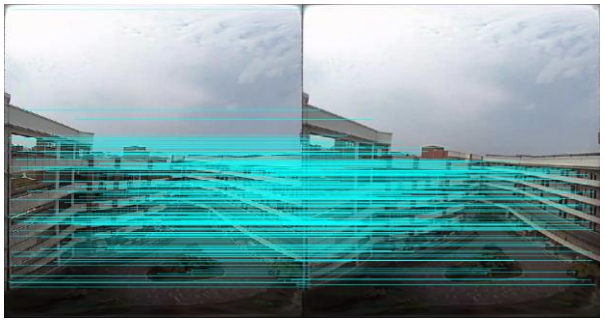
图 3.25 尺度变化测试原图



(a) SIFT 尺度变化



(b) SURF 尺度变化



(c) 改进 ORB 尺度变化

图 3.26 尺度变化匹配

中国知网 https://www.cnki.net

数据如下表所示：

表 3.8 尺度因子为 2.54 时测试数据

算法	特征点数（对）	匹配的特征点数（对）	匹配时间（s）	千特征点耗时（ms）
SIFT	306	228	0.256614s	838.6ms
SURF	546	260	0.374458s	685.5ms
改进 ORB	498	302	0.161900s	325.1ms

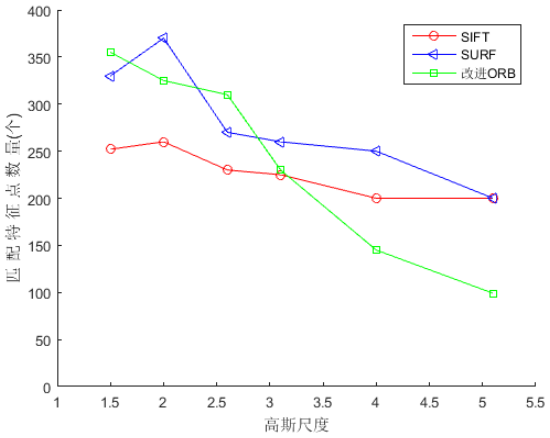


图 3.27 尺度变化对 SIFT、SURF、改进 ORB 算法的影响

从尺度测试结果数据可以看出，SIFT 和 SURF 都能检测出较多的特征点，并且匹配效果也很好，主要是建立尺度空间金字塔的原因。而改进 ORB 匹配正确的点数量最多，其原因在于 ORB 算法中的 R-BRIEF 描述子有着较大的方差和不相关性，这些特点使 ORB 可以检测出更多的特征点。改进 ORB 在高斯尺度因子 2.54 之后的曲线开始明显下降（图 3.27），反应了此时该算法的性能开始快速下降。图 3.28 对每个算法获得每千个特征点所耗费时间的统计，并用直方图的形式做出对比。

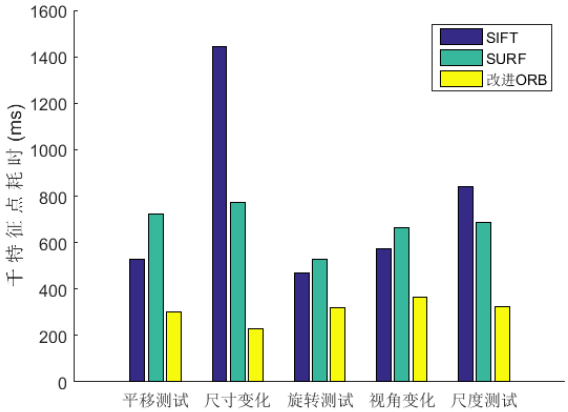


图 3.28 各个算法获取 1000 个特征点所用时间

中国知网 https://www.cnki.net

3.7 本章小结

本章对图像配准的方法进行了详细的介绍，它们都是基于图像的局部特征而提出来的。首先对 Harris 的原理作了详细分析，并介绍了基于梯度直方图的 SIFT 算法。由于此过程中要进行大量的计算，所以导致该算法速度很慢。对比 SURF 和 SIFT 算法，前者算法的性能弱于后者，但是 SURF 在速度上却有着独特的优势，这在计算上降低了内存的消耗，从而节省了资源。但是计算的速度上有一定的提高。ORB 是由 FAST 和 BRIEF 结合而成的，但由于生成的特征点没有尺度不变的特性，所以导致传统 ORB 没有尺度不变性。改进的 ORB 在增加尺度不变性以后，能够很好地对尺度变化的图像进行匹配。实验的结果表明，改进的 ORB 能够在尺度发生变化时，依然有着较高的特征点匹配正确率，并且与 SIFT 和 SURF 在速度上有着很大的优势，适用于对速度要求较高的场景中。

中国知网 https://www.cnki.net

第 4 章 图像特征点提纯

鱼眼镜头采集的图像在经过畸变校正以后，利用改进 ORB 算法提取特征点并生成描述子。当用于特征点的图像匹配时，由于受到光照、平移、采集设备抖动造成的影响，这些特征点不一定都能得到匹配，甚至可能产生误匹配的情况，这就会降低图片间单应性矩阵的精度，进而影响最终的拼接效果，因此对于匹配精度要求高的场合，需要进行去除误匹配的工作。

特征描述算子可以通过判定近邻距离与次近邻距离的比值来初步筛选误匹配。但是这样不足以有效的剔除误匹配点对，为此一些数学领域的方法被引入到特征点提纯的领域中，并取得了良好的去除效果，如 RANSAC[43]和 PROSAC[44]算法。

4.1 RANSAC 算法

RANSAC 算法的原理是：先得到一组测量的数据，该组数据包含有一定量的异常数据，该异常数据会导致产生错误的数学模型，我们使用这样的一组数据去估算要求的数学模型，从描述中我们可以发现，该算法具有不确定性和随机选取数据的特点[45]。然后根据随机选取的点先得到一个数学模型，这个数学模型不一定是最终所求的结果，我们还需要对这个初始的模型进行迭代，随着选取数据的增加，这样得到正确数据模型的概率就会变得更大。在该随机抽样性算法的模型中，我们把测试过程中的数据分为两种，一种是可以得到正确数据模型的数据，叫做内点；另外一种是不适合所要求取的数学模型的点，即异常的数据，我们称之为外点；与最小二乘法不一样的是，RANSAC 在建立模型的时候，用的是样本的值进行拟合，包含内点和外点[46]。由于在估算的过程中包含噪声数据，这就会造成还原数学模型的失败。而 RANSAC 可以很好的把内点和外点这两种数据分开。

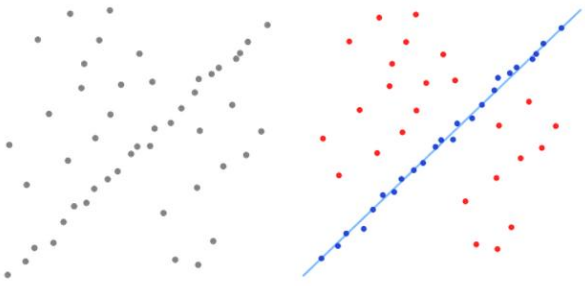


图 4.1 含有噪声的集合和 RANSAC 进行区分后的模型

中国知网 https://www.cnki.net

如图 4.1，左边是含有噪声的一组数据集合，具有较多的异常点数据。右图中为使用 RANSAC 算法来拟合的数学模型选出的内点，分布在蓝色直线周围的就为该模型计算出的内点，该直线外部的为筛选出的外点。右图看出，算法很好地进行了数据的区分，聚集了大部分正确的数据点，因此可以估算到准确的模型。

RANSAC 对数据进行采样的时候，各数据之间被采样到的概率是相同的，即采用的方法是随机进行采样，然后通过多次的迭代获取最优的数学模型，同时精炼出正确的数据。算法的步骤如下：

(1)在一个测试集 P 上估算该模型时，为了得到精准的结果模型，我们需要对该集合中的样本点的个数进行规定，设最少的个数为 n 。被用来作为采样集合中的采样点的个数 $Num(P)$ 要满足大于 n 这一条件。然后在集合 P 中进行采样，该采样的过程是进行的随机采样，把随机采样的点生成一个子集 S 。然后进行的是初始数学模型的估计，开始使用到的数据就是第一次采集到的子集 S 。

(2)对于剩下未被选取为采样点的集合 $SC = P / S$ 中的值与通过初始得到的模型 T 估算的值来进行对比，经过对比我们可以得到一个误差，对于误差在一定的阈值 t 内时，我们认为它是正确的数据，并把它加入到集合 S 中去，否则舍弃该数据。我们把这些正确的数据形成一个新的数据集 $S*$，即 $S*$ 为生成的内点集。

(3)当 $Num(S*) \geq N$ （ N 为给出的限定数量）时，则可认为得到了正确的模型。当使用原始的包含异常点的数据集合得出 T 之后，我们再使用新的经过筛选的数据集合去重新估算模型 T ，即是校正产生这个模型的数据偏差，然后使用这个经过修正的数学模型 T 去循环执行步骤（2）和（3）。

(4)当采取样本的次数达到我们设定 N_0 这一终止条件时，然后取出这个数学模型，并把它作为最终估算出的模型。

随机抽样一致性算法能很好的从包含有噪声的集合中除去这些异常的数据，从而使循环的过程更加的高效，进一步提升获取正确数学模型的可能性，使获取到的模型对于异常的数据由更强的鲁棒性[47]。因而该算法有着广泛的应用，特别是在图像配准、图像重建以及干扰较多的云点图下，由于误匹配点影响配准过程中的计算匹配点对的精度，因此需要对产生的误匹配点进行提纯操作。

4.2 PROSAC 算法

2005 年的 CVPR 会议上，Ondrej Chum 等人提出了 PROSAC(Progressive Sample Consensus)算法，它与 RANSAC 算法不同在于：RANSAC 使用同等的选取顺序获取特征点，而 PROSAC 是先根据采样点的互相关联性从大到小进行排列，先从相关性最大的点开始作迭代计算，然后进行假设、验证，获得数学模型的解。PROSAC 假设的前

中国知网 https://www.cnki.net

提是内点要大于外点的数量。而对于噪声不是很大的场景，都能满足这个条件。若一个采样集合中的数据不满足内点个数大于外点的个数，此时的 PROSAC 计算出的模型性能与 RANSAC 相比较，也是具有一定优势的。因此无论测试集合中的噪声数据的百分比有多大，PROSAC 算法的效率都是高于 RANSAC 的。这里使用的 ORB 算法，其具有内点多于外点的特点，使用 PROSAC 出去误匹配点对是可行的。

U_N 为采样点集，其中共有 N 个点，里面是按照点的质量降序存放的，质量 $q()$ 为:

$$u_i, u_j \in U_N : i < j \Rightarrow q(u_i) < q(u_j) \qquad \text{公式（4.1）}$$

PROSAC 算法的两个关键问题：生长集的大小设置和终止条件。

1.增长函数和抽样

为了采集到适合模型的点，要有采样的策略，并且它可以获取到正确的集合。由于每次进行抽样时的子集合的大小是变动的，因此我们需要对这个子集合的大小进行限制，所以我们以此来定义了 PROSAC 算法控制子集合增长的函数。

设 u_i 为正确的数据点即内点，$P\{u_i\}$ 表示该点被采样到的概率，对 $P\{u_i\}$ 和 u_i 提出这样的假设：在集合中，按照相关性从大到小进行排序以后，则排序越靠前的集合点，是内点的可能性也就越大，被采样的概率也就比排序靠后的点的大：

$$q(u_i) < q(u_j) \Rightarrow P\{u_i\} \geq P\{u_j\} \text{，也即} i < j \Rightarrow P\{u_i\} \geq P\{u_j\} \qquad \text{公式（4.2）}$$

采样策略：采用和 RANSAC 相同的策略，从几何 U_N 选出 T_N 个子集合，这些子集合的质量为 m，记为 $\{M_i\}_{i=1}^{T_N}$ 在得到采样点的质量以后，我们按照质量从大到小对这些对应的采样点进行从大到小的排序：

$$i < j \Rightarrow q(M_{(i)}) \geq q(M_{(j)}) \qquad \text{公式（4.3）}$$

若对集合的采样根据 $M_{(i)}$ 来进行，那么质量大的内点就容易先被提取到并用作模型的估计。相反，质量越低的外点的位置就越靠后，因而就不容易被抽取到。U_n 为上步选取的从小到大排序好的点的集合，记 T_n 为这些质量高的点的平均值，这些点为 $\{M_i\}_{i=1}^{T_N}$ 和 U_n 的交集：

$$T_n = T_n \frac{\binom{n}{m}}{\binom{N}{m}} = T_N \prod_{i=0}^{m-1} \frac{n-i}{N-i} \qquad \text{公式（4.4）}$$

中国知网 https://www.cnki.net

则有：$\frac{T_{n+1}}{T_n}=\frac{T_N}{T_N}\prod_{i=0}^{m-1}\frac{n+1-i}{N-i}\prod_{i=0}^{m-1}\frac{N-i}{n-i}=\frac{n+1}{n+1-m}$

可以得到T_{n+1}的递归关系式：

$$T_{n+1}=\frac{n+1}{n+1-m}T_n \qquad \text{公式（4.5）}$$

因此T_n中有着质量更高的内点，其原因在于这些点取自于集合T_n，此时T_n中的高质量的数据已经进行了排序。又有集合关系$U_{n+1}=U_n\cup\{u_{n+1}\}$，则$T_{n+1}-T_n$数据点都可以在$U_{n+1}$和$U_{n+1}$个$U_n$中找到对应的点。通常$T_n$的值不是整数，设$T'_{n+1}=1$，$T'_{n+1}=T'_n+\left|T_{n+1}-T_n\right|$。

定义增长函数：$g(t)=\min\{n:T'_n\geq t\}$，在该算法中，第$t$次获得的集合$M_t$是$M_t=\{u_{g(t)}\}\cup M'_t$，这里$M'_t\subset U_{g(t)}$为一个从$U_{g(t)}$中随机抽取的势为$\left|M'_t\right|=m-1$的集合[48]。$T_N$是经过多少采样后 PROSAC 和 RANSAC 达到的同样状态，在 PROSAC 的试验中$T_N=200000$。

PROSAC 算法步骤如下：

(1) 设置初始值$t:=0$，$n^*:=N$，执行(2)(3)(4)直到达到终止条件；

(2) 选择假设生成集合

令$t:=t+1$，如果$t=T_n'$，且$n<n^*$，有$n:=n+1$；

(3) 势为m的半随机采样集合M_t

如果$T_n'<t$，从U_{n-1}集合中随机抽取$m-1$m-1 个点和U_n，

否则，从U_n中随机抽取m个点；

(4) 模型参数预测

M_t为采样的集合，我们需要以这个采样的集合为基准，再进一步计算来得出模型参数 p_t；

(5) 模型检验

使用上步根据采样集合计算出的数学模型，在未进行搜索的集合中寻找与模型进行匹配的点，并设置终止的条件，以判断在满足条件时，终止算法的进行。

2.终止条件

若合U_n中获得的内点个数I满足下面两因素时，该算法就停止循环，并给出此时计算出的数学模型:

第一，当非正确模型计算的内点数量I_n的概率低于百分之五时。

第二，在设定的抽样次数k以后，仍然有一定数目的内点未被该模型检测到。

中国知网 https://www.cnki.net

4.3 去除误匹配试验

使用改进 ORB 与上文介绍的两种误匹配去除算法配合起来使用，进行实验并统计实验数据，对比两组实验数据得到的效果，如下图所示:

图 4.2 匹配原图

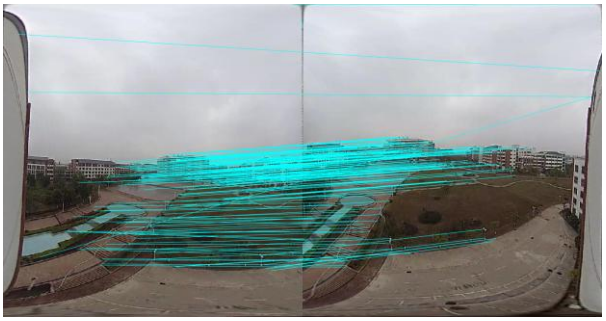


图 4.3 改进 ORB+RANSAC 效果

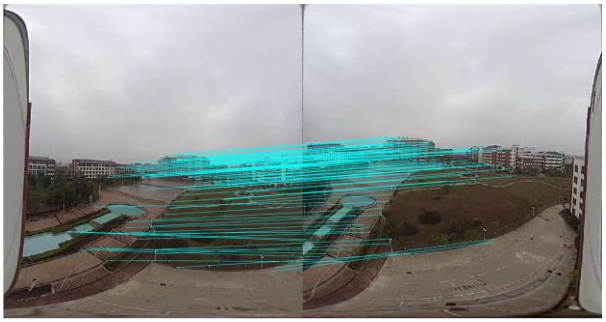


图 4.4 改进 ORB+PROSAC 效果

表 4.1 改进 ORB+RRANSAC

匹配总时间	改进ORB时间	RANSAC时间	特征点数	内点个数
89.931ms	45.325ms	43.568ms	265	205

表 4.2 改进 ORB+PROSAC 匹配结果

匹配总时间	改进ORB时间	RANSAC时间	特征点数	内点个数
49.125ms	44.568ms	4.981ms	265	198

中国知网 https://www.cnki.net

经过 ORB 算法进行特征点的初步筛选以后，我们结合本章节的实验算法，使用 RANSAC 和 PROSAC 在此基础上再次进行错误配对点的去除操作。在图 4.4 中看出错误匹配点对，在数次迭代后被删除。RANSAC 和 PROSAC 算法的都能很好地除去该过程产生的误匹配点。最后获得了相近的内点个数，如表 4.1 和表 4.2 所示，可以看出与改进的 ORB 一起进行运算后也没增加太多匹配过程所需时间。

4.4 本章小结

本章研究了采样一致性方法 RANSAC 和 PROSAC 算法，通过实验仿真对比了两种方法的性能。与 RANSAC 的随机抽取不同，PROSAC 算法先对采样点用质量函数进行一次排序，先从质量好的点采样，从而降低获得相同内点时需要的采样次数，节省去除误匹配时间。实验结果表明：改进 ORB 与 PROSAC 可以明显的减少误匹配，提高匹配的准确性，为后续图像拼接的进行提供了精度上的保证。

中国知网 https://www.cnki.net

第 5 章 鱼眼图像的拼接与融合

鱼眼图像的校正和特征点的匹配完成以后，下一步要进行的就是对图像的拼接。经过融合的图像往往会出现拼接缝或两张图片的亮度不一致的情况，这是由于拍摄图像时的光照、角度等不一样而造成的。若要解决以上拼接产生的问题，我们可以使用图像融合技术。该技术是通过一定的算法来实现各部分信息的融合，最后获得一幅视野广阔的鱼眼拼接图像。

融合后的图像应该满足[49]：拼接后图像的信息损失量应该最少，或者在拼接过程中尽量减少原图细节信息的丢失；图像的重叠的区域可以很好地互相融合，并且不会出现明显的明亮不一的现象。

5.1 图像的插值技术

图像插值指的是依据已经确定的像素点的值，来确定与之相关的点的值的过程。由于经过转换矩阵计算后点的值往往不是整数，所以要对转换后的坐标值取整，导致原有图像的像素会出现丢失，某些区域就会出现视觉上的不完整。因此，我们就需要使用插值的方法，来恢复这些像素点的值，这样才能使拼接后的图像看起来平滑，不会使一幅图像有明暗现象或者拼接缝的出现。

灰度的插值技术主要有两种：向前和向后映射法[50]。向前映射是指把原像素点的坐标值，使用一定的算法计算并赋值给新的坐标点。该算法为差值算法，它可以将源图像某点周围四个像素点的值进行平均，并赋值给该点。向后映射法：与向前映射正好相反它从所要得到的图像开始，按照像素点映射回源图像。因此我们可以看出，在两个映射方法的过程中，所使用的映射矩阵之间的关系是互逆的。

向前与向后映射法的对比，向前映射法的缺陷：第一，若对源像上的像素直接使用插值，会超出融合画布的范围，这样画布就不能容纳所有的源图像的信息，从而导致无法显示完整的图像。第二，根据向前映射的算法可知，若要得到一个像素的值，那么需要计算它周围像素的值，这样叠加在一起，运算量就会很大。向后映射法：每个输出像素的值，至多可以由周围四个像素点的值来计算得到，并且是按照每个像素、每行为单位进行输出。当图像的大小没有发生变化，即几何变化时，我们采用向前映射。若一个图像存在着大小的变化，比如图像进行了旋转、缩放，那么我们采用向后映射。由于实际的图像变化中，很难保证图像不发生大小和旋转等的变化，因此我们通常使用向后映射的方法，来得到需要的图像。

被广泛采用的插值方法：

（1）最近邻的插值方法。

中国知网 https://www.cnki.net

（2）双线性插值的方法。

（3）三次线性插值的方法[51]。

5.1.1 最近邻插值法

该方法是把距映射位置距离最短的对应点的像素灰度值，看成是输出图像的灰度值，它的过程简单并且易于计算，所以也为零阶的插值插值方法。表达式为:

$$\begin{cases} f(x)=f(x_t)\ \dfrac{1}{2}(x_{t-1}+x_t)<x<\dfrac{1}{2}(x+x_{t+1}) \\ f(y)=f(y_t)\ \dfrac{1}{2}(y_{t-1}+y_t)<y<\dfrac{1}{2}(y+y_{t+1}) \end{cases} \qquad \text{公式（5.1）}$$

式中：(x,y)分别代表的是源图像的两个坐标的值，经矩阵转换后的输出图像的坐标值的大小为(x_i,y_i)。

5.1.2 双线性插值

它是由未知像素点的四个方位的点的值来求得中心点处点的值，即求四个方位像素点值的求得。在两个坐标轴的方向上进行插值，将获得的灰度值作为输出的灰度值。该方法具体的插值如下图所示：

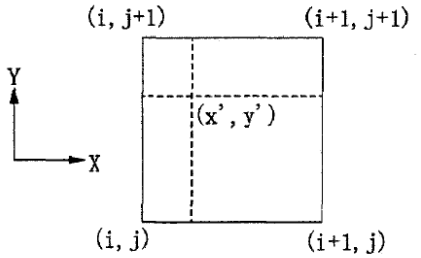


图 5.1 双线性插值算法

设点$(i+m,\ j+n)$是通过映射得到的值，则$f(i+m,j+n)$就为该像素点的值，$f(i+m,\ j+n)$由源图像周围点的值$f(i,j)$，$f(i+1,j)$，$f(i,j+1)$，$f(i+1,j+1)$确定。则插值函数为:

$$\begin{aligned} f(x,y)=f(i+m,j+n)=(1-m)(1-n)f(i,j)+ \\ (1-m)nf(i,j+1)+m(1-n)f(i+1,j)+mnf(i+1,j+1) \end{aligned} \qquad \text{公式（5.2）}$$

中国知网 https://www.cnki.net

式中：i, j 表示的是像素点的两坐标值的大小，u, v 分别为非整数型的两坐标的值，它们的取值范围为[0, 1]。

5.1.3 三次卷积插值法

三次卷积插值法是利用三次多项式，使函数 $T(x)$ 靠近最优化的 $sin(x) / x$。该算法根据像素点周围四个方位上的像素点，以及外围共 16 个点的像素值的大小，进行计算，这样可以更精确地计算像素点的值。函数 $T(s)$ 的表达式为:

$$T(s)=\begin{cases}1-2|s|^2+|s|^3 & 0\le|s|<1\\ 4-8|s|+5|s|^2-|s|^3 & 1\le|s|<2\\ 0 & |s|\ge 2\end{cases} \qquad \text{公式（5.3）}$$

则所该像素的值为：

$$f(x,y)=f(i+m,j+n)=UVW \qquad \text{公式（5.4）}$$

式中:i，j 分别为行、列的整数部分；m, n 分别为行、列的小数部分。（5.4）中，U、V、W 分别定义为:

$$U=[T(1+n)T(n)T(1-n)T(2-n)] \qquad \text{公式（5.5）}$$

$$V=\begin{bmatrix} f(i-1,j-1) & f(i-1,j) & f(i-1,j+1) & f(i-1,j+2)\\ f(i,j-1) & f(i,j) & f(i,j+1) & f(i,j+2)\\ f(i+1,j-1) & f(i+1,j) & f(i+1,j+1) & f(i+1,j+2)\\ f(i+2,j-1) & f(i+2,j) & f(i+2,j+1) & f(i+2,j+2)\end{bmatrix} \qquad \text{公式（5.6）}$$

$$W=[T(1+m)T(m)T(1-m)T(2-m)]^T \qquad \text{公式（5.7）}$$

综合以上的分析，最近邻插值法运算过程简便，并且在整个过程中不会造成原图像细节信息的缺失，但该算法是由邻近点计算得到的，所以获得的像素点的值精准度不高；三次线性插值法能够保证图像的细节不受到破坏，并且运算后能得到精确的像素点的值，但是由于计算过程的复杂性，导致求像素点时的运算量很高，这就影响了图像插值算法的速度和性能；双线性插值法的可以在保证计算所得的像素值高的前提下，使求解像素值的过程的运算量不是很大，能符合本文对算法性能的要求[52]。因此，结合实验的需要及算法性能的表现，本文使用双线性插值的方法进行图像的插值运算。

中国知网 https://www.cnki.net

5.2 图像的融合方法

为了消除拼接痕迹并使拼接后的区域看起来自然，平滑。需要使用合适的融合算法处理拼接后产生的拼接缝等效果。研究者们主要提出以下的三种融合的方法[53]，下面对这些技术进行介绍：

5.2.1 取平均值法

所谓均值法，指的是先获取拼接区域对应点的像素值，然后把两者进行算术加法，最后求两个像素点的均值，M_1,M_2是两幅原始图像，M为拼接图像。$M_1(x,y)$、$M_2(x,y)$和$M(x,y)$分别是这三幅图像上点(x,y)处的灰度值，则融合之后点的对应关系如下:

$$M(x,y)=\begin{cases} M_1(x,y) & (x,y)\in N_1 \\ \frac{1}{2}[M_1(x,y)+M_2(x,y)] & (x,y)\in(N_1\cap N_2) \\ M_2(x,y) & (x,y)\in N_2 \end{cases} \qquad \text{公式（5.8）}$$

式子中：N_1和N_2为两幅待拼接图像没用公共区域的部分，N_3代表的是两幅图像公共的区域。

5.2.2 渐入渐出法

该算法是目前使用的较广泛的融合算法。该算法不直接对响应的像素点实行平局值的求取[54]。与其它直接进行平均的算法不一样的是，首先算法先进行加权，然后再使用相加平均法取得重叠区域像素各像素点的值。图像重叠区域像素的权值不是由周围的像素点的值决定的，它取决于一个边界距离。这个边界距离指的是由该点到两幅图像有公共区域部分边界的长度。设$M1$和$M2$为待拼接图像，M为融合后的图像这些图像在点(x,y)处的像素值为$M_1(x,y)$、$M_2(x,y)$、$M(x,y)$。则融合后各点的像素值为:

$$M(x,y)=\begin{cases} M_1(x,y) & (x,y)\in N_1 \\ d_1(x,y)M_1(x,y)+d_2(x,y)M_2(x,y) & (x,y)\in(N_1\cap N_2) \\ M_2(x,y) & (x,y)\in N_2 \end{cases} \qquad \text{公式（5.9）}$$

式中：$d_1(x,y)$、$d_2(x,y)$两式为原图像有公共区域的像素点的值。如图 5.2，设公共区域x轴上的最大的值、最小的值分别表示为$x_{\max}$和x_{min}，d_i为权值因子i的取值

中国知网 https://www.cnki.net

为 1 或者 2。且 $d_l + d_2 = 1,\ 0 \le d_l,\ d_2 \le 1$，则：

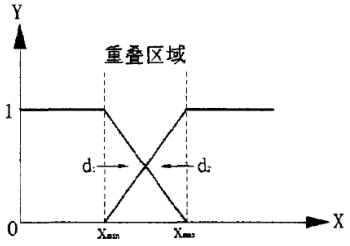


图 5.2 渐入渐出法示意图

5.2.3 多分辨率样条法

该算法是利用金字塔的方式，不同于其它的算法，该金字塔称为 Lpalacian 多分辨率。它把图像分为多组频率不相同部分，每组内的频率也是不一样。对各组图像上相同频率的部分使用平均的方法，该处使用的平均方法是加权平均的方法。并且该算法是融合是在有公共区域的边界处进行的，然后把每个频率上得到的拼接图像叠加起来，进而把两张图片拼接成一幅完整的图像。

融合最方便的就是对像素点的值直接进行求均值的操作。该算法的简单之处还在于运算量较小，缺点是该算法在公共的区域融合以后，会有带状条纹的出现[55]。多分辨率法虽然可得到效果好的融合图像，算法的准确度高，不会使融合后的图像丢失大量的信息，但是算法的运算过程复杂，会导致运算速度慢，从而很难应用到实时性场合较高的场景中；渐入渐出法较为简单且运算量小，最后的融合效果也是比较好。所以，依据实验的要求，本文融合部分采用渐入渐法来实现图像的融合。

5.3 本文采用的图像拼接与融合的方法

设 M_1、M_2 分别为两张待融合图像，它们相互之间有着公共的区域。H 为该过程中要用到的变换矩阵，求出该矩阵的 8 个参数。就可以获得 M_1 图像在区域的映射范围的大小，这个区域通常称为画布。可以通过公式（5.10）和（5.11） 求出这个“画布”的大小:

$$x_i' = \frac{m_{11}x_i + m_{12}y_i + m_{13}}{m_{31}x_i + m_{32}y_i + 1} \qquad \text{公式（5.10）}$$

$$y_i' = \frac{m_{21}x_i + m_{22}y_i + m_{23}}{m_{31}x_i + m_{32}y_i + 1} \qquad \text{公式（5.11）}$$

中国知网 https://www.cnki.net

式子中：(x_i', y_i')是待配准的图像M_2上的像素点(x,y)在参考图像M_1上的对应点。

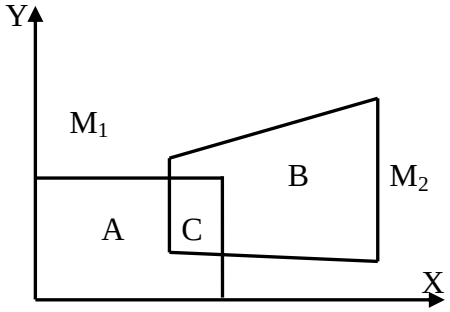


图 5.3 图像拼接与融合示意图

本文首先根据变换矩阵H求出图像M_2映射到图像M_1所在平面的投影区域，确定拼接图像的坐标范围；然后将拼接图像分为A、B和C三个区域，其中A区表示只属于参考图像M_1的部分，B区表示只属于配准图像M_2的部分，C区表示参考图像M_1和配准图像M_2的重叠区域；最后针对每个区的不同情况使用不同的方法确定A,B,C三个区的像素值。

(1) A区域: 由于A只属于参考图像M_1，所以A上各点的像素值对应于参考图像M_1各点的像素值。

(2) B区域: B区域是经过变换后映射到参考图像上的只属于配准图像M_2的部分，该区域像素的值采用下面的方法进行计算:

1）首先采用向后映射法，由公式（5.12）和（5.13）计算出区域B中的点(x',y')在M_2上所对应的点(x,y)：

$$x=\frac{bf-ce}{bd-ae} \qquad \text{公式（5.12）}$$

$$y=\frac{cd-af}{bd-ae} \qquad \text{公式（5.13）}$$

其中：

$$a=m_{31}x'-m_{11}\quad b=m_{32}x'-m_{12}\quad c=m_{13}x'-x'$$
$$d=m_{31}y'-m_{21}\quad e=m_{32}y'-m_{22}\quad f=m_{23}-y'$$

2）接下来使用双线性插值来进行，其主要的原因就是上一步计算出的坐标点(x,y)常常不是整数，使(x,y)点处的像素值可能不能直接从M_2上读取出来。因此需要根据式（5.14）使用双线性插值法来计算图像M_2在(x,y)点的值，并将其作为拼接图在点(x',y')处的像素值。

中国知网 https://www.cnki.net

$$M_2(x,y)=(1-\Delta x)(1-\Delta y)M_2([x],[y]+1)+\Delta x(1-\Delta y)M_2([x]+1,[y]) \\ +\Delta x\Delta y M_2([x]+1,[y]+1)$$ 公式（5.14）

（3）C 区域：区域 C 是待拼接图像 M_1 和 M_2 的重叠区域，为了使重叠区域平滑过渡，这部分区域各点的像素值采用渐入渐出的加权平均法。设 M 为最终拼接完成的图像，则 $M(x,y)$ 可以表示为：

$$M(x,y)=d_1M_1(x,y)+d_2M_2(x,y)$$ 公式（5.15）

其中：$d_1=\dfrac{x_{\max}-x}{x_{\max}-x_{\min}}$，$d_2=1-d_1$

5.4 实验结果与分析

本系统实验环境为：32 位，Windows7 操作系统，I5 酷睿，内存 4G，基于 matlab2015b+Visual Studio2015+Opencv2.3.4。实验对象为：使用海康威视 DS-2CD3942F-I 型号的鱼眼摄像头对学校的图书馆进行采集的图像，即获取的两张具有重叠区域的鱼眼图像。实验如下图所示：

(a) 鱼眼图像 1　　(b) 鱼眼图像 2

图 5.4 原始鱼眼图像

使用经度定位校正法对图 5.4 中的两幅鱼眼图像进行校正，校正后的图像有较大的重叠区域，可以满足拼接的要求。鱼眼图像拼接结果如图 5.5 所示：

中国知网 https://www.cnki.net

5.5 经过拼接后的鱼眼图像

通过实验，获得了一幅拼接的图像，由于先对右边图像使用变换矩阵 H 进行形变来与左侧图像进行拼接，导致拼接后的图像的最左侧出现了黑色的空白区域。实验的结果表明：通过对校正后的鱼眼图像的处理，图像在重叠区域处并没有出现明显过渡的拼接缝，两边的亮度基本上一致，从而实现了图像像素的平滑过渡，实验取得了较好的拼接效果，达到了本文对鱼眼图像的采集、校正进而拼接的目的。

5.5 本章小结

图像融合的目的就是消除在融合过程中产生的亮度不一致等问题，从而使图像看起来更加自然。本章介绍了常用的图像插值的方法：首先是简便，计算量较小的最近邻插值法；其次是计算精确但计算量较大的双线性插值法；最后是效果好，但是计算过程复杂的三次线性插值法。本章把待融合的图像分为三个不同的部分：两个没有公共区域的部分和一个有着公共区域的部分。无公共区域的部分使用向后映射法来处理，接下来使用的插值的法为双线性插值。有相同部分处的像素点的值使用的是渐入渐出法来确定像素点的值。从实验中可以看到，拼接后的图像没有出现明显拼接缝隙，并且进行融合部分的亮度平滑过渡，整幅图像的明暗比较一致。

中国知网 https://www.cnki.net

第 6 章 总结与展望

6.1 总结

本文主要以鱼眼镜头采集的图像为研究对象，展开图像拼接技术的研究工作，通过对实验的分析和研究取得了以下的结论：

(1) 基于经度坐标定位的校正法在图像的上下边缘保留着少量的形变，且该算法在校正的过程中没有像素的丢失，从而使源图像的细节内容能够很完美地保留下来。

(2) 改进的 ORB 算法进行校正图像的特征点提取，它能在保留传统 ORB 算法优点的基础上，使传统的 ORB 算法具有尺度不变性，提高了特征点检测的精度。

(3) 使用 PROSAC 去除在特征点的匹配过程中产生的误匹配点，与 RANSAC 不同之处在于 PROSAC 算法进行采样的次数要比随机情况下少得多。

(4) 采用渐入渐出以及图像插值的方案消除融合后图像产生的重影，拼接线等现象，进而使拼接后的图像更加自然。为实际环境中鱼眼镜头下的自然场景中的图像拼接系统的开发提供了理论基础和可行性分析，具有一定的实际应用价值。

6.2 展望

当前虽然已经基本实现了鱼眼图像拼接的主要的功能，但由于鱼眼图像的校正比较复杂，并且校正的过程涉及到不同领域内的知识，因此对于鱼眼图像校正的工作还需要进行进一步的深入研究。图像特征点的提取算法也是众多，其算法的选择是本着效果好、速度快等来进行选择的，本文选择的 ORB 算法在速度方面有着独特的优势，但在准确度和速度上仍然有提升的空间，需要今后进一步的探讨和研究。

(1) 鱼眼图像的运算量问题。本文选择的基于经度的鱼眼图像校正算法虽然能完成校正的工作，但是其运算量大，对于高质量图像的速度校正时间长，无法达到实时性要求高的场景使用的要求。

(2) 特征点检测的问题。改进的 ORB 算法虽然能够在一定的程度上解决尺度变化的问题，但也因此增大了运算所需要的时间，因此还需要对该算法的性能进一步优化，从而使检测特征点的效率更高。

(3) 系统完整性的问题。文章目前只是以静态的图像为对象进行了研究，凭借本文算法的优越性，可以迁移到鱼眼视频拼接上，因此对于鱼眼视频拼接相关的内容还需要进行学习和探讨。

中国知网 https://www.cnki.net

参考文献

[1] 吕丽军, 吴学伟. 鱼眼镜头初始结构的设计[J]. 光学学报, 2017(2):97-106.

[2] 曹芳, 王小攀, 朱永丰,等. 基于 SIFT 特征和改进融合策略的图像拼接算法[J]. 测绘与空间地理信息, 2016, 39(5):94-97.

[3] 何源. 鱼眼图像无缝全景拼接技术研究[D]. 山西农业大学, 2015.

[4] Garcia-Fidalgo E, Ortiz A, Bonnin-Pascual F, et al. Fast image mosaicing using incremental bags of binary words[C]. IEEE International Conference on Robotics and Automation. IEEE, 2016:1174-1180.

[5] 范莹. 基于双目视觉的图像匹配与定位技术的研究[D]. 江南大学, 2016.

[6] 宋涛, 褚光宇, 侯培国,等. 基于质心点优化的鱼眼摄像机标定[J]. 光子学报, 2016, 45(5):100-105.

[7] 刘亿静, 苗长云, 杨彦利. 基于经纬映射的径向畸变快速校正算法的研究[J]. 激光杂志, 2015(1):1-4.

[8] 唐思源, 白金牛, 杨敏. 红细胞 SEM 图像的三维重构及形状特征提取[J]. 微型机与应用, 2016, 35(18):45-47.

[9] 曲瑾. 基于角点检测的目标跟踪算法研究与应用[D]. 哈尔滨工业大学, 2015.

[10] 陈磊. 图像配准中基于特征提取和匹配的方法研究[D]. 吉林大学, 2016.

[11] 李效周, 赵洋, 贾景强,等. 一种基于双边滤波的高动态范围图像再现[J]. 齐鲁工业大学学报: 自然科学版, 2016, 30(6):6-11.

[12] 何敬. 基于点线特征匹配的无人机影像拼接技术[D]. 西南交通大学, 2013.

[13] 朱奇光, 张朋珍, 李昊立,等. 基于全局和局部特征融合的图像匹配算法研究[J]. 仪器仪表学报, 2016, 37(1):170-176.

[14] 张峥. 基于词袋模型的高分辨率影像宗建筑物提取方法研究[D]. 西安科技大学, 2015.

[15] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, 2001:I-511-I-518 vol.1.

[16] Kohler T, Heinrich A, Maier A, et al. Super-Resolved Retinal Image Mosaicing[J]. 2016:1063-1067.

[17] Zouhir I, Abdelhai L, Samir S, et al. FPGA implementation of the RANSAC based image mosaicing algorithm using the Nios II softcore[C]. International Conference on Systems, Signals and Image Processing. 2016:1-4.

[18] 张志宝, 孙微涛, 罗文峰. 基于 HSI 空间改进的彩色图像边缘检测方法[J]. 计算机与数字工程, 2016, 44(11):2257-2262.

[19] 陈德勇. 大幅面 CIS 扫描仪的自动图像拼接技术研究[D]. 电子科技大学, 2015.

中国知网 https://www.cnki.net

[20] 王文润，王阳萍，WANGWen-run,等. 移动增强现实三维注册中特征提取算法改进[J]. 计算机仿真, 2015, 32(12):231-234.

[21] Vaghela D, Naina K. A Review of Image Mosaicing Techniques[J]. Eprint Arxiv, 2014, volume 2(3).

[22] Bind V S, Muduli P R, Pati U C. A Robust Technique for Feature-based Image Mosaicing using Image Fusion[J]. International Journal of Advanced Computer Research, 2013, 3(8).

[23] Pushpakar A, Dube N, Dhar D, et al. Improved image mosaicing technique using marker controlled watershed segmentation[C]. International Conference on Signal Processing and Communication Engineering Systems. IEEE, 2015:62-66.

[24] Joshi H, Sinha K L. A Survey on Image Mosaicing Techniques[J]. International Journal of Advanced Research in Computer Engineering & Technology, 2013, 2(2).

[25] 袁杰. 基于 SIFT 的图像配准与拼接技术研究[D]. 南京理工大学, 2013.

[26] 孙中旭. 基于双目视觉的运动物体测距测速研究[D]. 青岛理工大学, 2014.

[27] 惠记庄，罗丽，杨永奎,等. 基于 SURF-BRISK 的目标识别匹配与定位方法研究[J]. 长安大学学报自然科学版, 2016, 36(3):93-101.

[28] 崔国影. 基于局部特征的显著区域检测及其在图像检索中的应用[D]. 合肥工业大学, 2014.

[29] 佟昊. 基于偏二叉树双支持向量机的遥感图像分类研究与应用[D]. 安徽大学, 2015.

[30] Ghosh D, Kaabouch N, Fevig R A. Robust Spatial-Domain Based Super-Resolution Mosaicing of CubeSat Video Frames: Algorithm and Evaluation[J]. International Journal of Computer & Information Sciences, 2014, 7(2).

[31] 石凯丽. 基于局部不变特征方法的图像匹配算法研究及其应用[D]. 安徽大学, 2016.

[32] Abraham R, Simon P. Review on Mosaicing Techniques in Image Processing[C]. Third International Conference on Advanced Computing & Communication Technologies. IEEE Computer Society, 2013:63-68.

[33] Rosten E, Drummond T. Machine learning for high-speed corner detection[J]. 2006, 3951:430-443.

[34] Zhu J, Ren M. Image mosaic method based on SIFT features of line segment[J]. Computational & Mathematical Methods in Medicine, 2014, 2014(6):926312.

[35] 王博杨，刘海燕. 图像角点检测配准的研究[J]. 价值工程, 2017, 36(4):195-196.

[36] 刘铭. 基于 ORB 算法的双目视觉测量与跟踪研究[D]. 哈尔滨工业大学, 2014.

[37] Rosin P L. Measuring Corner Properties[J]. Computer Vision & Image Understanding, 1999, 73(2):291-307.

[38] Prados R, Garcia R, Neumann L. Image Blending Techniques and their Application in Underwater Mosaicing[M]. Springer International Publishing, 2013.

[39] 张家伟. 光学表面疵病检测系统及图像处理技术研究[D]. 南京理工大学, 2016.

[40] Peris M, Maki A, Martull S, et al. Towards a simulation driven stereo vision system[C]. International Conference on Pattern Recognition. 2012:1038-1042.

中国知网 https://www.cnki.net

[41] Jiang J, Hu R, Zhang F, et al. Experimental Demonstration Of The Fixed-Point Sparse Coding Performance[J]. Cybernetics & Information Technologies, 2014, 14(5):40-50.

[42] Kaehler, Adrian. Learning OpenCV[M]. O'Reilly, 2008.

[43] Levy N. In-camera panorama image stitching assistance[J]. 2016..

[44] Chum O, Matas J. Matching with PROSAC " Progressive Sample Consensus[C]. Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. DBLP, 2005:220-226 vol. 1.

[45] KKimura T. Method and apparatus for processing images acquired by camera mounted in vehicle[J]. 2010.

[46] Ehlgen T, Thom M, Glaser M. Omnidirectional Cameras as Backing-Up Aid[C]. IEEE, International Conference on Computer Vision, ICCV 2007, Rio De Janeiro, Brazil, October. DBLP, 2007:1-5.

[47] Szeliski R. Computer Vision: Algorithms and Applications[M]. Springer-Verlag New York, Inc. 2010.

[48] Kurtulmuş F , AliBaş I , KavdiR I . Classification of pepper seeds using machine vision based on neural network.[J]. International Journal of Agricultural & Biological Engineering, 2016, 9(1):51-62.

[49] 万年红, 王雪蓉. 基于边缘特征点的全景图像拼接算法[J]. 温州大学学报(自然科学版), 2016, 37(1):42-51.

[50] 李晓娟. 图像拼接技术研究[D]. 西安电子科技大学, 2007.

[51] 龚声蓉. 数字图像处理与分析[M]. 清华大学出版社, 2014.

[52] 周镇镇, 孙丰荣, 宋尚玲,等. 冠脉造影图像序列的时空滤波[J]. 计算机应用, 2015, 35(6):1734-1738.

[53] Scherer-Negenborn N, Schaefer R. Model Fitting with Sufficient Random Sample Coverage[J]. International Journal of Computer Vision, 2010, 89(1):120-128.

[54] 陈令羽, 贾奋励, 宋国民. 基于全景图的地理现实增强技术研究[J]. 测绘通报, 2015(4):14-17.

[55] Shen, Junxing. Color matching and color correction for images forming a panoramic image[J]. 2010.

中国知网 https://www.cnki.net

中国知网 https://www.cnki.net

致　谢

时光飞逝，两年的研究生生活即将画上圆满的句号，回忆起这两年学习上的点点滴滴，心中颇多感慨。毕业之时，首先要衷心地感谢我的导师李新教授。李老师为人和蔼，平易近人，在我的工作、学习以及生活都给予了极大的支持和帮助。当我在写论文的过程中遇到了很多困难和障碍时，李老师一直孜孜不倦地指导，使我能够增长见识的同时得到了长足的进步。对李老师这两年来对我的关心的耐心培养深表感谢。

感谢同一个实验室的杨乐乐师姐、王宇宸师弟，当我遇到困难时，是他们给了我鼓励，让我坚持了下来，让我在学习之余体会到了温暖。是他们在学习、科研和生活上给了我极大帮助和关心，与他们共同度过的研究生生活，让我感受到学习的乐趣和生活的快乐，这将会是我一辈子的财富。

最后，我要感谢我的父母对我学业的大力支持和鼓励，有了你们的理解，我才能够有今天的成绩。同时也要感谢所有关心和帮助过我的老师、同学和朋友们。在此还要对审阅本论文的老师表示衷心感谢，能够得到各位老师点评，我深深地感荣幸，欢迎各位老师提出宝贵的意见！

中国知网　https://www.cnki.net