一、实验项目名称：

图片的关键点提取与特征匹配

二、实验原理：

利用 Harris 算法进行关键点提取，并利用 SIFT 算法描述关键点特征，最后实现特征匹配。

（1） 利用 Harris 算法提取关键点

a) 算法原理

当窗口[u,v]发生滑动时，滑动前后对应窗口中的像素点灰度变化为：

$$E(u,v)=\sum_{x,y\in W} w(x,y)[I(x+u,y+v)-I(x,y)]^2$$

** [u,v] 为窗口 W 的偏移量*

** I(x,y) 为像素坐标位置(x,y)的图像灰度值；*

** I(x+u,y+v)为像素坐标位置(x+u,y+v)的图像灰度值；*

** w(x,y)是窗口函数，例如窗口 W 内的所有像素所对应的 w 权重系数均为 1 或以窗口 W 中心为原点的二元正太分布。若窗口 W 中心点是角点时，移动前与移动后，该点在灰度上将变化剧烈*

之后对 E(u,v)进行化简，

$$E(u,v)=\sum_{x,y\in W} w(x,y)[I(x+u,y+v)-I(x,y)]^2$$

$$\approx \sum_{x,y\in W} w(x,y)[I(x,y)+uI_x+vI_y-I(x,y)]^2$$

$$=\sum_{x,y\in W} w(x,y)[u^2I_x^2+2uvI_xI_y+v^2I_y^2]$$

$$=\sum_{x,y\in W} w(x,y)[u \quad v]\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix}$$

$$=[u \quad v](\sum_{x,y\in W} w(x,y)\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix})\begin{bmatrix} u \\ v \end{bmatrix}$$

$$=[u \quad v]M\begin{bmatrix} u \\ v \end{bmatrix}$$

**I_x, I_y 分别为窗口内像素点(x,y)在 x，y 方向上的梯度值*

接下来对每个窗口计算得到一个分数 R，并根据 R 的大小判断窗口内是否存在 Harris 特征角。R 的计算方法如下：

$$R = \det(M) - k(trace(M))^2$$

$$\det(M) = \lambda_1\lambda_2$$

$$trace(M) = \lambda_1 + \lambda_2$$

λ_1, λ_2 为 M 的两个特征值，k 为经验参数

通常来说，角点的$|R|$很大，平坦区域的$|R|$很小，边缘的 R 为负值。

最后通过设定的 R 来对角点进行判断。基于非极大值抑制的原理，在一个窗口内，如果有很多角点则用值最大的那个角点，其他的角点都删除。

b) 具体实现

首先计算每个像素点的梯度值I_x, I_y：

$$I_x = I \times [-1\ 0\ 1]$$

$$I_y = I \times [-1\ 0\ 1]^T$$

接着计算 M 矩阵：

$$A = \sum_{x,y\in W} g\left(I_x^2\right) = \sum_{x,y\in W} I_x^2 \times w(x,y)$$

$$B = \sum_{x,y\in W} g\left(I_y^2\right) = \sum_{x,y\in W} I_y^2 \times w(x,y)$$

$$C = \sum_{x,y\in W} g(I_x I_y) = \sum_{x,y\in W} I_x I_y \times w(x,y)$$

其中$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$

接着对每个窗口计算得到分数 R，根据 R 的大小来判定窗口内是否存在 Harris 特征角。

最后选取分数大于阈值的像素点并基于非极大值抑制的原理删除多余角点即可。

（2） SIFT 算法描述关键点特征

SIFT 通过对关键点周围图像区域分块，计算块内梯度直方图，生成具有独特性的向量。

对于上面选出来的每一个关键点，都拥有三个信息：位置、尺度以及方向。接下来为每个关键点建立一个描述符，也就是生成特征点描述子，使其不随各种变化而改变，并且也含有特征点周围对其有

贡献的像素点。特征描述符的生成大致有三个步骤：

1、校正旋转主方向，确保旋转不变性。

2、生成描述子，最终形成一个 128 维的特征向量

3、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

（3） 特征匹配

由于描述子向量已经归一化，所以可直接用欧氏距离作为相似度的度量，将最短距离的点视为匹配成功。

三、实验目的：

SIFT 特征的具体细节理解和简单的距离计算

四、实验内容：

本次实验直接利用了 VLFeat 库进行。根据官网的介绍，VLFeat 开源库可以实现流行的计算机视觉算法专业图像的理解和局部特征提取和匹配。算法包括 Fisher 向量，VLAD，SIFT，MSER，k 均值，分层 k 均值，聚集信息瓶颈，SLIC 超像素，快速移位超像素，大规模 SVM 训练等。它是用 C 编写的，以提高效率和兼容性，并带有 MATLAB 的接口。

官网文档对于本次需要使用的 vl_sift 函数使用方法介绍如下：

F = VL_SIFT (I)

It computes the SIFT frames (keypoints) F of the image I. I is a gray-scale image in single precision. Each column of F is a feature frame and has the format [X;Y;S;TH], where X,Y is the (fractional) center of the frame, S is the scale and TH is the orientation (in radians).

需要注意的是，vl_sift 函数的输入是一个单精度的灰度图像，因此对于输入的图像 I 要先利用 I = single(rgb2gray(I))转换成相应的格式。

五、实验步骤：

（1）安装 VLFeat 库

a）下载 VLFeat 的安装包在其解压到任意目录下。

b）在 matlab 中新建 startup.m 文件

c）在 startup.m 文件中输入 run('D:\ ...\vlfeat-0.9.18\toolbox\vl_setup')并运行，即可安装

d）可在 matlab 中输入 vl_version ，得到 vlfeat 的版本号以检验是否安装成功

（2）用 VLFeat 实现特征提取与匹配

代码如下：

```
clear
clc
image1=imread('NotreDame1.jpg');
img1=single(rgb2gray(image1));
image2=imread('NotreDame2.jpg');
img2=single(rgb2gray(image2));
[f1,d1]=vl_sift(img1);
[f2,d2]=vl_sift(img2);
[matches,scores]=vl_ubcmatch(d1,d2);
[dump,scoreid]=sort(scores);
newpicture=zeros(size(img1,1),size(img1,2)+size(img2,2),3);
newpicture(:,1:size(img1,2),:)=image1;
newpicture(1:size(img2,1),(size(img1,2)+1):end,:)=image2;
newpicture=uint8(newpicture);
figure(1);
image(newpicture);
axis image
mv=f2;
m=size(img1,2);
mv(1,:)=mv(1,:)+size(img1,2);
vl_plotframe(f1(:,end-200:end),'color','g','linewidth',2);
vl_plotframe(mv(:,end-200:end),'color','r','linewidth',2);
title('特征点');
figure(2);
image(newpicture);
axis image
hold on
plot_rate=0.1;
for i=1:fix(plot_rate*size(matches,2))
      id=scoreid(i);
      line([f1(1,matches(1,id)) mv(1,matches(2,id))],...
            [f1(2,matches(1,id)) mv(2,matches(2,id))],'linewidth',1,'color','b');
end
hold off
title('特征匹配');
```

六、实验数据及结果分析：

对于 data 文件夹中的 NotreDame 和 Mount_Rushmore 进行了实验，结果如下：

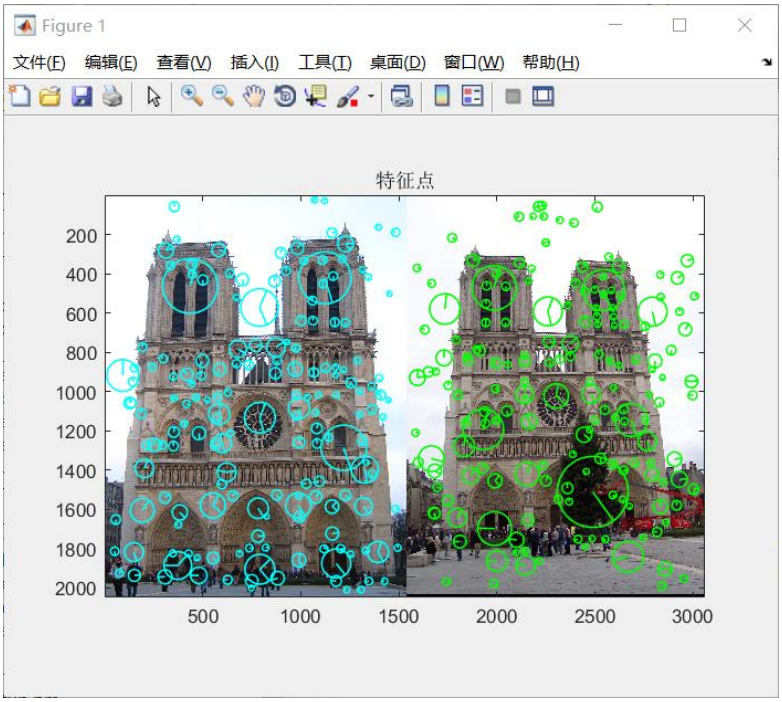

图 1-NotreDame 特征点检测结果

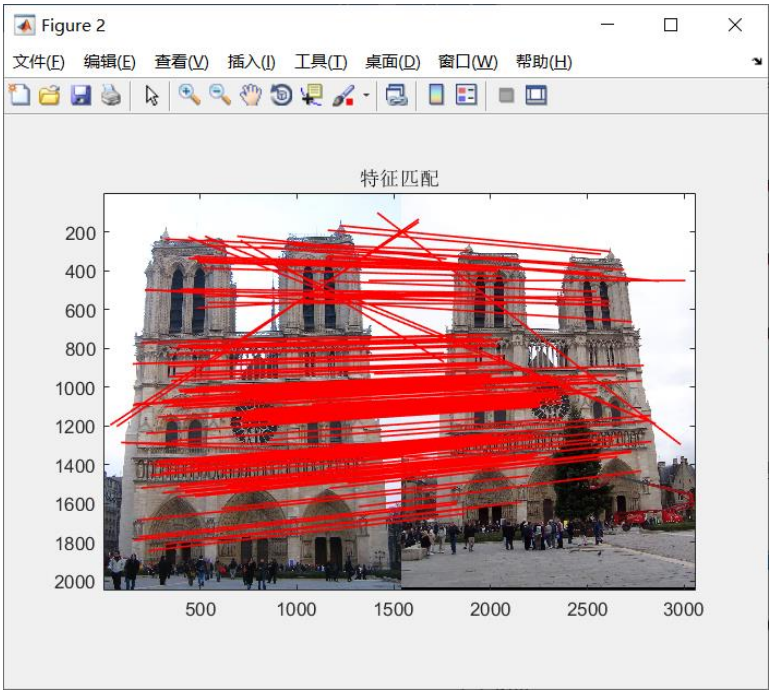

图 2-NotreDame 特征点匹配结果

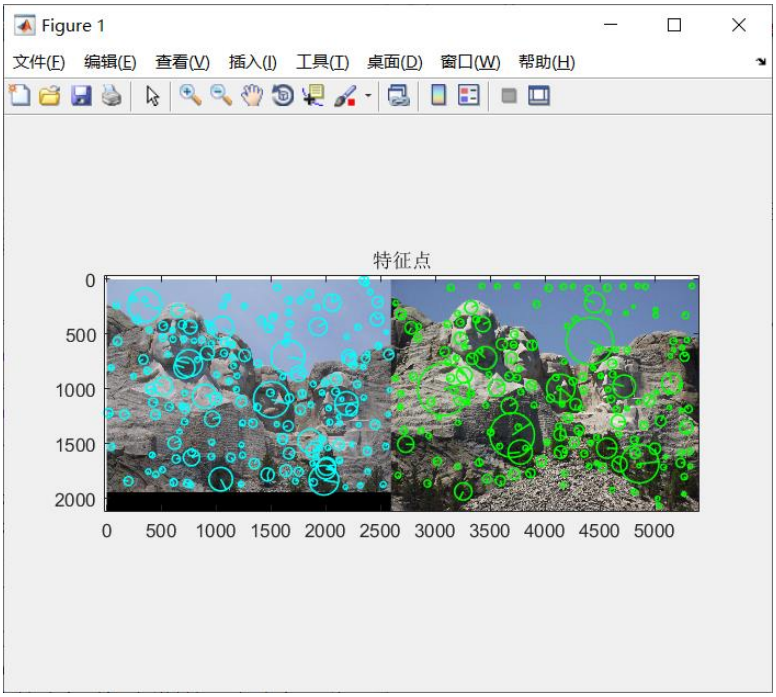

图 3-Mount_Rushmore 特征点检测结果

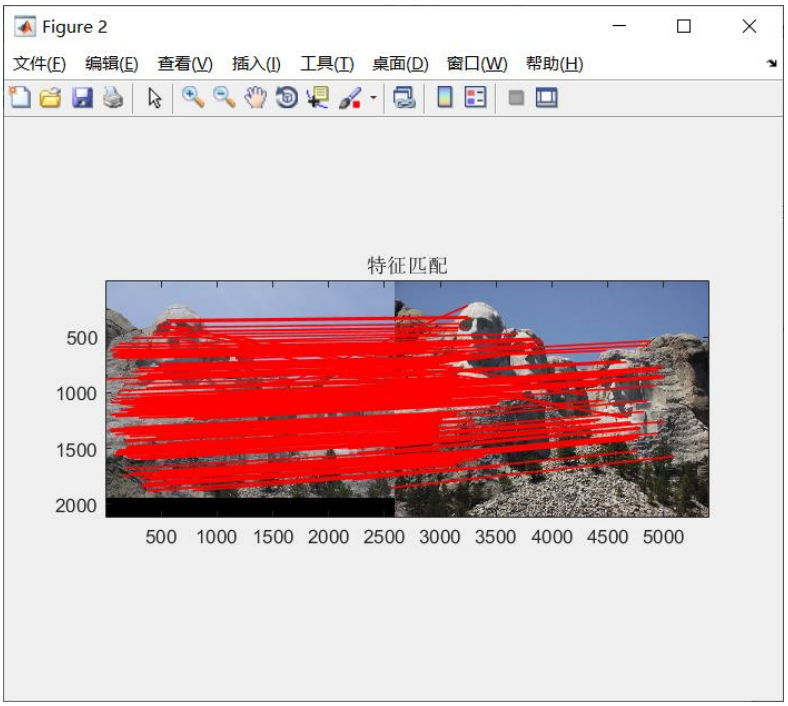

图 4-Mount_Rushmore 特征点匹配结果

可见，两个都实现了特征点的检测和匹配，但是可以看到特征点的检测中存在一些并非特征点的地方也被错误当成特征点，并且存在一些错误匹配的地方。

七、实验结论：

本实验利用 vlfeat 库实现了特征点的检测和匹配，但仍存在一些精度不

高及匹配错误的问题。

九、对本实验过程及方法的改进建议：

（1）在研究 vlfeat 时发现 vlfeat 库中自带了实现 Harris 的函数，文档中对其描述如下：

H = VL_HARRIS（I，SI）

It computes the Harris corner strength of the image I at ``integration" scale SI.
之后可以尝试利用 vl_harris 函数实现 Harris 算法。

（2）除了 SIFT，MSER 也是一种十分有效的特征点检测方法。MSER（Maximally Stable Extremal Regions）即区域特征提取，是一种基于分水岭的思想来做图像中斑点检测的方法。该方法首先使用一系列灰度阈值对图像进行二值化处理，得到相应的黑色区域与白色区域。在比较宽的灰度阈值范围内保持形状稳定的区域称为稳定区域。MSER 的基本原理是对一幅灰度图像（灰度值为 0～255）取阈值进行二值化处理，阈值从 0 到 255 依次递增。阈值的递增类似于分水岭算法中的水面的上升，随着水面的上升，有一些较矮的丘陵会被淹没，如果从天空往下看，则大地分为陆地和水域两个部分，这类似于二值图像。在得到的所有二值图像中，图像中的某些连通区域变化很小，甚至没有变化，则该区域就被称为最大稳定极值区域——MSERs。这类似于当水面持续上升的时候，有些被水淹没的地方的面积没有变化。MSER 对于图像灰度的仿射变化具有不变性和稳定性，并且可以检测不同精细程度的区域。之后可以尝试采用 MSER 方法进行特征点检测，并与 SIFT 方法进行比较。