# 图像处理与识别

# ——Part 7 图像特征描述

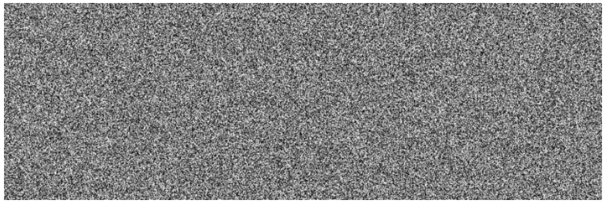
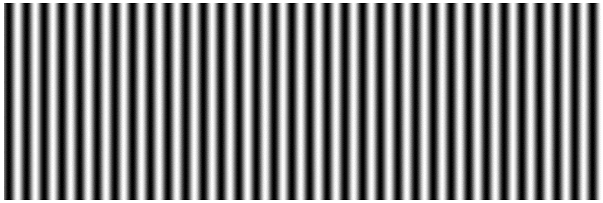
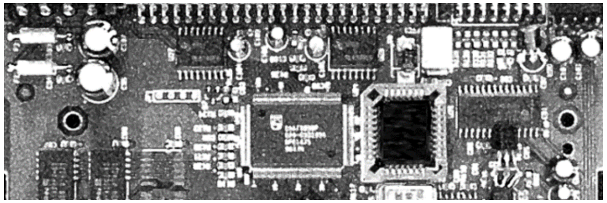
主讲：张磊

# Main Content

- Texture description
- Local features
- Image scale space
- SIFT
- Bag of Visual Words

# Texture

- Texture is for Region description
- No specific definition (smoothness, roughness, regularity)

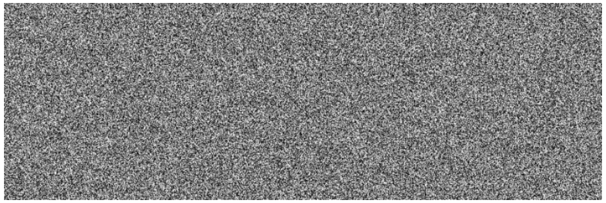




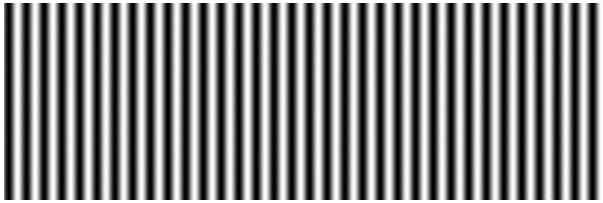
纹理反映物体表面缓慢变化或周期变化的表面结构组织排列属性，但不能反映物体本质属性。

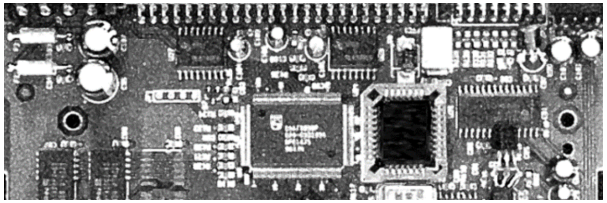
# Texture

Methods for descripting Texture of an image region

- Statistical method (used frequently)
- Structural method
- Spectrum method







A widely used metric for texture of an image or region is the statistical moments (grayscale level histogram)

# Texture

Methods for descripting Texture of an image region

- **Statistical method**

For a random variable $z$ (grayscale value)

Its n-order moment is

$$\mu_n(z) = \sum_{i=0}^{L-1} (z_i - \bar{z})^n p(z_i)$$

$p(z_i)$ is gray histogram

where

$$\bar{z} = \sum_{i=0}^{L-1} z_i p(z_i)$$

Note $\mu_0 = 1$, $\mu_1 = 0$, $\mu_2(z) = \sigma^2(z)$ (important and for gray contrast)

Higher order moment……

# Texture

Methods for descripting Texture of an image region

- **Statistical method**

A widely used metric for texture of an image or region is the statistical moments (grayscale level histogram)

Gray histogram based Consistency metric (灰度一致性)

$$U(z) = \sum_{i=0}^{L-1} p^2(z_i),$$

$achieves\ to\ maximum\ for\ constant\ image$

Gray histogram based Entropy metric(熵)

$$e(z) = -\sum_{i=0}^{L-1} p(z_i) log_2 p(z_i),$$

$achieves\ zero\ for\ constant\ image\ (no\ variation)$

# Texture

Methods for descripting Texture of an image region

- **Statistical method**

Problem: no spatial information (relative position among pixels) is used based only on histogram. (直方图不能反映像素空间的相对位置信息，且直方图对应图像的不唯一性)

Both gray distribution and spatial relation are important.

Let Q be an operator, describing the spatial relation of two pixels $z_i$ and $z_j$ $(0 \leq i, j \leq L-1)$.

Given a matrix **G** $\in \Re^{L \times L}$, where entry $g_{i,j} = Num\left\{Q\left(z_i, z_j\right)\right\}$

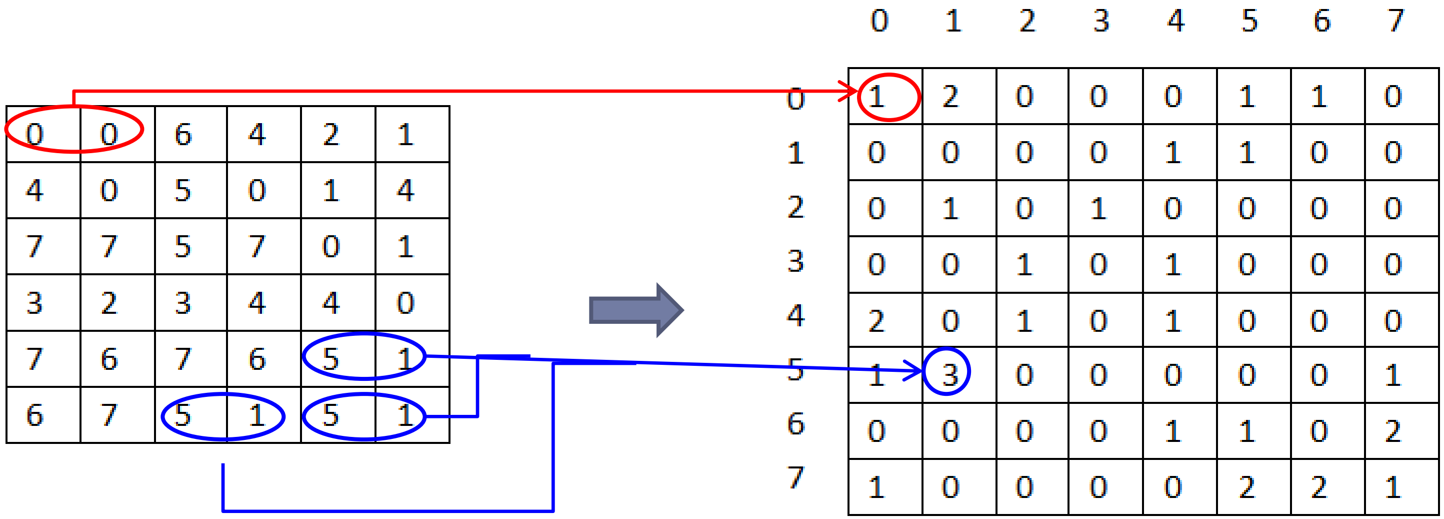
$Gray\ level\ Co-occurrence\ matrix\ (GLCM)$: 共生矩阵

# Texture

Methods for descripting Texture of an image region

For a gray image with gray lever L=8

$Q(z_i, z_j)$ represents the neighborhood relation



|  |  |  |  |  |  |
|---|---|---|---|---|---|
| 0 | 0 | 6 | 4 | 2 | 1 |
| 4 | 0 | 5 | 0 | 1 | 4 |
| 7 | 7 | 5 | 7 | 0 | 1 |
| 3 | 2 | 3 | 4 | 4 | 0 |
| 7 | 6 | 7 | 6 | 5 | 1 |
| 6 | 7 | 5 | 1 | 5 | 1 |

Image with L=8

|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 4 | 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 5 | 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 |
| 6 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 |
| 7 | 1 | 0 | 0 | 0 | 0 | 2 | 2 | 1 |

Gray level Co-occurrence matrix 8x8

# Texture

## Methods for descripting Texture of an image region

### Co-occurrence matrix Analysis

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 2 |
| 1 | 0 | 0 | 0 | 0 | 2 | 2 | 1 |

Gray level Co-occurrence matrix G 8x8

→ *depend on Q operator*

*Specific descriptors of G:*

*Maximum probability*: $\max(p_{i,j})$

*Correlation*: $$\sum_{i=1}^{L}\sum_{j=1}^{L}\frac{(i-m_r)(j-m_c)p_{ij}}{\sigma_r\sigma_c}$$

*Contrast*: $\sum_{i=1}^{L}\sum_{j=1}^{L}(i-j)^2 p_{ij}$

*Energy*: $\sum_{i=1}^{L}\sum_{j=1}^{L}p_{ij}^2$

*Homogeneity(同质性)*: $\sum_{i=1}^{L}\sum_{j=1}^{L}\frac{p_{ij}}{1+|i-j|}$

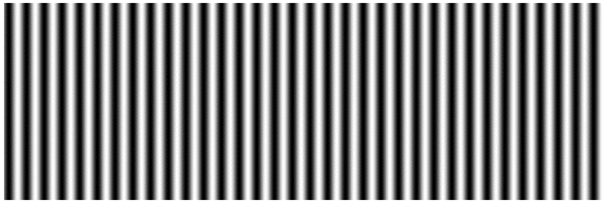
*Entroy*: $$-\sum_{i=0}^{L}\sum_{j=0}^{L}p_{ij}log_2 p_{ij}$$

# Texture

## Methods for descripting Texture of an image region

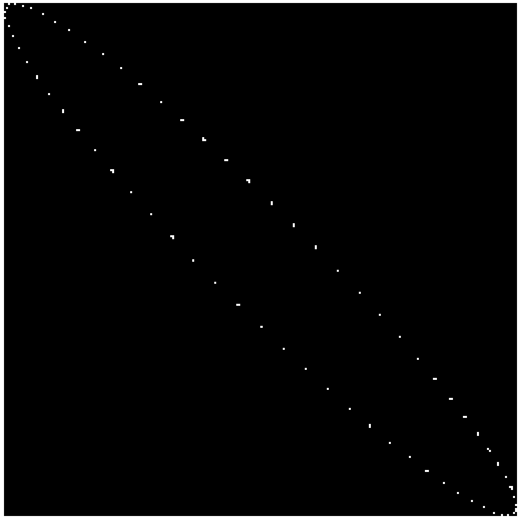
Co-occurrence matrix Analysis

Steps for texture description:



Q：右侧紧邻一个像素

→



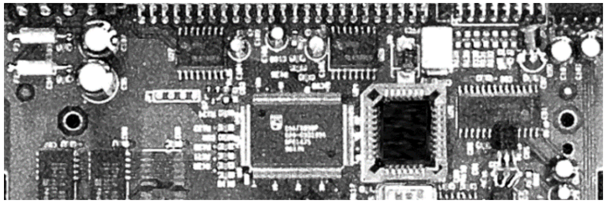
灰度共生矩阵

# Texture

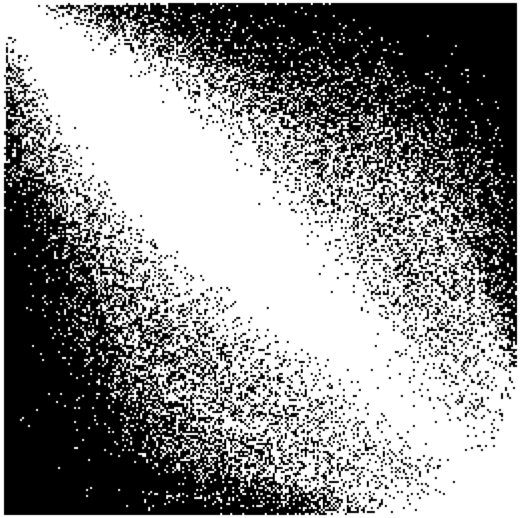
Methods for descripting Texture of an image region

Co-occurrence matrix Analysis

Steps for texture description:



Q：右侧紧邻一个像素



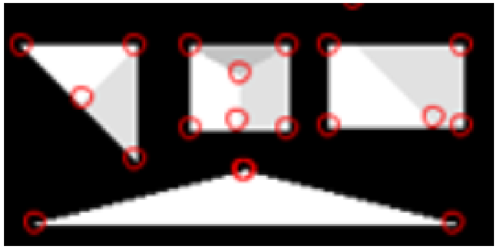
灰度共生矩阵

扩展：灰度-梯度共生矩阵(Gray-Gradient Co-occurrence Matrix)

# Local Features

History of Local description (Key point extraction):

1977, Moravec proposed corner point feature, the origin of “points of interest” concept. Sensitive to noise.



且不具有旋转不变性！

1988, Harris proposed a robust Harris corner feature. Invariant to rotation and gray change. (still in use)

但不具有尺度不变性！

1990, Lindeberg proposed discrete signal scale space theory, and proved that only Gaussian kernel filters can be used for image smooth (image scale space)

[1] Witkin A.P.“Scale Space Filtering,” IJCAI, 1983.
[2] Koenderink J.J.“The Structure of Image,” Biological Cybernetics, 1984.

# Local Features

History of Local description:

Next, Mikolajczyk proposed Harris-Laplacian and Harris-Affine detectors.  The former combines Harris corner detector with Gaussian scale space, such that the corner features are scale-invariant; the latter can detect the feature under affine transformation, and affine-invariant.

（Milestone） in 2000, David Lowe proposed very efficient SIFT (Scale Invariant Feature Transform) local feature descriptor.  SIFT is invariant to rotation, scale, affine Transformation and view-angle.

# Local Features

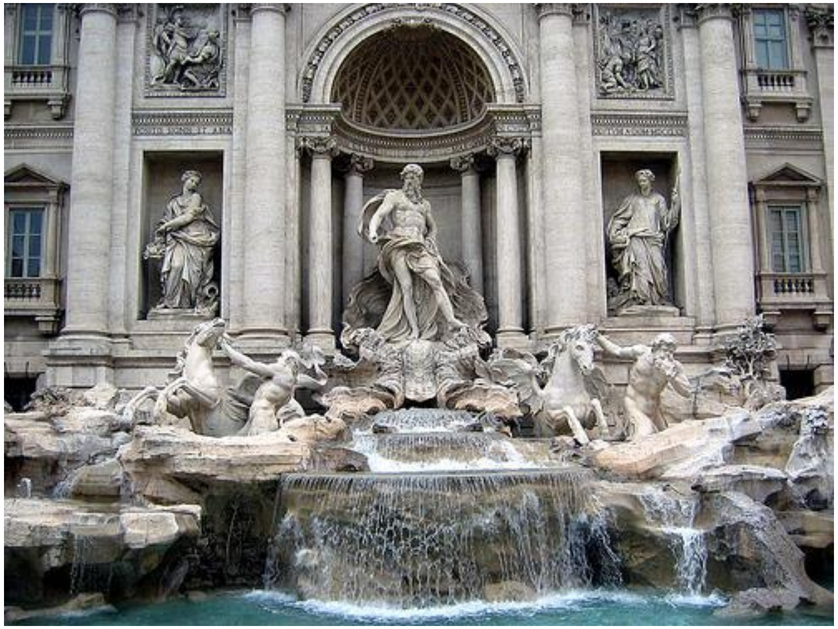
History of Local description:

In 2006, Bay proposed a SURF (speed up robust features) by following the idea of David Lowe, the velocity of feature extraction was improved by combining integral image and Haar wavelet.

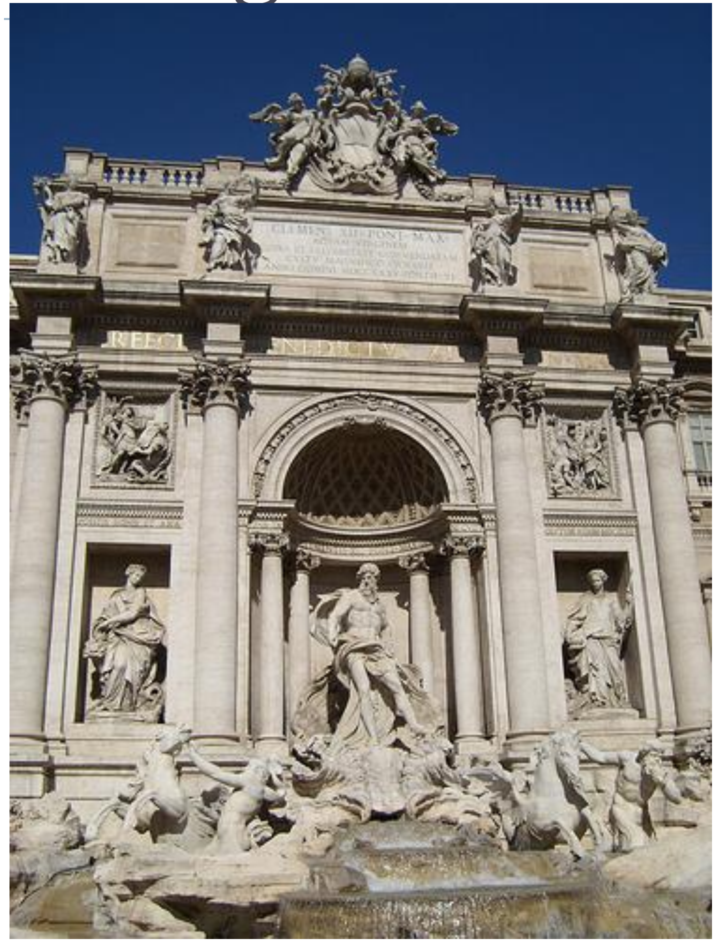
The key property of local feature:

Invariant to rotation, scale, affine transform, grayscale value, intensity, etc.

# Image matching

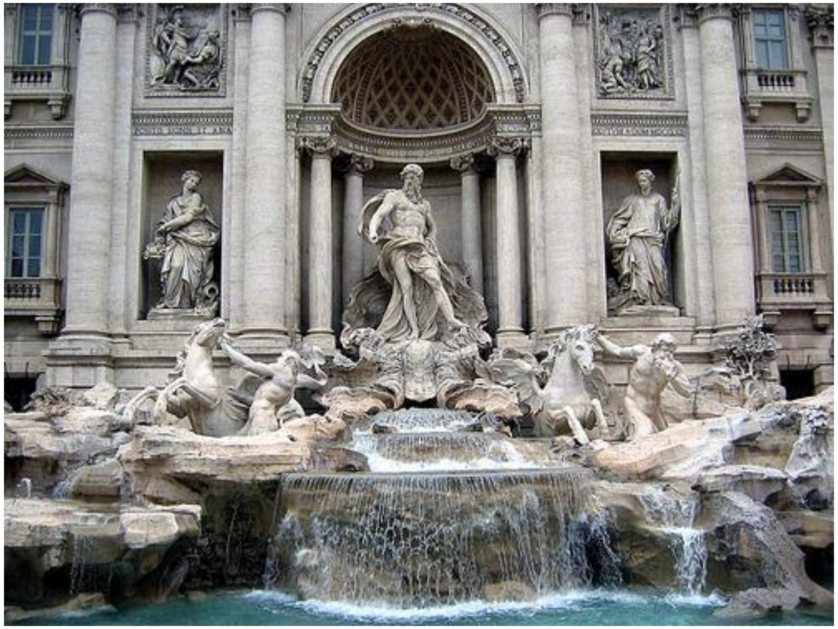


by Diva Sian

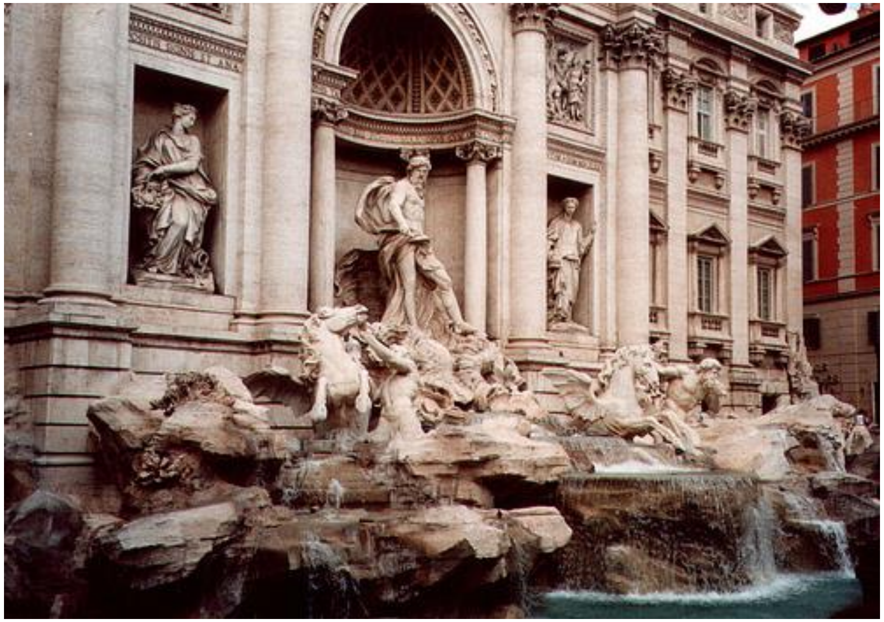


by swashford

# Harder case

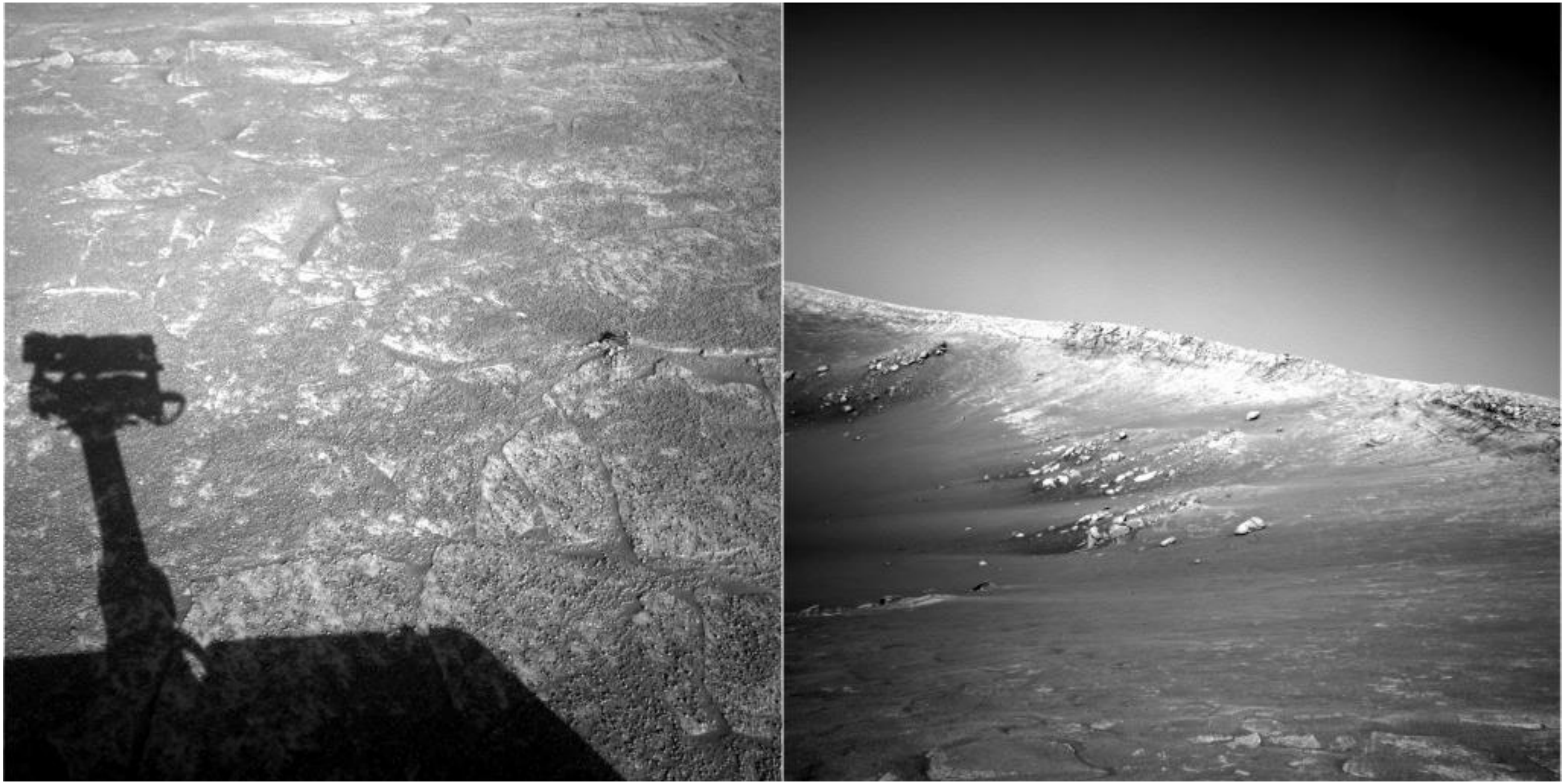


by Diva Sian



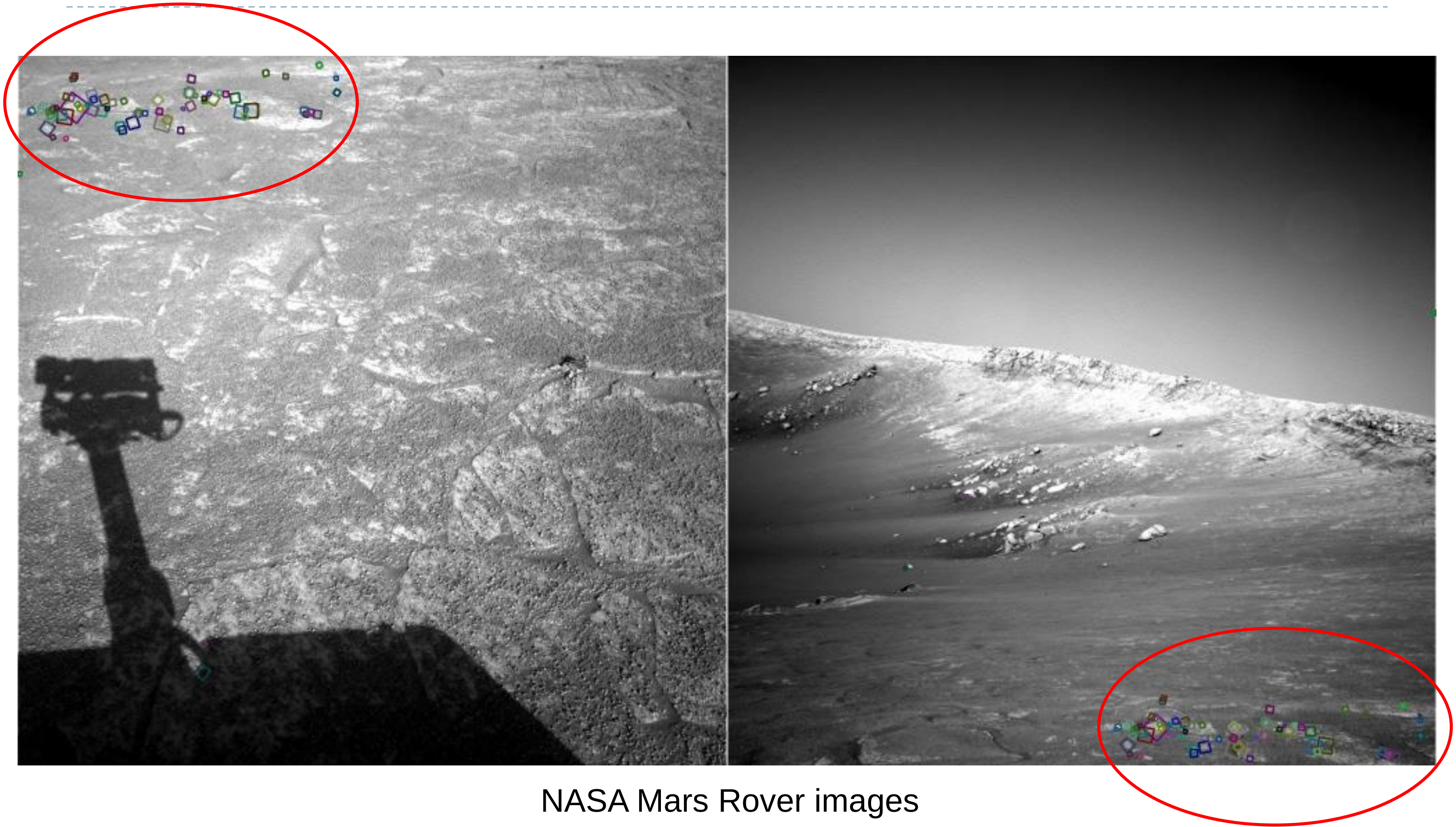
by scgbt

# Harder still?



NASA Mars Rover images

# Answer below (look for tiny colored squares…)



NASA Mars Rover images
with SIFT feature matches
Figure by Noah Snavely

# Features

"ALL IS VANITY"
C. Allan Gilbert

*All is Vanity*, by C. Allan Gilbert, 1873-1929

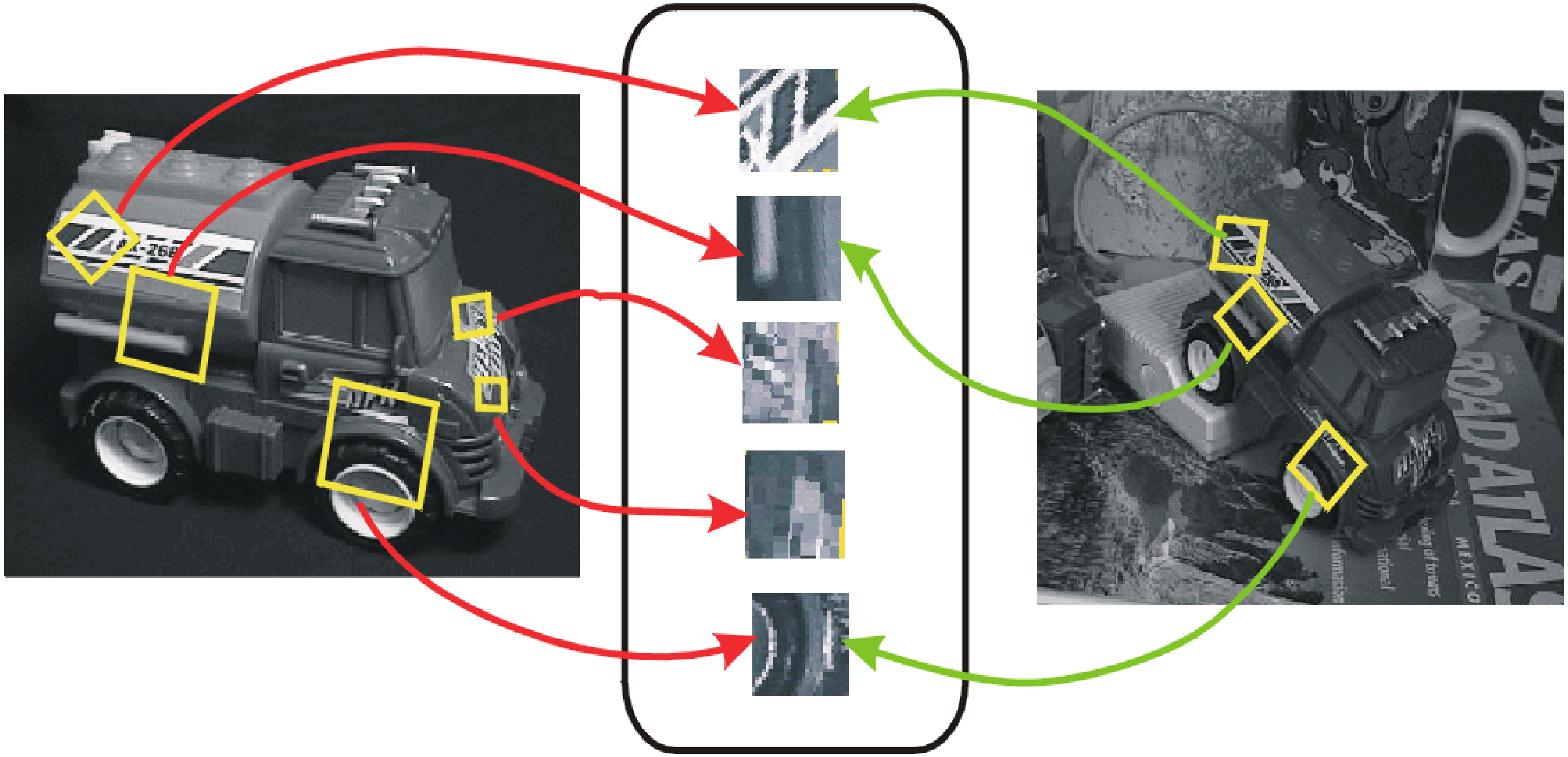
Readings

- Szeliski, Ch 4.1
- (optional) K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. In PAMI 27(10):1615-1630
  - http://www.robots.ox.ac.uk/~vgg/research/affine/det_eval_files/mikolajczyk_

# Invariant local features

Find features that are invariant to transformations

- geometric invariance:  translation, rotation, scale
- photometric invariance:  brightness, exposure, …



Feature Descriptors

# More motivation…

Feature points are used for:

- Image alignment (e.g., mosaics)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- … other

# Invariance

Suppose we are comparing two images $I_1$ and $I_2$

- $I_2$ may be a transformed version of $I_1$
- What kinds of transformations are we likely to encounter in practice?

# Invariance

Suppose we are comparing two images $I_1$ and $I_2$

- $I_2$ may be a transformed version of $I_1$
- What kinds of transformations are we likely to encounter in practice?

We'd like to find the same features regardless of the transformation

- This is called transformational *invariance* 尺度不变性和旋转不变性
- Most feature methods are designed to be invariant to
  - Translation, 2D rotation, scale
- They can usually also handle
  - Limited 3D rotations (SIFT works up to about 60 degrees)
  - Limited affine transformations (some are fully affine invariant)
  - Limited illumination/contrast changes

# How to achieve invariance

Need both of the following:

1. Make sure your detector is invariant
   - Harris is invariant to translation and rotation
   - Scale is trickier(棘手)
     - common approach is to detect features at many scales using a Gaussian pyramid (e.g., MOPS, multi-scale oriented patches)
     - More sophisticated methods find “the best scale” to represent each feature (e.g., SIFT)
2. Design an invariant feature *descriptor*
   - A descriptor captures the information in a region around the detected feature point
   - The simplest descriptor: a square window of pixels
     - What’s this invariant to?
   - Let’s look at some better approaches…

# Scale Space

Earliest: Signal Pyramid (信号金字塔化)

Step 1: Low-pass filter (e.g. Gaussian filters) for signal smooth

Step 2: Down-sampling on the smooth signal with 1/2.

Then, signals with different scales can be obtained.

*But, it is lack of theory*

# Scale Space

In 1983, Witkin proposed Scale Space Representation of signals by using a series of Gaussian filters with increasing width σ.

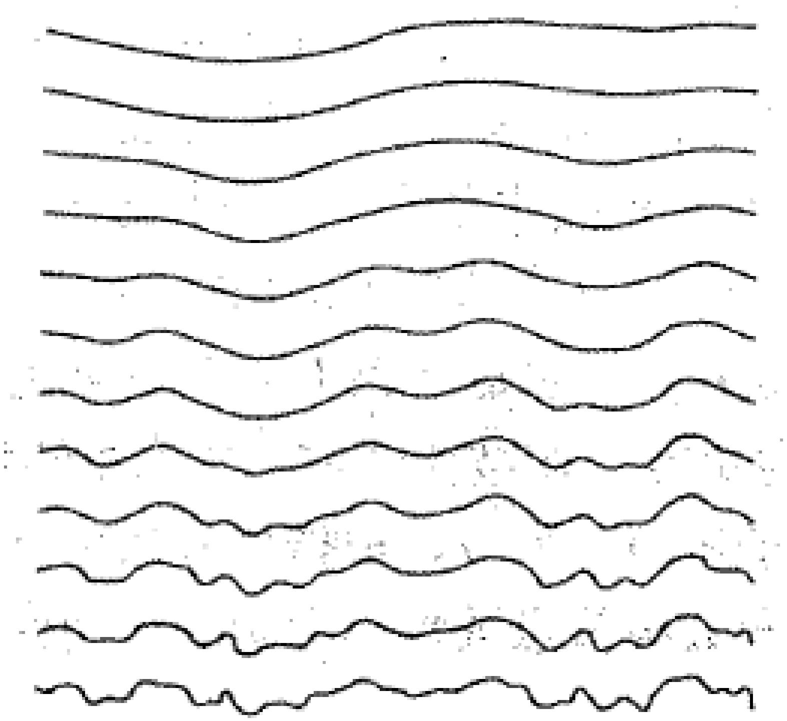
*Q: How about other filters other than Gaussian filters?*

*A: Lindeberg et al., proved that Gaussian kernel is the only one for Scale Transformation.*

*Invariance to translation, scale, rotation*

不同高斯核组成的尺度空间是规范的和线性的，并且满足性质{加权平均和有限孔径效应、层叠平滑、局部极值递减、尺度不变性}。

# 图像特征描述



一维信号平滑，σ(尺度参数)从大到小(上－下)

A well-known Local Feature Descriptor in Pre-deep learning era

SIFT (Scale Invariant Feature Transform)

David Lowe
University of British Columbia

# Scale Invariant Feature Transform

**Construct Scale Space**:

Perform scale transformation on the raw image, and obtain multiple image sequence with different scales

Scale space with Gaussian kernel (filter)

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

where $G(x,y,\sigma)$ is Gaussian convolution kernel,$\sigma$ means the smoothness,

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$$

Objective: find out the positions of key points in the scale space.

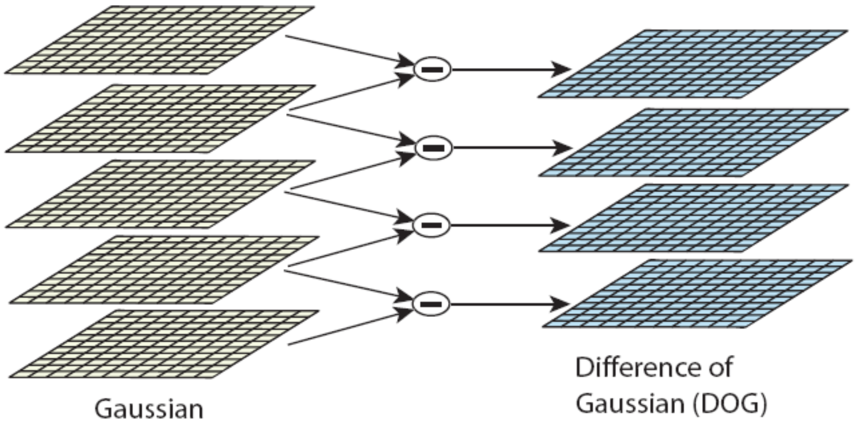
Action: To quickly find out these key points, David Lowe proposed the difference of Gaussian (DoG) scale space D

# Scale Invariant Feature Transform

**Construct Difference Scale Space(构造高斯差分尺度空间)**:

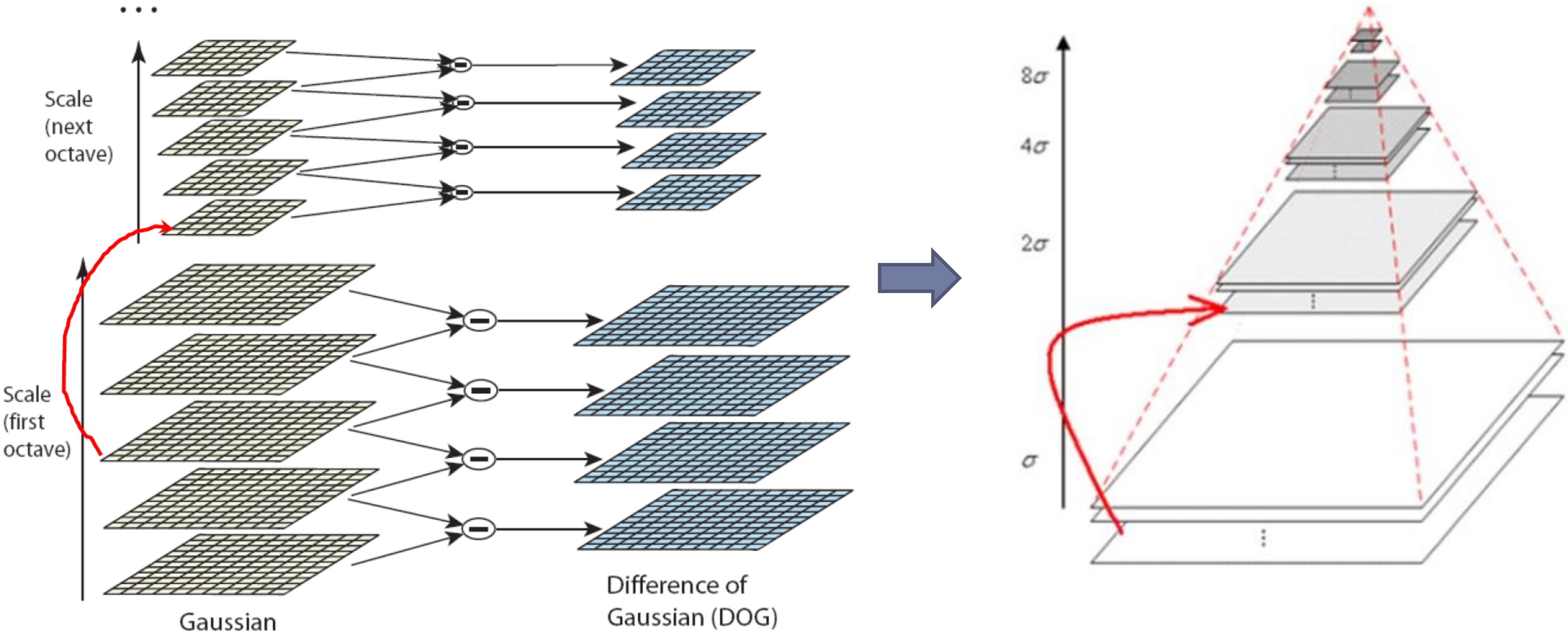
How to construct difference of Gaussian scale space D(x,y,σ) ?

$$D(x,y,\sigma)=(G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y)$$
$$=L(x,y,k\sigma)-L(x,y,\sigma)$$



# Scale Invariant Feature Transform

## Construct Gaussian Pyramid (构造高斯金字塔):



尺度 $= 2^{i-1}k^{j-1}\sigma$，第$i$个octave，第$j$层，$k = 2^{\frac{1}{S}}$ ($S = 2$表示每组要检测的尺度数)

# Scale Invariant Feature Transform

**Construct Gaussian Pyramid (高斯金字塔)**:

Search the local minima feature point in DoG scale space (高斯差分尺度空间)

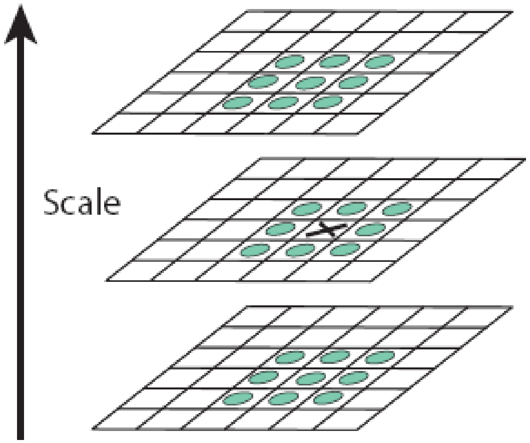
9+9+8=26 points

If the current point is larger

or smaller than 26 points,

local minima (false)

End



在差分尺度空间内找极值

Remove the points of low contrast and edge effect points.

Then, the key points have been found.

# Scale Invariant Feature Transform

Basic idea:

- Take **16x16** square window around detected feature（key points）
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

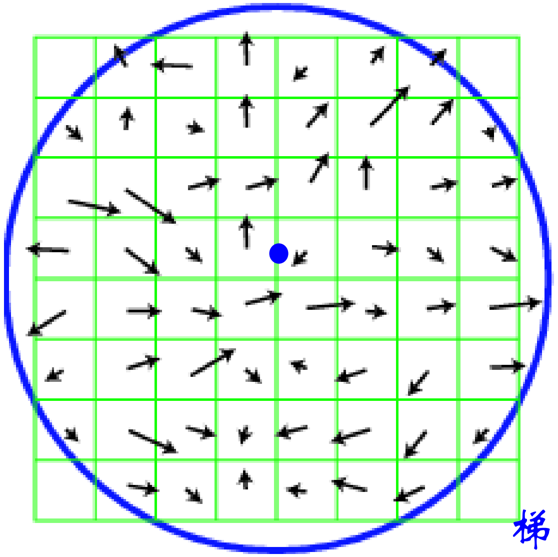
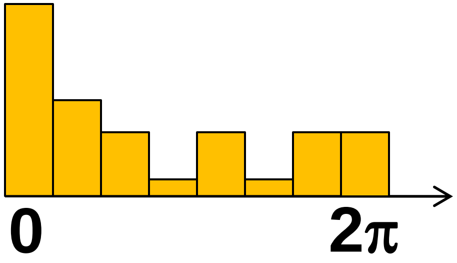
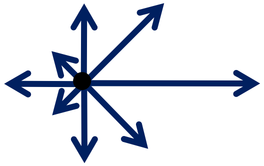


Image gradients



angle histogram



梯度幅度：
箭头长度

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2}$$

梯度方向：
箭头方向

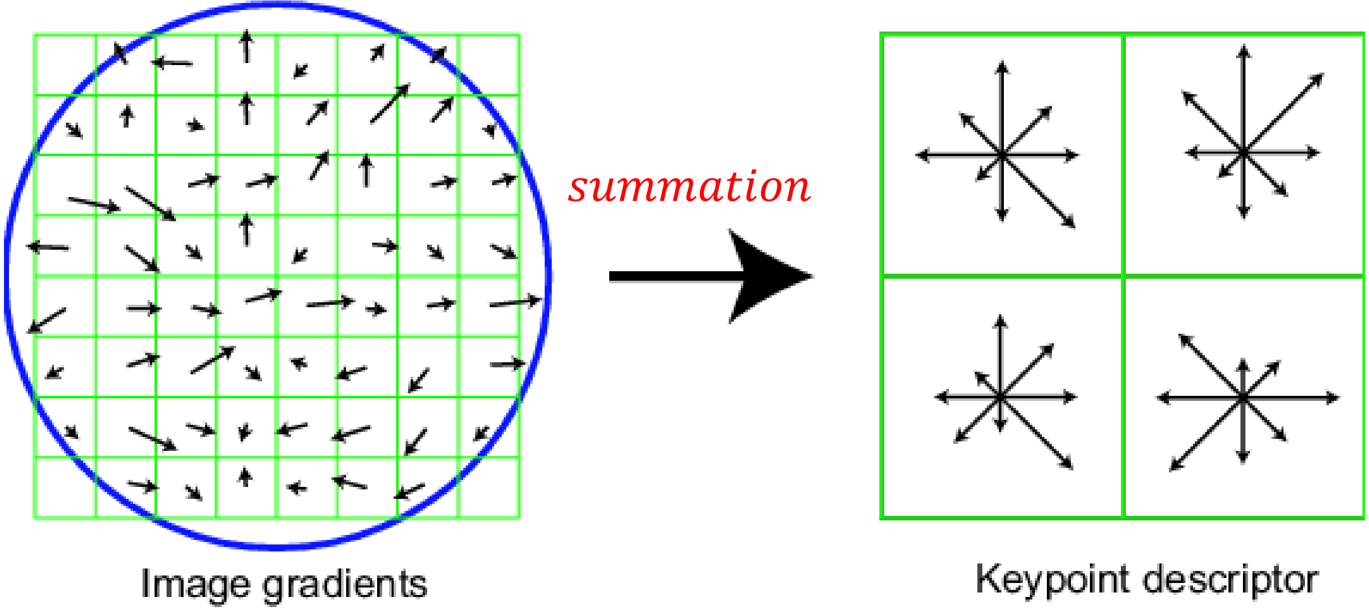
$$\theta(x,y)=\arctan\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right)$$

Adapted from slide by David Lowe

# SIFT descriptor

## Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram (summation of gradient) for each cell
- 16 cells * **8 orientations** = 128 dimensional descriptor for each keypoint
- Normalize for illumination reduction



- SIFT feature points match based on Euclidean distance ($d_{nearest}/d_{subnearest}<T$)

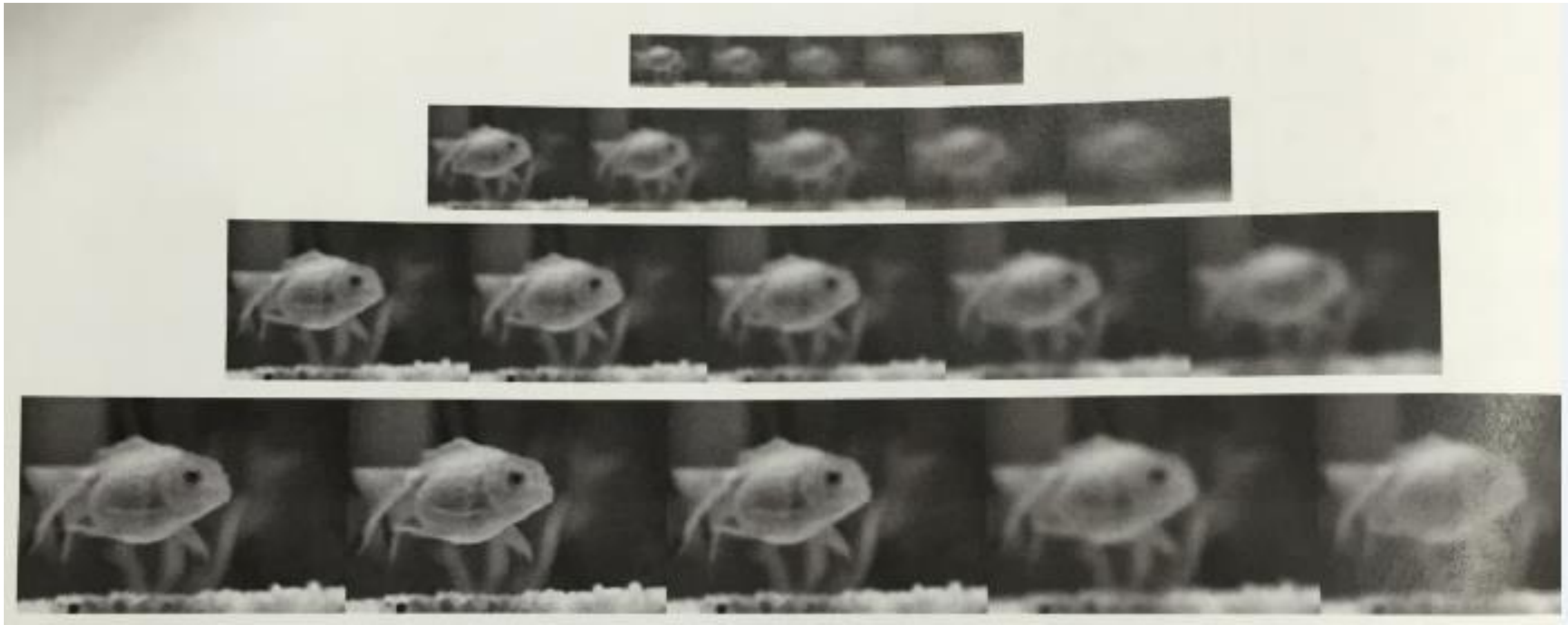
T=0.4~0.6;

Adapted from slide by David Lowe

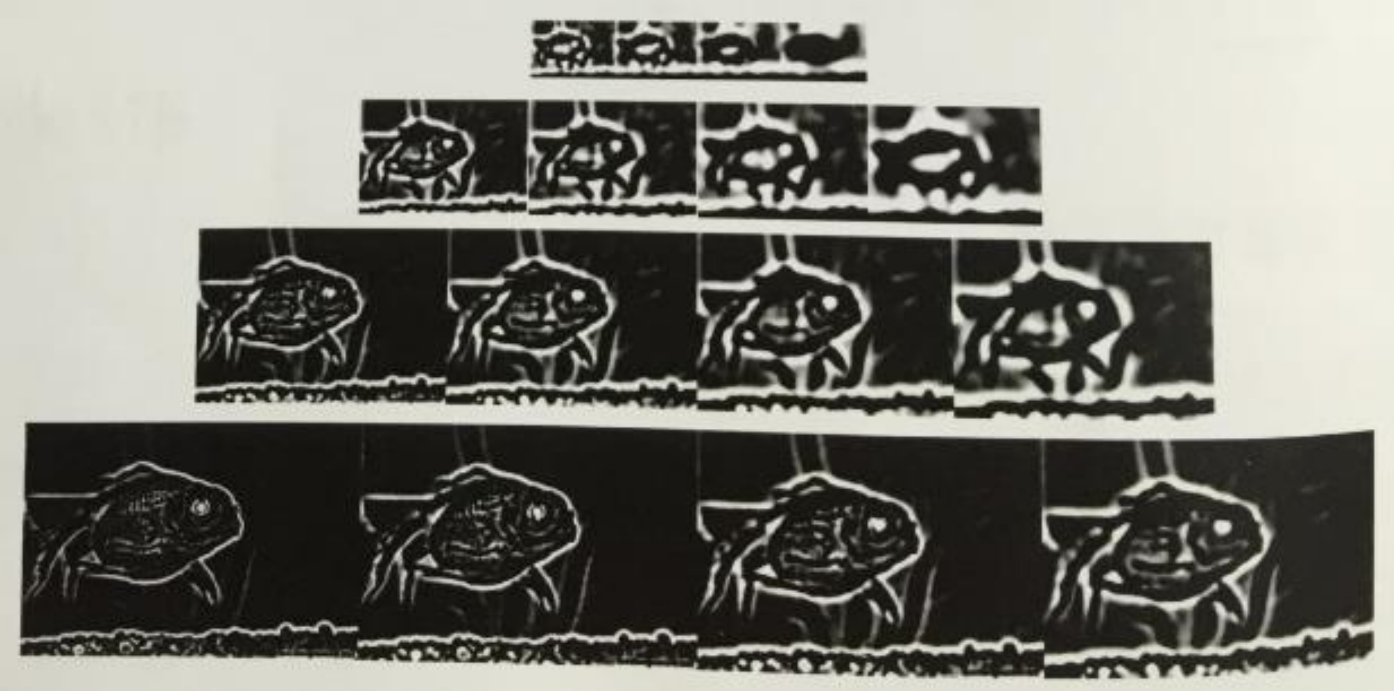
# Example of SIFT

- Local feature points detection

*raw image*



*Gaussian Scale Space* 高斯尺度空间构建

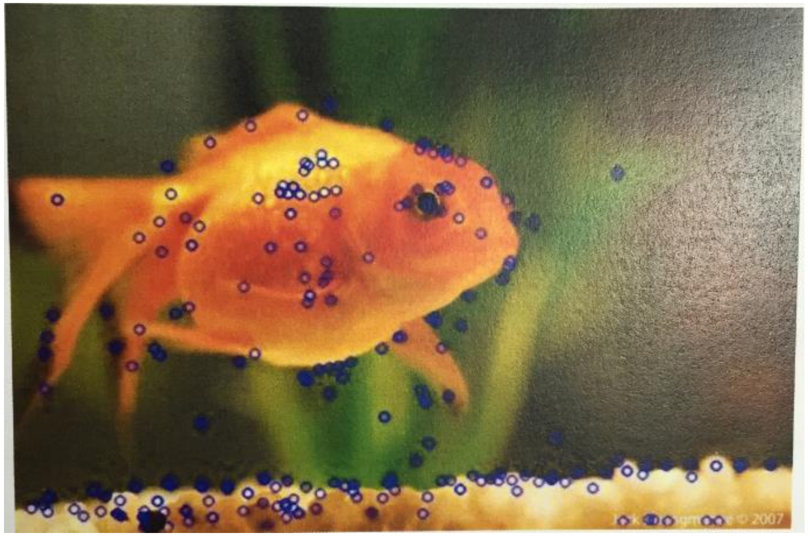


*Difference of Gaussian Scale*高斯差分尺度空间图

# Example of SIFT

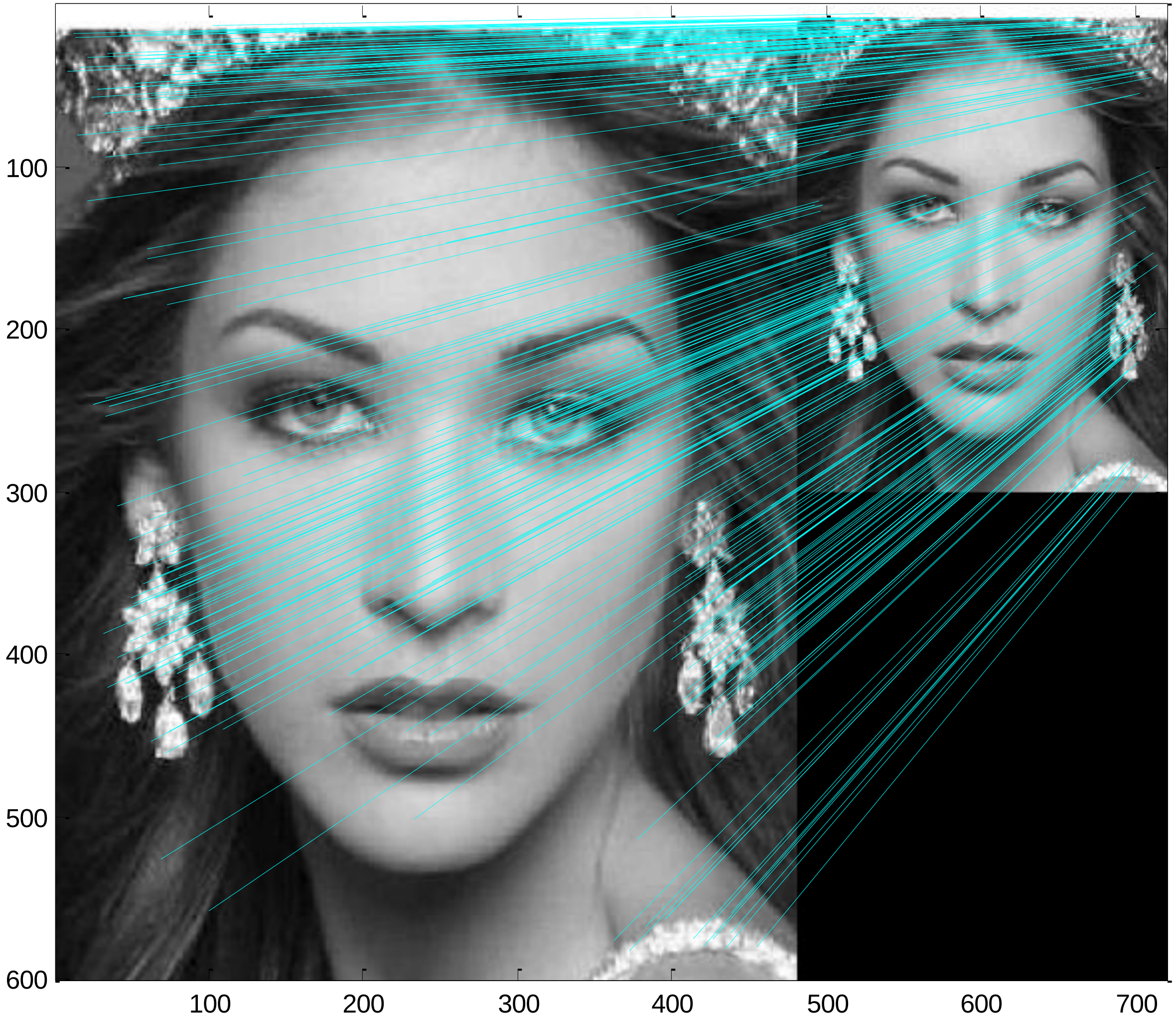
- Local feature points detection

极值检测、关键点精确定位



*Extracted key points based on SIFT*

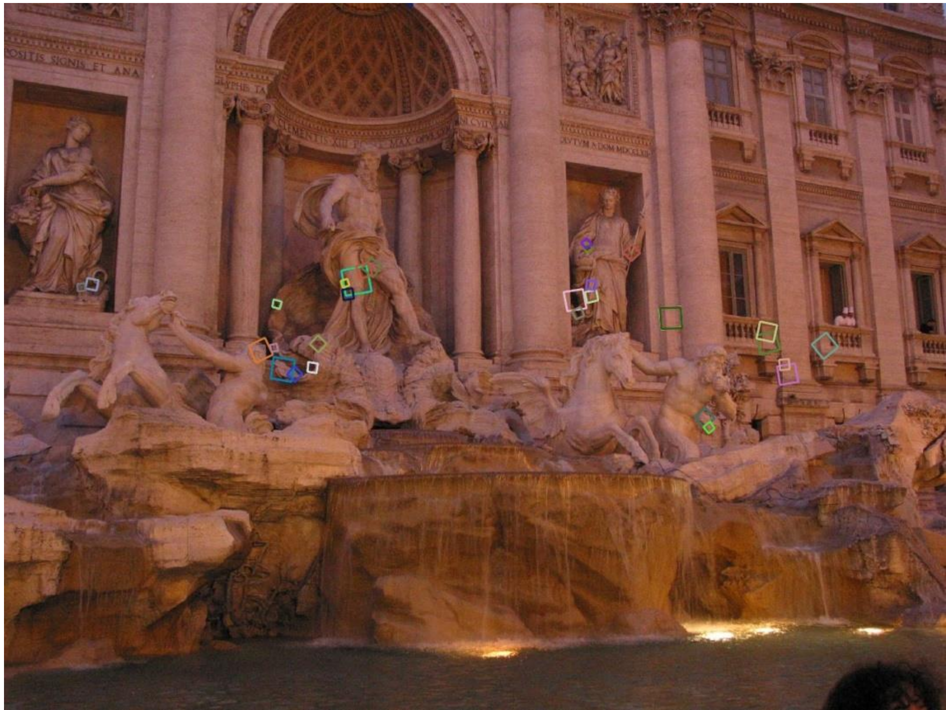
# Feature match: Example of SIFT



# Properties of SIFT

Extraordinarily robust matching technique

- Can handle changes in viewpoint (视角)
  - Up to about 60 degree out of plane rotation
- Can handle significant changes in illumination (光照)
  - Sometimes even day vs. night (below)
- Fast and efficient—can run in real time
- Lots of code available
  - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT



# Bag of Words (BoW) Feature Model

- Common Feature representation

In high-level image understanding, image features are commonly represented as a feature vector in $\mathfrak{R}^D$.

VQ (矢量量化编码) methods, such as clustering, PCA, Hashing.

BoW(词袋模型) is one method based on clustering, proposed in text processing and retrieval.

# Bag of Words (BoW) Feature Model

- Common Feature representation

In 2004, Fei-fei Li proposed a Bag of Visual Words(BoVW) in computer vision, based on BoW.

Princeton Univ, B.A.

California Ins. of Tech. M.S.

California Ins. of Tech. Ph.D, 2005

Stanford Univ. Professor

http://vision.stanford.edu/people.html

# Bag of Words (BoW) Feature Model

- Text Information Retrieval

A text can be viewed as a set with multiple words in some dictionary.

For example, the following two texts

1. Jack wants to play basketball, John wants too.
2. Jack also wants to play football.

The dictionary can be constructed as

Dictionary={1: Jack, 2: want, 3: to, 4: play, 5: basketball, 6: John, 7: too, 8: also, 9: football}

# Bag of Words (BoW) Feature Model

- Text Information Retrieval

Texts:

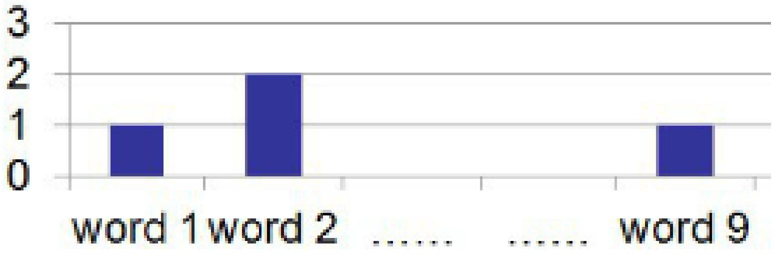
1. Jack wants to play basketball, John wants too.
2. Jack also wants to play football.

The dictionary can be constructed as

Dictionary={1: Jack, 2: want, 3: to, 4: play, 5: basketball, 6: John, 7: too, 8: also, 9: football}

Vector representation (9 dimensions) of the two texts: record the frequency, the words of the dictionary happen in each text. (histogram)

Vector 1=[1, 2, 1, 1, 1, 1, 1, 0, 0]
Vector 2=[1, 1, 1, 1, 0, 0, 0, 1, 1]

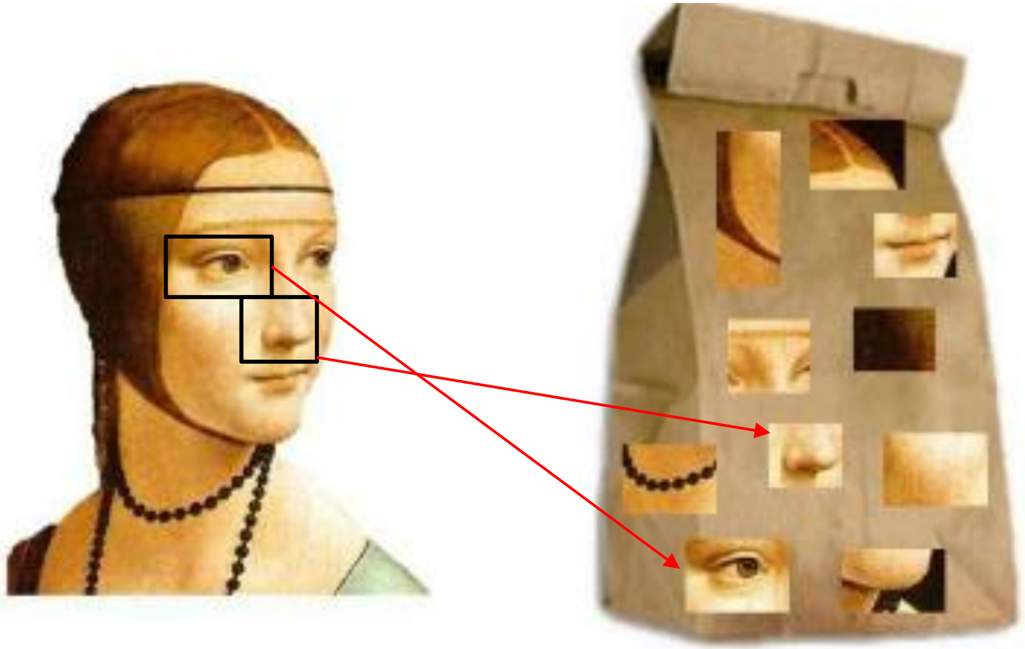


# Bag of Visual Words (BoVW)

- BoVW Model

**Similarly, one image can be viewed as a "text".**

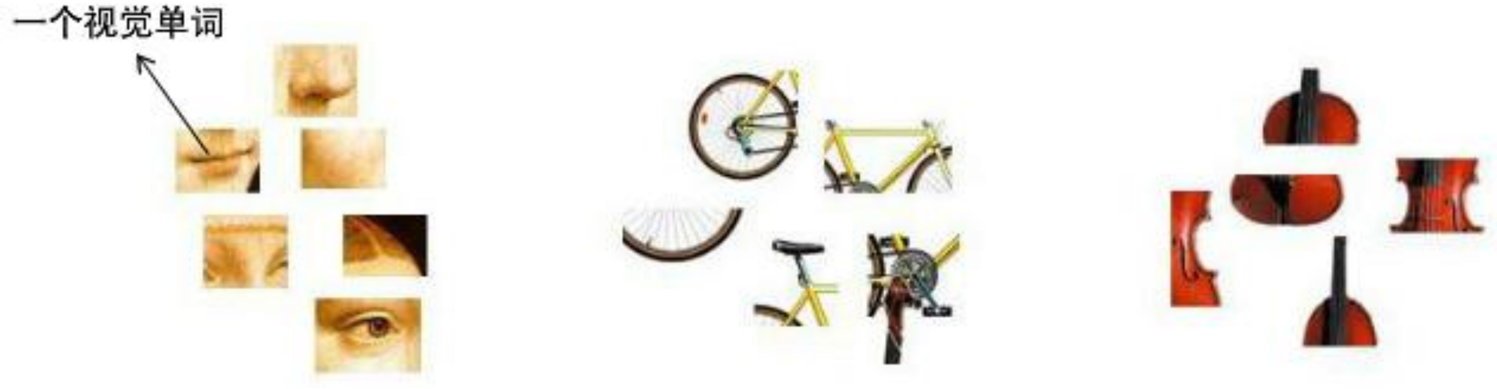
The first step is to construct the visual dictionary, where the visual words are independent.



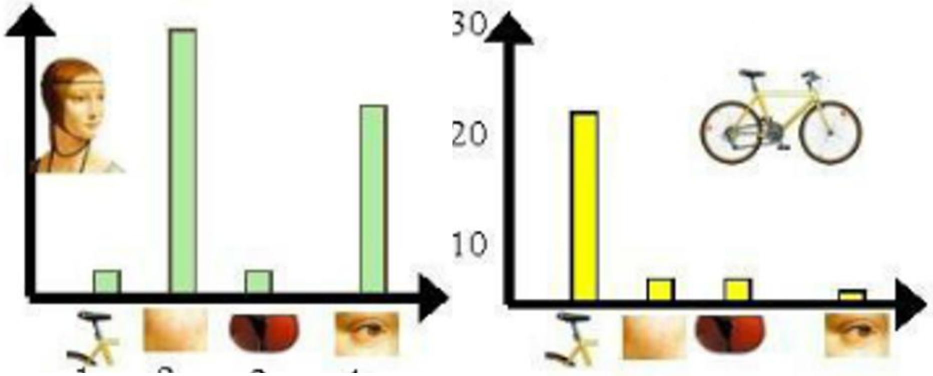
# Bag of Visual Words (BoVW)

- BoVW Model

Differently, the visual words should be manually generated by local feature detection, feature representation (e.g. SIFT local descriptor)



*one image is constituted by multiple words*



# Bag of Visual Words (BoVW)

- BoVW Implementation

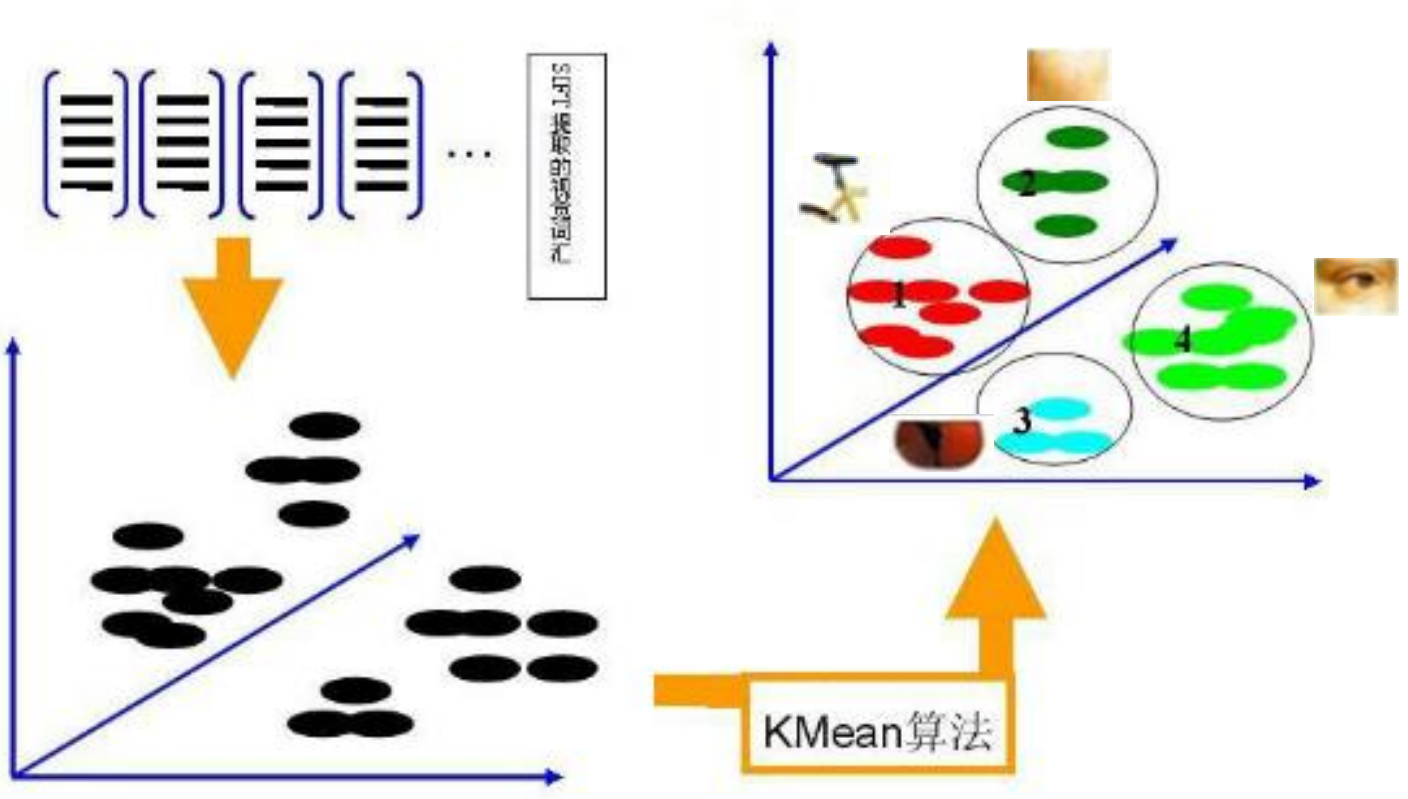
For example, there are 3 training images.

**Step 1**: Based on SIFT, $n_1, n_2, n_3$ feature points with 128-dimensions are detected on the 3 images.

**Step 2**: Based on *K*-means clustering on the SIFT feature points, *K* centers (words) are formulated into a visual dictionary. (*K*=4 for example)

# Bag of Visual Words (BoVW)

- BoVW Implementation



# Bag of Visual Words (BoVW)

- BoVW Implementation

For example, there are 3 training images.

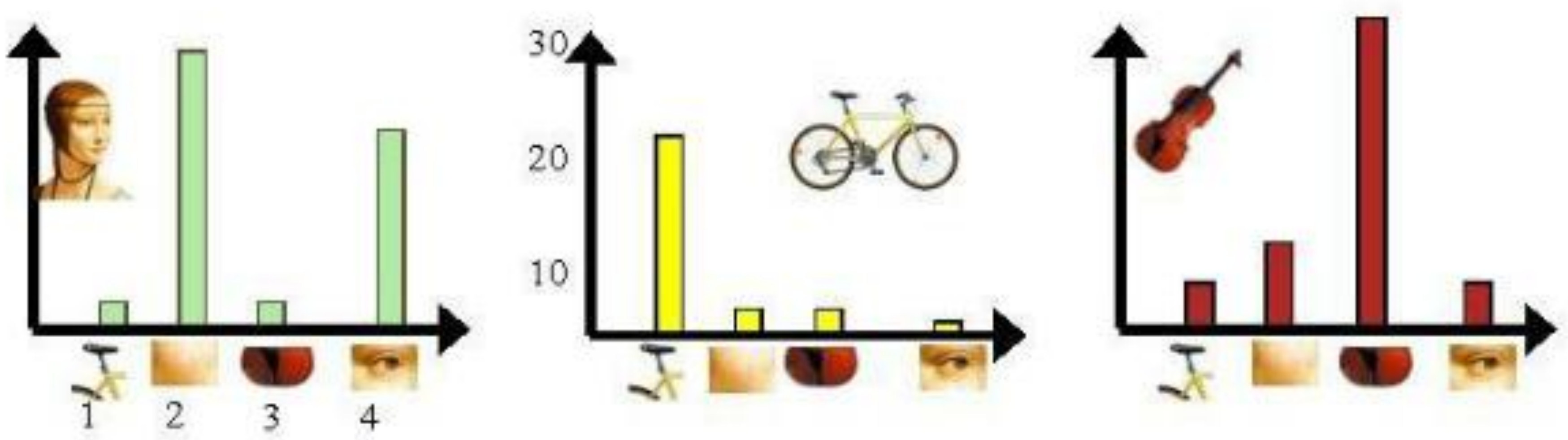
**Step 3**: Image feature vector  based on BoVW.

Similarly to text, compute the frequency of each visual word appeared in each image.

Based on the SIFT feature points of each image, the distance between each points and each word is computed.

# Bag of Visual Words (BoVW)

- BoVW Feature vector



*4 words in the visual dictionary*

The visual word histogram is different from each other

# Bag of Visual Words (BoVW)

- BoVW Testing

For a new image, the feature vector is constructed based on the visual dictionary from the training data.

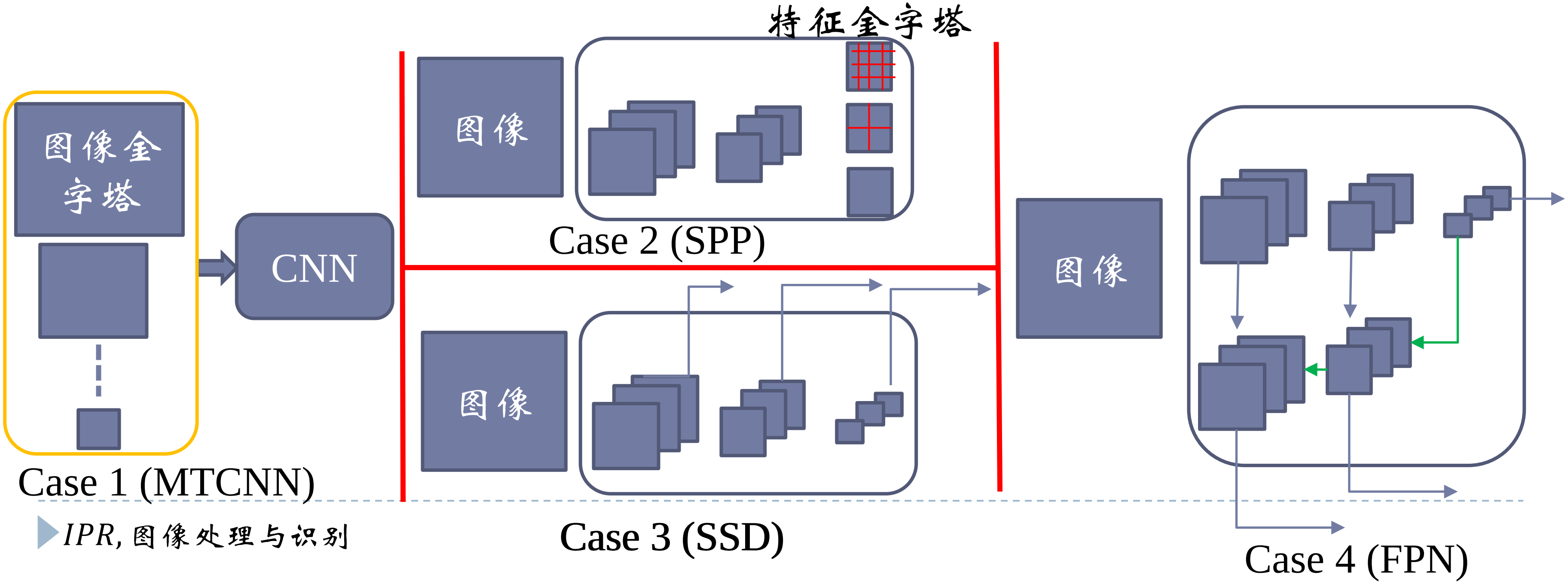
Step 1: extract SIFT feature points: 128-dimensional

Step 2: Compute the frequency that each word happens in the SIFT points

Step 3: Classification

# Multi-scale in Deep Learning

- □ Case 1: Multi-scale in the inputs
- □ Case 2: Multi-scale in the feature map level
- □ Case 3: Multi-scale in the prediction level
- □ Case 4: Multi-scale in both feature map and prediction levels



# 电影中的体现<推理笔记>

把这些数据库关联到一起
then linked all the databases together,

用word2vec获取文本特征
used word2vec to extract textual information

# 电影中的体现

用AGDL提取环境因素

and AGDL to extract environmental factors,

并将这些特征

then they ran this information

# 电影中的体现

代入多个神经网络中进行训练
through a neural net repeatedly to train it,

SEARCHING......
Molecular analysis
A01
A02
A03
A02
用split lbi来代替BP做优化算法
then used split LBI instead of BP as the optimization algorithm

Split Linearized bregman iteration

# 电影中的体现

然后将各种模型融合

and finally integrated the various models

得到最终概率

to get the final probability.

# 第八部分：图像特征描述（II）