

问题

在我的一个传统图像处理项目中用到了图像配准技术，太久都忘了，为了防止面试被问到答不上来，这里还是要简要总结下。关于图像配准的概念，在另一个问题“13_图像拼接原理介绍”中也大体上介绍了一下，不过没那么详细。

随着技术的发展，图像配准已经有了深度学习的方法，但是我们这里讨论的还是传统的基于特征的方法。

图像配准流程

假设我们要对一张参考图像和一张待配准图像之间进行图像配准，主要基于三个步骤：**关键点检测和特征描述**，**特征匹配**，**图像变形**。简而言之，我们在两幅图像中选择兴趣点，将参考图像中的每个兴趣点和它在待配准图像中的对应点关联起来，然后对待批准图像进行变换，这样两幅图像就得以对齐。

关键点检测和特征描述

关键点就是感兴趣的点。它定义了一幅图像中重要并且有特点的地方（如角，边等）。每个关键点都由一个描述子（包含关键点本质特点的特征向量）表征。描述子应该对图像变换（如位置变换、缩放变换、亮度变换等）是鲁棒的。很多算法都要执行关键点检测和特征描述，主流的关键点检测算法有：

SIFT (Scale-invariant feature transform, 尺度不变的特征变换) 是用于关键点检测的原始算法，但是它并不能免费地被用于商业用途。SIFT 特征描述子对均衡的缩放，方向、亮度变化是保持不变的，对仿射形变也是部分不变的。SURF (Speeded Up Robust Features, 加速鲁棒特征) 是受到 SIFT 深刻启发设计的检测器和描述子。与 SIFT 相比，它的运行速度要快好几倍。当然，它也是受专利保护的。ORB (定向的 FAST 和旋转的 BRIEF) 是基于 FAST (Features from Accelerated Segment Test) 关键点检测器和 BRIEF (Binary robust independent elementary features) 描述子的组合的快速二值描述子，具有旋转不变性和对噪声的鲁棒性。它是由 OpenCV Lab 开发的高效、免费的 SIFT 替代方案。AKAZE(Accelerated-KAZE) 是 KAZE 的加速版本。它为非线性尺度空间提出了一种快速多尺度的特征检测和描述方法。它对于缩放和旋转也是具有不变性的，可以免费使用。

下面来介绍下 SIFT (Scale Invariant Feature Transform) 即尺度不变特征转换匹配算法。

SIFT

SIFT算法的实质是在不同的尺度空间上查找关键点(特征点)，并计算出关键点的方向。SIFT所查找到的关键点是一些十分突出，不会因光照，仿射变换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。

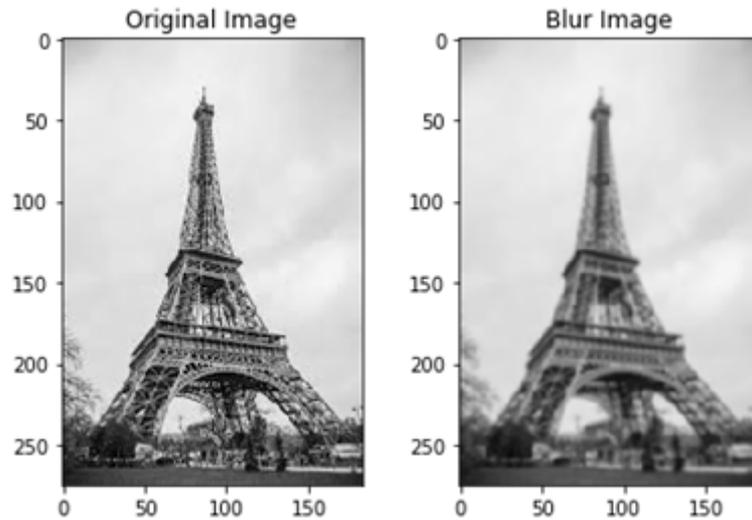
广义上讲，整个过程可以分为四个部分：

1. **创建比例空间**：确保要素与比例无关
2. **关键点本地化**：确定合适的特征或关键点
3. **方向分配**：确保关键点是角度不变
4. **关键点描述符**：为每个关键点分配唯一的指纹

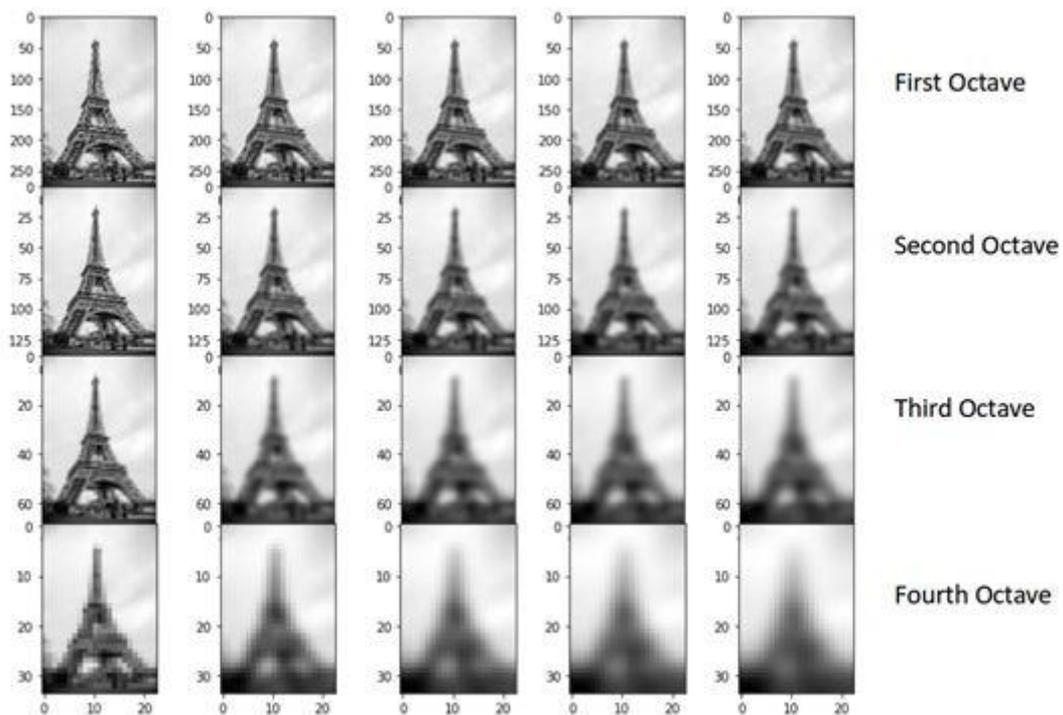
(1) 创建比例空间

使用高斯模糊技术 (Gaussian Blur) 来降低图像中的噪点。

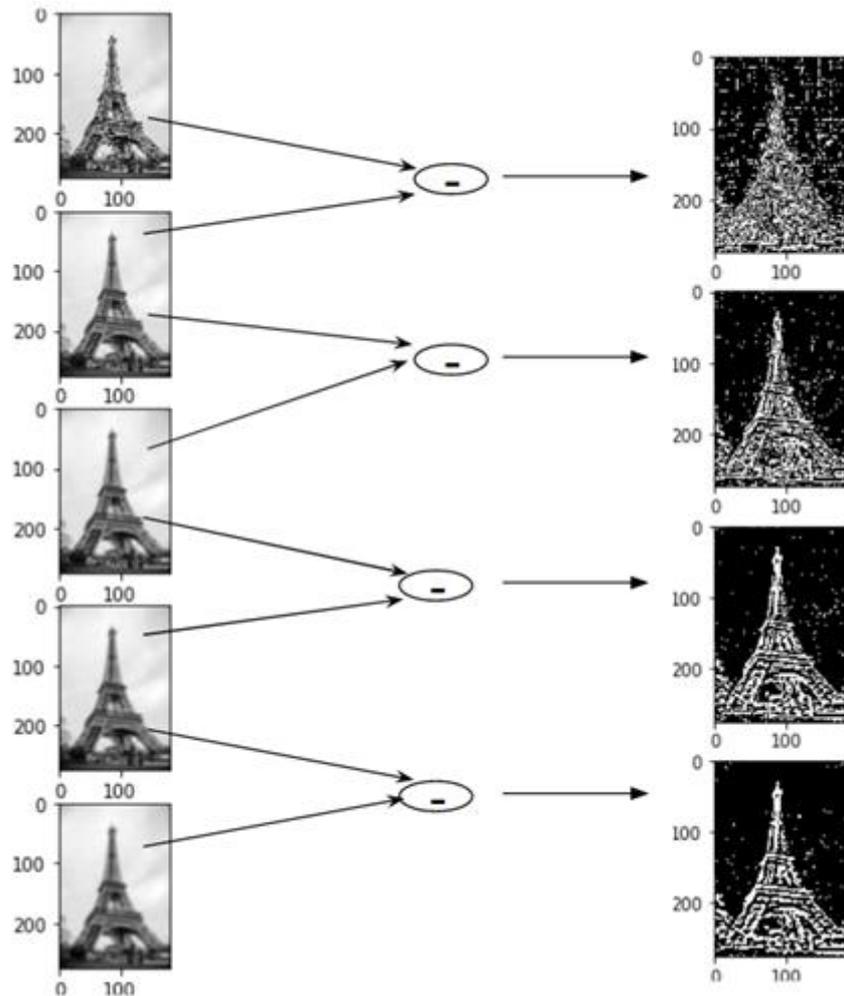
因此，对于图像中的每个像素，高斯模糊技术会基于其相邻像素计算一个值。以下是应用高斯模糊之前和之后的图像示例。如图所示，纹理和次要细节将从图像中删除，并且仅保留诸如形状和边缘之类的相关信息：



为了保证之后提取的特征与图像大小无关，需要创建“比例空间”，比例空间是从单个图像生成的具有不同比例的图像的集合。通常对原始图像进行四次缩放图像，并为每个缩放图像创建 5 个后续的模糊图像。



接下来需要使用高斯差异 (DoG) 的技术来增强特征。DoG指的是在相同的比例尺下，前一个图像减去后一个图像，为每个octave创建另一组图像，每组图像剩下4个图像。



(2) 关键点检测

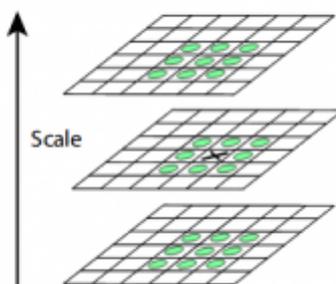
创建图像后，下一步就是从图像中找到可用于特征匹配的重要关键点。即找到图像的局部最大值和最小值。分为两个步骤：

1. 找到局部最大值和最小值
2. 删除低对比度的关键点（关键点选择）

为了定位局部最大值和最小值，仔细检查图像中的每个像素，并将其与相邻像素进行比较。

当我说“邻近”时，它不仅包括该图像的周围像素（像素所在的尺度），还包括八度中上一张和下一张图像的九个像素。

这意味着将每个像素值与其他26个像素值进行比较，以确定是否为局部最大值/最小值。例如，在下图中，我们从第一个八度获得了三个图像。将标记为x的像素与相邻像素（绿色）进行比较，如果它是相邻像素中最高或最低的像素，则将其选择为关键点：



由此便找到了那些尺度不变形的关键点。

我们已经成功地生成了尺度不变的关键点。但是这些关键点中的一些可能对噪声没有鲁棒性。这就是为什么需要进行最终检查以确保我们拥有最准确的关键点来表示图像特征的原因。

因此，将消除对比度低或非常靠近边缘的关键点。

为了处理低对比度关键点，将为每个关键点计算二阶泰勒展开（second-order Taylor expansion）。如果结果值小于0.03（大小），则剔除该关键点。

那么，如何处理其余关键点呢？再次检查以确定位置不佳的关键点。这些是具有高边缘度但对少量噪点无鲁棒性的关键点。使用二阶Hessian矩阵来识别此类关键点。

(3) 方向匹配

现在为每个关键点分配一个方向值以使旋转角度不变。再次将该步骤分为两个较小的步骤：

1. 计算幅度和方向
2. 创建大小和方向的柱状图

看下面的例图：

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

假设要找到红色像素值的大小和方向。为此，通过提取55和46与56和42之间的差值来计算x和y方向上的梯度。得出的分别是 $G_x = 9$ 和 $G_y = 14$ 。

一旦有了梯度，就可以使用以下公式找到幅度和方向：

$$Magnitude = \sqrt{[(G_x)^2 + (G_y)^2]} = 16.64$$

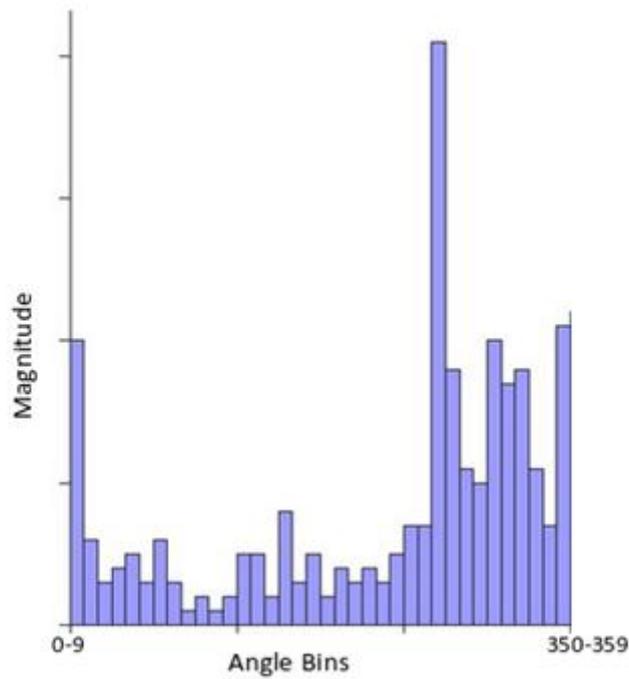
$$\Phi = \text{atan}\left(\frac{G_y}{G_x}\right) = \text{atan}(1.55) = 57.17$$

大小表示像素的强度，方向表示像素的方向。

现在，假设我们具有这些大小和方向值，就可以创建柱状图。

在x轴上，有一个角度值的区间，例如0-9、10-19、20-29，最大为360。假设现在角度值为57，它会落在第6个区间中。第6个bin值与像素的幅度成正比，即16.64。我们将对关键点周围的所有像素执行此操作。

这样就得到了下面的柱状图：

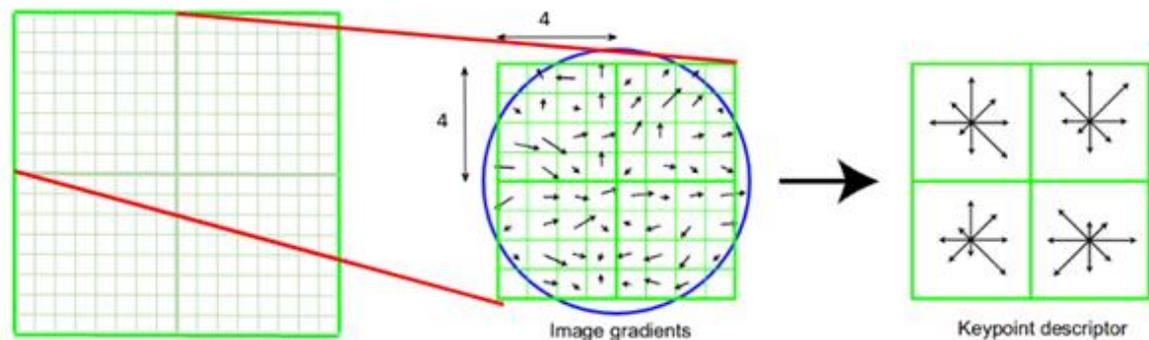


(4) 关键信息描述符

这是SIFT的最后一步。到目前为止，我们有稳定的关键点——不变的比例以及旋转角度。在本部分中，我们将使用相邻像素，它们的方向和大小为该关键点生成一个唯一的指纹，称为“描述符”。

另外，由于我们使用周围的像素，因此描述符对于图像的照度或亮度部分不变。

首先在关键点周围采用 16×16 的邻域。将该 16×16 区域进一步划分为 4×4 子块，对于这些子块中的每一个小块，使用幅度和方向生成柱状图。



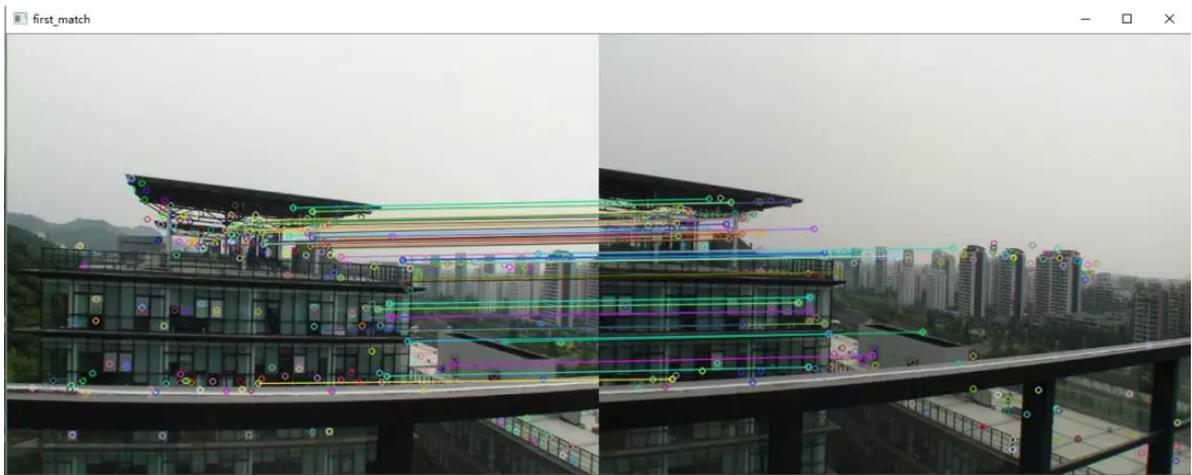
在此阶段，bin的大小增加，只占用8个bin(不是36个)。每一个箭头代表8个bin，箭头的长度定义了幅度。因此，每个关键点总共有128个bin值。

上面只是非常简略的描述，如果不是很熟悉的话根本理解不了，想要更加详细的了解请移步博客：[SIFT算法详解](#)

特征匹配

当组成一个图像对的两张图的关键点都被识别出来以后，我们需要将它们关联（或称「匹配」）起来，两张图像中对应的关键点在现实中是同一个点。一个可以实现该功能的函数是

「BFMatcher.knnMatch()」。这个匹配器（matcher）会衡量每一对关键点的描述子之间的距离，然后返回与每个关键点距离最小的k个最佳匹配结果。



图像变形

在匹配到至少 4 对关键点之后，我们就可以将一幅图像相对于另一幅图像进行转换。这个过程被称作图像变形（image warping）。空间中同一平面的任意两幅图像都是通过单应性变换关联起来的。单应性变换是具有 8 个参数的几何变换，通过一个 3×3 的矩阵表征。它们代表着对一幅图像整体所做的任何变形（与局部形变不同）。因此，为了得到变换后的待配准图像，我们计算了单应矩阵，并将它应用在了待配准图像上。

为了保证最优的变形，我们使用了 RANSAC 算法来检测轮廓，并且在进行最终的单应性变换之前将轮廓删除。该过程直接内置于 OpenCV 的「findHomography()」函数中。目前也有一些 RANSAC 的替代方案，例如 LMED（Least-Median robust method，最小中值鲁棒方法）。

参考资料

[不能错过！超强大的SIFT图像匹配技术详细指南（附Python代码）](#)
[SIFT算法详解](#)