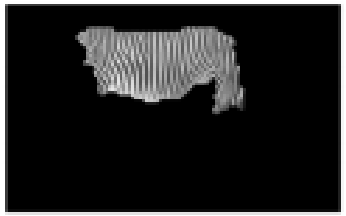
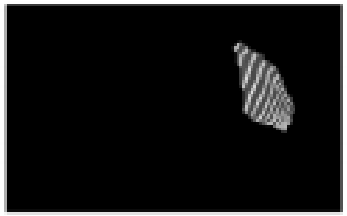
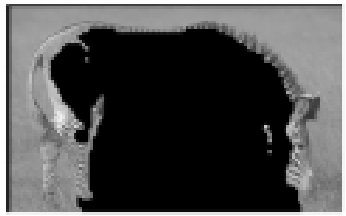
图像处理与识别

——Part 6 图像分割

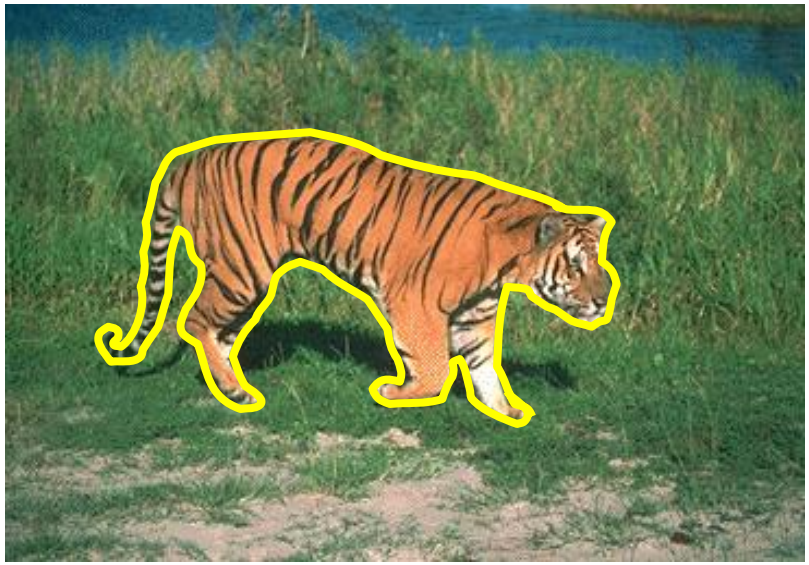
主讲：张磊

图像分割

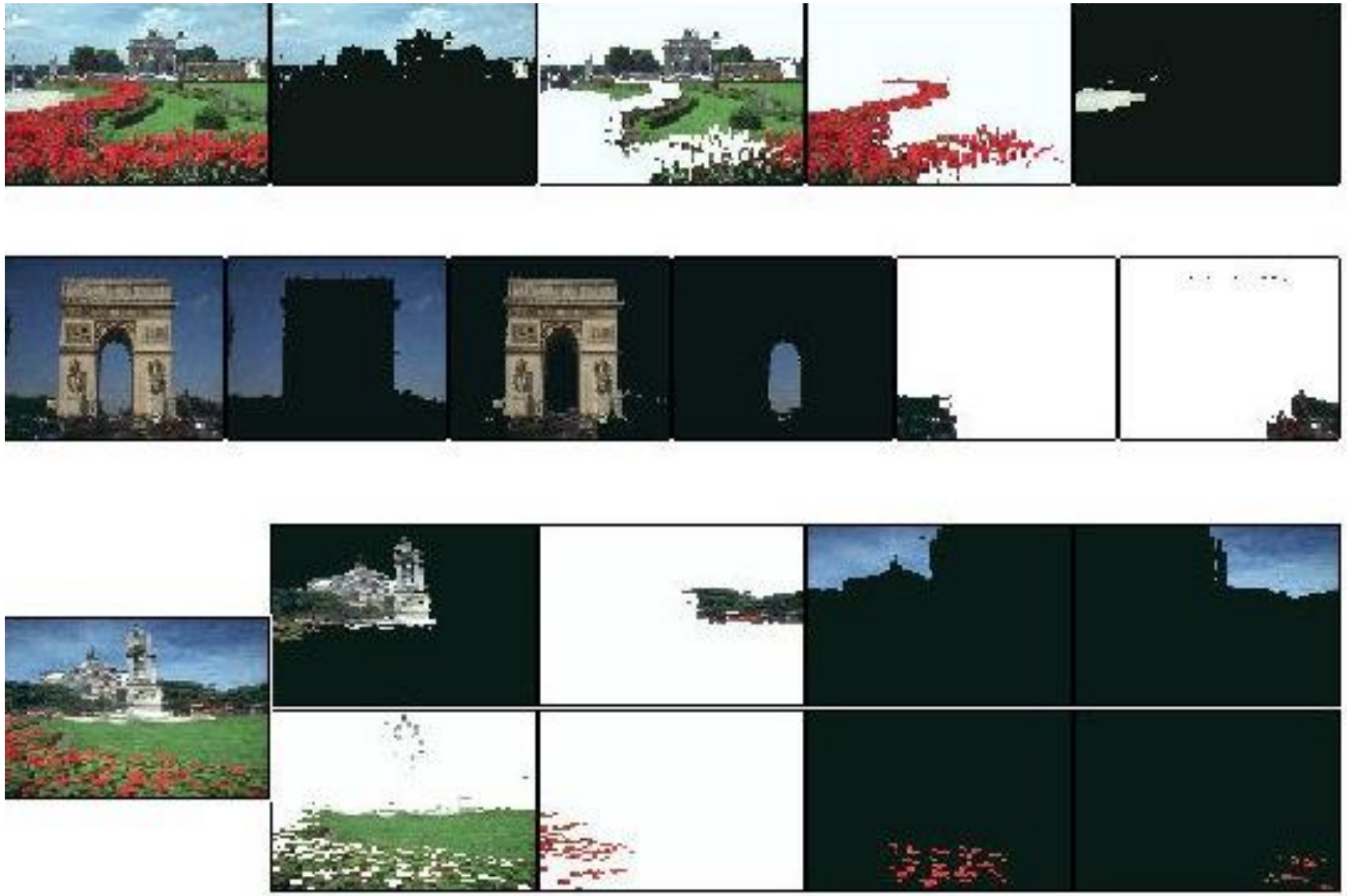







图像分割



图像分割



课外分享

图像分割基础

分割意义

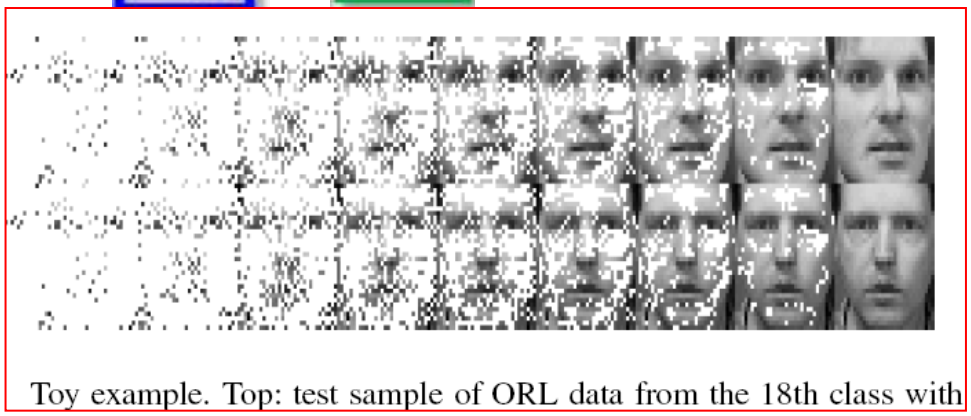
- 数字图像处理的两个目的：图像的处理与加工；图像中目标物的分析与理解。目标物分析过程包括：
  - ✓ 把图像分割成不同目标物和背景区域(检测)
  - ✓ 对分割出来的目标物进行特征描述
  - ✓ 对目标物进行识别和理解
- 图像分析涉及到“感兴趣的目标物区域”，若只对感兴趣区域进行特征描述和识别分析，那么需要对图像进行分割，把目标物区域与背景分开。
- 图像分割是图像分析和理解的关键步骤，分割效果的好坏将影响目标物的特征描述和识别。

图像分割基础

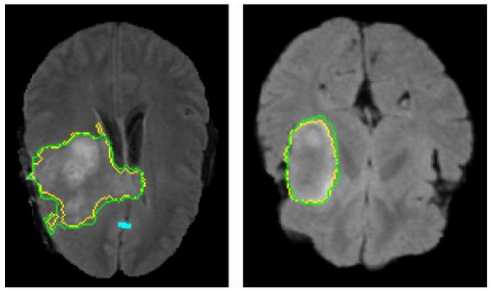
- 分割意义

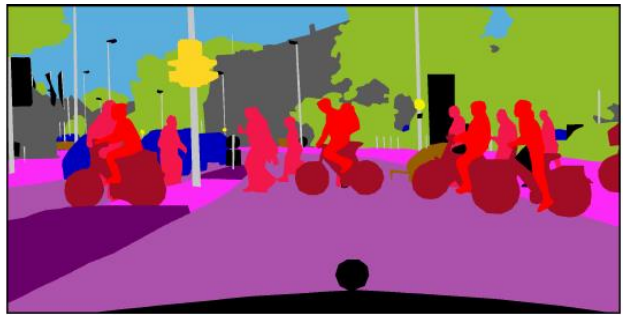
实际分割应用：

- ✓ 图像中的人脸识别
- ✓ 图像中的文字识别
- ✓ 图像中的汽车识别
- ✓ 车牌号识别
- ✓ 医学图像分割
- ✓ 语义分割。。。

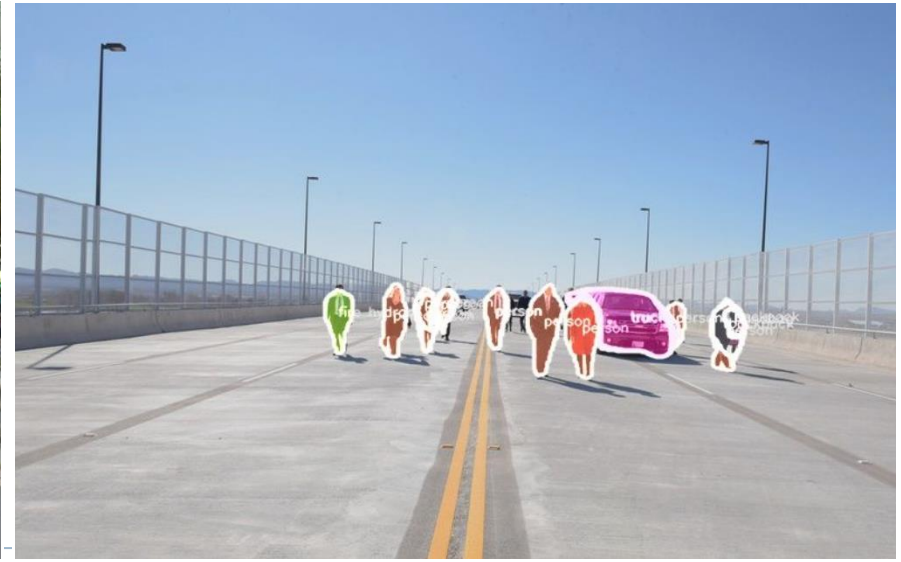


Toy example. Top: test sample of ORL data from the 18th class with





图像实例分割(instance segmentation)



图像分割基础

- 图像分割的定义

令R表示整个图像区域，对R的分割可看作将R分成若干个满足如下5个条件的非空子集或子区域R1，R2，...,Rn

- ✓ **完备性** $\bigcup_{i=1}^{n} R_i = R$。分割出的全部子区域的总和包括图像的所有像素，即每个像素都被划分到一个子区域(分割必须是完全的)
- ✓ **独立性** $R_i \cap R_j = \emptyset, \forall i, j, i \neq j$。一个像素不能同时属于两个及以上子区域，即每个子区域互不重叠
- ✓ **相似性** $P(R_i) = true, i = 1,2,\ldots,n$。同一子区域的像素应具有相同或相近的特性

图像分割基础

- 图像分割的定义
  - ✓ **互斥性**$P(R_i \cup R_j) = false, i \neq j$。不同子区域应具有不同的特性
  - ✓ **连通性**。$R_i, \forall i$是一个连通的区域，即同一子区域的像素点是连通的

通常，对于一幅图像，按照背景区域(background)和前景区域(foreground)进行划分和分割。前景通常是指目标物，可以有多个目标物区域。

图像为什么可以分割？需满足的条件

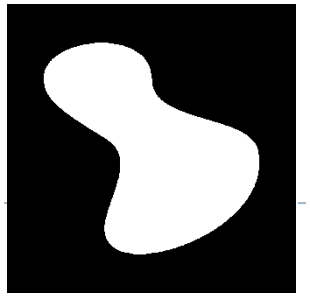
图像分割基础

- 图像分割的依据
  - ✓ 图像局部特征的相似性和互斥性
  - ✓ 每个子区域的自相似性，而相邻子区域的互斥性

对于一幅灰度图像，分割的依据是区域内的像素灰度值相似性和区域间的灰度突变性(不连续性)

对于分割算法，多数基于灰度的两个性质：不连续性和相似性。

- ✓ 不连续性：以灰度突变为基础，比如目标物边缘
- ✓ 相似性：以预定义的相似准则，将一幅图像分割为相似区域。



图像分割算法

- 边缘检测法

边缘点检测

边缘是图像分割的重要特征，通过边缘点检测，将边缘点连接成边缘线，围成的区域就是分割结果。

- 边缘点检测基本原理

思想：确定图像中是否有无边缘点？可利用某个检测算子，对图像的每一个像素进行检测，确定该像素点是否为边缘点。

在一幅图像中，边缘有方向和幅度两个特性，一般沿着边缘走向的灰度值缓慢变化或不变，而垂直于边缘走向的灰度则突变

图像分割算法

- 边缘检测法

边缘点检测

对一幅图像进行平均模板平滑时，类似于积分。而对于灰度的局部突变，采用微分检测十分合理，一阶微分和二阶微分通常较为合适。

以阶跃式的边缘为例：

- ✓ **一阶微分**：边缘的一阶导数在图像由暗变亮的突变位置，出现正峰值；相反，在图像由亮变暗的突变位置，出现负峰值。也就是说，可以用一阶导数的幅度值来检测边缘的存在。（对于灰度不变的位置，其导数应为0）

注：对于离散函数，一阶导数相当于差分

$$\frac{\partial f}{\partial x} = f'(x) = f(x+1) - f(x)$$

(考虑f(x+ △x)在x展开泰勒级数，并令△x=1)

图像分割算法

- 边缘检测法
  - 边缘点检测
    - ✓ 二阶微分

在图像由暗变亮的突变位置，边缘的二阶导数出现一个由正脉冲到负脉冲过0点；相反，在图像由亮变暗的突变位置，边缘的二阶导数出现一个由负脉冲到正脉冲过0点。

因此，二阶导数过0点也可以用于检测边缘的存在

注：$\frac{\partial^2 f}{\partial x^2} = f''(x) = f'(x+1) - f'(x) = f(x+1) - f(x) - [f(x) - f(x-1)] = f(x+1) + f(x-1) - 2f(x)$

图像分割算法

- 边缘检测法

边缘点检测

在图像处理中，一阶导数是通过梯度实现，利用一阶导数检测边缘点的方法称为梯度算子法。

问题：如何计算图像f中(x,y)位置的边缘强度及方向？

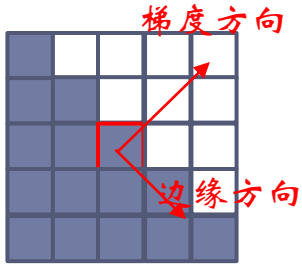
分析：采用梯度。

假定梯度采用∇f来表示，那么可以定义为$\nabla f(x,y)=\begin{bmatrix}g_x\\g_y\end{bmatrix}=\begin{bmatrix}\frac{\partial f}{\partial x}\\\frac{\partial f}{\partial y}\end{bmatrix}$

几何性质：表明了图像f在位置(x,y)处的最大变化率方向。

向量的大小(长度)M(x,y)和方向α(x,y)分别表示为

$$M(x,y)=\sqrt{g_x^2+g_y^2},\qquad \alpha(x,y)=arctan\left(\frac{g_y}{g_x}\right)$$



梯度方向与边缘方向相互正交。

图像分割算法

- 边缘检测法
  - 边缘点检测

在图像处理中，一阶导数是通过梯度实现，利用一阶导数检测边缘点的方法称为梯度算子法。

梯度实现时，有正交梯度(正交模板)和方向梯度。

✓ 正交梯度(水平和垂直两个方向)

在图像f(m,n)中，计算水平和垂直方向的梯度，

$$\begin{cases} G_h(m,n) = f(m,n) - f(m,n-1) \\ G_v(m,n) = f(m,n) - f(m-1,n) \end{cases}$$

那么水平和垂直方向的梯度模板

$$W_h = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, W_v = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

图像分割算法

- 边缘检测法
  - 边缘点检测
    - ✓ 正交梯度

利用梯度模板，与图像求卷积，可获得水平和垂直方向的梯度$G_h(m,n)$和$G_v(m,n)$。

检测点(m,n)处的梯度的幅度通常表示为以下三种

$$\begin{cases} G(m,n) = \left(G_h^2(m,n) + G_v^2(m,n)\right)^{0.5} \\ G(m,n) = |G_h(m,n)| + |G_v(m,n)| \\ G(m,n) = max\{|G_h(m,n)|, |G_v(m,n)|\} \end{cases}$$

梯度的幅度即为边缘的强度。

图像分割算法

- 边缘检测法
  - 边缘点检测
    - ✓ 正交梯度

当计算出(m,n)点的梯度幅度后，如何判断该点是否为边缘点？

首先，得出梯度图像$G(m,n)$，然后选取适当的阈值T，

$$B(m,n)=\begin{cases}1, G(m,n)\geq T\\ 0, otherwise\end{cases}$$

满足大于T的所有的(m,n)即为边缘点。

注：原图像(m,n)处的像素值被该点的梯度幅度取代后，即为梯度图像(考查)

图像分割算法

- 边缘检测法
- 边缘点检测
- ✓ 正交梯度

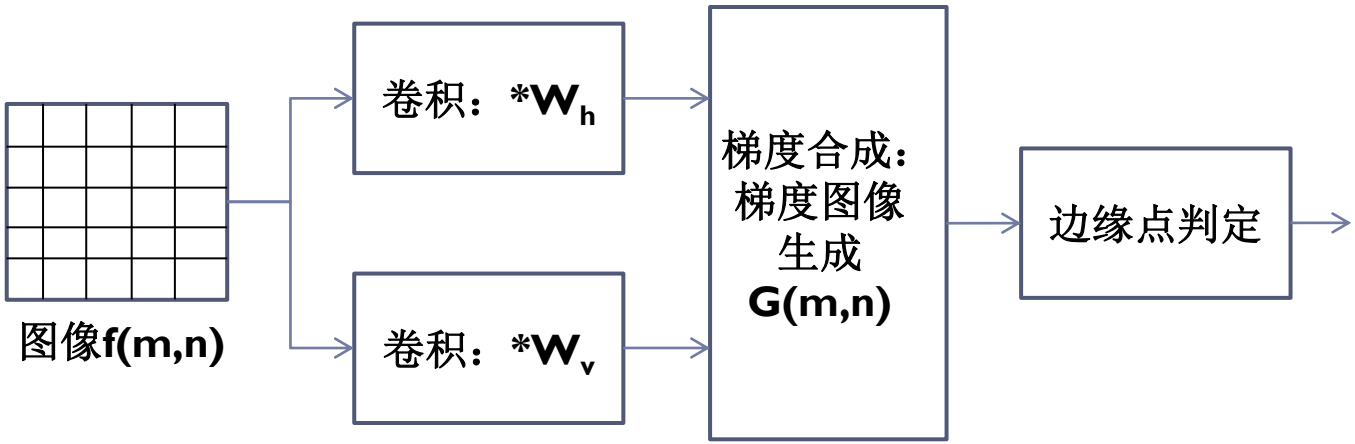
正交梯度法检测边缘点的过程,包括三个步骤:

- 利用水平和垂直梯度模板，计算每个像素点的水平和垂直方向梯度$G_h(m,n)$和$G_v(m,n)$
- 梯度合成，即梯度幅度计算，获得梯度图像
- 利用阈值对梯度图像进行边缘点的判断，获得边缘点位置

图像分割算法

- 边缘检测法
- 边缘点检测
- ✓ 正交梯度

将以上三个步骤，画成流程图：



其他梯度算子：Sobel算子，Roberts算子，Prewitt算子

图像分割算法

- 边缘检测法
- 边缘点检测
- ✓ 正交梯度
  - 一般的正交梯度算子(水平和垂直方向)

$$W_h = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, W_v = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

  - Roberts梯度算子(四点差分法),对噪声敏感

$$W_h = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, W_v = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

图像分割算法

- 边缘检测法
  - 边缘点检测
    - ✓ 正交梯度
      - Prewitt梯度算子(平均差分法)，抗噪声，但边缘模糊

$$W_h = \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, W_v = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- Sobel梯度算法(加权平均差分法)，抗噪声，边缘稍好

$$W_h = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, W_v = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

图像分割算法

- 边缘检测法
- 边缘点检测
- ✓ 方向梯度法

正交梯度算子是在两个正交(垂直)方向上计算梯度，通常沿水平x轴和垂直y轴方向进行，梯度模板分别仅对像素的水平突变和垂直突变敏感。

在实际应用中，还有其他除了水平和垂直以外的方向走向，如图所示

该图像表示在水平和垂直方向无突变

注：正交梯度法不能体现边缘方向，仅体现梯度的幅度。

20	30	200
30	30	30
50	30	100

图像分割算法

- 边缘检测法
- 边缘点检测
- ✓ 方向梯度法

如果不知道边缘的方向，那么需要确定边缘方向。因此，可以尝试一系列的不同方向边缘的方向梯度模板集，其中每一个方向的梯度模板仅对该特定方向的突变敏感。

利用每个方向的模板$\mathbf{W}_i$分别与图像卷积,即$F(m,n)*\mathbf{W}_i$，其最大模值就是边缘点的强度(即边缘梯度)，对应的模板方向就是边缘点方向。

边缘梯度值(梯度幅度)计算：

$$G(m,n)=\max_{i=1,\dots N}\{|F(m,n)*\mathbf{W}_i|\}$$

20	30	200
30	30	30
50	30	100

图像分割算法

- 边缘检测法
  - 边缘点检测
    - ✓ 方向梯度法

对于$G(m,n)=\max_{i=1,\ldots N}\{|F(m,n)*\mathbf{W}_i|\}$，G为获得的梯度图像，利用阈值判定，可得到边缘点及其方向。

与正交梯度的区别：

其计算梯度幅度时，采用了两个方向(垂直和水平)的模。即

$$\begin{cases} G(m,n)=\left(G_h^2(m,n)+G_v^2(m,n)\right)^{0.5} \\ G(m,n)=|G_h(m,n)|+|G_v(m,n)| \\ G(m,n)=max\{|G_h(m,n)|,|G_v(m,n)|\} \end{cases}$$

实际上，正交梯度是方向梯度法的特例。

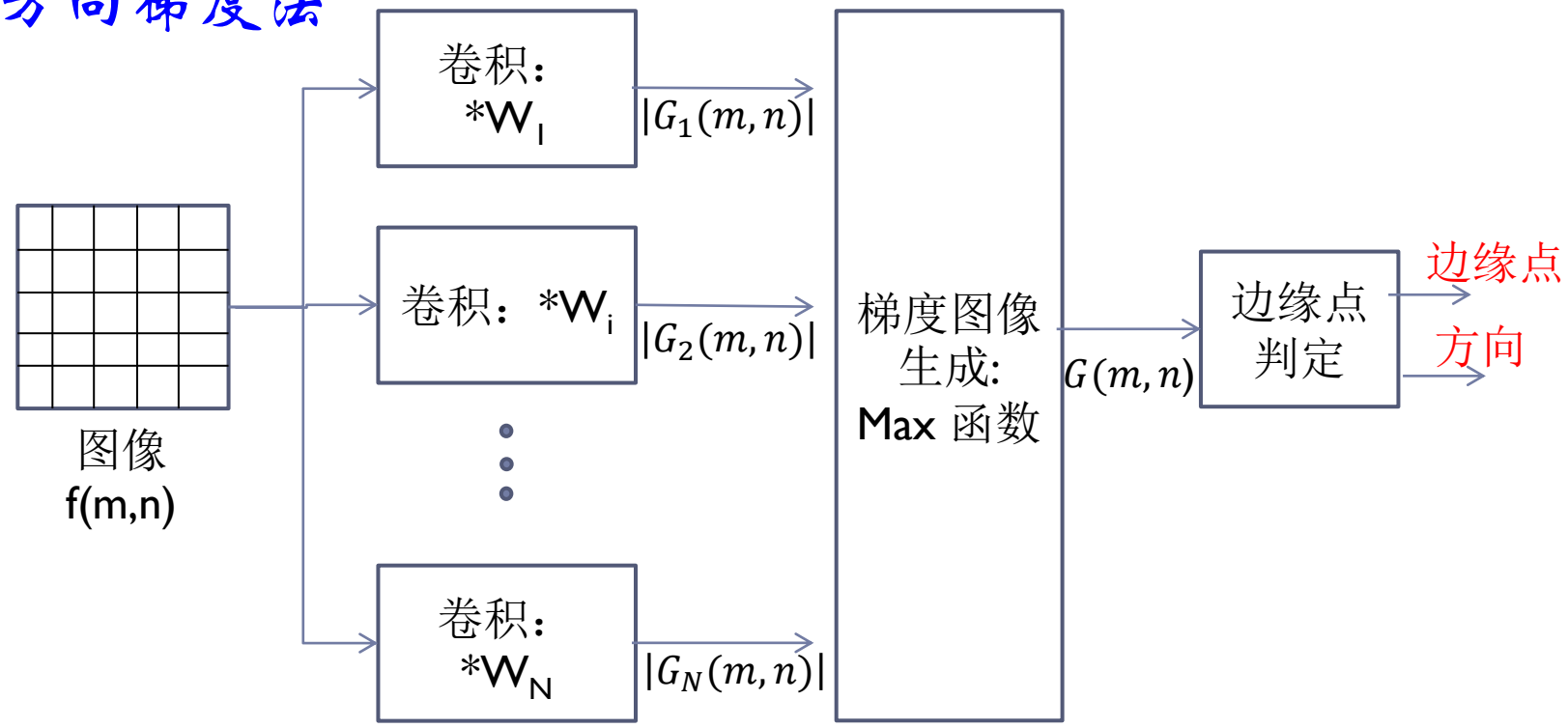
即当方向模板集中只选择垂直和水平方向时，并

$G(m,n)=max\{|G_h(m,n)|,|G_v(m,n)|\}$

20	30	200
30	30	30
50	30	100

图像分割算法

- 边缘检测法
- 边缘点检测
- ✓ 方向梯度法



图像分割算法

- 边缘检测法
  - 边缘点检测
    - ✓ 方向梯度法

对于8方向模板集，包括东、南、西、北、东南、西南、东北、西北。

以平均差分算子(Prewitt算子)为例，$W=\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}$，比例因子为1/3

灰度突变方向描述：

东方向：表示灰度从西向东突变 $\begin{bmatrix}-1 & 0 & 1\\ -1 & 0 & 1\\ -1 & 0 & 1\end{bmatrix}$

西方向：表示灰度从东向西突变 $\begin{bmatrix}1 & 0 & -1\\ 1 & 0 & -1\\ 1 & 0 & -1\end{bmatrix}$

经45°顺时针旋转后，为东南方向梯度模板，每次旋转45°，会得到8个方向梯度模板

图像分割算法

- 边缘检测法
  - 边缘点检测
  - ✓ 方向梯度法

东南方向：表示灰度从西北向东南突变 $\begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

。 。 。

西南方向：表示灰度从东北向西南突变 $\begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$

图像分割算法

- 边缘检测法
- 边缘点检测
- ✓ 方向梯度法

例：对一幅5x5大小的二值图像，如下所示，

1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1

分别利用一般正交梯度算子、Sobel梯度算子,Roberts梯度算子,和Prewitt梯度算子，(1)计算该图的梯度，(2)求出梯度幅度图(给出像素值)，(3)利用阈值法，给出检测出的边缘点(给出判定的二值化结果)

图像分割算法

- 边缘检测法
  - 边缘点检测
    - ✓ 方向梯度法

3x3的8方向模板可用于检测45°增量下的边缘，若有方向更细，即减小增量角度，比如30°，一般需要增大模板尺寸，合理构造模板内系数。

Nevatia和Babu提出了5x5的12方向模板，用于检测30°增量下的边缘点和方向。

不管多少个方向梯度模板，边缘点的检测方法与3x3的8方向梯度方法相同。

图像分割算法

- 边缘检测法
- 边缘点检测
- ✓ 方向梯度法

如何利用二阶导数进行边缘点检测？

方法同一阶类似，先求二阶差分，获得二阶梯度，根据二阶梯度的模$\left|\nabla^2 f(x,y)\right|$，利用阈值判定法，获得边缘化的二值化结果，从而获得边缘点。

二阶导数计算：$\nabla^2 f(x,y)=\frac{\partial^2 f}{\partial x^2}+\frac{\partial^2 f}{\partial y^2}$

$$\frac{\partial^2 f}{\partial x^2}=f(x+1,y)+f(x-1,y)-2f(x,y)$$

$$\frac{\partial^2 f}{\partial y^2}=f(x,y+1)+f(x,y-1)-2f(x,y)$$

$$\nabla^2 f(x,y)=f(x+1,y)+f(x-1,y)+f(x,y+1)+f(x,y-1)-4f(x,y)$$

图像分割算法

- 采用8方向梯度法进行图像分割




原图　　T=50　　T=100　　T=200




原图　　T=50　　T=100　　T=200

图像分割算法

- 采用8方向梯度法进行图像分割

原图

T=50

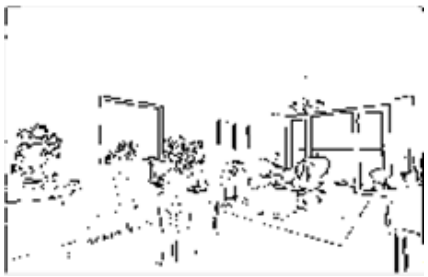
T=100

T=200

原图

T=50

T=100



T=200

图像分割算法

- 边缘检测法

边缘点的线检测法

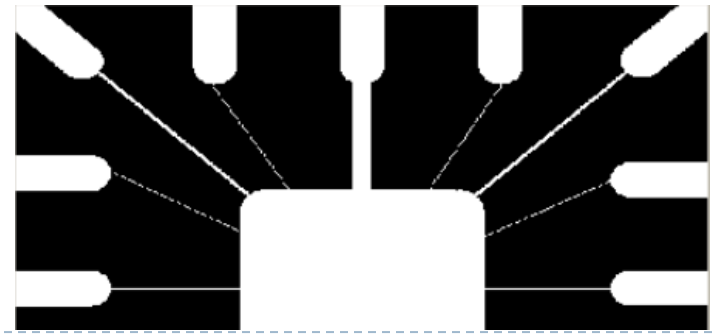
基本思想：利用方向梯度模板，可以设计检测不同方向线的方向模板，利用阈值判定其梯度幅度(梯度的模),完成线的检测，并确定方向。

与边缘点的检测原理类似。

线的梯度幅度定义

$$G(m,n) = \max_{i=0,\ldots,N-1} \{|F(m,n) * \mathbf{W}_i|\}$$

其中，W_i是线检测模板。



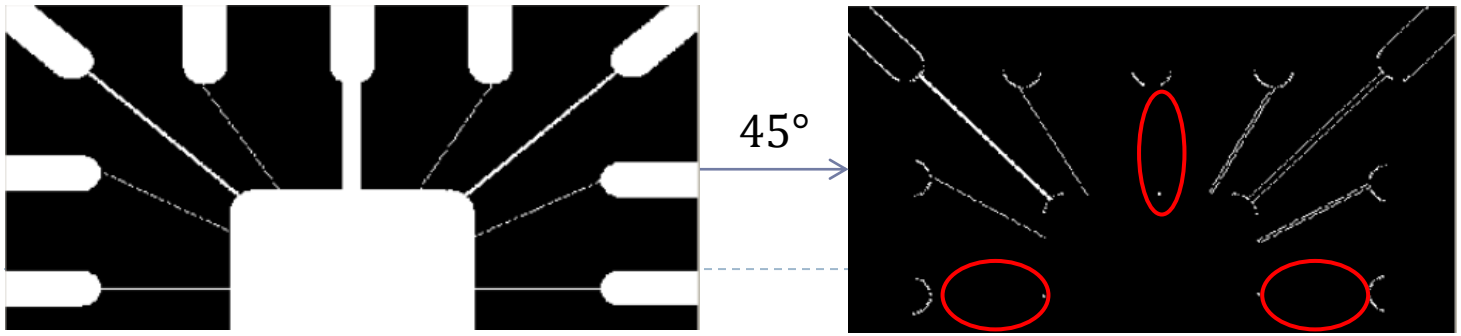
图像分割算法

- 边缘检测法
  - 边缘点的线检测法

4方向的3x3线检测模板：

$$0°：W_0=\frac{1}{6}\begin{bmatrix}-1&-1&-1\\2&2&2\\-1&-1&-1\end{bmatrix}；45°：W_1=\frac{1}{6}\begin{bmatrix}-1&-1&2\\-1&2&-1\\2&-1&-1\end{bmatrix}$$

$$90°：W_2=\frac{1}{6}\begin{bmatrix}-1&2&-1\\-1&2&-1\\-1&2&-1\end{bmatrix}；-45°：W_3=\frac{1}{6}\begin{bmatrix}2&-1&-1\\-1&2&-1\\-1&-1&2\end{bmatrix}$$



图像分割算法

- 边缘检测法
- 边缘点的线检测法(二阶导数算子法)
- ✓ Laplacian算子法

连续图像f(x,y)的Laplacian边缘检测算子定义为

$$G(x,y) = -\nabla^2 f(x,y)$$

其中，$\nabla^2 f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$

对于数字图像f(m,n),用差分代替二阶偏导，Laplacian边缘检测算子可写为？

图像分割算法

- 边缘检测法

边缘点的线检测法(二阶导数算子法)

✓ Laplacian算子法

$$
\begin{aligned}
&G(m,n)\\
&=-\{f(m+1,n)+f(m-1,n)+f(m,n+1)\\
&+f(m,n-1)-4f(m,n)\}\\
&=4f(m,n)-f(m+1,n)-f(m-1,n)-f(m,n+1)\\
&-f(m,n-1)
\end{aligned}
$$

因此，Laplacian边缘检测模板(四邻域模板)

$$
W=\begin{bmatrix}0 & -1 & 0\\ -1 & 4 & -1\\ 0 & -1 & 0\end{bmatrix}
$$

(仅计算水平和垂直方向上的梯度)

图像分割算法

- 边缘检测法
  - 边缘点的线检测法(二阶导数算子法)
    - ✓ Laplacian算子法

Laplacian边缘检测模板 (八邻域模板)

$$W = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

除了水平和垂直方向上的梯度，对角线方向的梯度也用于检测。

Laplacian检测模板的特点是各向同性，对孤立点的检测效果较好，但对高斯噪声敏感,丢失边缘信息，会把噪声当成边缘点检测出来。

孤立点与噪声的区别？可以先去噪声，再利用Laplacian模板？

导数对噪声的敏感性特点。

图像分割算法

- 边缘检测法

边缘点的线检测法(二阶导数算子法)

✓ LoG算子法(Marr and Hildreth)

LoG(Laplacian of Gaussian)高斯-拉普拉斯算子，简称LoG算子。

方法思想：先采用Gaussian算子对原图像进行平滑，去掉噪声，然后再利用Laplacian算子进行边缘检测。克服了Laplacian算子对噪声敏感的特点。

图像分割算法

- 边缘检测法
- 边缘点的线检测法(二阶导数算子法)
  - ✓ LoG算子法

已知二维高斯函数$h(x,y)=exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$

则，高斯去噪过程即为卷积，

$$z(x,y)=h(x,y)*f(x,y)$$

对去噪后的图像$z(x,y)$利用Laplacian边缘梯度算子，计算二阶梯度：

$$G(x,y)=-\nabla^2[z(x,y)]=-\nabla^2[h(x,y)*f(x,y)]$$
$$=[-\nabla^2 h(x,y)]*f(x,y)=H(x,y)*f(x,y)$$

图像分割算法

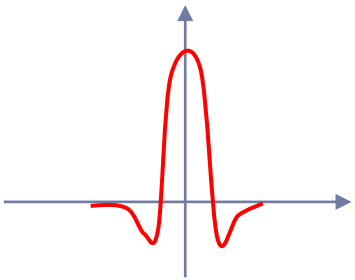
- 边缘检测法

边缘点的线检测法(二阶导数算子法)

✓LoG算子法

$H(x,y) = -\nabla^2 h(x,y)$为LoG算子(由Gaussian函数推出的Laplacian算子)。LoG算子的表达式为

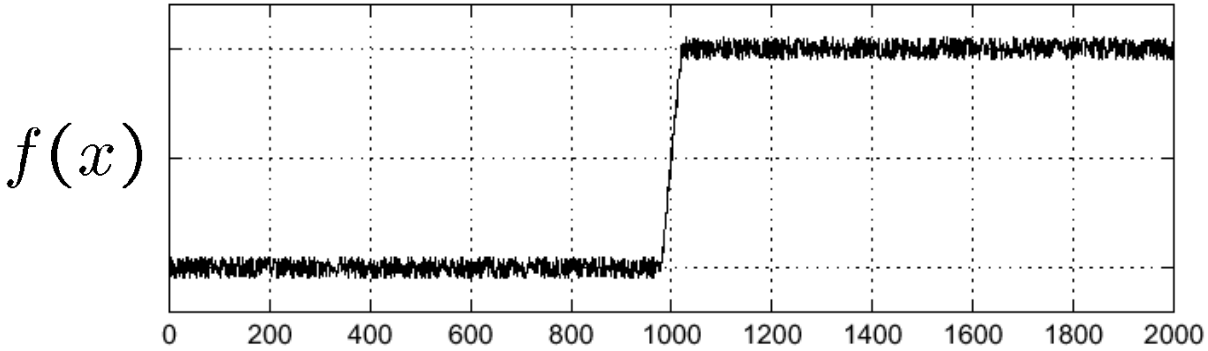
$$H(x,y) = -\nabla^2 h(x,y)$$
$$= \frac{2\sigma^2 - x^2 - y^2}{\sigma^4} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

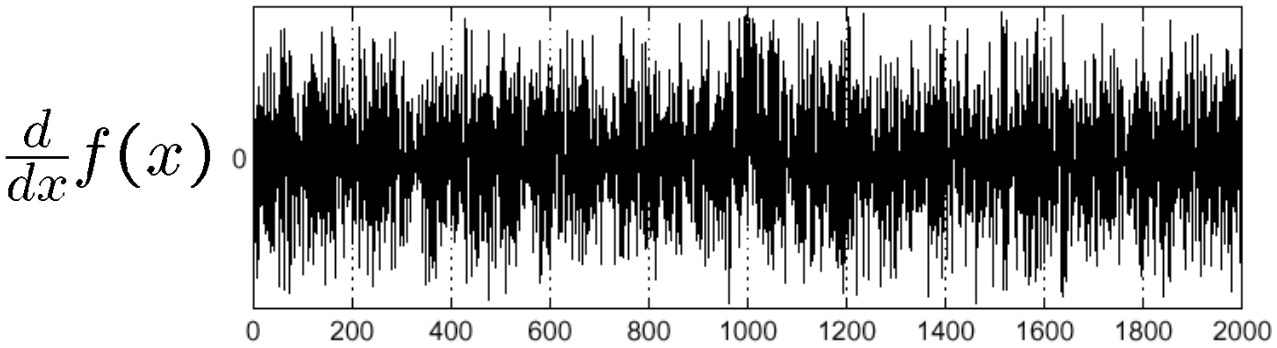


这种LoG算子能够有效降低噪声的影响，当边缘模糊或噪声较大时，依然能够提取边缘位置。

图像分割算法

- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal





Where is the edge?

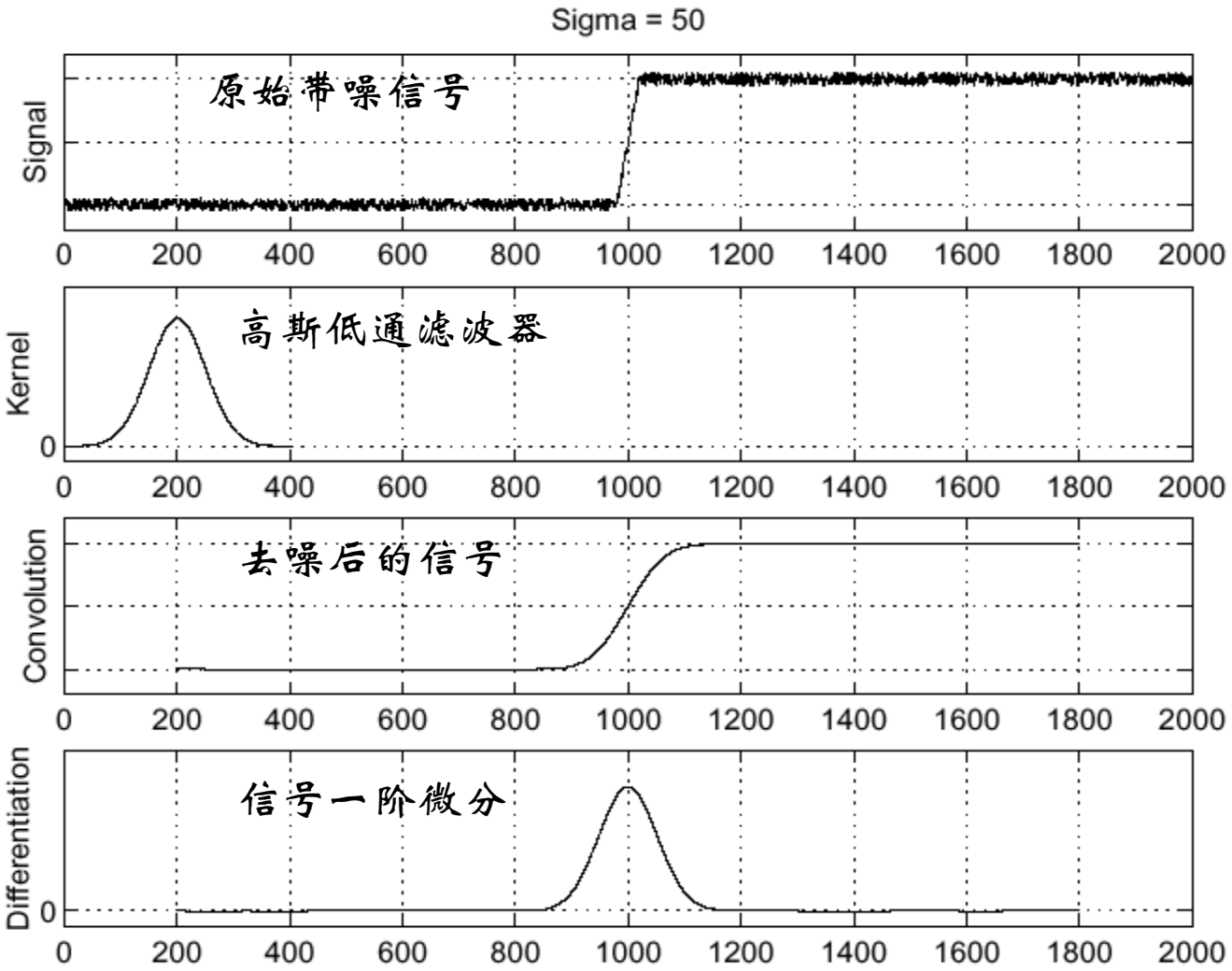
Solution: smooth first

f

h

$h \star f$

$\frac{\partial}{\partial x}(h \star f)$



Where is the edge? Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Associative property of convolution

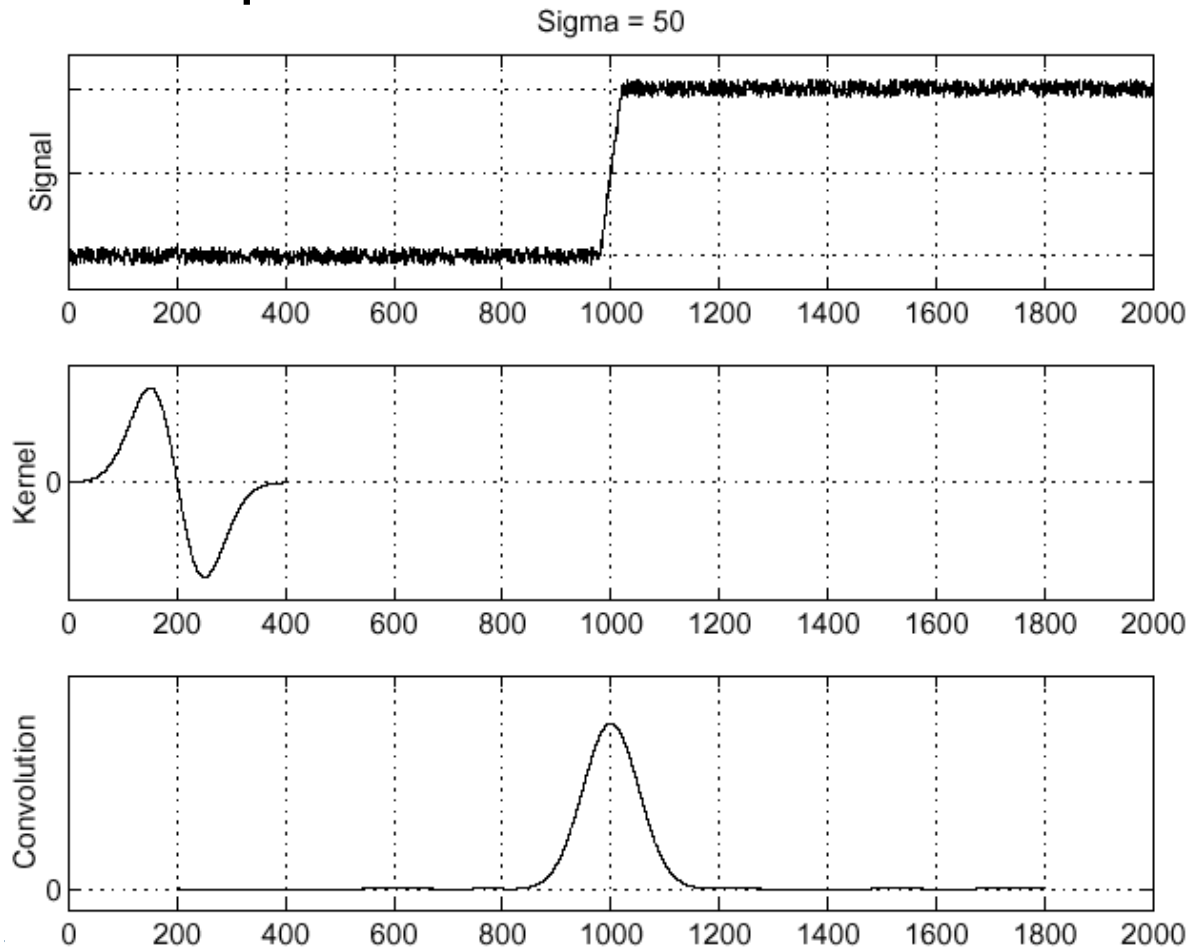
$$\frac{\partial}{\partial x}(h \star f) = (\frac{\partial}{\partial x}h) \star f$$

- This saves us one operation:

f

$\frac{\partial}{\partial x}h$

$(\frac{\partial}{\partial x}h) \star f$



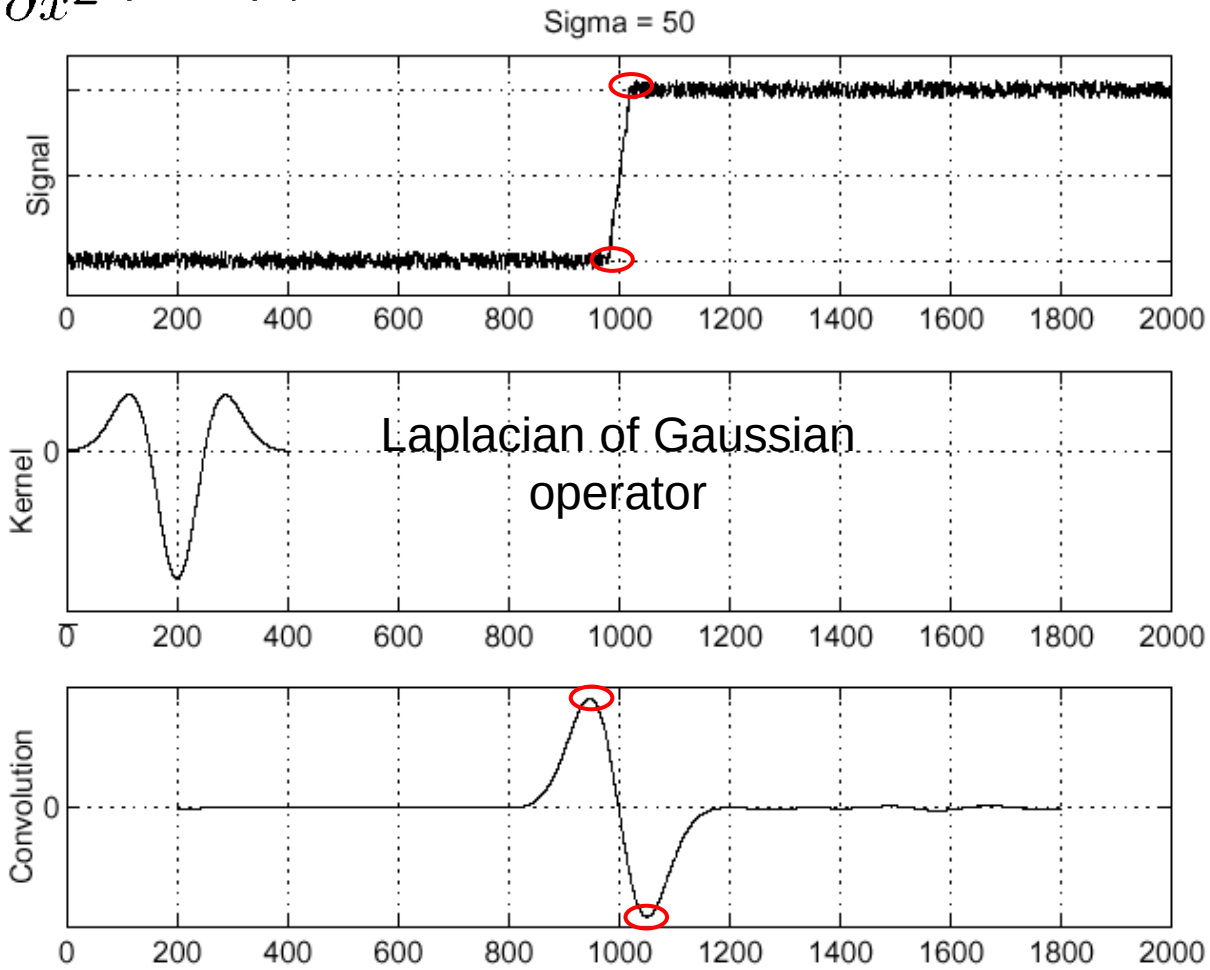
Laplacian of Gaussian

- Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

f

$\frac{\partial^2}{\partial x^2}h$

$(\frac{\partial^2}{\partial x^2}h) \star f$



- Where is the edge? Zero-crossings of bottom graph

图像分割算法

Canny edge detector [1986]，优秀的边缘检测算子：

关键步骤：

$$W_h = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, W_v = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

1. 高斯低通平滑图像I;
2. 计算图像I的梯度(一阶导数，如sobel算子)，获得梯度幅值图像g和梯度方向图像α;
3. 对梯度幅度图像采用NMS (non-maximum suppression)非极大值抑制 (边缘抑制，相同方向保最大);
4. 采用双阈值检测边缘 (强边缘，弱边缘。主要由噪声和颜色变化导致)。

Self-learning…..

图像分割算法

- 边缘线跟踪(边缘连接、边界检测)

问题：对于边缘点检测，是通过梯度算子、方向梯度算子、线检测模板、二阶导数算子、LoG算子，检测结果是满足阈值条件的离散点，比如边缘点、孤立点等。当连接边缘点时，会出现断裂(噪声点的影响)。

方法：如何将边缘点连接成边缘线？即边界。核心思想是如何判断两个边缘点是否可以连接？需要定义某种相似准则，进行局部处理。

图像分割算法

- 边缘线跟踪(边缘连接、边界检测)
  - 局部边缘连接法

思想：

以局部梯度算子处理后的梯度图像作为输入，对每个边缘点,进行**二次判定**，比如以3x3的小邻域，找出最大的梯度值，作为候选边缘点。再对每一个候选点，利用方向梯度的方法确定边缘方向。

然后进一步判定小邻域内的候选边缘点是否属于同一个边缘线，连接起来。

图像分割算法

- 边缘线跟踪(边缘连接、边界检测)
  - 局部边缘连接法

方法：

第一步：确定候选边缘点。在边缘点(m,n)的一个小邻域内(3x3,5x5),具有最大梯度值的点被称为候选点。并对每个候选点，利用方向梯度模板，确定其边缘方向。

第二步：对相邻的候选边缘点(m,n)和(i,j)，根据事先确定的相似准则，(即两个候选点的梯度和方向的差值小于某个阈值)，判定是否连接。该相似准则定义为

$$\begin{cases} |G_1(m,n) - G_2(i,j)| \le E \\ |\phi_1(m,n) - \phi_2(i,j)| \le A \end{cases}$$

其中，$G_1(m,n)$，$G_2(i,j)$表示两个边缘点梯度的模；
$\phi_1(m,n)$，$\phi_2(i,j)$表示两个边缘点的方向值。

图像分割算法

- 边缘线跟踪(边缘连接、边界检测)
- 光栅扫描跟踪法(顺序扫描跟踪)

思想：通过设定检测门限和跟踪门限，对一幅图像，进行顺序扫描，高于检测门限的像素位置赋值为1，然后对该位置的下一行的三个近邻像素进行跟踪判决，若高于跟踪门限，则赋值为1。依次对每个像素进行扫描。最终获得一个二值化图像。

图像分割算法

- 边缘线跟踪(边缘连接、边界检测)
- 光栅扫描跟踪法(顺序扫描跟踪)

方法：

第一步：设定两个门限：检测门限d和跟踪门限t (d>t)；

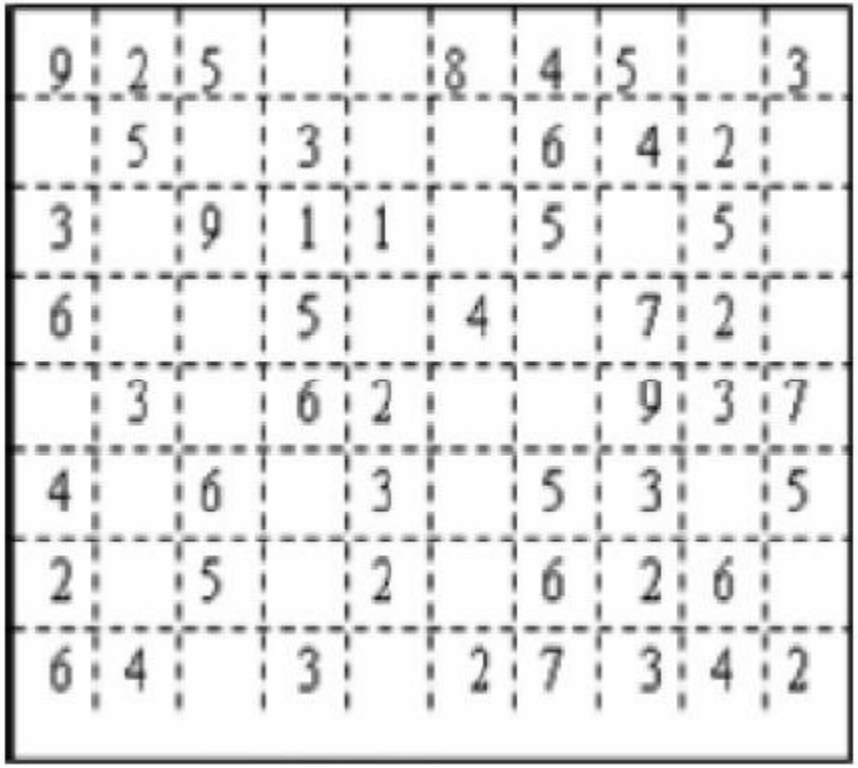
第二步：扫描整幅图像，像素值达到检测门限d的位置，标记为1，作为下一步的跟踪起点；

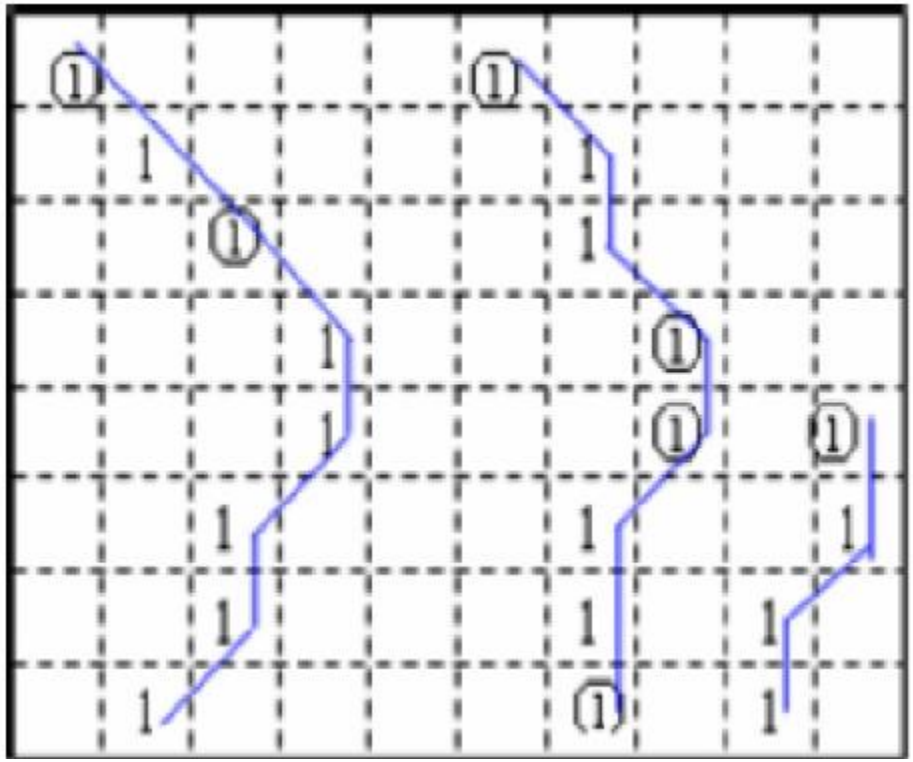
第三步：从第m行上首次被标记为1的点(m,n)开始，在其下一行(m+1,n-1), (m+1,n), (m+1,n+1)点上进行跟踪判决，只要这三个点的灰度值达到跟踪门限，则也被标记为1。

直到所有标记为1的点被扫描完成，跟踪结束。

图像分割算法

- d=7, t=4





图像分割算法

- Hough变换法

对于边缘检测：首先，边缘点的检测；其次，将边缘点连接成边缘线。

由于噪声(光照等)的干扰，很难检测到一组完整的描述边缘线的点集。Hough变换则能根据待检测曲线对应的像素间的整体关系，检测出边缘线，并能够进行参数方程描述。

优点：抗噪声、抗断点，是将边缘点连成边缘线的全局最优方法，而非局部边缘连接方法。

图像分割算法

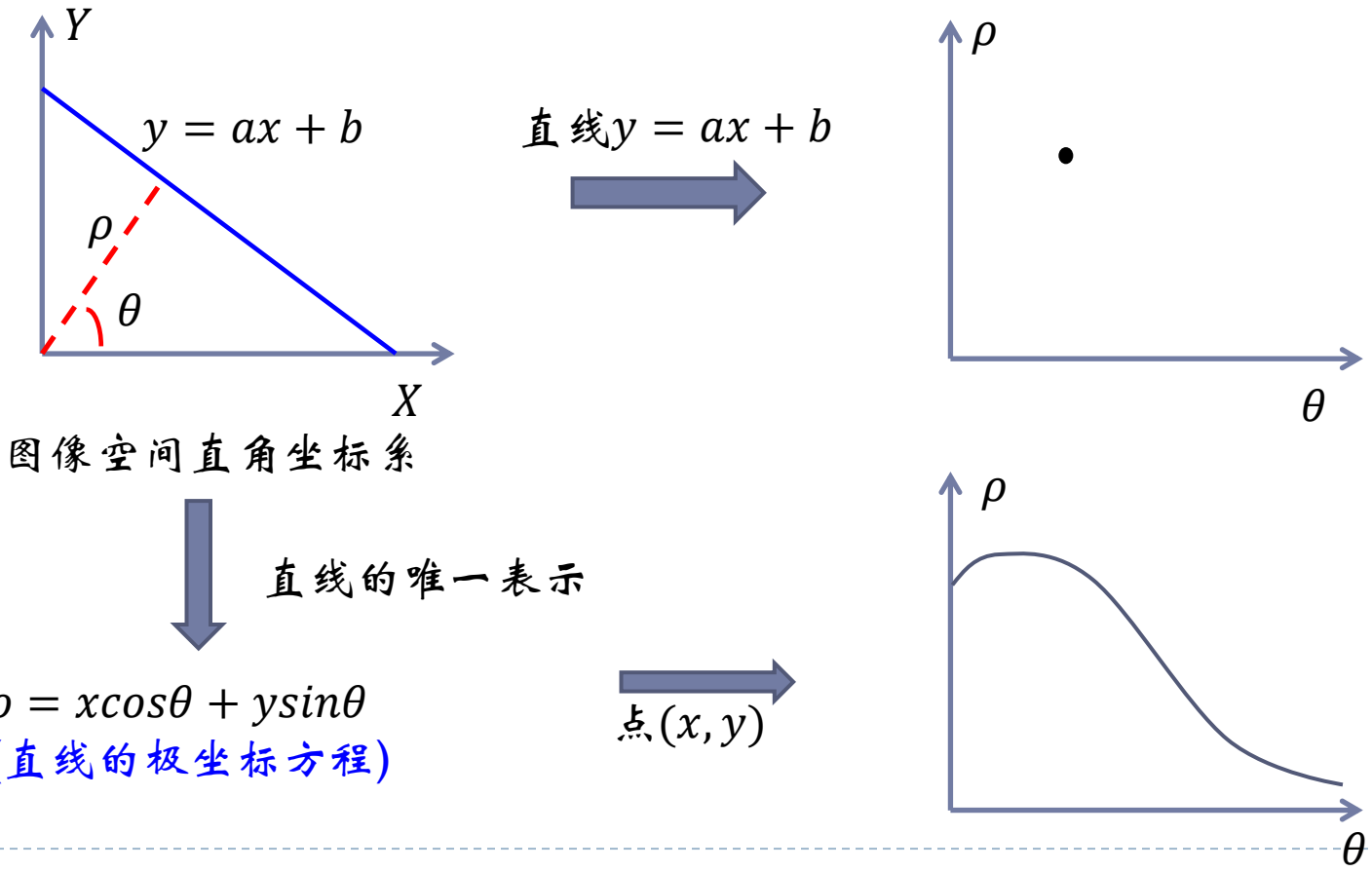
- Hough变换法

背景：已知检测出的n个边缘点，目的是找到位于同一条直线上的点集。最直接的方式，是利用两点一线，对n个点，组成的n(n-1)/2条直线，分别计算每条直线上的共线点，那么共线点较多的直线，即为要找的直线。但要对如此多的直线进行分别判定，运算量极大，实际无法满足。

Hough思考：坐标空间变换？从图像空间中的直角坐标系中某直线y=ax+b，变换到该直线的极坐标系。

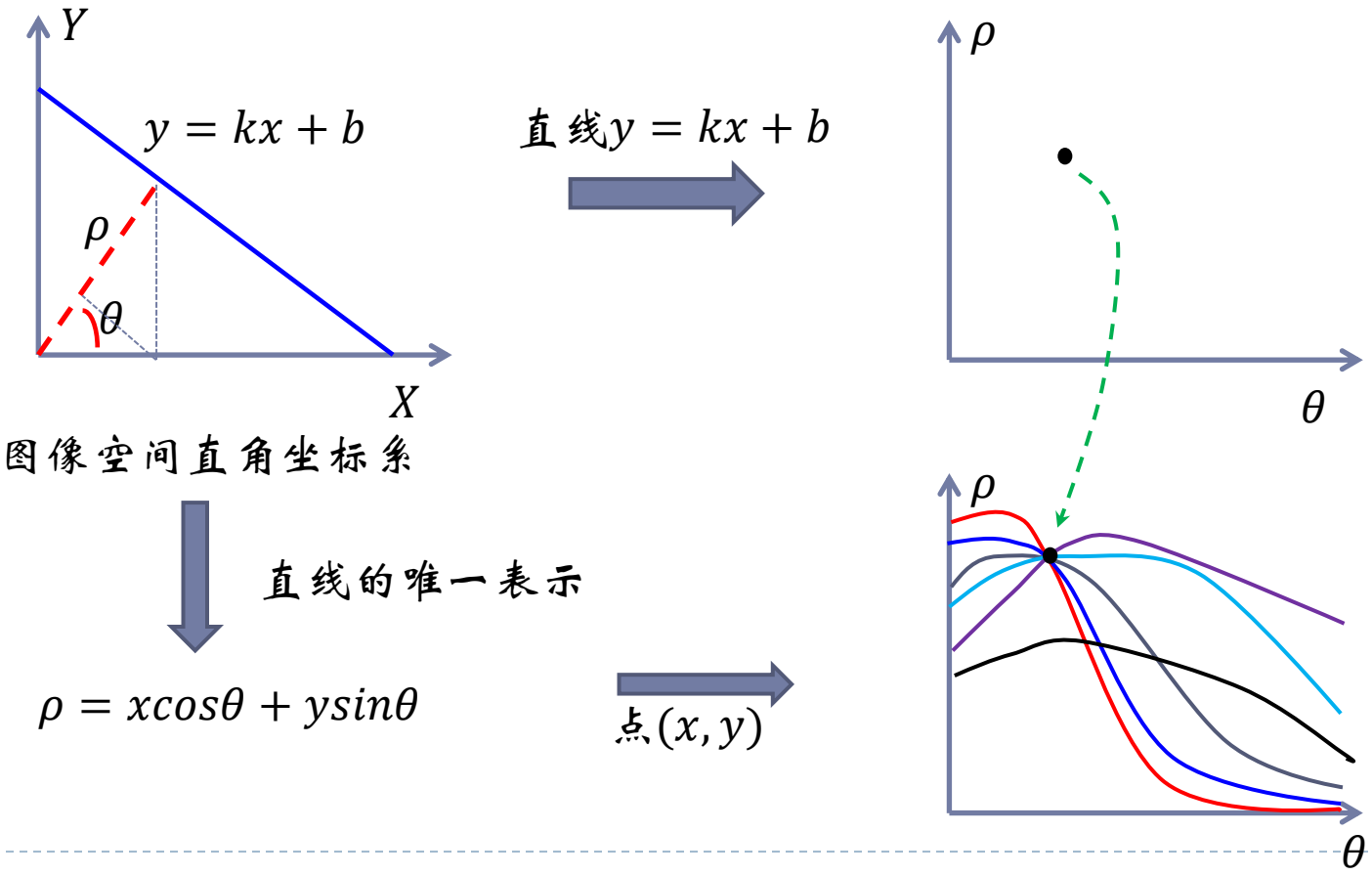
图像分割算法

- Hough变换法



图像分割算法

- Hough变换法



图像分割算法

- Hough变换法

观察：以(x,y)为坐标的图像空间(直角坐标系)和以(ρ,θ)为坐标的参数空间(极坐标系)，有如下对应关系：

- 图像空间中的一条直线$y = ax + b$，对应参数空间中的一个点(θ,ρ)
- 图像空间中的一个点(x,y)，映射为参数空间中的一条正弦曲线 $\rho = x\cos\theta + y\sin\theta$
- 图像空间中一条直线上的多个共线点，映射为参数空间相较于一点的多条正弦曲线。

图像分割算法

- Hough变换法

这种图像空间的点到参数空间的线的映射关系，称为Hough变换。

因此，寻找图像空间中共线点最多的直线，就相当于寻找参数空间中相交于一点的最多的正弦曲线。找到该点坐标(θ, ρ)，即可确定共线点的直线方程。

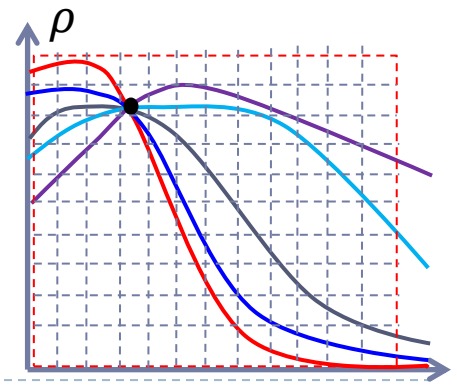
确定该交点的方法：网格法

图像分割算法

Hough变换法

网格法：

- 将参数空间按ρ和θ量化成多个小格。对每个边缘点(x_i, y_i)，都有一条正弦曲线$\rho = x_i cos\theta + y_i sin\theta$
- 将每个小格的θ,代入方程，计算出ρ,若ρ值落在划分好的小格时，该小格计数增加1
- 所有边缘点经以上两步计算完后，统计每个小格的计数值，若检测一条直线，则最大计数值对应的(ρ,θ)即为交点。从而得出图像空间中的直线方程，$y = \frac{-xcos\theta+\rho}{sin\theta}$



图像分割算法

- 灰度门限分割法

利用灰度直方图的波峰数量，设定一个或多个阈值，直接进行灰度的二值或多值判定。波峰的数量表示可分割的多个对象，包括目标或背景。(不同目标物之间的灰度值有明显差异)

方法：单阈值分割、多阈值分割、迭代阈值分割

1、阈值数量由直方图波峰数量确定

2、阈值选取，通常采用极小值点阈值（直方图包络谷值）

3、最优阈值法，以最小错分概率为准则(OTSU大律法，1979；类间方差最大准则，$T^* = \underset{T}{\operatorname{argmax}}\,\sigma_B^2(T)$。

$$\sigma_B^2(T) = P_1(T)(m_1 - m)^2 + P_2(T)(m_2 - m)^2。$$

注：$P_1(T)$和$P_2(T)$表示前景和背景的像素比例，m_1,m_2和m表示前景、背景和整幅图的平均灰度

图像分割算法

- 灰度门限分割法

一种迭代阈值法：通过不断更新阈值T，直到该阈值T不再变化，表示分割已完成。

- 找出图像中的最大和最小灰度值a和b,那么T=(a+b)/2
- 根据阈值将图像分成目标物和背景两部分，并计算两部分的像素数，以及各部分的平均灰度值c和d,那么更新T=(c+d)/2.
- 利用T重新分割目标物和背景两部分，并计算各部分的平均灰度值e和f,更新T=(e+f)/2，直到T不再变化。

$$g(x,y)=\begin{cases}1, f(x,y)>T\\ 0, f(x,y)\le T\end{cases}$$

图像分割算法

- 区域生长-分割法

基本思想：利用灰度相似性，根据像素的某种相似准则，进行像素区域扩张(生长)。

基本方法：从一组“种子”点开始，根据某种定义的准则，将种子的邻域像素添加进来，逐渐形成生长区域。

几个问题：

- 种子的选择：确定分割的区域数量，种子点为能正确代表每个区域灰度取值的像素点
- 生长方式：像素方式和区域方式
- 相似性准则
- 生长终止准则

图像分割算法

- 区域生长-分割法

简单生长(像素)

种子点(第一个生长点(s,t))按某种相似准则接受其邻域(4-邻域)的像素点(m,n)；接受后的像素点变成生长点，其像素值取为种子点的值。重复该过程，直到生长终止。

$$|f(m,n)-f(s,t)|<T$$

质心生长(像素+区域)

对生长点的4邻域进行判别时，是与生长区域像素的灰度平均值，进行比较，而非种子点的值。

$$\left|f(m,n)-\bar{f}(s,t)\right|<T$$

混合生长(区域)

是对生长区域之间的合并与否。比如对于两个区域，通过区域的平均灰度值进行判定。

$$\left|\bar{f}_i-\bar{f}_j\right|<T$$

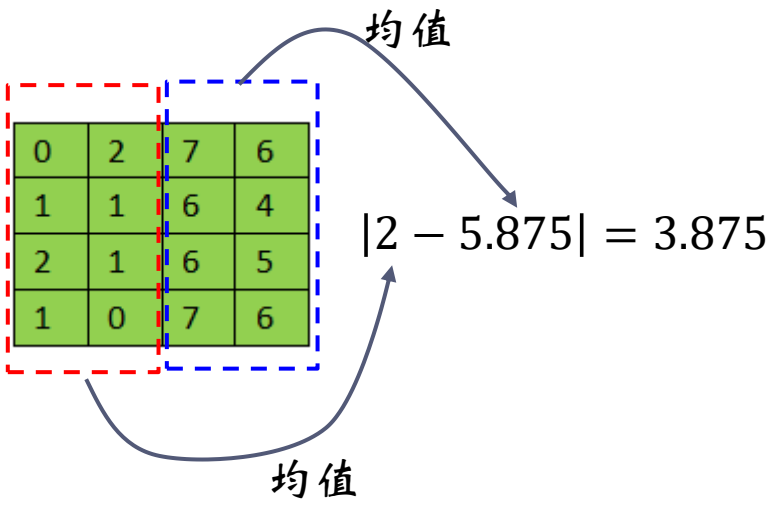
图像分割算法

- 区域生长-分割法
- 简单生长(像素)

0	2	7	6
1	1	6	4
2	1	6	5
1	0	7	6

1	1	6	6
1	1	6	6
1	1	6	6
1	1	6	6

- 混合生长(区域)



图像分割算法

- 区域生长-分割法
- 简单生长(像素)

0	2	7	6
1	1	6	4
2	1	6	5
1	0	7	6

1	1	6	6
1	1	6	6
1	1	6	6
1	1	6	6

- 质心生长(像素+区域)

0	2	7	6
1	1	6	4
2	1	6	5
1	0	7	6

- 混合生长(区域)

0	2	7	6
1	1	6	4
2	1	6	5
1	0	7	6

图像分割算法

- 区域分裂合并-分割法

对于不了解区域形状和区域数目时，分裂合并较为适用。该方法将图像分解成互不重叠的区域(分裂)，再按某种相似性准则进行合并。

四叉树分裂合并法实现：

1. 对一幅图像R，给定相似准则P，若图像中的任一个区域R_i，有$P(R_i) = false$，则将该区域等分成四个子区域。

0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

图像分割算法

- 区域分裂合并-分割法

四叉树分裂合并法实现：

2. 对相邻区域R_i, R_j，给定相似准则P，若$P(R_i \cup R_j) = true$，则合并这两个区域。

3. 当进一步分裂与合并不能进行时，分割结束。

0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

图像分割算法

- 区域分裂合并-分割法

四叉树分裂合并法实现：

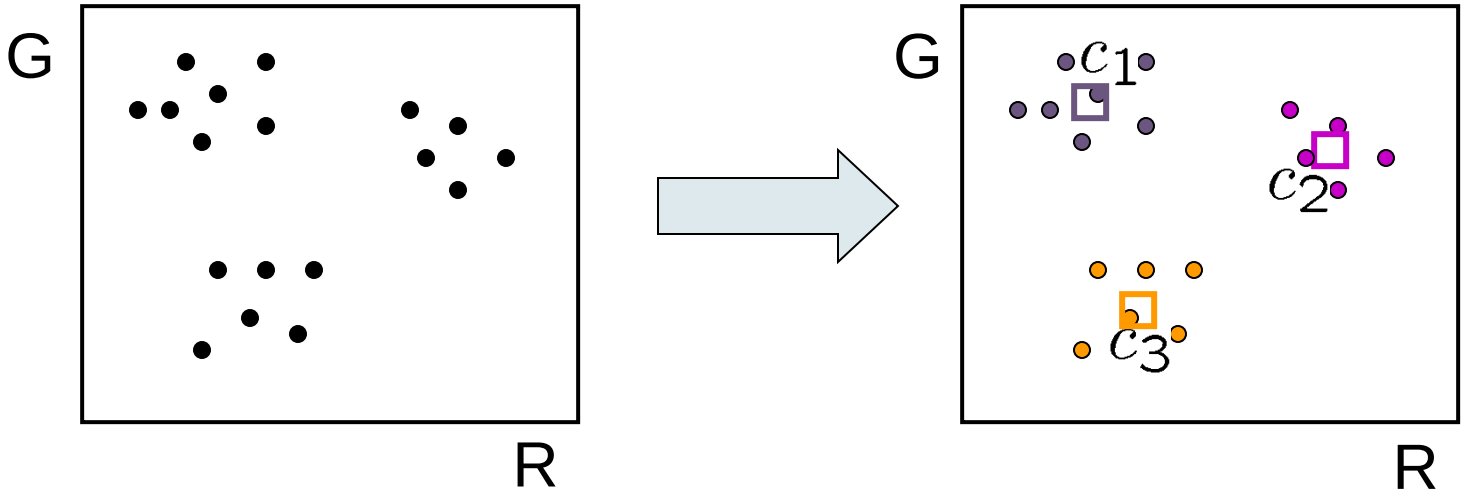
2. 对相邻区域R_i, R_j，给定相似准则P，若$P(R_i \cup R_j) = true$，则合并这两个区域。

3. 当进一步分裂与合并不能进行时，分割结束。

0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	1
0	0	0	0	1	1	1	1
0	0	0	0	1	1	1	1
0	0	1	1	1	1	1	1
0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Clustering

- How to choose the representative colors?
  - This is a clustering problem!



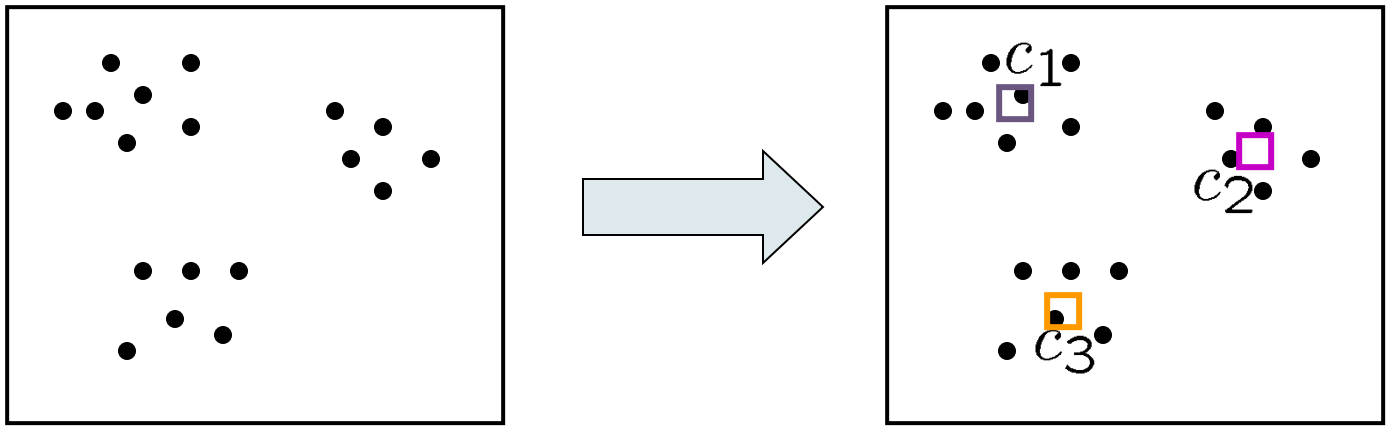
Objective

- Each point should be as close as possible to a cluster center
  - Minimize sum squared distance of each point to closest center

$$\sum_{\text{clusters } i} \quad \sum_{\text{points p in cluster } i} \|p - c_i\|^2$$

Break it down into subproblems

- Suppose I tell you the cluster centers c_i
  - Q: how to determine the points associated with each c_i?
    - A: for each point p, choose closest c_i



Suppose I tell you the points in each cluster

- Q: how to determine the cluster centers?
- A: computer the cluster center c_i as the mean of all points in the cluster

K-means clustering

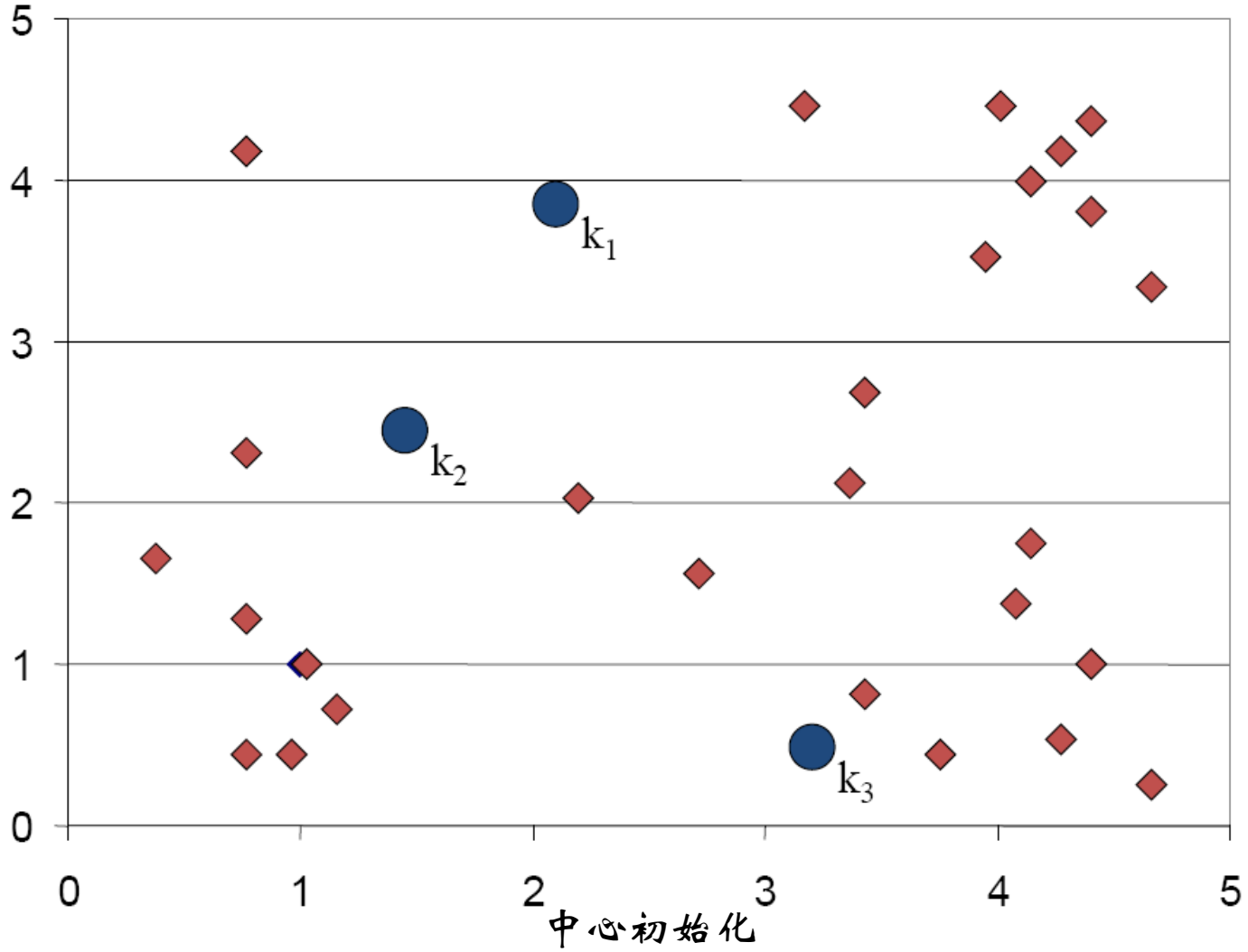
- K-means clustering algorithm
  1. Randomly initialize the cluster centers, $c_1, \ldots, c_K$
  2. Given cluster centers, determine points in each cluster
     - For each point p, find the closest c_i. Put p into cluster i
  3. Given points in each cluster, solve for c_i
     - Set c_i to be the mean of points in cluster i
  4. If c_i have changed, repeat Step 2

- Java demo: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

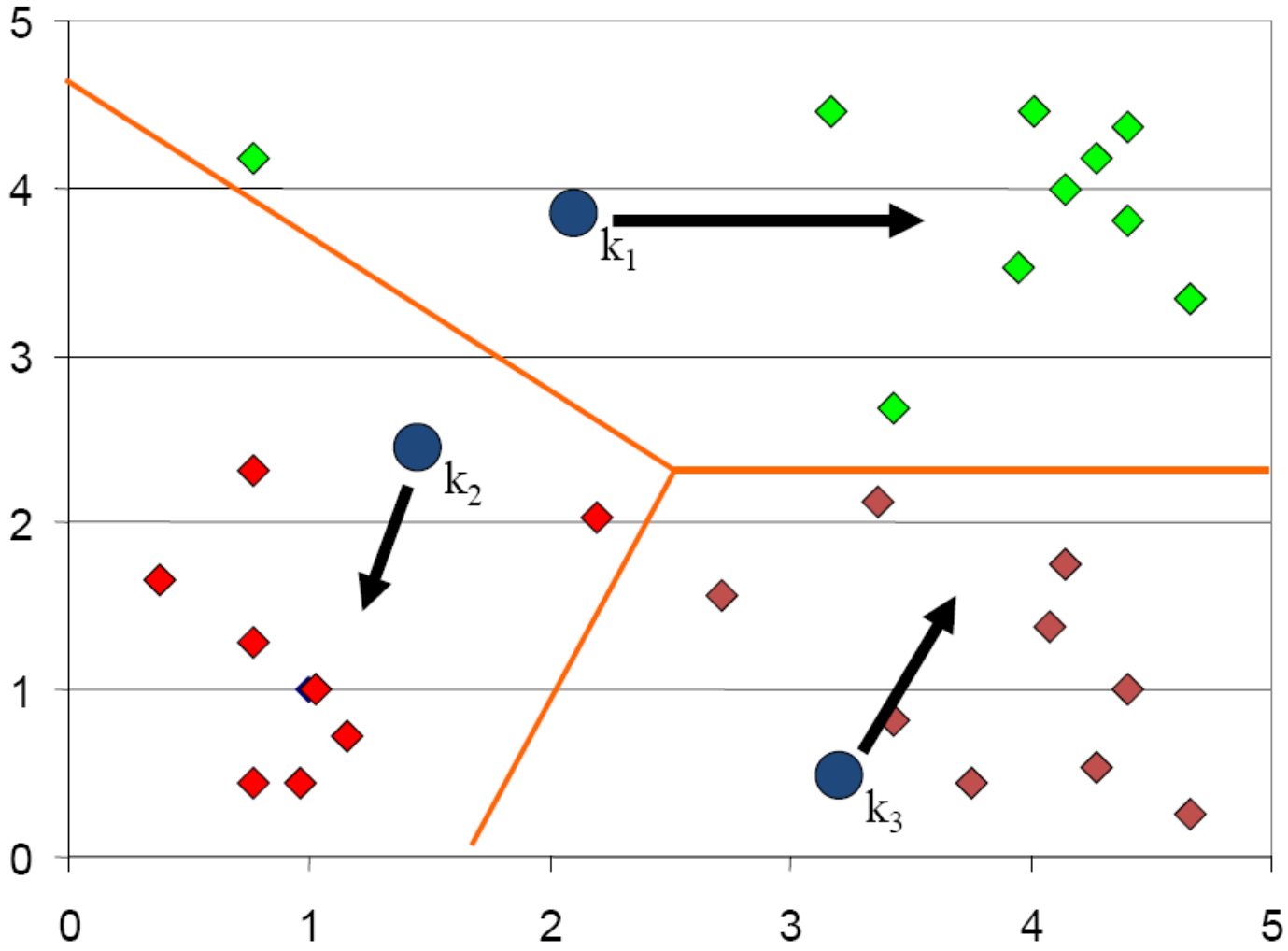
- Properties
  - Will always converge to *some* solution
  - Can be a "local minimum"
    - does not always find the global minimum of objective function:

$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} \|p - c_i\|^2$$

K-means clustering-Step 1

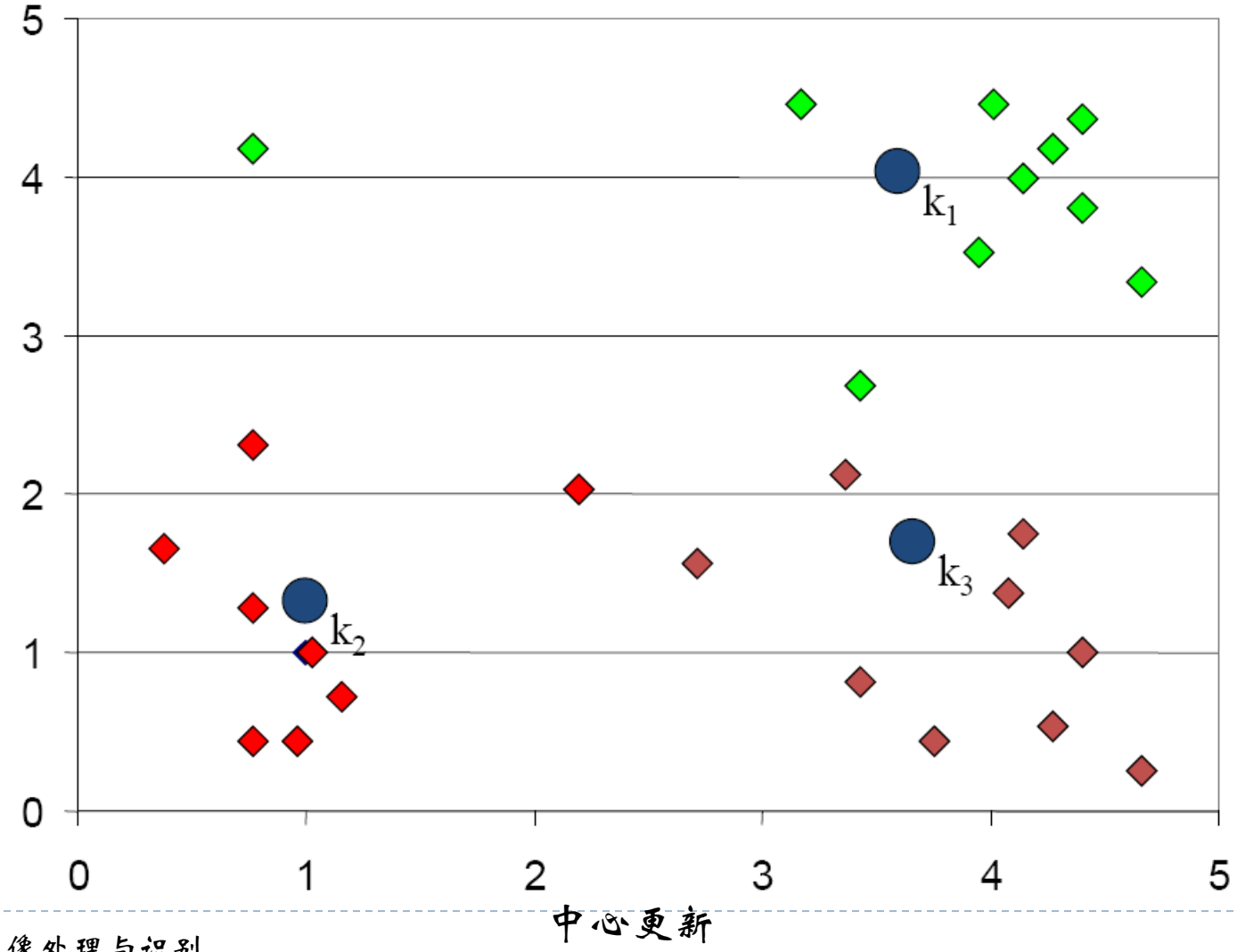


K-means clustering-Step 2

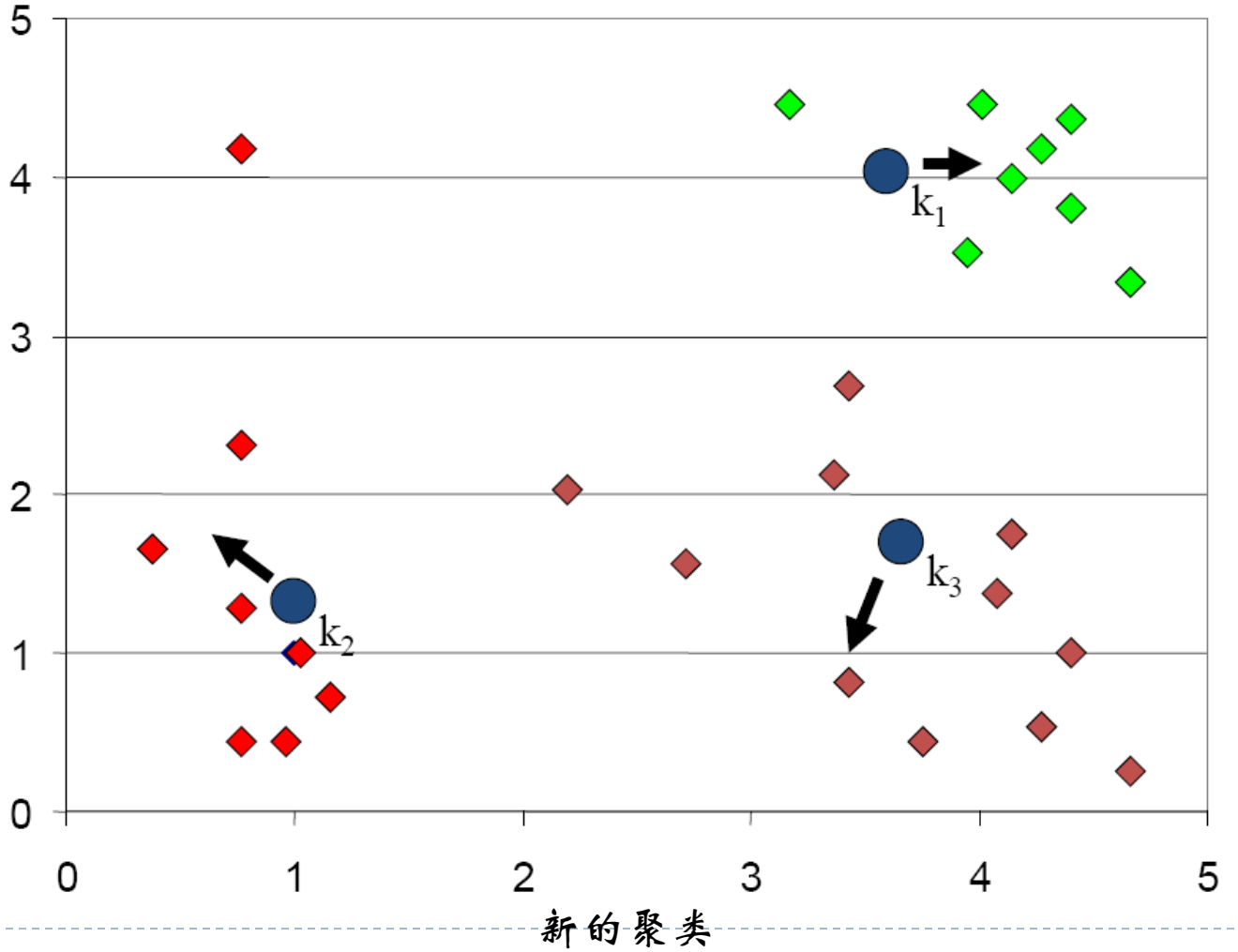


新的聚类

K-means clustering-Step 3

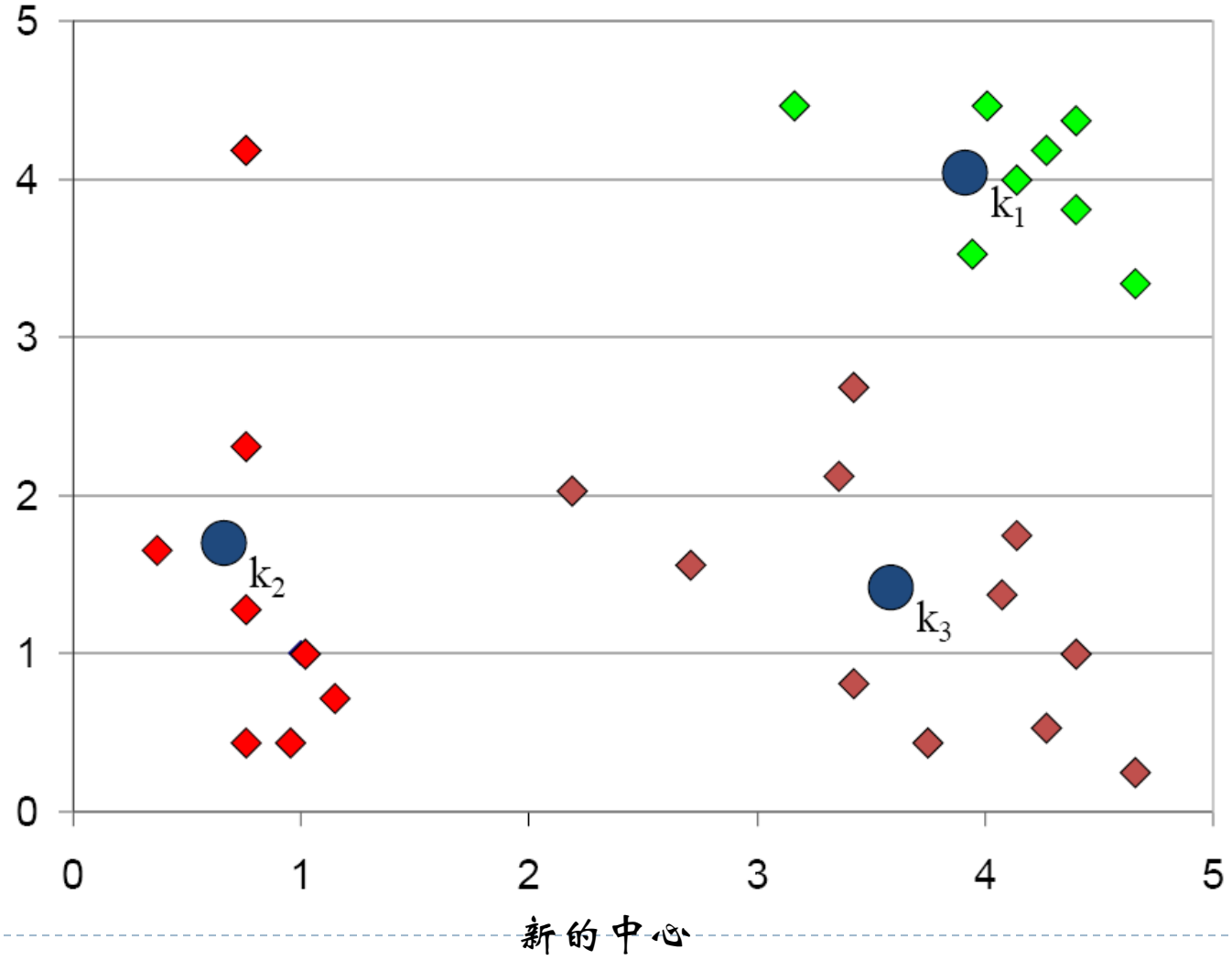


K-means clustering-Step 4



新的聚类

K-means clustering-Step 5



Improvment

Spectral clustering (谱聚类)

自行学习。。。。。

图像分割算法

- 小结
- 图像分割是把图像分成各具一定特性的区域并提取感兴趣目标区域的过程。本课程介绍了常用的分割方法：
  - 边缘点检测 (考虑了像素间的不连续性)
  - Hough变换 (利用了点-线对偶原理)
  - 灰度门限法(利用了图像直方图分布特性)
  - 区域分割法包括区域生长和区域分裂合并(利用了像素间的相似性)
  - 了解基于聚类的图像分割

Part 7: 图像特征描述