



Feature Detection and Matching

任课教师：李贤芝

CONTENTS

目录

01

特征检测

Feature Detector

02

特征描述

Feature Descriptor

03

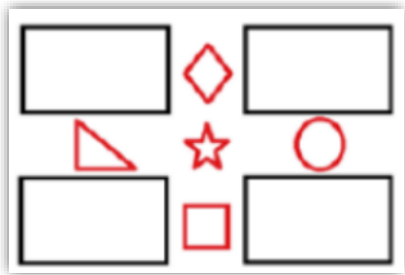
特征匹配

Feature Matching

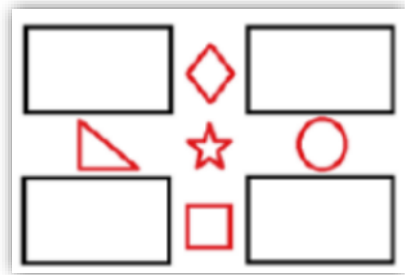


Example

小蓝看到的图像：



小红看到的图像：



我们看到的是否是同一幅图像呢？

对话1: **Step1: 寻找那些能在其他图像中较好匹配的位置，即特征检测**

小蓝：我的图像里有五个很明显的特征，分别在图像的上下左右中五个位置。

小红：我的图像里也有五个很明显的特征，分别在图像的上下左右中五个位置。

对话2: **Step2: 把检测到的关键点周围区域转化为一个紧凑和稳定的描述，即特征描述**

小蓝：我的一个特征左边是三角形，右边是圆形，上面是菱形，下面是正方形，中间是星星。

小红：我也有一个特征左边是三角形，右边是圆形，上面是菱形，下面是正方形，中间是星星。

对话3: **Step3: 对特征的相似性进行判断，即特征匹配**

小蓝和小红：那我们看到的就是一副图像了！

Motivation

同一场景下视角、光线剧烈变化

by Diva Sianby [Sarah Washford](#)by scgbt

Motivation

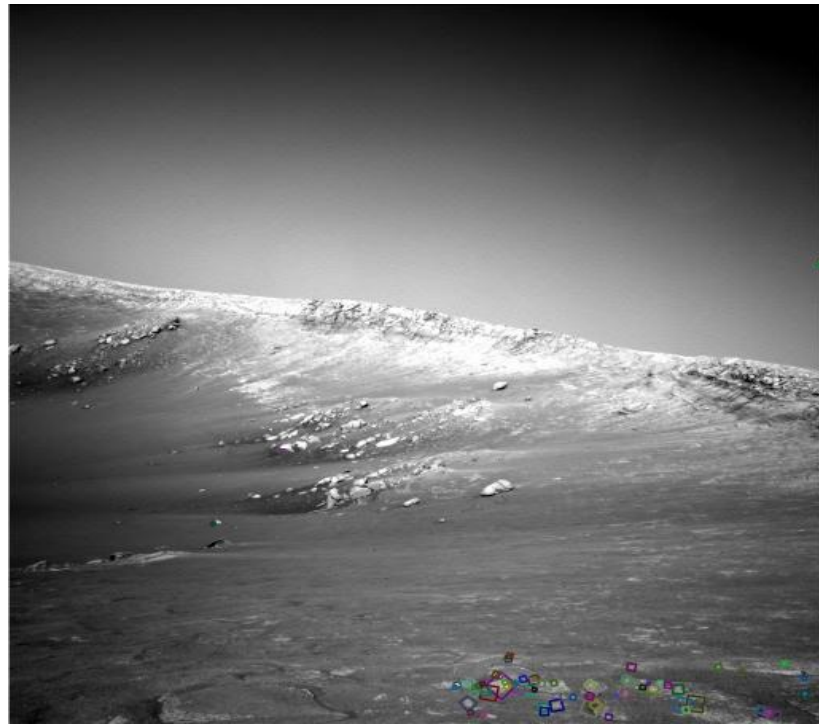
这两幅图像有重叠区域吗？



美国宇航局火星探测器图像

Motivation

仔细看彩色方框.....他们对应同一区域！



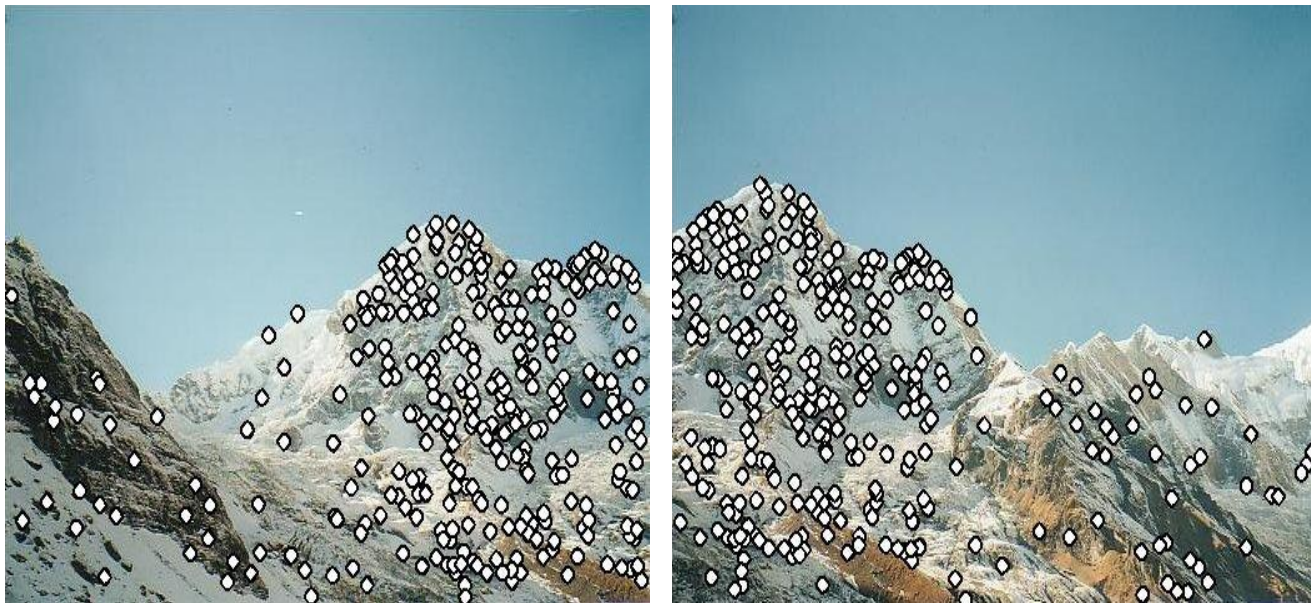
SIFT 特征匹配结果

Motivation

如何将这两幅图像进行拼接（对齐）？

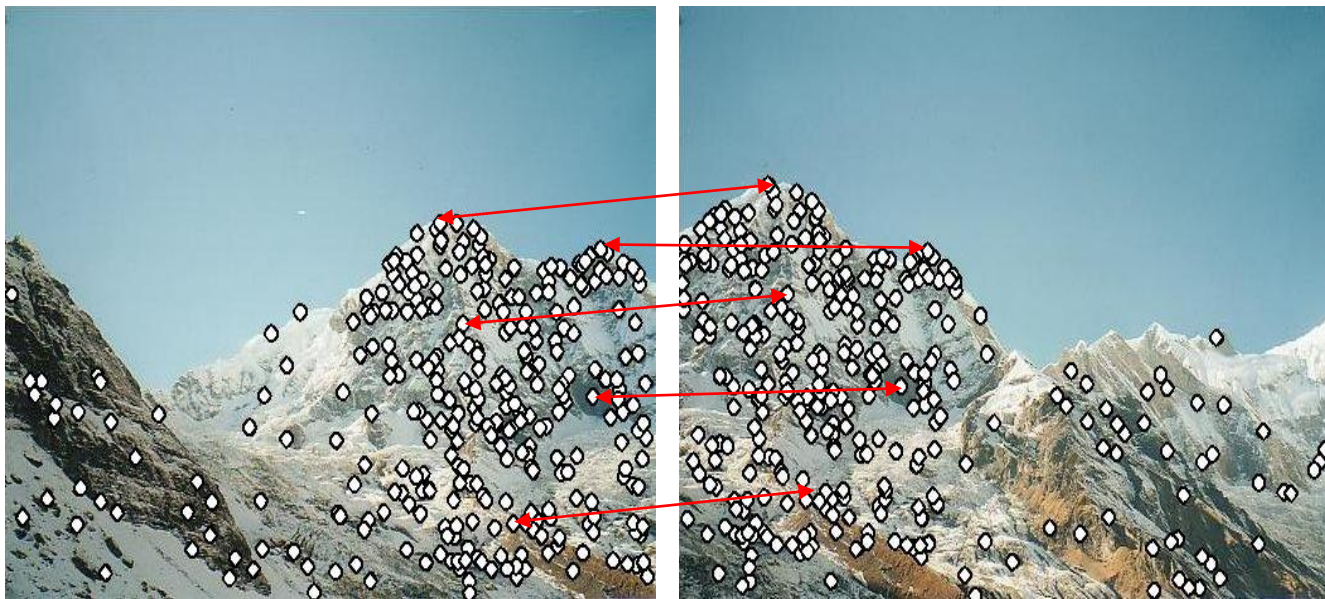


局部特征检测与匹配



1) 在两幅图像中检测局部特征（特征检测+特征描述）

局部特征检测与匹配



- 1) 在两幅图像中检测局部特征（特征检测+特征描述）
- 2) 找到对应匹配的特征点

局部特征检测与匹配



- 1) 在两幅图像中检测局部特征（特征检测+特征描述）
- 2) 找到对应匹配的特征点
- 3) 利用匹配的点集进行图像对齐

局部特征检测与匹配

- 问题 1:
 - 如何在两幅图像的相同区域检测到关键点？



以上检测结果可行吗? No Way!

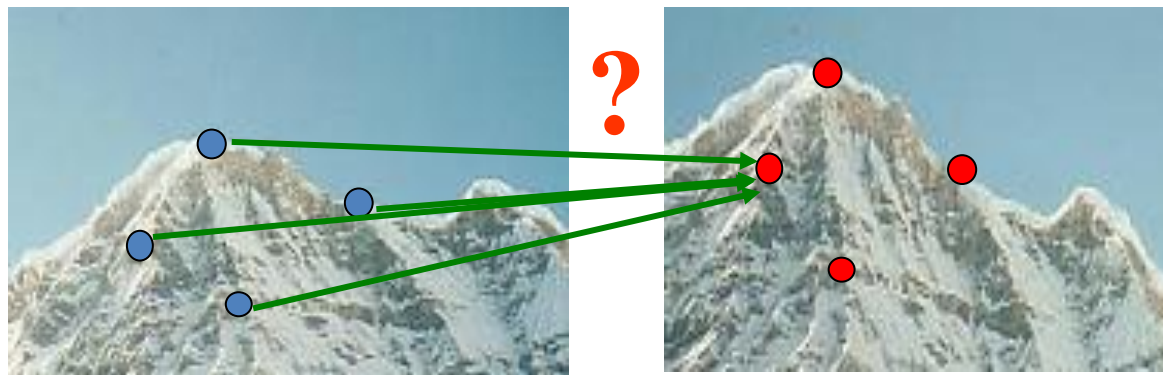


We Need This !

需要重复 (repeatable) 检测相同区域的特征点检测器!

局部特征检测与匹配

- 问题 2:
 - 如何使得每一个点都可以找到正确匹配的点？



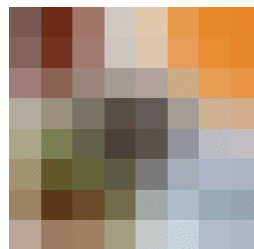
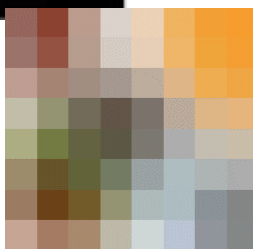
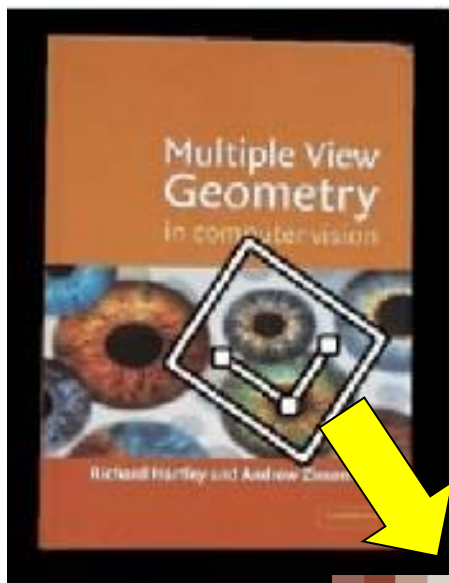
每一个点需要一个可靠且独特（distinctive）的描述子！

局部特征检测与匹配是困难的.....



Figure from T. Tuytelaars ECCV 2006 tutorial

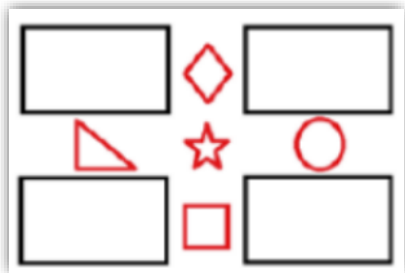
局部特征检测与匹配是困难的.....



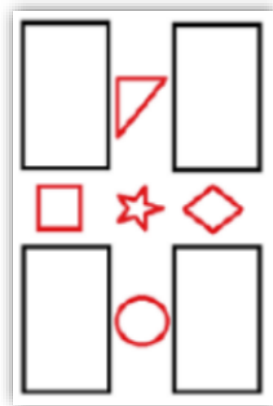
视角变换

Example

小蓝看到的图像：



小红看到的图像：



向右旋转90°

我们看到的是否是同一幅图像呢？

对话1：

小蓝：我的图像里有五个很明显的特征，分别在图像的上下左右中五个位置。

小红：我的图像里也有五个很明显的特征，分别在图像的上下左右中五个位置。

对话2：

小蓝：我的一个特征左边是三角形，右边是圆形，上面是菱形，下面是正方形，中间是星星。

小红：我的一个特征左边是正方形，右边是菱形，上面是三角形，下面是圆形，中间是星星。

对话3：

小蓝和小红：那我们看到的就是同样一副图像了！我们看到的不是同一幅图像！

局部特征检测与匹配是困难的.....

其他问题:

- 噪声
- 遮挡
- 人为压缩
- ...

好的局部特征应具备的性质

- **可重复性**：一张图像上有某个特征，若该图像经过各种几何和光线变换后，该特征仍然可以在变换后的图像上被检测出来，那么这个特征的可重复性强。
 - 旋转不变性
 - 尺度不变性
- **显著性**：一个特征容易被识别到，则它的显著性强。
- **表达紧凑而高效**：一个特征若能用较少的像素信息表现显著的内容，则它的表达紧凑而高效。
- **局部性**：特征只会在小区域中与周围信息产生关联，局部的“关系”不会受全局变化的影响。

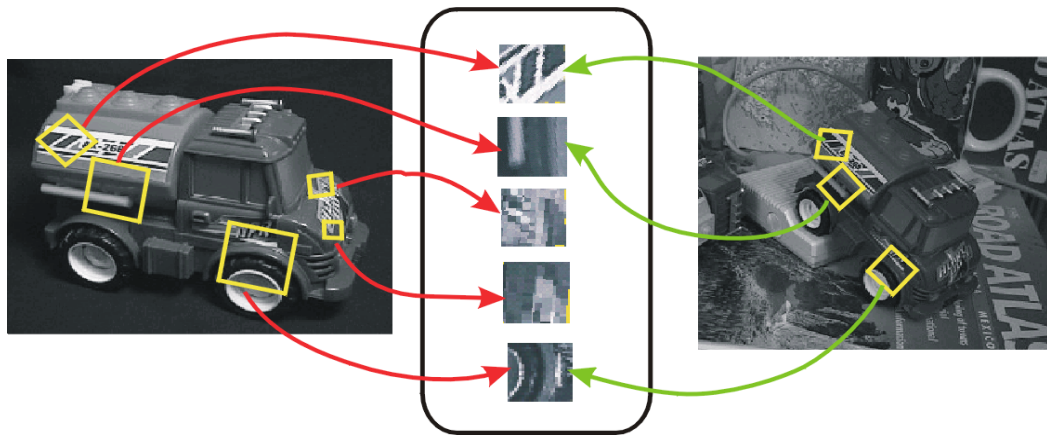
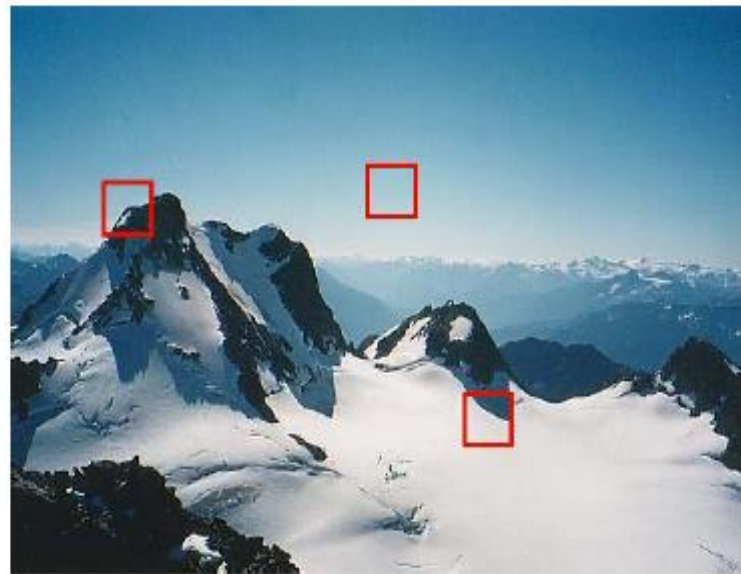
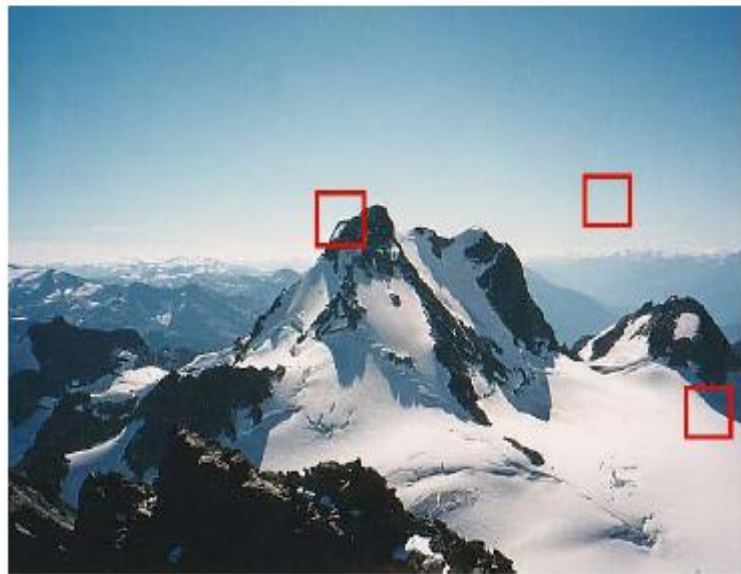


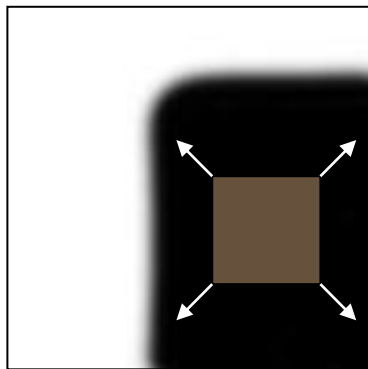
Figure: David Lowe

Main questions

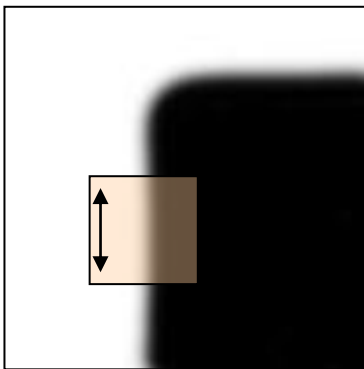
- 特征点从何而来？
 - 怎么在多幅图像中检测相同的、显著性的特征点？
- 如何描述局部特征使其保证独特性？
- 如何建立特征点之间的对应关系？, i.e., 如何计算匹配？

哪一种区域更容易识别匹配？

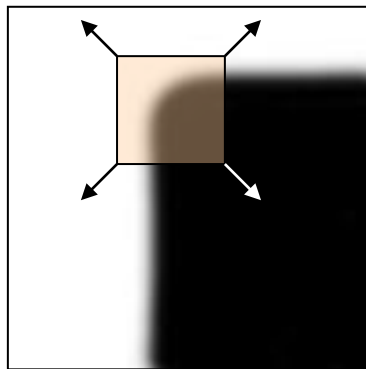




(1)



(2)

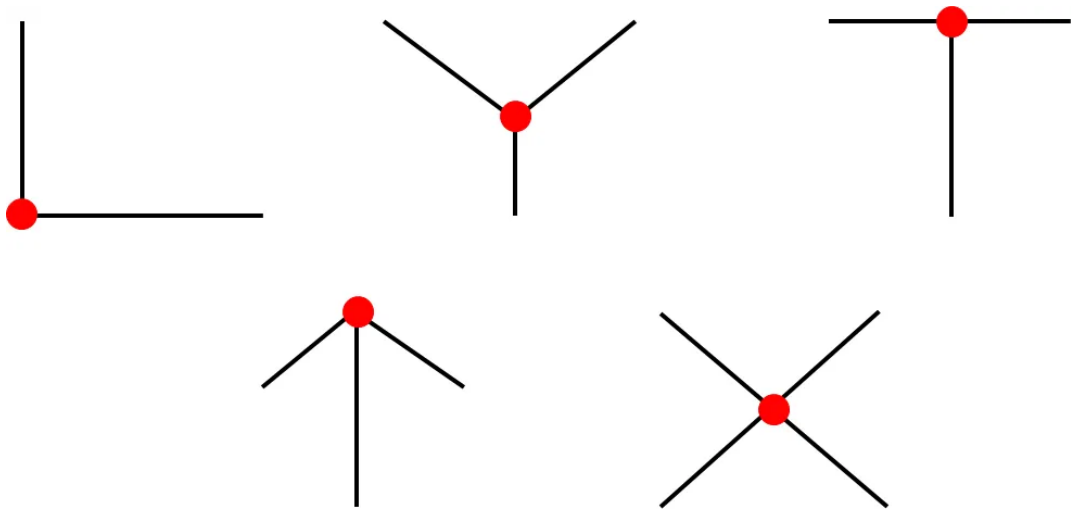


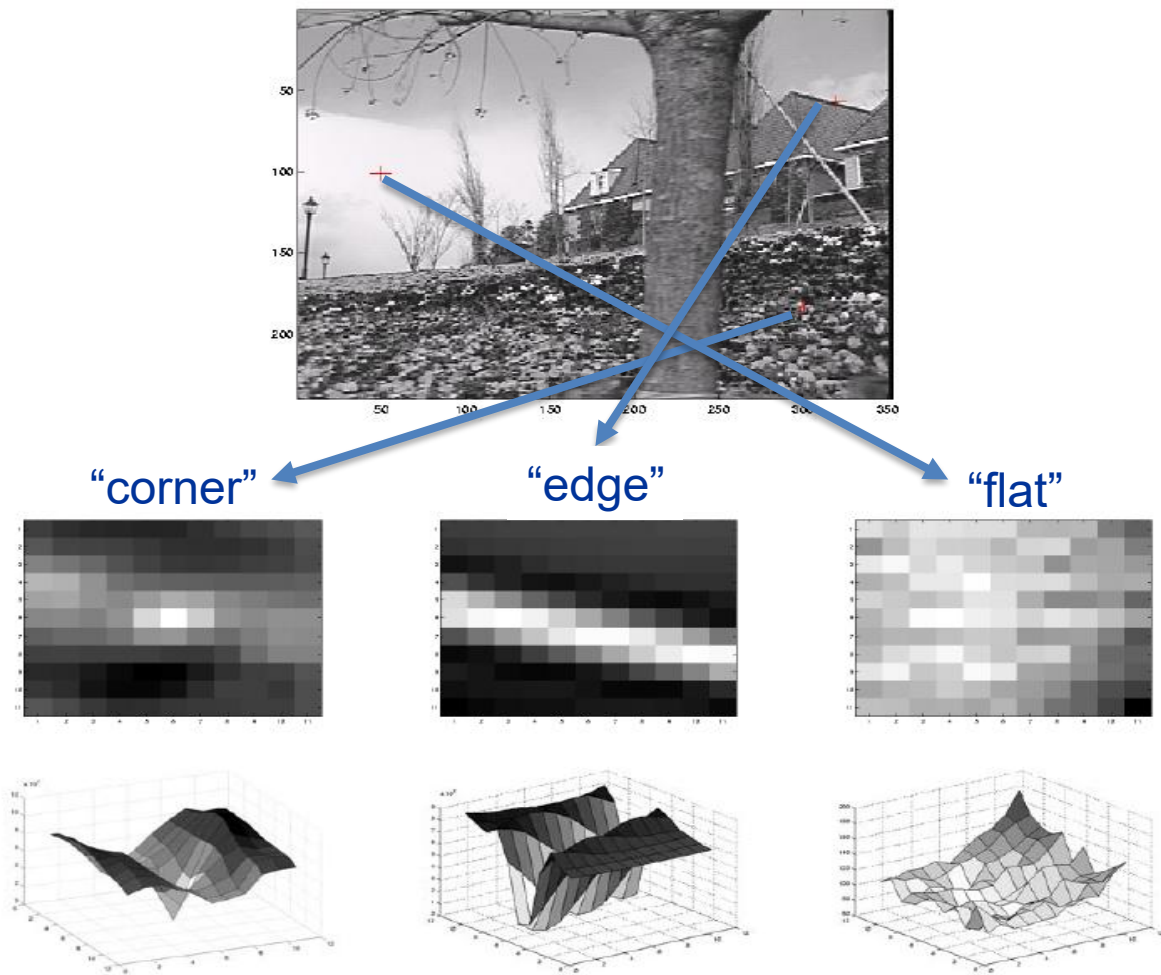
(3)

- (1) 平坦区域：任意方向移动，无明显灰度变化；
- (2) 边缘：沿着边缘方向移动，无明显灰度变化；
- (3) 角点：沿着任意方向移动，灰度都会明显改变。

角点作为独特的兴趣点

- 容易在图像的局部区域检测得到
- 在任何方向上移动窗口都应该在强度上产生很大的变化





检测窗口位置发生微小变化后，灰度值强度的变化

Harris角点检测

将图像窗口平移 $[u, v]$ 产生的灰度变化 $E(u, v)$ 如下：

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

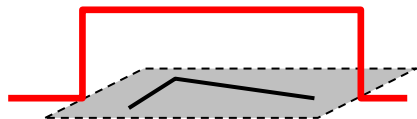
窗口函数

(空间上变化的权重函数)

平移后的
图像灰度

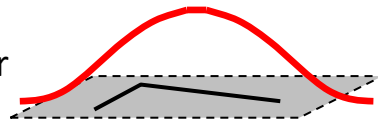
图像灰度

Window function $w(x, y) =$



1 in window, 0 outside

or



Gaussian

Harris角点检测

$$\begin{aligned} E(u, v) &= \sum_{(x,y)} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \\ &\approx \sum_{(x,y)} w(x, y) [I(x, y) + \frac{\partial I}{\partial x}(x, y)u + \frac{\partial I}{\partial y}(x, y)v - I(x, y)]^2 \text{ (一阶泰勒展开)} \\ &\approx \sum_{(x,y)} w(x, y) [\frac{\partial I}{\partial x}(x, y)u + \frac{\partial I}{\partial y}(x, y)v]^2 \text{ (消除重复项)} \\ &= \sum_{x,y} w(x, y) (u^2 f_x^2(x, y) + 2uv f_x(x, y) f_y(x, y) + v^2 f_y^2(x, y)) \text{ (二阶泰勒展开)} \\ &= \sum_{x,y} w(x, y) (u^2 I_x^2 + 2uv I_x I_y + v^2 I_y^2) \text{ (简化)} \\ &= [u \quad v] (\sum w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}) \begin{bmatrix} u \\ v \end{bmatrix} \\ &= \begin{bmatrix} u \\ v \end{bmatrix}^T H \begin{bmatrix} u \\ v \end{bmatrix} \end{aligned}$$

Harris矩阵

$$H = \begin{bmatrix} \sum_{(x,y)} w(x, y) (\frac{\partial I}{\partial x}(x, y))^2 & \sum_{(x,y)} w(x, y) (\frac{\partial I}{\partial x}(x, y)) \frac{\partial I}{\partial y}(x, y) \\ \sum_{(x,y)} w(x, y) (\frac{\partial I}{\partial x}(x, y)) \frac{\partial I}{\partial y}(x, y) & \sum_{(x,y)} w(x, y) (\frac{\partial I}{\partial y}(x, y))^2 \end{bmatrix}$$

Harris矩阵特征值分析

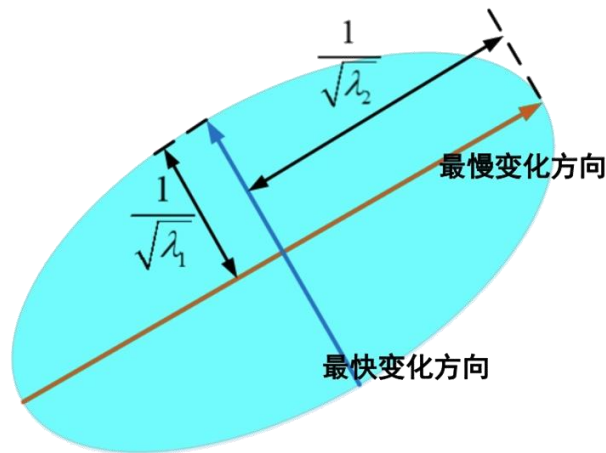
$$H = \begin{bmatrix} \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right)^2 & \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right) \frac{\partial I}{\partial y}(x,y) \\ \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right) \frac{\partial I}{\partial y}(x,y) & \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial y}(x,y) \right)^2 \end{bmatrix}$$

$$SVD(H) = U \Sigma V^T$$

λ_1 : 最大特征值

λ_2 : 最小特征值

- 两个特征值对应的特征向量可以确定变化最快的方向和变化最慢的方向；
- 两个特征值的大小可以反映在最快变化方向和最慢变化方向上变化的快慢。



Harris矩阵特征值分析

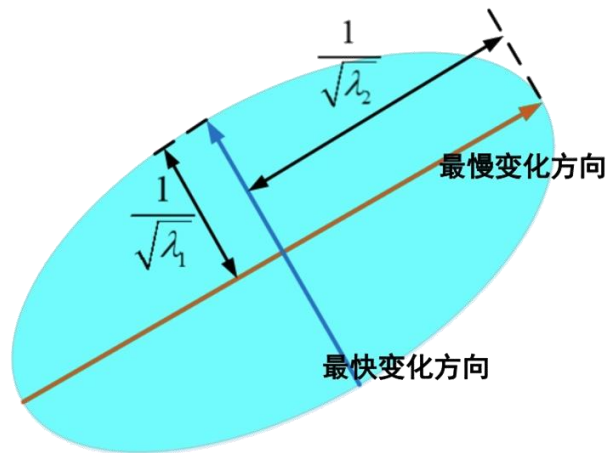
$$H = \begin{bmatrix} \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right)^2 & \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right) \frac{\partial I}{\partial y}(x,y) \\ \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right) \frac{\partial I}{\partial y}(x,y) & \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial y}(x,y) \right)^2 \end{bmatrix}$$

$$SVD(H) = U \Sigma V^T$$

λ_1 : 最大特征值

λ_2 : 最小特征值

- $\lambda_1 \approx \lambda_2 \approx 0$: 两个方向上灰度变化都很小, 兴趣点位于光滑区域
- $\lambda_1 > 0, \lambda_2 \approx 0$: 一个方向变化快, 一个方向变化慢, 兴趣点位于边缘区域
- $\lambda_1, \lambda_2 > 0$: 两个方向变化都很快, 兴趣点位于角点区域



Harris角点响应值

$$H = \begin{bmatrix} \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right)^2 & \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right) \frac{\partial I}{\partial y}(x,y) \\ \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial x}(x,y) \right) \frac{\partial I}{\partial y}(x,y) & \sum_{(x,y)} w(x,y) \left(\frac{\partial I}{\partial y}(x,y) \right)^2 \end{bmatrix} = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

$SVD(H) = U\Sigma V^T$ 特征值分解，计算量大！

Harris角点响应值： $R = \det(M) - \alpha \text{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$

- α 是一个经验常数，通常取0.04到0.06之间
- $\det(M)$ 表示矩阵 M 的行列式， $\det(M) = \lambda_1 \lambda_2 = AC - B^2$
- $\text{trace}(M)$ 表示矩阵 M 的迹， $\text{trace}(M) = \lambda_1 + \lambda_2 = A + B$
- 只有当 λ_1 和 λ_2 同时取最大值时， R 值才最大

Harris角点检测算法流程

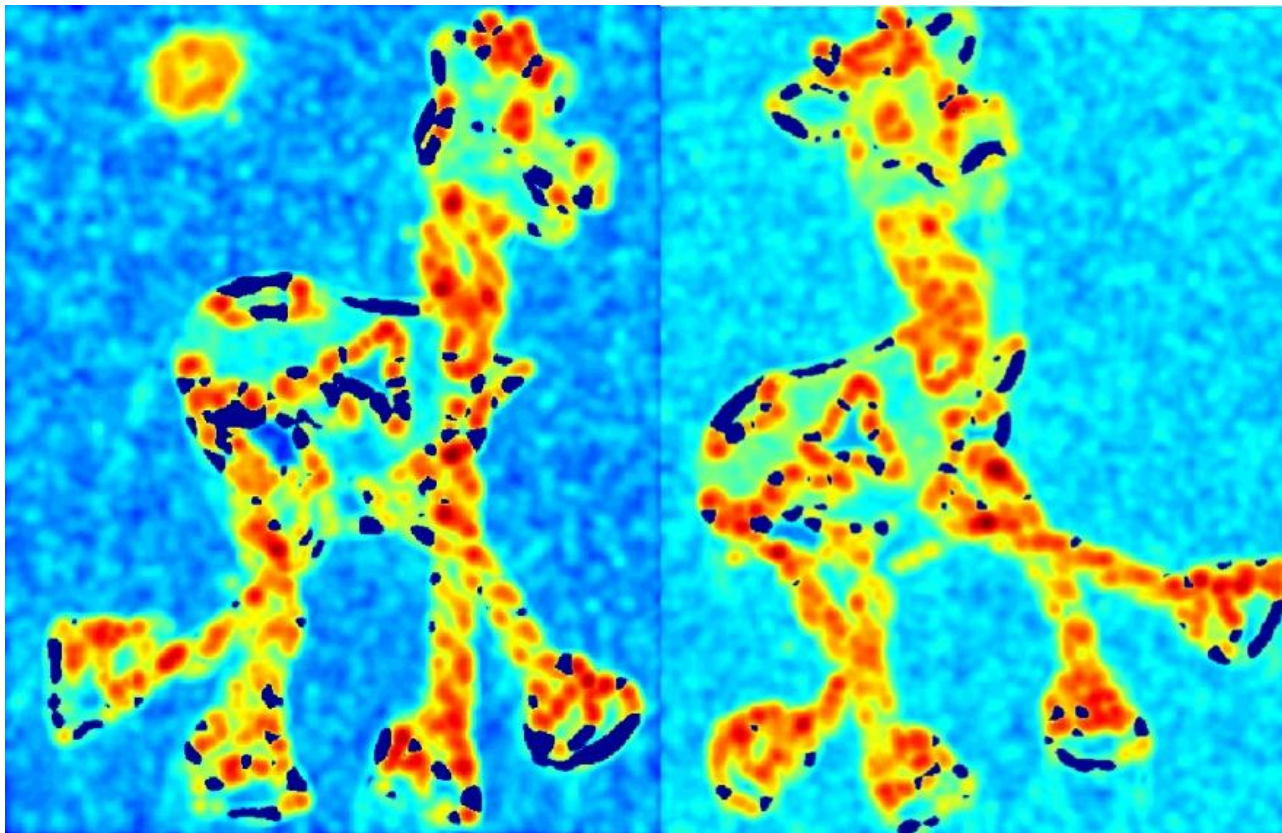
1. 计算每个像素位置的Harris角点响应值
2. 找到Harris角点响应值大于给定阈值的像素点
3. 非极大值抑制：选取局部响应最大值作为特征点，避免重复的检测

Harris Detector: 样例



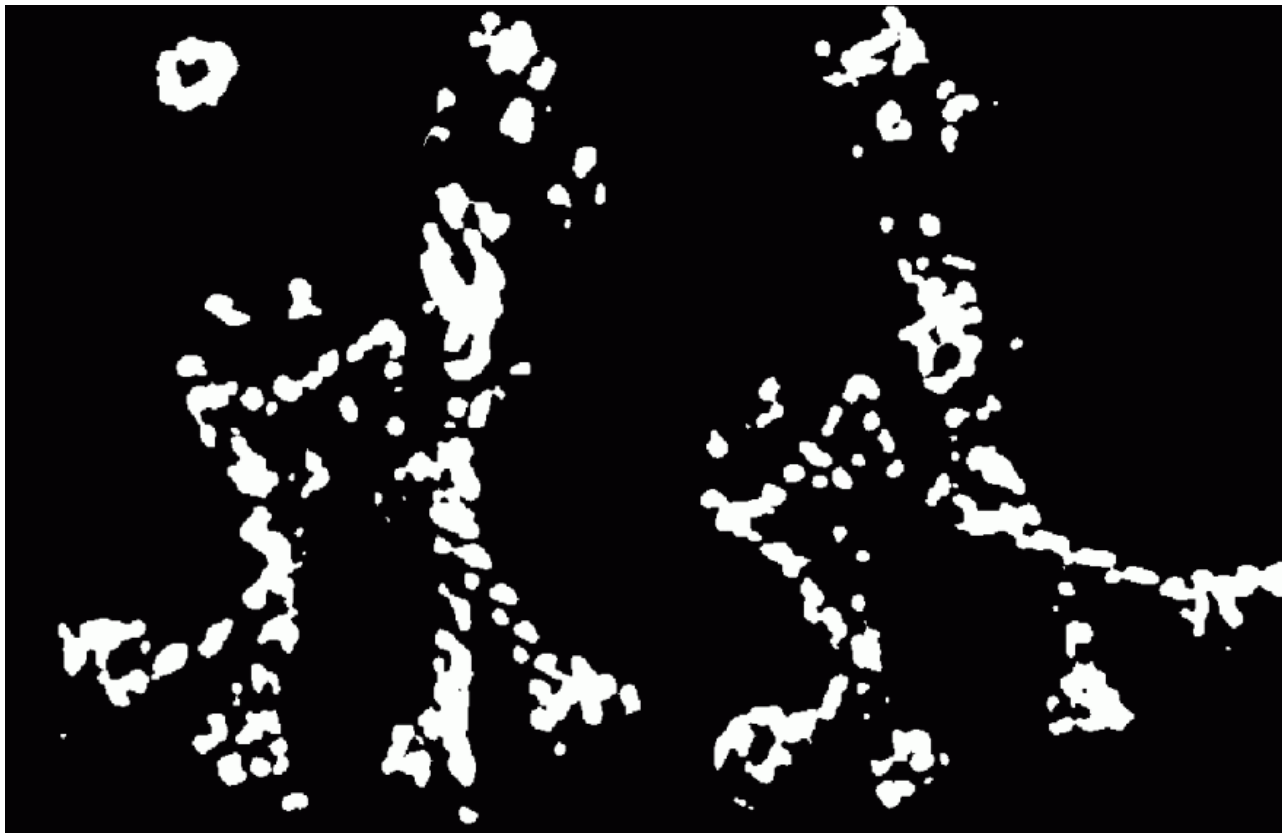
Harris Detector: 样例

每个像素的角点响应值 R



Harris Detector: 样例

找到角点响应值 $R > \text{threshold}$ 的像素点



Harris Detector: 样例

只保留角点响应值局部最大的像素点

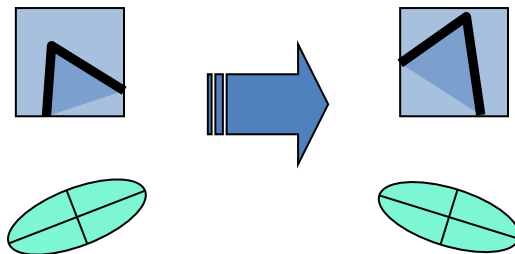


Harris Detector: 样例



Harris角点性质

- 旋转不变性

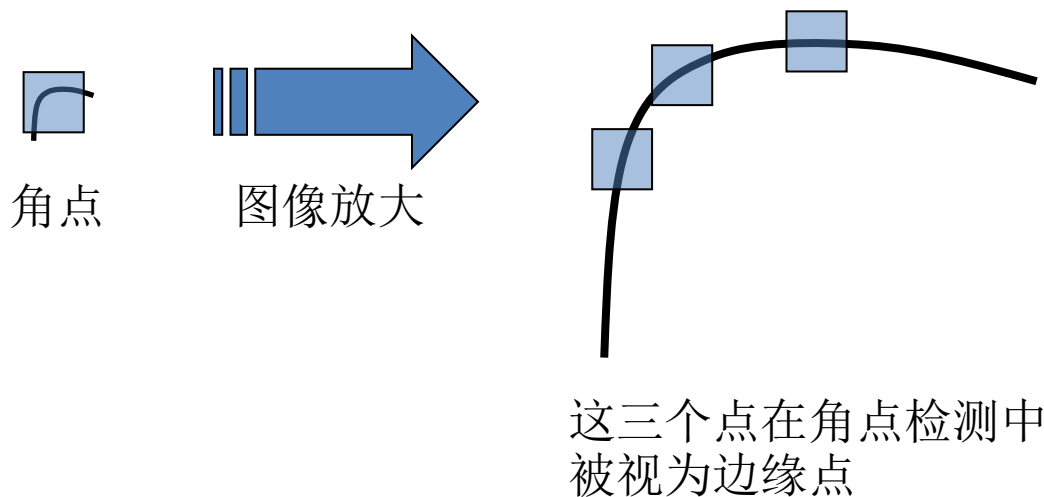


椭圆经过旋转形状保持不变 (i.e. 特征值不变)

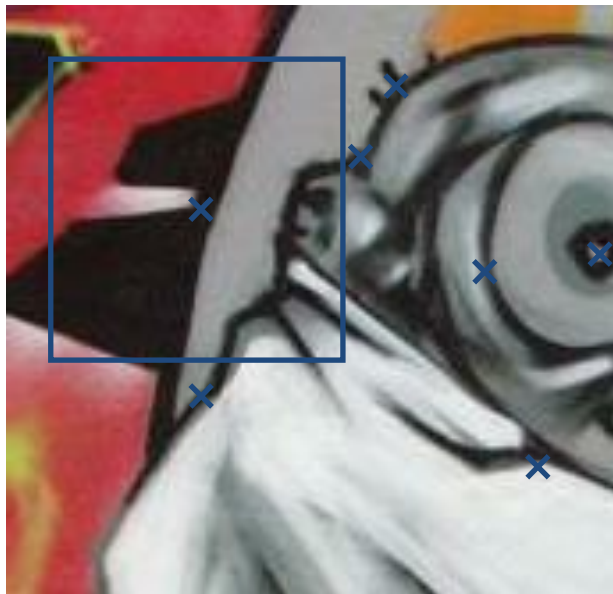
角点响应值 R 对于图像的旋转具有不变性！

Harris角点性质

- 难以应对图像几何尺度的变化

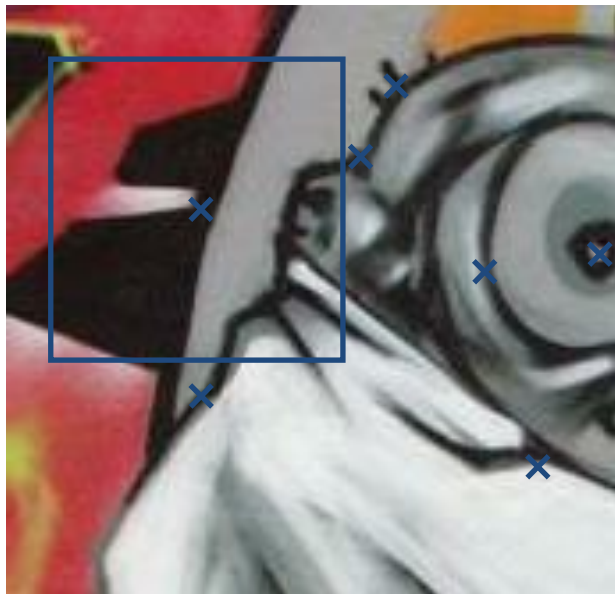


图像尺度变化而检测窗口尺度不变时，会导致检测窗口对特征的检测结果不同。



如何检测具有尺度不变性的特征点？

思路：为了检测具有尺度不变性的特征点，我们需要找到一种方法，让检测窗口可以对同一张图像的尺度变化作出响应，也进行尺度上的改变，从而可以在尺度变换后的同一张图像上识别到同一个目标。



- “变”的东西是一张图像（或者说图像上的检测对象）的尺度；
- “不变”的东西是一个相对区域内的内容；
- 目标：寻求一个合理的检测窗口的绝对尺度以迎合内容的相对不变。

问题建模

- “变”的东西是一张图像（或者说图像上的检测对象）的尺度；
- “不变”的东西是一个相对区域内的内容；
- 目标：寻求一个合理的检测窗口的绝对尺度以迎合内容的相对不变。

方案：

把检测窗口内的“内容”作为一个函数，把检测窗口的“尺度”作为自变量。

问题建模

把检测窗口内的“内容”作为一个函数，把检测窗口的“尺度”作为自变量。



$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

对上述数学方程的解释：在两张图像上，不同的两个自变量值（i.e., 窗口尺度 σ ），对应某种函数 f 的函数值相等。

如何量化所谓的“内容”？

把检测窗口内的“内容”作为一个函数，把检测窗口的“尺度”作为自变量。

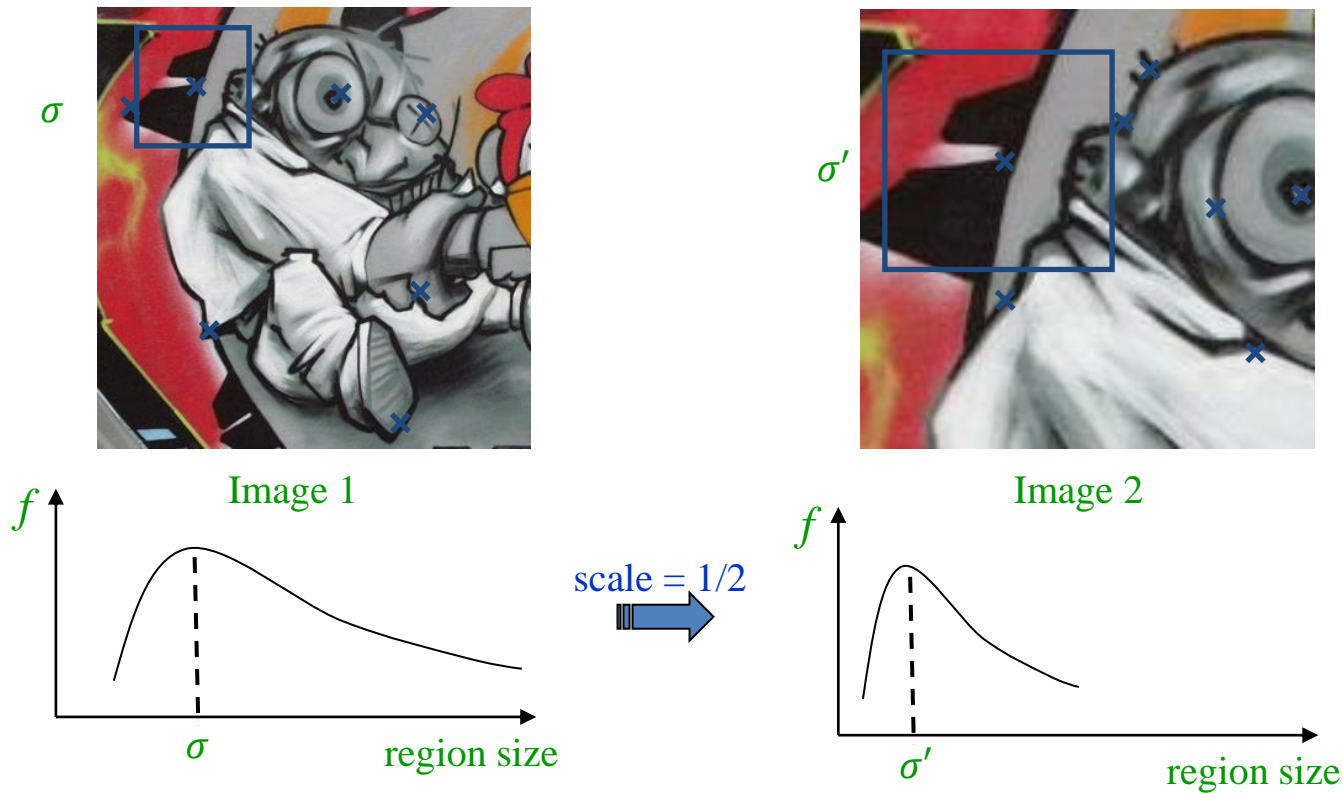


$$f(I_{i_1 \dots i_m}(x, \sigma)) = f(I_{i_1 \dots i_m}(x', \sigma'))$$

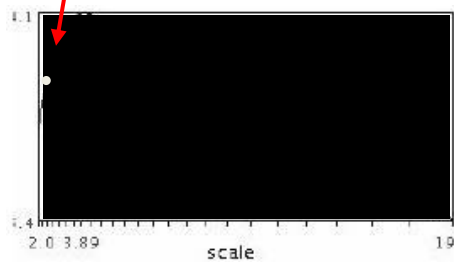
让计算机去感知检测窗口中“内容”的极值！
设计某种函数 f ，当内容相同时达到极值。

如何量化所谓的“内容”？

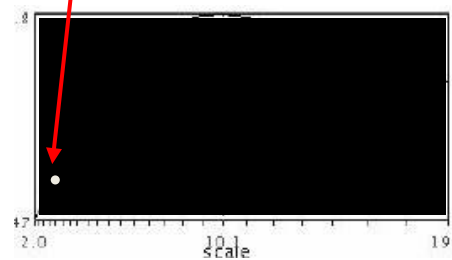
把检测窗口内的“内容”作为一个函数，把检测窗口的“尺度”作为自变量。



“窗口尺度-内容”函数的工作过程

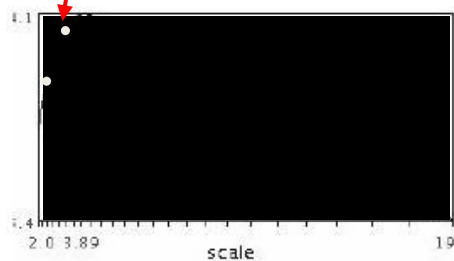


$$f(I_{i_1...i_m}(x, \sigma))$$

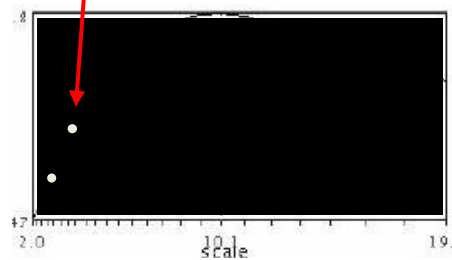


$$f(I_{i_1...i_m}(x', \sigma))$$

“窗口尺度-内容”函数的工作过程

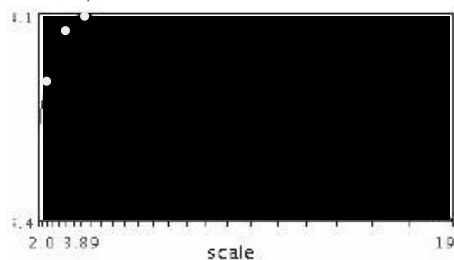


$$f(I_{i_1...i_m}(x, \sigma))$$

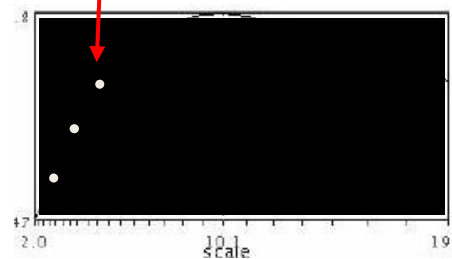


$$f(I_{i_1...i_m}(x', \sigma))$$

“窗口尺度-内容”函数的工作过程

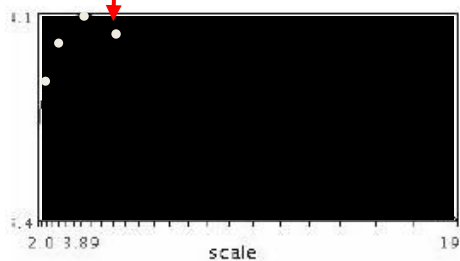
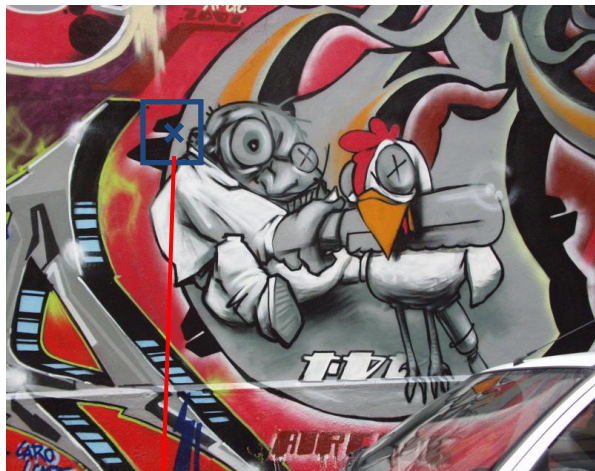


$$f(I_{i_1...i_m}(x, \sigma))$$

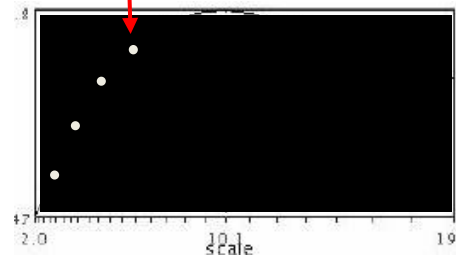


$$f(I_{i_1...i_m}(x', \sigma))$$

“窗口尺度-内容”函数的工作过程

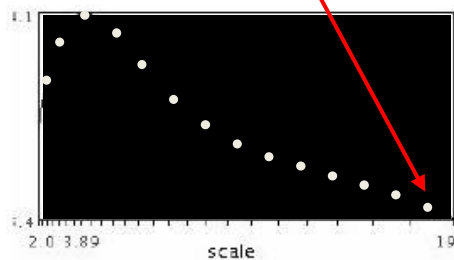


$$f(I_{i_1...i_m}(x, \sigma))$$

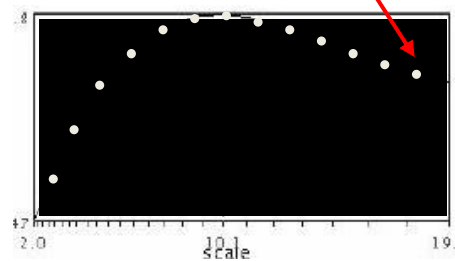


$$f(I_{i_1...i_m}(x', \sigma))$$

“窗口尺度-内容”函数的工作过程

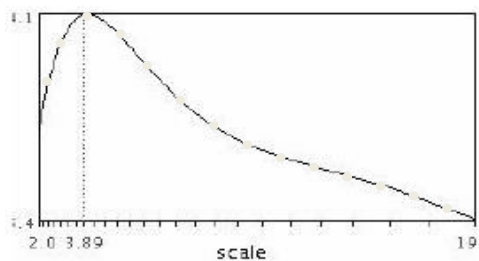


$$f(I_{i_1...i_m}(x, \sigma))$$

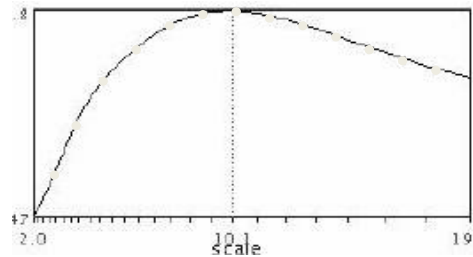


$$f(I_{i_1...i_m}(x', \sigma))$$

“窗口尺度-内容”函数的工作过程



$$f(I_{i_1 \dots i_m}(x, \sigma))$$



$$f(I_{i_1 \dots i_m}(x', \sigma'))$$

高斯拉普拉斯算子 (LoG)

- 拉普拉斯算子：对图像进行二阶微分的算子，可以突出图像中强度发生快速变化的区域。通过求取二阶导数的零交叉点来进行图像边缘检测。

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- 由于微分对噪声比较敏感，可以先对图像进行高斯平滑滤波，再使用拉普拉斯算子对图像进行二阶微分，以降低噪声的影响。常用的二维高斯函数如下：

$$G_{\sigma}(x, y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

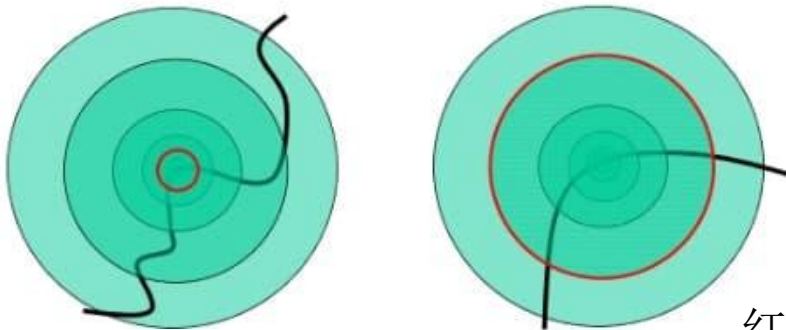
- 原图像与高斯核函数卷积后再做拉普拉斯运算：

$$\Delta[G_{\sigma}(x, y) * f(x, y)] = \underbrace{[\Delta G_{\sigma}(x, y)]}_{\text{LoG算子}} * f(x, y)$$

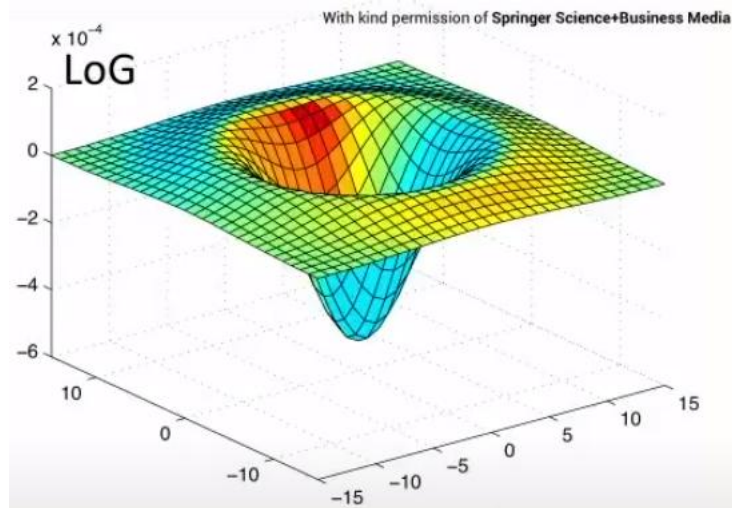
$$LoG = \Delta G_{\sigma}(x, y) = \frac{\partial^2 G_{\sigma}(x, y)}{\partial x^2} + \frac{\partial^2 G_{\sigma}(x, y)}{\partial y^2} = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} e^{-(x^2 + y^2)/2\sigma^2}$$

高斯拉普拉斯算子 (LoG)

- σ 越大, 算子尺度越大, 直观上看, 算子在三维空间中形成的那个洞的直径就越大, 在图像上的检测窗口也就越大。
- 无论图像初始尺寸多大, 都可以通过调整 LoG 算子的 σ 使窗口内的“内容”达到极值, 在达到极值的尺度下, 窗口所框出来的信息量是相同的。



红色框内信息量相同



LoG特征检测算法流程

计算不同尺度上的尺度归一化LoG函数值

$$\nabla^2_{\text{norm}} L(x, y, \sigma_D) = \sigma_D^2 \left(\frac{\partial^2 G(x, y, \sigma_D)}{\partial x^2} + \frac{\partial^2 G(x, y, \sigma_D)}{\partial y^2} \right) * I(x, y)$$



同时在位置和尺度构成的三维空间上寻找
尺度归一化LoG的极值点



进行非极大值抑制，减少重复检测

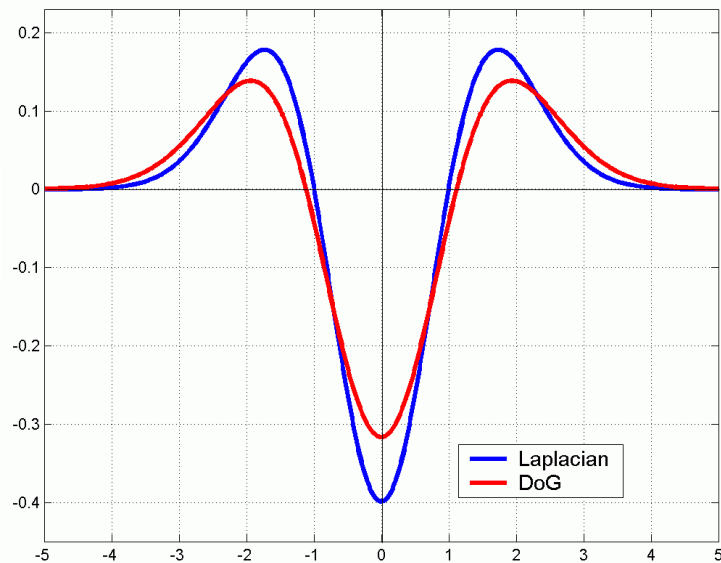
高斯差分算子 (Difference of Gaussians, DoG)

- DoG: 高斯函数的差分, 具体到图像中, 就是将图像在不同参数下的高斯滤波结果相减, 得到差分图。

$$DoG = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}} \left[\frac{1}{\sigma_1} e^{-(x^2+y^2)/2\sigma_1^2} - \frac{1}{\sigma_2} e^{-(x^2+y^2)/2\sigma_2^2} \right]$$

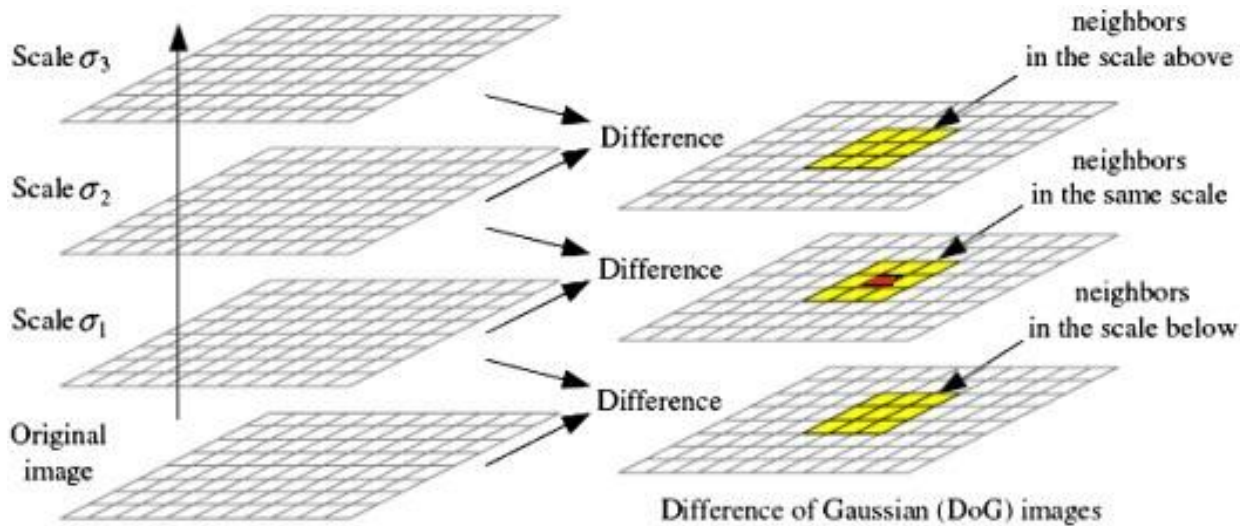
$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G$$

- DoG相比LoG计算更简单, 极值点对应的自变量相同, 因此常用DoG替代LoG。



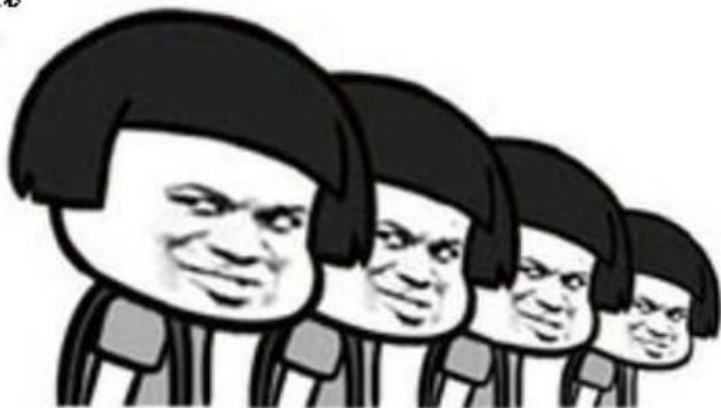
DoG角点检测算法流程

1. 将同一图像在三个不同参数下进行高斯滤波。
2. 将相邻参数的高斯滤波图两两相减，得到差分图。
3. 求中间尺度对应的差分图像的每个像素与该像素同尺度的8个相邻点以及上下相邻尺度对应的9*2共26个点的极值。如果该点像素值为最大值或最小值，就认为该点是图像在该尺度下的一个特征点。



参考资料

<https://zhuanlan.zhihu.com/p/144875290>



等忙完这一阵
就可以接着忙下一阵了

知乎 @小小

Main questions

- 特征点从何而来？
 - 怎么在多幅图像中检测相同的、显著性的特征点？
- 如何描述局部特征使其保证独特性？
- 如何建立特征点之间的对应关系？, i.e., 如何计算匹配？

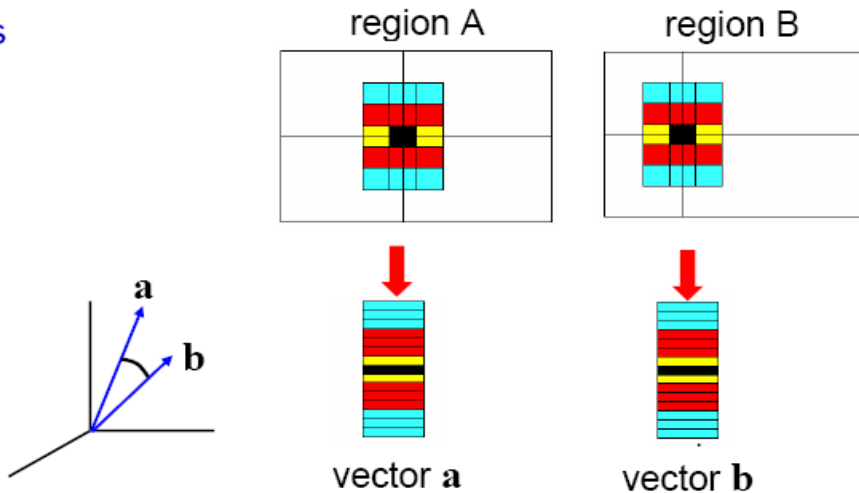
特征描述子 (Feature Descriptor)

最简单的方式:

- 检测窗口内的像素值有序排列作为特征描述子

Write regions as vectors

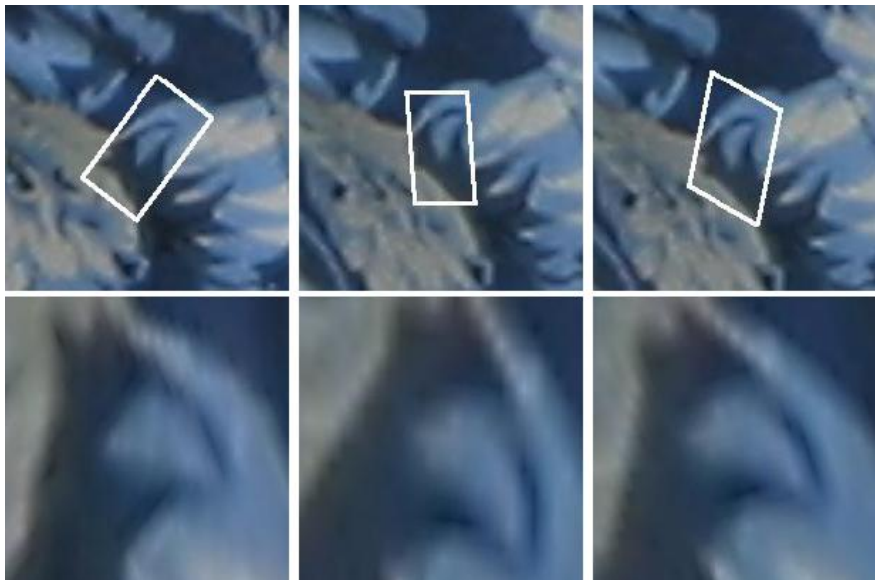
$A \rightarrow \mathbf{a}$, $B \rightarrow \mathbf{b}$



特征描述子（Feature Descriptor）

最简单的方式：

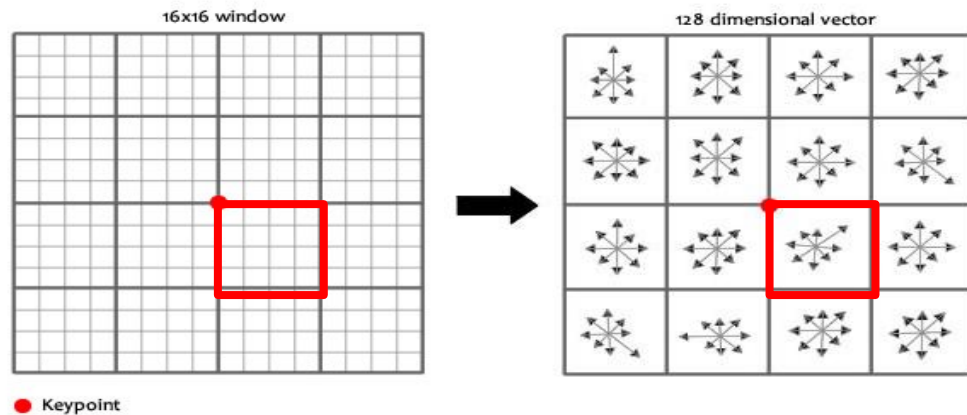
- 检测窗口内的像素值有序排列作为特征描述子
- 缺点：微小的移动会对匹配产生巨大的影响！



SIFT方法 (Scale Invariant Feature Transform)

步骤:

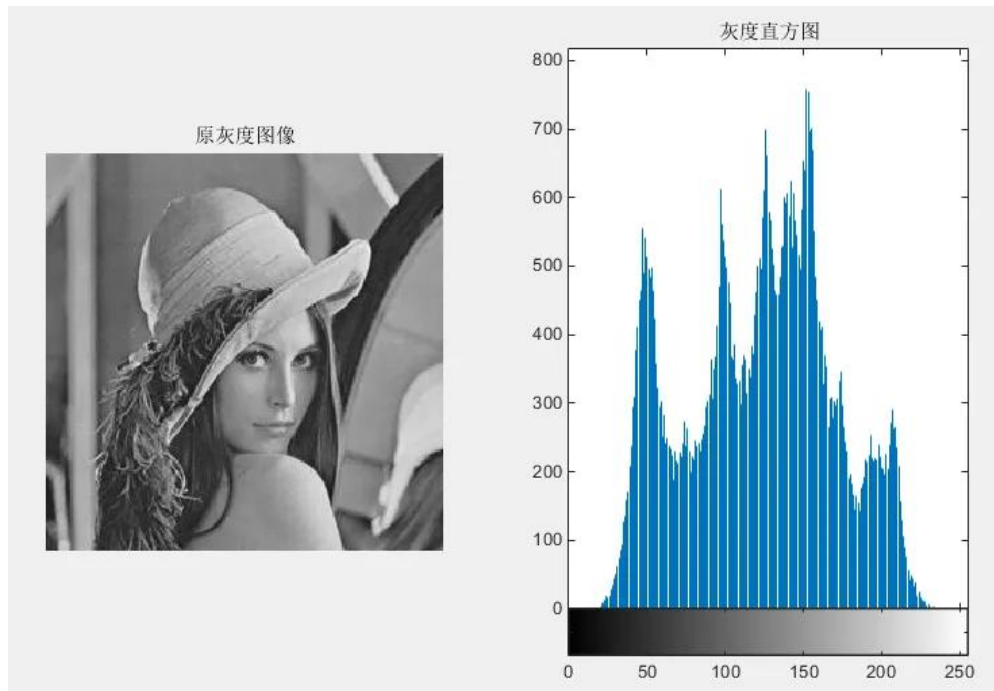
1. 以特征点为中心取 16×16 的窗口
2. 将窗口进一步分成16个 4×4 的图像小块
3. 在每一个图像小块中计算方向梯度直方图，含有8个梯度方向的向量信息
4. 形成 $4 \times 4 \times 8 = 128$ 维的特征向量



什么是直方图

- 直方图：一种统计报告图，横轴表示数据类型，纵轴表示分布情况

灰度直方图：将数字图像中的所有像素，按照灰度值的大小，统计其出现的频率



什么是方向梯度直方图

1. 计算区块内每个像素的梯度，箭头长度表示梯度大小，箭头方向表示梯度方向

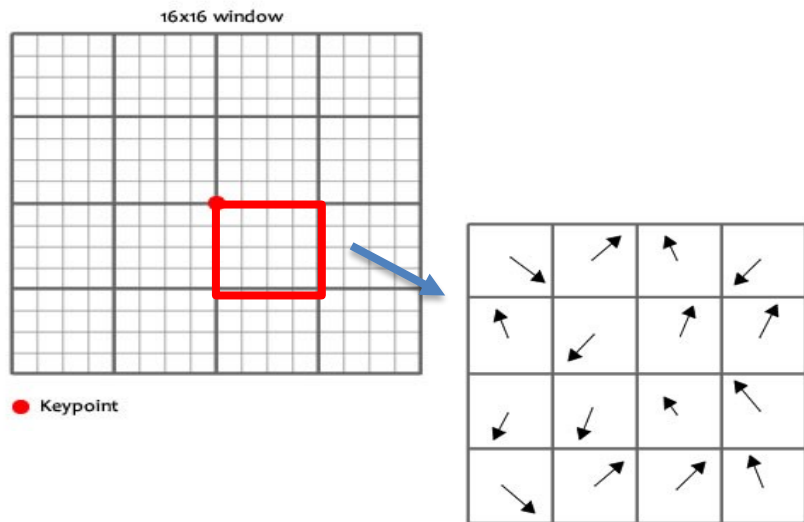
1) 计算图像中某个像素点 (x, y) 的水平方向梯度和垂直方向梯度：

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y)$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1)$$

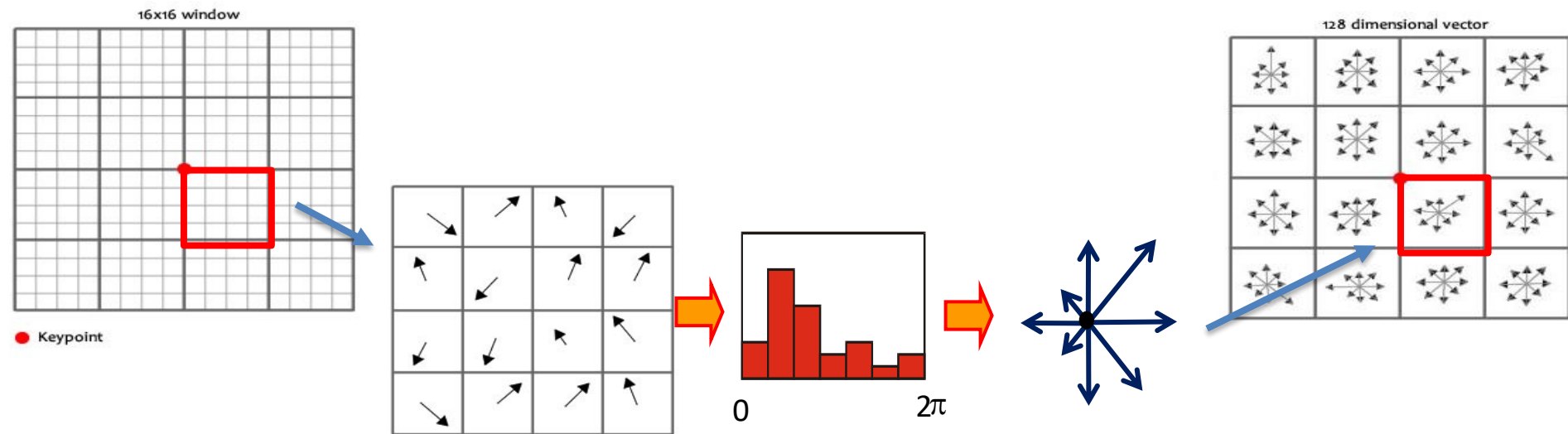
2) 通过该像素点的水平梯度和垂直梯度计算其梯度大小和方向：

$$G(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)}$$
$$\theta(x, y) = \tan^{-1} \frac{G_y(x, y)}{G_x(x, y)}$$



什么是方向梯度直方图

1. 计算区块内每个像素的梯度，箭头长度表示梯度大小，箭头方向表示梯度方向
2. 将 360° 按照每隔 45° 划分为8个区间
3. 将属于每个区间的所有像素梯度值累加



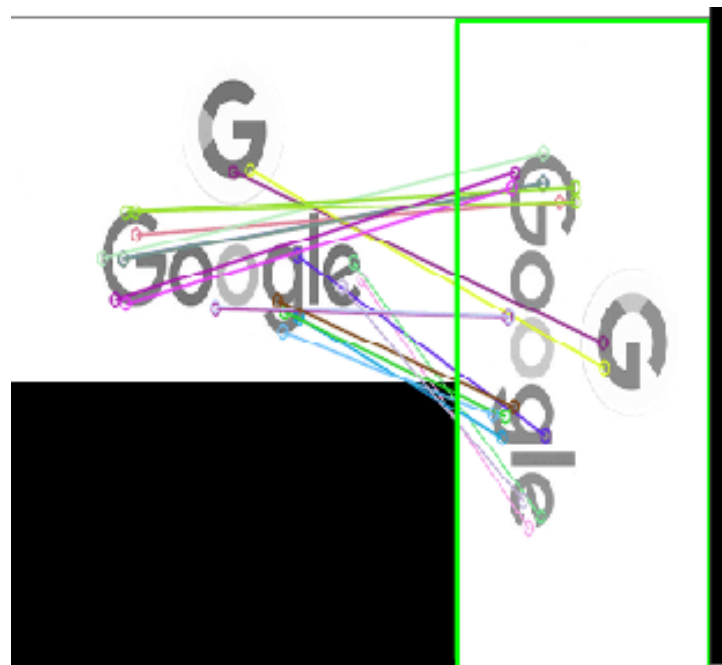
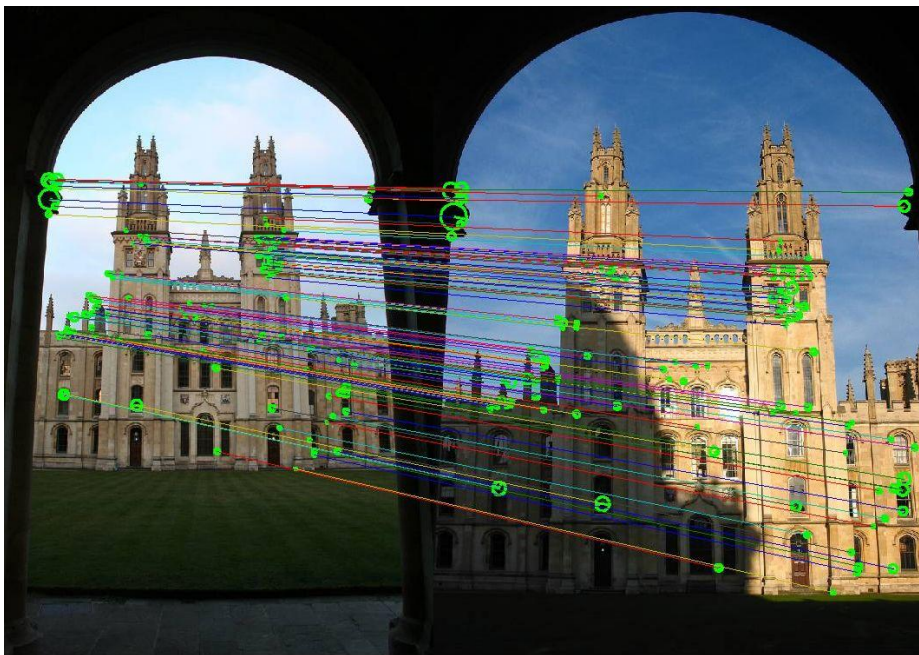
SIFT的特性

- 可以应对视角变化：能够识别超过 60° 的视角变化
- 能够应对显著的光线变化：有时甚至可以将同一场景的白天和黑夜图像进行匹配
- 有许多可用的代码：

<https://github.com/deepanshut041/feature-detection>

<https://blog.csdn.net/wangxinsheng0901/article/details/79676081>

基于SIFT的图像匹配结果



图片来源：谷歌图片

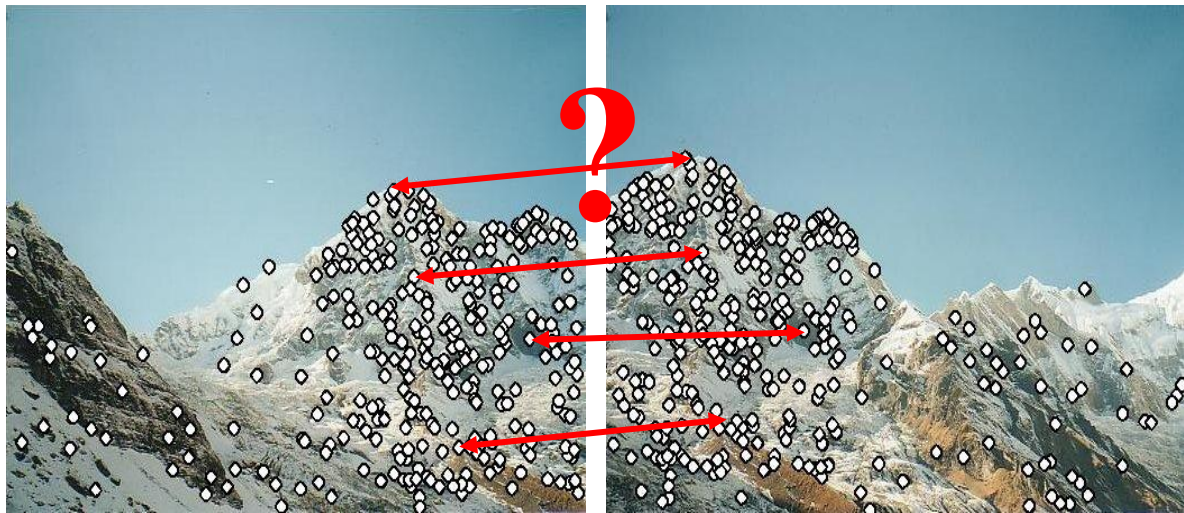
Main questions

- 特征点从何而来？
 - 怎么在多幅图像中检测相同的、显著性的特征点？
- 如何描述局部特征使其保证独特性？
- 如何建立特征点之间的对应关系？, i.e., 如何计算匹配？

特征匹配

给定图像1中的一个特征点及其特征向量，如何在图像2中找到最佳匹配？

1. 定义特征向量距离函数
2. 对于图1中的每一个特征点，与图2中的所有特征点的特征向量计算距离函数，距离最小的特征点为最佳匹配

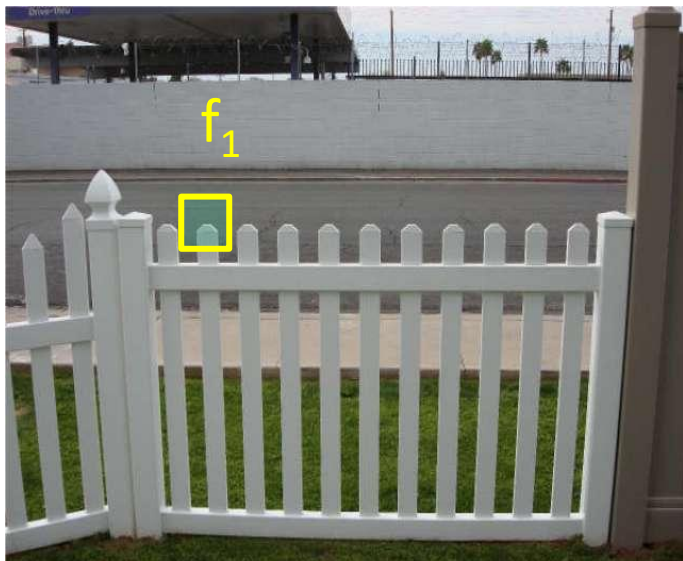


如何定义特征向量距离函数？

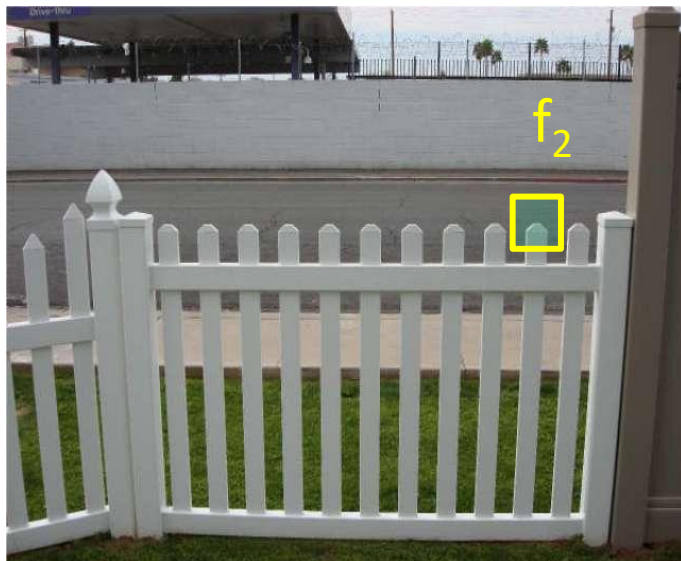
最简单的度量方式：平方差和 (Sum of Squared Differences, SSD)

特征向量对应维度计算平方差，而后求和

缺点：可能存在错误匹配，如下图所示



I_1



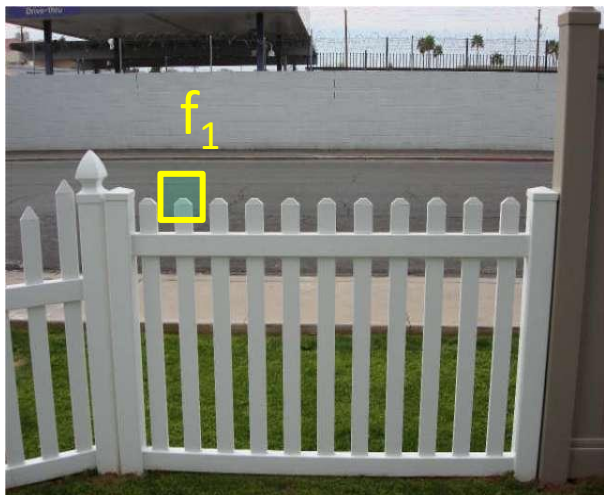
I_2

如何定义特征向量距离函数？

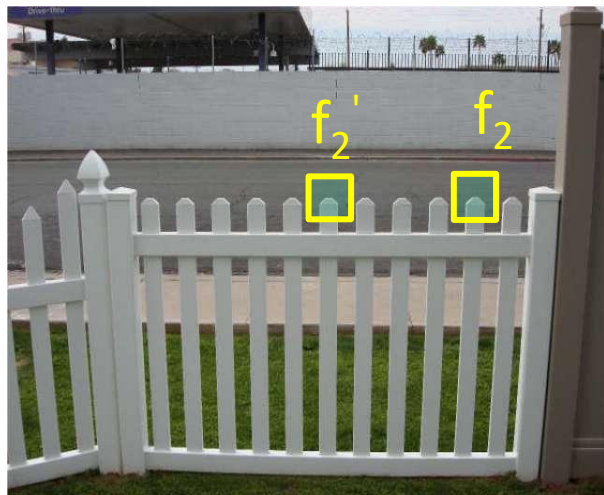
最简单的度量方式：平方差和 (Sum of Squared Differences, SSD)

更好的方式：ratio distance = $SSD(f_1, f_2) / SSD(f_1, f_2')$

- $SSD(f_1, f_2)$ 为最佳匹配对应的最小距离； $SSD(f_1, f_2')$ 为次优匹配对应的次接近距离
- 如果ratio distance很大，意味着将 f_2 作为最佳匹配的可信度不高，可以选择舍弃

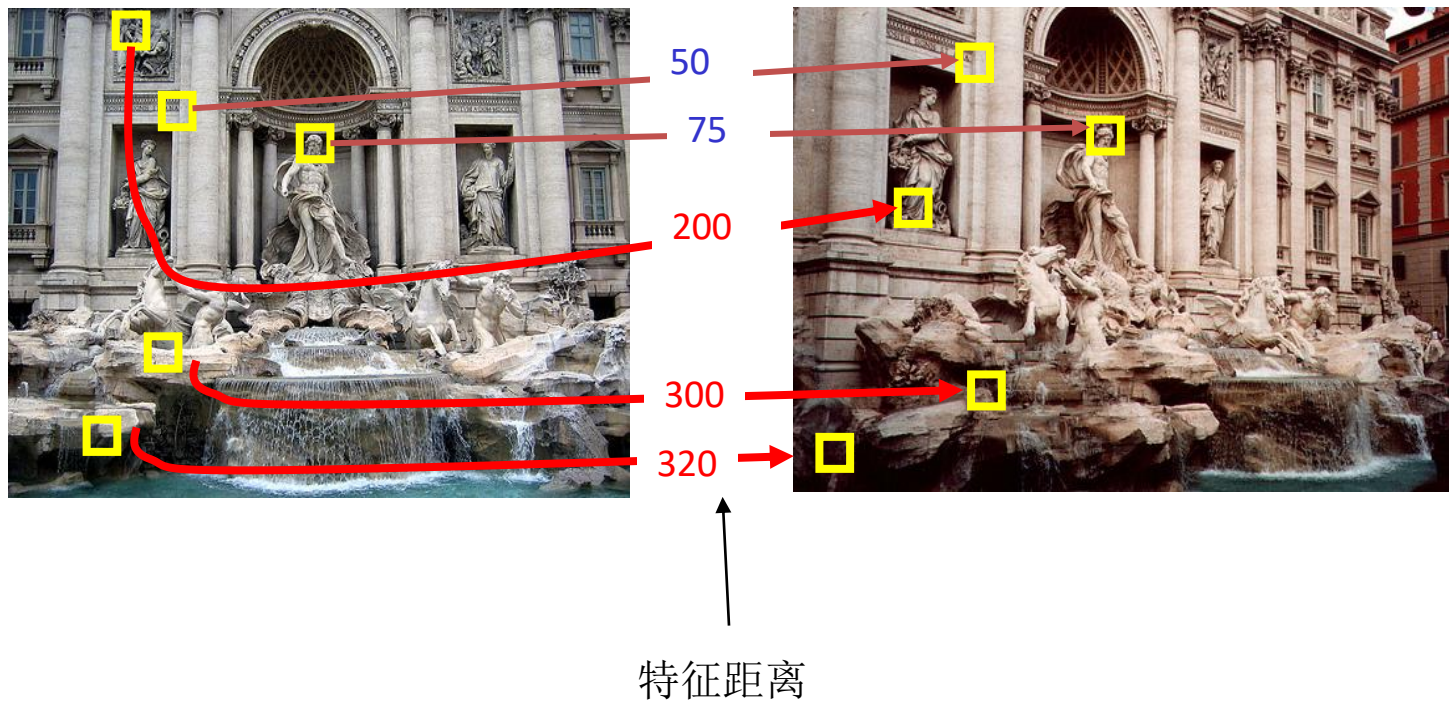


I_1

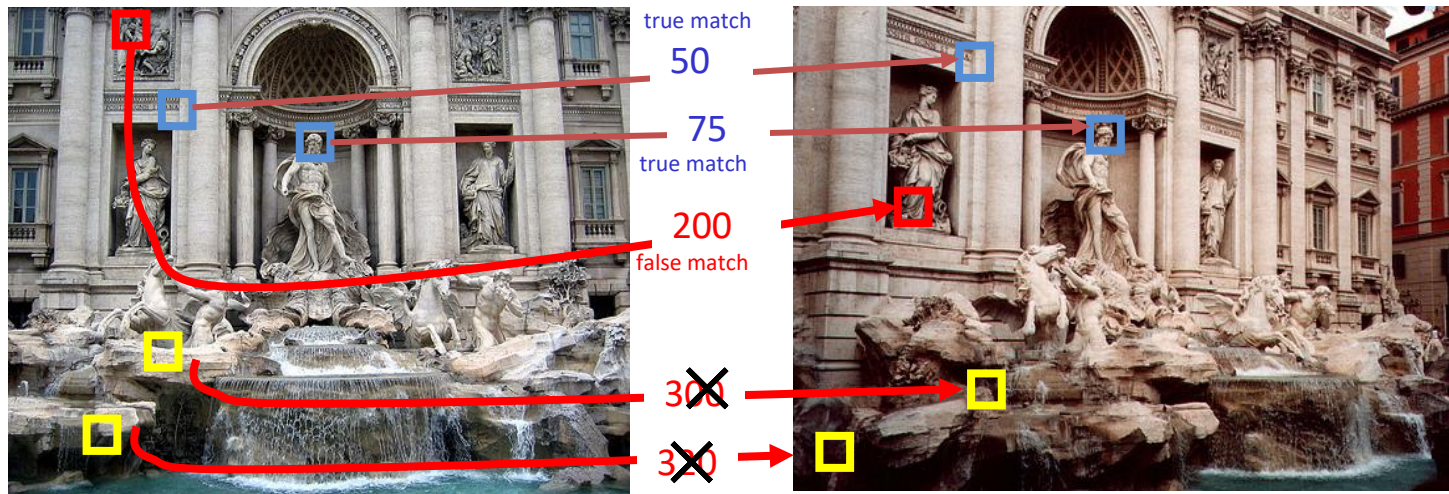


I_2

如何评价特征匹配的好坏？



如何评价特征匹配的好坏？



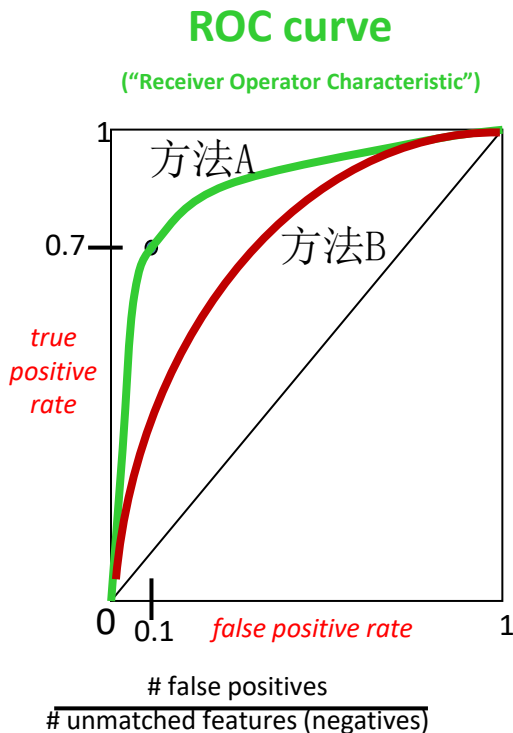
通过设置特征距离阈值确定保留哪些互相匹配的特征点
阈值的设置直接影响匹配的性能：

- True positives = 所得匹配中正确匹配的数量（如蓝色标记特征点）
- False positives = 所得匹配中错误匹配的数量（如红色标记特征点）

特征匹配方法性能评价

1. 计算不同阈值下的真阳率和假阳率，连成ROC曲线
2. 计算不同的特征匹配方法曲线下面积AUC
3. AUC面积越大性能越好：方法A优于方法B

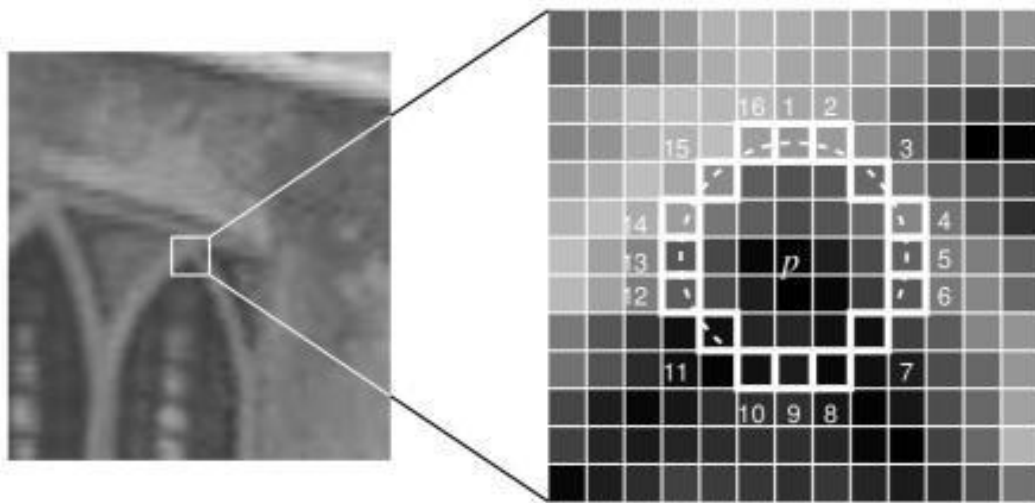
$$\frac{\text{\# true positives}}{\text{\# matching features (positives)}}$$



如何提高检测速度，保证实时性？

FAST特征点检测

- 从图像中选取一个像素点 P ，其强度为 I_p ；
- 考虑以该像素点为中心的一个半径等于3个像素的离散化的Bresenham圆，这个圆的边界上有16个像素；
- 如果在这个大小为16个像素的圆上有连续 N 个像素 (N 一般取9或12) 都比候选点 P 的强度高或者比 P 的强度暗，那么它就是一个角点。

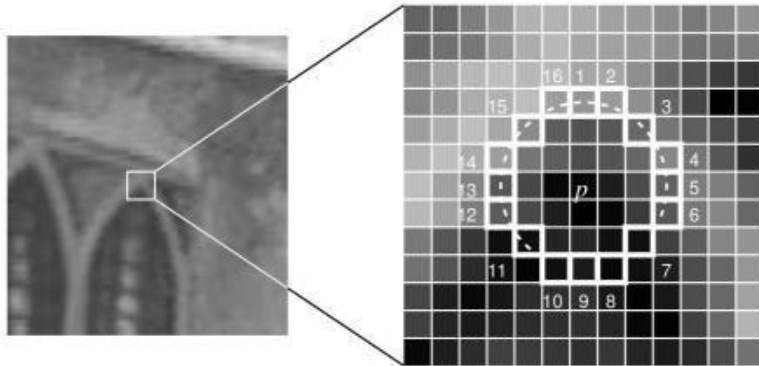


FAST特征点检测

一种更快速的判断方式：仅检查位置在1、9、5、13这四个位置的像素可以快速排除大部分非角点的像素！

- 首先检查位置1和位置9，如果它们都比 I_p 大 (或者小)，再检测位置5和位置13；
- 如果位置5和位置13中至少有一个比 I_p 大 (或者小)，则该点是候选角点，否则剔除。
- 对于成为候选的角点，再做完整的测试，即检测圆上的所有点。

原理说明：在 $N=12$ 的情况下，如果 P 是一个角点，超过四分之三圆的像素值必须大于或小于 I_p 。因此，1、9、5、13这四个像素中至少有3个必须大于或小于 I_p 。



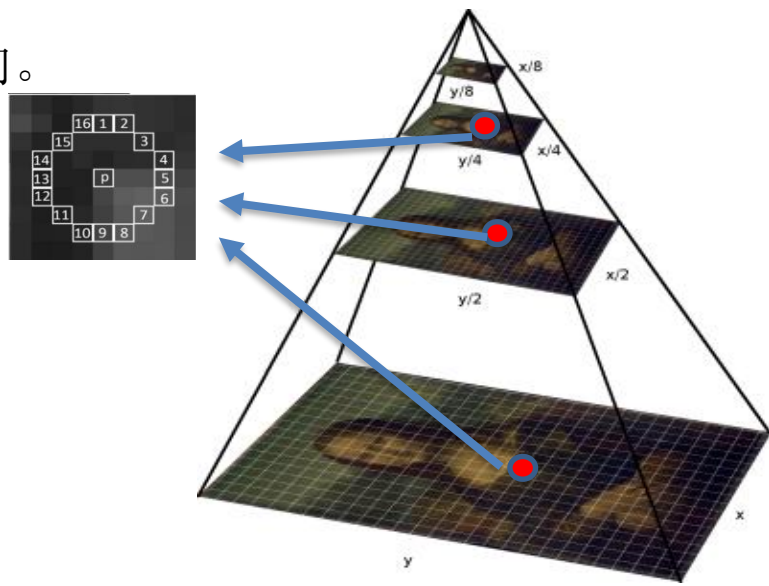
ORB特征检测

ORB算法性能与SIFT一样好，速度却快了两个数量级：

1. 构建图像特征金字塔；
2. 在不同尺度上检测FAST特征点 (为了让检测到的角点具有尺度不变性)；
3. 对FAST提取到的角点计算Harris角点响应值，选取前N个具有最大响应值的角点作为特征点集合；
4. 非极大值抑制去除局部较密集特征点；
5. 使用矩（moment）法来确定FAST特征点的方向。
(为了让检测到的特征具有旋转不变性)

图像金字塔的构建过程：

- 原始图像作为金字塔的最底层；
- 对第 k 层图像进行高斯滤波，以一定的采样率对其进行降采样，生成第 $k + 1$ 层。



ORB特征检测

ORB算法性能与SIFT一样好，速度却快了两个数量级：

5. 使用矩（moment）法来确定FAST特征点的方向。

(为了让检测到的特征具有旋转不变性)

1、在一个小的图像块 B 中，定义图像块的矩。

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y)$$

2、通过矩可以找到图像块的质心

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

3、连接图像块的几何中心 O 与质心 C，得到一个方向向量 \overrightarrow{OC} ，这就是特征点的方向

$$\theta = \text{atan2}(m_{01}, m_{10})$$



THANKS