

CS131 Lecture08: 特征描述符和大小调整

by: 斯坦福大学计算机科学系

github: https://github.com/zhaoxiongjun/CS131_notes_zh-CN (包含中英文版课件及相关课程视频)

1 尺度不变关键点检测

1.1 动机

到目前为止，我们已经在单个图像中讨论了关键点的检测，但是更广泛的应用需要在不同尺度的类似图像中进行这种检测。例如，我们可能希望在不了解行人大小的情况下，从自动车辆的视频源中搜索行人。同样，我们可能希望使用不同比例的照片缝合全景图。在这两种情况下，我们都需要在不同的尺度上独立地检测相同的关键点。

1.2 通用方法

目前，我们使用窗口（例如 Harris 角检测）来检测关键点。使用相同大小的窗口将无法在不同大小的图像中检测相同的关键点（图 1）。但是，如果窗口大小适当，则可以捕获相同的内容。

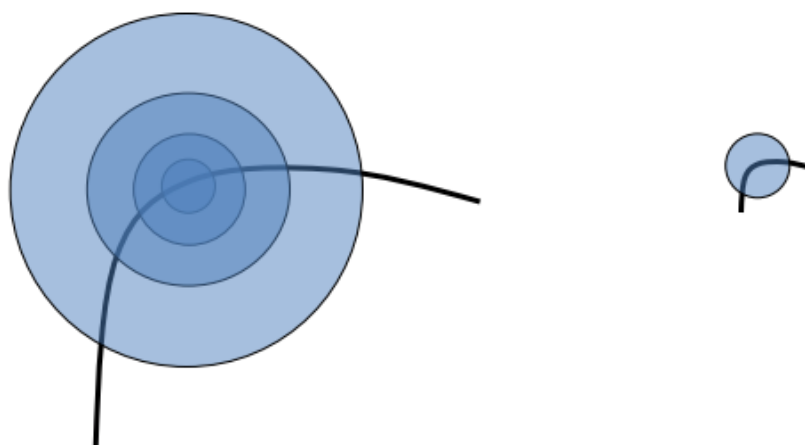


图 1：曲线的角以两个比例显示。请注意，右曲线上的圆形窗口捕获整个角，而左曲线上相同大小的窗口则不捕获。相反，我们必须在左曲线上选择一个多得多的圆形窗口来获得相同的信息。

我们如何独立地为每个图像找到正确缩放的窗口？我们需要以一种比例不变的方式描述“捕获相同的内容”的含义。更具体地说，考虑一个函数 $f(\text{window})$ ，它接受一个内容区域并为该区域的所有尺度输出相同的值。

现在考虑两个不同比例的相似图像。我们可以独立地改变每个图像的窗口大小，并将 $f(\text{window})$ 的响应绘制为窗口大小的函数：

在这两个图中，我们都可以独立地确定局部极值作为关键点。与这些极值相对应的窗口大小（在图 2 中、 s_1 和 s_2 ）为我们提供了两个图像之间的比例差。

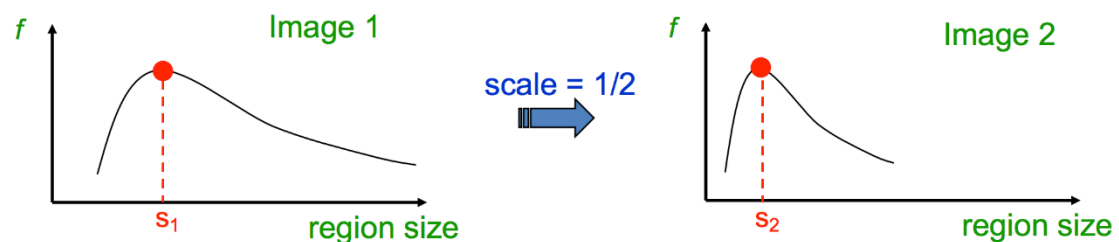


图 2: $f(\text{window})$ 响应函数随图像 1 和图像 2 的窗口大小而变化的两个图，其中图像 2 与图像 1 相似，但按 1/2 缩放。

1.2.1 平均强度

一个函数 $f(\text{window})$ 的一个候选者是窗口中像素的平均强度；这是因为当我们向上或向下缩放窗口时，平均强度不会改变。然而，平均强度并不能很好地捕捉窗口内的对比度或急剧变化；这使得在比较两幅图像中的 f 时很难找到清晰的极值。为了获得对比度，我们需要在组合中引入衍生品。

1.2.2 高斯差

另一个候选方案是使用高斯差。

考虑图像 I 。首先， I 与不同的高斯滤波器 s 重复卷积。这些卷积与 i 的缩小（即下采样）版本进行同样操作。生成了不同 s 和不同图像尺寸的高斯金字塔（图 3）。然后对相邻的高斯卷积图像进行相减，以计算高斯（DOG）图像的差异：

$$DOG(\sigma) = (G(k\sigma) - G(\sigma)) * I$$

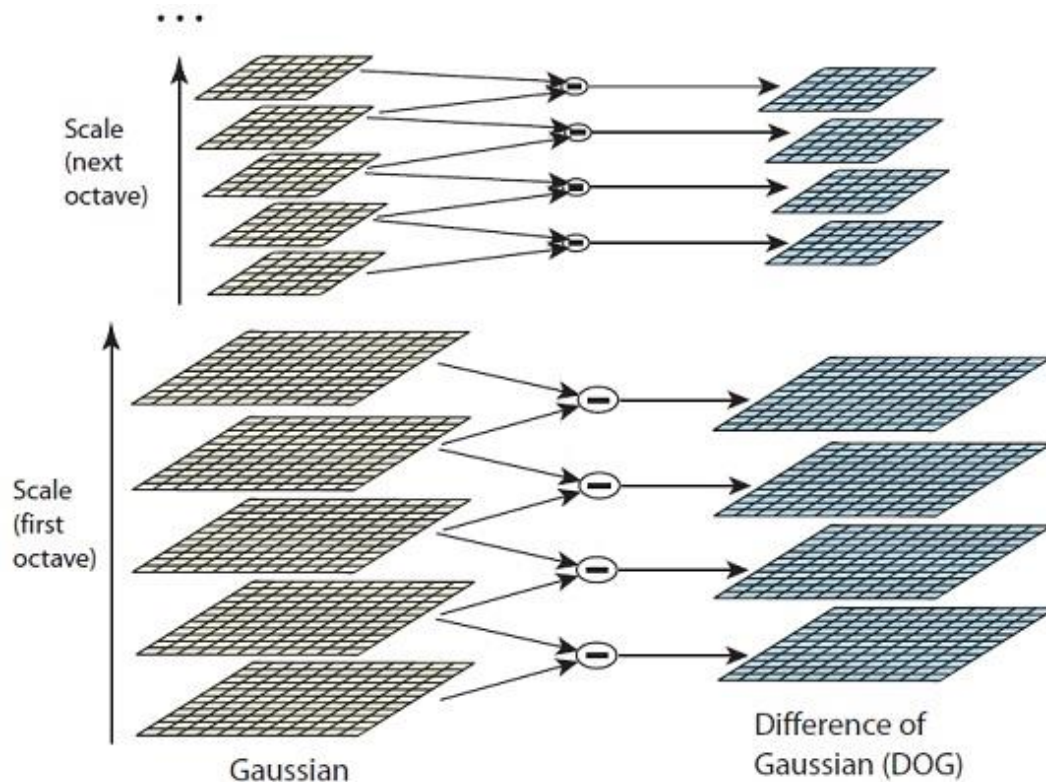


图 3：左边：不同高斯滤波器大小和不同图像的高斯金字塔。右边：相邻的高斯图的区别。

这些不同的高斯滤波器在不同的尺度上捕捉到了图像的细节。更具体地说，两个 Gaussians 1 和 2 的区别是删除所有同时出现在 1 和 2 处的细节，只保留出现在 1 或 2 之间的细节。高斯对小和大的差异分别捕捉了细微和粗糙的细节。鉴于高斯金字塔在 x - y 尺度空间上的不同，我们现在可以确定该三维空间内的局部极值，以确定关键点及其相关尺度。我们将一个给定的坐标与它的 26 个相邻坐标（在三维空间中）进行比较，如果它小于或大于所有相邻坐标，则认为它是一个极值（见图 4）。

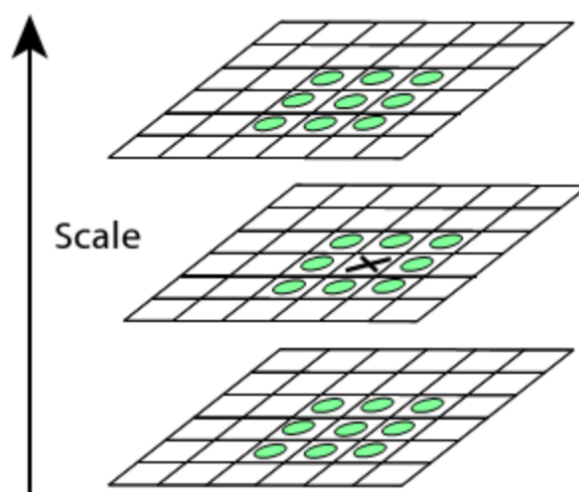


图 4: 给定 x - y 尺度空间中的坐标 (用黑色 x 表示), 检查其 26 个相邻坐标 (用绿色圆圈表示) 以确定原始坐标是否为局部极值。

1.2.3 Harris-Laplacian

第三个候选方案是使用 Harris-Laplacian 方法[2], 该方法在尺度不变关键点检测方面比 DOG 更有效, 但计算成本也可能更高。

再考虑一个图像 i 。首先, 我们创建多个 i 尺度, 并在每个尺度上运行 Harris 检测器, 以定位每个尺度级别的关键点。然后我们选择关键点, 最大化拉普拉斯跨越所有尺度。

转换到尺度不变描述符: 现在我们有几个方法来检测跨多个尺度的一致性关键点, 我们可以继续开发方法来以一种尺度不变的方式描述这些关键点, 以便它们可以匹配。

2 图像区域描述符

2.1 不变局部特征

点描述符应该是不变的和独特的。为了实现点描述符的鲁棒性, 我们将图像内容转换为对平移、旋转、缩放和其他成像参数不变的局部特征坐标。

不变局部特征的优点包括:

- 位置: 特征描述部件, 对遮挡和杂乱具有鲁棒性 (没有先前的分段)。
- 独特性: 特征可以从一个大型的对象数据库中识别出来。数量: 即使是很小的物体, 也能产生许多特征。
- 效率: 接近实时性能。
- 可扩展性: 可以很容易地扩展到各种不同的特性类型, 每种类型都增加了对更改的稳健性。

2.1.1 尺度不变性

- 唯一合理的尺度空间核是高斯。(Koenderink, 1984 年; Lindeberg, 1994 年)
- 一个有效的选择是检测高斯金字塔差异的峰值 (Burt&Adelson, 1983; Crowley&Parker, 1984-但是检查更多的尺度)
- 高斯与等尺度比的差与林德伯格尺度归一化拉普拉斯方程 (可从热扩散方程中看出) 非常接近。

2.1.2 旋转不变性

给 DOG 一个关键点及其刻度,

- 1: 平滑（模糊）与关键点比例关联的图像。
- 2: 计算关键点附近的图像渐变。
- 3: 通过负关键点方向旋转渐变方向和位置。换句话说，描述与方向相关的所有特征。

2.1.3 SIFT 描述符构成

使用精确的梯度位置是脆弱的，因此我们希望在允许泛化的同时生成类似的描述符。我们创建了一个方向柱状图数组，并将梯度放置到 8 个方向箱的局部方向柱状图中。建议将坡度分为 8 个垃圾箱，通过实验发现，这些垃圾箱数量表现出最佳性能。

具体如下：

- 1: 创建方向直方图数组
- 2: 将旋转的渐变放到局部方向直方图中，其中每个渐变根据距离贡献到附近的直方图；远离中心的渐变按比例缩小。SIFT 作者[3]发现，使用每个柱状图 8 个方向箱和 4x4 柱状图阵列（图 2）可以获得最佳结果。
- 3: 比较两幅图像之间的每一个矢量，找出匹配的关键点。
- 4: 为了增加高对比度照片中照明变化的鲁棒性，在比较之前对向量进行规格化。这可以减轻不可靠的三维照明效果，例如由非常大的图像渐变引起的眩光；这是通过在再次标准化之前将矢量中的值限制在 0.2 以下（实验调整的值）来实现的。

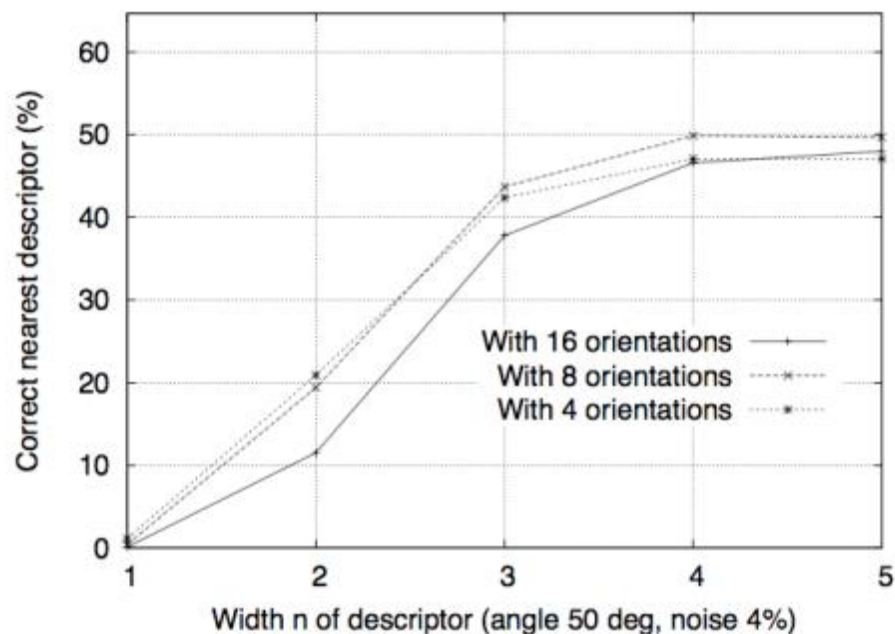


图 5: 这个图显示了正确匹配的关键点的百分比，它是描述符宽度和柱状图箱数的函数。

3 HOG：另一个图像区域描述

3.1 定向梯度柱状图

定向梯度的柱状图 (hog) [4] 描述符在图像中找到一个弹出的对象，这个对象可以被区分。HOG 的一般算法如下：

- 1: 将图像窗口划分为小的空间区域或单元。
- 2: 对于每个单元，累积一个局部柱状图；将渐变方向分组到均匀间隔的箱中，并将像素渐变的大小分配到对应于渐变方向的适当箱中（图 6）。
- 3: 对由多个单元格组成的较大区域（称为“块”）上的局部柱状图进行规格化。

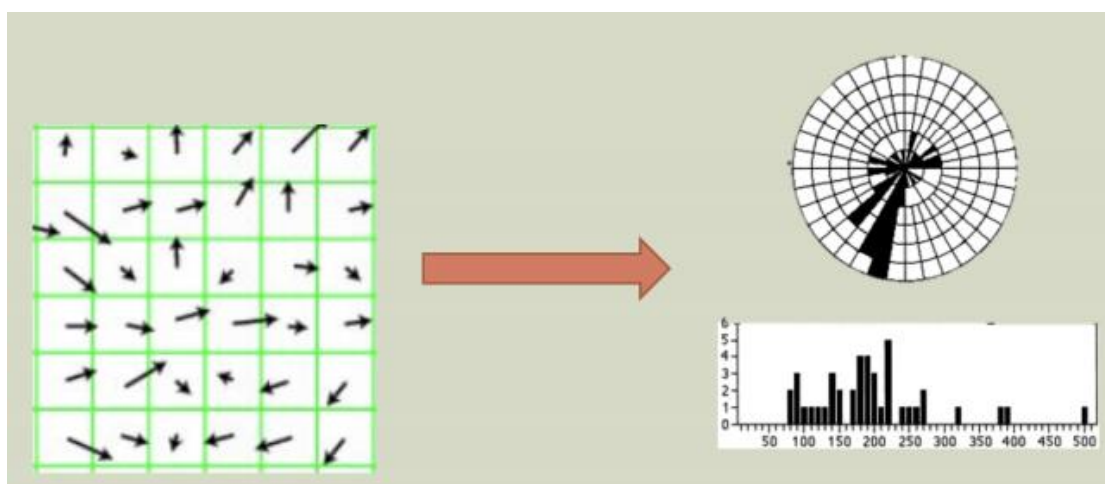


图 6：这里我们看到了一个跟踪每个梯度方向梯度大小的可视化示例。

使用 HOG 有几个缺点：

- 1: 检测时变化幅度大
- 2: 非常慢
- 3: 当背景有不同的光照时，没有很好的组织

尽管有这些缺点，HOG 还是很有效的。请注意下图中应用 HOG 的结果。

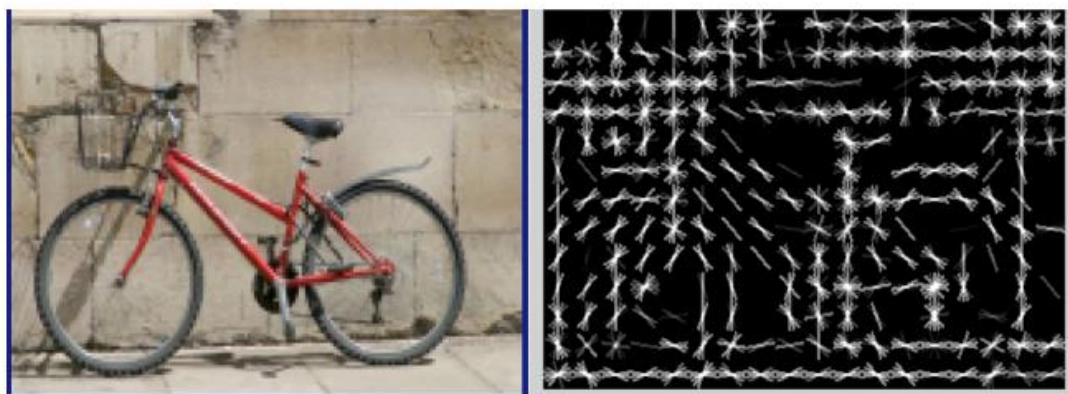


图 7：HOG 应用在自行车

3.2 HOG 和 SIFT 的差别

两者之间有一些细微的差别。HOG 用于在整个图像上查找渐变。SIFT 用于关键点匹配。SIFT 柱状图指向自然正梯度，而 HOG 没有。HOG 使用邻域，而 SIFT 使用权重计算不同的描述符。

4 衔接图像尺寸调整

由于屏幕大小不同，我们需要根据显示功能调整内容大小。通常，我们试图通过拉伸或收缩图像来强制内容填充任何类型的显示。然而，这产生的结果并不理想。那么解决方案是什么呢？内容感知重新定位操作员。

重定目标意味着我们接受一个输入并“重新定位”到一个不同的形状或大小。将输入想象为大小为 $n \times m$ 的图像，将所需输出想象为大小为 $n' \times m'$ 的图像。重定目标的想法是：

- 1: 遵守几何约束（例如，展弦比）
- 2: 保存重要的内容和结构
- 3: 限制工件。

然而，被认为“重要”的东西是非常主观的，因为对一个观察者来说可能重要的东西对另一个观察者来说可能不重要。

4.1 像素能量

决定什么是“重要”的方法是使用显著性措施。有许多不同类型的显著性度量，但其概念是相同的：每个像素 p 都有一定数量的“能量”，可以用函数 $E(p)$ 表示。

其概念是，具有较高能量值的像素比具有较低能量值的像素更突出或更重要。真正进入 E 中心的是旁观者。

一个很好的例子是使用像素 p 的梯度幅度来严重影响 $E(p)$ ，因为这通常表示一条边。由于人类特别容易接受边缘，这是图像的一部分，与具有较低梯度量级的图像相比，具有潜在的价值和趣味性。因此，它保留了很强的轮廓，并且总体上非常简单，能够产生很好的效果。图像 I 的 E 示例可以表示为

$$E(I) = \left| \frac{\partial I}{\partial x} \right| + \left| \frac{\partial I}{\partial y} \right|.$$

4.2 缝衔接

假设我们有一个分辨率为 $m \times n$ 的输入图像，我们正在寻找一个输出图像 $m' \times n'$ ，其中 $n' < n$ 。我们如何知道要删除哪些像素？我们可以使用像素能量的这个概念来识别相邻像素的路径，或者接缝，它们的组合像素能量最低，可以从图像中去

除。

请注意，接缝不必是严格的行和列。事实上，大多数时间接缝是水平或垂直穿过图像的曲线。如果接缝从图像的下边缘到上边缘，则它是水平的。同样，如果接缝从图像的左边缘延伸到右边缘，则它是垂直的。但是，接缝的布局方式总是这样：如果接缝是垂直的，则每行只有一个像素；如果接缝是水平的，则每列只有一个像素。

本质上，当选择要从图像中删除的内容时，接缝可以避免图像的所有重要部分，以便在删除时对图像造成最少的干扰。关于接缝雕刻和使用案例，还有更多的事情需要考虑，可以用类似的技术加以改进，但这是接缝如何操作的核心思想。