****

计算机视觉实验

**实验三**

**专 业：**

**班 级：**

**姓 名：**

**学 号：**

**指导教师：**

**成 绩：**

**计算机学院**

**年 月 日**

# 实验目的

Harris角点检测

阅读代码，为代码添加备注，即大致描述每一部分代码分别实现的什么功能

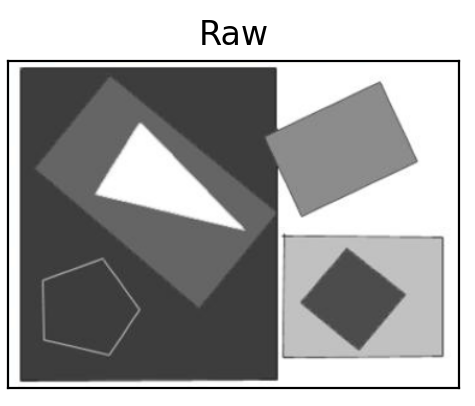
SIFT关键点检测及特征匹配

调试代码，根据代码进一步掌握算法相关步骤

# 实验过程

## Harris角点检测

实验图片



所用代码

import cv2

import numpy as np

import matplotlib.pyplot as plt

if \_\_name\_\_ == "\_\_main\_\_":

# 1. 读取图像文件

img = cv2.imread('corner.png')

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

gray = np.float32(gray)

# 2. Harris角点检测

dst = cv2.cornerHarris(gray, 5, 5, 0.04)

dst = cv2.dilate(dst, None) # 膨胀， 非极大值抑制

thres = 0.01 \* dst.max()

finalImg = img.copy()

finalImg[dst > thres] = [255, 0, 0]

imgs = [img, finalImg]

titles = ['Raw', 'Corner']

for i in range(2):

plt.subplot(1, 2, i + 1)

plt.imshow(imgs[i], 'gray')

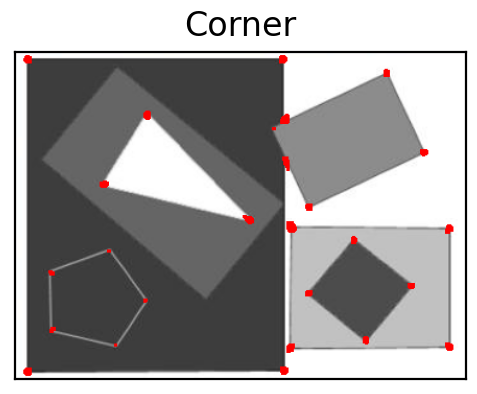
plt.title(titles[i])

plt.xticks([]), plt.yticks([])

# 3. 显示结果

plt.show()

实验效果



代码分析及现象

算法基本思想是使用一个固定窗口在图像上进行任意方向上的滑动，比较滑动前与滑动后两种情况，窗口中的像素灰度变化程度，如果存在任意方向上的滑动，都有着较大灰度变化，那么我们可以认为该窗口中存在角点。

Harris 角点检测是基于图像像素灰度值变化梯度的。灰度值图像的角点附近，是其像素灰度值变化非常大的区域，其梯度也非常大。换句话说，在非角点位置邻域里，各点的像素值变化不大，甚至几乎相等，其梯度相对也比较小。

## SIFT关键点检测及特征匹配

实验图片



所用代码

import cv2

# 1. 读取图像文件

image\_path = 'i1.jpg'

image = cv2.imread(image\_path, cv2.IMREAD\_GRAYSCALE)

# 2. SIFT角点检测与特征描述

detector = cv2.SIFT\_create()

keypoints1, descriptors1 = detector.detectAndCompute(image, None)

# 3. 在图中画出关键点

result\_image1 = cv2.drawKeypoints(image, keypoints1, image, color=(0, 0, 255))

# 5. 仿射变换，将原图旋转给定角度

angle = 15

M = cv2.getRotationMatrix2D(((image.shape[1]-1)/2.0, (image.shape[0]-1)/2.0), angle, 1.0)

rotated\_image = cv2.warpAffine(image, M, (image.shape[1], image.shape[0]))

# 6. SIFT关键点检测

keypoints2, descriptors2 = detector.detectAndCompute(rotated\_image, None)

result\_image2 = cv2.drawKeypoints(rotated\_image, keypoints2, rotated\_image, color=(0, 0, 255))

# 7. BFMatcher：计算匹配图层的一个特征描述子与待匹配图层的所有特征描述子的距离返回最近距离。

bf = cv2.BFMatcher()

matches = bf.knnMatch(descriptors1, descriptors2, k=2)

match\_thresh = 0.1

keep = []

for m, n in matches:

if m.distance <= (match\_thresh \* n.distance):

keep.append([m])

result = cv2.drawMatchesKnn(image, keypoints1, rotated\_image, keypoints2, keep, None, flags=2)

# 8. 显示结果

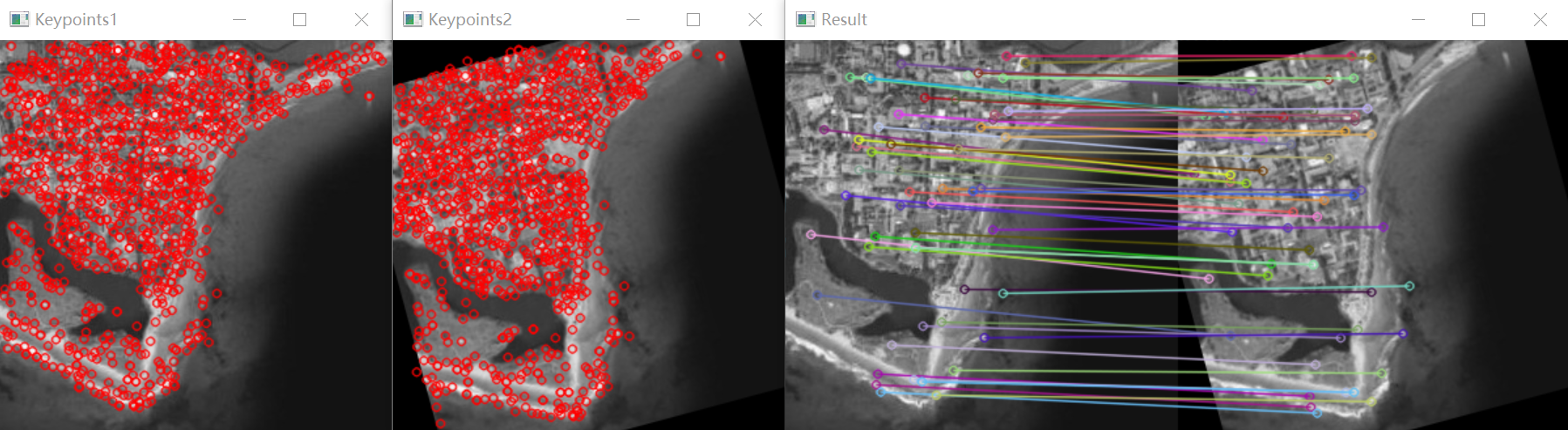
cv2.imshow('Keypoints1', result\_image1)

cv2.imshow('Keypoints2', result\_image2)

cv2.imshow('Result', result)

cv2.waitKey(0)

实验效果



代码分析及现象

SIFT算子是把图像中检测到的特征点用一个128维的特征向量进行描述，因此一幅图像经过SIFT算法后表示为一个128维的特征向量集，该特征向量集具有对图像缩放，平移，旋转不变的特征，对于光照、仿射和投影变换也有一定的不变性，是一种非常优秀的局部特征描述算法。实现步骤为：1.检测尺度空间极值点 2.精确定位极值点 3.为每个关键点指定方向参数 4.关键点描述子的生成

# 实验心得

角点就是极值点，即在某方面属性特别突出的点。一般的角点检测都是对有具体定义的、或者是能够具体检测出来的兴趣点的检测。这意味着兴趣点可以是角点，是在某些属性上强度最大或者最小的孤立点、线段的终点，或者是曲线上局部曲率最大的点。

SIFT即尺度不变特征转换，它用来检测图像的局部性特征，在空间尺度中寻找极值点，提取这点的位置、尺度、旋转不变量。这些关键点是一些十分突出，不会因光照和噪音等因素而变化的点，如角点、边缘点、暗区的亮点及亮区的暗点等，所以与影像的大小和旋转无关，对光线、噪声、视角改变的容忍度也很高。