Feature detection and matching

Bing Gong
（巩冰）
gongbing@shnu.edu.cn

Outline

- Points and patches
  - Feature detectors
  - Feature descriptors
  - Feature matching
- Lines
  - Hough transforms
  - RANSAC

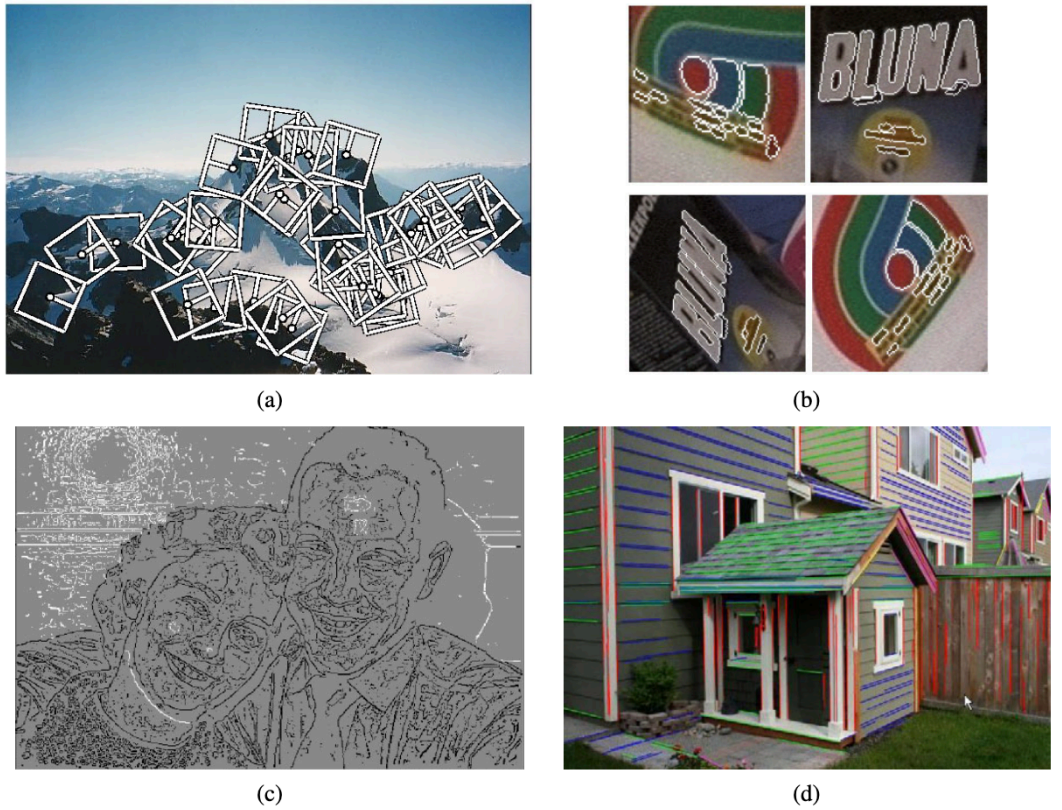


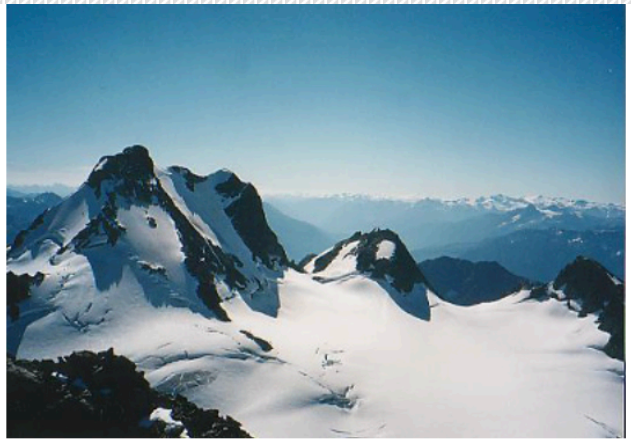
Figure 4.1 A variety of feature detectors and descriptors can be used to analyze, describe and match images: (a) point-like interest operators (Brown, Szeliski, and Winder 2005) © 2005 IEEE; (b) region-like interest operators (Matas, Chum, Urban *et al.* 2004) © 2004 Elsevier; (c) edges (Elder and Goldberg 2001) © 2001 IEEE; (d) straight lines (Sinha, Steedly, Szeliski *et al.* 2008) © 2008 ACM.

Points and patches

- Where would you tell your friend to meet you

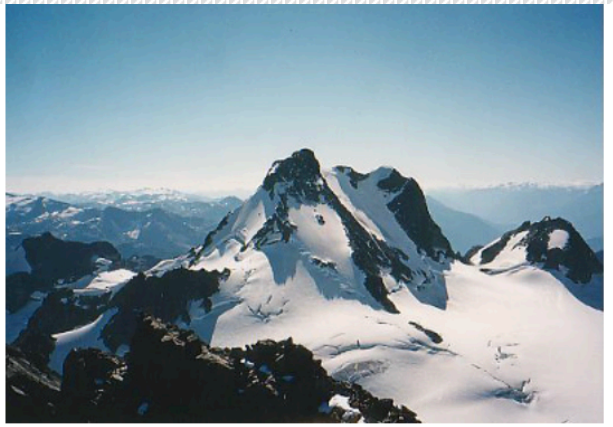
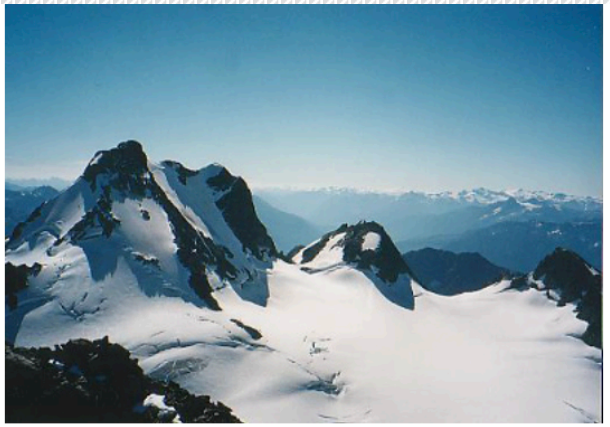
Points and patches

- What features are needed if you want to match two images for the two examples?

Points and patches

- What features are needed if you want to match two images for the two examples?
  - Mountain peaks,
  - Building corners,
  - Doorways,
  - Interestingly shaped patches of snow

These kinds of localized feature are often called *keypoint features* or *interest points*

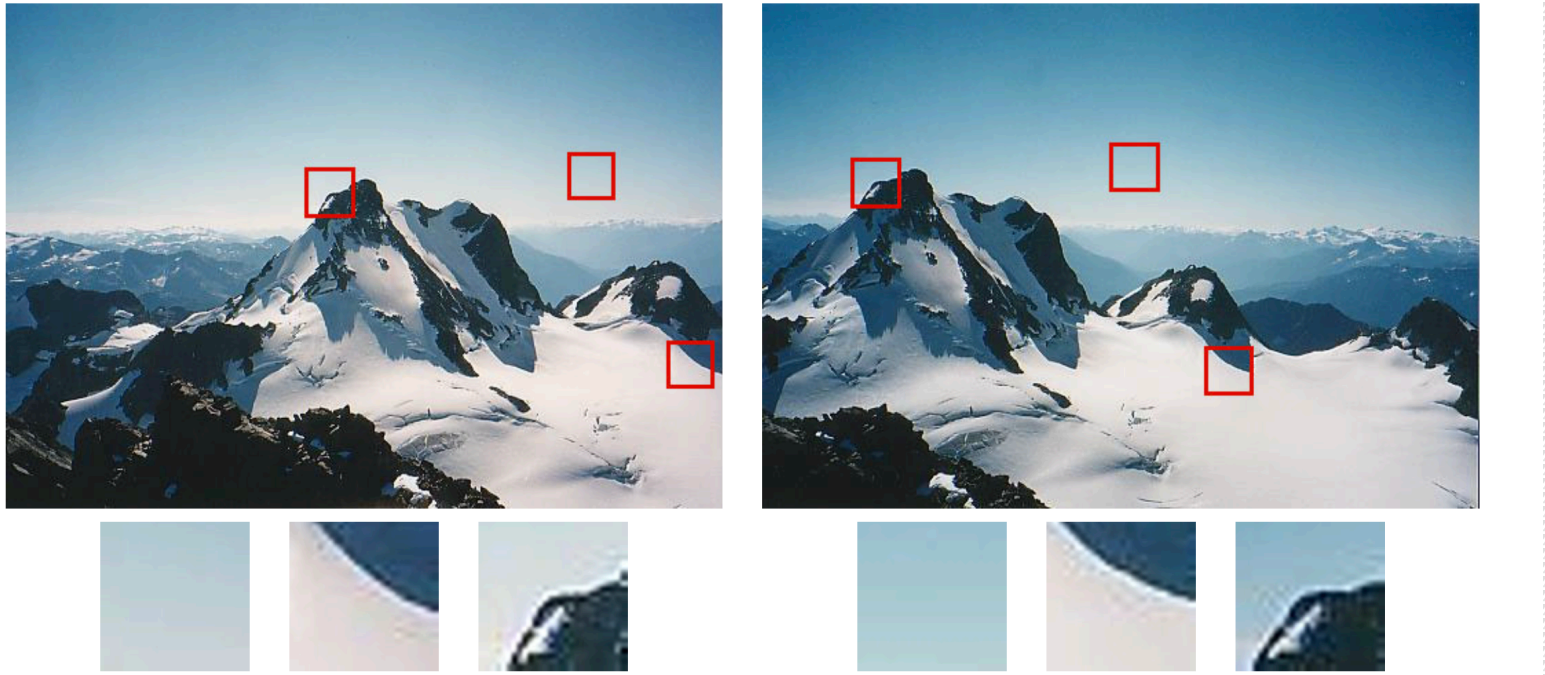
Points and patches

- What features are needed if you want to match two images for the two examples?
  - Edges
  - Lines

  These kinds of features can be matched based on their *orientation and local appearance (edge profiles)* and can also be good indicators of object boundaries and occlusion events in image sequence.

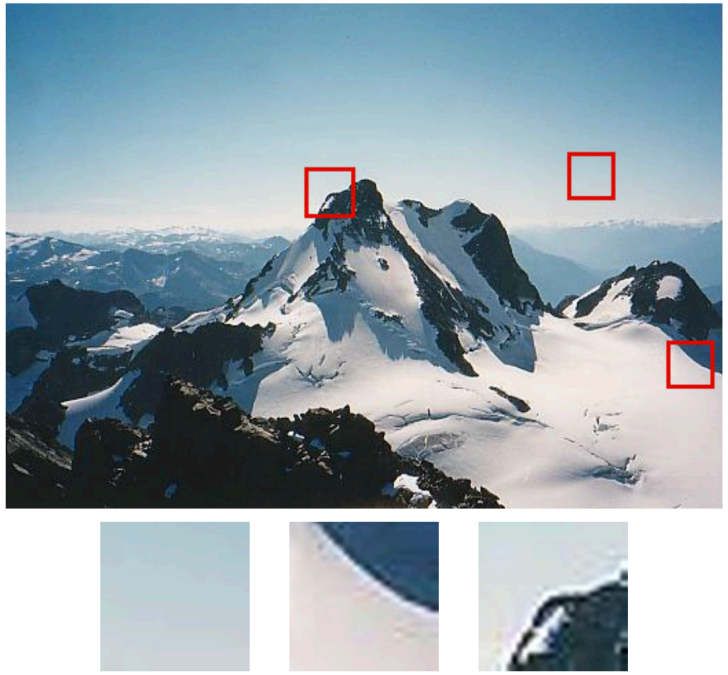
Points and patches

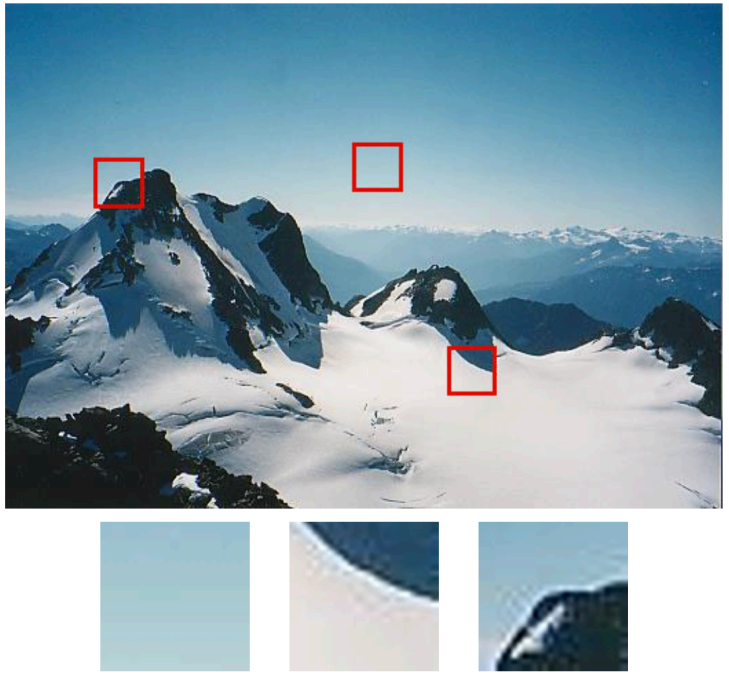
- How can we find image locations where we can reliably find correspondences with other images, i.e., what are good features to track in the following image?



Feature detectors

- How can we find image locations where we can reliably find correspondences with other images, i.e., what are good features to track in the following image?





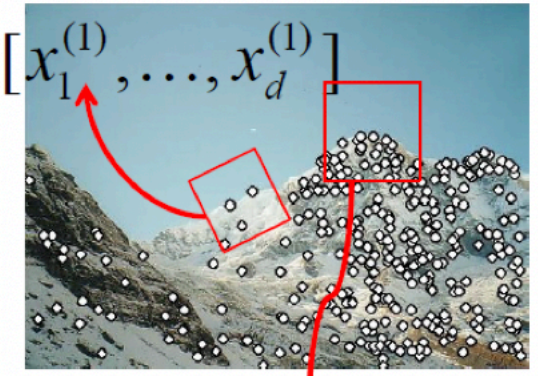
✗ Textureless patches

✓ Large contrast changes (gradients)

Points and patches

- Keypoint detection and matching pipeline:
  - (1) Feature detection (extraction) stage:
    - Search for the locations that are likely to match well in other images **特征检测/提取**
  - (2) Feature description stage:
    - Each region around detected keypoint locations is converted into a more compact and stable (invariant) *descriptor* that can be matched against other descriptors **特征描述**
  - (3) Feature matching stage:
    - Searches for likely matching candidates in other images **特征匹配**

  Alternatively for (3) Feature tracking for vide processing.



Feature detectors

- Patches with large contrast changes (gradients) are easier to localize. 对比度大的容易定位
  - **Straight line** segments at a single orientation **suffer from the aperture problem**. 直线定位的孔径问题
  - Patches with gradients in **at least two (significantly) different orientations** are the easiest to localize 有两个以上明显梯度变化的图像块最容易定位.

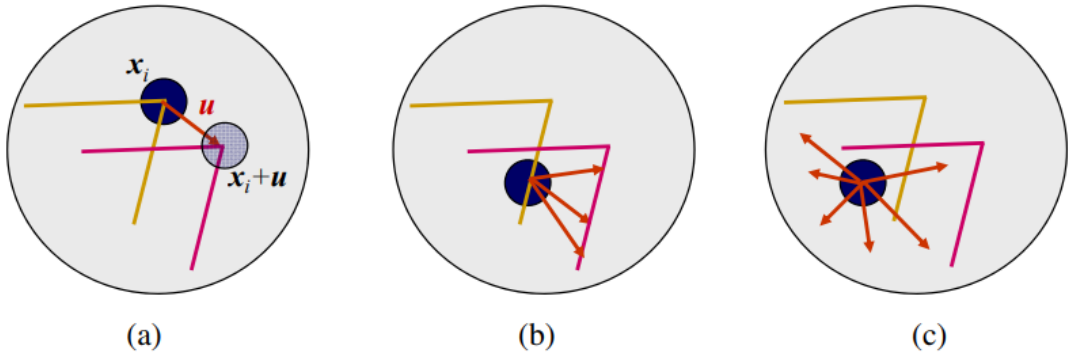


Figure 7.4 *Aperture problems for different image patches: (a) stable ("corner-like") flow; (b) classic aperture problem (barber-pole illusion); (c) textureless region. The two images I_0 (yellow) and I_1 (red) are overlaid. The red vector* $\mathbf{u}$ *indicates the displacement between the patch centers and the $w(\mathbf{x}_i)$ weighting function (patch window) is shown as a dark circle.*

a) "角点" – 稳定的，各方面都发生了重大变化； (b) "边 (edge)" – 经典的孔径问题，沿边缘方向没有变化； © "平坦/无纹理的区域 (flat region)" – 各方向都没有变化。从上面的描述我们认为通过移动一个小窗口会导致窗口中图像灰度变化剧烈，那么这个窗口中易于识别的特征是我们认为的角点。

Feature detectors

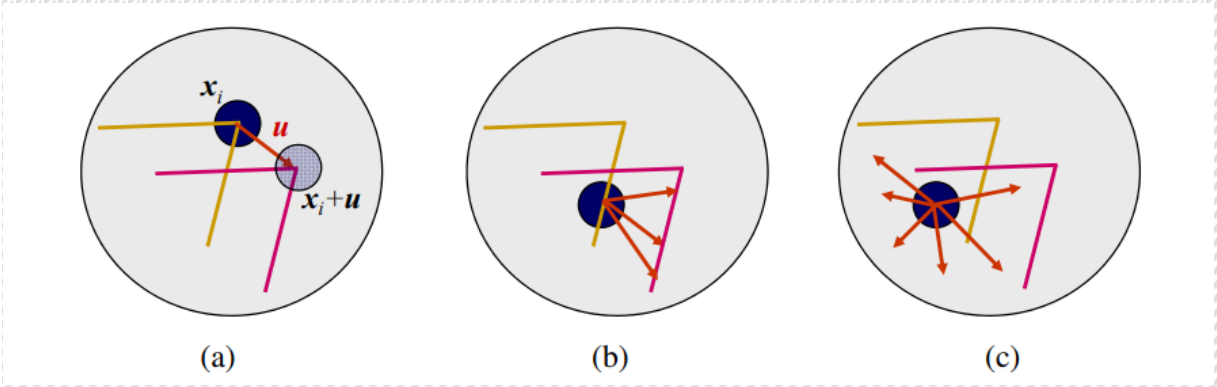
- These intuitions can be formalized by looking at the simplest possible matching criterion for comparing two image patches, i.e., their (weighted) summed square difference用加权差的平方和形式化表述:

$$EWSSD(u) = \sum_i w(x_i)[I_1(xi + u) - I_0(x_i)]^2$$

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

Window function | Shifted intensity | Intensity

where I_0 and I_1 are the two images being compared, $\mathbf{u} = (u, v)$ is the *displacement* vector, $w(\mathbf{x})$ is a spatially varying weighting (or window) function, and the summation i is over all the pixels in the patch. Note that this is the same formulation we later use to estimate motion



Feature detectors

- These intuitions can be formalized by looking at the simplest possible matching criterion for comparing two image patches, i.e., their (weighted) summed square difference用加权差的平方和形式化表述:

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

Window function

Shifted intensity

Intensity

这是一个二元函数，其要素解释如下：

> 目标因变量是窗口内整体像素值变化
> 自变量为：
> 窗口向水平方向移动的像素距离（u)
> 窗口向垂直方向移动的像素距离（v)
> x和y的范围是窗口的长和宽
> $I(x,y)$ 表示窗口内 (x,y) 处的像素值，$I(x+u,y+v)$ 表示原窗口中 (x,y) 处像素在窗口移动后对应位置的像素值
> $w(x,y)$ 表示窗口内 (x,y) 处的位置权重（这体现了视觉的注意力特征）

Feature detectors

- Auto-correlation function or surface (自相关函数或自相关表面)

$$E_{\mathrm{AC}}(\Delta\mathbf{u}) = \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i + \Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

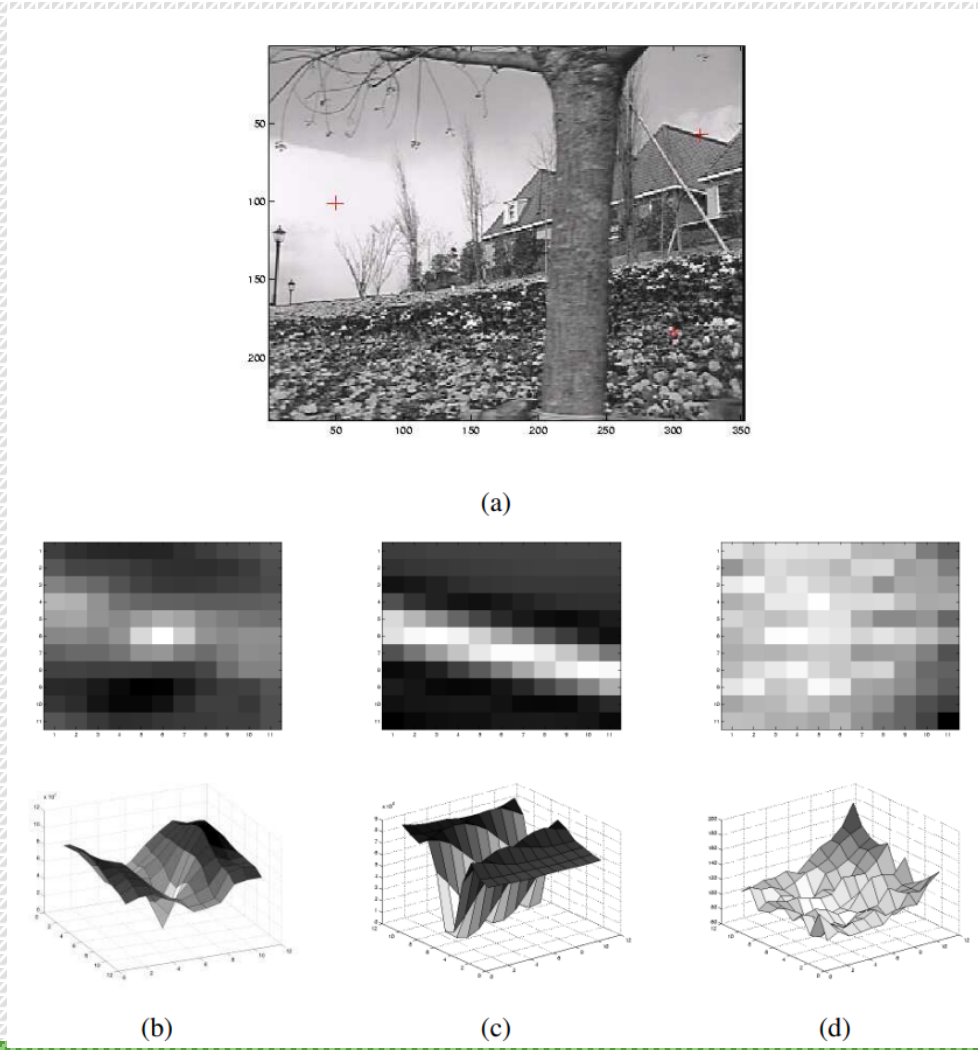
$$\approx \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i)\cdot\Delta\mathbf{u} - I_0(\mathbf{x}_i)]^2$$

$$= \sum_i w(\mathbf{x}_i)[\nabla I_0(\mathbf{x}_i)\cdot\Delta\mathbf{u}]^2$$

$$= \Delta\mathbf{u}^T\mathbf{A}\Delta\mathbf{u},$$

The auto-correlation matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix},$$



Feature detectors

- Auto-correlation function or surface (自相关函数或自相关表面)

$$E_{\mathrm{AC}}(\Delta\mathbf{u}) = \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i + \Delta\mathbf{u}) - I_0(\mathbf{x}_i)]^2$$

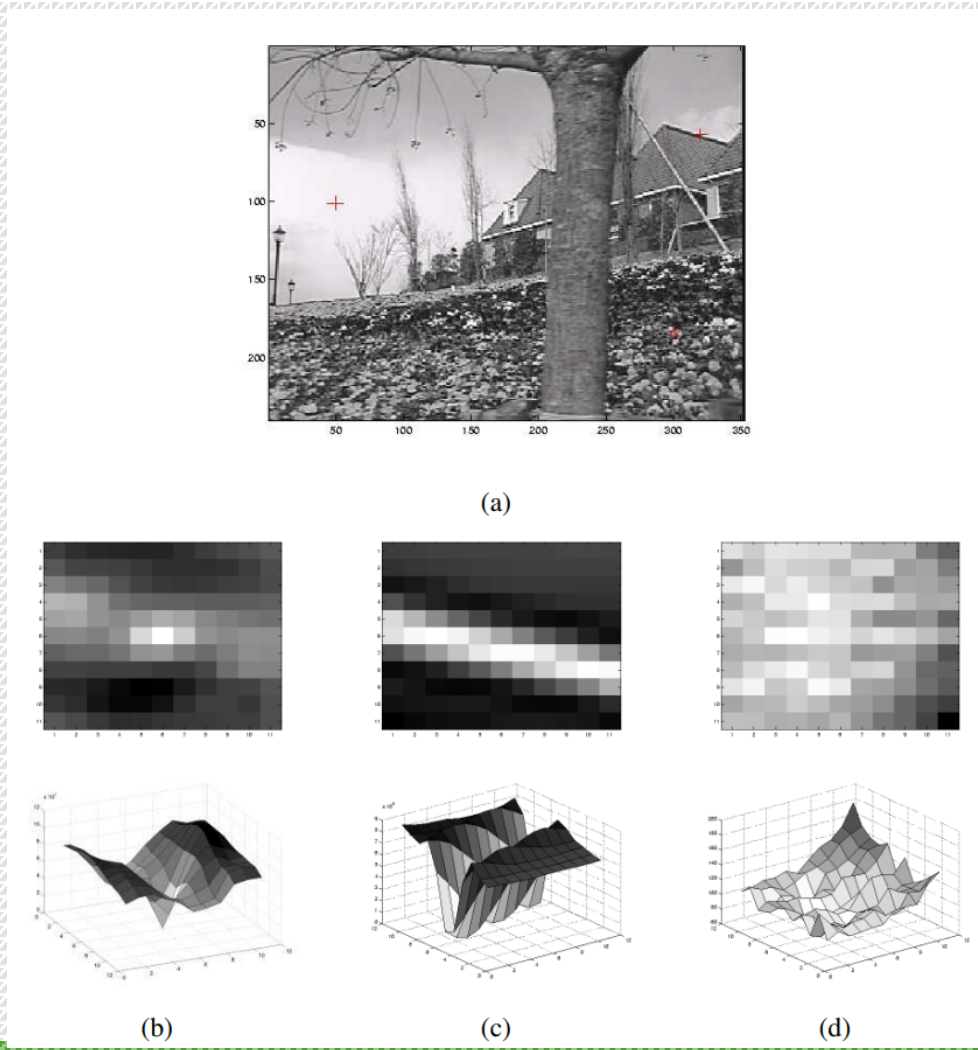
$$\approx \sum_i w(\mathbf{x}_i)[I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i)\cdot\Delta\mathbf{u} - I_0(\mathbf{x}_i)]^2$$

$$= \sum_i w(\mathbf{x}_i)[\nabla I_0(\mathbf{x}_i)\cdot\Delta\mathbf{u}]^2$$

$$= \Delta\mathbf{u}^T\mathbf{A}\Delta\mathbf{u},$$

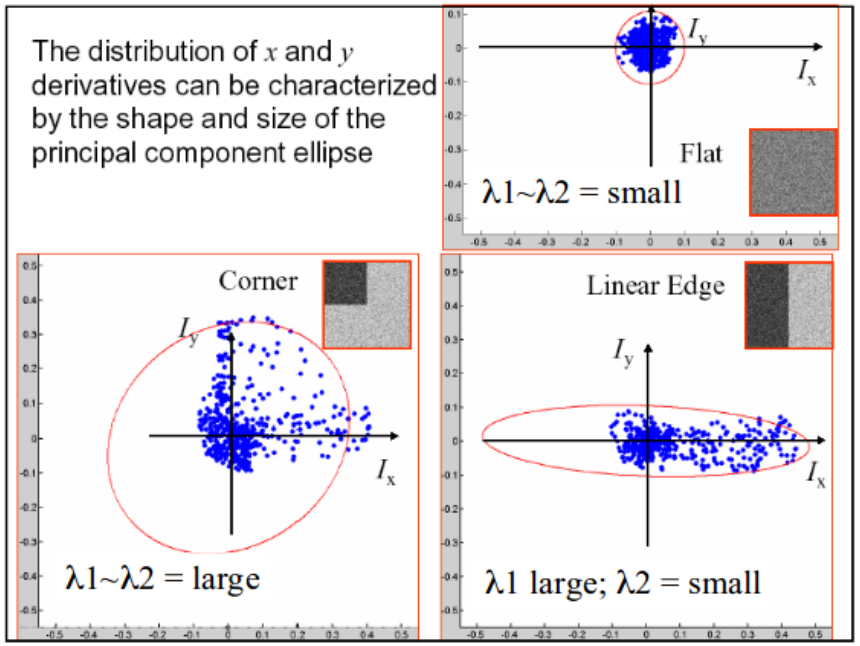
The auto-correlation matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix},$$



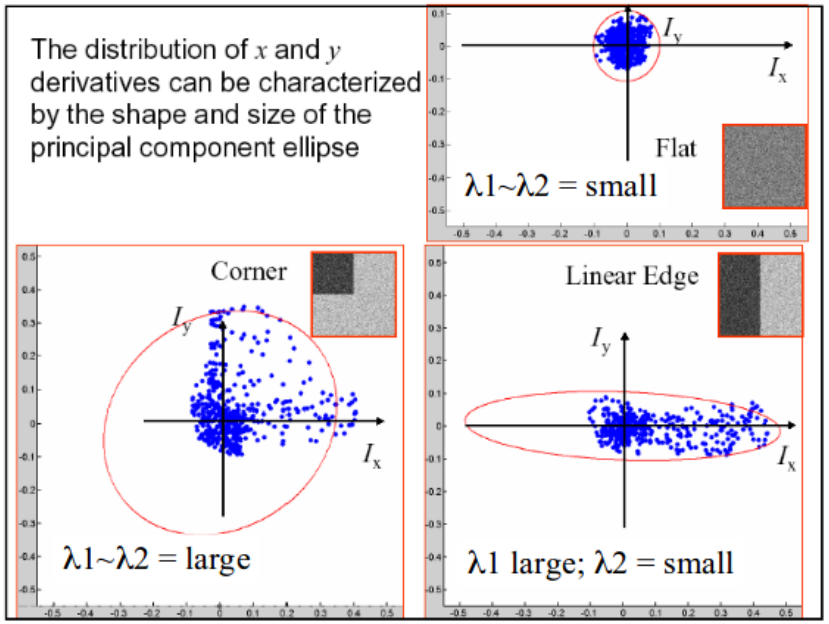
Feature detectors

- 对自相关矩阵进行特征值分析得到两个特征值（λ_0，λ_1）和两个特征向量方向
- 较大不确定度取决于较小特征值特征值（ $\lambda_0^{-1/2}$ ），特征值大，半轴短。*(Shi and Tomasi 1994)*



https://zhuanlan.zhihu.com/p/354749490
https://www.zhihu.com/question/484256180

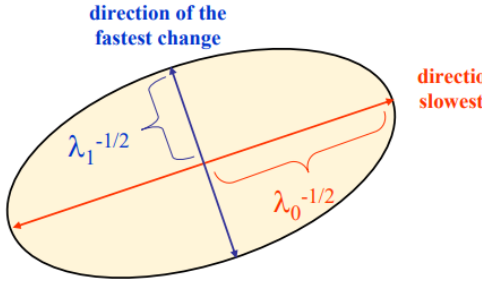
rs



征值分析得到两个特征值（λ_0，λ_1）和两个特征

较小特征值特征值（ $\lambda_0^{-1/2}$ ），特征值大，半轴

si 1994)

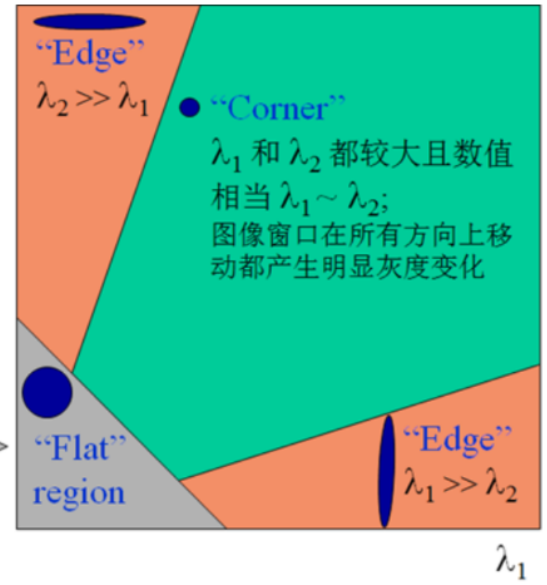


边界：一个特征值大，另一个特征值小，λ1≫λ2λ1≫λ2或λ2≫λ1。自相关函数值在某一方向上大，在其他方向上小。

平面：两个特征值都小，且近似相等；自相关函数数值在各个方向上都小。

角点：两个特征值都大，且近似相等，自相关函数在所有方向都增大。

如果λ_1 和 λ_2 都很小，图像窗口在所有方向上移动都无明显灰度变化



https://zhuanlan.zhihu.com/p/354749490

https://www.zhihu.com/question/484256180

Feature detectors

- Harris 角点检测算法
  - 利用水平，竖直差分算子对图像 I（x, y）的每个像素进行滤波以求得 X 和 Y方向的梯度 Ix和Iy，进而求得M中的四个元素的值。

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

  - 使用M计算对应每个像素的角点响应函数R，其中角点响应函数R 为：

$$R = \frac{[I_x^2 * I_y^2 - (I_x * I_y)^2]}{(I_x^2 + I_y^2)}$$

  - 在矩阵R中，同时满足 R（i, j）大于一定阈值 threshold 和 R(i, j) 是某邻域内的局部极大值，则被认为是角点。当角点量小于阈值值，则不是候选角点。

https://zhuanlan.zhihu.com/p/354749490
https://www.zhihu.com/question/484256180

Feature detectors

- Harris 角点检测算法 (Opencv: cv2.cornerHarris())

```
import cv2
import numpy as np

img = cv2.imread('test_1.jpg')
print('imgshape', img.shape)
# imgshape (800, 1200, 3)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# gray = np.float32(gray)
dst = cv2.cornerHarris(gray, 2, 3, 0.04)
print('dst.shape', dst.shape)
# dst.shape (800, 1200)

img[dst>0.01*dst.max()] = [0, 0, 255]
cv2.imshow('dst', img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

(a) (b)

https://zhuanlan.zhihu.com/p/354749490

https://www.zhihu.com/question/484256180

Feature detectors

- Given the large number of feature detectors that have been developed in computer vision, how can we decide which ones to use?
  - ✓ **Measuring repeatability 衡量可重复性**
    - the frequency with which keypoints detected in one image are found within x (say, x = 1.5) pixels of the corresponding location in a transformed image.
  - ✓ **Scale invariance 尺度不变**
    - In many situations, detecting features at the finest stable scale possible may not be appropriate.
  - ✓ **Rotational invariance and orientation estimation 旋转不变和方向估计**
    - most image matching and object recognition algorithms need to deal with (at least) in-plane image rotation
  - ✓ **Affine invariance 仿射不变**
    - Affine invariant detectors not only respond at consistent locations after scale and orientation changes, they also respond consistently across affine deformations such as (local) perspective foreshortening

Feature detectors

- **Common feature detectors**

Common feature detectors and their classification:

Feature detector	Edge	Corner	Blob	Ridge
Canny[3]	Yes	No	No	No
Sobel	Yes	No	No	No
Harris & Stephens / Plessey[4]	Yes	Yes	No	No
SUSAN[5]	Yes	Yes	No	No
Shi & Tomasi[6]	No	Yes	No	No
Level curve curvature[7]	No	Yes	No	No
FAST[8]	No	Yes	Yes	No
Laplacian of Gaussian[7]	No	Yes	Yes	No
Difference of Gaussians[9][10]	No	Yes	Yes	No
Determinant of Hessian[7]	No	Yes	Yes	No
Hessian strength feature measures[11][12]	No	Yes	Yes	No
MSER[13]	No	No	Yes	No
Principal curvature ridges[14][15][16]	No	No	No	Yes
Grey-level blobs[17]	No	No	Yes	No

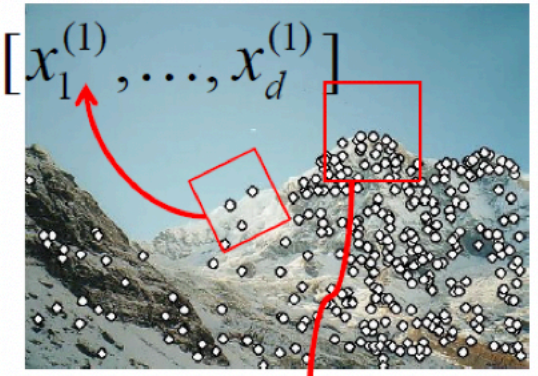
Here are some commonly used image feature detectors:

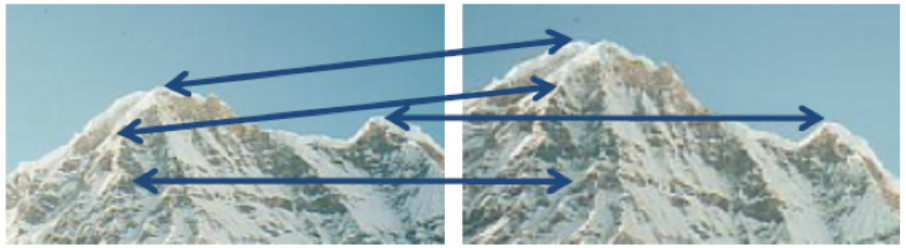
- **Edge Detectors:**
  - Sobel operator
  - Prewitt operator
  - Canny edge detector
- **Corner Detectors:**
  - Harris corner detector
  - Shi-Tomasi corner detector
- **Blob Detectors:**
  - Laplacian of Gaussian (LoG)
  - Difference of Gaussians (DoG)
  - Determinant of Hessian (DoH)
- **Scale-Invariant Feature Transform (SIFT):**
  - SIFT is a feature detection algorithm that identifies key points in an image, which are invariant to scaling, rotation, and translation.
- **Speeded-Up Robust Features (SURF):**
  - SURF is similar to SIFT but is designed to be computationally more efficient.
- **Histogram of Oriented Gradients (HOG):**
  - HOG is often used for object detection. It calculates the distribution of gradient orientations in an image.
- **Local Binary Pattern (LBP):**
  - LBP is a texture descriptor used for texture classification and facial recognition.
- **Convolutional Neural Networks (CNNs):**

Points and patches

- Keypoint detection and matching pipeline:
  - (1) Feature detection (extraction) stage:
    - Search for the locations that are likely to match well in other images **特征检测/提取**
  - (2) Feature description stage:
    - Each region around detected keypoint locations is converted into a more compact and stable (invariant) *descriptor* that can be matched against other descriptors **特征描述**
  - (3) Feature matching stage:
    - Searches for likely matching candidates in other images **特征匹配**

Alternatively for (3) Feature tracking for vide
processing.





Feature descriptors描述子

- **Feature descriptor:** numerical representations of local image regions or keypoints. They encode information about the intensity, color, texture, or other visual characteristics of the region. 特征描述子是图像的简化表示，仅包含图像最重要的信息。

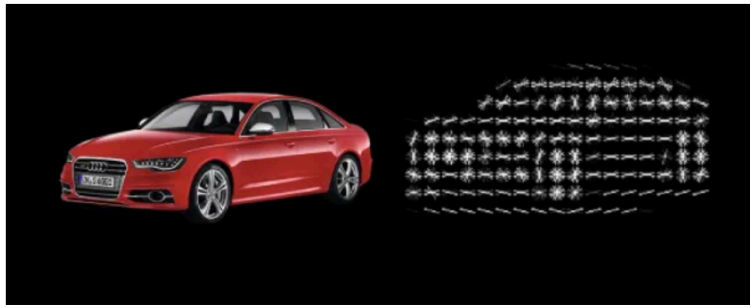
  - **Key/Interest Points:** Feature descriptors are often **associated with key points or interest points in an image**. 关键点/兴趣点
    - These keypoints are locations where the local **visual information is distinctive or unique**, making them suitable for matching and recognition tasks.
  - **Localization**: Feature descriptors are typically designed to be **invariant or robust to certain transformations** like rotation, scaling, and changes in illumination. 关键点不受变形影响
    - This ensures that the descriptors can match corresponding features in different images despite variations.

Feature descriptors描述子

- **Common Descriptors:**
  - **HOG (Histogram of Oriented Gradients) 方向梯度直方图:** Describes the distribution of gradient orientations in localized regions, commonly used in object detection.
  - **SIFT (Scale-Invariant Feature Transform)尺度不变特征变换:** Describes keypoints based on their scale and orientation, providing invariance to scaling and rotation.
  - **SURF (Speeded-Up Robust Features) 加速鲁棒性特征:** Similar to SIFT but designed for faster computation.
  - **ORB (Oriented FAST and Rotated BRIEF):** Combines the speed of FAST keypoint detection with the robustness of BRIEF descriptors.
  - **BRISK (Binary Robust Invariant Scalable Keypoints):** A binary descriptor designed for real-time applications.
  - **MOPS(Bias and gain normalization)**
  - **....**

Feature descriptors

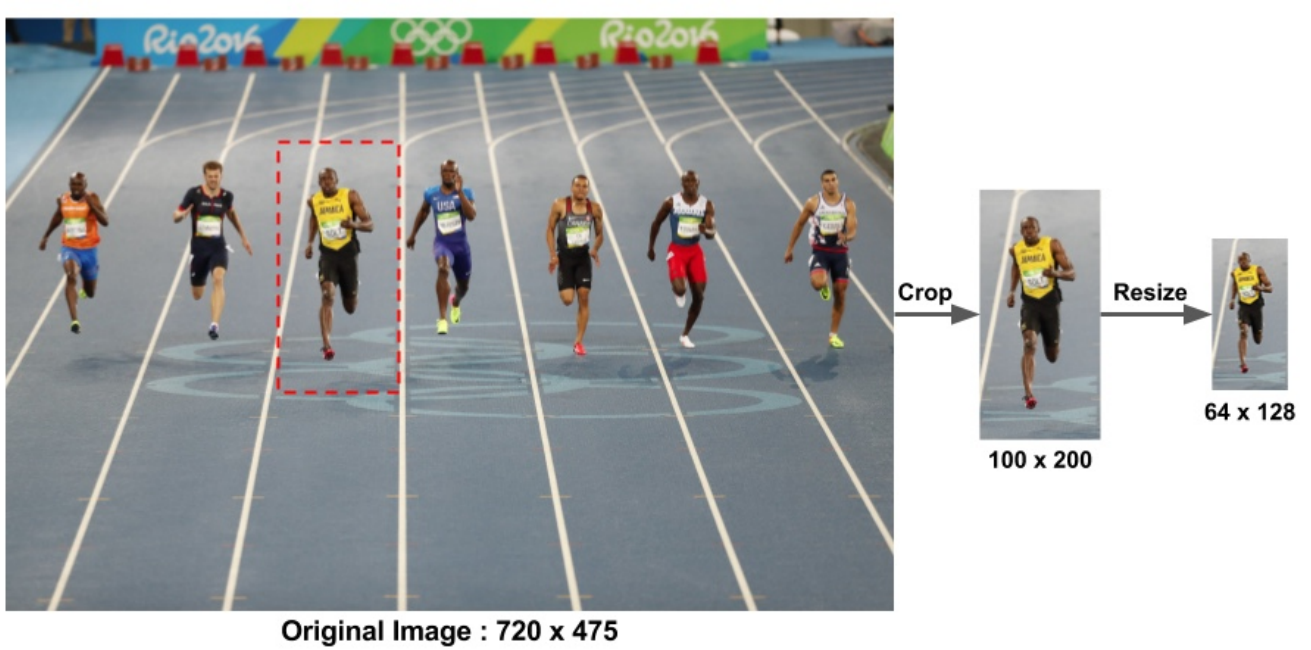
- **HOG (Histogram of Oriented Gradients) 方向梯度直方图**
  - 方向梯度直方图（HOG）特征是一种在计算机视觉和图像处理中用来进行物体检测的特征描述子。HOG特征通过**计算和统计图像局部区域的梯度方向直方图**来构成特征。
  - 方向梯度直方图(HOG)特征描述子常和线性支持向量机(SVM)配合使用，用于训练高精度的目标分类器，在行人检测中获得了极大的成功。
  - **主要思想：**
    - ✓ 在一副图像中，局部目标的表象和形状能够被梯度或边缘的方向密度分布很好地描述。
    - ✓ 其本质为梯度的统计信息，而梯度主要存在于边缘的地方。
  - 在HOG中，对一幅图进行如下划分：
    - ✓ 图像(image)->检测窗口(win)->图像块(block)->细胞单元(cell)



Example of HOG feature descriptor on images. Credit: Analytics Vidhya

Feature descriptors

- **HOG特征提取算法流程图**
  - **Preprocessing**
    - Preprocess the image so that it has a fixed aspect ratio (A common aspect ratio (width:height) is 1:2, so your images can be 100x200, 500x1000, etc)



https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa
https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Feature descriptors

- **HOG特征提取算法流程图**
  - **The gradient of the image is calculated.**
    - The gradient is obtained by combining magnitude and angle from the image. Considering a block of 3x3 pixels, first Gx and Gy is calculated for each pixel. First Gx and Gy is calculated using the formulae below for each pixel value .

$$G_x(r,c) = I(r,c+1) - I(r,c-1) \quad G_y(r,c) = I(r-1,c) - I(r+1,c)$$

https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa
https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Feature descriptors

- **HOG特征提取算法流程图**
  - **The gradient of the image is calculated.**
    - The gradient is obtained by combining magnitude and angle from the image. Considering a block of 3x3 pixels, first Gx and Gy is calculated for each pixel. First Gx and Gy is calculated using the formulae below for each pixel value :

$$G_x(r,c) = I(r,c+1) - I(r,c-1) \quad G_y(r,c) = I(r-1,c) - I(r+1,c)$$

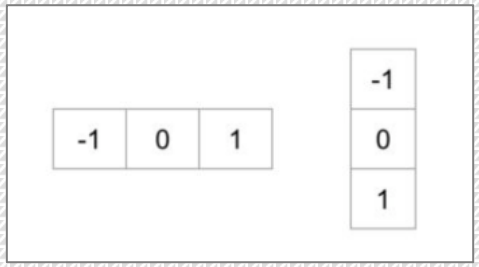
    - After calculating Gx and , magnitude and angle of each pixel is calculated using the formulae mentioned below.

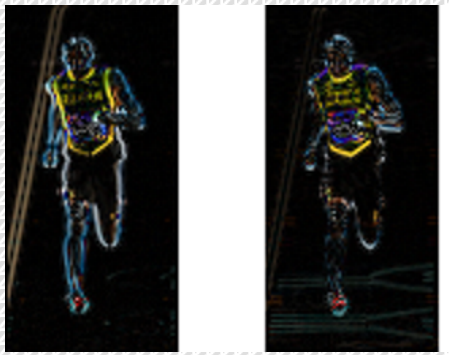
$$Magnitude(\mu) = \sqrt{G_x^2 + G_y^2} \quad Angle(\theta) = |\tan^{-1}(G_y/G_x)|$$

https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa
https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Feature descriptors

- **HOG特征提取算法流程图**
  - **The gradient of the image is calculated.**





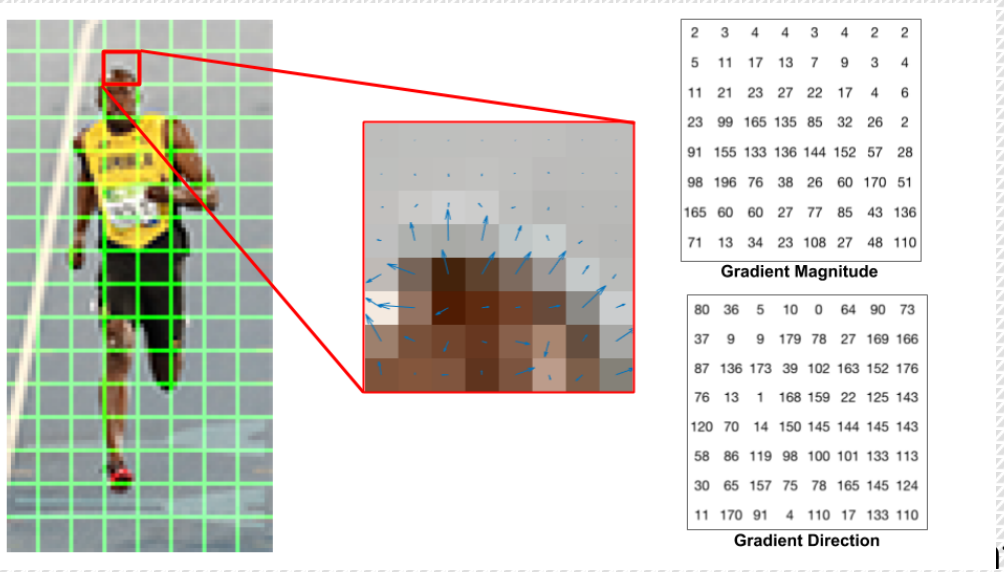
$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa
https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Feature descriptors

- **HOG特征提取算法流程图**
  - After obtaining the gradient of each pixel, the gradient matrices (magnitude and angle matrix) are divided into 8x8 cells to form a block. For each block, a 9-point histogram is calculated. A 9-point histogram develops a histogram with 9 bins and each bin has an angle range of 20 degrees.

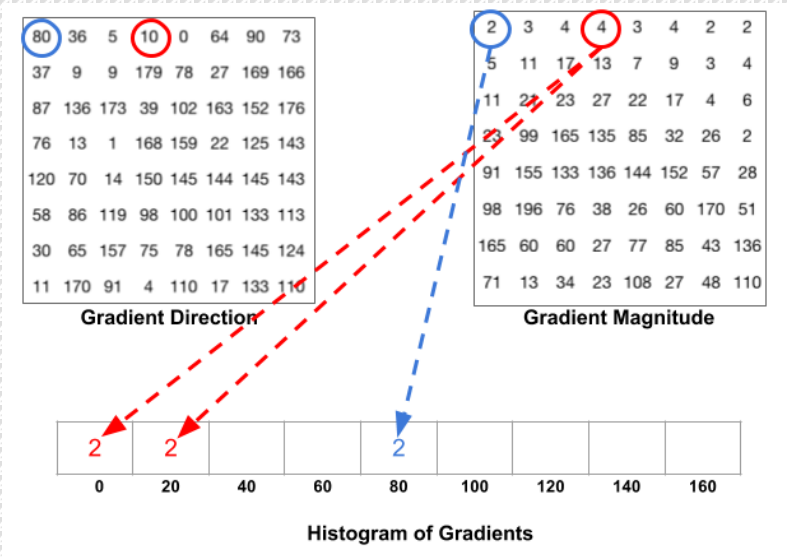


✓中间图中的箭头表示梯度，箭头方向表示梯度方向，箭头长度表示梯度大小。
✓右图是 8×8 的cell中表示梯度的原始数字，注意角度的范围介于0到180度之间，而不是0到360度，这被称为“无符号”梯度，因为两个完全相反的方向被认为是相同的。

https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa
https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Feature descriptors

- **HOG特征提取算法流程图**
  - After obtaining the gradient of each pixel, the gradient matrices (magnitude and angle matrix) are divided into 8x8 cells to form a block. For each block, a 9-point histogram is calculated. A 9-point histogram develops a histogram with 9 bins and each bin has an angle range of 20 degrees.



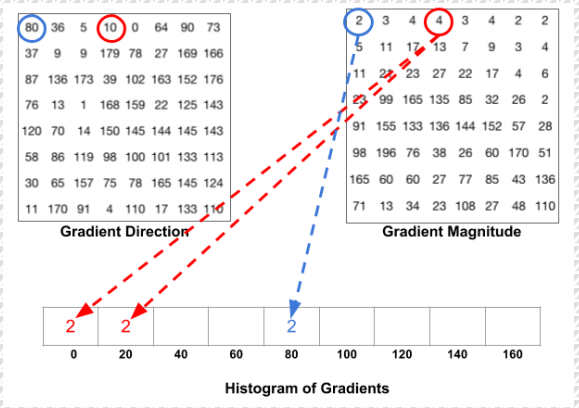
✓ 图中蓝圈包围的像素，角度为80度，这个像素对应的幅值为2，所以在直方图80度对应的bin加上2。红圈包围的像素，角度为10度，介于0度和20度之间，其幅值为4，那么这个梯度值就被按比例分给0度和20度对应的bin，也就是各加上2。

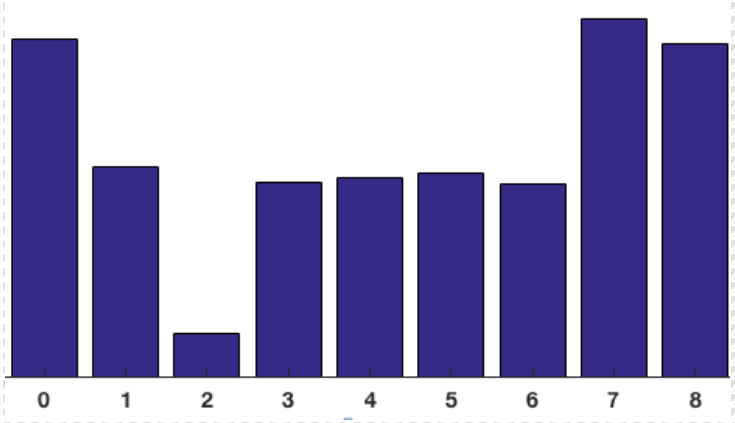
✓ 还有一个细节需要注意，如果某个像素的梯度角度大于160度，也就是在160度到180度之间，那么把这个像素对应的梯度值按比例分给0度和160度对应的bin。

https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa
https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Feature descriptors

- **HOG特征提取算法流程图**
  - After obtaining the gradient of each pixel, the gradient matrices (magnitude and angle matrix) are divided into 8x8 cells to form a block. For each block, a 9-point histogram is calculated. A 9-point histogram develops a histogram with 9 bins and each bin has an angle range of 20 degrees.

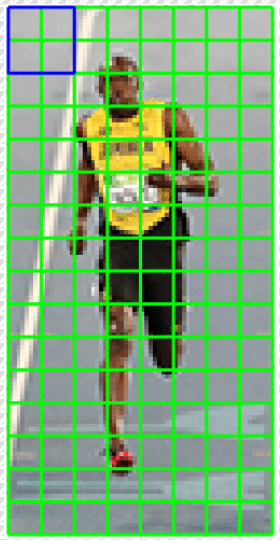


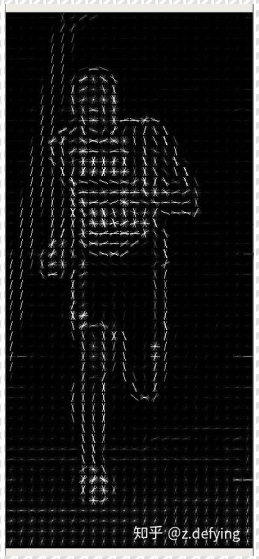


https://medium.com/analytics-vidhya/a-gentle-introduction-into-the-histogram-of-oriented-gradients-fdee9ed8f2aa
https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f

Feature descriptors

- **HOG特征提取算法流程图**
  - 每个block包含4个cell, 就得到一个长度为36的特征向量
  - 将整幅图像划分成cell的个数为8x16，就是横向有8个cell，纵向有16个cell。每个block有2x2个cell的话，那么block的个数为：(16-1)x(8-1)=105。即有7个水平block和15个竖直block。
  - 再将这105个block合并，就得到了整个图像的特征描述符，长度为 105×36=3780






```
from skimage import feature, exposure
import cv2
image = cv2.imread('/home/zxd/Pictures/Selection_018.jpg')
fd, hog_image = feature.hog(image, orientations=9, pixels_per_cell=(16, 16),
                    cells_per_block=(2, 2), visualize=True)

# Rescale histogram for better display
hog_image_rescaled = exposure.rescale_intensity(hog_image, in_range=(0, 10))

cv2.imshow('img', image)
cv2.imshow('hog', hog_image_rescaled)
cv2.waitKey(0)==ord('q')
```

Feature descriptors

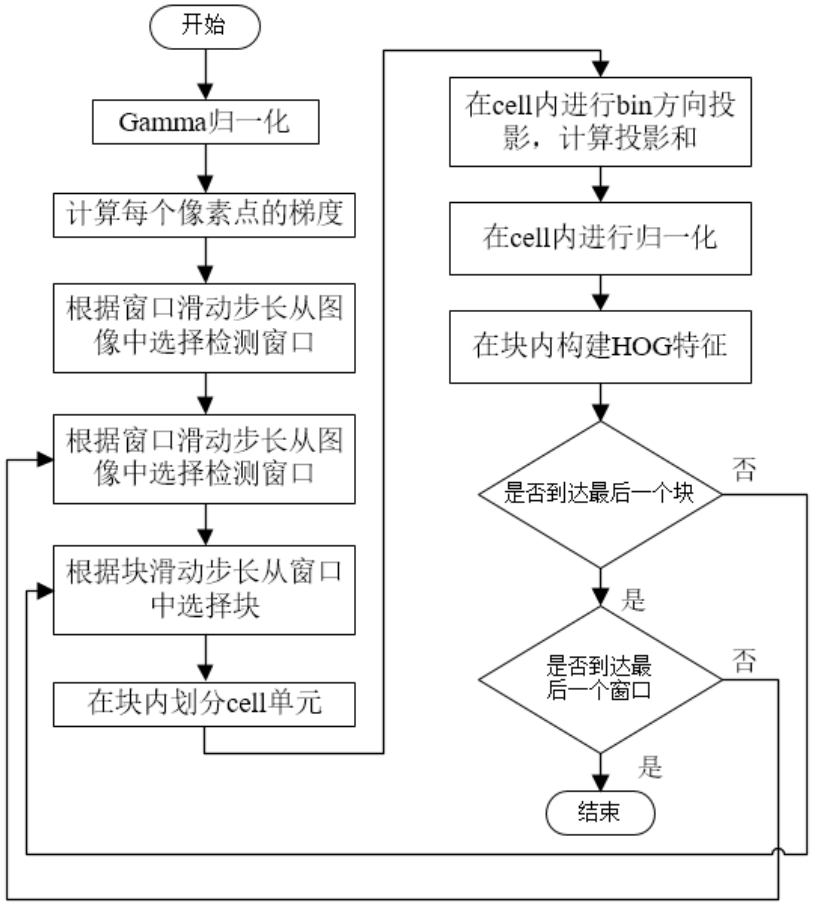
• HOG特征提取算法流程图

- **需要注意的地方：**

1) 色彩和**Gamma归一化**为了减少光照因素的影响，首先需要将整个图像进行规范化（归一化）。在图像的纹理强度中，局部的表层曝光贡献的比重较大，所以，这种压缩处理能够有效地**降低图像局部的阴影和光照变化。**

2) 图像的**梯度针对的是每一个像素**计算得到，然后再**cell中进行方向梯度直方图**的构建，在**block中进行对比度归一化**操作。

3) 由于窗口的滑动性与块的滑动行，窗口与块都会出现不同程度的重叠（由步长决定），此时在块内划分出的cell就会多次出现，这就意味着：每一个细胞单元的输出都多次作用于最终的描述器。



Feature descriptors

- **Scale invariant feature transform (SIFT).**
  - SIFT features are formed by computing the gradient at each pixel in a 16×16 window around the detected keypoint, using the appropriate level of the Gaussian pyramid at which the keypoint was detected .

Feature descriptors

- Major advantages of SIFT:

  - **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation)
  - **Distinctiveness:** individual features can be matched to a large database of objects
  - **Quantity:** many features can be generated for even small objects
  - **Efficiency:** close to real-time performance
  - **Extensibility:** can easily be extended to a wide range of different feature types, with each adding robustness

Feature descriptors

• SIFT（Scale-invariant feature transform）尺度不变特征变换

- SIFT是用于描述图像中的局部特征，在空间尺度中寻找**极值点**，并且提取出其位置、尺度、旋转不变量，因此**具有尺度和旋转不变的性质。**
- **特点：**
  - ✓ 图像的局部特征，对旋转、尺度缩放、亮度变化保持不变，对视角变化、仿射变换、噪声也保持一定程度的稳定性。
  - ✓ 独特性好，信息量丰富，适用于海量特征库进行快速、准确的匹配。
  - ✓ 多量性，即使是很少几个物体也可以产生大量的SIFT特征。
  - ✓ 高速性，经优化的SIFT匹配算法甚至可以达到实时性
  - ✓ 扩展性，可以很方便的与其他的特征向量进行联合。

Feature descriptors

- **SIFT（尺度不变特征变换）- major steps**
  - **Scale-space peak selection:** Potential location for finding features.
    - 生成高斯差分金字塔（DOG金字塔），尺度空间构建, 尺度空间的极值检测.
  - **Keypoint Localization:** Accurately locating the feature keypoints.
    - 特征点定位: 在每个候选的位置上，通过一个拟合精细模型来确定位置尺度，关键点的选取依据他们的稳定程度
  - **Orientation Assignment:** Assigning orientation to keypoints.
    - 特征方向赋值： 基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向，后续的所有操作都是对于关键点的方向、尺度和位置进行变换，从而提供这些特征的不变性。
  - **Keypoint descriptor:** Describing the keypoints as a high dimensional vector.
    - 特征点描述：在每个特征点周围的邻域内，在选定的尺度上测量图像的局部梯度，这些梯度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变换。
  - **Keypoint Matching:**
    - 特征点匹配

https://www.cnblogs.com/wangguchangqing/p/4853263.htm
https://blog.csdn.net/dcrmg/article/details/52577555
https://medium.com/@deepanshut041/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40

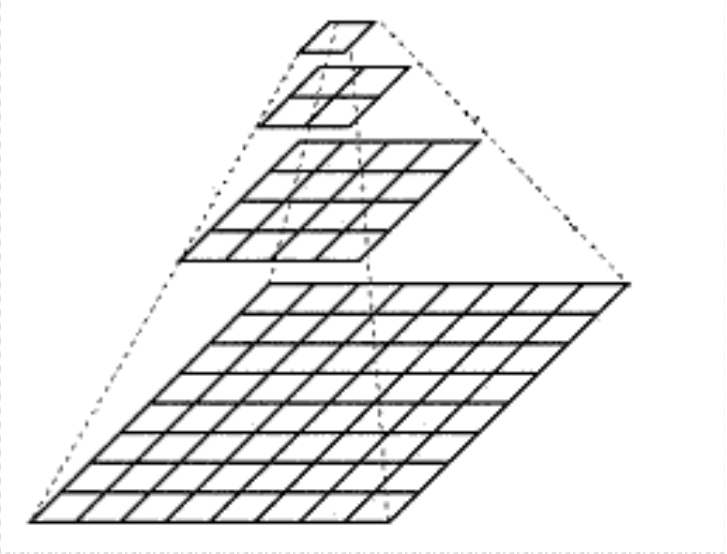
Feature descriptors

- **SIFT（尺度不变特征变换）- major steps**
  - **Scale-space peak selection:** Potential location for finding features.
    - 生成高斯差分金字塔（DOG金字塔），尺度空间构建, 尺度空间的极值检测.
  - **Keypoint Localization:** Accurately locating the feature keypoints.
    - 特征点定位: 在每个候选的位置上，通过一个拟合精细模型来确定位置尺度，关键点的选取依据他们的稳定程度
  - **Orientation Assignment:** Assigning orientation to keypoints.
    - 特征方向赋值： 基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向，后续的所有操作都是对于关键点的方向、尺度和位置进行变换，从而提供这些特征的不变性。
  - **Keypoint descriptor:** Describing the keypoints as a high dimensional vector.
    - 特征点描述：在每个特征点周围的邻域内，在选定的尺度上测量图像的局部梯度，这些梯度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变换。
  - **Keypoint Matching:**
    - 特征点匹配

https://www.cnblogs.com/wangguchangqing/p/4853263.htm
https://blog.csdn.net/dcrmg/article/details/52577555
https://medium.com/@deepanshut041/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40

Feature descriptors

- **Scale-space peak Selection(尺度空间极值检测)**
  - ✓ 以多分辨率来解释图像的结构，通过对原始图像进行多尺度像素采样的方式，生成N个不同分辨率的图像。
  - ✓ 把具有最高级别分辨率的图像放在底部，以金字塔形状排列，往上是一系列像素（尺寸）逐渐降低的图像，一直到金字塔的顶部只包含一个像素点的图像，这就构成了传统意义上的图像金字塔。

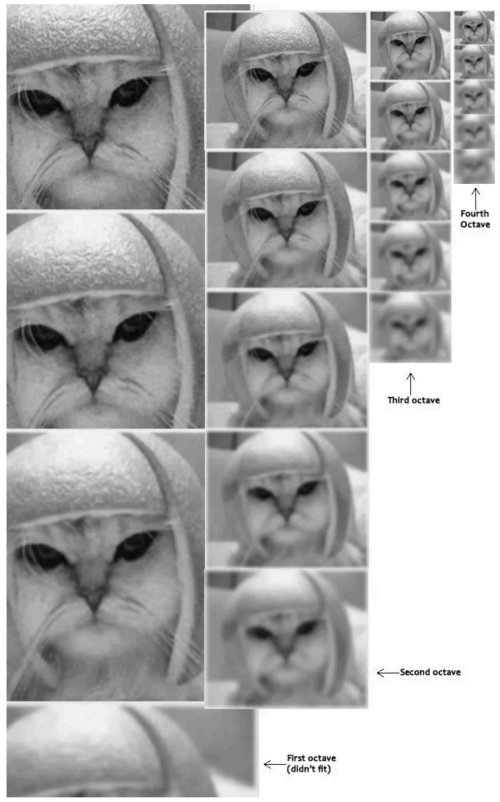


- 获得图像金字塔一般包括二个步骤：
  1) 利用低通滤波器平滑图像
  2) 对平滑图像进行抽样（采样）

  有两种采样方式——上采样（分辨率逐级升高）和下采样（分辨率逐级降低）

Feature descriptors

- **Scale-space peak Selection:**
  - The scale space of an image is a function L(x,y,σ) that is produced from the convolution of a **Gaussian kernel(Blurring) at different scales** with the input image.



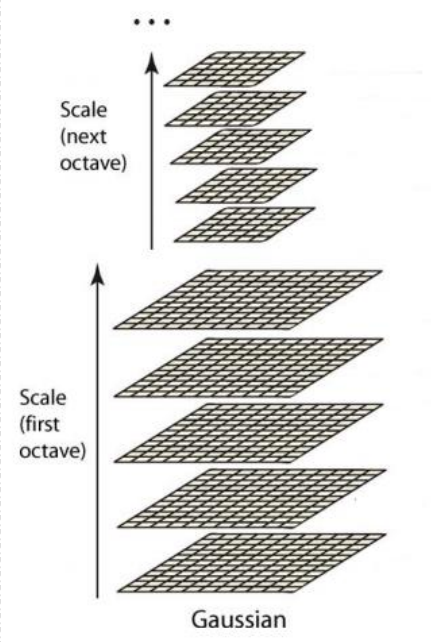
获得图像金字塔一般包括二个步骤：

1) 利用低通滤波器平滑图像
2) 对平滑图像进行抽样（采样）

有两种采样方式——上采样（分辨率逐级升高）和下采样（分辨率逐级降低）

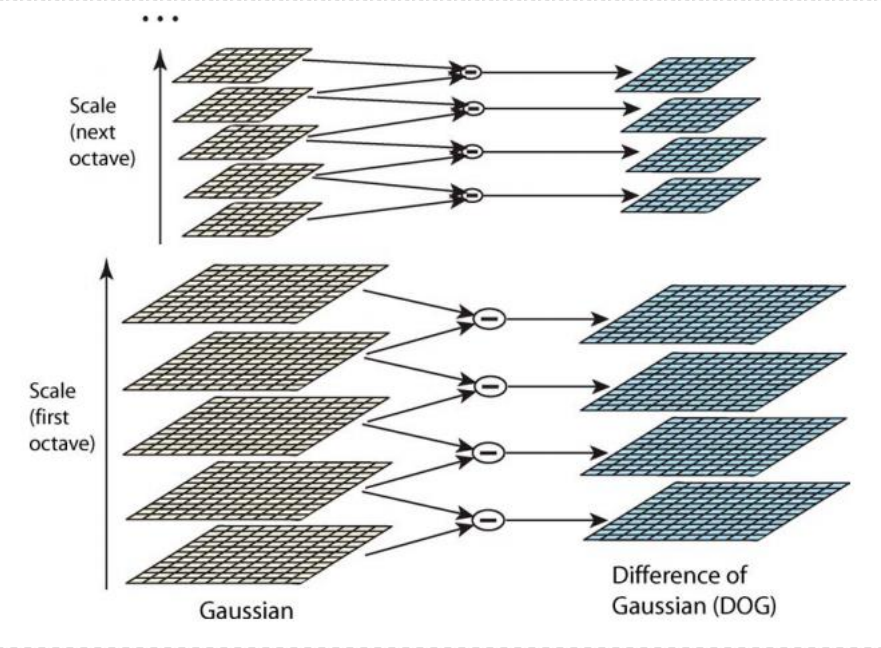
Feature descriptors

- **构建图像高斯金字塔：**
  - ✓ 高斯金字塔式在Sift算子中提出来的概念，首先高斯金字塔并不是一个金字塔，而是有很多组（Octave）金字塔构成，并且每组金字塔都包含若干层（Interval）.
  - ✓ 同一组内，不同层图像的**尺寸一样**，后一层图像的高斯平滑因子σ是前一层图像平滑因子的k倍；
  - ✓ 不同组内，后一组第一个图像是前一组倒数第三个图像的**二分之一采样**，图像大小是前一组的一半。



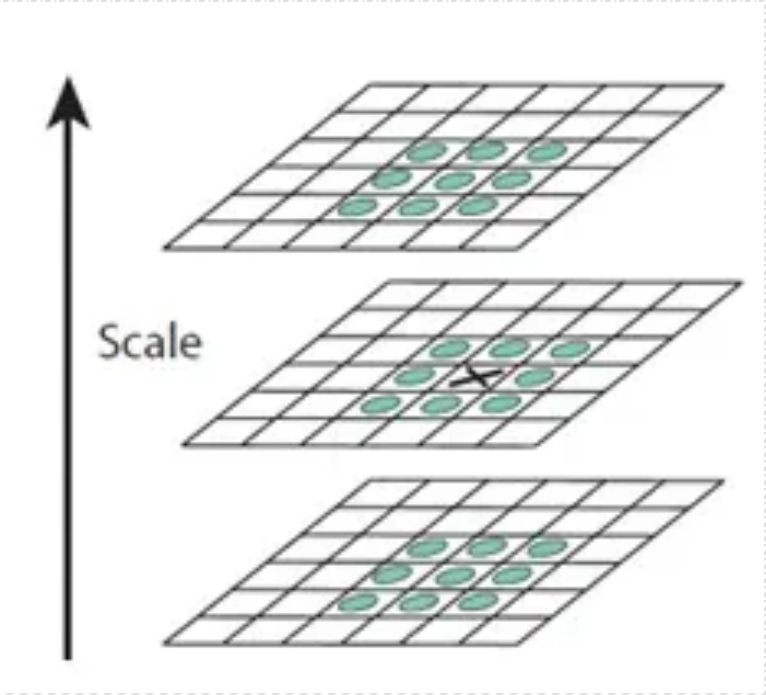
Feature descriptors

- **DOG(Difference of Gaussian kernel)差分金字塔：**
  - The difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different σ, let it be σ and *kσ*. This process is done for different octaves of the image in the Gaussian Pyramid.



Feature descriptors

- **Finding keypoints:**
  - One pixel in an image is compared with its 8 neighbors as well as 9 pixels in the next scale and 9 pixels in previous scales. This way, a total of 26 checks are made. If it is a local extrema, it is a potential keypoint. It basically means that keypoint is best represented in that scale.

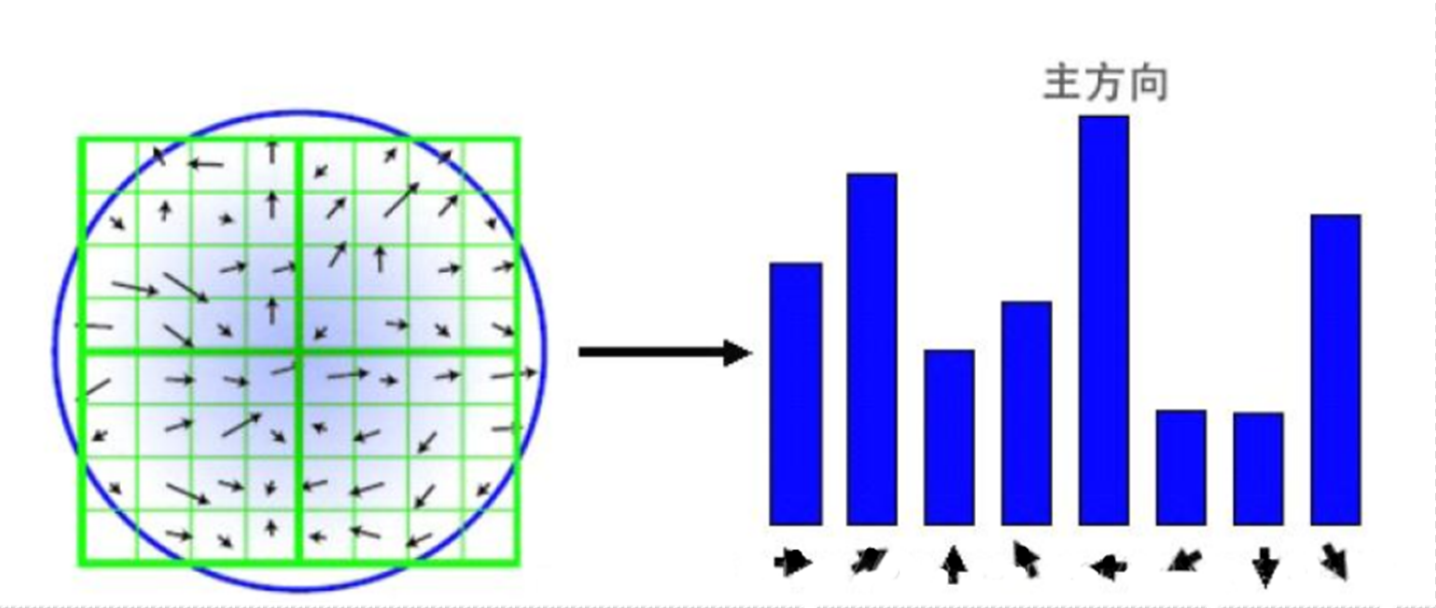


Feature descriptors

- **Keypoint Localization:**
  - ✓ 在每个候选的位置上，通过一个拟合精细模型来确定位置尺度，关键点的选取依据他们的稳定程度。
  - ✓ DOG值对噪声和边缘比较敏感，所以在第2步的尺度空间中检测到的局部极值点还要经过进一步的筛选，去除不稳定和错误检测出的极值点，另一点就是在构建高斯金字塔过程中采用了下采样的图像，在下采样图像中提取的极值点对应在原始图像中的确切位置，也是要在本步骤中解决的问题。

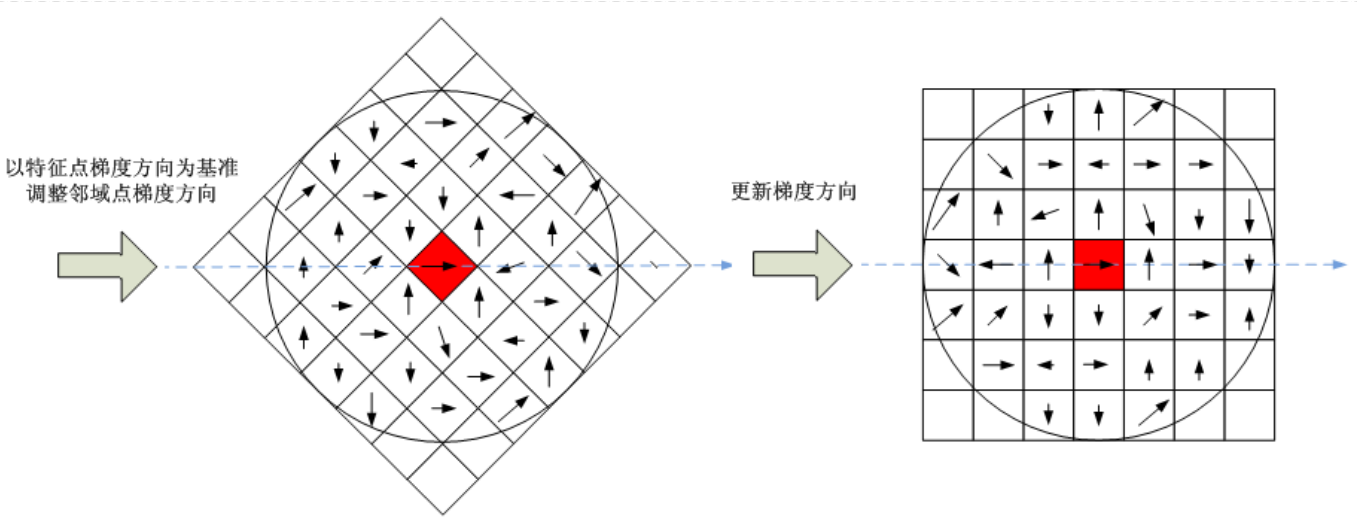
Feature descriptors

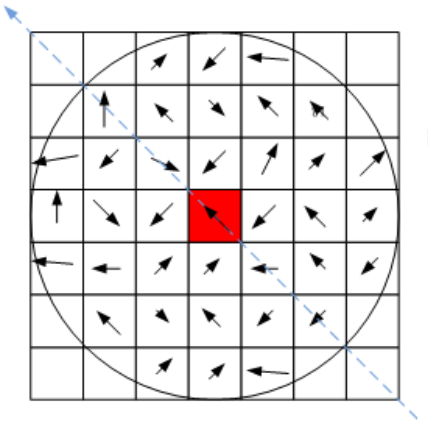
- **Orientation Assignment(确定特征点方向):**
  - 确定特征点方向：利用直方图统计领域内像素对应的梯度和幅值
    - 在以关键点为中心的邻域窗口内采样，并用直方图统计邻域像素的梯度方向。梯度直方图的范围是0～360度，其中每45度一个柱，总共8个柱, 或者每10度一个柱，总共36个柱
    - 梯度方向角为横轴刻度，取45度为一个单位，那么横轴就有8个刻度；纵轴是对应梯度的幅值累加值。



Feature descriptors

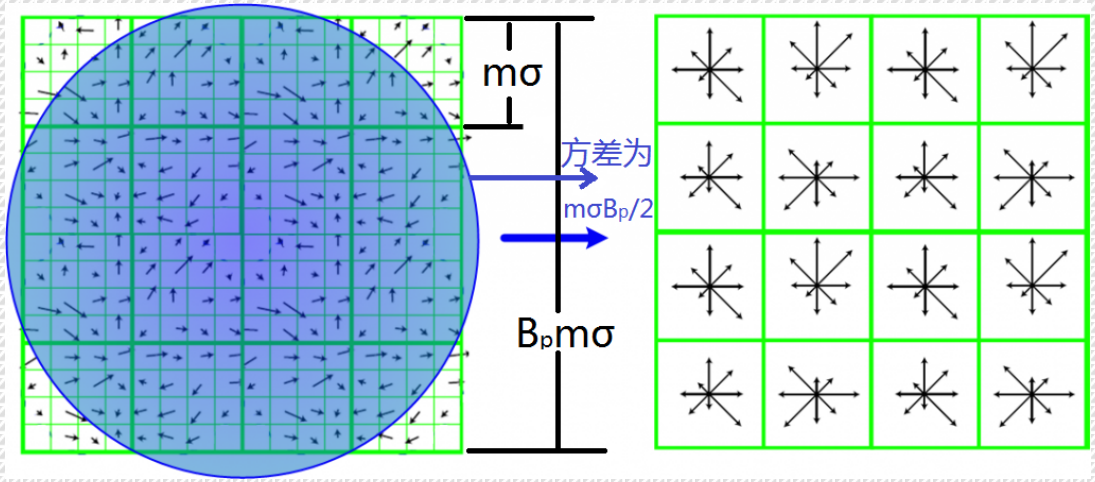
- **Keypoint descriptor生成描述子:**
  - ✓ 旋转图像至主方向
    - ○ 为了保证特征矢量具有旋转不变性，需要以特征点为中心，将特征点附近邻域内图像梯度的位置和方向旋转一个方向角θ，即将原图像x轴转到与主方向相同的方向。





Feature descriptors

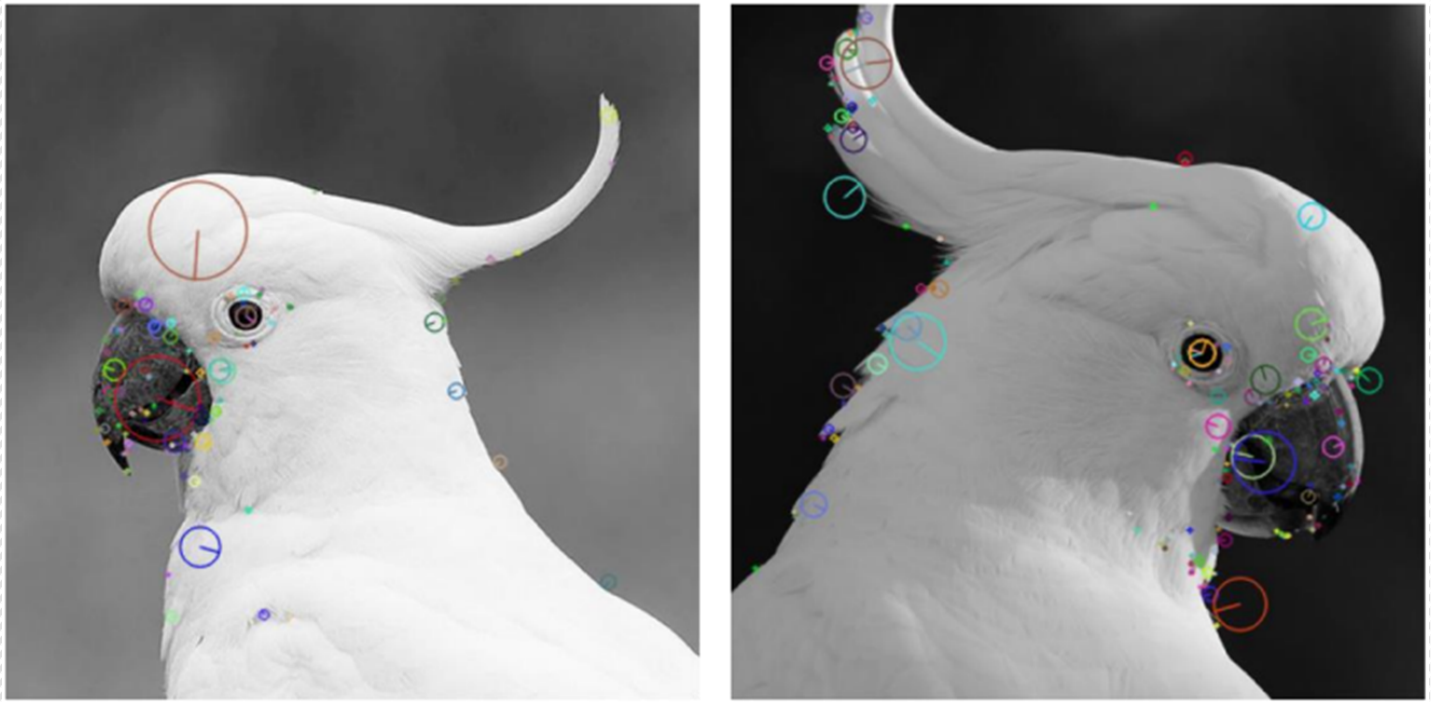
- **Keypoint descriptor生成描述子:**
  - ✓ 生成特征向量
    - 在每子区域内计算8个方向的梯度方向直方图，绘制每个梯度方向的累加值，形成一个**种子点**，与求特征点主方向时有所不同，此时，每个子区域的梯度方向直方图将0° ~360°划分为8个方向范围，每个范围为45°，这样，**每个种子点共有8个方向的梯度强度信息。**由于存在4X4个子区域，所以，共有4X4X8=128个数据，最终形成**128维的SIFT特征矢量。**



每一维都可以表示4*4个格子中一个的scale/orientation. 将这个向量归一化之后，就进一步去除了光照的影响

Feature descriptors

- SIFT可视化



Detector *vs* Descriptor

- 关键点检测器是一种根据函数的局部最大值从图像中选择点的算法。
- 关键点描述子/符是用于描述关键点周围的图像补丁值的向量。描述方法有比较原始像素值的方法也有更复杂的方法，如梯度方向的直方图。
- 关键点检测器一般是从一个帧图片中寻找到特征点。而描述符帮助我们在“关键点匹配”步骤中将不同图像中的相似关键点彼此分配。
- 如图所示，一个帧中的一组关键点被分配给另一帧中的关键点，以使它们各自描述符的相似性最大化，并且这些关键点代表图像中的同一对象。除了最大化相似性之外，好的描述符还应该能够最大程度地减少不匹配的次数，即避免将彼此不对应于同一对象的关键点分配给彼此。

https://cloud.tencent.com/developer/article/1689075

Feature descriptors

- **Others: Bias and gain normalization (MOPS)**
  - sampled at a spacing of five pixels relative to the detection scale, using a coarser level of the image pyramid to avoid aliasing. To compensate for affine photometric variations (linear exposure changes or bias and gain, (3.3)), patch intensities are re-scaled so that their mean is zero and their variance is one.
-

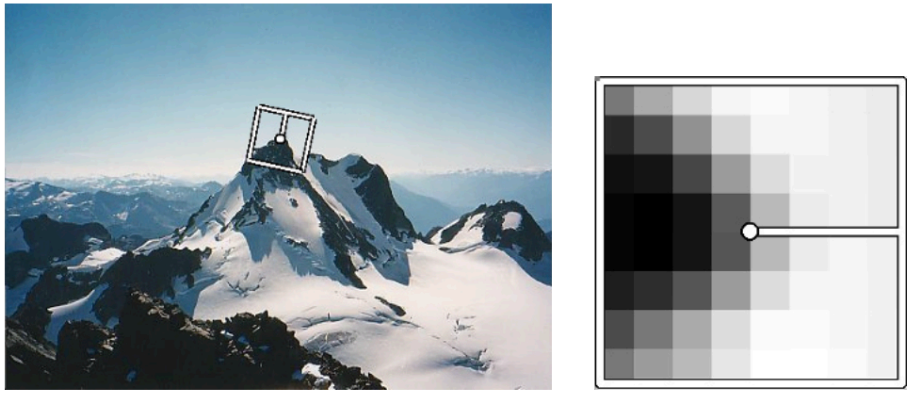
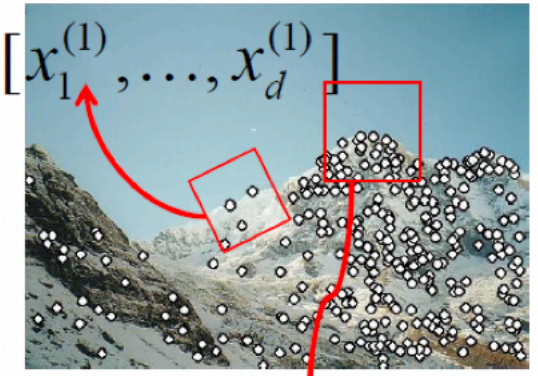


Figure 4.17 MOPS descriptors are formed using an 8×8 sampling of bias and gain normalized intensity values, with a sample spacing of five pixels relative to the detection scale (Brown, Szeliski, and Winder 2005) © 2005 IEEE. This low frequency sampling gives the features some robustness to interest point location error and is achieved by sampling at a higher pyramid level than the detection scale.

Points and patches

- Keypoint detection and matching pipeline:
  - (1) Feature detection (extraction) stage:
    - Search for the locations that are likely to match well in other images **特征检测/提取**
  - (2) Feature description stage:
    - Each region around detected keypoint locations is converted into a more com- pact and stable (invariant) *descriptor* that can be matched against other descriptors **特征描述**
  - (3) Feature matching stage:
    - Searches for likely matching candidates in other images **特征匹配**

Alternatively for (3) Feature tracking for vide processing.



Feature matching

- Feature Matching: finding corresponding points or regions between two or more images. The goal is to establish associations between distinctive features in different images. 建立不同图像之间显著特征的关联
- Given a Euclidean distance metric, the simplest matching strategy is to set a threshold (maximum distance) and to return all matches from other images within this threshold.
- 特征点之间的匹配过程到该步骤其实就是特征向量之间的距离计算，如欧式距离、汉明距离、余弦距离等等。
- 每个图片1的向量都要和 图片2的向量去进行距离的计算，进而得出两个图片特征点之间的距离关系。

计算特征描述算子之间的欧式距离

如果距离小于阈值 则匹配成功

$des_1 = (x_1, \ldots, x_{128})$

$des_2 = (y_1, \ldots, y_{128})$

$$d(des_1, des_2) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_{128} - y_{128})^2}$$

Feature matching

- Setting the threshold too high results in too many false positives, i.e., incorrect matches being returned. Setting the threshold too low results in too many false negatives, i.e., too many correct matches being missed
- **Error rates**

- true positive rate (TPR),

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} = \frac{\mathrm{TP}}{\mathrm{P}};$$

- false positive rate (FPR),

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} = \frac{\mathrm{FP}}{\mathrm{N}};$$

- positive predictive value (PPV),

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} = \frac{\mathrm{TP}}{\mathrm{P'}};$$

- accuracy (ACC),

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{P+N}}.$$

	True matches	True non-matches		
Predicted matches	TP = 18	FP = 4	P' = 22	PPV = 0.82
Predicted non-matches	FN = 2	TN = 76	N' = 78	
	P = 20	N = 80	Total = 100	
	TPR = 0.90	FPR = 0.05		ACC = 0.94

Feature matching

- Setting the threshold too high results in too many false positives, i.e., incorrect matches being returned. Setting the threshold too low results in too many false negatives, i.e., too many correct matches being missed
- **Error rates**

- true positive rate (TPR),

$$TPR = \frac{TP}{TP+FN} = \frac{TP}{P};$$

→ *Recall*

- false positive rate (FPR),

$$FPR = \frac{FP}{FP+TN} = \frac{FP}{N};$$

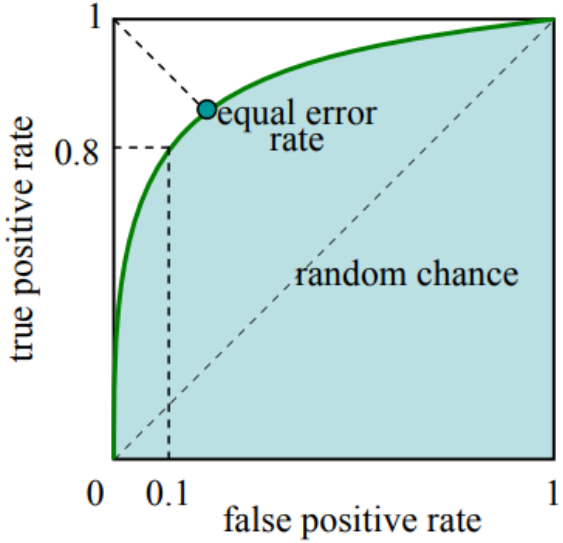
→ *Precision*

- positive predictive value (PPV),

$$PPV = \frac{TP}{TP+FP} = \frac{TP}{P'};$$

- accuracy (ACC),

$$ACC = \frac{TP+TN}{P+N}.$$



Feature matching

- **匹配策略:**
  - **Brute-Force Matching 暴力匹配**: The simplest method involves comparing every feature in one image with every feature in another image. However, this approach can be computationally expensive.
  - **FLANN (Fast Library for Approximate Nearest Neighbors) Matching**: Utilizes efficient data structures to speed up the matching process.
  - **RANSAC (Random Sample Consensus) 随机抽样一致性:** Used for robustly estimating a transformation model (e.g., affine or homography) from noisy data, filtering out outliers in the matching process.
- **Applications**: Image Stitching, Object Recognition, Augmented Reality, Image Registration

Outline

- Points and patches
  - Feature detectors
  - Feature descriptors
  - Feature matching
- Lines
  - Hough transforms
  - RANSAC

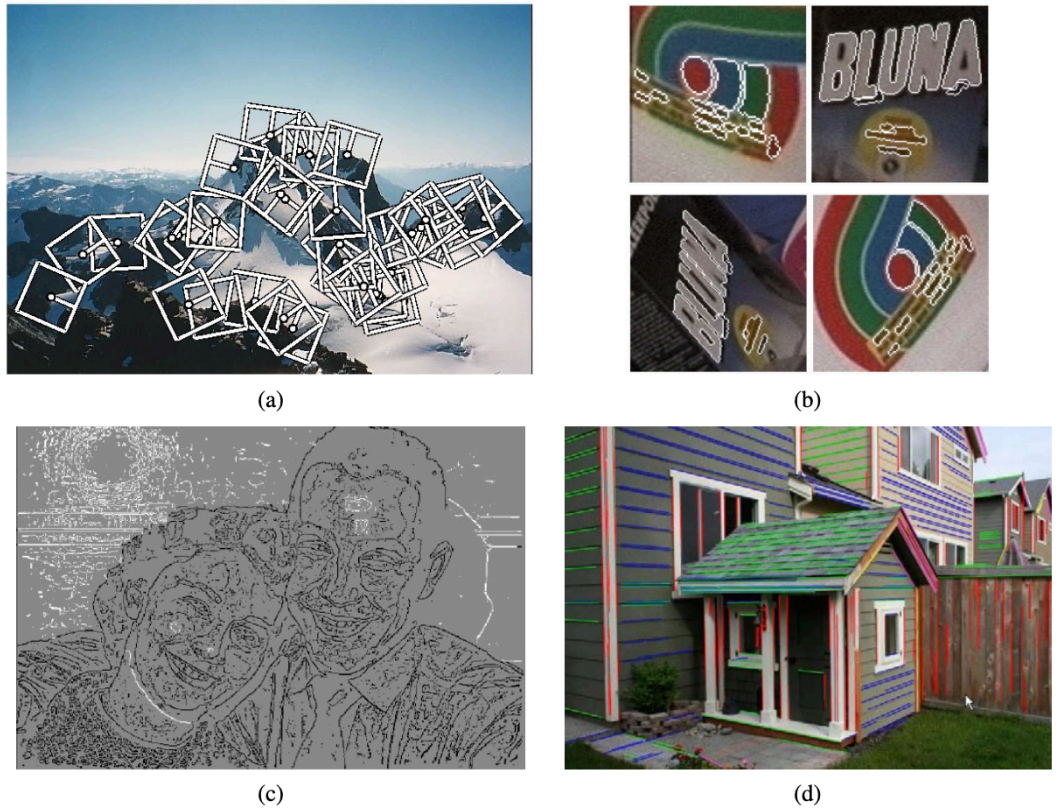


Figure 4.1 A variety of feature detectors and descriptors can be used to analyze, describe and match images: (a) point-like interest operators (Brown, Szeliski, and Winder 2005) © 2005 IEEE; (b) region-like interest operators (Matas, Chum, Urban *et al.* 2004) © 2004 Elsevier; (c) edges (Elder and Goldberg 2001) © 2001 IEEE; (d) straight lines (Sinha, Steedly, Szeliski *et al.* 2008) © 2008 ACM.

An edge is not a line…

How can we detect ***lines*** ?

Lines

- Option 1:
  - Search for the line at every possible position/orientation
  - What is the cost of this operation?
- Option 2:
  - Use a voting scheme: Hough transform

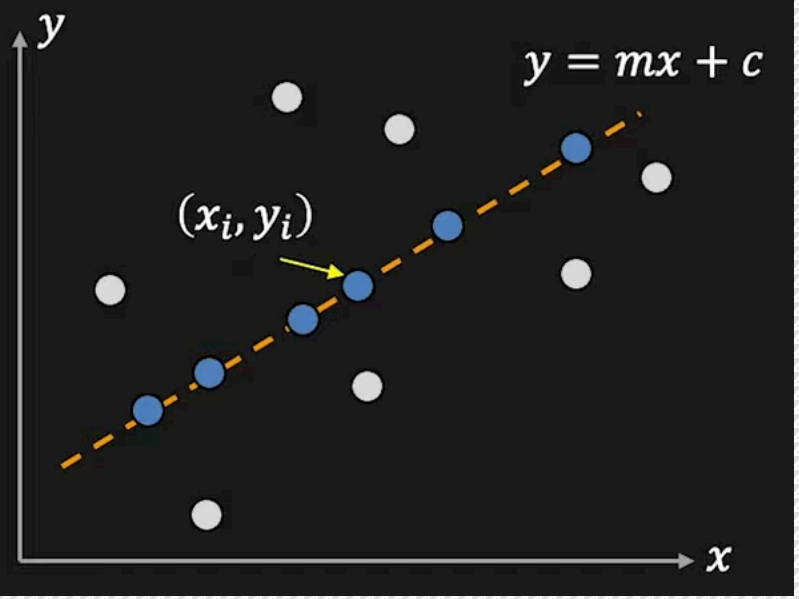
- **线检测——Hough变换**
  - 通过一种投票算法检测具有特定形状的物体
  - Hough变化是一个重要的检测间断点边界形状的方法，它通过将图像坐标空间变化到参数空间来实现直线和曲线的拟合。
  - 从图像中识别几何形状的基本方法之一
  - 直线检测任务中
    - ✓ 图像空间中的直线与参数空间中的点一一对应
    - ✓ 参数空间中的直线与图像空间中的点一一对应
    - ✓ 把在图像空间中的直线检测问题转换到参数空间中对点的检测问题，在参数空间中寻找峰值

Lines

Given: Edge Points (x_i, y_i)

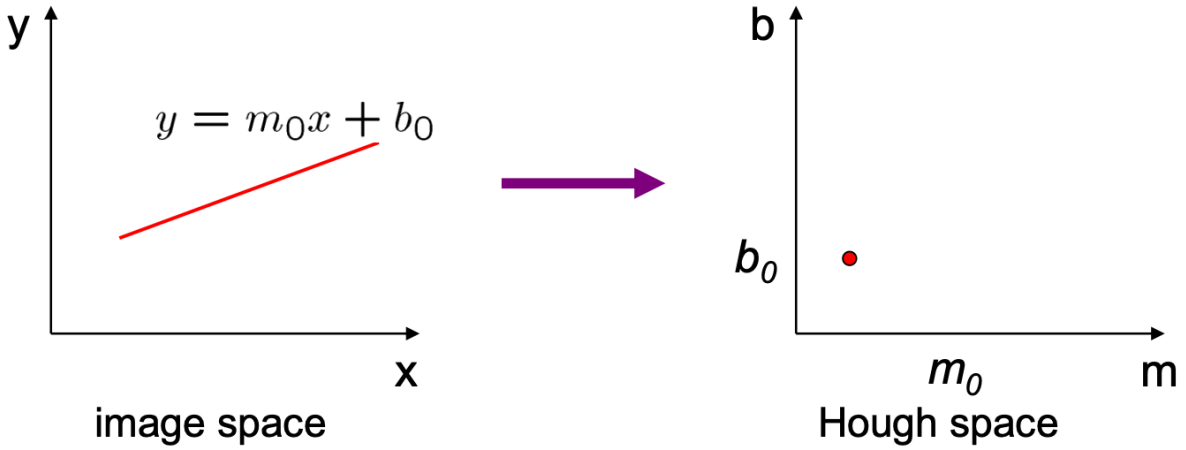
Task: Detect line

$$y = mx + c$$



Lines

Hough transform:

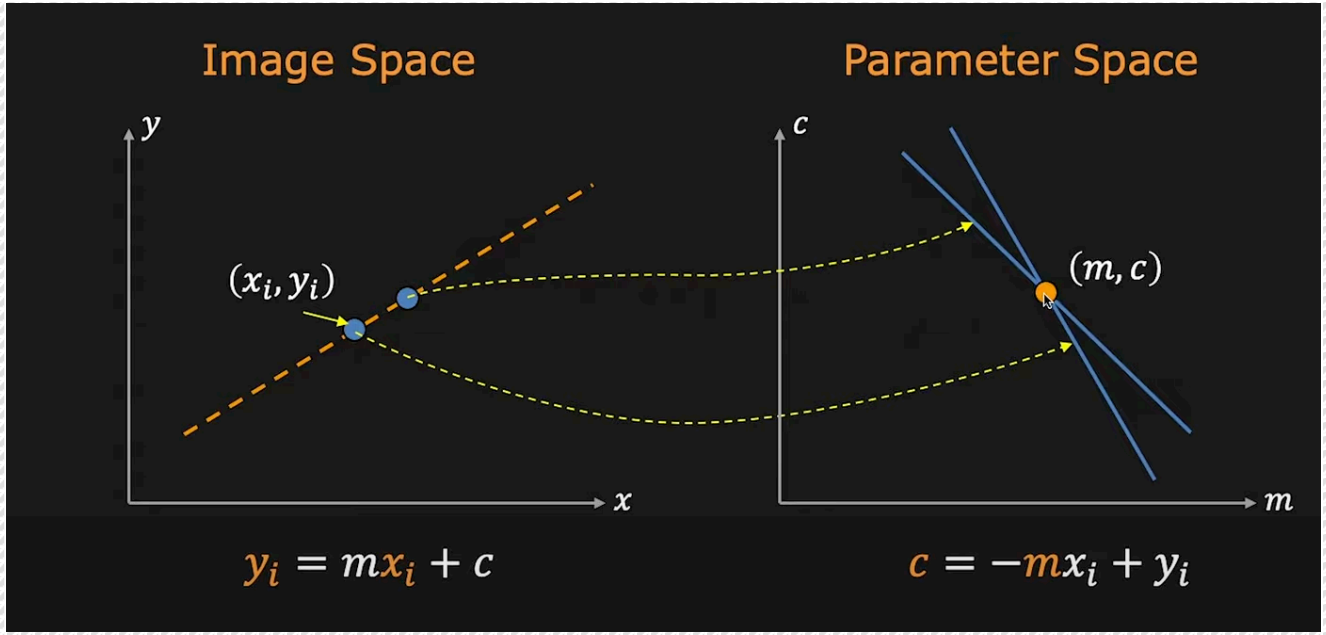


- Connection between image (x,y) and Hough (m,b) spaces
  - A line in the image corresponds to a point in Hough space
  - To go from image space to Hough space:
    - given a set of points (x,y), find all (m,b) such that y = mx + b

图像空间x-y中的点A和点B在参数空间k-b中对应的直线相交于一点，这也就是说AB所确定的直线，在参数空间中对应着唯一一个点，而这个点的坐标值(k 0 , b 0 ）也就是直线AB的参数。

Lines

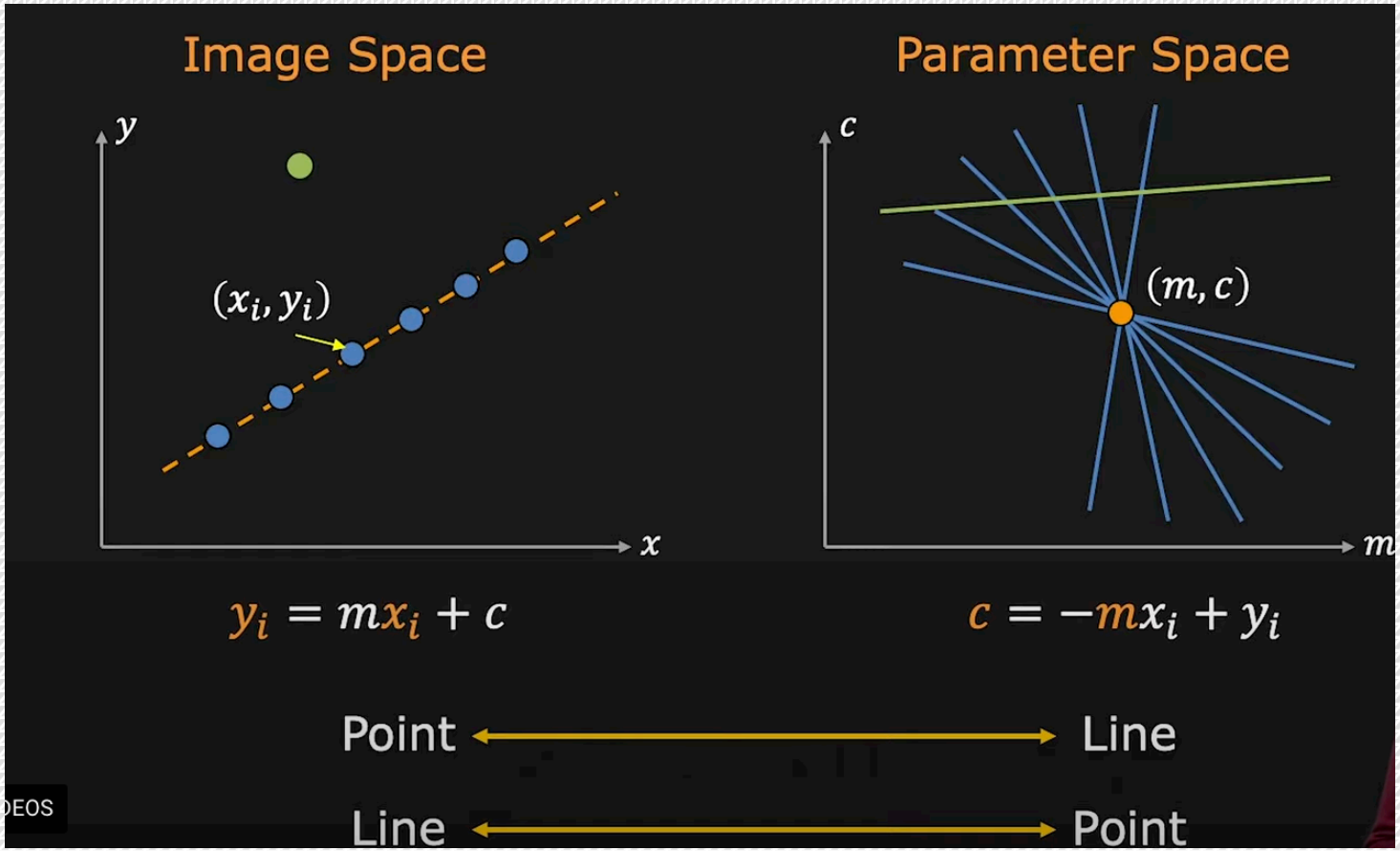
- Hough transform:
  - 参数空间转换
    - ✓ 图像空间中的直线与参数空间中的点一一对应



图像空间x-y中的点A和点B在参数空间k-b中对应的直线相交于一点，这也就是说AB所确定的直线，在参数空间中对应着唯一一个点，而这个点的坐标值(k 0，b 0）也就是直线AB的参数。

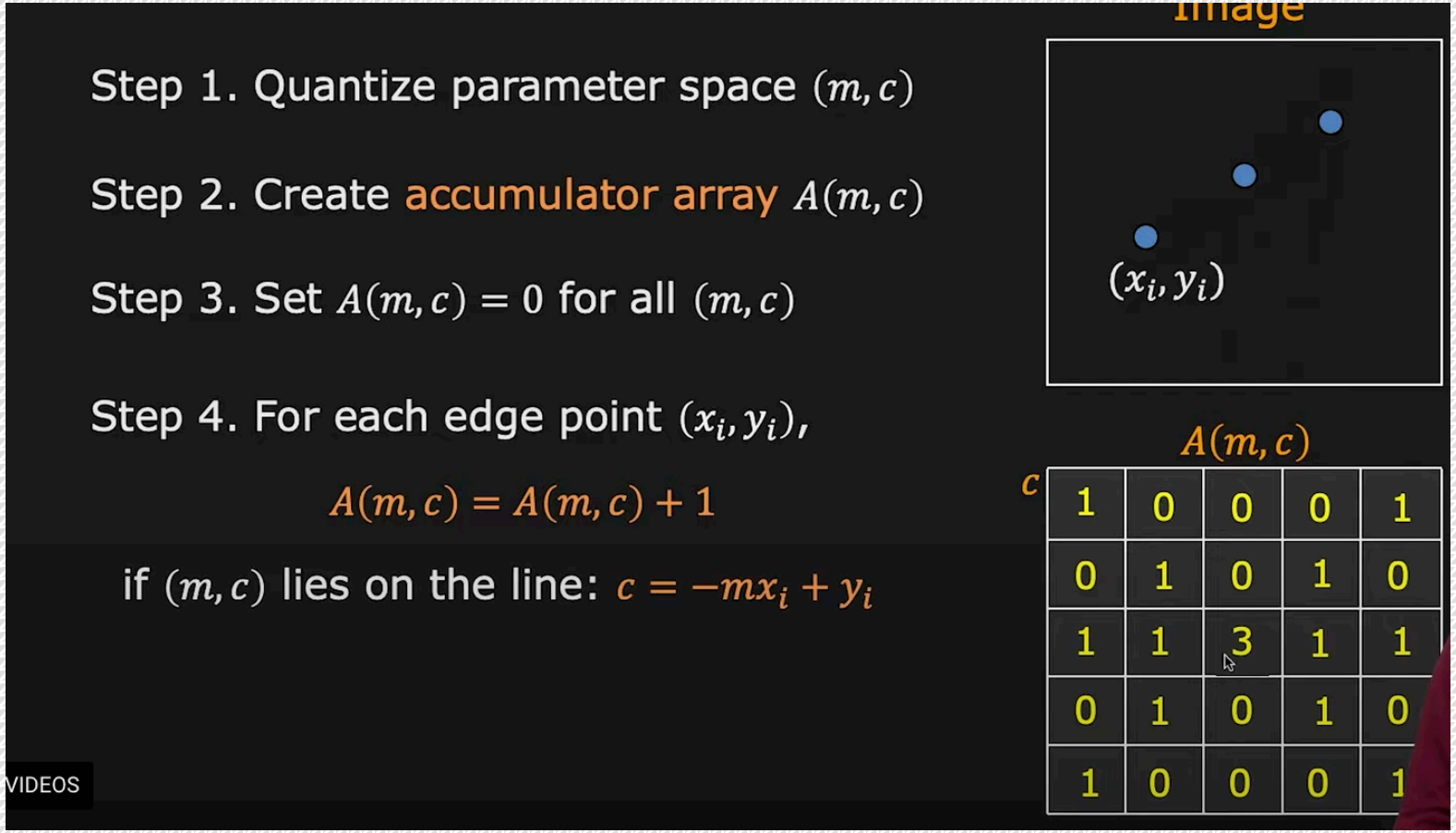
Lines

- Hough transform:
  - **直线检测**：参数平面相交直线最多的点对应图像空间中的直线



Lines

- Hough transform:



Lines

- Hough transform:

Issue: Slope of the line $-\infty \leq m \leq \infty$

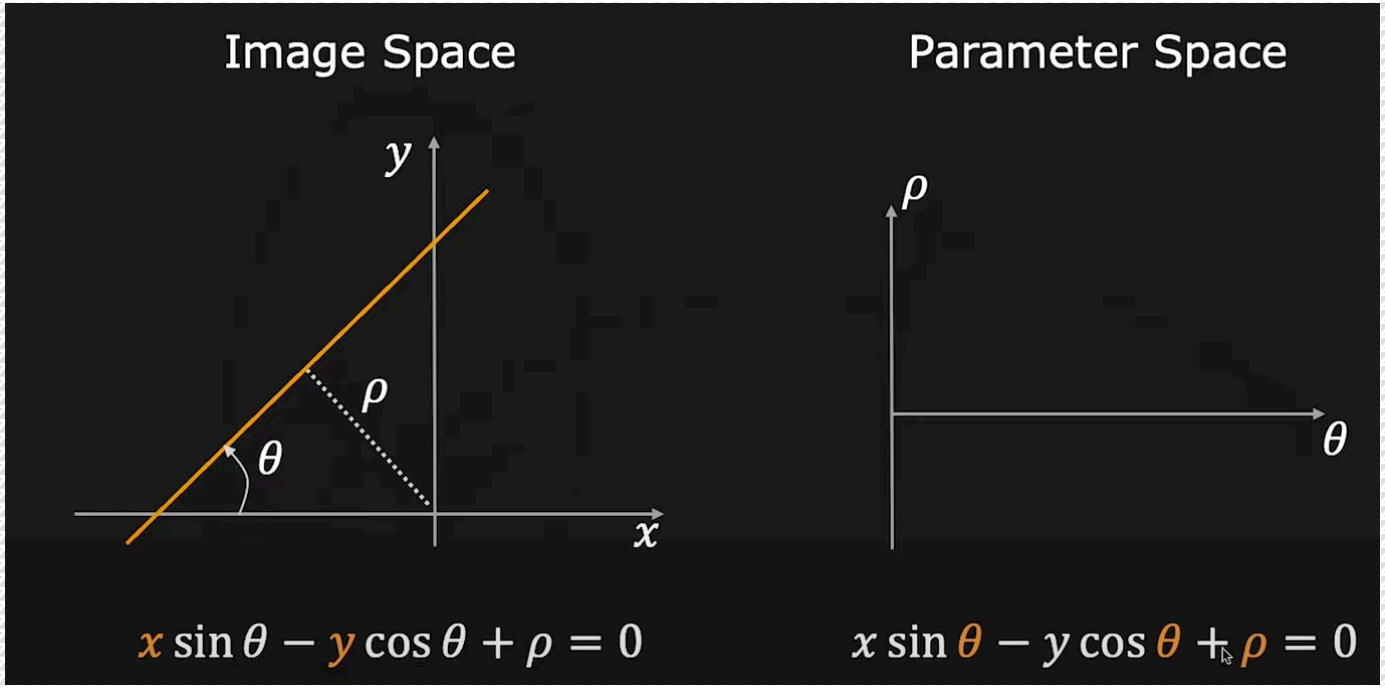
- Large Accumulator
- More Memory and Computation

Solution: Use $x \sin\theta - y \cos\theta + \rho = 0$

- Orientation θ is finite: $0 \leq \theta < \pi$
- Distance ρ is finite

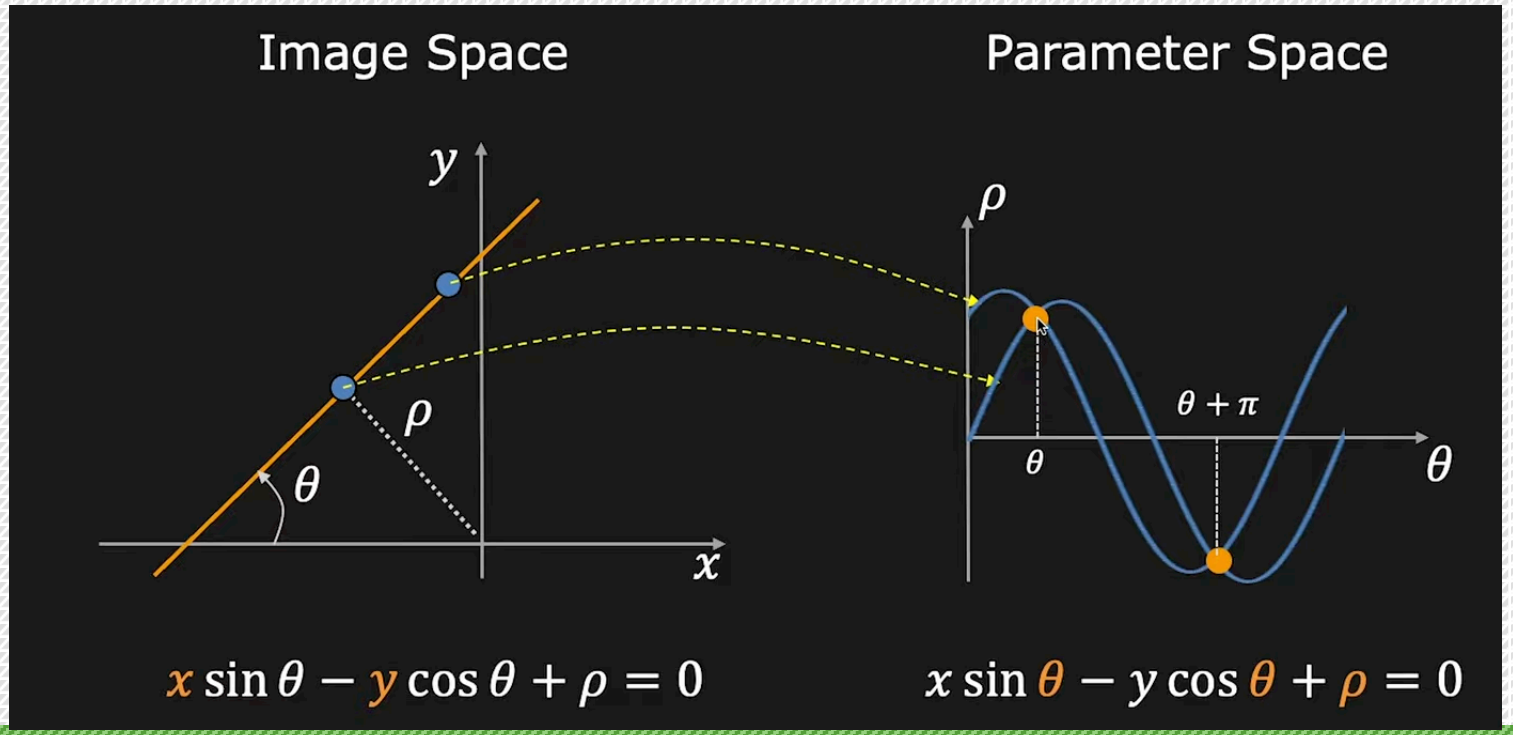
Lines

- Hough transform:
  - 参数空间的选择
    - ✓ 采用极坐标方式作为参数空间



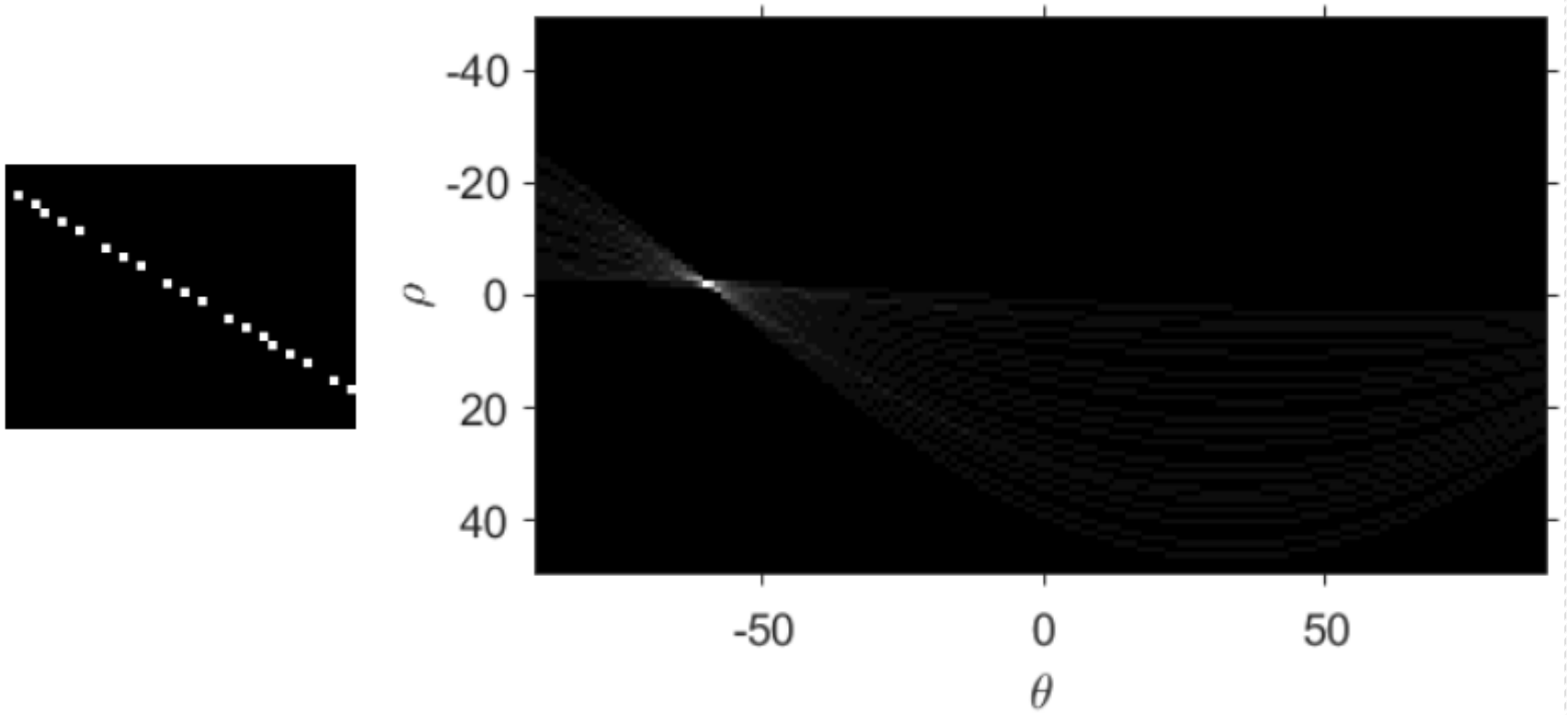
Lines

- Hough transform:
  - 参数空间的选择——极坐标
    - ✓ 图像空间中的每个点(x,y)被映射为一个(p,θ)极坐标空间中的正弦曲线
    - ✓ 图像空间中共线的点所对应的(p,θ)极坐标空间中正弦曲线相交于一点(p′ ,θ′)



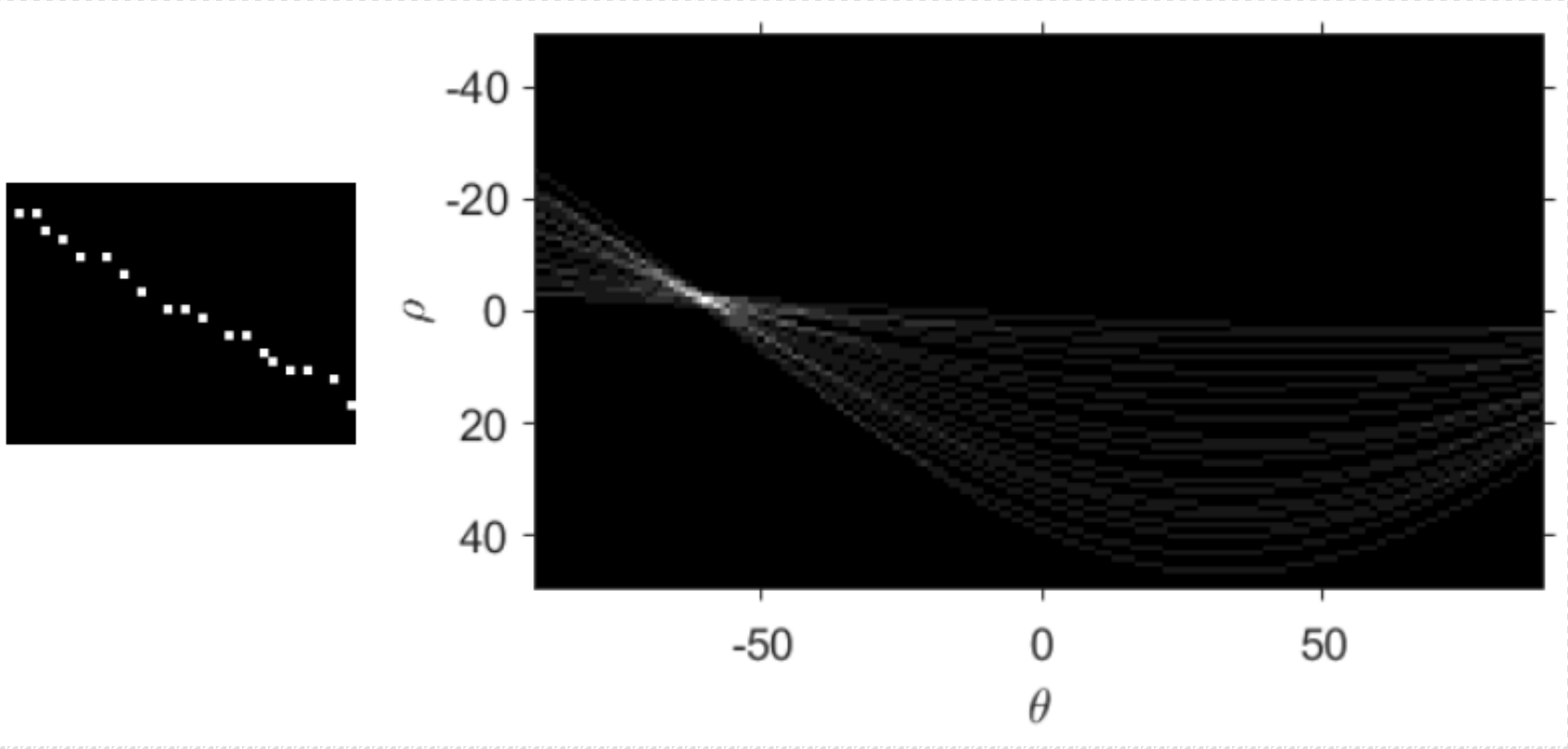
Lines

- Hough变换



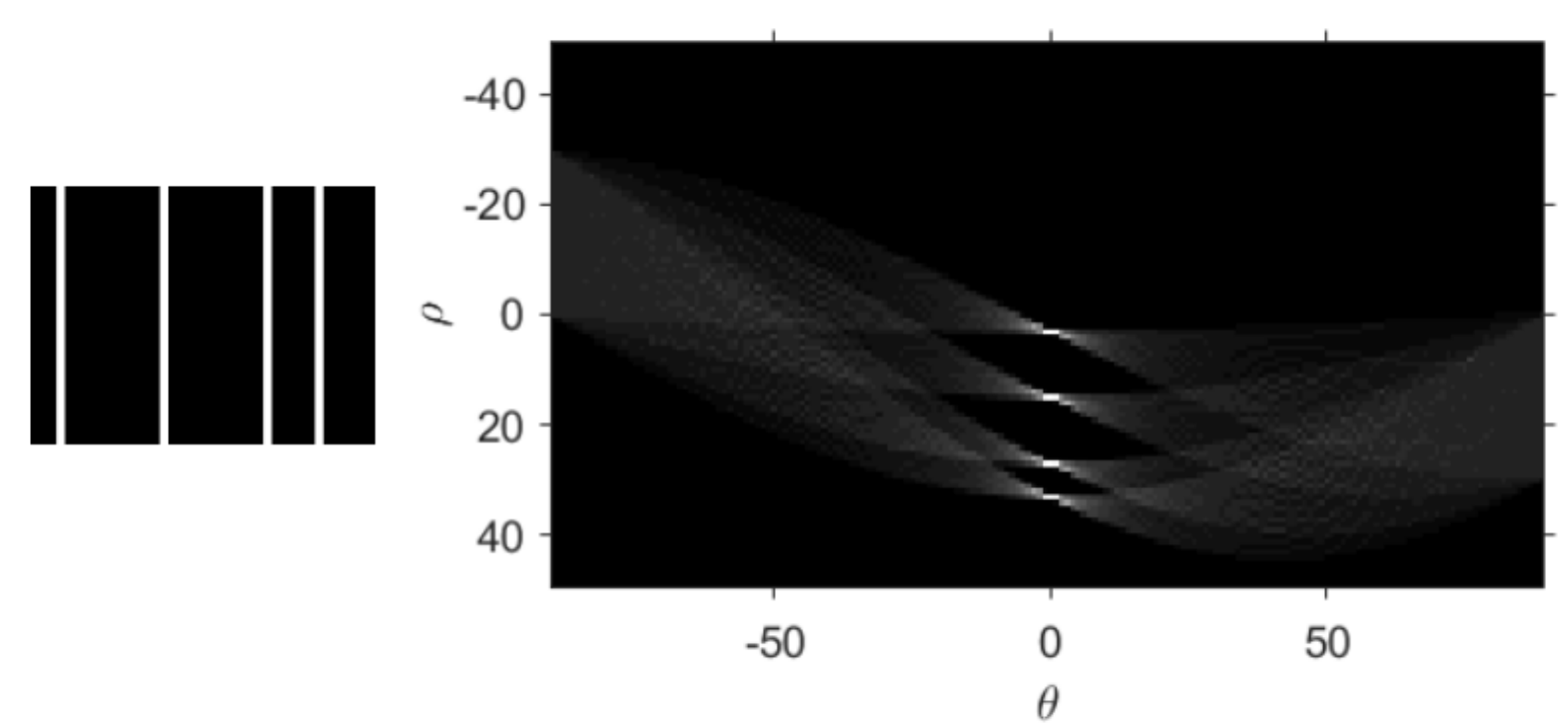
Lines

- Hough变换



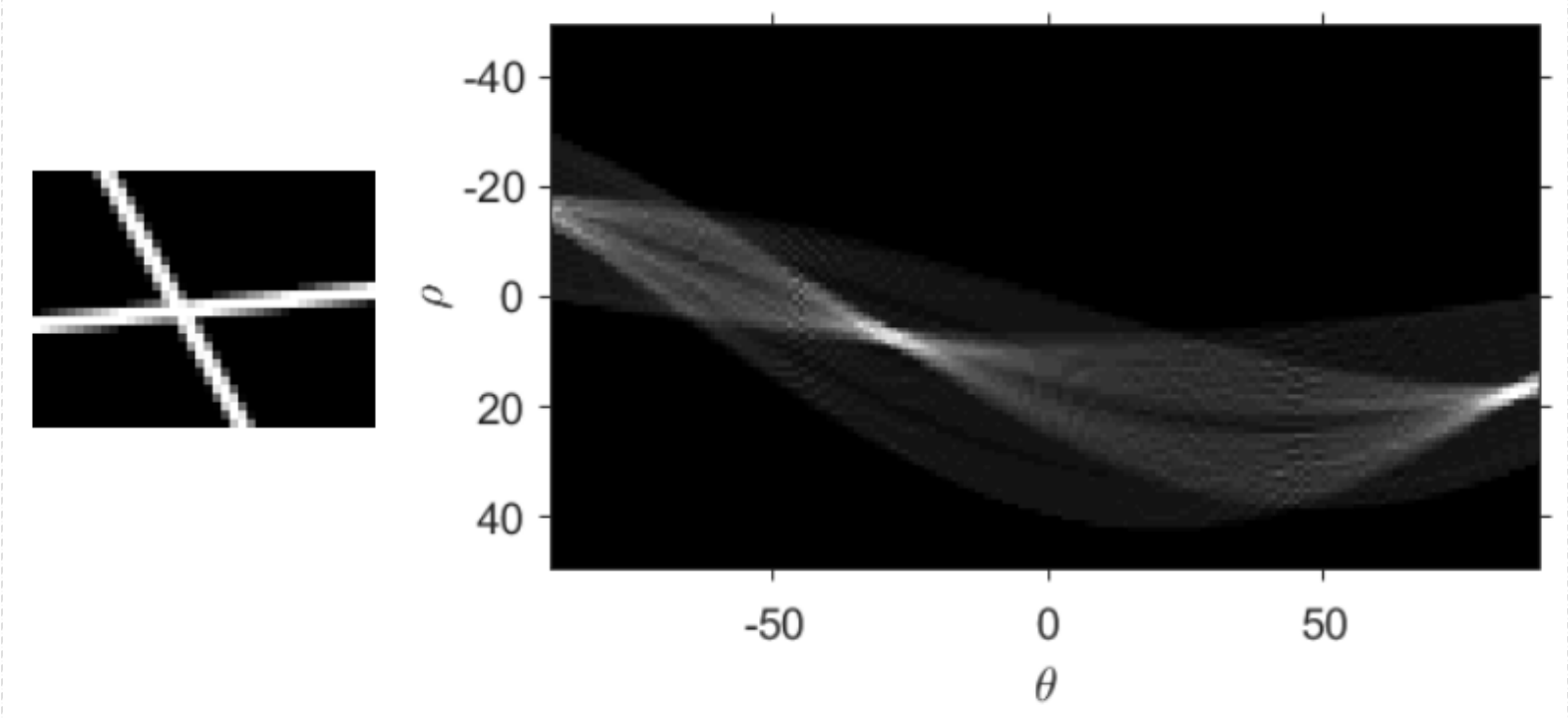
Lines

- Hough变换



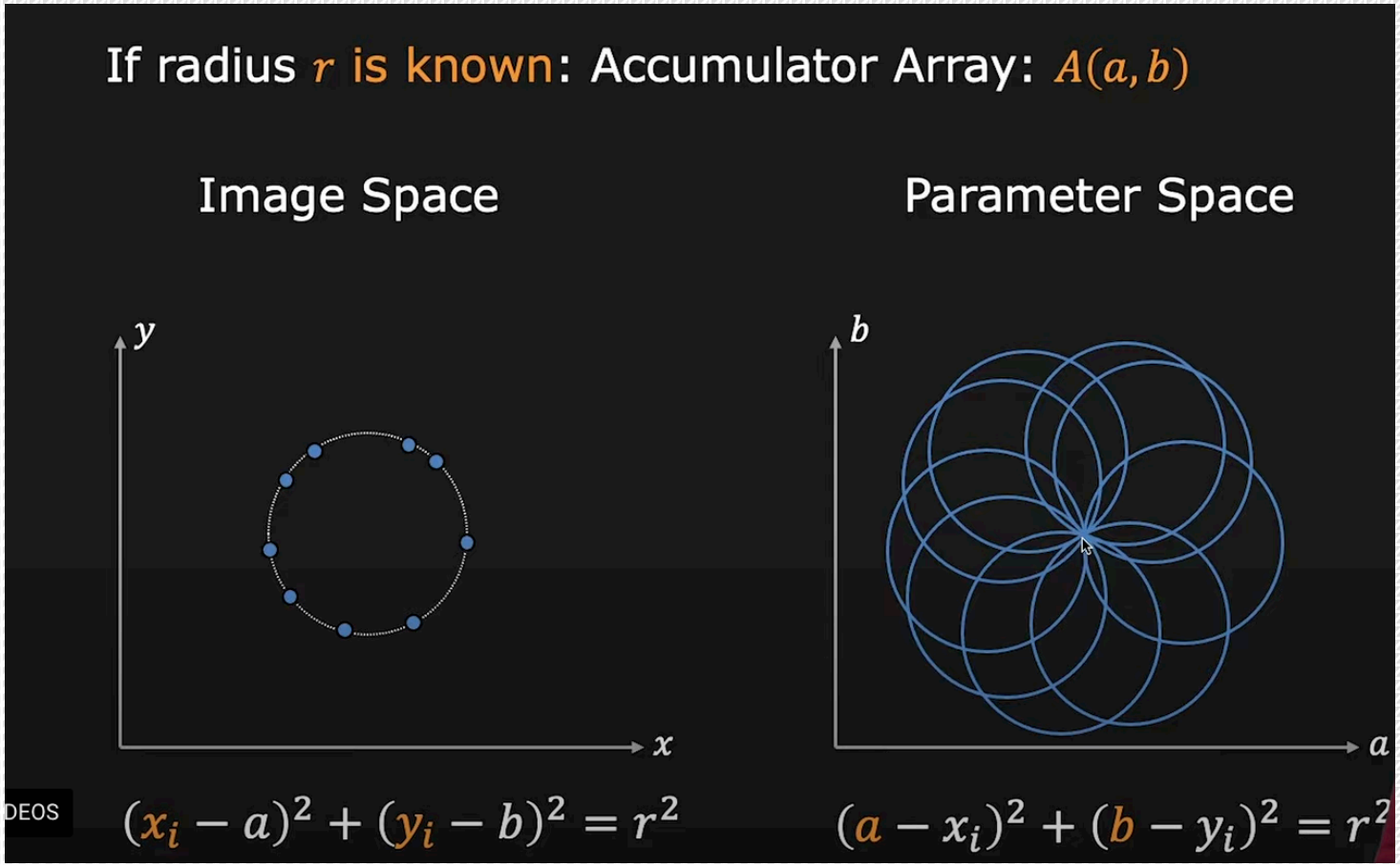
Lines

- Hough变换



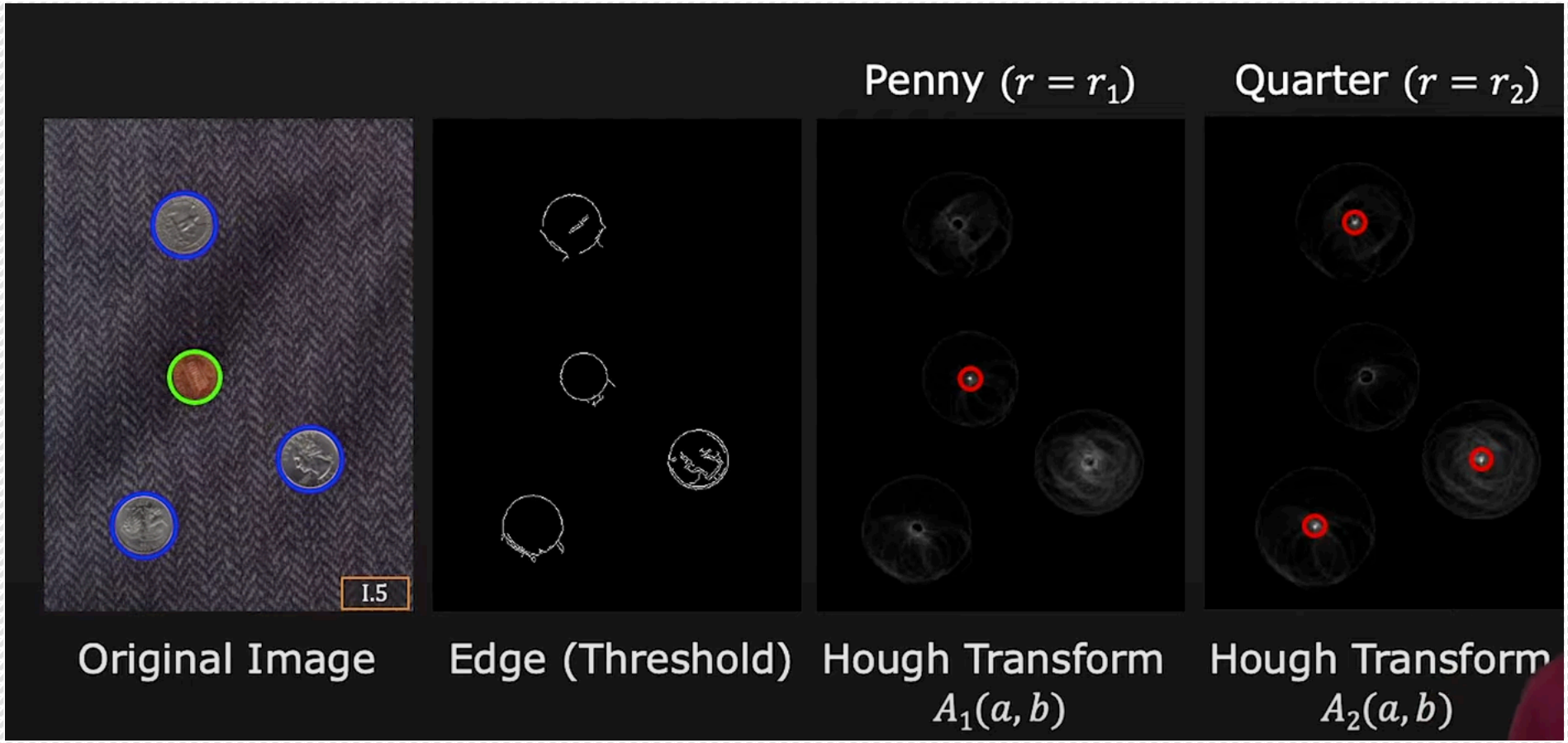
Circle Detection

- Hough变换



Circle Detection

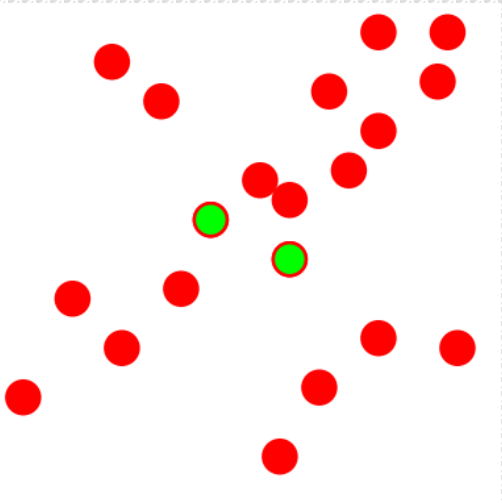
- Hough变换



Lines

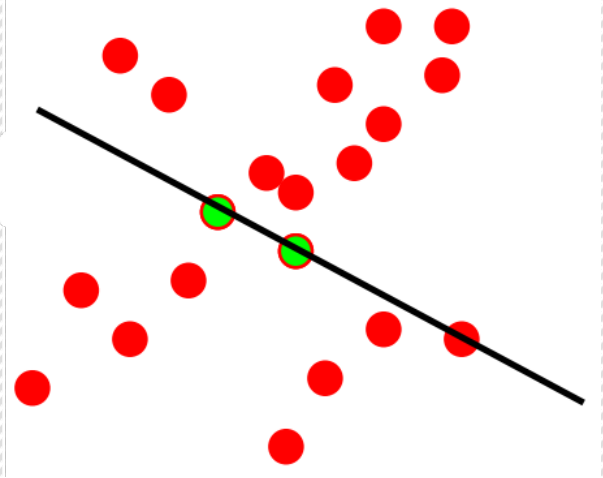
线检测——RANSAC (RANdom SAmple Consensus) 随机采样一致性

- 特征匹配时遇到误匹配，使用RANSAC算法进行过滤
- 从一组包含局外点(outlier)的观测数据中，通过迭代方式估计数学模型的参数
- RANSAC 试图踢掉那些outliers并找到一个在其计算时仅使用inliers的线性模型
- 步骤简单总结：
  - ✓ 随机采样K个点（K=2）



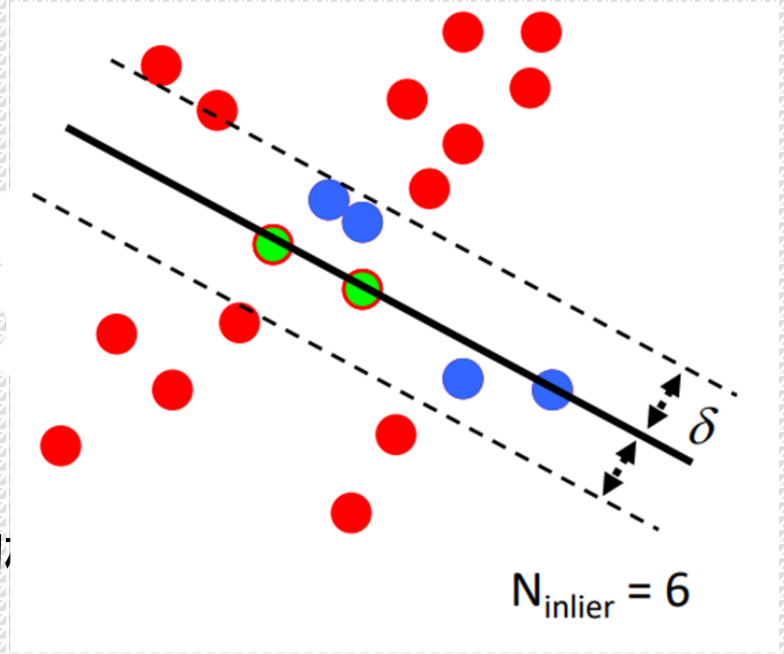
线检测——RANSAC(RANdom SAmple Consensus)

- 步骤简单总结：
  - ✓ 随机采样K个点（K=2）
  - ✓ 拟合一条直线



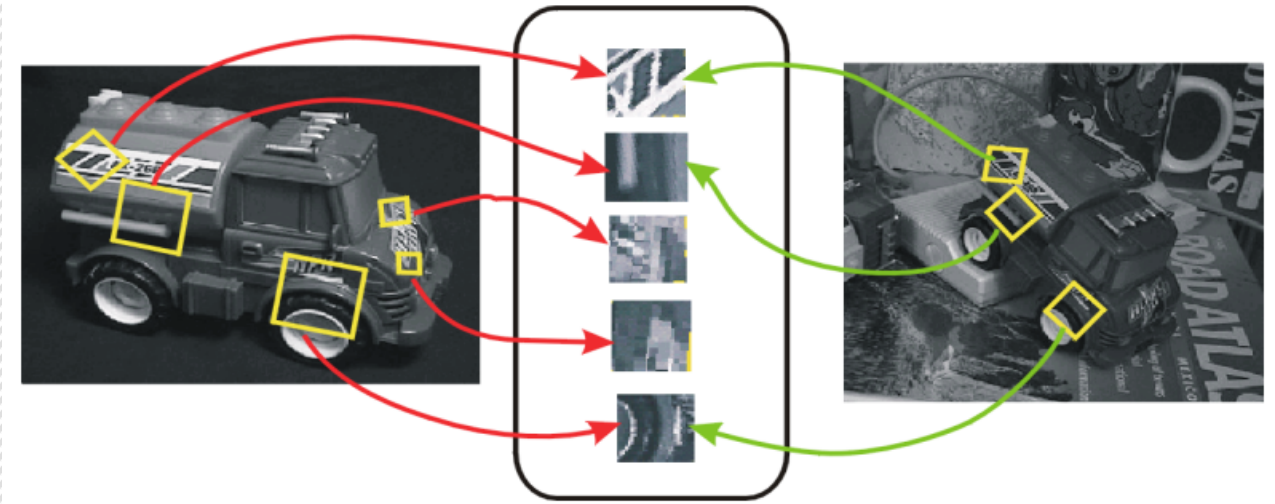
线检测——RANSAC

- 步骤简单总结：
  - ✓ 随机采样K个点（K=2）
  - ✓ 拟合一条直线
  - ✓ 统计局内点的个数
  - ✓ 重复上述过程M次，找到局内点数最大的
    型统计局内点的个数
  - ✓ 利用所有局内点重新估计直线



线检测——RANSAC

- 应用：特征匹配
  - ✓ 两张图中是否包含相同的特征？



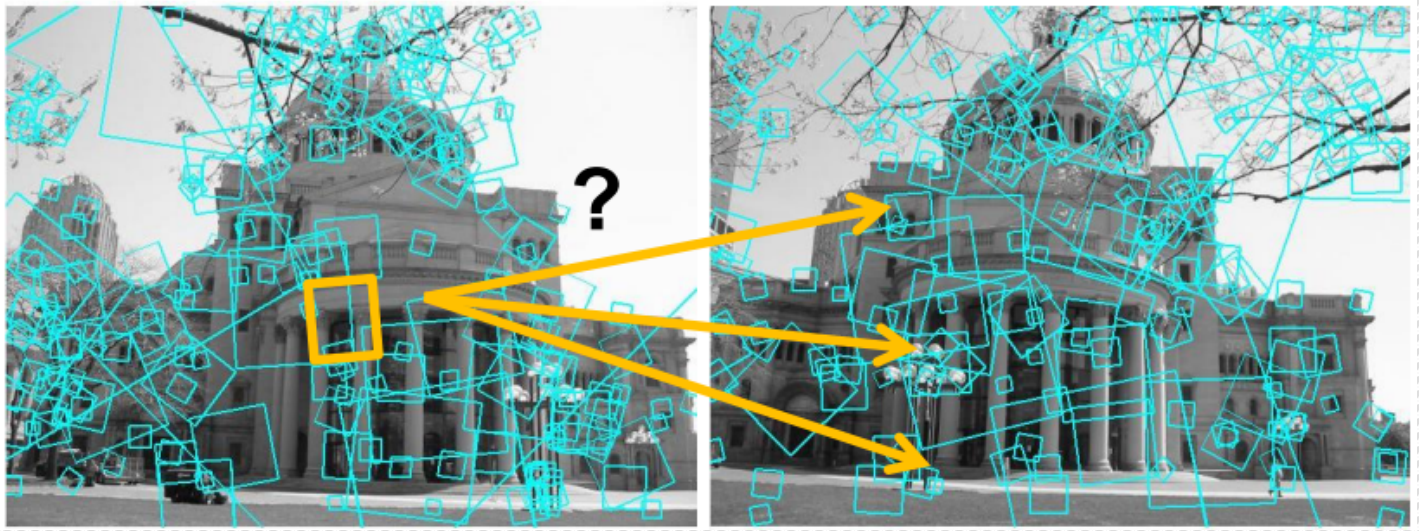
线检测——RANSAC

- 应用：特征匹配
  - ✓ 在每张图中找关键点和描述子

Lines

线检测——RANSAC

- 应用：特征匹配
  - ✓ 查找候选匹配：对于图像 1 中的每个关键点，在图像 2 中找到最相似的匹配



Lines

- **线检测——RANSAC**
  - 应用：特征匹配
    - ✓ 存在问题：关键点和相似特征较多，可能出现错误匹配

Talk from the top scientist

Andrew Paul McAfe:
https://v.youku.com/v_show/id_XNzY4MDQ2MzIw.html?playMode=pugv&frommaciku=1