

运动分析

董秋雷

中国科学院自动化研究所

qldong@nlpr.ia.ac.cn

运动分析

- 什么是运动分析？

就是在不需要人为干预的情况下，综合利用计算机视觉、模式识别、图像处理、人工智能等诸多方面的知识对摄像机拍摄的图像序列进行自动分析，实现对动态场景中人的定位、跟踪和识别，并在此基础上分析和判断人的行为。

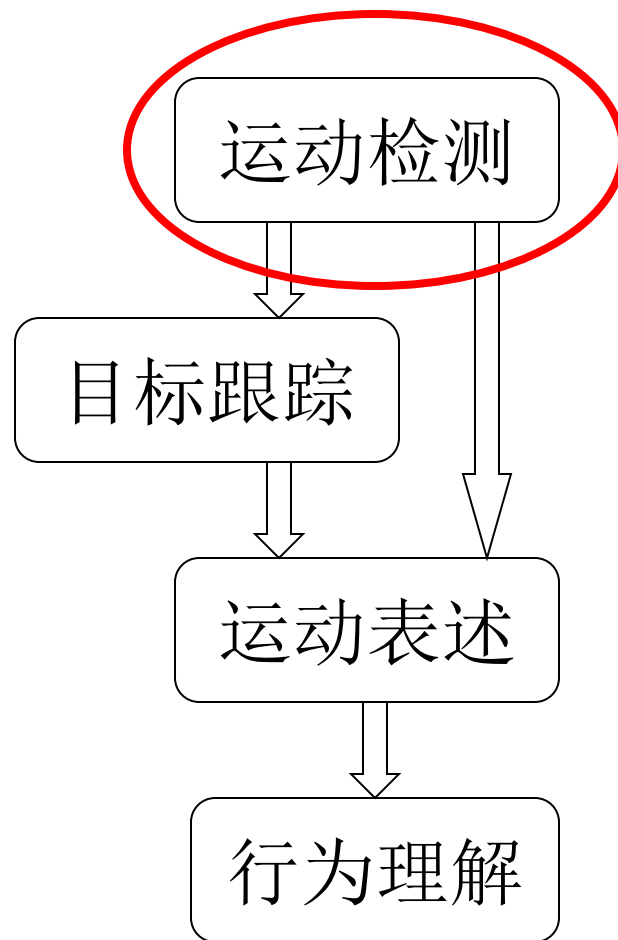


运动分析



运动分析

- 运动分析框图



运动检测

运动检测

- 什么是运动检测 (Motion detection) ?
 - 将运动前景从图像序列中提取出来，也就是说将背景与运动前景分离开。



运动检测

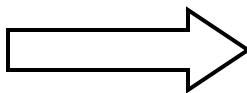
- 运动检测的重要性：
 - 是目标跟踪、运动表述和行为理解等后续处理的基础。
- 运动检测的难点：
 - 受天气、光照、阴影等诸多外界因素以及背景物体内在因素的影响，图像中的背景也常常是动态变化的。

运动检测

- 两种总体思路：
 1. 直接利用前景所特有的信息检测前景；

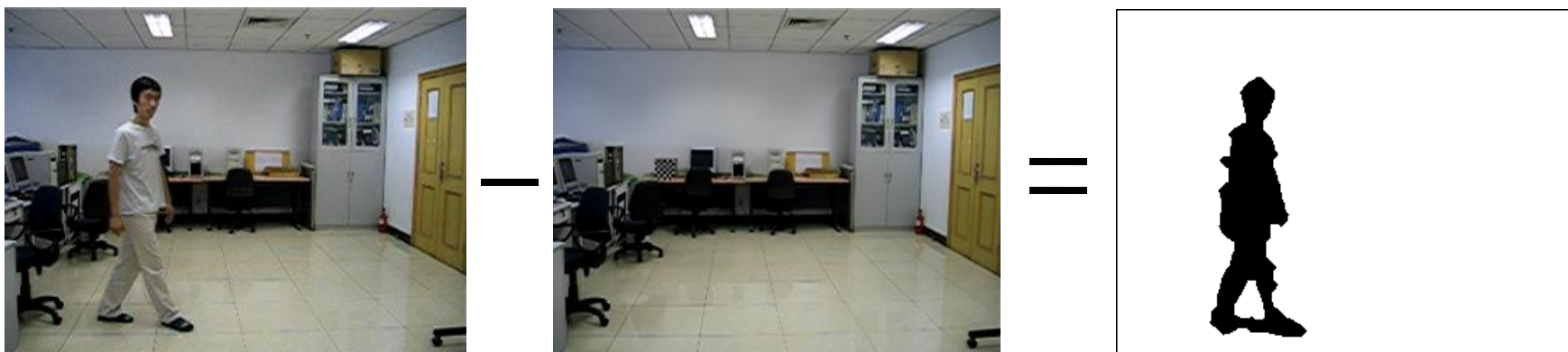


基于前景信息



运动检测

2. 先得到背景图象，然后将输入图象减去背景图像从而得到前景图象。



运动检测

- 到底哪种思路好呢？也就是说，前景和背景哪一个相对好检测一些呢？

没有绝对的好与不好，要视具体环境而定：

- 一些场景的背景相对固定，而前景变化较大；
- 一些场景的背景有较大变化，而前景的一些特征变化不大。

运动检测

- 常规的运动检测方法：
 - 背景差法 (background subtraction)
 - 光流 (optical flow)
 - 帧间差分 (frame differencing)
 - 前景建模方法

背景差法

- 原理：计算当前图像与背景图像的逐像素的灰度差，再通过设置阈值来确定运动前景区域。



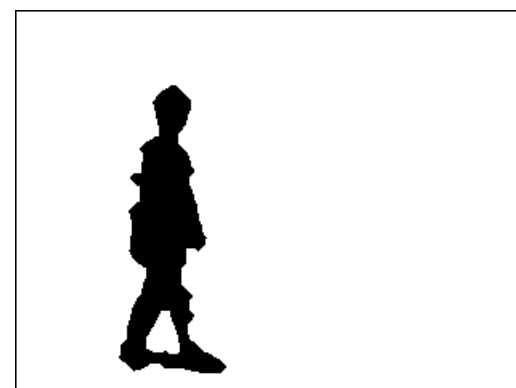
输入图象



背景图象

—

=



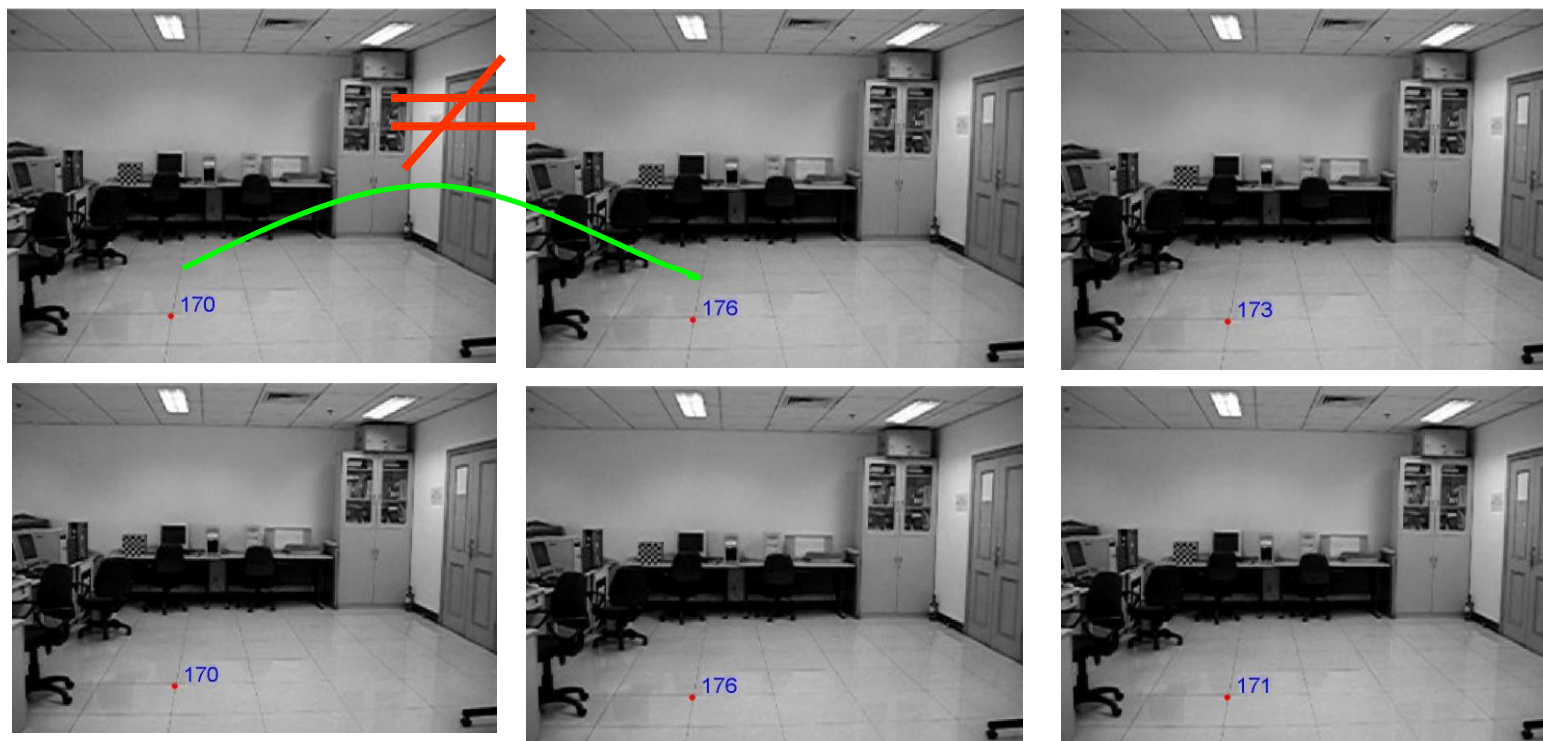
已知背景的情况——均值图像

- 均值图像：将若干背景图像求逐点的灰度均值。
- 什么是逐点的灰度均值？



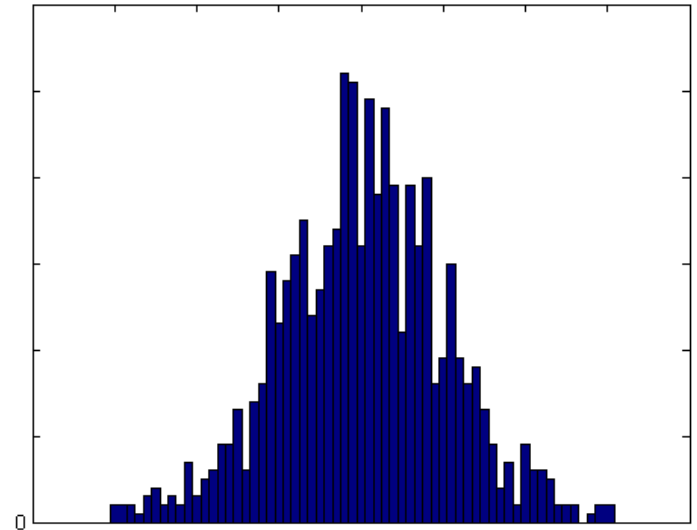
均值图像

- 为什么要将若干背景图像的均值图像作为背景，而不是直接从中选一张作为背景呢？

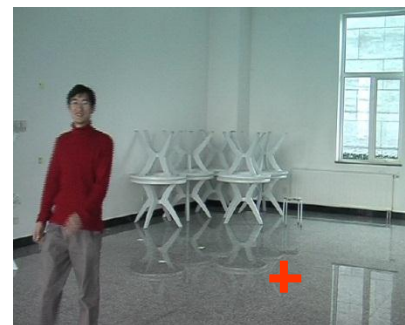
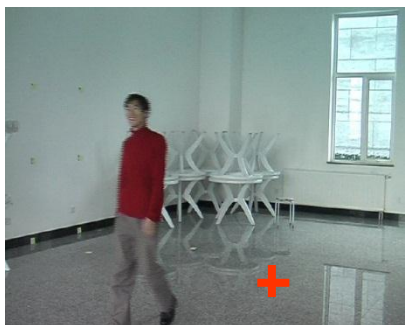
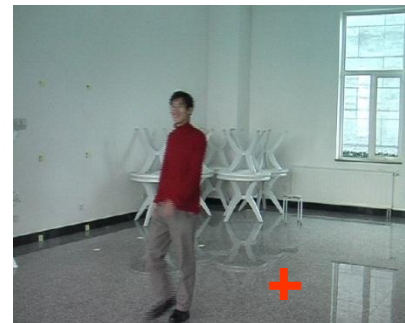
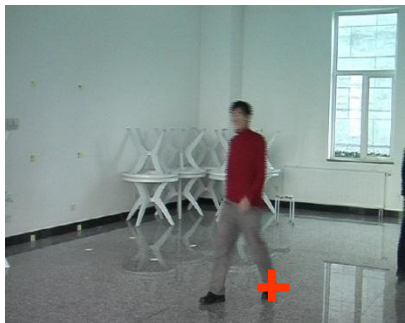


单高斯模型

- 什么是单高斯模型？
 - 对于每一个像素，用一个高斯分布来描述其在不同时刻的灰度分布情况的背景模型。
- 使用高斯分布来描述背景的假设是什么呢？
 - 背景在图像序列中总是最经常被观测到



单高斯模型



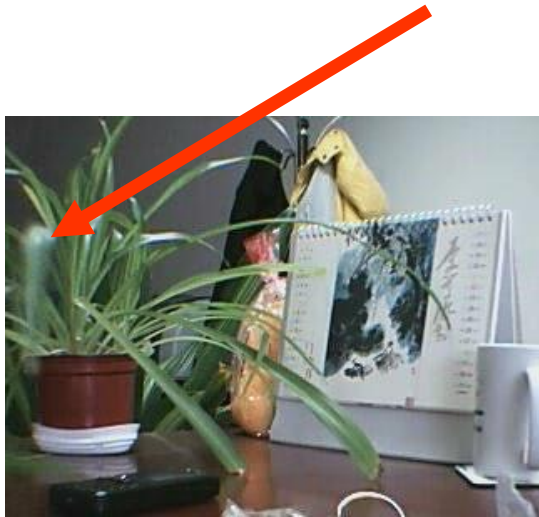
单高斯模型



单高斯模型

- 单高斯模型有没有不足呢？
- 背景往往不是绝对静止的，而是时常变化的！尤其对于室外场景。
- 例子：
 - 树枝摇曳；
 - 窗帘晃动；
 - 等等。





混合高斯模型

- 与单高斯模型类似，混合高斯模型采用混合高斯分布来描述每个像素在不同时刻的灰度分布情况。
- 其假设依然是背景在图像序列中总是最经常被观测到。

混合高斯模型

- 设用来描述每个像素的高斯分布共 K 个（ K 通常取3 – 7个），像素 z_{uv} 的概率函数：

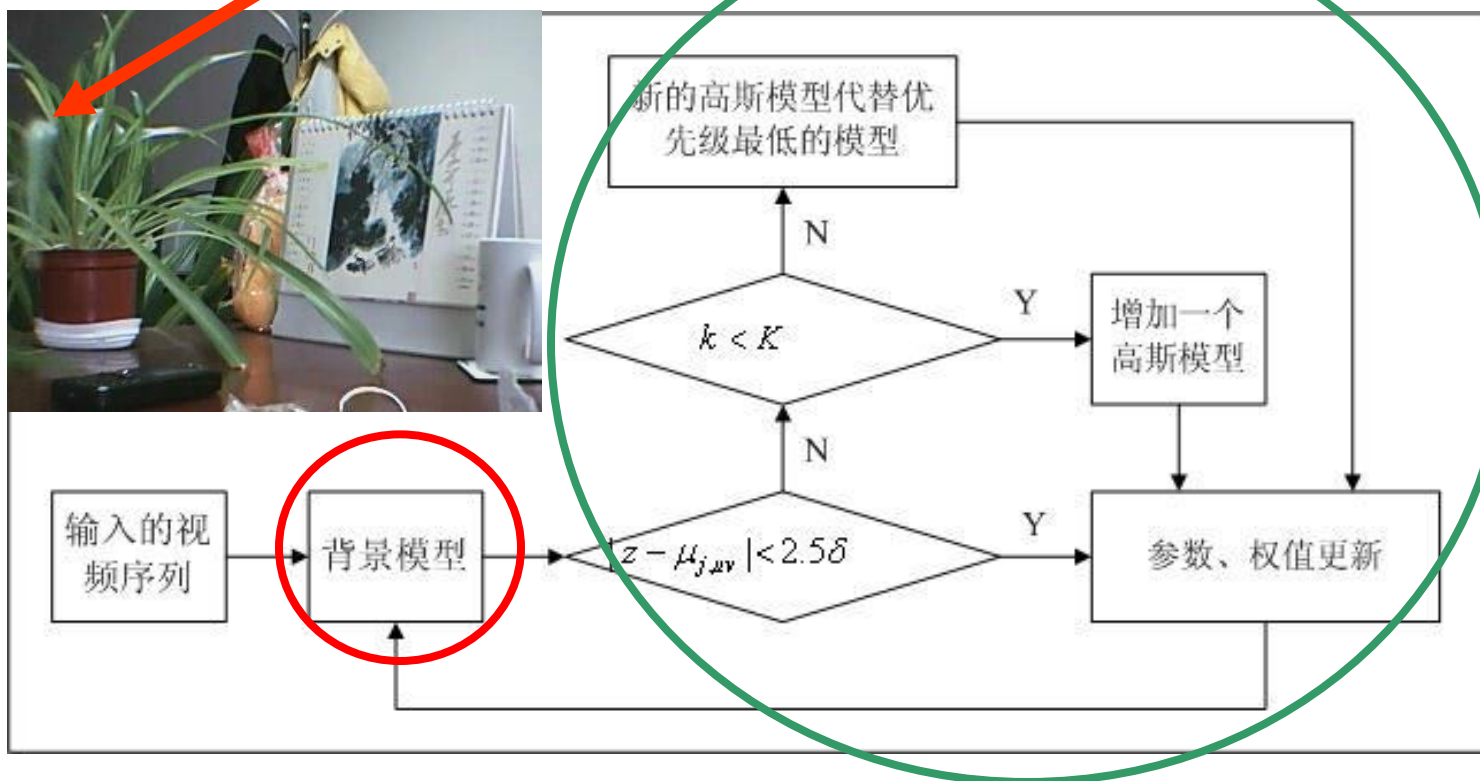
$$P(z_{uv}) = \sum_{j=1}^K \omega_{j,uv} N(z_{uv} | \mu_{j,uv}, \Sigma_{j,uv})$$

其中 $\omega_{j,uv}$ 是第 j 个高斯分布权重

$$N(z | \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} \Sigma^{1/2}} e^{-\frac{1}{2}(z-\mu)^T \Sigma^{-1} (z-\mu)}$$

混合高斯模型

- 流程图



混合高斯模型

背景建模过程（仅针对单个像素）：

1. 初始化：用第一幅图像该点的像素值作为均值，给定一个较大的方差和较小的权值。

2. 模型学习

$\omega_{j,w,t} = (1 - \alpha)\omega_{j,w,t-1} + \alpha(M_{j,w,t-1})$, $M_{j,w,t-1}$ 是个0/1函数

$$\mu_{j,w,t} = (1 - \rho)\mu_{j,w,t-1} + \rho z_{j,w,t}$$

$$\sigma_{j,w,t}^2 = (1 - \rho)\sigma_{j,w,t-1}^2 + \rho(z_{j,w,t} - \mu_{j,w,t})^T (z_{j,w,t} - \mu_{j,w,t})$$

$$\rho = \alpha N(z_{j,w,t} | \mu_{j,w,t}, \Sigma_{j,w,t})$$

混合高斯模型

- 这 K 个高斯模型是否应该都用上呢？
 - 由于噪声的影响或前景物体的存在，某些像素值并不能代表背景，因此由这些像素值构造的高斯分布应该去掉。
- 定义各个高斯分布的优先级：

$$p_{j,uv} = \omega_{j,uv} / \sigma_{j,uv}$$

混合高斯模型

- 如何去掉多余的高斯分布？
 - 前面提到的假设：前景和噪声不会在同一位置太长时间，这样，前景和噪声对应的高斯模型的权值和优先级都比较小，因此可以将 K 个高斯分布按优先级由高到低排列，用如下策略选取前 B 个分布作为背景模型：
- B 的定义：

$$B = \min_b (\sum_{j=1}^b \omega_{j,uv} > M)$$

其中 M 为预设的阈值。（如果 M 较小，则为单高斯模型）

混合高斯模型

- 前景检测：将待测图像的每一个像素点与该像素点对应的混合高斯模型的各个模型分别进行比较，若有 $|z - \mu_{j,uv}| < a\sigma$ （ a 为一常数），则该点属于背景，否则属于前景。



运动检测

- 常规的运动检测方法：
 - 背景差法 (background subtraction)
 - 光流 (optical flow)
 - 帧间差分 (frame differencing)
 - 前景建模方法

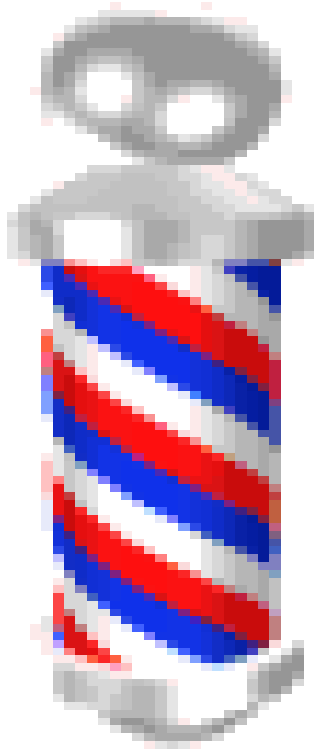
光流法

什么是光流？

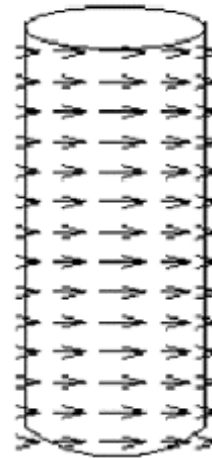
- 当人的眼睛观察运动物体时，物体的景象在人眼的视网膜上形成一系列连续变化的图像，这一系列连续变化的信息不断“流过”视网膜（即图像平面），好像一种光的“流”，故称之为光流（optical flow）。
- 光流是空间运动物体在观测成像面上的像素运动的瞬时速度。光流的研究是利用图像序列中的像素强度数据的时域变化和相关性来确定各自像素位置的“运动”。

光流法

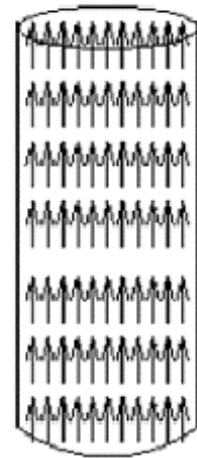
- Barber pole illusion



Barber's pole



Motion field

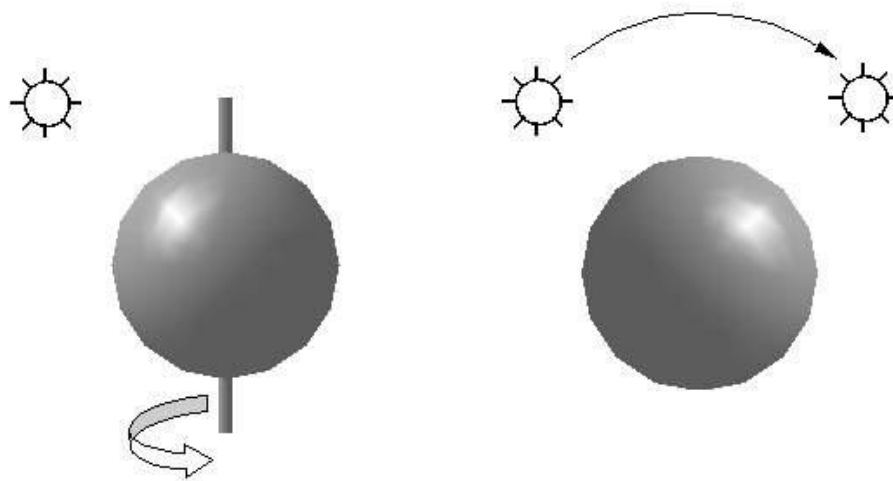


Optical flow

光流法

- 点的光流和点的实际运动有什么区别和联系呢？
 - 运动场(motion field)：一个运动物体在空间产生一个三维的速度场，运动前后空间对应点在图象上的投影形成一个二维运动场。
 - 光流场(optical flow field)：是指图像亮度模式的表观（或视在）运动，是二维矢量场。它包含的信息即是各像点的瞬时运动速度矢量信息。

光流法



光流等于零，但运动场
却不等于零。

光流不等于零，但运动
场为零。

光流法

- 为什么要研究光流呢？
 - 事实上，仅仅通过图像序列很难计算出物体的空间位置进而得到真实的运动场。而光流表达了图像的变化，包含了目标一定的运动信息，通过计算光流场可以从图像中近似计算不能直接得到的运动场。

光流法

- 如何计算光流呢？
 - 1981 年，Horn和Schunck创造性地将二维速度场与灰度相联系，引入光流约束方程，得到光流计算的基本算法。

光流约束方程

光流一致性假设 (Brightness constancy assumption)

$$I(x + u\Delta t, y + v\Delta t, t + \Delta t) = I(x, y, t)$$

方程左侧按Taylor级数展开，化简：

$$\frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0$$

光流约束方程

$$\text{令 } u = \frac{\partial x}{\partial t}, v = \frac{\partial y}{\partial t}, I_x = \frac{\partial I}{\partial x}, I_y = \frac{\partial I}{\partial y}, I_t = \frac{\partial I}{\partial t}$$

$$\text{则方程 } \frac{\partial I}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial I}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial I}{\partial t} = 0$$

化为如下光流约束方程：

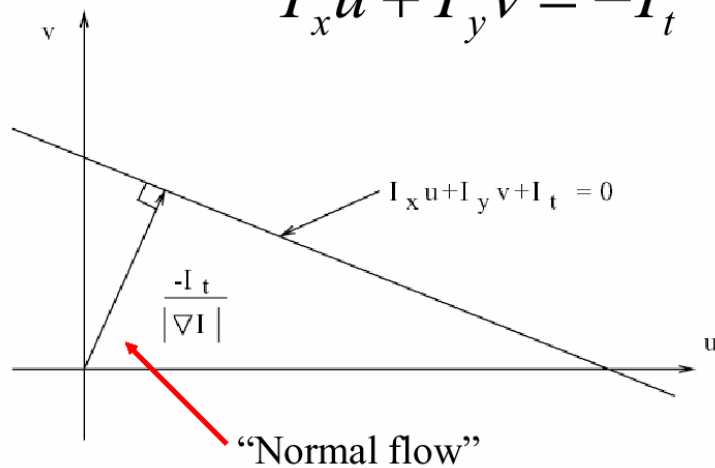
$$I_x u + I_y v + I_t = 0$$

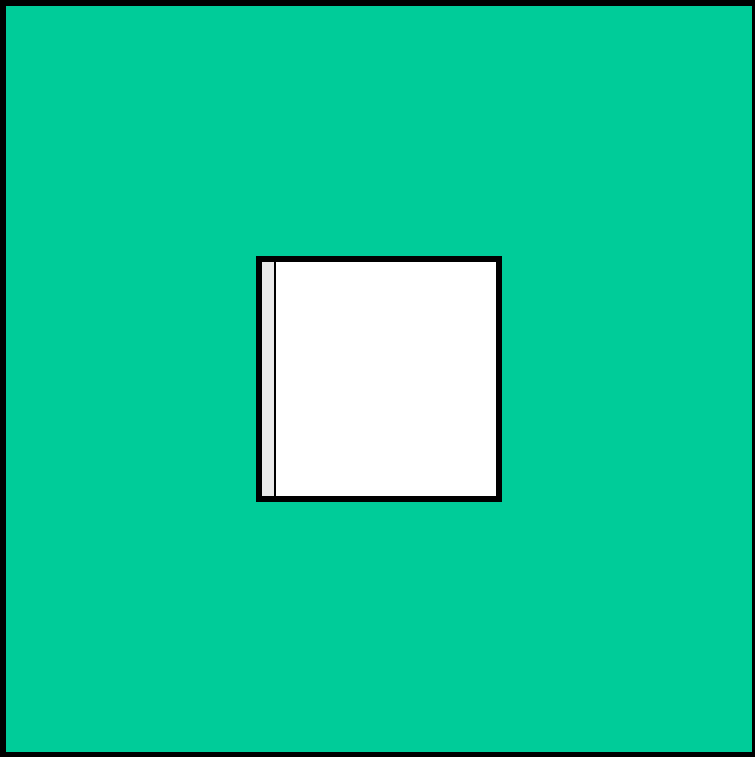
孔径问题

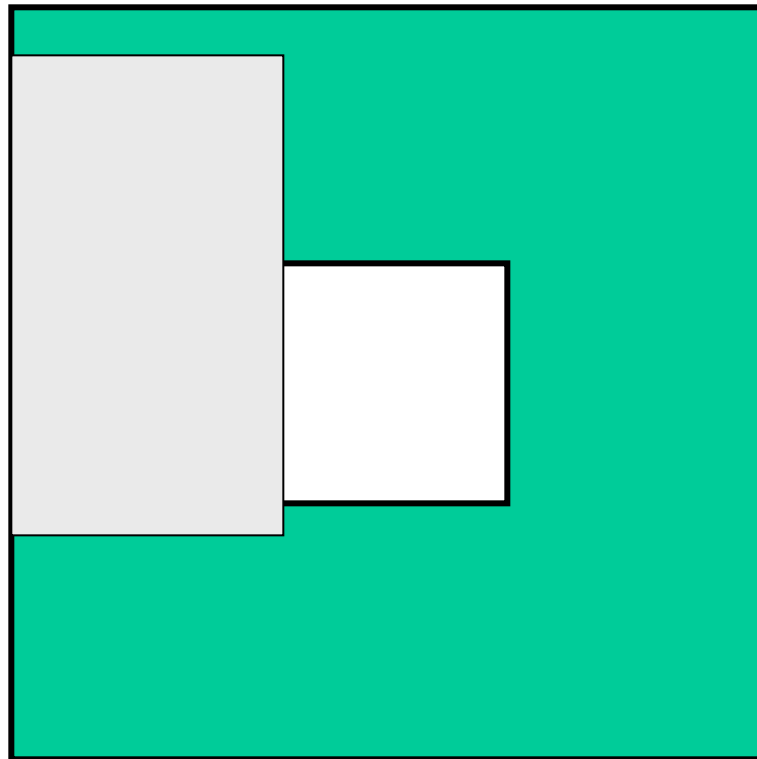
- 由光流约束方程可见：该方程两个未知数 u 和 v 。因此，只使用一个点上的信息是不能确定光流的。人们将这种不确定问题称为孔径问题 (aperture problem)。

At a single image pixel, we get a line:

$$I_x u + I_y v = -I_t$$







怎样为每个像素建立更多的方程呢？

光流计算

- 基本的解决思路：添加其它的约束。
- 光流的五种计算方法：
 - 基于梯度的方法
 - 基于匹配的方法
 - 基于能量的方法
 - 基于相位的方法
 - 神经动力学方法

基于梯度的方法——Lucas-Kanade方法

通常附加的约束是：假设光流场是局部平滑的。

Lucas-Kanade方法[3]的约束则更强一些：假设每个像素领域内的像素具有相对的速度。

$$E(u, v) = \sum_{(x,y) \in \Omega} (I_x(x, y)u + I_y(x, y)v + I_t)^2$$

对其两边求偏导：

Lucas-Kanade方法

$$\frac{\partial E(u, v)}{\partial u} = \sum_{(x, y) \in \Omega} 2I_x(I_x(x, y)u + I_y(x, y)v + I_t) = 0$$

$$\frac{\partial E(u, v)}{\partial v} = \sum_{(x, y) \in \Omega} 2I_y(I_x(x, y)u + I_y(x, y)v + I_t) = 0$$

$$\Rightarrow \begin{bmatrix} \Sigma I_x^2 & \Sigma I_x I_y \\ \Sigma I_x I_y & \Sigma I_y^2 \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \Sigma I_x I_t \\ \Sigma I_y I_t \end{bmatrix}$$

$$\Rightarrow (\Sigma \nabla I \nabla I^T) \vec{U} = -\Sigma \nabla I I_t$$

Lucas-Kanade方法

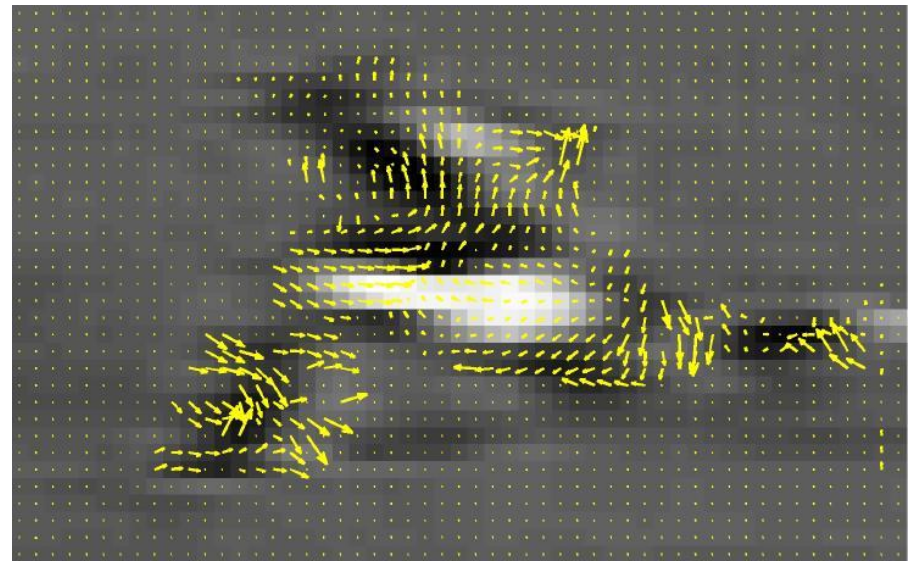
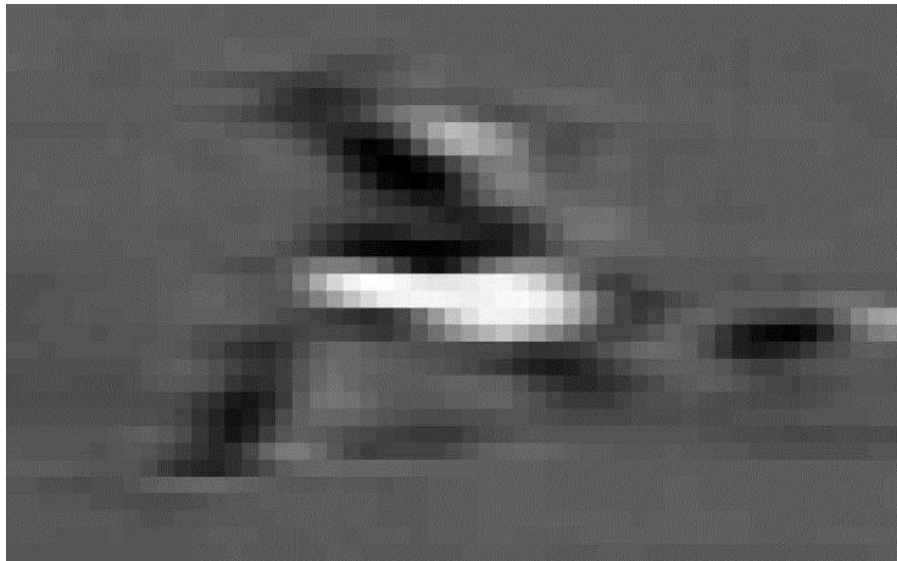
令 $A = \Sigma \nabla I \nabla I^T$, $\vec{U} = [u, v]^T$, $b = -\Sigma \nabla I I_t$

则有：

$$A\vec{U} = b$$

此时，原问题转化为一个最小二乘问题。

注意，考虑可解性！



光流法

- 光流法的优点
 - 直观的表述运动模式；
 - 不易受运动物体外表的干扰。
- 光流法的不足
 - 计算量比较大；
 - 易产生较大噪声。

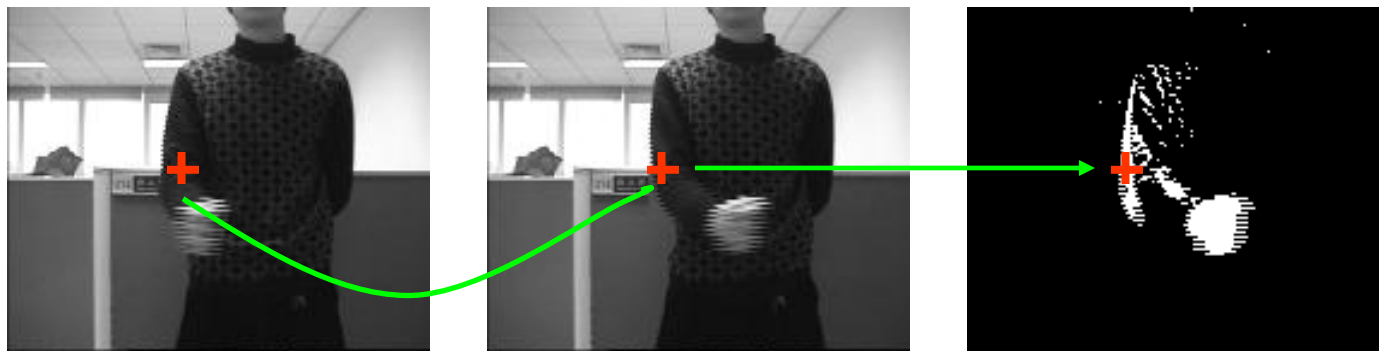
运动检测

- 常规的运动检测方法：
 - 背景差法 (background subtraction)
 - 光流法 (optical flow)
 - 帧间差分 (frame differencing)
 - 前景建模方法

帧间差分

- 原理：

在相邻两帧（也可以为多帧）间计算逐像素的灰度差，并通过设置阈值来确定对应运动前景的像素，进而得到运动前景区域。



双帧差分

- 数学表述：

如果 I_n, I_{n-1} 表示图像序列中任意相邻的两帧图像，则逐像素的差分图 D_n 可以定义为：

$$D_n(i, j) = |I_n(i, j) - I_{n-1}(i, j)|$$

对上述差分图阈值化（假设预先设定的阈值为 T ），即可确定运动前景区域 M_n ：

$$M_n(i, j) = \begin{cases} 0 & D_n(i, j) < T \\ I_n(i, j) & D_n(i, j) \geq T \end{cases}$$

三帧差分

- 扩展——三帧差分：其原理与双帧差分的原理是一样的，只是在具体构造差分图上有些许不同。



I_{n-1}



I_n



I_{n+1}

三帧差分方法

- 三帧差分方法的数学表述：

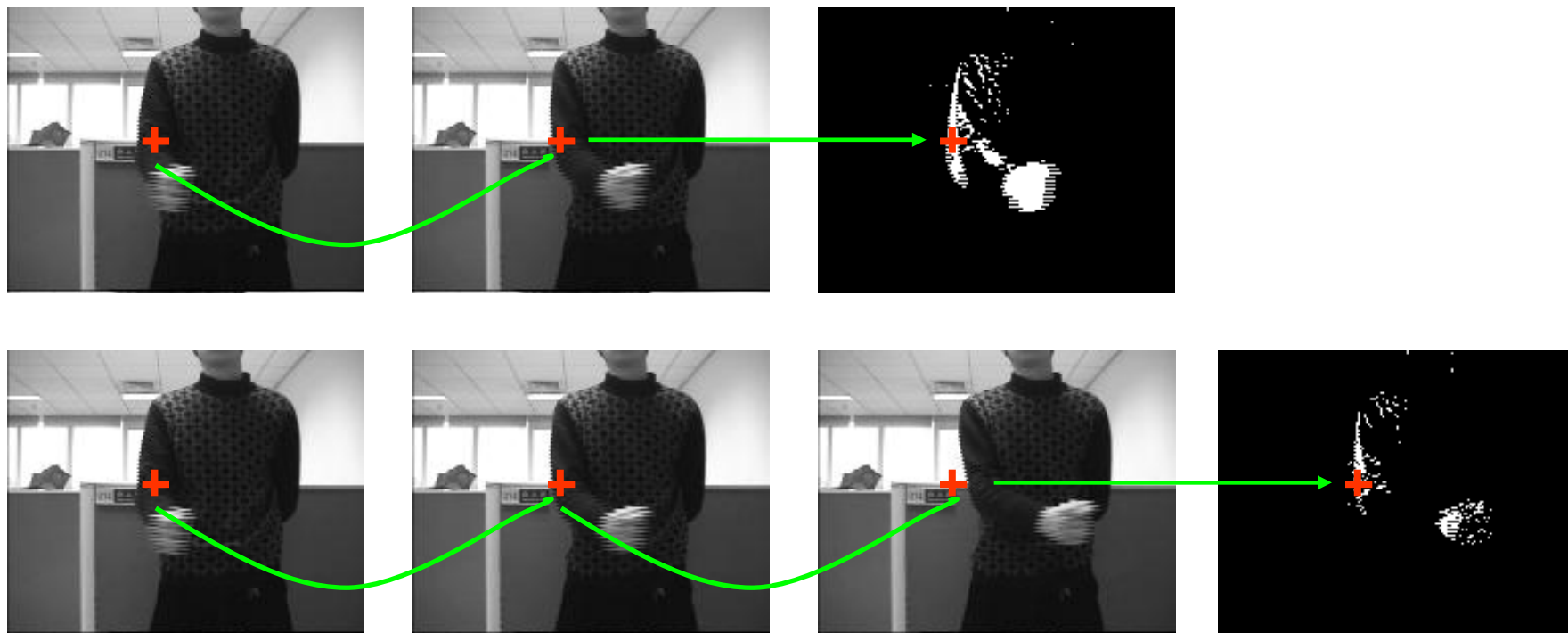
如果 I_{n-1}, I_n, I_{n+1} 表示图像序列中任意连续的三帧图像，则逐像素的差分图定义为：

$$D_n(i, j) = |I_{n-1}(i, j) - I_n(i, j)| \times |I_n(i, j) - I_{n+1}(i, j)|$$

对上述差分图阈值化（假设预先设定的阈值为 T ），即可确定运动前景区域 M_n ：

$$M_n(i, j) = \begin{cases} 0 & D_n(i, j) < T \\ I_n(i, j) & D_n(i, j) \geq T \end{cases}$$

对比效果图



帧间差分

- 优点：
 - 适用于动态变化的背景环境。
- 不足：
 - 较难准确检测运动速度过快的物体；
 - 较难准确检测运动速度过慢的物体；
 - 较难准确检测场景中同时存在的多个运动物体。

运动检测

- 常规的运动检测方法：
 - 背景差法 (background subtraction)
 - 光流法 (optical flow)
 - 帧间差分 (frame differencing)
 - 前景建模方法

前景建模方法

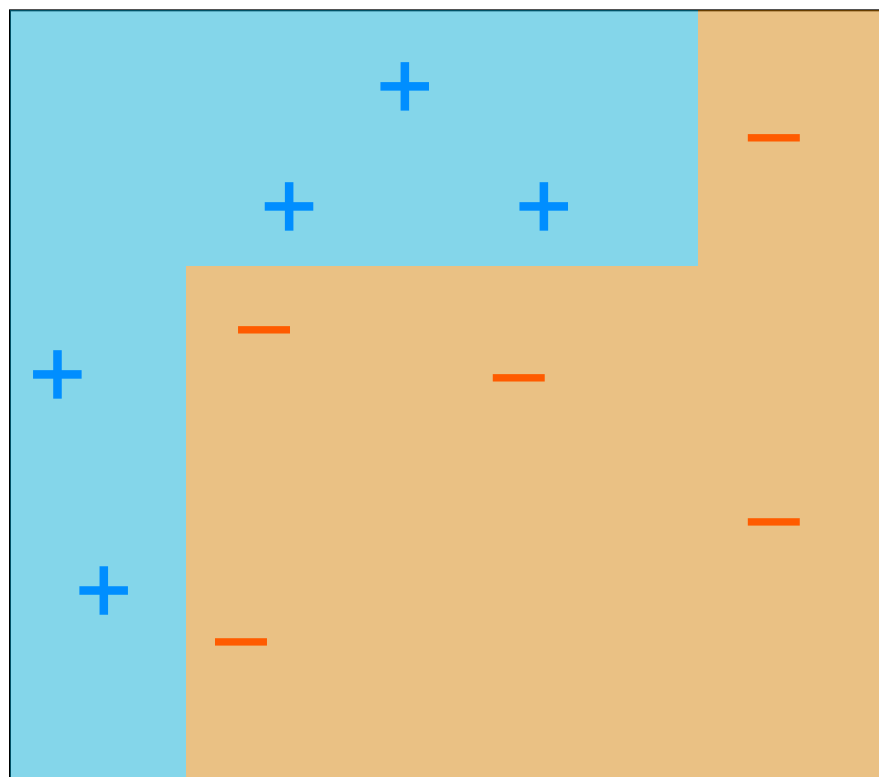
- 前景建模方法：直接针对前景所特有的特征进行建模。
- 此处，我们仅以人脸检测为例，介绍一种流行的学习方法——AdaBoost，以及其在人脸检测方面的应用。



AdaBoost

Toy problem

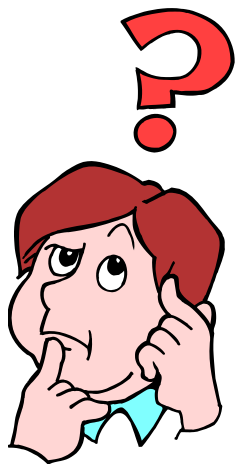
- 怎样把两类点区分开呢？



AdaBoost



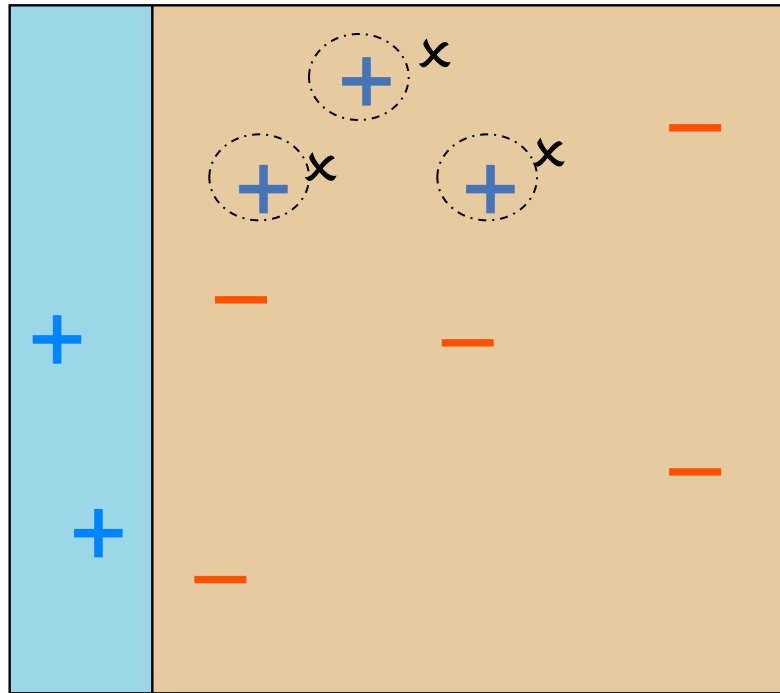
“要善于‘退’，足够地‘退’，退到最原始而不失去重要性的地方，退到我们最容易看清楚的地方……”



哪里是“最原始而不失去重要性的地方”呢？

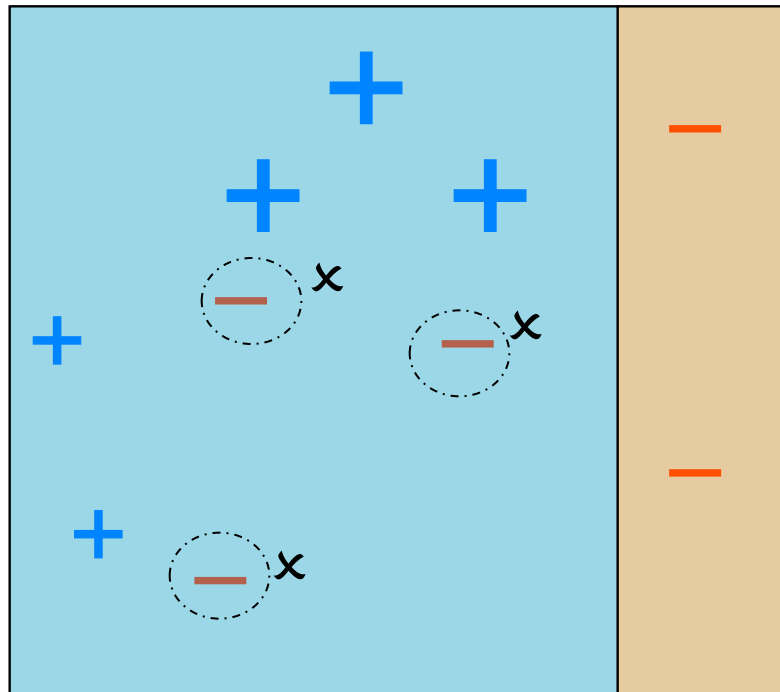
Toy problem

- 第一种划分:



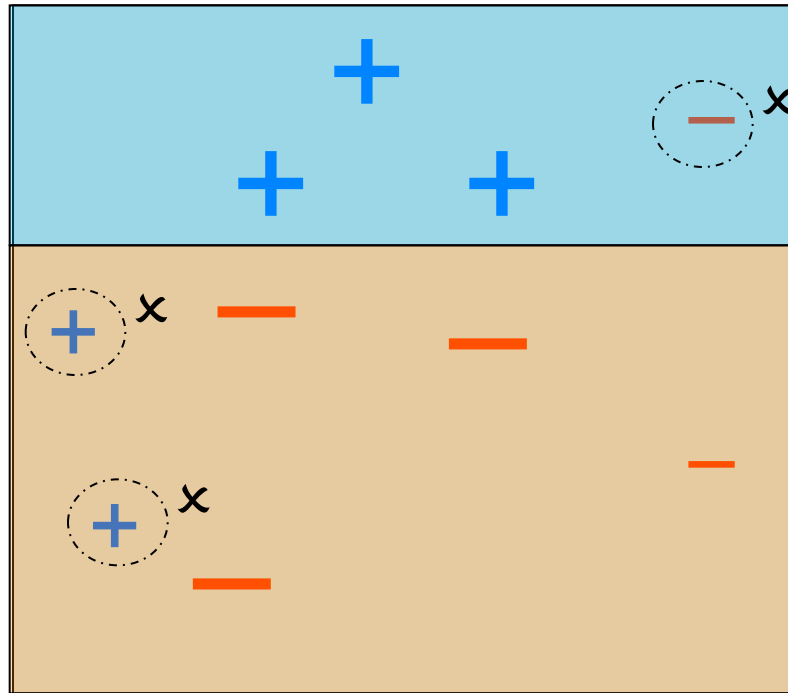
Toy problem

- 第二种划分:

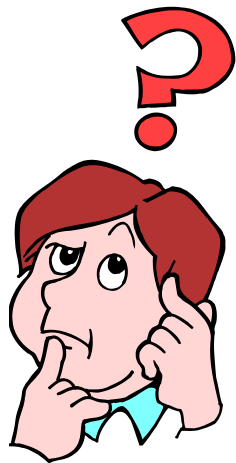


Toy problem

- 第三种划分:



AdaBoost



“最原始而不失去重要性的地方”找到了，然后怎么办呢？

换言之：

- 弱分类器：准确率不高（仅比随机猜测略好）；
- 强分类器：准确率很高的分类器；
- 怎么样将弱分类器提升为强分类器呢？

AdaBoost

- AdaBoost (adaptive boosting) Boosting 算法中的一种变形。
- 目的：就是根据已有的样本，融合多个弱分类器形成一个整体的强分类器，并提高分类准确率。

AdaBoost

- 发展历史：
 - Proved the fact: learners, each performing only slightly better than random, can be combined to form an arbitrarily good ensemble hypothesis. (Kearns and Valiant 1989)
 - Boost-by-majority algorithm (Freund 1990)
 - AdaBoost (Freund and Schapire 1994-1996)
 - Generalized version of AdaBoost (Schapire and Singer 1997)
 - AdaBoost in face detection (Viola and Jones 2001)

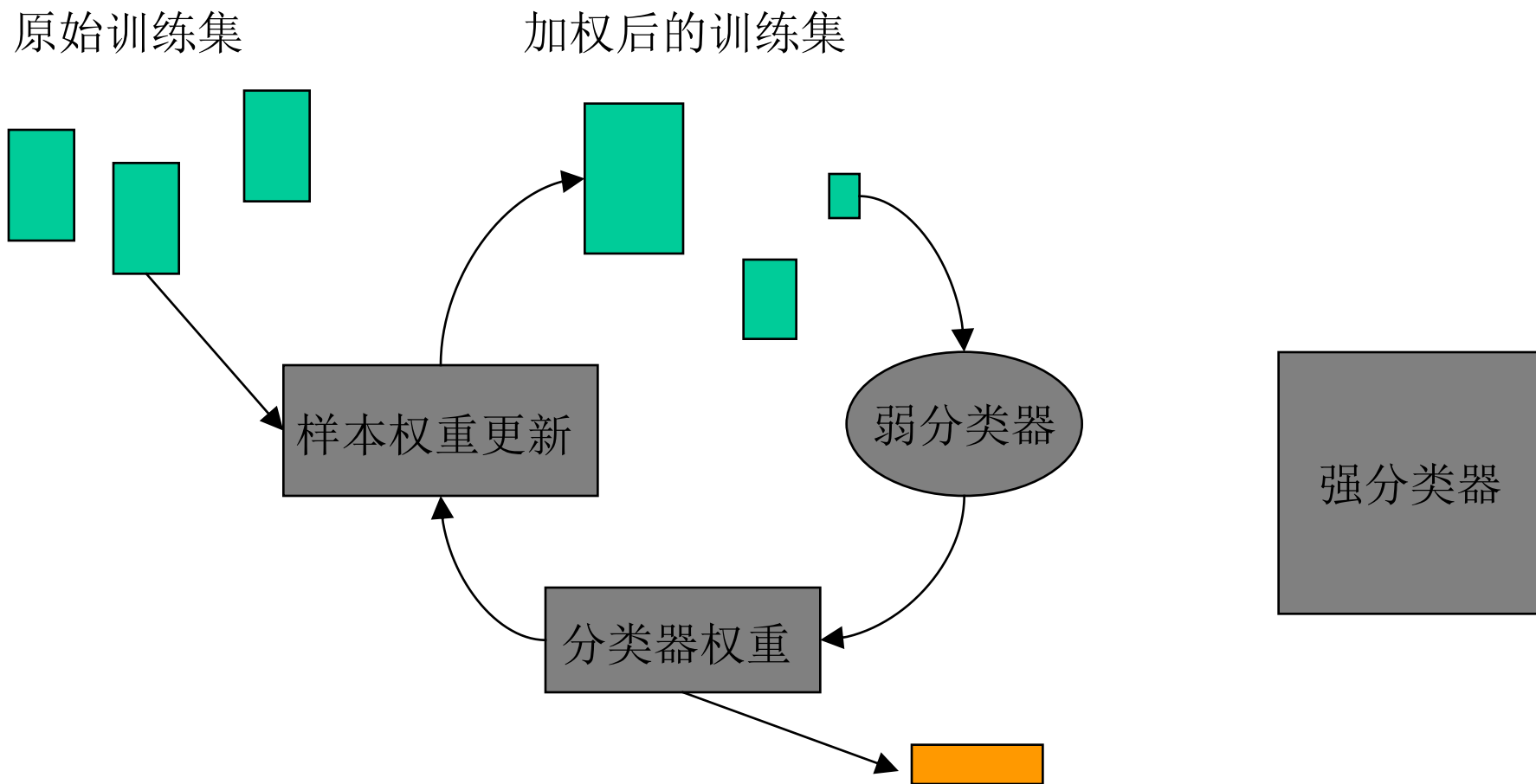
两个问题

- AdaBoost具体解决两个问题：
 - 怎样处理训练样本？
 - 怎样合并弱分类器成为一个强分类器？

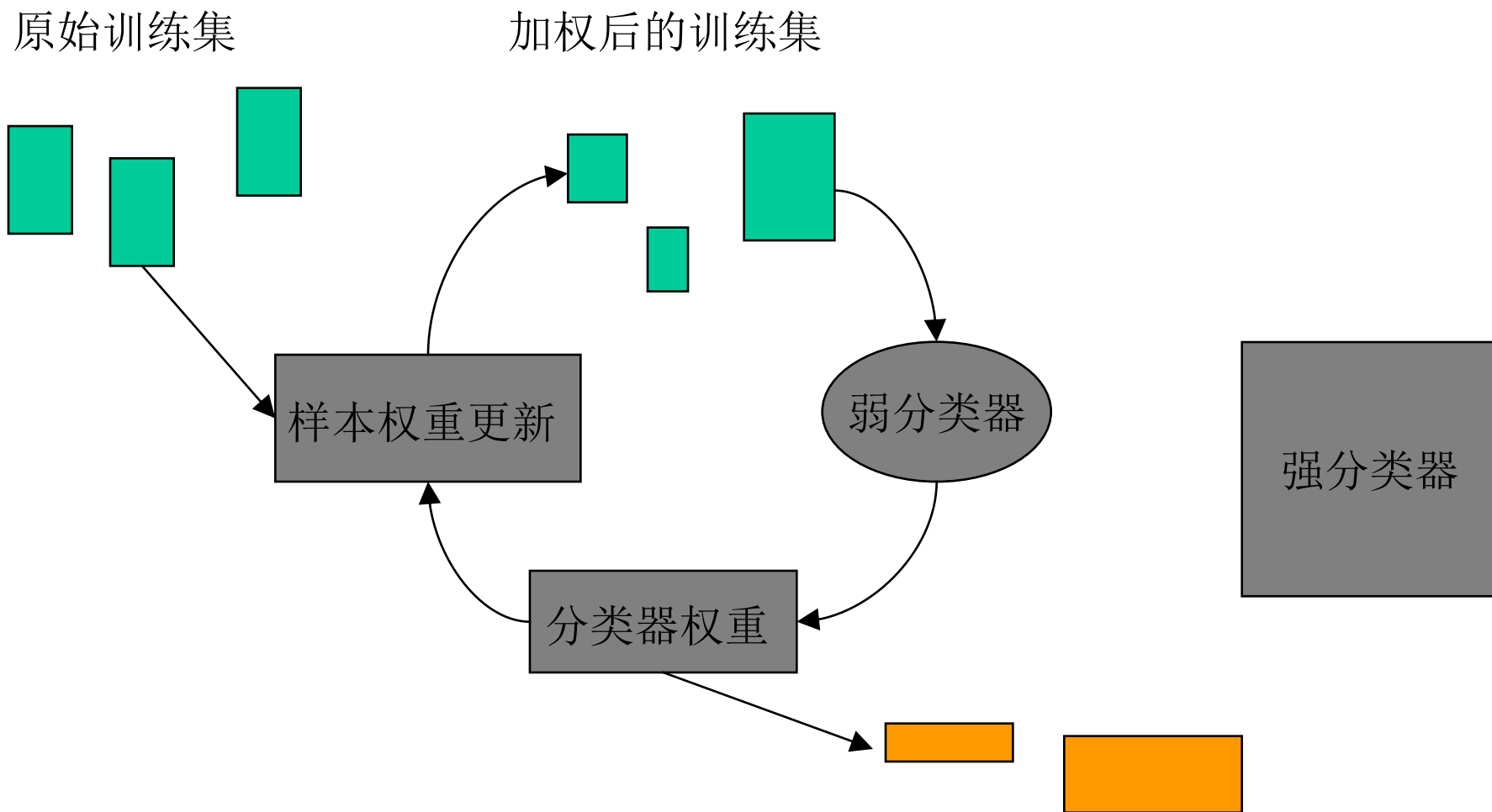
问题答案

- 怎样处理样本？
 - 在AdaBoost中，每个样本都被赋予一个权重。如果某个样本没有被正确分类，它的权重就会被提高，反之则降低。这样，AdaBoost方法将注意力更多地放在“难分”的样本上。
- 怎样合并弱分类器？
 - 强分类器表示为若干弱分类器的线性加权和形式，准确率越高的弱学习机权重越高。

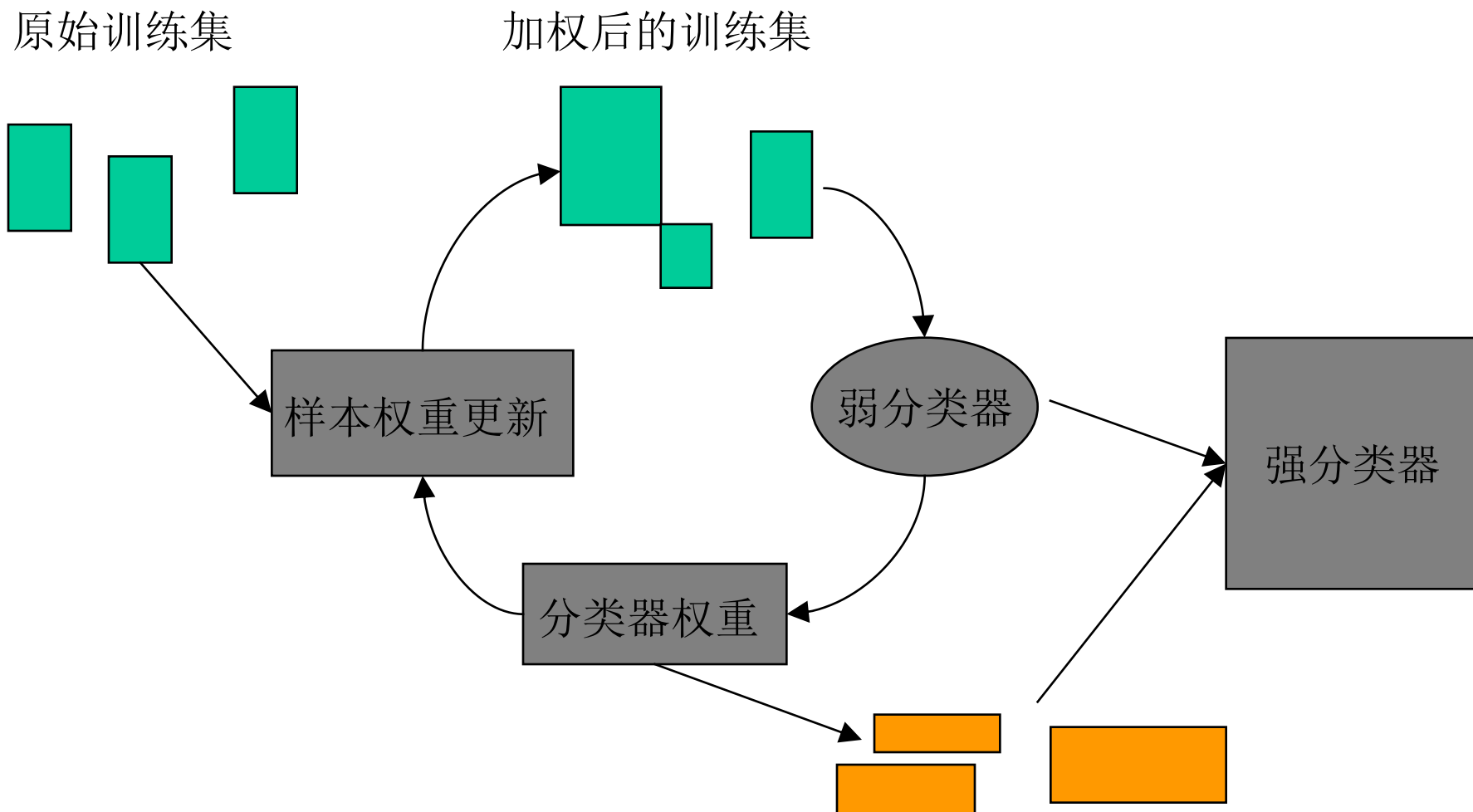
AdaBoost流程 (loop1)



AdaBoost流程 (loop2)



AdaBoost流程 (loop3)



AdaBoost法则

输入： m 个样本 $(x_1, y_1), \dots, (x_m, y_m)$, 其中 $x_i \in X, y_i \in Y = \{-1, 1\}$

初始化样本权重 $D_{t+1}(i) = 1/m$

For $t = 1$ to T

1. 找到弱分类器 h_t , 并得到其分类误差 ε_t

ε_t 越大, α_t 越小

2. 计算分类器权重 $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$

3. 更新样本权重

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

输出:

Z_t 是归一化因子

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$$

AdaBoost法则

怎样获得每次循环的弱分类器？

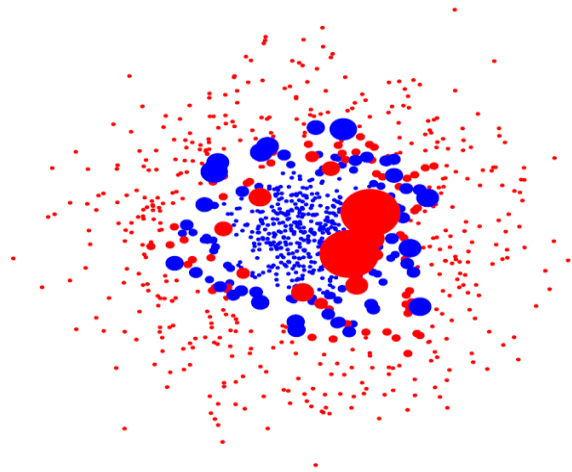
1. $h_t = \operatorname{argmin}_{h_j} \{ \varepsilon_j = \sum_{i=1}^m D_t(i) [y_i \neq h_j(x_i)] \}$
2. 先决条件： $\varepsilon_t < 1/2$ ，否则终止循环

弱分类器的例子： 如决策树等等。

AdaBoost法则

样本权重更新过程：

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$



增加被错分样本的权重，
同时减小被正确划分的
样本的权重

AdaBoost法则再回顾

初始化;

For $t = 1$ to T

1. 找到弱分类器:

$$h_t = \operatorname{argmin}_{h_j} \{ \varepsilon_j = \sum_{i=1}^m D_t(i) [y_i \neq h_j(x_i)] \}$$

2. $\varepsilon_t < 1/2$, 否则终止循环;

3. 计算分类器权重: $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$

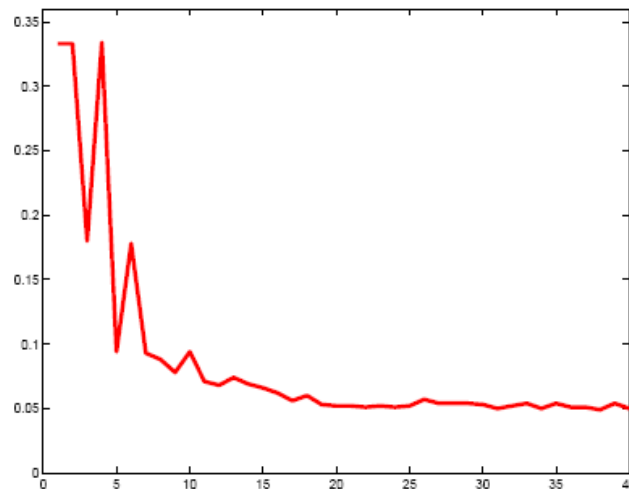
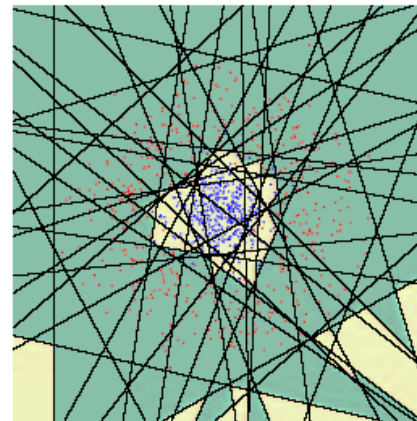
4. 更新样本权重:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

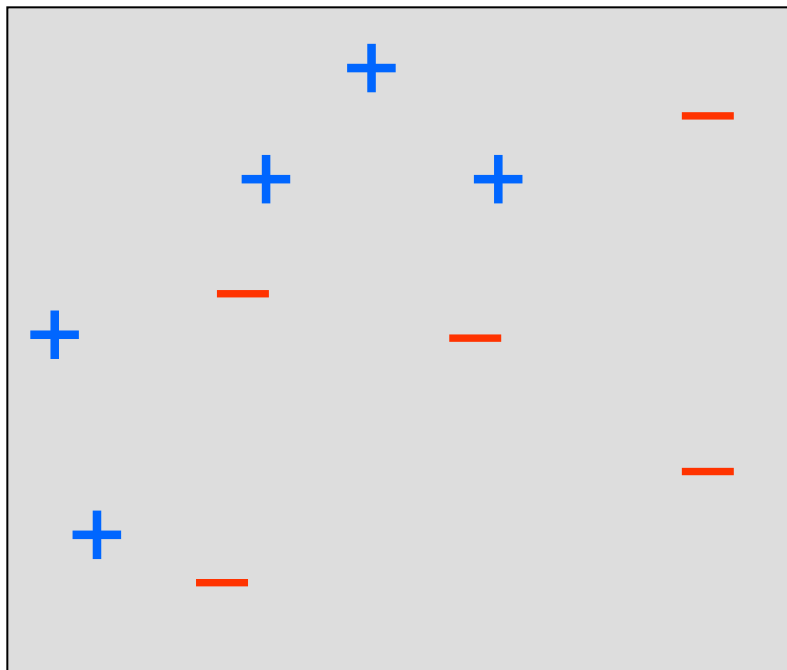
输出强分类器:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$$

$t = 40$



Toy problem再回顾



Train data

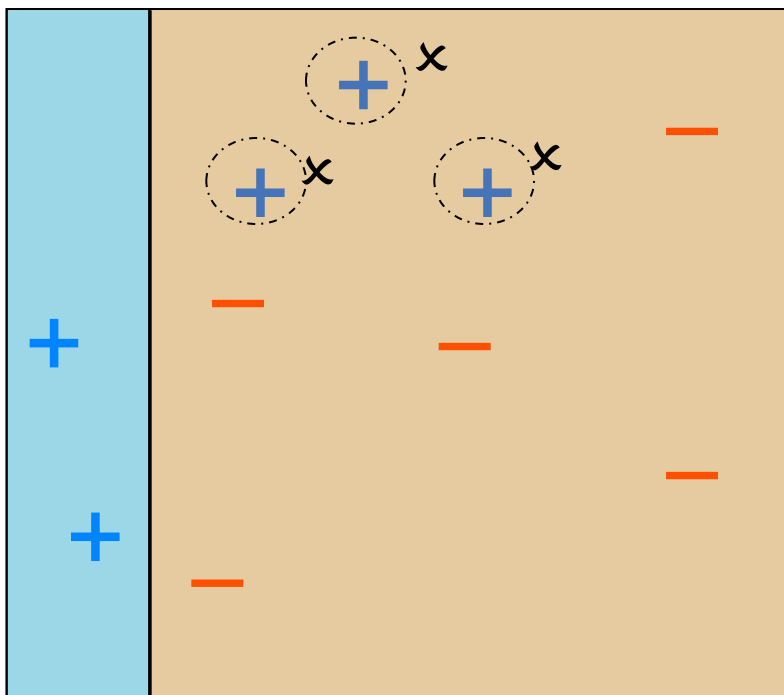
x1	x2	y
1	5	+
2	3	+
3	2	-
4	6	-
4	7	+
5	9	+
6	5	-
6	7	+
8	5	-
8	8	-

Initialization

[illegible]

Toy problem再回顾

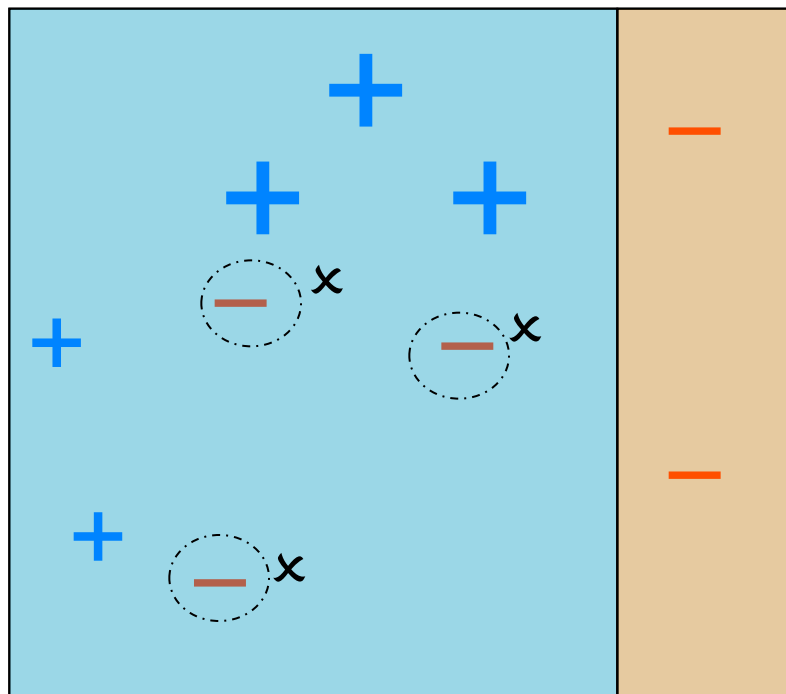
第一次循环：



h1e	ε	D2
0	0.00	0.07
0	0.00	0.07
0	0.00	0.07
0	0.00	0.07
1	0.10	0.17
1	0.10	0.17
0	0.00	0.07
1	0.10	0.17
0	0.00	0.07
0	0.00	0.07
$\varepsilon 1$	0.30	1.00
$\alpha 1$	0.42	\updownarrow
Z_t		0.92

Toy problem再回顾

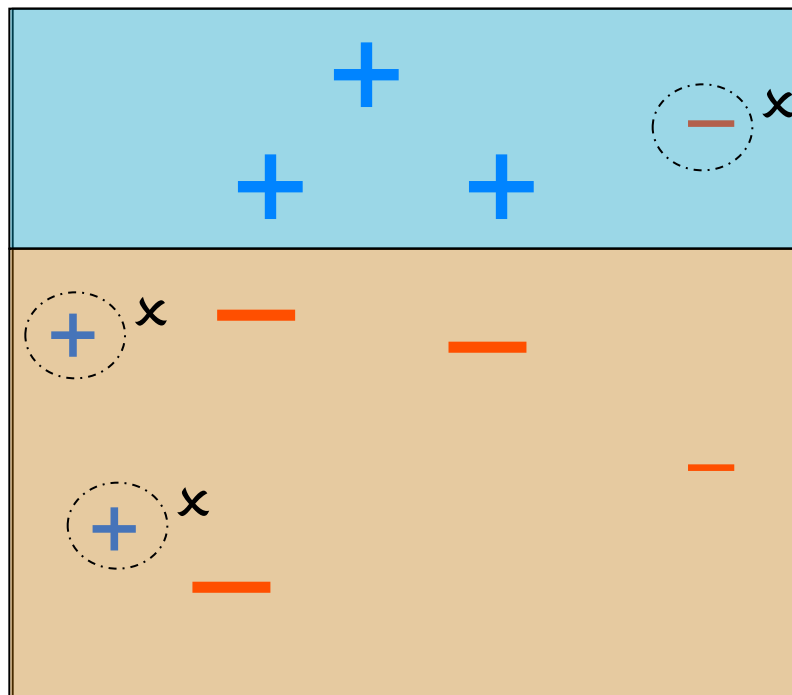
第二次循环：



h2e	ε	D3
0	0.00	0.05
0	0.00	0.05
1	0.07	0.17
1	0.07	0.17
0	0.00	0.11
0	0.00	0.11
1	0.07	0.17
0	0.00	0.11
0	0.00	0.05
0	0.00	0.05
ε_2	0.21	1.00
α_2	0.65	\updownarrow
Zt		0.82

Toy problem再回顾

第三次循环：



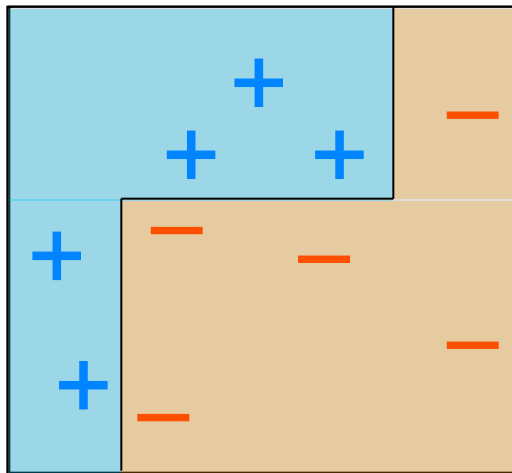
$h_3 e$	ϵ
1	0.05
1	0.05
0	0.00
0	0.00
0	0.00
0	0.00
0	0.00
0	0.00
0	0.00
0	0.00
1	0.05
ϵ_3	0.14
α_3	0.92

Toy problem 再回顾

$$H(x) = \text{sign} \left(0.42 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{orange} \end{array} + 0.65 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{orange} \end{array} + 0.92 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{orange} \end{array} \right)$$

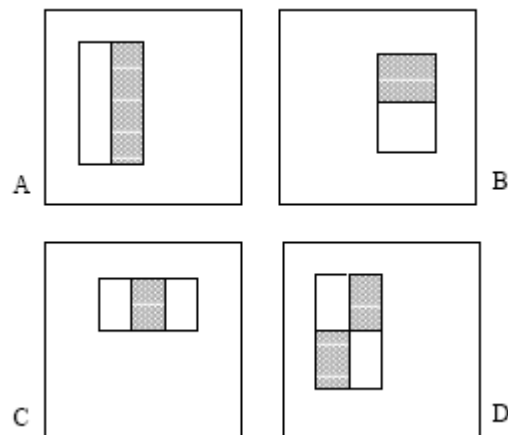
$$= \text{sign}[0.42(h_1 - 1) + 0.65(h_2 - 1) + 0.92(h_3 - 1)]$$

=



实例——人脸检测

- 特征选取：



实例——人脸检测

- 面临的挑战：
 - 特征数目太多！！！！
- 是否其中少量特征的组合可以构成一个有效的分类器？
- 如何找到这些有区分能力的特征？

一个AdaBoost的变体

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{l,i} = 1/(2m), 1/(2l)$ for training example i , where m and l are the number of negatives and positives respectively.

For $t = 1 \dots T$

- 1) Normalize weights so that w_t is a distribution
- 2) For each feature j train a classifier h_j and evaluate its error ε_j with respect to w_t
- 3) Chose the classifier h_j with lowest error.
- 4) Update weights according to:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-\varepsilon_i}$$

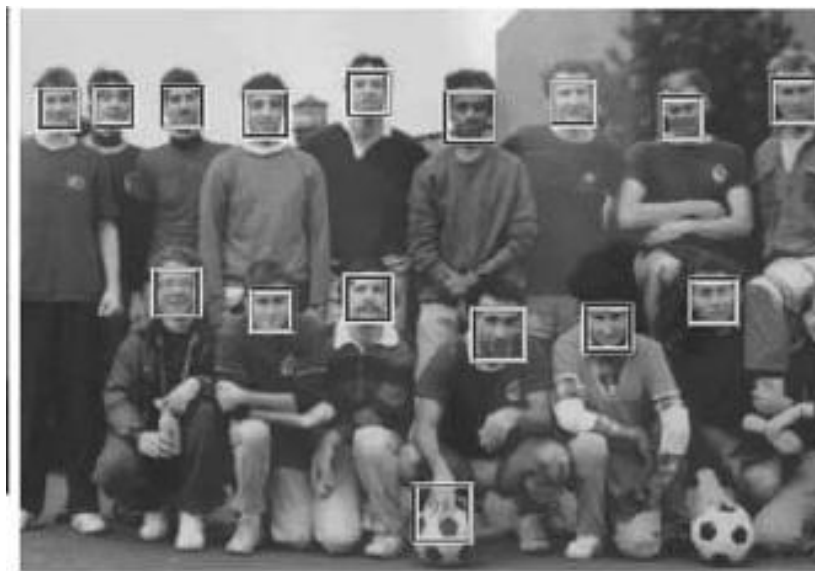
where $\varepsilon_i = 0$ if x_i is classified correctly, 1 otherwise, and

$$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t, \\ 0 & \text{otherwise} \end{cases} \quad \text{where} \quad \alpha_t = \log\left(\frac{1}{\beta_t}\right)$$

实例——人脸检测



小节

- AdaBoost的思想：
 - 将一系列粗略的规则加权组合起来得到高度精确的规则。
- 优点：
 - 易于执行；
 - 分类精度较高；
- 不足：
 - 易受噪声干扰；
 - 执行效果依赖于弱分类器的选择。

小节

- AdaBoost的应用
 - 人脸识别
 - 文本分类
 - 图像数据库中的图像识别
 - 手写体字符识别
 - 语音识别

参考文献

1. I. Haritaoglu, D. Harwood and L. Davis, "W4: Real-Time Surveillance of People and Their Activities," PAMI, vol.22, no.8, 2000.
2. C. Stauffer and W. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," PAMI, vol.22, no.8, pp. 747-757, 2000.
3. B. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", Proceedings of the 7th International Joint Conference on Artificial Intelligence, pp. 674-679, 1981.
4. P. Viola, M. Jones: Robust Real-Time Face Detection. International Journal of Computer Vision 57(2): 137-154, 2004.