

# 计算机视觉报告：SIFT 特征提取与匹配

*Technical Report on Computer Vision Programming Assignment: SIFT detection and matching*

---

叶翰樵

中国科学院大学-人工智能学院-202218020629017

中国科学院自动化研究所

[yehanqiao22@mails.ucas.ac.cn](mailto:yehanqiao22@mails.ucas.ac.cn)

特征提取是计算机视觉底层任务中最具受关注的基础算法，在其后续进行的特征描述、匹配与之共同构成了一整套复杂而又清晰的处理流程，为物体识别、图像匹配等任务提供了解决思路。尺度不变特征变换（Scale-Invariant Feature Transform, SIFT）是迄今使用最为广泛的一种特征提取算法，其能够在不同尺度和旋转角度下检测到具有独特性质的关键点，并进行鲁棒、高效的特征描述和匹配。

此份报告围绕 SIFT 特征提取与匹配，首先在 *Part I* 介绍高斯拉普拉斯核的空间尺度选择特性，以及图像尺度空间这两个重要概念，作为后续算法的铺垫；*Part II* 则具体描述了 SIFT 特征提取的基本流程，其中包括高斯差分金字塔构建以及关键点定位两个关键步骤；*Part III* 介绍 SIFT 特征描述子；最后，在 *Part IV* 进一步解释多视角图像间特征描述子匹配的过程，并通过个人代码实现展示了 SIFT 特征提取与匹配基础流程的最终效果。

---

## Contents

<b>I Preliminaries</b>	<b>1</b>
1 LoG 的空间选择特性	1
2 图像尺度空间	3
<b>II Keypoint Detection</b>	<b>4</b>
3 构造高斯差分金字塔	4
4 定位关键点	6
<b>III Keypoint Descriptor</b>	<b>6</b>
5 主方向分配	6
6 特征描述子	7
<b>IV Using SIFT for Image Matching</b>	<b>8</b>
7 描述子匹配	8
8 实验：图像 SIFT 特征匹配	9
8.1 代码库介绍	9
8.2 提取特征点	10
8.3 特征点匹配	10
<b>V Summary</b>	<b>11</b>
参考文献	12

---

# Preliminaries

对图像底层的局部特征进行检测和描述是解决很多计算机视觉问题的基础，例如物体识别、图像匹配和图像复原。在对这些局部特征进行匹配后，算法就能够对不同视角图像当中的特殊区域进行识别和比较。

在图像底层局部特征提取的诸多算法当中，SIFT 是迄今使用最为广泛的一种特征，它具有以下优点：

- 旋转不变性 (*rotation invariance*) 和尺度不变性 (*scale invariance*)：对图像的旋转和尺度变化具有不变性；
- 对三维视角变化、光照变化以及噪声具有很强的适应性；
- 在存在遮挡或场景杂乱的情况下，底层的局部特征具有不变性；
- 辨别力强：特征之间相互区分的能力强，有利于匹配；
- 扩展性强：能够与其他形式的特征向量联合
- 易获取：一般  $500 \times 500$  的图像能提取出约 2000 个特征点

下面对 SIFT 背后的原理进行简单论述。

Section 1

## LoG 的空间选择特性

回忆当我们在进行边缘检测时，会使用高斯偏导（Derivative of Gaussian）对图像做卷积，得到的图像与在原图像基础上先后进行高斯平滑与求导两个操作等价，最终响应值最高处即被确定为边缘所在位置。

其实，高斯二阶导也可用于边缘检测。高斯二阶导也被叫做拉普拉斯核（Laplacian of Gaussian, LoG），与之卷积所获得的图像与在原图像基础上先后进行高斯平滑与连续求二阶导两个操作等价，最终响应值过零点处即被确定为边缘所在位置。

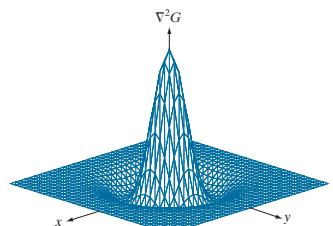


图 1. 3-D plot of the negative of the LoG.

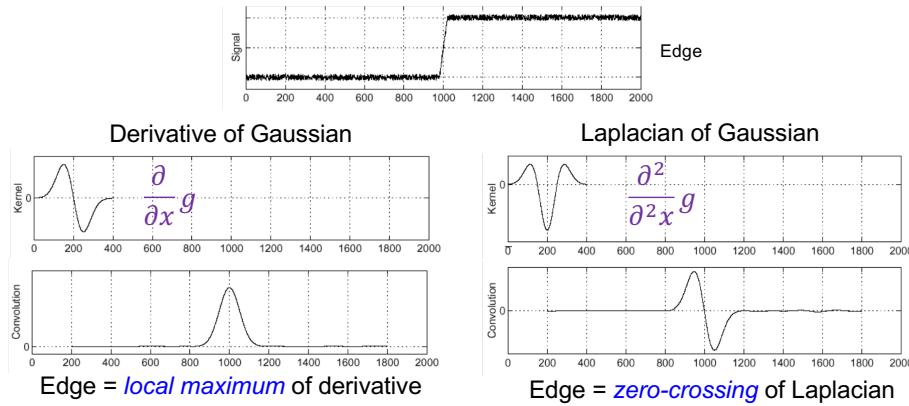


图 2. 高斯偏导和 LoG 分别进行边缘检测 [1]。

不仅如此，LoG 相比于高斯偏导还拥有一个更好的特性，即具有空间选择特性 (*Spatial Selection*)：当 LoG 的尺度  $\sigma$  与信号的宽度匹配时，两者卷积会在原信号的中心位置有最大响应。

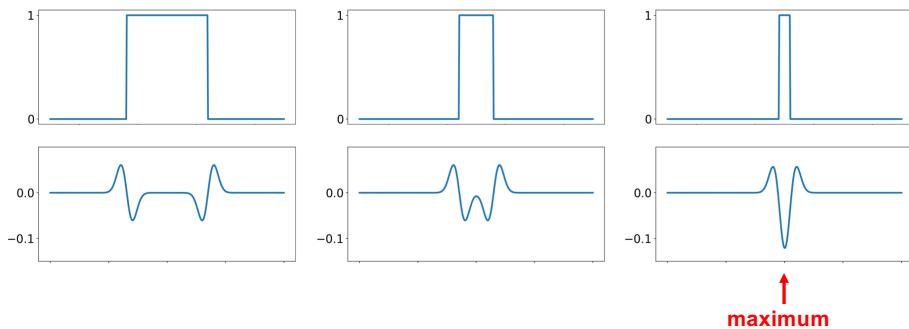


图 3. LoG 具有空间选择特性。

因此容易想到由 LoG 的空间选择特性，用一组尺度不一的 LoG 分别与某一尺度未知信号做卷积，通过判断响应值的大小最终确定该信号的尺度。而响应值最大时 LoG 与信号在尺度上到底存在什么数量关系呢？

#### Intuition 1

当 LoG 的过零点刚好与信号卡住的时候，会在信号的中心点产生最大响应。于是，令

$$\nabla_{\text{norm}}^2 g = 0$$

得到  $x^2 + y^2 = 2\sigma^2$ ，即要想产生最大响应，信号的半径  $r$  应等于 LoG 过零点围成的半径  $\sqrt{2}\sigma$ 。换句话说，与半径为  $r$  的信号产生最大响应的 LoG 参数  $\sigma$  应当满足： $\sigma = r/\sqrt{2}$ 。

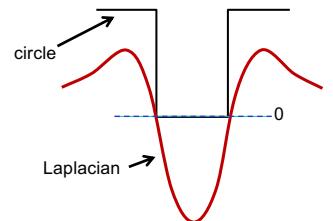


图 4. LoG 的过零点刚好与信号卡住的情形。

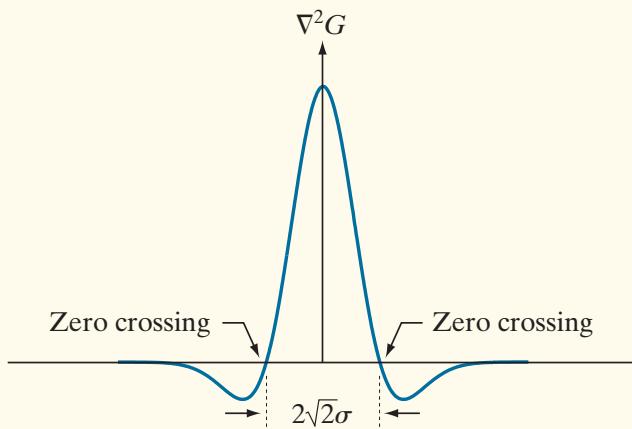


图 5. LoG 过零点

直觉上看，信号的尺度和与之产生最大响应值的 LoG 的  $\sigma$  存在上述关系。但事实上，卷积的响应值会随着尺度  $\sigma$  的增大不断衰减，因此在此之前还必须对响应值进行尺度补偿（scale normalize）：

$$\nabla_{\text{normalized}}^2 g = \sigma^2 \left( \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \right) \quad (1.1)$$

## Section 2

# 图像尺度空间

对于图像信号，如果希望找到图像上以某一点为中心所在局部特征的尺度  $r$ ，就可以使用一组尺度 ( $\sigma$ ) 不同的归一化 LoG 分别对图像做卷积。图 6 表示了一组用多尺度 LoG 相卷积得到的图像尺度空间，在纵向找到该位置处响应值最大时的 LoG，然后根据其参数  $\sigma$  即可计算局部特征的尺度  $r = \sqrt{2}\sigma$ 。

要对图像提取局部特征时，就对每一个像素坐标进行上面的操作。但是实际中并不会将 LoG 组的尺度划分得过细以逼近响应极值，这样将造成过大的计算消耗，同时意味着每一个像素坐标只能对应一个局部特征。因此，通常每三个 LoG 尺度进行比较，如果中间的响应值大于上下，则认为中间这个位置为响应极值，从而可确定出其所对应局部特征的尺寸。

然后横向在响应极值所在的尺度平面上看，周围像素其对应的响应也可能是极值，这也就导致了图像中局部特征可能会大量重叠。因此还需要进行非极大值抑制，其思路就是在确定响应极值时，不仅纵向跟自己位置进行比较，还需要跟图像尺度空间当中周围的 26 个像素相比，如果该响应值仍然最大，则保留这一处这一尺度的局部特征，否则遗弃该位置。

至此介绍了在图像尺度空间进行尺度搜索和非极大值抑制两个关键环节。但是上述过程仍然需要进行大量计算，为此 [2] 提出了 Harris-Laplacian 方法，先找出图像当中所有的 Harris 角点，然后只在这些点周围建立尺度空间，进行 LoG 的尺度分析。Harris

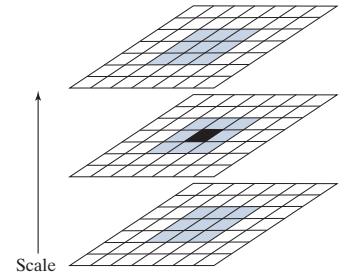


图 6. 图像尺度空间特征检测

# Keypoint Detection

正如上一章所讲，因为我们希望提取到的局部特征具有尺度不变性，所以既要找局部特征所在位置，还要确定它们的尺度。LoG 已经被说明能够很好地完成上述任务，但它还不是最优的。本章将介绍 SIFT 是如何优化的，以及程序在实现关键点检测时的一些细节。

Section 3

## 构造高斯差分金字塔

SIFT 注意到一个明显的问题是当 LoG 尺度越来越大  $\sigma$  时，卷积窗口  $w$  也会越来越大（一般认为  $w \approx 2 \times \lceil 3\sigma \rceil + 1$ ），计算消耗也会越来越明显。为了解决这一问题，SIFT 指出可以采用高斯差分（Difference of Gaussian, DoG）完成如下近似：

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1) \sigma^2 \nabla^2 G \quad (3.1)$$

↑ DoG                                   ↑ scale-normalized LoG

上式表明，在不考虑尺度因子  $k-1$  的情况下，DoG 约等于尺度归一化后的 LoG。利用 DoG 近似的好处是，由高斯卷积的性质，大的高斯核 ( $G(x, y, k\sigma)$ ) 可以拆成两个小高斯核的卷积（定理 1），即要想获得方差更大高斯滤波，只需要在小方差高斯滤波的结果上继续进行一个小方差高斯滤波即可。

**Theorem 1**

用两个方差分别为  $\sigma_1^2$  和  $\sigma_2^2$  的高斯核与图像进行连续的卷积滤波操作，该过程等价于直接用一个方差为  $\sigma_1^2 + \sigma_2^2$  的高斯核与图像进行一次卷积滤波操作。即：

$$G(x, y, \sigma_1^2 + \sigma_2^2) = G(x, y, \sigma_1^2) * G(x, y, \sigma_2^2)$$

于是  $G(x, y, k\sigma)$  可以表示为：

$$G(x, y, k\sigma) = G(x, y, \sqrt{k^2 - 1}\sigma) * G(x, y, \sigma) \quad (3.2)$$

这是提升 DoG 图像金字塔创建效率的有效方法。

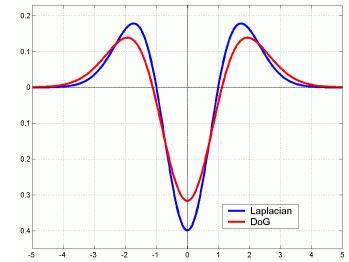


图 7. Approximate LoG with a difference of Gaussians

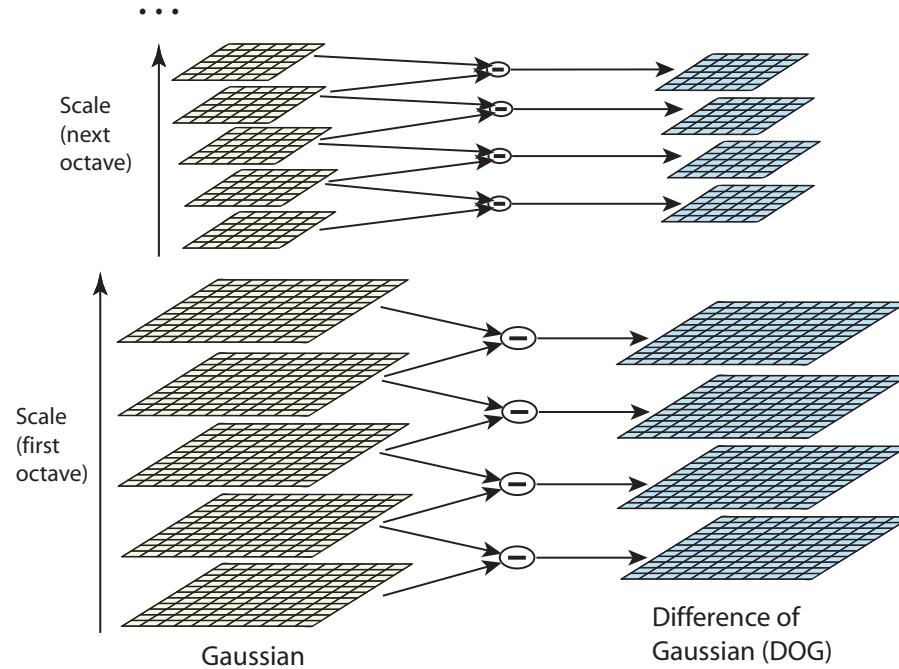


图 8. DoG 图像金字塔

DoG 图像金字塔的结构如图 8，它基本由如下三个参数确定：

- 每一个 *octave* 中包含的尺度个数  $S$ ；
- *octave* 的个数  $O$ ；
- 标准差基本值  $\sigma_0$ 。

如图 8 所示，在每一个 *octave* 的高斯金字塔（左侧）内一共需要进行  $S + 3$  次高斯操作，高斯核的标准差从下往上依次扩大  $k = 2^{1/S}$  倍（这样设置能够确保待检测的尺度在每个 *octave* 之间刚好连续），相邻两层相减就得到了右侧的 DoG。通过直接对一个 *octave* 的高斯金字塔的第  $S$  层进行两倍下采样，获得下一个 *octave* 的最底层。下面是构建 DoG 图像金字塔的步骤：

1. 加载灰度图像，对图像进行灰度归一化，也可缩小图像以加快处理效率；
2. 对于索引  $s = [0, 1, \dots, S + 2]$ ，用标准差  $k^s \sigma_0$  对图像进行高斯模糊。这里可以采用式 (3.2) 介绍的方法提高效率，每一层直接在上一层的基础上进行模糊；
3. 相邻的两张高斯模糊图像做差得到 DoG，每三层 DoG 就可以叠放成图 6 所示的 3D 张量；
4. 重复 2-3 步以最终得到  $O$  个 *octave*。后面每个 *octave* 的最底层高斯模糊图像可以直接从上一 *octave* 的第  $S$  层经两倍下采样获得。

## 定位关键点

在上一章第 2 节中提到，关键点即在 DoG 尺度空间比它的领域都要大的极值点。搜索关键点的过程同时也需要进行非极大值抑制，因为图像中的相邻像素在 DoG 中的响应很可能相同，这会对后续特征点匹配的稳定性造成影响。同时还可以设置阈值，去除对比度低的点，因为这些点可能来自噪声。下面是程序中搜索定位关键点的步骤：

1. 抑制所有响应小于阈值  $C$  的点；
2. 将每个  $octave$  中的  $S + 2$  个 DoG 叠成一个  $H \times W \times (S + 2)$  的张量，比较张量中每一个位置和它  $26(8 + 9 + 9)$  个邻域之间的大小关系。如果一个位置比它邻域都要大，则它是一个关键点。忽略 DoG 金字塔最底层和最高层中检测到的关键点，即只在张量的  $[,:,1] \sim [,:,S]$  部分提取关键点。
3. 对每个  $octave$  重复上述过程。

PART

III

## *Keypoint Descriptor*

在计算机视觉中，为了进行图像处理、目标检测、图像匹配等任务，需要从图像中提取出能够代表图像内容的特征。关键点描述就是这些特征之一，它能够描述图像中的局部特征，通常具有不变性、区分性和可重复性等优良性质，因此在图像匹配和目标识别等任务中被广泛应用。

对于某个关键点，其描述子可以看作是对该点周围像素的一种统计特征表达，它可以用于描述该点的形状、颜色、纹理等信息。在构建关键点描述子时，需要保证其具有一定的不变性，即对于图像的旋转、缩放、平移、光照等变换具有一定的容忍度，这样才能在实际应用中获得更好的效果。因此，SIFT 算法采用了一些特殊的技巧来确保描述子的不变性，如使用 DoG 金字塔来实现尺度不变性，使用主方向来实现旋转不变性等。前面部分介绍了如何构建 DoG 金字塔并从中提取关键点，这一部分介绍主方向分配，以及构建具有旋转不变性的关键点描述子。

## 主方向分配

在 SIFT 算法中，主方向被定义为是指关键点周围像素的梯度方向直方图中的最大峰值所对应的方向。在找到这个主方向之后，通过对关键点周围的图像区域进行旋转的方式，将该主方向变换到统一的标准方向。该过程的具体步骤如下：

- 计算梯度幅值和方向：首先，以每个 SIFT 特征关键点为中心提取出  $16 \times 16$  的局部图像块，然后计算其中每个像素的梯度幅值和方向。这可以通过对图像进行梯度运算来实现。通常使用 Sobel 算子或 LoG 算子来计算图像的梯度。对于像素点  $(x, y)$ ，其梯度的幅值和方向分别为：

$$m(x, y) = \sqrt{g_x^2(x, y) + g_y^2(x, y)} \quad (5.1)$$

$$\theta(x, y) = \arctan \frac{g_y(x, y)}{g_x(x, y)} \quad (5.2)$$

其中， $g_x(x, y)$  和  $g_y(x, y)$  分别表示像素点  $(x, y)$  处的水平和垂直方向梯度。需要注意的是，在计算梯度方向时，可以将其划分为 8 个方向（即 0 度、45 度、90 度、135 度、180 度、225 度、270 度、315 度）。

- 构建方向直方图：接下来，需要将局部图像块的每个像素的梯度按方向分配到 8 个区间中，统计每个方向区间内的梯度幅值，最后形成一个长度为 8 的方向直方图。具体来说，对于像素点  $(x, y)$ ，可以根据其梯度方向  $\theta(x, y)$  将其梯度幅值  $m(x, y)$  分配到相应的方向区间中。如果  $\theta(x, y)$  位于第  $i$  个方向区间内，则该区间的计数器  $bin_i$  需要增加  $m(x, y)$  的值。这样，就可以得到一个长度为 8 的方向直方图：

$$H = [bin_1, bin_2, \dots, bin_8]$$

- 寻找主方向：在得到关键点周围的方向直方图后，需要在其中寻找主方向。通常采用最大值法来确定主方向，即在方向直方图中找到最大的那个值，并将其对应的方向作为主方向。同时，可以考虑一些容错机制，例如将大于最大值一半的值也视为主方向，以确保算法的稳定性。

## Section 6

# 特征描述子

---

经过主方向分配后，各幅图像对应特征子周围像素之间的旋转角度不确定性得以消除，所以到目前为止 SIFT 特征已经解决了尺度问题、旋转问题。对于光照变化，如果直接用 SIFT 关键点周围的像素直接做比较是明显不合理的，因此 SIFT 进一步提出了一种更为鲁棒的描述子，即用一个 128 维向量为关键点进行特征编码，而编码过程考虑的是关键点周围像素的梯度赋值和方向。具体实现过程如下：

- 旋转每个 SIFT 关键点周围像素，使其主方向能与指定的标准方向对齐，并重新提取以每个 SIFT 特征关键点为中心的  $16 \times 16$  的局部图像块；
- 计算局部图像块每个像素上的梯度幅值与方向。同时按照到中心点的距离对梯度幅值进行高斯加权，高斯函数的标准差按经验一般取  $\sigma_w = 1.5 \times 16$ ；
- 将  $16 \times 16$  的局部图像块再划分为 16 个  $4 \times 4$  的子块，然后在每个子块中统计一个长度为 8 的梯度方向直方图，每一个方向的权重为上一步高斯加权后的梯度幅值；

4. 将这 16 个长度为 8 的梯度方向直方图拼成一个 128 维向量，这个高维向量经过归一化就获得了 SIFT 特征描述子，它对线性光照变化具有鲁棒性。

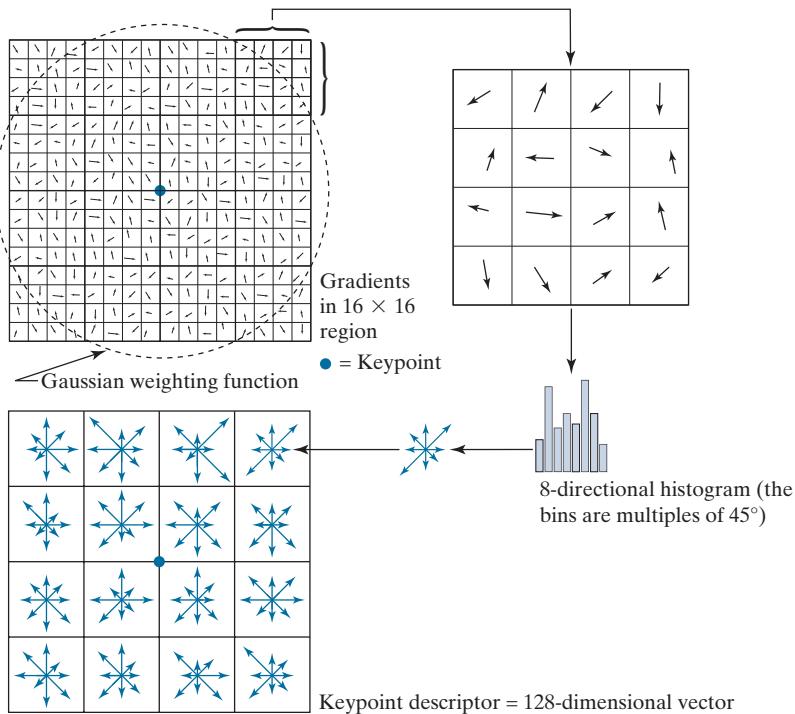


图 9. SIFT 特征描述子

*Remark* 对比确定主方向的过程，同样是构建梯度方向的直方图，但在特征描述子当中是需要对 16 个子块分别构建，并且梯度幅值还被进行了高斯加权。划分子块的方式可以保留更精细的局部细节。

## Using SIFT for Image Matching

PART

IV

基于前面提取 SIFT 特征点、描述 SIFT 特征点的过程，本部分进一步介绍如何进行多视角图像间特征点的匹配，并展示一个最终的实验结果。

Section 7

### 描述子匹配

寻找一幅图像中某个特征点在另一幅图像中的对应，即寻找与之描述子最为接近的特征点。衡量两个描述子（即特征向量）之间距离的方式有很多，例如欧氏距离（2-范数），

汉明距离等。这里，可直接使用欧氏距离，并且以暴力匹配的方式寻找描述子之间的匹配关系。

同时，不一定一幅图像中的每一个描述子都能在另一幅图像中找到对应点，因此可以对描述子之间的距离设置一个上限；并且可以每次选取距离最近的两个描述子，如果最接近的描述子距离上要明显小于次接近的描述子，则可认为最接近的描述子是可靠的。

#### Section 8

## 实验：图像 SIFT 特征匹配

---

该部分大致介绍代码框架及代码实现，并通过从命令行执行脚本的方式，对 SIFT 特征点提取、图像 SIFT 特征点匹配算法进行了实验验证。实验平台 **NVIDIA® Jetson AGX Orin™ Developer Kit, CUDA 11.4**，实验图像分辨率  $760 \times 570$ 。

#### Subsection 8.1

### 代码库介绍

---

目前使用 **Python** 实现了前面介绍的关于 SIFT 特征提取、描述与匹配等基本方法。代码参考自苏黎世大学 (UZH) 2022 年秋季移动机器人视觉算法课程<sup>1</sup>，在其提供代码的基础上，对构造高斯金字塔、生成特征描述子等过程均进行了优化。算法被封装在 **sift** 目录下，形成一个 **Python** 包，可以在其他脚本文件中通过 **import** 调用其所实现的 SIFT 特征提取和匹配方法。

**sift** 目录下各文件介绍：

```
.  
├── __init__.py # 包 (package) 标识文件  
├── descriptor.py # 计算特征描述子  
└── mysift.py # 匹配描述子  
└── pyramid.py # 构建DoG Pyramid，并提取SIFT特征点
```

环境安装说明：

```
# 创建conda虚拟环境  
conda create -n mysift python=3.8 numpy  
# 进入conda虚拟环境  
conda activate mysift  
# 安装opencv  
pip install opencv-python
```

<sup>1</sup>课程网站：<https://rpg.ifi.uzh.ch/teaching.html>

Subsection 8.2

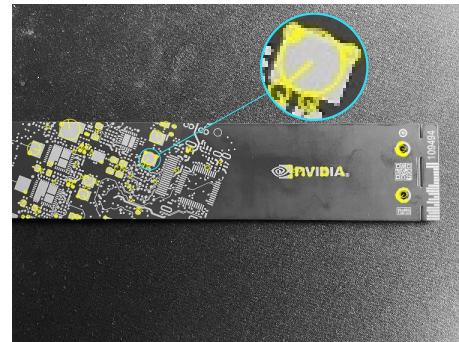
## 提取特征点

输入一幅图像，提取其中的 SIFT 关键点。脚本运行命令示例：

```
python sift_detect.py \
    --file_dir images/nvidia-3.jpg \ # input image
    -o 5 \ # number of octaves
    -s 3 \ # number of scales per octave
    --rescale 0.3 \ # rescale images to make it faster
    --sigma 1.0 \ # use for Gaussian blurring
    --output_dir images/experiment/nvidia-3-keypoints.jpg
    \ # output destination
    -t 5e-2 # threshold for detection
```



(a) 输入的原图像



(b) 提取到的 SIFT 特征点

图 10. SIFT 特征提取结果。观察提取到的关键点，发现它们大多位于角点附近。平均用时 0.1s。

Subsection 8.3

## 特征点匹配

输入两幅图像，分别对它们进行 SIFT 特征提取与描述，并在两组描述子之间进行匹配。  
脚本运行命令示例：

```
python image_matching.py \
    --file_dir_1 images/nvidia-3.jpg \ # input image 1
    --file_dir_2 images/nvidia-4.jpg \ # input image 2
    -o 5 \ # number of octaves
    -s 3 \ # number of scales per octave
    --rescale 0.3 \
    --output_dir images/experiment/image_matching.pdf \ #
    \ output destination
```

```
--sigma 1.0 \ # use for Gaussian blurring
-t 0.05 # threshold for detection
```

结果：

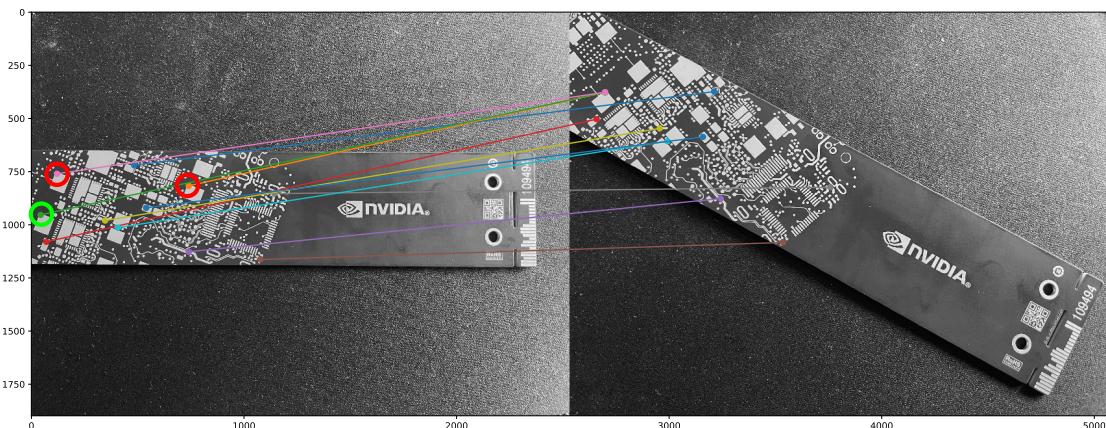


图 11. 特征点匹配。右侧的查询图像相比左侧的数据库图像，发生了尺度、旋转上的变化。一共得到 9 对匹配点，其中错误的匹配有 2 对（红圈标注），匹配正确率 77.8%。平均用时 0.3s。

上述实验中一共获得了 9 对匹配点，其中错误的匹配有 2 对（红圈所示），均重复匹配到了另一位置上的特征（绿圈所示）。其原因在于该三处特征的确十分近似，仅仅依靠局部特征将存在定位歧义性。可见 SIFT 无法对重复的纹理特征进行有效匹配。

## Summary

PART

V

SIFT 是一种在计算机视觉领域中传统而又常用的图像处理算法。SIFT 算法的主要意义在于其能够有效地提取图像中的特征，并通过特征匹配来实现图像的识别、匹配、定位等应用。最后再总结一下其最为基本的五大步骤：

**1. 构建尺度空间** LoG 所具有的空间选择特性为特征的尺度分析提供了方法。SIFT 当中用高斯差分近似 LoG 以高效构建尺度空间（式 (3.1)）。其具体过程首先构建了高斯金字塔，然后在每个尺度上，通过相邻两个高斯模糊后的图像相减，得到差分图像。将每个尺度的差分金字塔中相同位置的差分图像组合成一个 3D 张量，构成尺度空间。每个张量代表了在一个特定尺度下图像的局部特征。在构建尺度空间的过程中，*octave* 的层数  $o$ ，每层 *octave* 包含的尺度个数  $S$ ，以及高斯差分函数的  $\sigma$  参数是三个重要的经验参数，它们决定了所能检测到的尺度范围，以及每个尺度之间的差异程度。为了保证图像的尺度不变性，它们的值需要根据特征尺度进行自适应调整，以便提取到不同尺度下的关键点。

**2. 定位关键点** 定位关键点的过程就是进行非极大值抑制的过程，即既要保证特征在某个高斯差分上的响应超过阈值，还要保证其在 DoG 尺度空间比它的领域都要

大。首先，对每个尺度的差分金字塔中的每个像素，如果其响应值超过了阈值，则继续检查其周围  $3 \times 3 \times 3$  邻域内的 26 个像素，确定其是否为该邻域的极值点。对于每个尺度的差分金字塔，将极值点的坐标和所处的尺度记录下来，形成关键点集合。

3. **主方向分配** 分析以关键点为中心的局部图像块的梯度主方向。图像的梯度分析对光照变化更为鲁棒，而梯度主方向的确定可以消除特征的旋转不确定性。按式(5.2)方式计算以每个 SIFT 特征关键点为中心提取出  $16 \times 16$  的局部图像块的梯度。接下来通过构建梯度方向直方图的方式确定主方向。在找到这个主方向之后，通过对关键点周围的图像区域进行旋转的方式，将该主方向变换到统一的标准方向。
4. **特征描述** 每个关键点生成一个具有区分度和鲁棒性的特征描述符，用于表示关键点周围的图像信息。首先将关键点所在的尺度空间的图像分为  $4 \times 4 = 16$  个大小相等的子区域，为每个子区域构建长度为 8 的梯度方向直方图。将这 16 个梯度方向直方图拼接在一起后进行归一化，得到最终的 128 维特征向量。
5. **描述子匹配** 通过比较两张图像的特征描述符，找出相似的特征点，实现图像间的匹配和识别。可直接使用欧氏距离，并且以暴力匹配的方式寻找描述子之间的匹配关系。

实验结果表明，所实现的 SIFT 算法的特征匹配正确率 77.8%，能够较为鲁棒地解决尺度、旋转等变化条件下的特征匹配任务。另外，实验中出现的两对错误匹配也表明，仅仅依靠局部特征描述无法区分重复的纹理特征。

可以看出，SIFT 特征提取与匹配算法有着清晰而又复杂的处理流程，并且依靠许多经验参数，但这还只是最为基础的一种实现方式。将来在此基础还可以做出如下改进：

- **关键点的精确定位** 在离散采样中搜索到的极值点不一定是真实空间的极值点，因此对尺度空间 DoG 函数进行泰勒展开，在近似连续域上计算其极值点，从而实现关键点的精确定位。
- **增加关键点对视角的鲁棒性** 对视角变化具有一定的鲁棒性，但是在大视角变化下其性能会受到影响。由于透视变换的作用，在数据库图像上原本接近圆形的特征会在查询图像上畸变为椭圆，从而导致无法匹配。[\[4\]](#) 中提出了基于二阶协方差矩阵的仿射不变特征点检测算法。
- **去除边缘上的点** 由于 DoG 对图像中的边缘有比较强的响应值，而一旦特征点落在图像的边缘上，这些点就是不稳定的点。它们很难定位，具有定位歧义性，且容易受到噪声的干扰而变得不稳定。根据主曲率比值来判断关键点是否为稳定的特征点，如果不稳定则删除。

## 参考文献

- [1] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

- [2] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 1. IEEE, 2001, pp. 525–531.
- [3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [4] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, pp. 63–86, 2004.