

Introduction to Image Representation

Yang Cao

© Chang Wen Chen

University of Science and Technology of China

Outline

- **Fourier and Wavelet Analysis**
- **Covariance Matrix of Gradients**
- **Deep Representation**

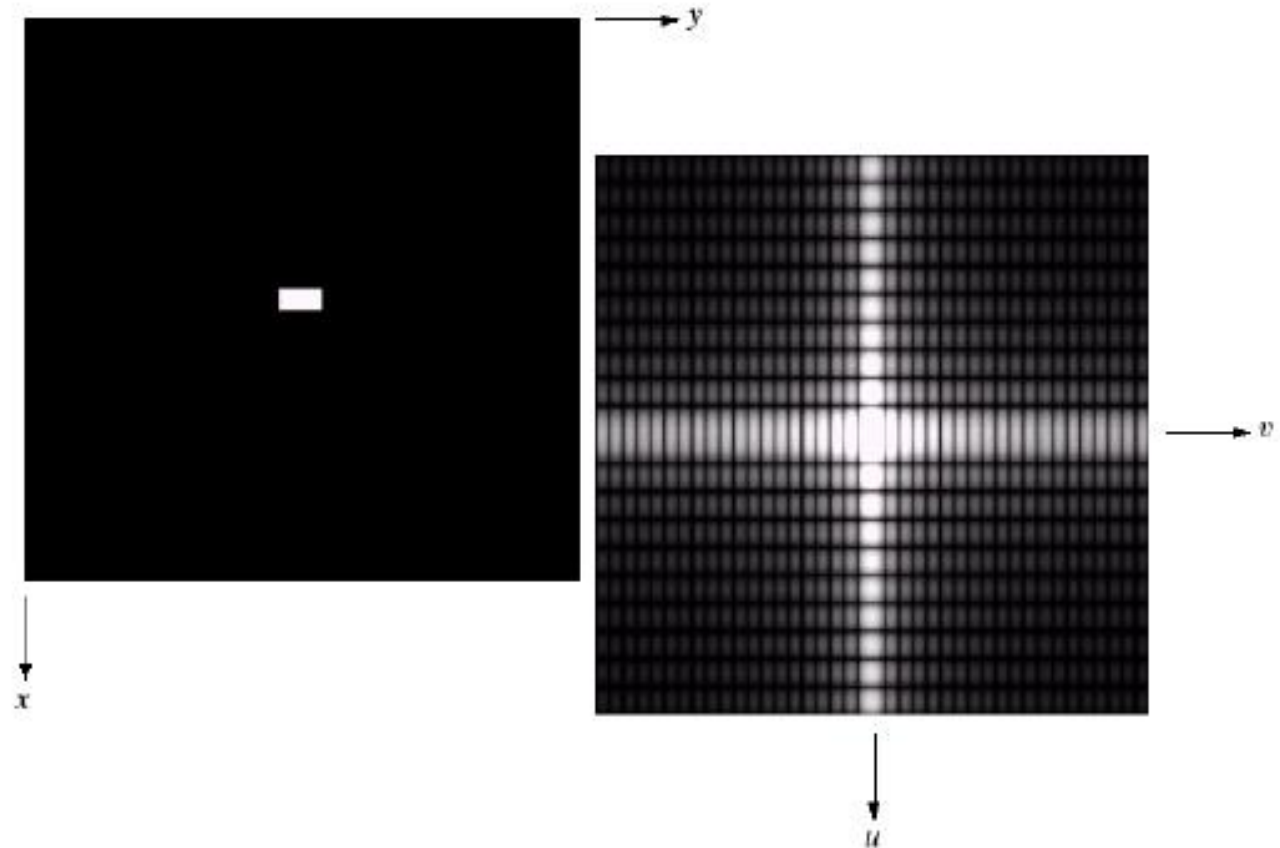
Fourier Analysis

a b

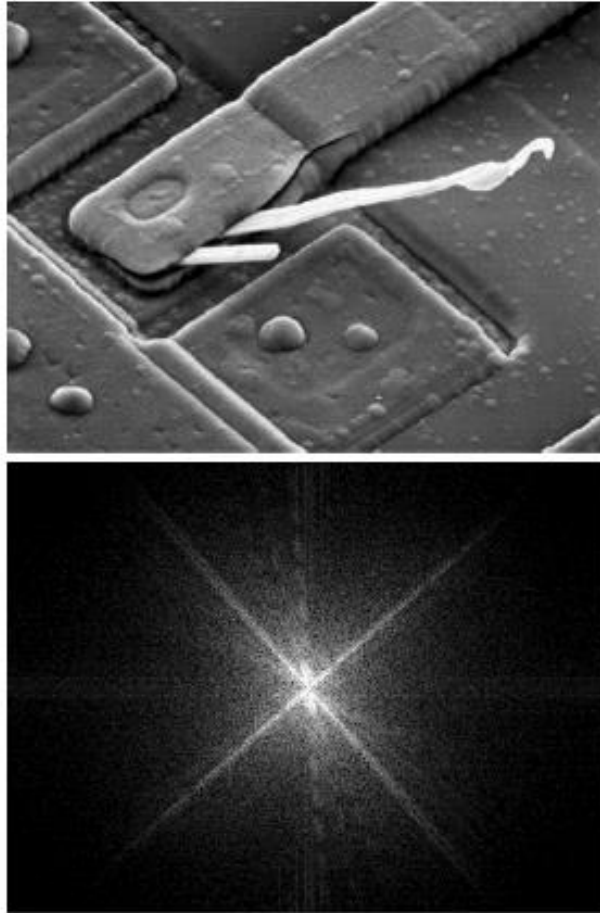
FIGURE 4.3

(a) Image of a 20×40 white rectangle on a black background of size 512×512 pixels

(b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.



Fourier Analysis



a
b

FIGURE 4.4

(a) SEM image of a damaged integrated circuit.

(b) Fourier spectrum of (a).

(Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

Complex Signal from Sum of Sinusoids

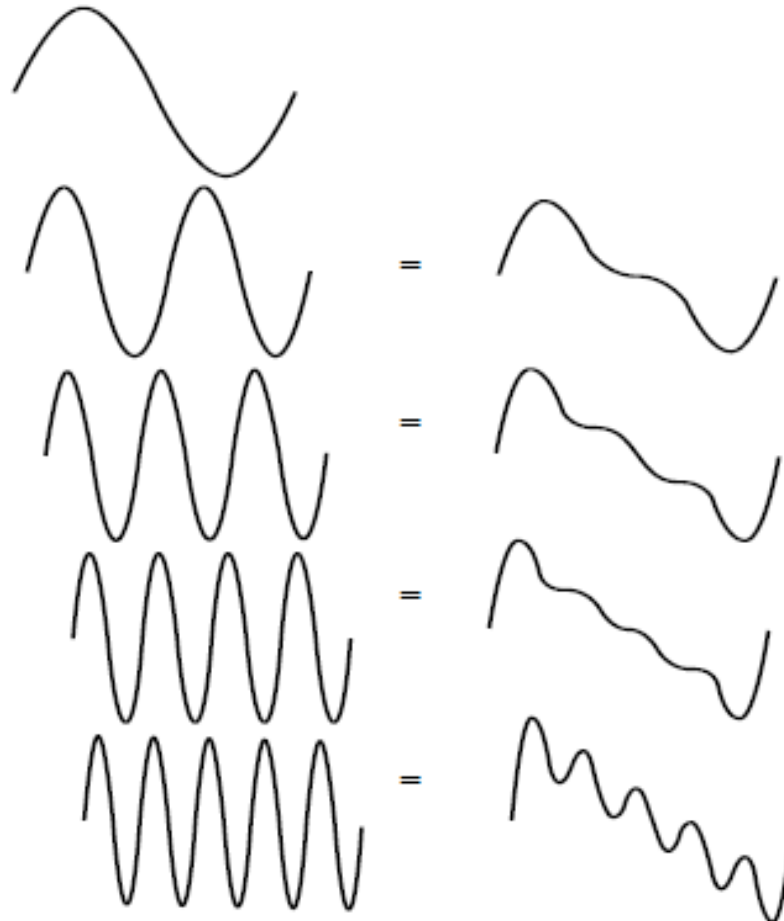
Fundamental
frequency

+ 0.5 \times
2 \times fundamental

+ 0.33 \times
3 \times fundamental

+ 0.25 \times
4 \times fundamental

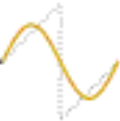
+ 0.5 \times
5 \times fundamental



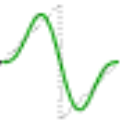
Complex Signal from Sum of Sinusoids



$\frac{2 \sin \theta}{-\pi}$



$\frac{2 \sin 2\theta}{2\pi}$



$\frac{2 \sin 3\theta}{-3\pi}$



$\frac{2 \sin 4\theta}{4\pi}$



Fourier Series Representations

- **Joseph Fourier (1768–1830), a great French mathematician, discovered:**
 - The representation of **periodic signals** with the sum of a set of simple oscillating functions (possibly infinite many) , namely sines and cosines (or **complex exponentials**).
- **Fourier analysis of signals and systems, including applications to image processing and computer vision, has been active research field for many years**
- **Major Variants of Fourier Analysis**
 - **Fourier Series, Fourier Transform (Continuous), Discrete-time Fourier Transform, Discrete Fourier Transform, Time-Frequency Transform**

General Idea of Transforms

- ▶ Suppose that you have an orthonormal (orthogonal, unit length) basis set of vectors $\{\bar{e}_k\}$.
- ▶ Any vector in the space spanned by this basis set can be represented as a weighted sum of those basis vectors:

$$\bar{v} = \sum_k a_k \bar{e}_k$$

- ▶ To get the weights:

$$a_k = \bar{v} \cdot \bar{e}_k$$

- ▶ In other words:
 - ▶ The vector can be *transformed* into the weights a_i .
 - ▶ Likewise, the transformation can be *inverted* by turning the weights back into the vector.

General Idea of Transforms

- ▶ The inner (dot) product of two vectors is the sum of the point-wise multiplication of each component:

$$\bar{u} \cdot \bar{v} = \sum_j \bar{u}[j] \bar{v}[j]$$

- ▶ Can't we do the same thing with functions?

$$f \cdot g = \int_{-\infty}^{\infty} f(x) g(x) dx$$

- ▶ *Functions satisfy all of the linear algebraic requirements of vectors.*

General Idea of Transforms

	Vectors $\{\bar{e}_k\}$	Functions $\{e_k(t)\}$
Transform	$a_k = \bar{v} \cdot \bar{e}_k$ $= \sum_j \bar{v}[j] e_k[j]$	$a_k = f \cdot e_k$ $= \int_{-\infty}^{\infty} f(t) e_k(t) dt$
Inverse	$\bar{v} = \sum_k a_k \bar{e}_k$	$f(t) = \sum_k a_k e_k(t)$

Basis Set: Generalized Harmonics

The set of generalized harmonics we discussed earlier form an orthonormal basis set for functions:

$$\{e^{j2\pi ut}\}$$

where each harmonic has a different frequency u .

Remember:

$$e^{j2\pi ut} = \cos(2\pi ut) + i \sin(2\pi ut)$$

- ▶ The real part is a cosine of frequency u .
- ▶ The imaginary part is a sine of frequency u .

Fourier Series

For a finite set of basis frequencies $\{u_k\}$:

	All Functions $\{e_k(t)\}$	Harmonics $\{e^{j2\pi u_k t}\}$
Transform	$a_k = f \cdot e_k$ $= \int_{-\infty}^{\infty} f(t) e_k(t) dt$	$a_k = f \cdot e^{j2\pi u_k t}$ $= \int_{-\infty}^{\infty} f(t) e^{-j2\pi u_k t} dt$
Inverse	$f(t) = \sum_k a_k e_k(t)$	$f(t) = \sum_k a_k e^{j2\pi u_k t}$

Fourier Series

- The Fourier Series is a signal analysis tool that allows for any **periodic signal** to be decomposed into an infinite sum of everlasting sinusoids

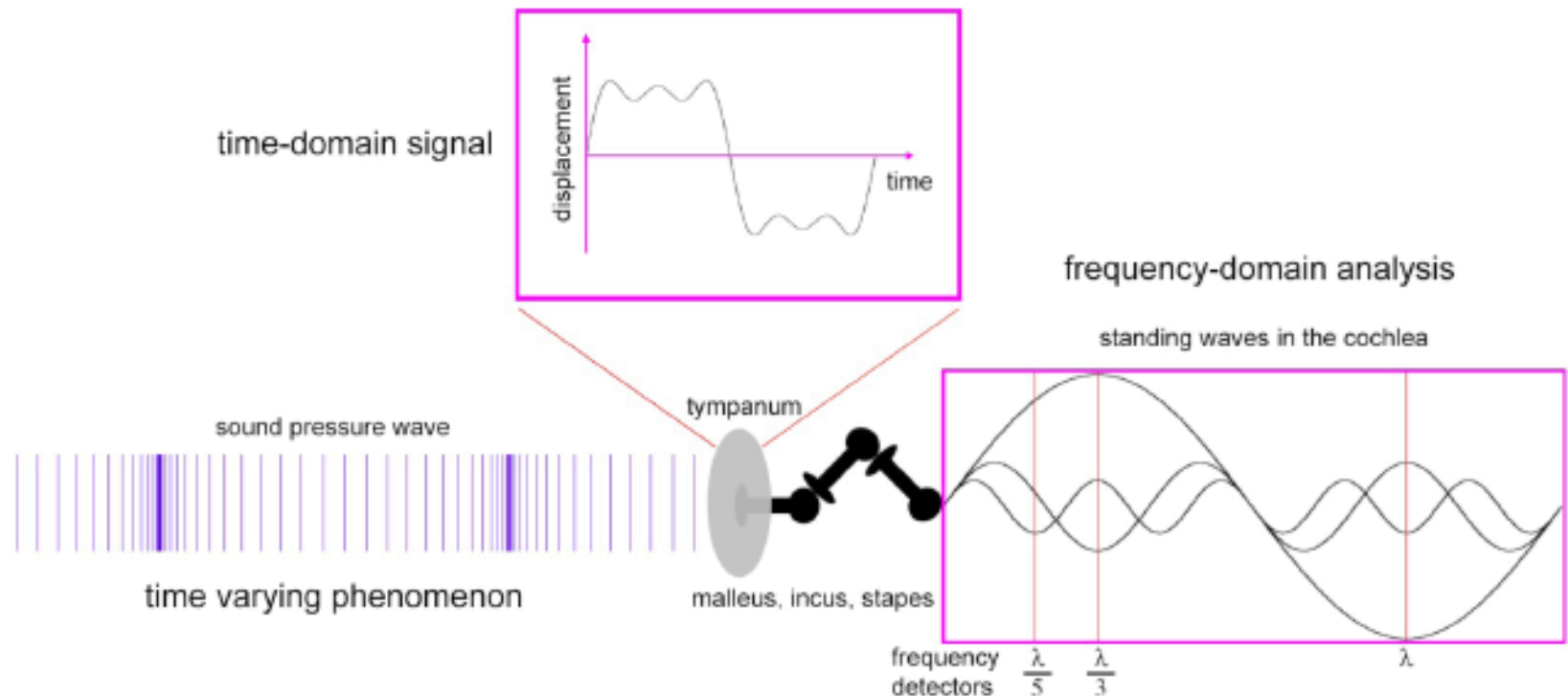
$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{n\pi x}{L}\right) + b_n \sin\left(\frac{n\pi x}{L}\right) \right]$$

- Coefficients can be computed as follows:

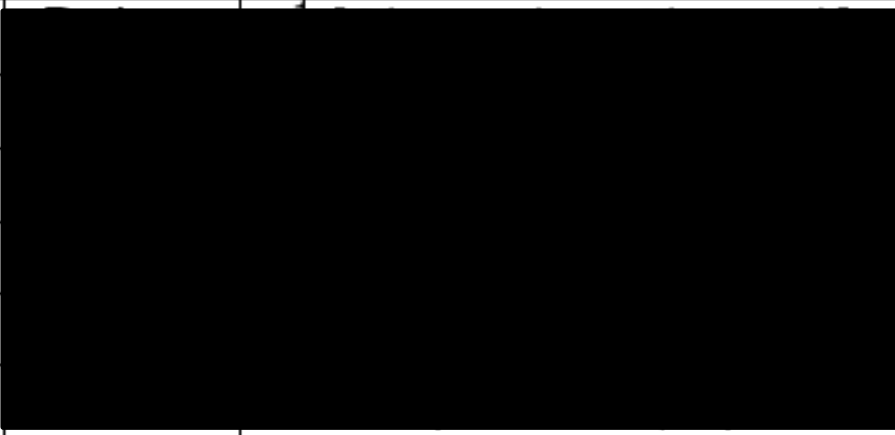
$$a_n = \frac{1}{L} \int_{-L}^L f(x) \cos\left(\frac{n\pi x}{L}\right) dx$$

$$b_n = \frac{1}{L} \int_{-L}^L f(x) \sin\left(\frac{n\pi x}{L}\right) dx$$

Mammals Use the FT in Hearing



Fourier Transform

Spatial Domain $f(t)$		Frequency Domain $F(u)$
Cosine	$\cos(2\pi st)$	
Sine	$\sin(2\pi st)$	
Unit	1	
Constant	a	
Delta	$\delta(t)$	
Comb	$\delta(t \bmod k)$	

Fourier Transform

Spatial Domain $f(t)$		Frequency Domain $F(u)$
Square	1 if $-a/2 \leq t \leq a/2$ 0 otherwise	
Triangle	$1 - t $ if $-a \leq t \leq a$ 0 otherwise	
Gaussian	$e^{-\pi t^2}$	
Differentiation	$\frac{d}{dt}$	

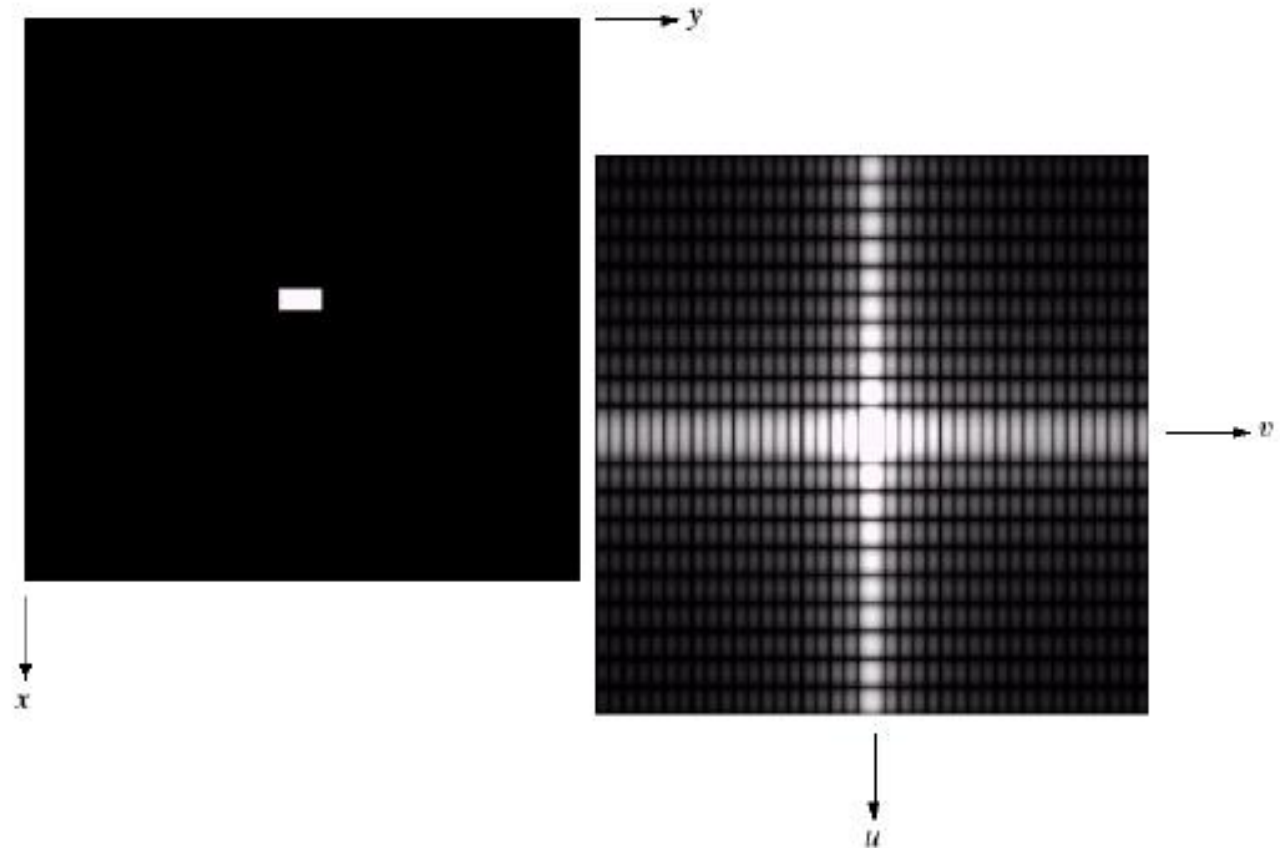
Fourier Analysis

a b

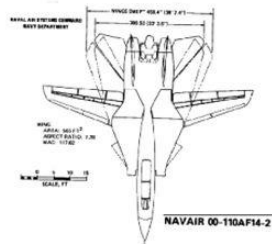
FIGURE 4.3

(a) Image of a 20×40 white rectangle on a black background of size 512×512 pixels

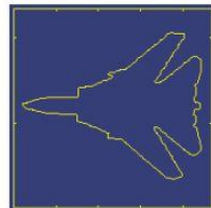
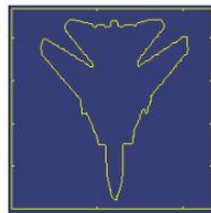
(b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.



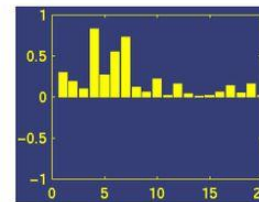
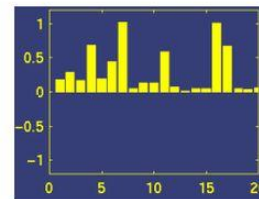
Fourier Descriptors



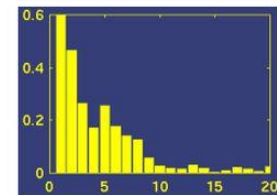
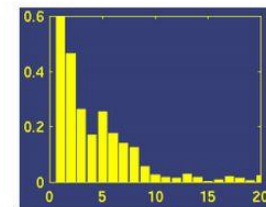
Model/Image
(F14 TOMCAT)



Curve



Fourier
Expansion



Fourier
Descriptors

此图片引自知乎答主“暗中观察”

Fourier Descriptors

Obtain a 1D function from the 2D curve

$$c(t) = x(t) + j y(t)$$



Fourier Descriptors

Perform a Fourier expansion

$$c(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos(k\omega t) + b_k \sin(k\omega t)$$

$$c(t) = x(t) + j y(t)$$

$$x(t) = \frac{a_{x0}}{2} + \sum_{k=1}^{\infty} a_{xk} \cos(k\omega t) + b_{xk} \sin(k\omega t) \quad y(t) = \frac{a_{y0}}{2} + \sum_{k=1}^{\infty} a_{yk} \cos(k\omega t) + b_{yk} \sin(k\omega t)$$

$$a_{xk} = \frac{2}{T} \int_0^T x(t) \cos(k\omega t) dt$$

$$b_{xk} = \frac{2}{T} \int_0^T x(t) \sin(k\omega t) dt$$

$$a_{yk} = \frac{2}{T} \int_0^T y(t) \cos(k\omega t) dt$$

$$b_{yk} = \frac{2}{T} \int_0^T y(t) \sin(k\omega t) dt$$

知乎 @BroGua

Fourier Descriptors

Rewrite to Matrix form

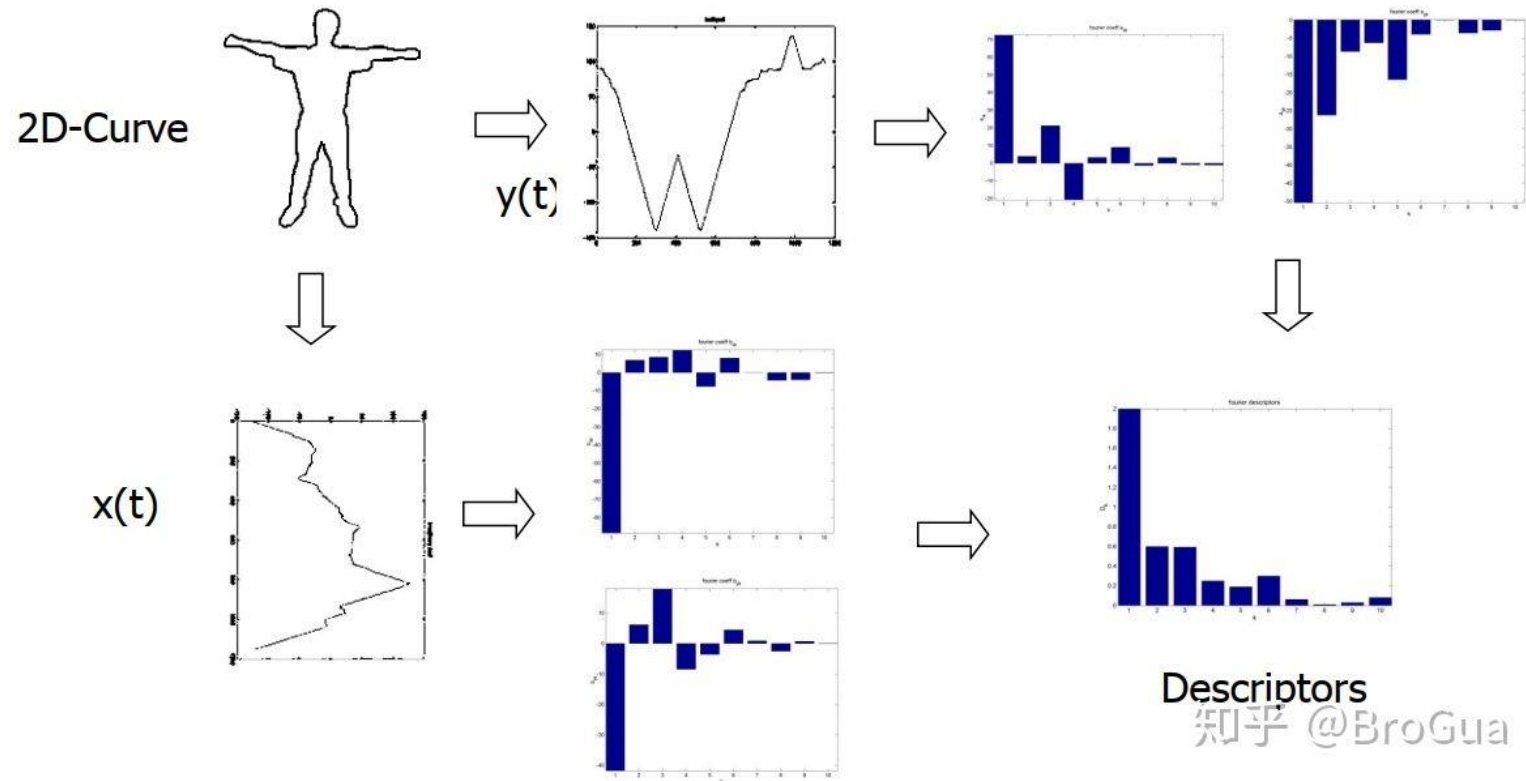
$$\begin{aligned}
 c(t) &= x(t) + j y(t) \\
 x(t) &= \frac{a_{x0}}{2} + \sum_{k=1}^{\infty} a_{xk} \cos(k\omega t) + b_{xk} \sin(k\omega t) \\
 y(t) &= \frac{a_{y0}}{2} + \sum_{k=1}^{\infty} a_{yk} \cos(k\omega t) + b_{yk} \sin(k\omega t) \\
 c(t) &= \frac{a_{x0}}{2} + \sum_{k=1}^{\infty} \left(a_{xk} \cos(k\omega t) + b_{xk} \sin(k\omega t) + j \left(\frac{a_{y0}}{2} + \sum_{k=1}^{\infty} a_{yk} \cos(k\omega t) + b_{yk} \sin(k\omega t) \right) \right)
 \end{aligned}$$

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} a_{x0} \\ a_{y0} \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_{xk} & b_{xk} \\ a_{yk} & b_{yk} \end{bmatrix} \begin{bmatrix} \cos(k\omega t) \\ \sin(k\omega t) \end{bmatrix}$$

知乎 @BroGua

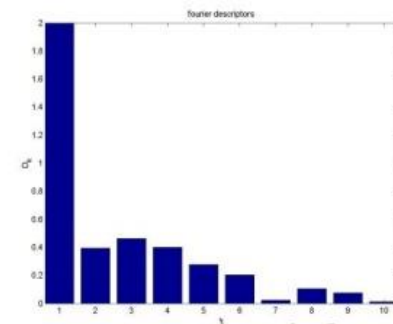
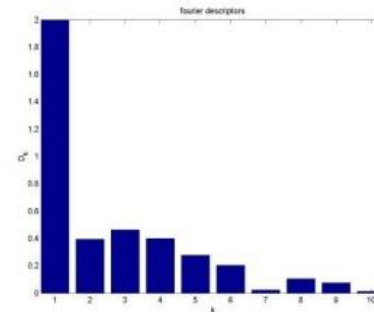
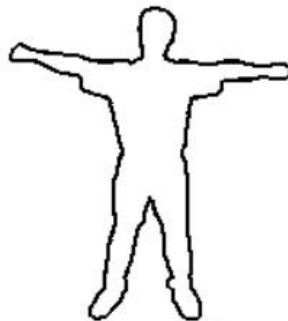
Fourier Descriptors

Incorporate Four coefficient vectors into Fourier descriptors



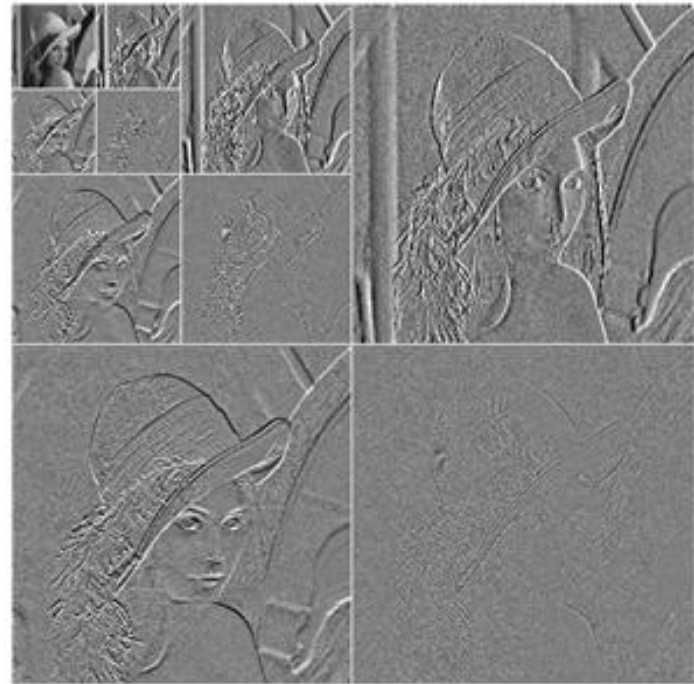
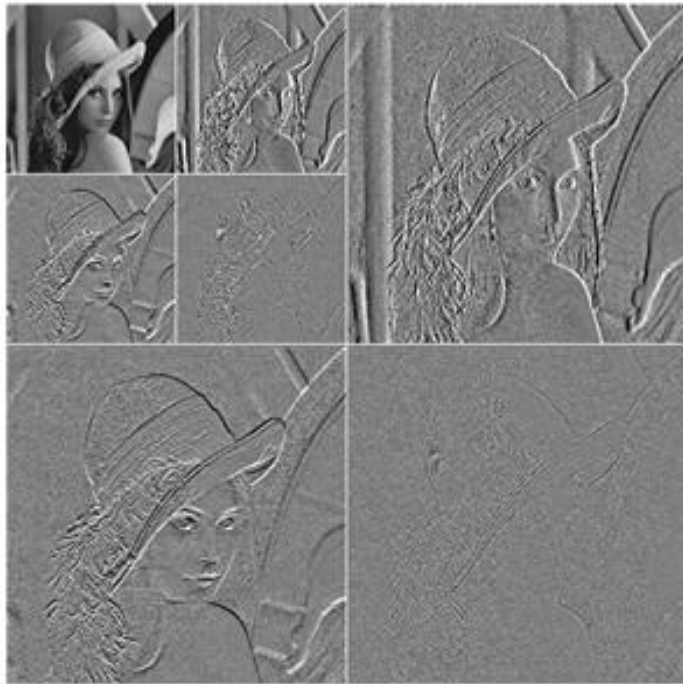
Fourier Descriptors

Invariant to scaling, translation and rotation

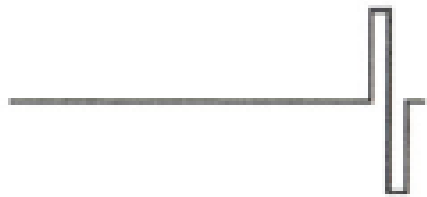


知乎 @BroGua

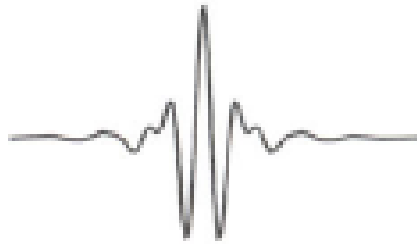
Wavelet Analysis



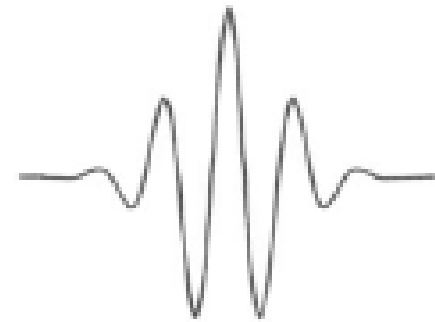
Wavelet Analysis



(a) Haar



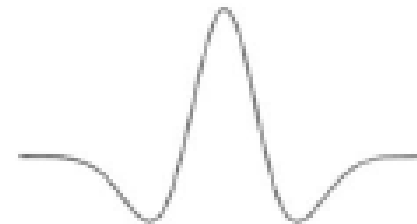
(b) Meyer



(c) Morlet



(d) Daubechies-4



(e) Mexican hat

Wavelet Transform

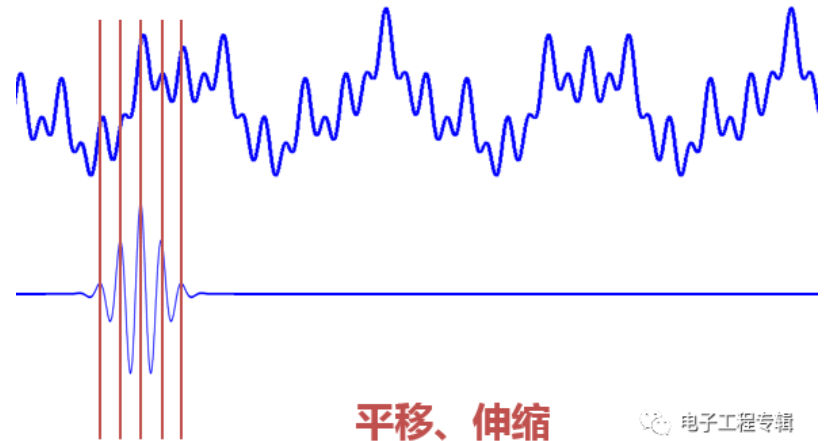
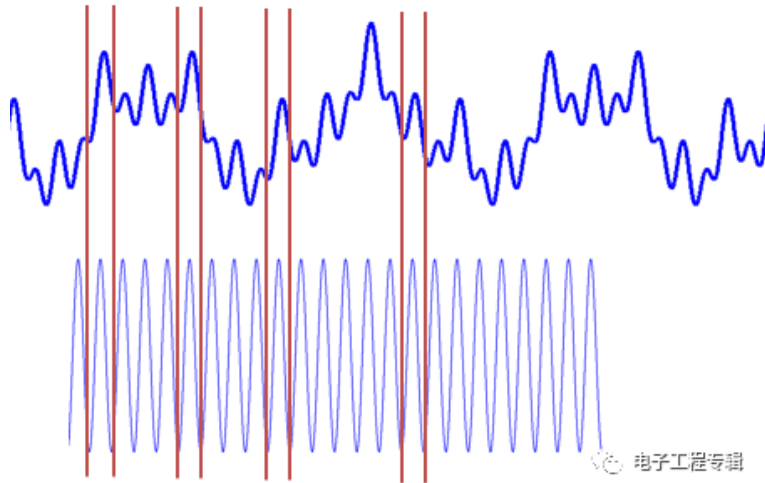
$$F(\omega) = \int_{-\infty}^{\infty} f(t) * e^{-i\omega t} dt$$

傅里叶变换只有一个变量 ω ， 对应频率上的变化。

$$WT(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) * \psi\left(\frac{t - \tau}{a}\right) dt$$

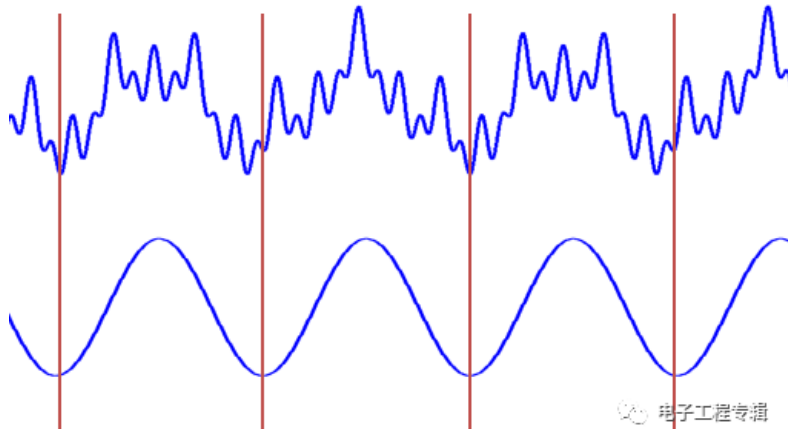
小波变换有两个变量，尺度 a 和平移 τ ，分别控制小波的伸缩和平移。尺度对应频率上的变化，平移对应时间上的变化。

Wavelet Transform



平移、伸缩

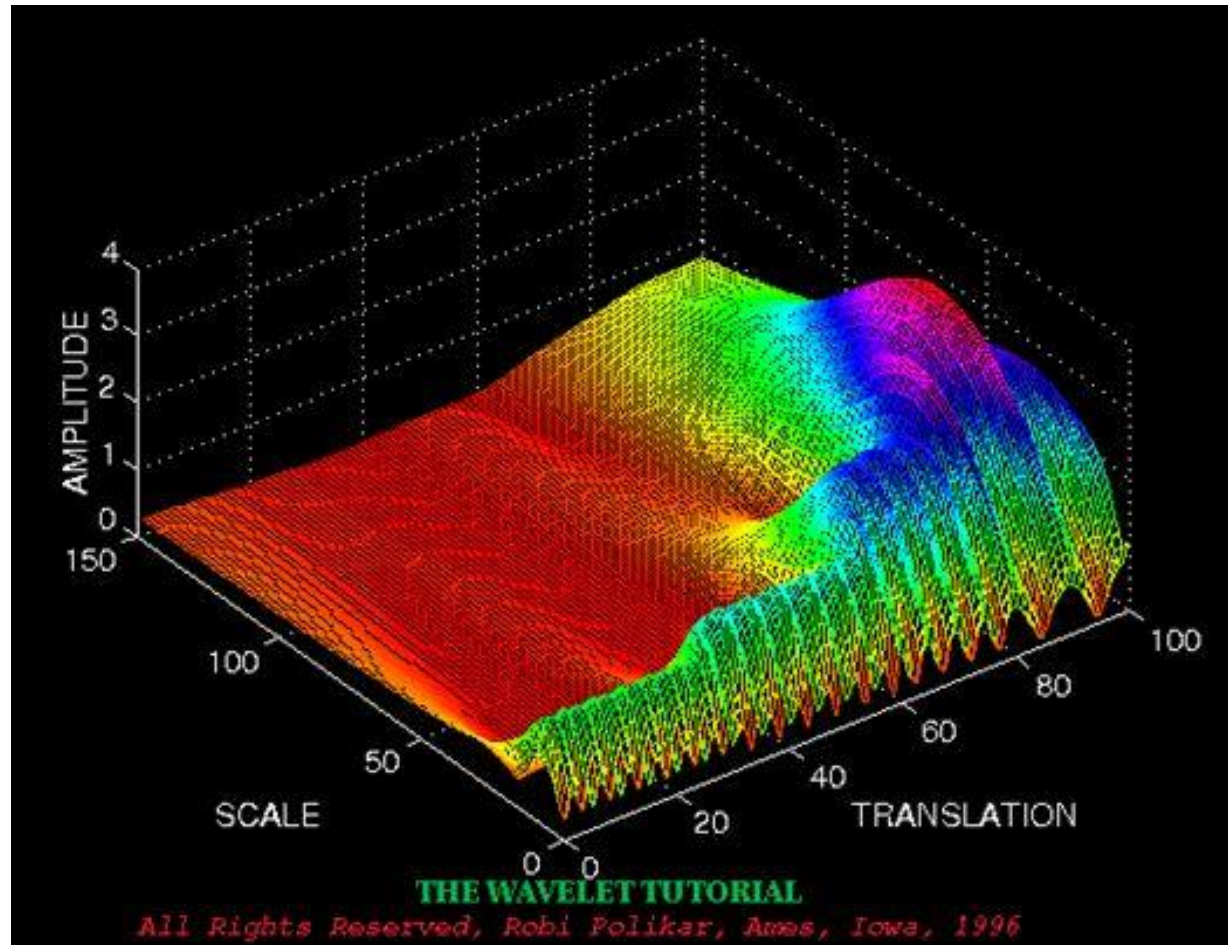
电子工程专辑



电子工程专辑

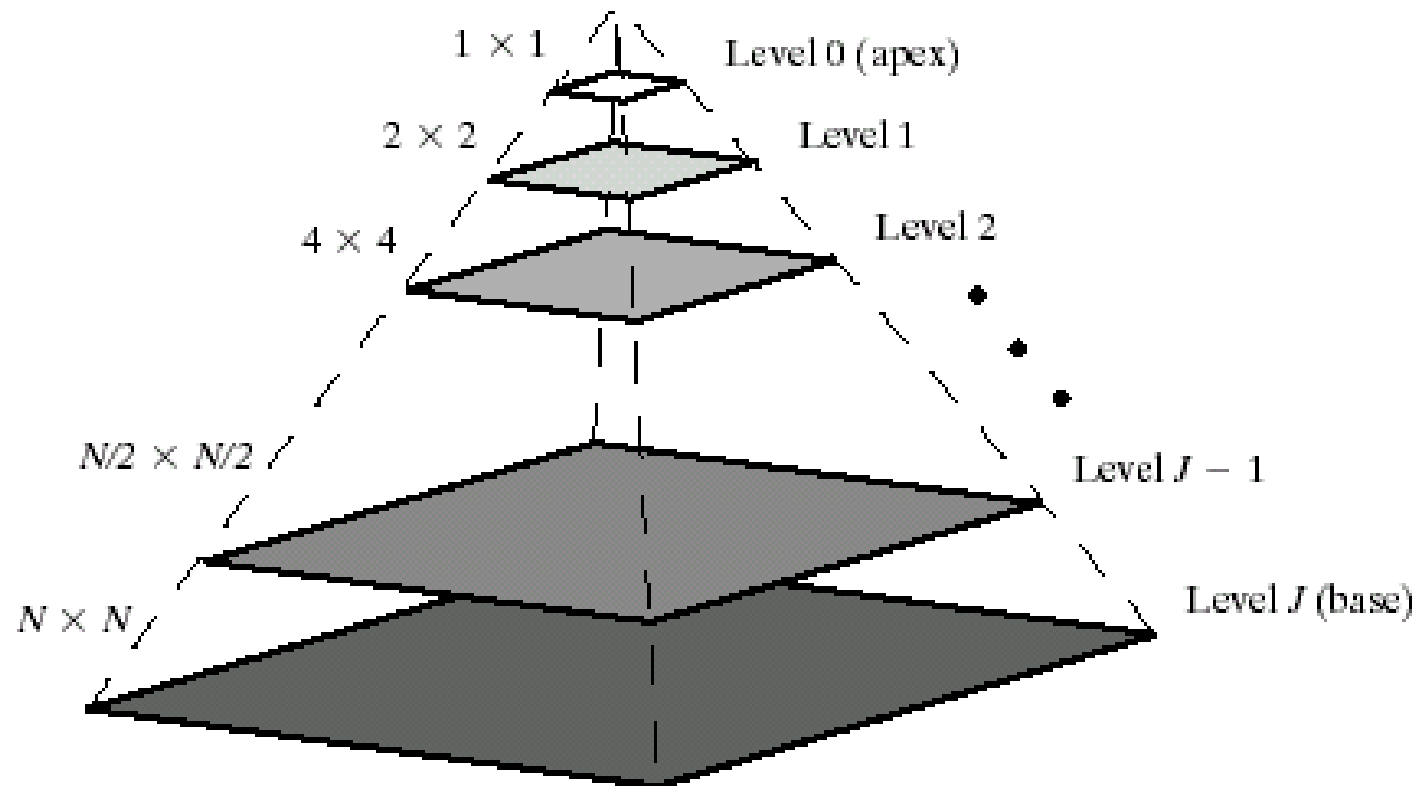
Wavelet Transform

此图片引自
“小波导论”

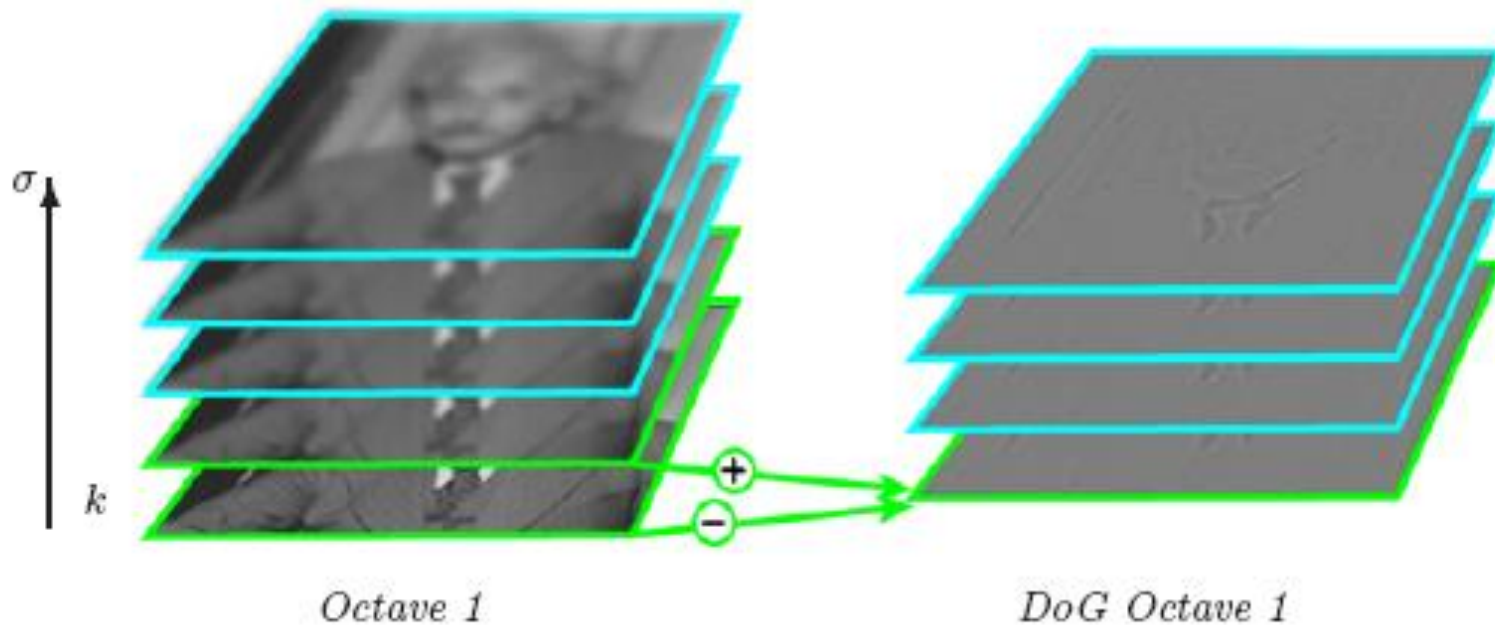


小波变换不仅可以得到频域谱，而且可以得到时域谱。

Image Pyramid

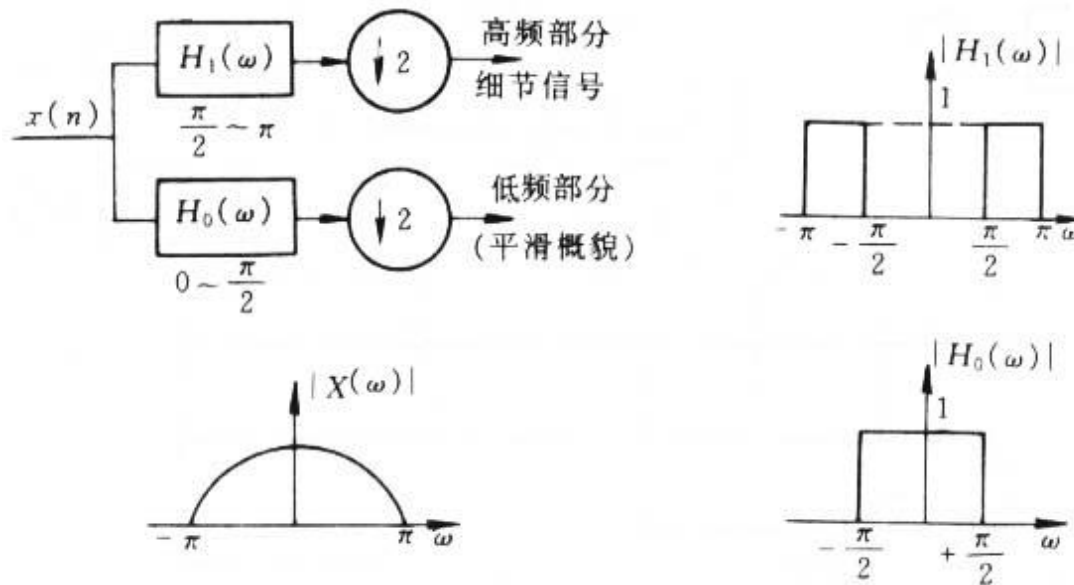


Scale Space



Wavelet Analysis

多分辨分析是小波分析中最重要的概念之一，它将一个函数表示为一个低频成分与不同分辨率下的高频成分，并且多分辨分析能提供一种构造小波的统一框架，提供函数分解与重构的快速算法。

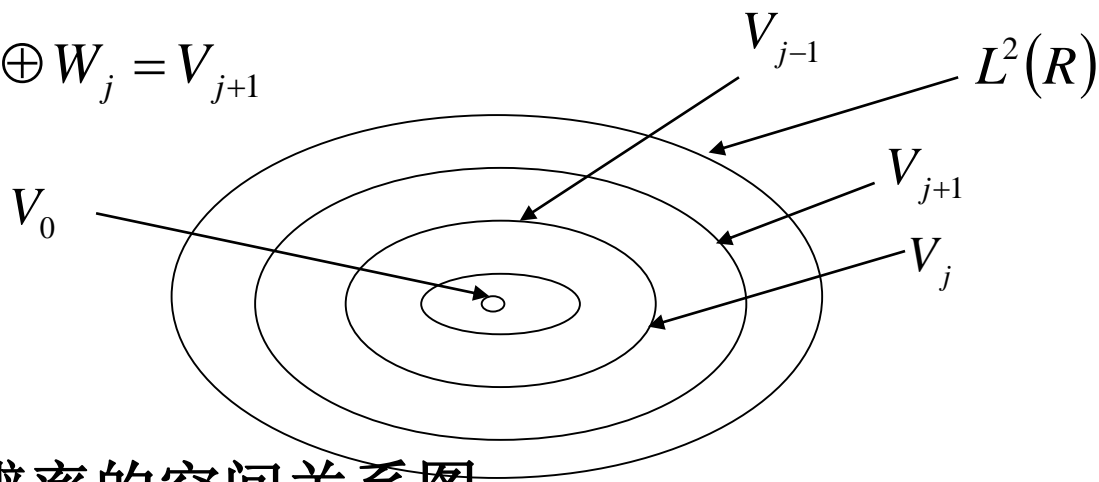


$L^2(R)$ 是一个无限维向量空间，称为平方可积空间，将 $L^2(R)$ 用它的子空间 $\{V_j\}_{j \in \mathbb{Z}}$ ， $\{W_j\}_{j \in \mathbb{Z}}$ 表示，其中 $\{V_j\}_{j \in \mathbb{Z}}$ 称为尺度空间， $\{W_j\}_{j \in \mathbb{Z}}$ 称为小波空间。

尺度空间的递归嵌套关系： $\{0\} \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \subset L^2(R)$

小波空间 W_j 是 V_j 和 V_{j+1} 之间的差，即 $V_j \oplus W_j = V_{j+1}$ ，它捕捉由 V_j 逼近 V_{j+1} 时丢失的信息。推出：

$$V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_j = V_{j+1}$$



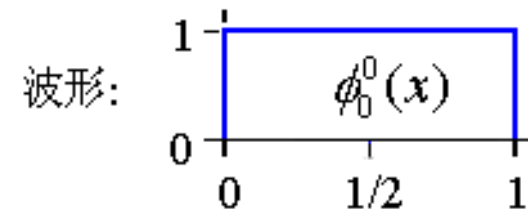
多分辨率的空间关系图

Harr Wavelet

- 哈尔基函数

- 基函数是一组线性无关的函数，可以用来构造任意给定的信号，如用基函数的加权和表示
- 哈尔基函数(Haar basis function)
 - » 定义在半开区间 $[0, 1)$ 上的一组分段常值函数(piecewise-constant function)集
 - » 生成矢量空间 V^0 的常值函数

$$V^0 : \phi_0^0(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$

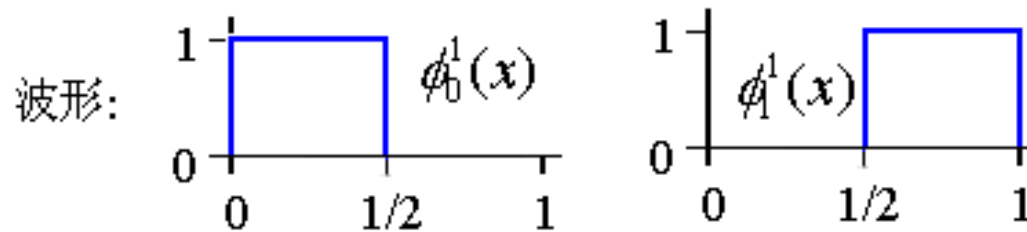


Harr Wavelet

» 生成矢量空间 V^1 的常值函数

$$V^1: \phi_0^1(x) = \begin{cases} 1 & 0 \leq x < 0.5 \\ 0 & \text{其他} \end{cases},$$

$$\phi_1^1(x) = \begin{cases} 1 & 0.5 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$



Harr Wavelet

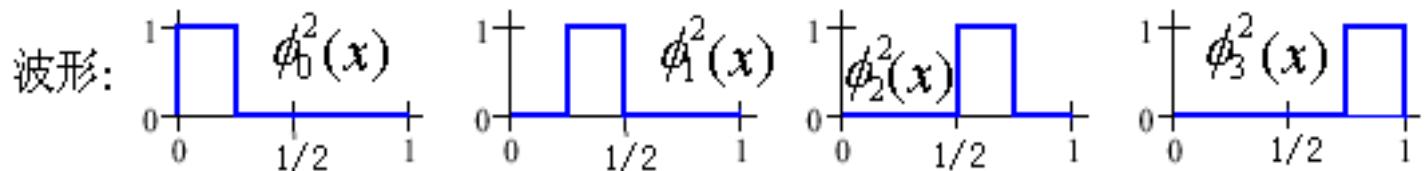
» 生成矢量空间 V^2 的常值函数

$$\phi_0^2(x) = \begin{cases} 1, & 0 \leq x < 1/4 \\ 0, & \text{其他} \end{cases}$$

$$\phi_1^2(x) = \begin{cases} 1, & 1/4 \leq x < 1/2 \\ 0, & \text{其他} \end{cases}$$

$$\phi_2^2(x) = \begin{cases} 1, & 1/2 \leq x < 3/4 \\ 0, & \text{其他} \end{cases}$$

$$\phi_3^2(x) = \begin{cases} 1, & 3/4 \leq x < 1 \\ 0, & \text{其他} \end{cases}$$



- 可按照以上方法继续定义哈尔基函数和由它生成的矢量空间 V_j, \dots

Harr Wavelet

- » 为了表示矢量空间中的矢量，每一个矢量空间都需要定义一个基(basis)，哈尔基定义为

$$\phi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$

- 为生成矢量空间而定义的基函数也叫做尺度函数(scaling function)。哈尔基尺度函数定义为

$$\phi_i^j(x) = \phi(2^j x - i), \quad i = 0, 1, \dots, (2^j - 1)$$

- 其中， j 为尺度因子，使函数图形缩小或放大
 i 为平移参数，使函数沿 x 轴方向平移

Harr Wavelet

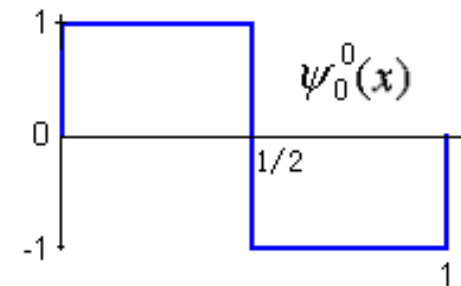
- 哈尔小波(函数)

- 最古老和最简单的小波, 定义为

$$\psi(x) = \begin{cases} 1 & \text{当 } 0 \leq x < 1/2 \\ -1 & \text{当 } 1/2 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$

- 生成矢量空间 W^0 的哈尔小波

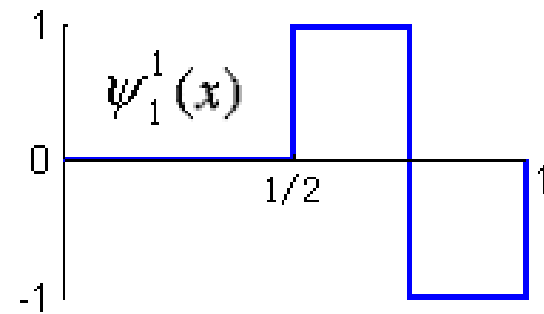
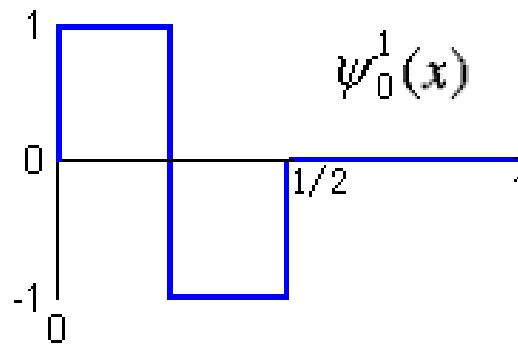
$$\psi_0^0(x) = \begin{cases} 1 & 0 \leq x < 1/2 \\ -1 & 1/2 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$



Harr Wavelet

– 生成矢量空间 W 的哈尔小波

$$\psi_0^1(x) = \begin{cases} 1 & 0 \leq x < 1/4 \\ -1 & 1/4 \leq x < 1/2 \\ 0 & \text{其他} \end{cases} \quad \psi_1^1(x) = \begin{cases} 1 & 1/2 \leq x < 3/4 \\ -1 & 3/4 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$



Harr Wavelet

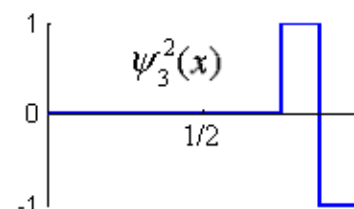
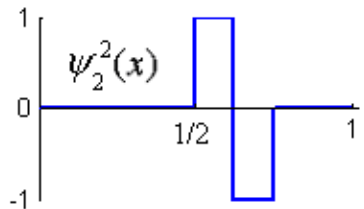
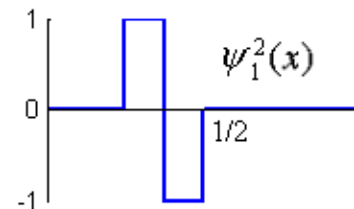
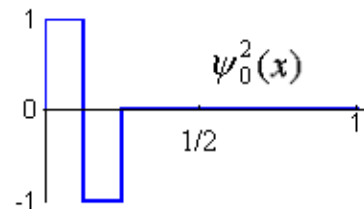
– 生成矢量空间 W^2 的哈尔小波

$$\psi_0^2(x) = \begin{cases} 1 & 0 \leq x < 1/8 \\ -1 & 1/8 \leq x < 2/8 \\ 0 & \text{其他} \end{cases}$$

$$\psi_1^2(x) = \begin{cases} 1 & 2/8 \leq x < 3/8 \\ -1 & 3/8 \leq x < 4/8 \\ 0 & \text{其他} \end{cases}$$

$$\psi_2^2(x) = \begin{cases} 1 & 4/8 \leq x < 5/8 \\ -1 & 5/8 \leq x < 6/8 \\ 0 & \text{其他} \end{cases}$$

$$\psi_3^2(x) = \begin{cases} 1 & 6/8 \leq x < 7/8 \\ -1 & 7/8 \leq x < 1 \\ 0 & \text{其他} \end{cases}$$



Harr Wavelet

- 求有限信号的均值和差值

- 假设有一幅分辨率只有4个像素 P_0 、 P_1 、 P_2 、 P_3 的一维图像，对应的像素值分别为

[9 7 3 5]

计算该图像的哈尔小波变换系数

- » 步骤1：求均值(averaging)。计算相邻像素对的平均值，得到一幅分辨率比较低的新图像，它的像素数目变成了2个，即新的图像的分辨率是原来的1/2，相应的像素值为

[8 4]

Harr Wavelet

» 步骤2: 求差值(differencing)。为能从2个像素组成的图像重构由4个像素组成的原始图像, 就需要存储一些图像的细节系数(detail coefficient)

- 方法是把像素对的第一个像素值减去这个像素对的平均值, 或者使用这个像素对的差值除以2

原始图像用两个均值和两个细节系数表示为

$$[8 \ 4 \ 1 \ -1]$$

» 步骤3: 重复步骤1和2, 把由第一步分解得到的图像进一步分解成分辨率更低的图像和细节系数。其结果, 整幅图像表示为

$$[6 \ 2 \ 1 \ -1]$$

Harr Wavelet

表1 哈尔变换过程

分辨率	平均值	细节系数
4	[9 7 3 5]	
2	[8 4]	[1 -1]
1	[6]	[2]

- 把由4个像素组成的一幅图像用一个平均像素值和三个细节系数表示，这个过程称为哈尔小波变换(Haar wavelet transform)，也称哈尔小波分解(Haar wavelet decomposition)。这个概念可以推广到使用其他小波基的变换
- 特点：(1) 变换过程中没有丢失信息，因为能够从所记录的数据中重构出原始图像。(2) 对这个给定的变换，可从所记录的数据中重构出各种分辨率的图像。(3) 通过变换之后产生的细节系数的幅度值比较小，为图像压缩提供了一种途径，如去掉微不足道的系数

Harr Wavelet

- 哈尔小波变换

- 在例1中的求均值和差值的过程实际上就是一维小波变换的过程，现在用数学方法重新描述哈尔小波变换

- » $I(x)$ 图像用 V^2 中的哈尔基表示

$$I(x) = 9\phi_0^2(x) + 7\phi_1^2(x) + 3\phi_2^2(x) + 5\phi_3^2(x)$$

$$\begin{aligned} I(x) = & 9 \times \begin{array}{c} \text{[Step Function]} \end{array} \phi_0^2(x) \\ & + 7 \times \begin{array}{c} \text{[Step Function]} \end{array} \phi_1^2(x) \\ & + 3 \times \begin{array}{c} \text{[Step Function]} \end{array} \phi_2^2(x) \\ & + 5 \times \begin{array}{c} \text{[Step Function]} \end{array} \phi_3^2(x) \end{aligned}$$

Harr Wavelet

» $I(x)$ 图像用 V 和 W 中的函数表示

生成 V 矢量空间的基函数为 $\phi_0^1(x)$ 和 $\phi_1^1(x)$ ，生成矢量空间 W 的小波函数为 $\psi_0^1(x)$ 和 $\psi_1^1(x)$ ， $I(x)$ 可表示为

$$I(x) = c_0^1 \phi_0^1(x) + c_1^1 \phi_1^1(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x)$$

$$\begin{aligned}
 I(x) = & 8 \times \text{[Step Function]} \phi_0^1(x) \\
 & + 4 \times \text{[Step Function]} \phi_1^1(x) \\
 & + 1 \times \text{[Step Function]} \psi_0^1(x) \\
 & + -1 \times \text{[Step Function]} \psi_1^1(x)
 \end{aligned}$$

Harr Wavelet

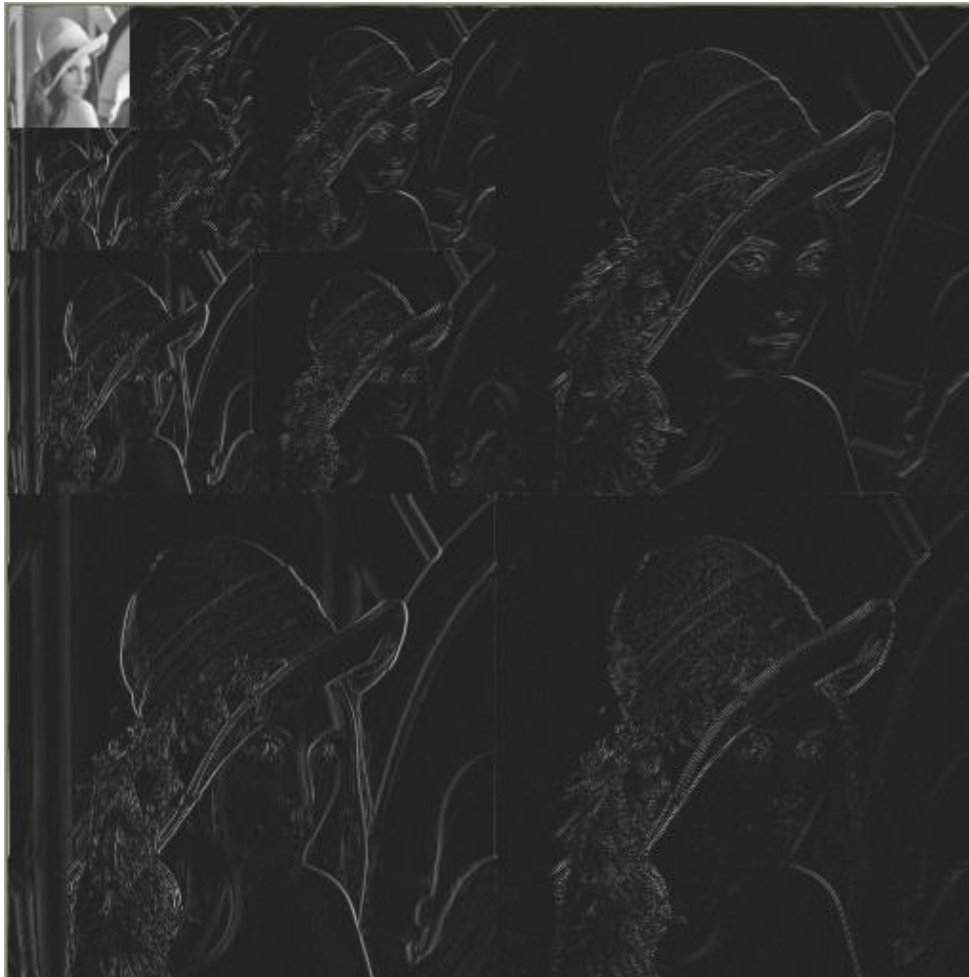
» $I(x)$ 图像用 V^0 、 W^0 和 W^1 中的函数表示

生成矢量空间 V^0 的基函数为 $\phi_0^0(x)$ ，生成矢量空间 W^0 的小波函数为 $\psi_0^0(x)$ ，
生成矢量空间 W^1 的小波函数为 $\psi_0^1(x)$ 和 $\psi_1^1(x)$ ， $I(x)$ 可表示为

$$I(x) = c_0^0 \phi_0^0(x) + d_0^0 \psi_0^0(x) + d_0^1 \psi_0^1(x) + d_1^1 \psi_1^1(x)$$

$$\begin{aligned}
 I(x) = & 6 \times \text{[rectangle]} \quad \phi_0^0(x) \\
 & + 2 \times \text{[step down]} \quad \psi_0^0(x) \\
 & + 1 \times \text{[step up]} \quad \psi_0^1(x) \\
 & + -1 \times \text{[step down]} \quad \psi_1^1(x)
 \end{aligned}$$

WT for Image



Outline

- **Fourier and Wavelet Analysis**
- **Covariance Matrix of Gradients**
- **Principle of CNN**

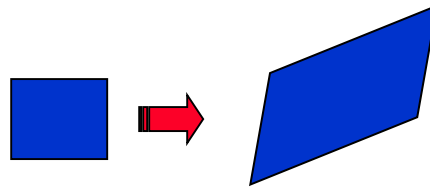
图像的空间变换

- 图像的空间变换

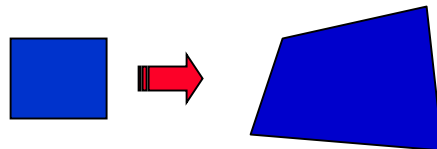
- Similarity
(translation,
scale, rotation)



- Affine



- Projective
(homography)



图像的空间变换

- 图像的相似性变换








$$(x', y') = T\{(x, y)\}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = T \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

homogeneous
coordinates

图像的空间变换

Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I & & t \end{bmatrix}_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$\begin{bmatrix} R & & t \end{bmatrix}_{2 \times 3}$	3	lengths	
similarity	$\begin{bmatrix} sR & & t \end{bmatrix}_{2 \times 3}$	4	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{2 \times 3}$	6	parallelism	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{3 \times 3}$	8	straight lines	

图像的空间变换






Transformation	Matrix	# DoF	Preserves	Icon
translation	$\begin{bmatrix} I & & t \end{bmatrix}_{3 \times 4}$	3	orientation	
rigid (Euclidean)	$\begin{bmatrix} R & & t \end{bmatrix}_{3 \times 4}$	6	lengths	
similarity	$\begin{bmatrix} sR & & t \end{bmatrix}_{3 \times 4}$	7	angles	
affine	$\begin{bmatrix} A \end{bmatrix}_{3 \times 4}$	12	parallelism	
projective	$\begin{bmatrix} \tilde{H} \end{bmatrix}_{4 \times 4}$	15	straight lines	

Image alignment

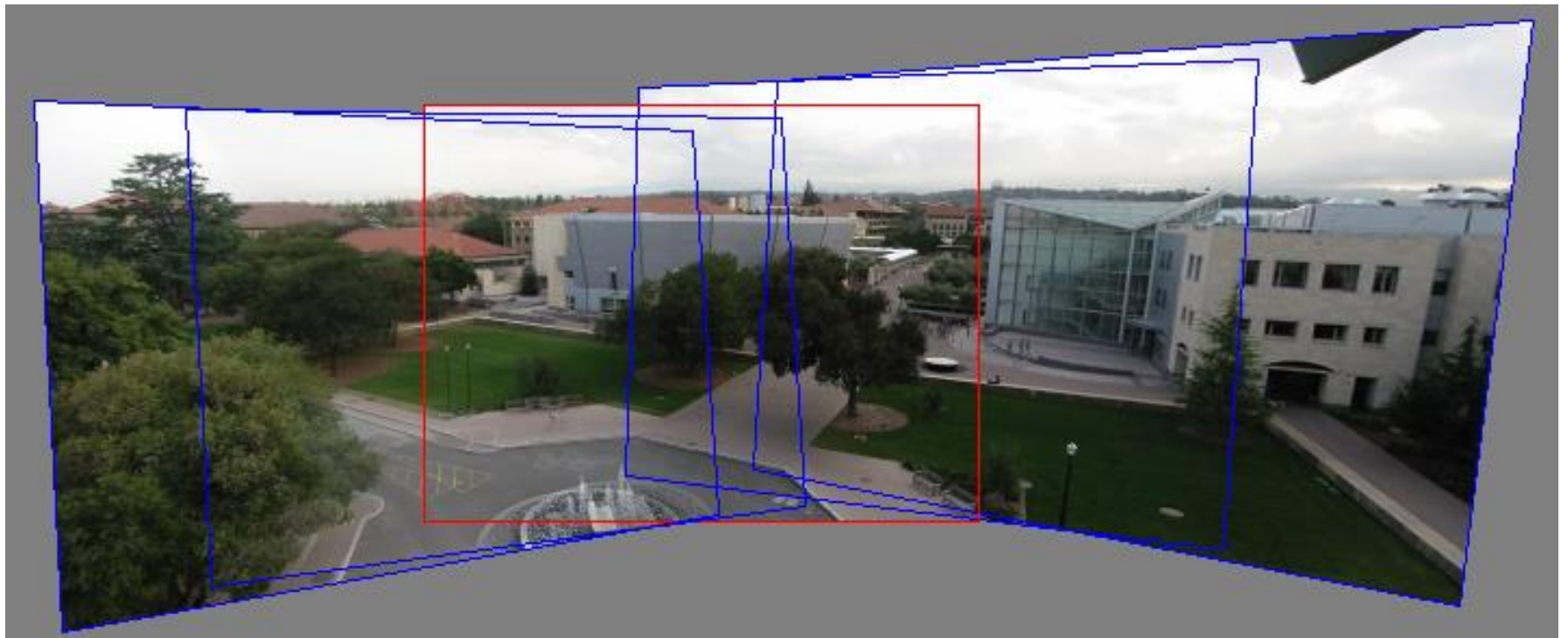


Image alignment

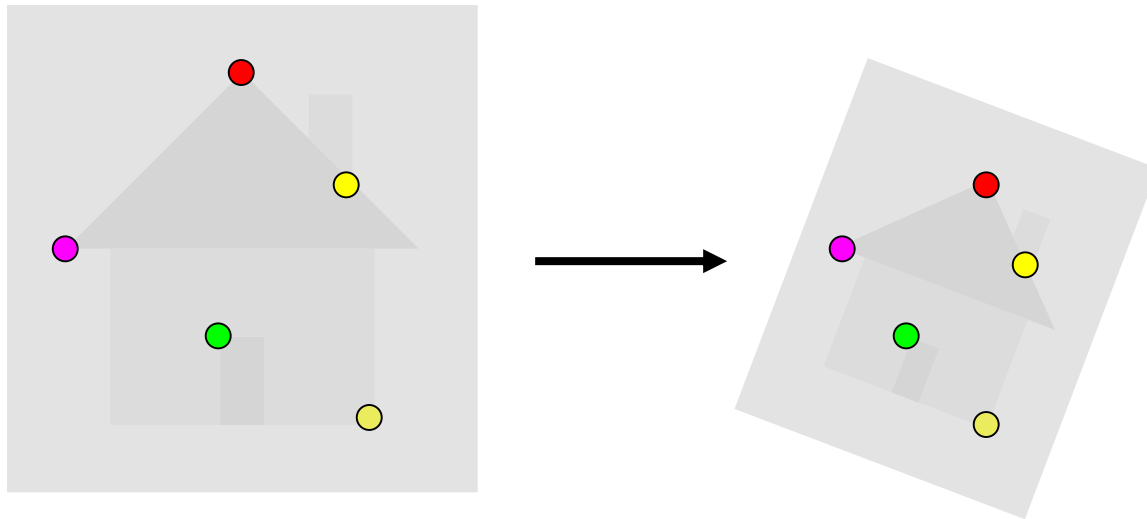
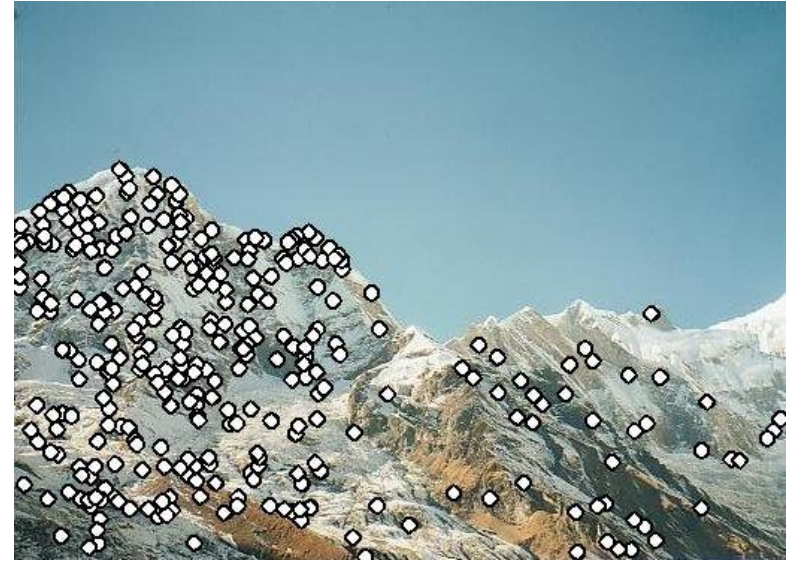
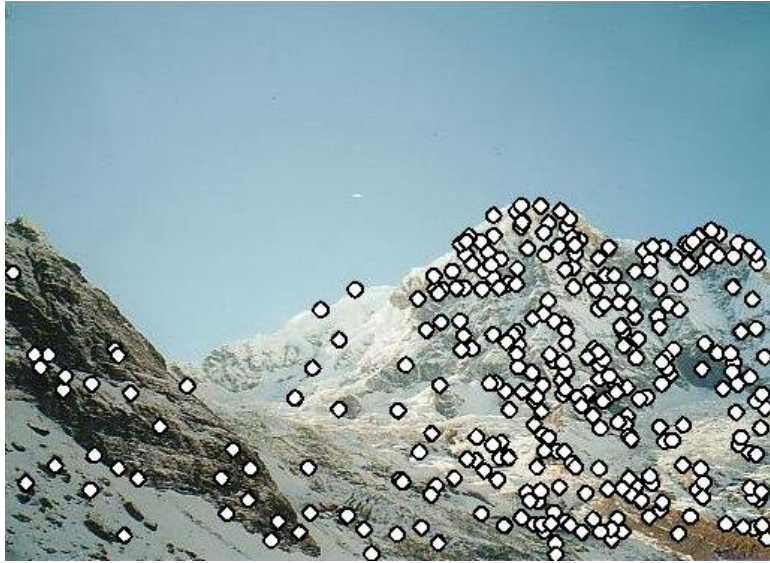


Image Alignment



Image Alignment



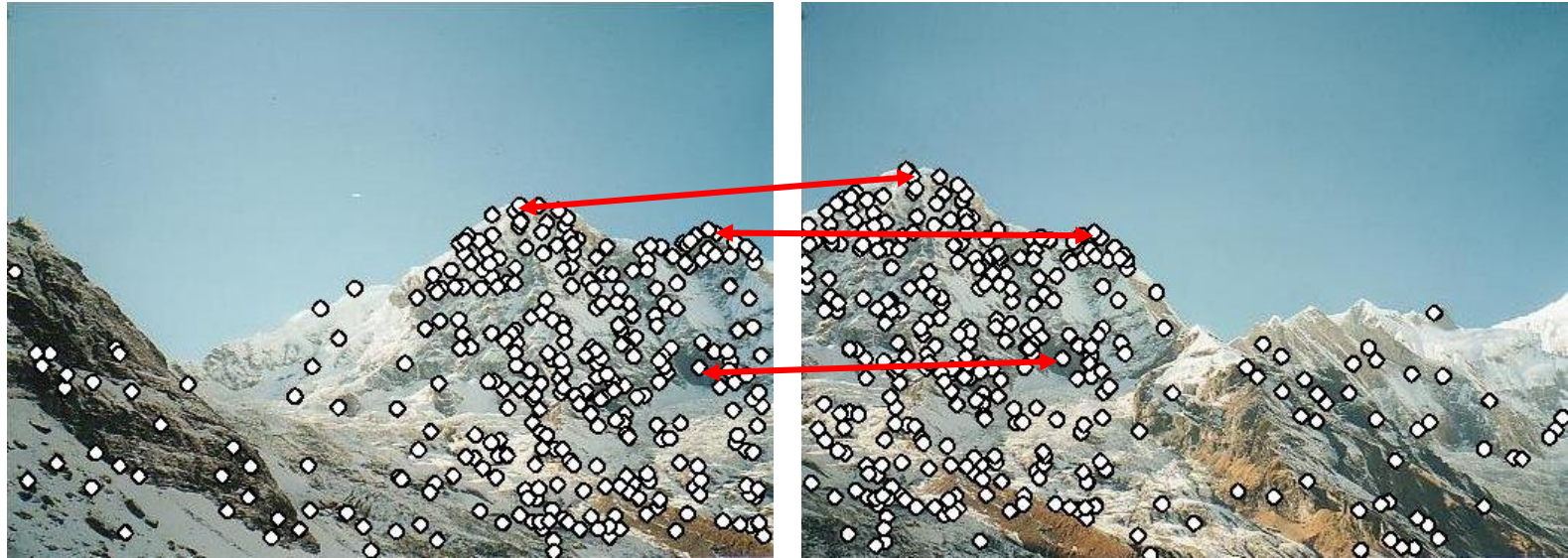
- 提取特征

Image Alignment



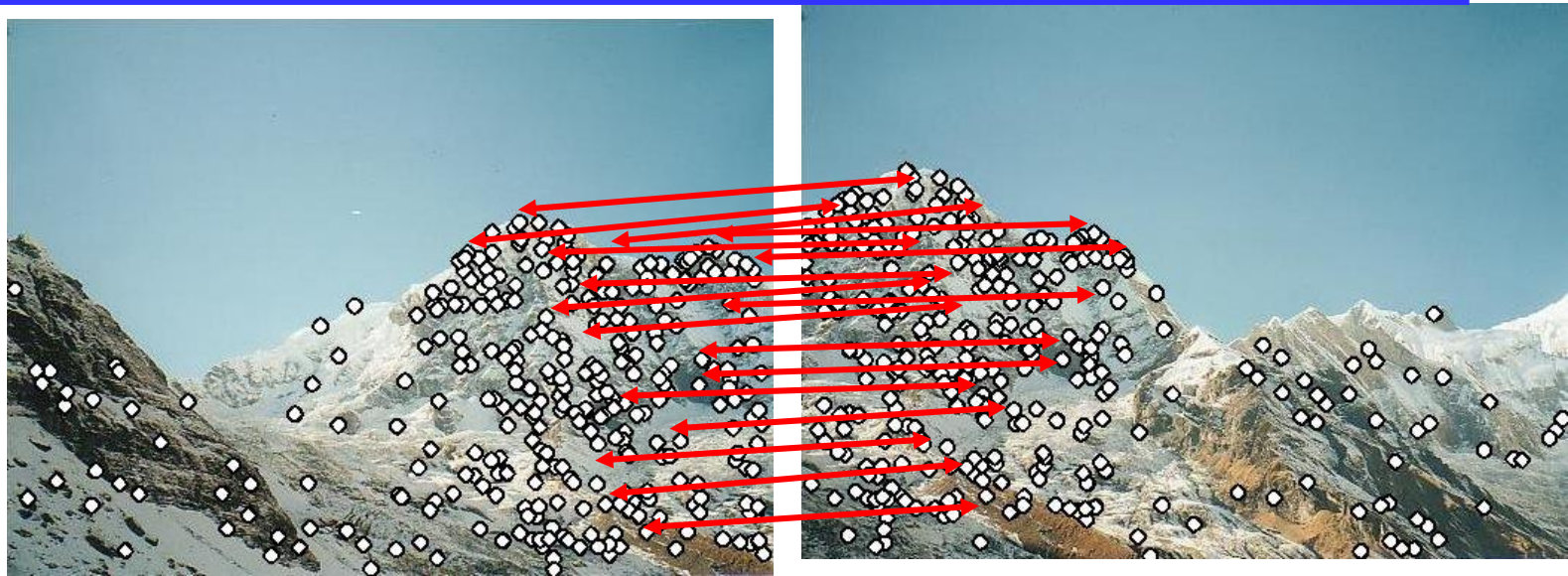
- 提取特征
- 特征匹配

Image Alignment



- 提取特征
- 特征匹配
- 迭代:
 - 选择三个特征点对, 得到初始变换 T

Image Alignment



- 提取特征
- 特征匹配
- 迭代:
 - 选择三个特征点对, 得到初始变换 T
 - 利用其它特征点对, 验证变换 T

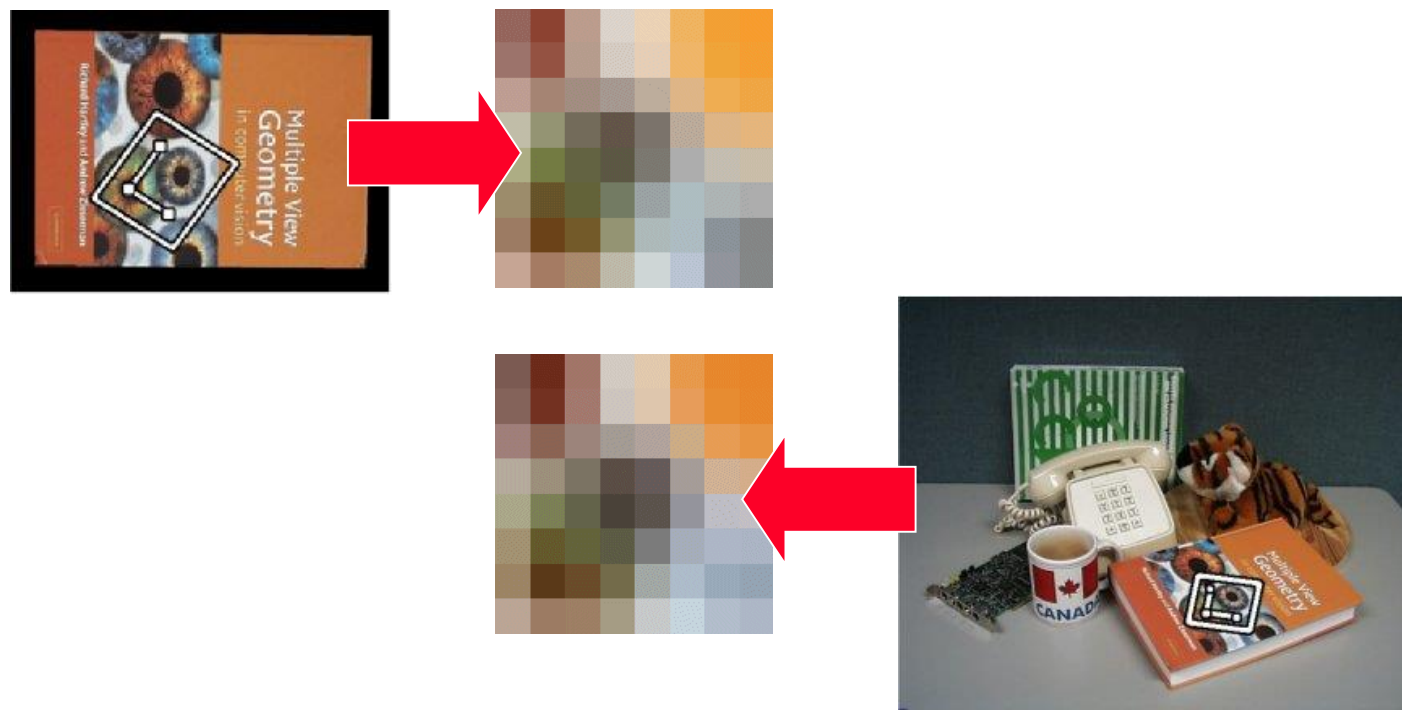
Image Alignment

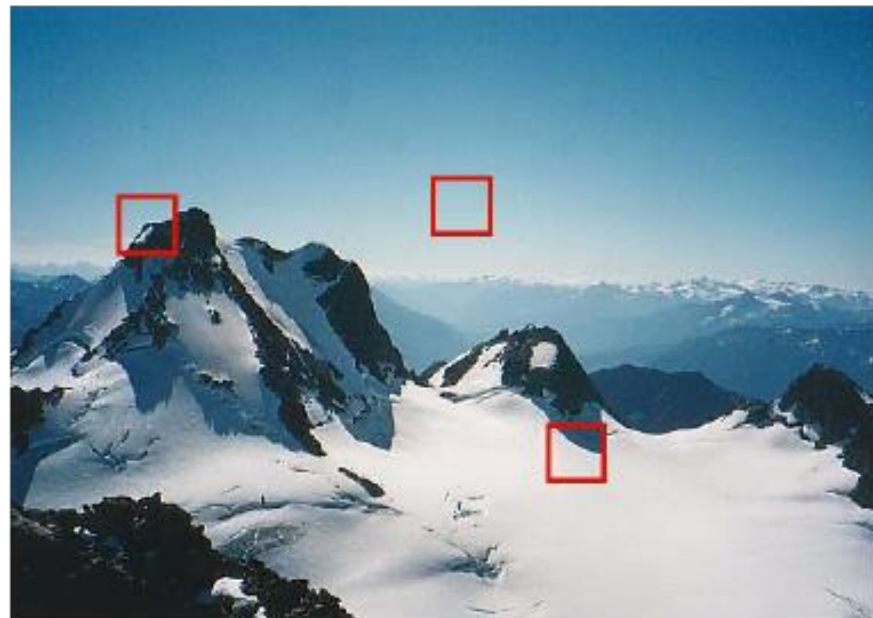
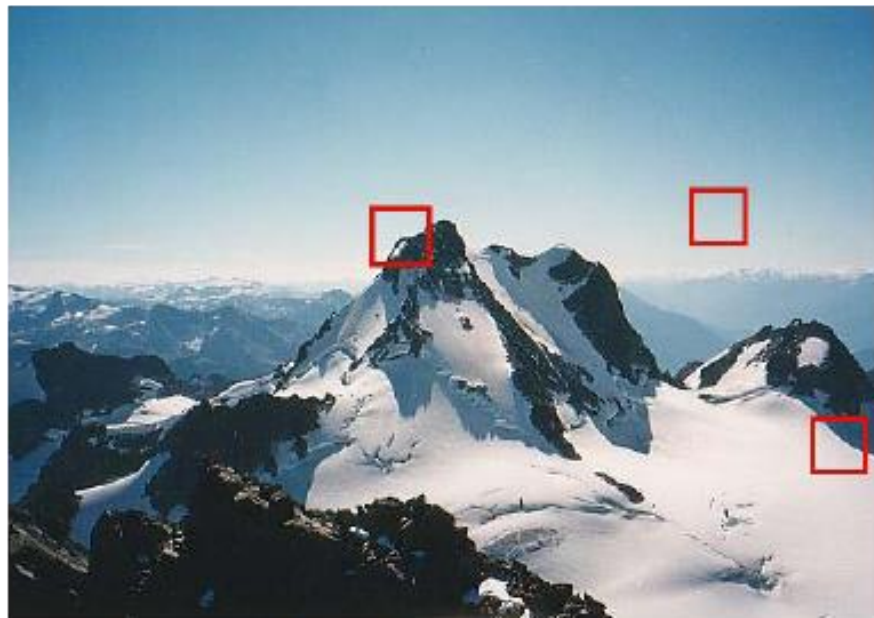


- 提取特征
- 特征匹配
- 迭代:
 - 选择三个特征点对, 得到初始变换 T
 - 利用其它特征点对, 验证变换 T

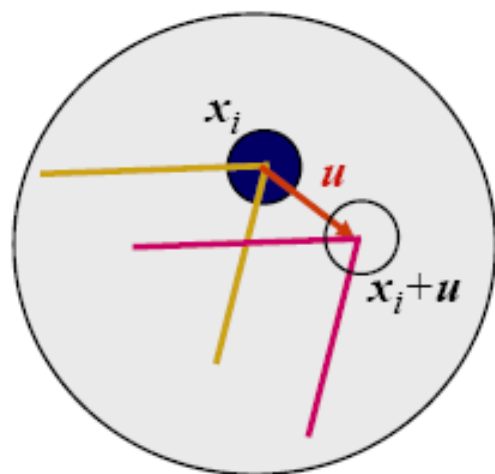
Feature Representation

- 特征描述需要具有对光照变化，噪声，视角的不变性

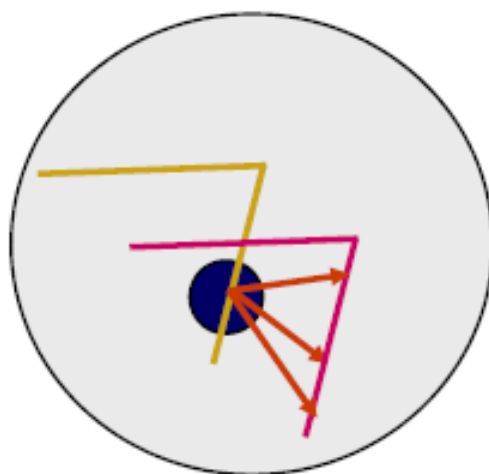




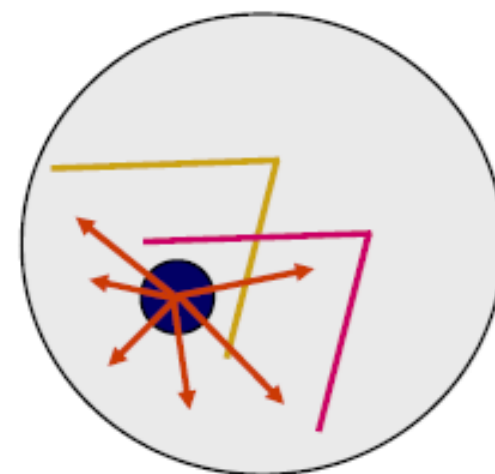
哪个图像块更容易被定位？



(a)



(b)



(c)

Aperture problems for different image patches:

判断两幅图像的相似度

$$E_{\text{WSSD}}(\mathbf{u}) = \sum_i w(\mathbf{x}_i) [I_1(\mathbf{x}_i + \mathbf{u}) - I_0(\mathbf{x}_i)]^2,$$

由于无法得到另一幅图像，只能在当前图像内寻找

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

根据泰勒级数展开，

$$I_0(\mathbf{x}_i + \Delta \mathbf{u}) \approx I_0(\mathbf{x}_i) + \nabla I_0(\mathbf{x}_i) \cdot \Delta \mathbf{u}$$

可得，

$$E_{\text{AC}}(\Delta \mathbf{u}) = \sum_i w(\mathbf{x}_i) [\nabla I_0(\mathbf{x}_i) \cdot \Delta \mathbf{u}]^2 = \Delta \mathbf{u}^T \mathbf{A} \Delta \mathbf{u},$$

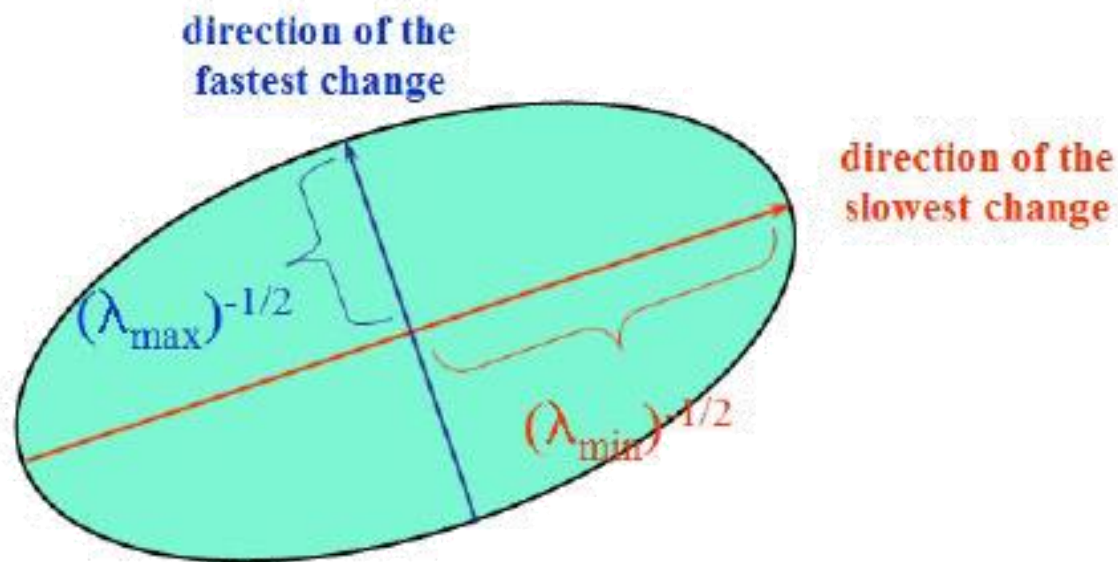
$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

图像自相关函数分析

- 矩阵A与信号处理中的自相关函数密切相关.
- 设 α , β 为矩阵A的特征值, 则 α , β 与局部自相关函数的主曲率成比例.
- 当两个曲率都低时, 局部自相关函数是平坦的, 那么窗口图像区域的灰度值近似为常量; 当只有一个曲率高而另一个曲率低时, 局部自相关函数呈脊形, 那么E只有当沿山脊移动时变化小, 这就表示是边缘; 当两个曲率都高时, 局部自相关函数是尖峰, 那么E在任意方向上移动都会增加, 这就表示是角点.
- 因此我们可以由 α , β 的值判断是否是角点。为了不对M进行分解求特征值, 可以采用 $\text{Tr}(A)$ 和 $\text{Det}(A)$ 来代替 α , β ,

$$\text{Tr}(A) = \alpha + \beta = A + B \quad A = I_x^2 \quad B = I_y^2$$

$$\text{Det}(A) = \alpha\beta = AB - C^2 \quad C = I_x I_y$$



$$\det(A) - \alpha \operatorname{trace}(A)^2 = \lambda_0 \lambda_1 - \alpha (\lambda_0 + \lambda_1)^2$$

$$\frac{\det A}{\operatorname{tr} A} = \frac{\lambda_0 \lambda_1}{\lambda_0 + \lambda_1}$$

(3) 计算兴趣值q与w

$$\omega = \frac{1}{\text{tr}Q} = \frac{\text{Det}N}{\text{tr}N}$$

Det N代表矩阵N之行列式

$$q = \frac{4\text{Det}N}{(\text{tr}N)^2}$$

tr N代表矩阵N之迹

(4) 确定待选点

$$\left. \begin{aligned} T_q &= 0.5 \sim 0.75 \\ T_w &= \begin{cases} f\bar{w} & (f = 0.5 \sim 1.5) \\ c\omega_c & (c = 5) \end{cases} \end{aligned} \right\}$$

当 $q > T_q$ 同时 $w > T_w$ ，该像素为待选点

(5) 选取极值点

即在一个适当窗口中选择最大的待选点

Harris角点算子

- 角点是景物轮廓线上曲率的局部极大点，对掌握景物的轮廓特征具有决定作用。一旦找到了景物的轮廓特征点也就大致掌握了景物的形状。直观的讲，角点就是图像上所显示的物体边缘拐角所在的位置点。
- Harris角点检测法是一种基于图像灰度的检测方法，这类方法主要通过计算点的曲率及梯度来检测角点。该方法是由Harris和Stephen于1988年提出来的，也叫Plessey角点检测法。其基本思想与Moravec角点算子相似，但对其作了许多改进。

Moravec角点算子

$$E_{x,y} = \sum_{u,v} W_{u,v} \left| I_{x+u,y+v} - I_{x,y} \right|^2$$

计算各像素沿同方向的平均灰度变化，选取最小值作为对应像素点的角点响应函数。定义在一定范围内具有最大角点响应的像素点为角点。

Moravec角点算子简单快速，但是它存在一些缺点：

- 各向异性, 只考虑四个方向
- 对强边界敏感, 只考虑了兴趣值的最小值

Harris角点算子

$$E_{x,y} = \sum_{u,v} W_{u,v} [I_{x+u,y+v} - I_{u,v}]^2 = \sum_{u,v} W_{u,v} [xX + yY + O(x^2, y^2)]^2$$

这里一阶微分可以由下面的式子近似

$$X = I \otimes (-1, 0, 1) = \partial I / \partial x$$

$$Y = I \otimes (-1, 0, 1)^T = \partial I / \partial y$$

因此, E可以表现成:

$$E(x, y) = Ax^2 + 2Cxy + By^2$$

$$A = X^2 \otimes W$$

$$B = Y^2 \otimes W$$

$$C = (XY) \otimes W$$

Harris角点算子

为消除噪声, 进行高斯平滑:

$$W_{u,v} = \exp - (u^2 + v^2) / 2\sigma^2$$

在平移方向 (x, y) 上的E可以表示如下:

$$E(x, y) = (x, y) M (x, y)^T$$

$$M = G(\tilde{s}) \otimes \begin{bmatrix} g_x & g_x g_y \\ g_x g_y & g_y \end{bmatrix}$$

M为 2×2 的矩阵:

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

Harris角点算子

兴趣值的计算公式为:

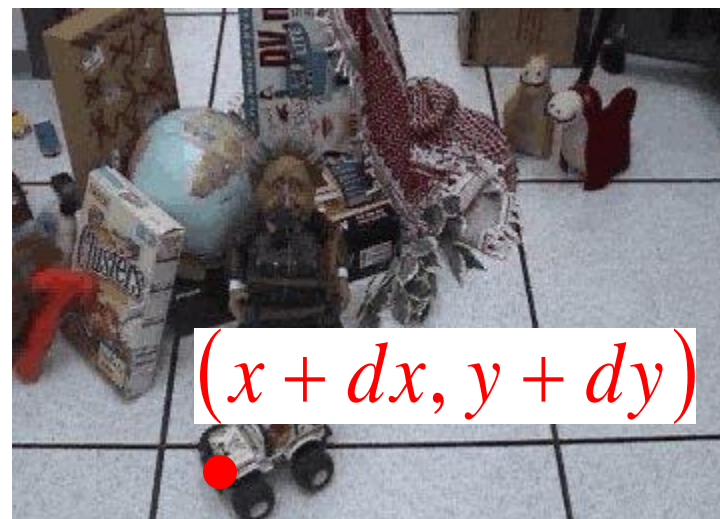
$$I = \text{Det}(M) - k\text{Trace}^2(M), k = 0.04$$

- Harris算法认为，特征点是局部范围内的极大兴趣值对应的像素。因此，在计算完各点的兴趣值后，要提取出原始图像中的所有局部兴趣值最大的点。
- 在提取特征点时，凡满足大于某一阈值的像素点均可被认为是特征点。
- 阈值选取的方法：通过确定图像中所能提取的最大可能的特征点数目 N 来选择值最大的若干像素点作为特征点。局部极值点的数目往往很多，根据 I 值进行排序，取其前 N 个为特征点。

二维运动估计

- 关键性假设

- 假设各幅图像的亮度具有一致性
- 运动较小



二维运动估计

- 基础性假设

点的亮度变化仅由运动引起

$$I(\mathbf{x}, t) \approx I(\mathbf{x} + \delta \mathbf{x}, t + \delta t)$$

↓ Taylor展开

$$I(\mathbf{x}, t) = I(\mathbf{x}, t) + \nabla I \cdot \delta \mathbf{x} + \delta t I_t + O^2$$

↓

$$\nabla I \cdot \mathbf{v} + I_t = 0 \quad \text{光流约束方程}$$

物理意义:如果一个固定的观察者观察一幅活动的场景,那么所得图象上某点灰度的(一阶)时间变化率是场景亮度变化率与该点运动速度的乘积。

Application of Covariance Matrix

- Prob: we have more equations than unknowns

$$\begin{matrix} A & d = b \\ 25 \times 2 & 2 \times 1 \quad 25 \times 1 \end{matrix} \longrightarrow \text{minimize } \|Ad - b\|^2$$

- Solution: solve least squares problem
 - minimum least squares solution given by solution (in d) of:

$$\begin{matrix} (A^T A) & d = A^T b \\ 2 \times 2 & 2 \times 1 \quad 2 \times 1 \end{matrix}$$

$$\begin{matrix} \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} & \begin{bmatrix} u \\ v \end{bmatrix} = - & \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} \\ A^T A & & A^T b \end{matrix}$$

- The summations are over all pixels in the K x K window

Application of Covariance Matrix

– Optimal (u, v) satisfies Lucas-Kanade equation

$$\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}$$

$A^T A$ $A^T b$

When is This Solvable?

- $A^T A$ should be invertible
- $A^T A$ should not be too small due to noise
 - eigenvalues λ_1 and λ_2 of $A^T A$ should not be too small
- $A^T A$ should be well-conditioned
 - λ_1 / λ_2 should not be too large (λ_1 = larger eigenvalue)

Application of Covariance Matrix

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

- Suppose (x,y) is on an edge. What is $A^T A$?
 - gradients along edge all point the same direction
 - gradients away from edge have small magnitude

$$\left(\sum \nabla I (\nabla I)^T \right) \approx k \nabla I \nabla I^T$$

$$\left(\sum \nabla I (\nabla I)^T \right) \nabla I = k \|\nabla I\| \nabla I$$

- ∇I is an eigenvector with eigenvalue $k \|\nabla I\|$

Application of Covariance Matrix

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] = \sum \nabla I (\nabla I)^T$$

- Suppose (x,y) is on an edge. What is $A^T A$?

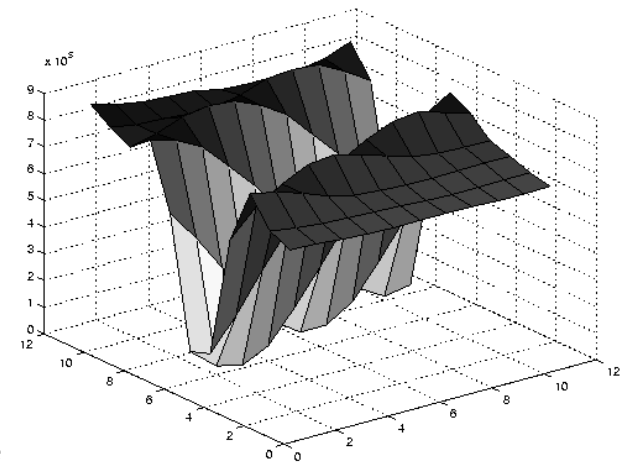
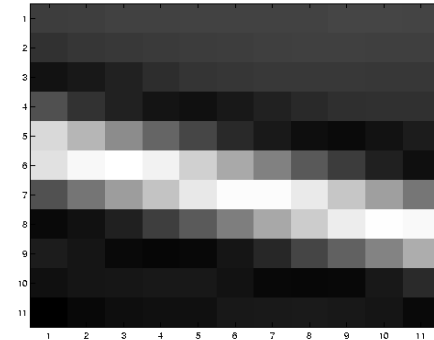
- What's the other eigenvector of $A^T A$?

- let N be perpendicular to ∇I

$$\left(\sum \nabla I (\nabla I)^T \right) N = 0$$

- N is the second eigenvector with eigenvalue 0
- The eigenvectors of $A^T A$ relate to edge direction and magnitude

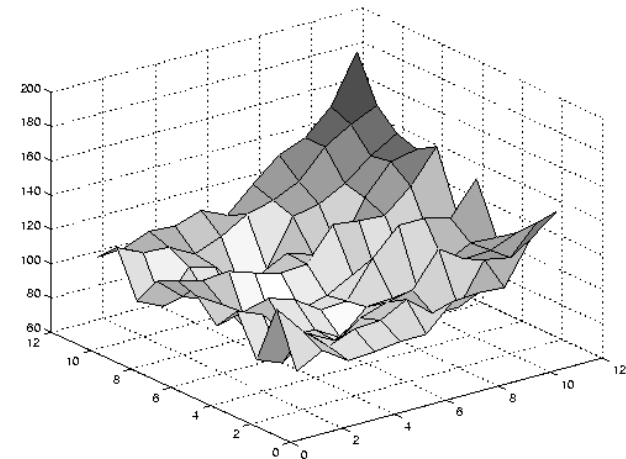
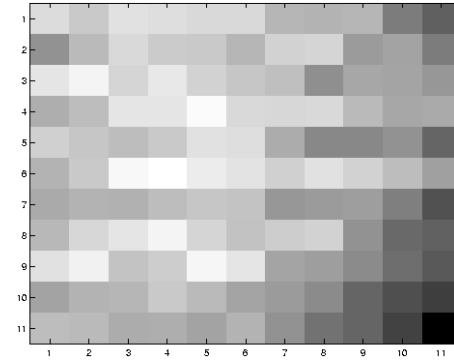
Covariance Matrix of Edge



$$\sum \nabla I (\nabla I)^T$$

- large gradients, all the same
- large λ_1 , small λ_2

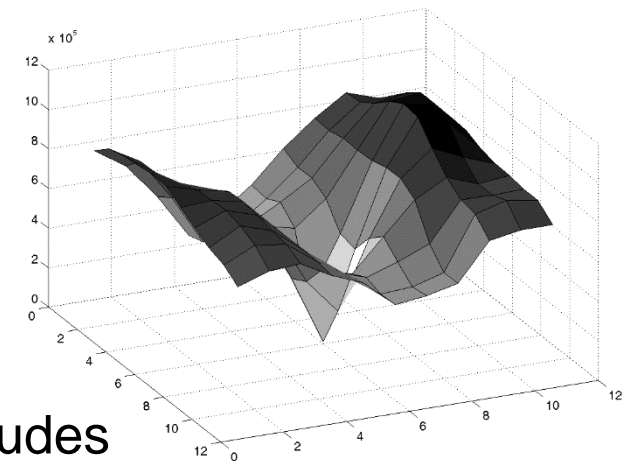
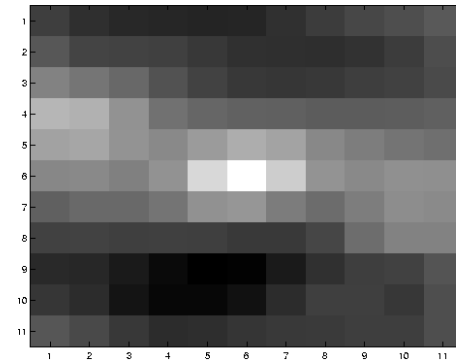
Covariance Matrix of Low Texture



$$\sum \nabla I (\nabla I)^T$$

- gradients have small magnitude
- small λ_1 , small λ_2

Covariance Matrix of High Texture



$$\sum \nabla I (\nabla I)^T$$

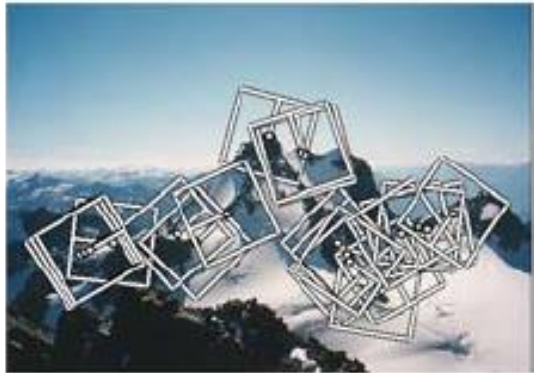
- gradients are different, large magnitudes
- large λ_1 , large λ_2

SIFT算法

- SIFT算法由加拿大University of British Columbia的D.G.Lowe 1999年提出, 2004年完善总结。
- Object Recognition from Local Scale-Invariant Features(ICCV,1999)
- Distinctive Image Features from Scale-Invariant Keypoints(IJCV,2004)

SIFT 主要思想

- SIFT算法是一种提取局部特征的算法，在尺度空间寻找极值点, 并确定关键点(Key points)的位置和关键点所处的尺度；
- 然后使用关键点邻域梯度的主方向作为该点的方向特征，以实现算子对尺度和方向的无关性。



SIFT算法的主要特点

- a) SIFT特征是图像的局部特征，其对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声也保持一定程度的稳定性。
- b) 独特性(Distinctiveness)好，信息量丰富，适用于在海量特征数据库中进行快速、准确的匹配。
- c) 多量性，即使少数的几个物体也可以产生大量SIFT特征向量。
- d) 高速性，经优化的SIFT匹配算法甚至可以达到实时的要求。
- e) 可扩展性，可以很方便的与其他形式的特征向量进行联合。

SIFT算法步骤

1. 检测尺度空间极值点
2. 精确定位极值点
3. 为每个关键点指定方向参数
4. 关键点描述子的生成
5. 特征匹配

1. 检测尺度空间极值点

- 在对有噪声的图像求取边缘点时，可以先用平滑滤波器对图像平滑，然后再对平滑后的图像求两阶微分，并检测局部极值点.
- 高斯卷积核是实现尺度变换的唯一线性核，所以SIFT算法中使用了高斯滤波器对图像进行平滑处理.

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

(x, y) 是空间坐标, σ 是尺度参数。

尺度空间的生成

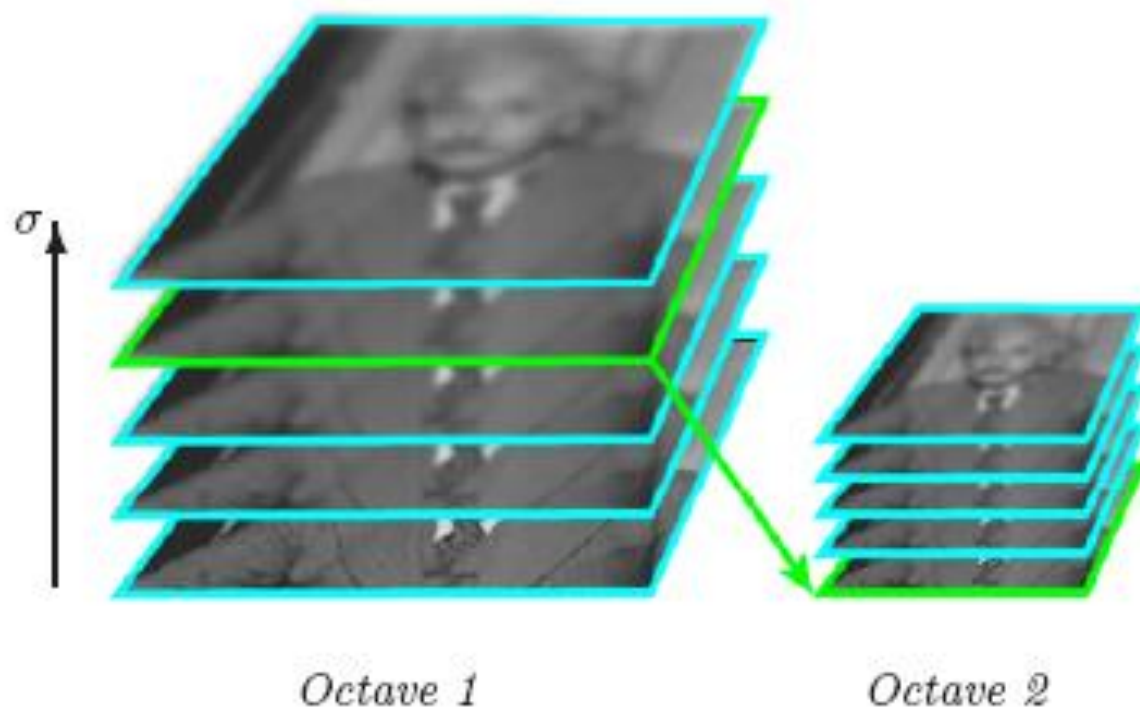
- 尺度空间理论是通过对原始图像进行尺度变换，获得图像多尺度下的尺度空间表示序列，对这些序列进行尺度空间主轮廓的提取，并以该主轮廓作为一种特征向量，实现边缘、角点检测和不同分辨率上的特征提取等。

- 一副二维图像的尺度空间定义为：

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

- 对原始图像做多次的高斯平滑处理，也就得到了一个多尺度图像金字塔
- 图像金字塔能够很好的显示图像多尺度空间的特征，SIFT算法中的图像金字塔共有o组，每组有S层，下一组的图像由上一组图像降采样得到。

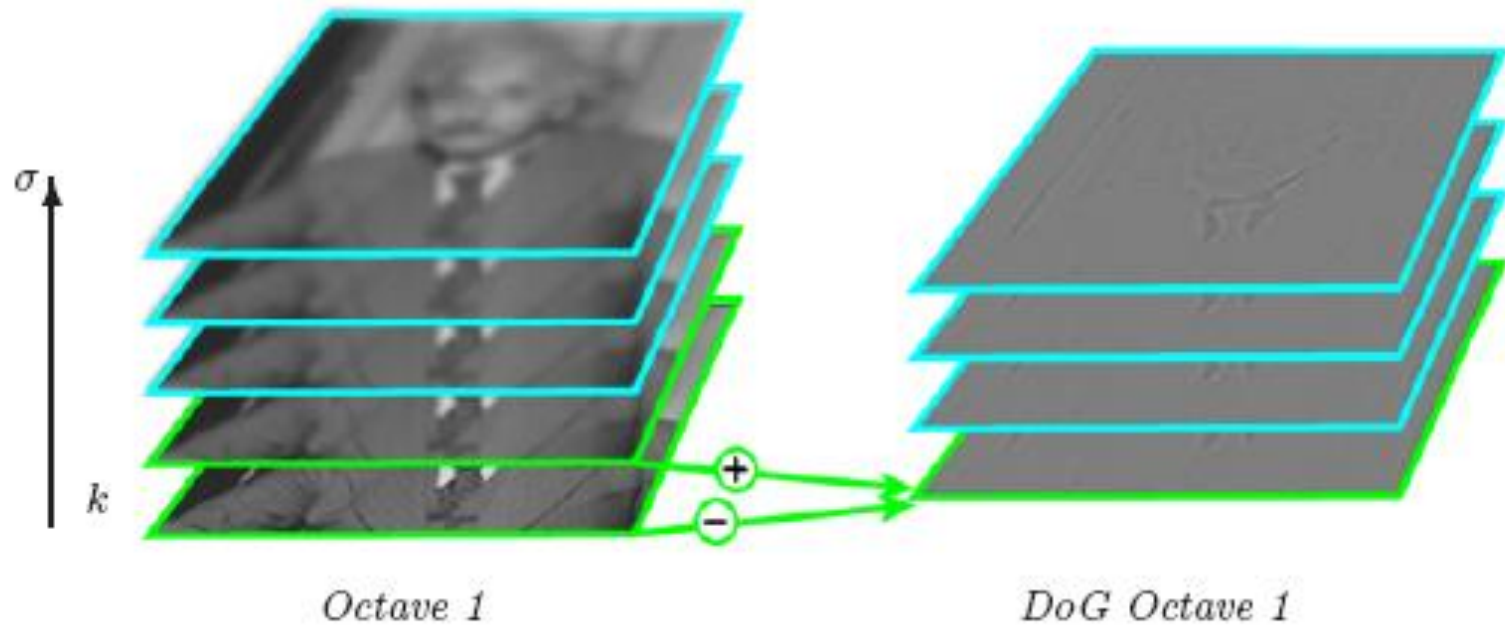
多尺度图像金字塔

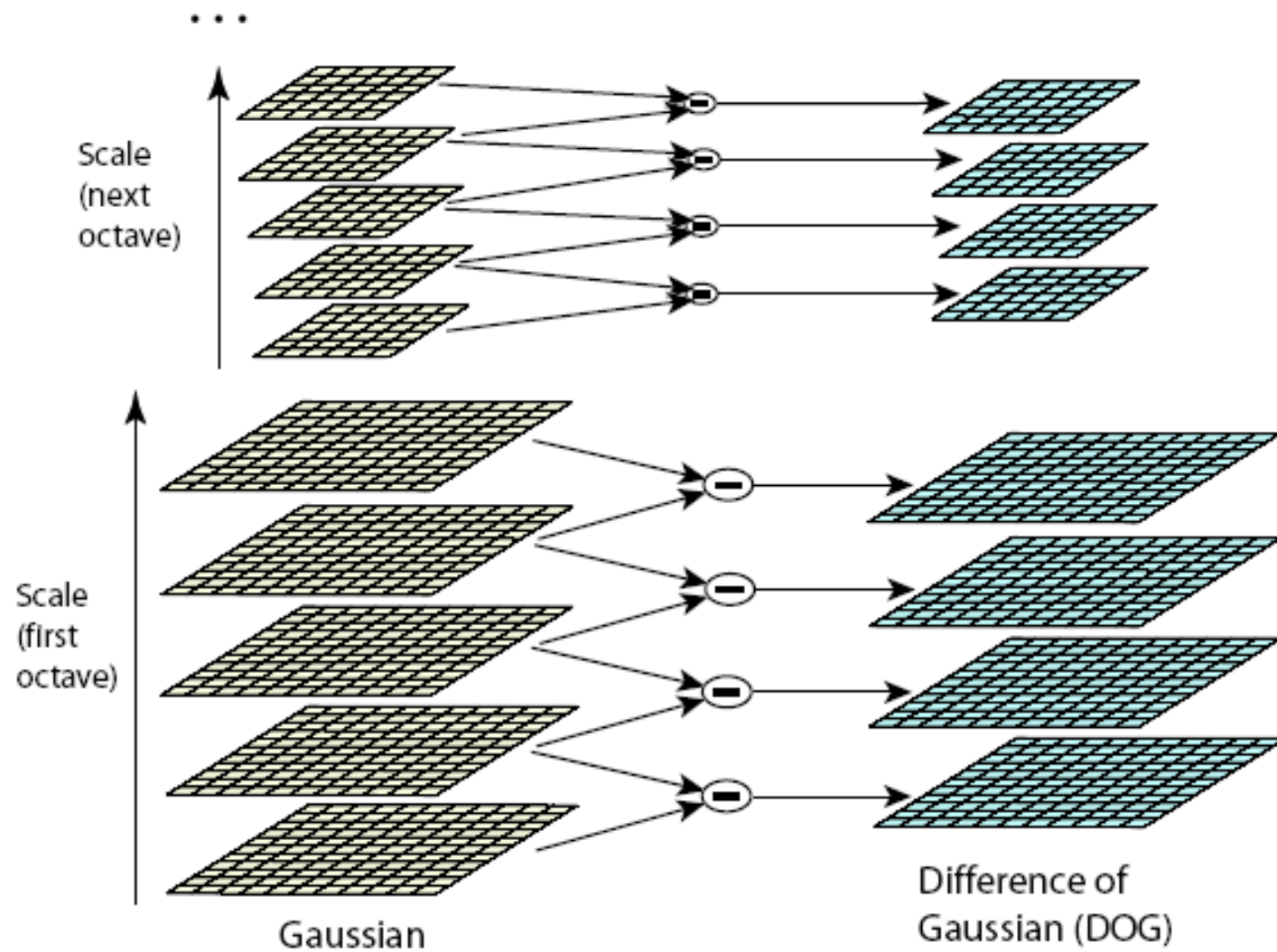


DOG图像金字塔

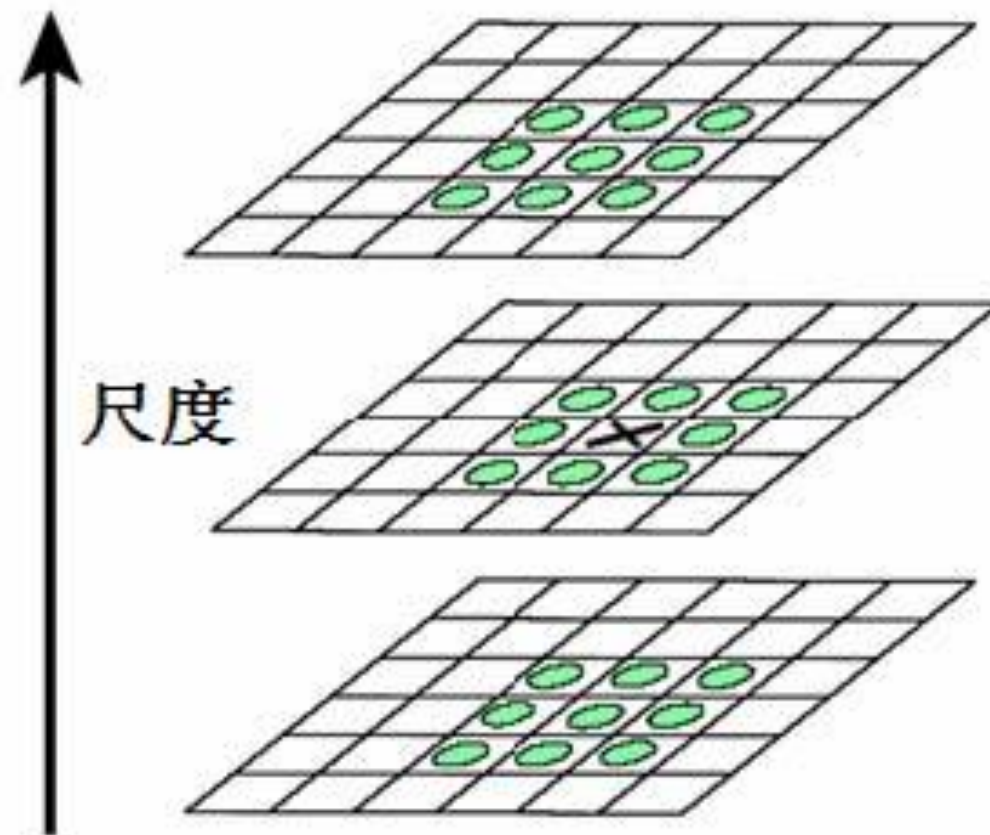
- 对平滑后的图像求两阶微分，并检测局部极值点.
- 对于高斯函数而言，其两阶微分可以用两个不同尺度高斯函数的差近似表示.
- 根据这一原理可以得到DOG图像金字塔

DOG图像金字塔





检测极值点



构建DOG尺度空间需确定的参数

σ — 尺度空间坐标; O — octave坐标; S — sub-level 坐标
 $o \in o_{\min} + [0, \dots, O-1]$,

- σ 和 O 、 S 的关系

$$\sigma(o, s) = \sigma_0 2^{o+s/S} \quad s \in [0, \dots, S-1] \text{ 为基准层尺度}$$

$$x = 2^o x_0, o \in \mathbb{Z}, x_0 \in [0, \dots, N_0-1] \times [0, \dots, M_0-1]$$

- 空间坐标 x 是组 octave 的函数, 设是 o 组的空间坐标, 则

$$(M_0, N_0) \quad N_0 = \left\lfloor \frac{N_0}{2^o} \right\rfloor, M_0 = \left\lfloor \frac{M_0}{2^o} \right\rfloor$$

- 如果 σ_0 是基础组 $o=0$ 的分辨率, 则其他组的分辨率由下式获得: $\sigma_n = 0.5, \sigma_0 = 1.6 \cdot 2^{1/S}, o_{\min} = -1, S=3$

2. 极值点定位

- 在某个像素点被确定为尺度空间的极值点以后，下一步的任务就是确定它的位置、尺度及主曲率，同时去除低对比度的关键点和不稳定的边缘响应点(因为DoG算子会产生较强的边缘响应)，增强匹配稳定性、提高抗噪声能力

极值点位置的确定

- Lowe99年的文章中极值点被简单定位在中间采样点上。
- 在2004的论文中对此进行了改进。通过拟和三维二次函数以精确确定关键点的位置和尺度（达到亚像素精度）。
- 这个改进减少了提取特征的数量, 增强了匹配的鲁棒性

边缘响应的去除

- 其主要思想是一个定义不好的高斯差分算子的极值在横跨边缘的地方有较大的主曲率，而在垂直边缘的方向有较小的主曲率。
- 主曲率通过一个2x2 的Hessian矩阵 H 求出，矩阵中的导数可以由采样点相邻差估计得到。

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

D的主曲率和**H**的特征值成正比，令 α 为最大特征值， β 为最小的特征值，则

$$\text{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$

$$\text{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

令 $\alpha = r\beta$ ，则：

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r},$$

$(r + 1)^2/r$ 的值在两个特征值相等的时候最小，随着 r 的增大而增大，因此，为了检测主曲率是否在某阈值 r 下，只需检测

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r + 1)^2}{r}.$$

在Lowe的文章中，取 $r=0.46$ 。

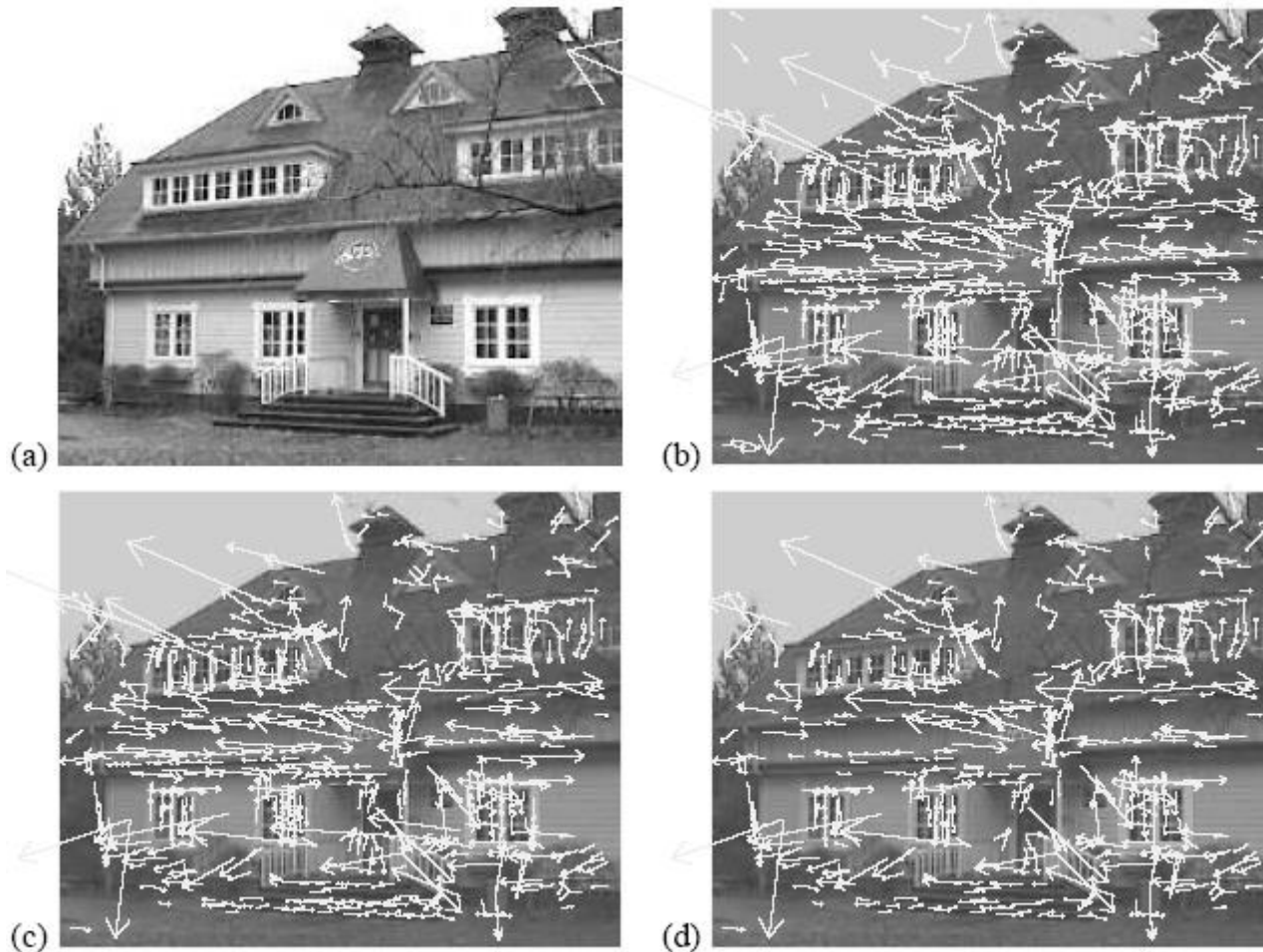


Figure 5: This figure shows the stages of keypoint selection. (a) The 233x189 pixel original image. (b) The initial 832 keypoints locations at maxima and minima of the difference-of-Gaussian function. Keypoints are displayed as vectors indicating scale, orientation, and location. (c) After applying a threshold on minimum contrast, 729 keypoints remain. (d) The final 536 keypoints that remain following an additional threshold on ratio of principal curvatures.

3. 关键点朝向

- 主要思想

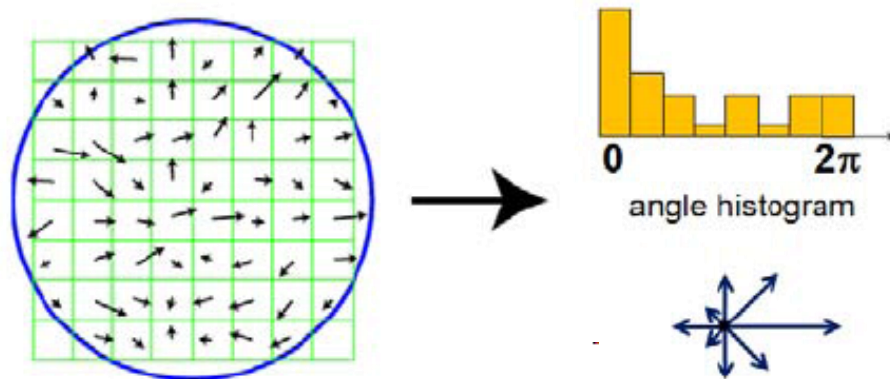
利用关键点邻域像素的梯度方向分布特性为每个关键点指定方向参数，使算子具备旋转不变性。

- (x, y) 处的梯度值和梯度方向公式

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

确定关键点朝向

- 在实际计算时，我们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。直方图的峰值则代表了该关键点处邻域梯度的主方向，即作为该关键点的方向。
- 在梯度方向直方图中，当存在另一个相当于主峰值80%能量的峰值时，则将这个方向认为是该关键点的辅方向。一个关键点可能会被指定具有多个方向（一个主方向，一个以上辅方向），这可以增强匹配的鲁棒性。



4. 关键点描述子的生成

- 至此，图像的关键点已检测完毕，每个关键点有三个信息：位置、所处尺度、方向。
- 由此可以确定一个SIFT特征区域, 用箭头表示

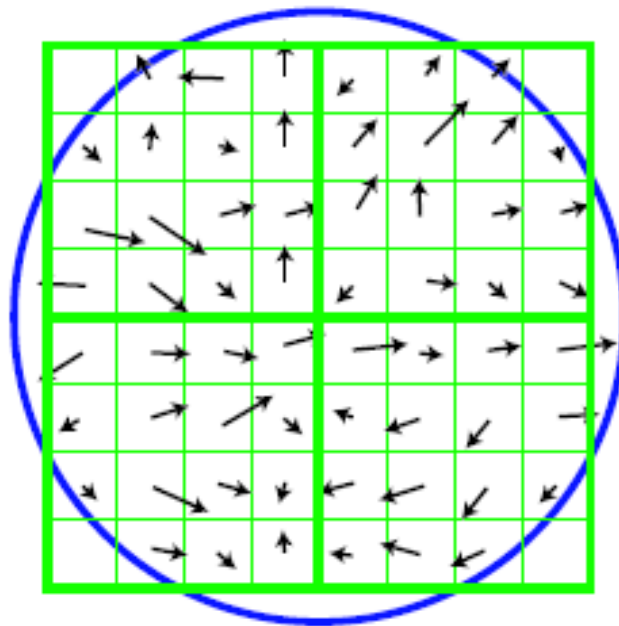
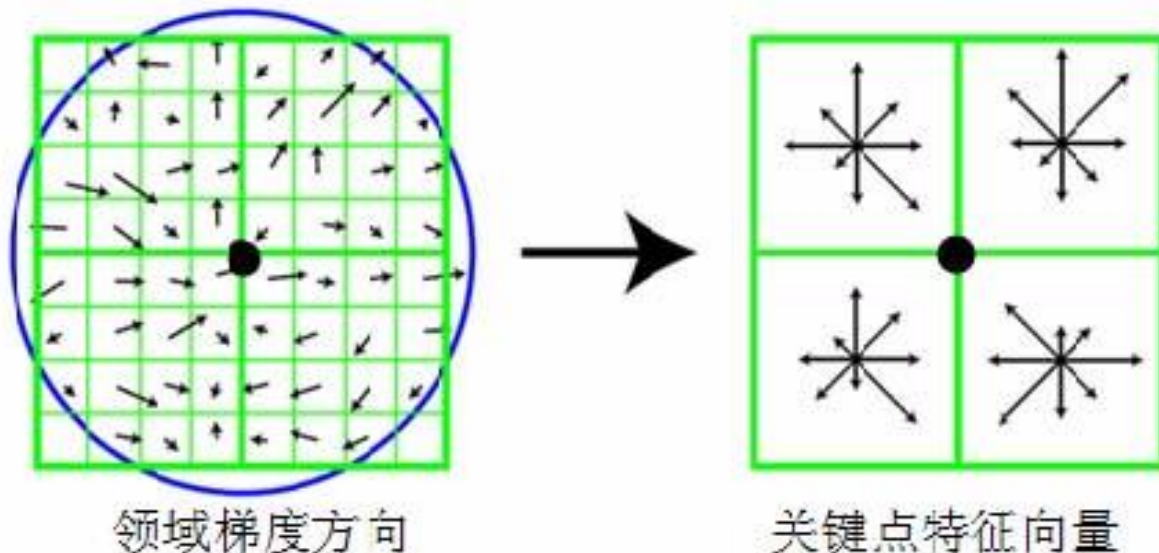


Image gradients

关键点描述子的生成

- 接下来以关键点为中心取 8×8 的窗口。图左部分的中央黑点为当前关键点的位置，每个小格代表关键点邻域所在尺度空间的一个像素，箭头方向代表该像素的梯度方向，箭头长度代表梯度模值，图中蓝色的圈代表高斯加权的范围（越靠近关键点的像素梯度方向信息贡献越大）。然后在每 4×4 的小块上计算8个方向的梯度方向直方图，绘制每个梯度方向的累加值，即可形成一个种子点，



关键点描述子的生成

- 首先将坐标轴旋转为关键点方向，以确保旋转不变性。
- 此图中一个关键点由 2×2 共4个种子点组成，每个种子点有8个方向向量信息。这种邻域方向性信息联合的思想增强了算法抗噪声的能力，同时对于含有定位误差的特征匹配也提供了较好的容错性。
- 实际计算过程中，为了增强匹配的稳健性，Lowe建议对每个关键点使用 4×4 共16个种子点来描述，这样对于一个关键点就可以产生128个数据，即最终形成128维的SIFT特征向量。
- 此时SIFT特征向量已经去除了尺度变化、旋转等几何变形因素的影响，再继续将特征向量的长度归一化，则可以进一步去除光照变化的影响。

5. 特征匹配

- 当两幅图像的SIFT特征向量生成后，下一步我们采用关键点特征向量的欧式距离来作为两幅图像中关键点的相似性判定度量。取图像1中的某个关键点，并找出其与图像2中欧式距离最近的前两个关键点，在这两个关键点中，如果最近的距离除以次近的距离少于某个比例阈值，则接受这一对匹配点。降低这个比例阈值，SIFT匹配点数目会减少，但更加稳定。

SIFT的应用



SIFT的应用

