



## 第七章：运动和跟踪

---

中国科学技术大学  
电子工程与信息科学系

主讲教师：李厚强 ([lihq@ustc.edu.cn](mailto:lihq@ustc.edu.cn))  
周文罡 ([zhwg@ustc.edu.cn](mailto:zhwg@ustc.edu.cn))



# 运动和跟踪

---

- 运动分析
- **目标跟踪**



# 目标跟踪

---

- 目标跟踪概述
- 贝叶斯跟踪框架
  - 卡尔曼滤波
  - 粒子滤波
- 均值漂移算法
- 相关滤波器

# 概述

## □ 目标跟踪：

目标跟踪是计算机视觉领域的重要研究方向之一，其目标是在连续的图像中对感兴趣物体进行检测、提取、识别和跟踪，从而获得目标物体的相关参数，如位置、速度、尺度、轨迹等，并对其进行进一步处理和分析，实现对目标物体的行为理解，或完成更高一级的任务。



初始化第一帧目标状态

.....



预测第N帧目标状态

# 概述

## □ 视觉跟踪技术典型应用

### ■ 安全监控

- ✓ 车站、机场、银行及超市等公共场所的实时监控。

### ■ 交通检测

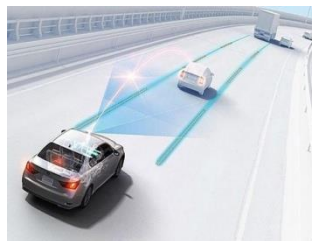
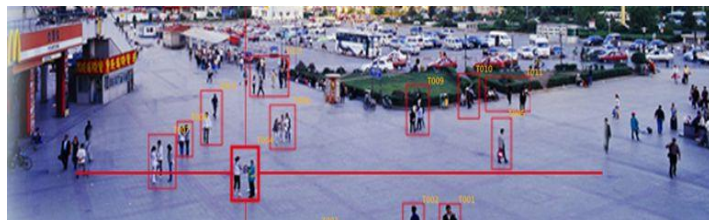
- ✓ 对行人及车辆行为进行判定，完善智能交通系统。

### ■ 军事领域

- ✓ 导弹制导、武器观测瞄准、敌方目标定位及跟踪。

### ■ 医学应用

- ✓ 标记、增强及跟踪生物特征来帮助医生诊断疾病。

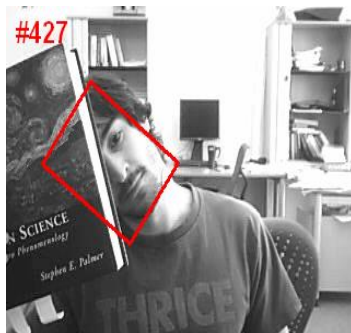
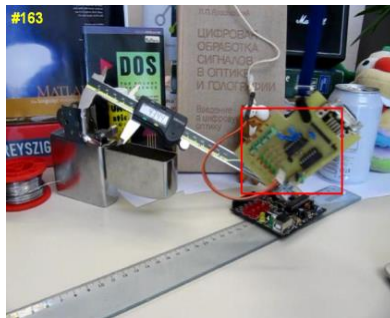




# 概述

## □ 主要技术挑战

- 光照变化
- 尺寸变化
- 快速运动
- 形变
- 背景杂乱
- 相机抖动
- 遮挡
- 运动模糊
- 旋转
- 超出视野
- .....



# 视频目标跟踪基本框架



表观模型

视觉目标表征

1. 生成式模型
2. 判别式模型

运动更新模型

跟踪目标定位

1. 粒子滤波
2. 全局搜索
3. 稠密采样



# 生成模型 (Generative Model)

## □ 特点:

- 通过前景信息拟合目标的外观表达

## □ 缺点:

- 缺少背景信息的利用, 容易漂移

## □ 经典算法:

- 均值漂移目标跟踪算法 (MeanShift)
- 基于颜色信息的粒子滤波
- 基于目标局部稀疏表达的粒子滤波
- .....





# 判别模型 (Discriminative Model)

## □ 特点:

- 通过前景和背景信息去拟合区分前景和背景的二分类器

## □ 缺点:

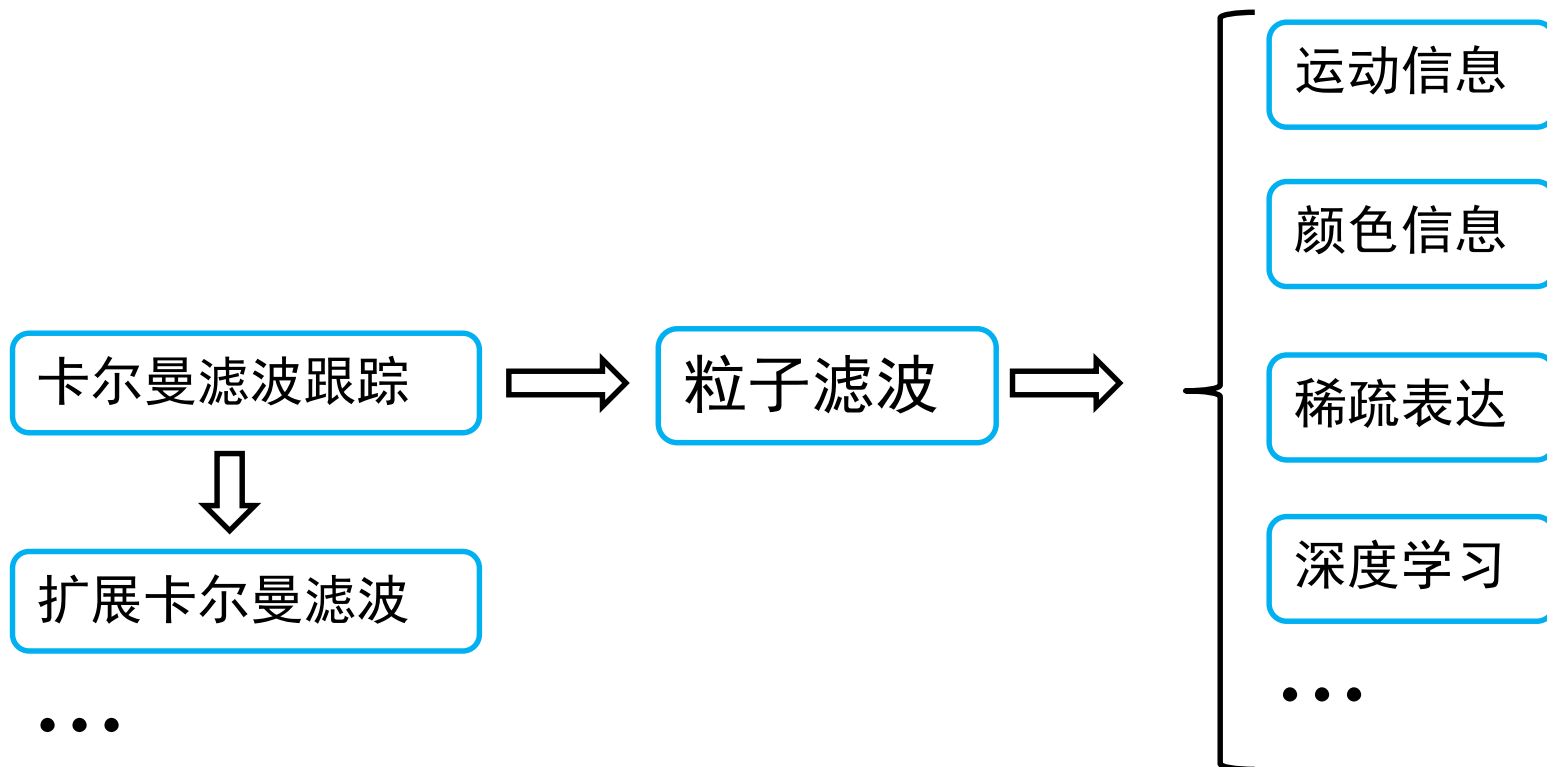
- 过分依赖训练样本及样本标签

## □ 经典算法:

- 基于支持向量机 (SVM) 的跟踪算法
- 基于相关滤波器的跟踪方法
- 基于深度学习的跟踪方法
- .....

# 贝叶斯跟踪框架

## □ 基于贝叶斯估计的跟踪框架：



# 贝叶斯跟踪框架

## □ 基于贝叶斯估计的跟踪框架

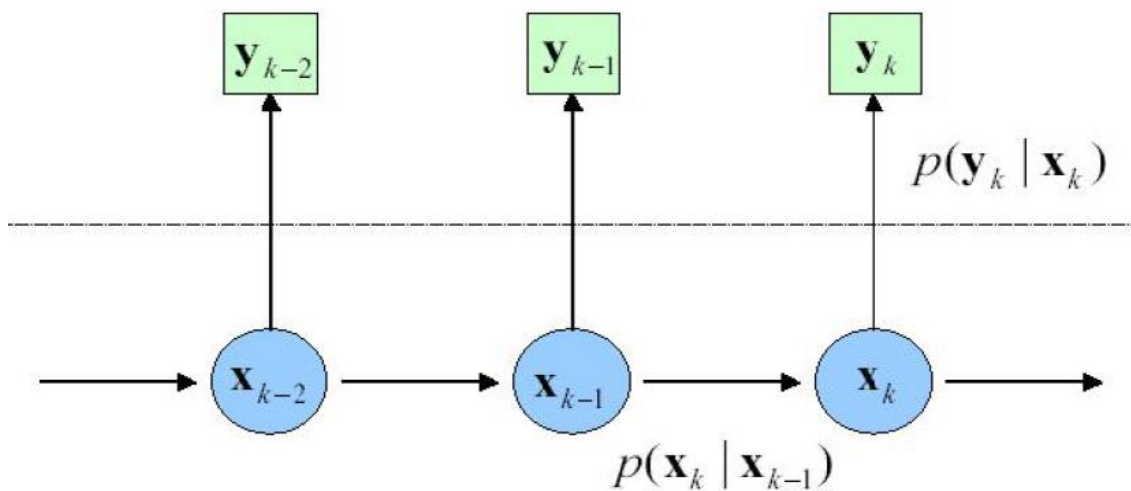
- 状态方程:  $x_k = f(x_{k-1}) + u_{k-1}$
- 观测方程:  $y_k = h(x_k) + v_k$

$x_k$  为系统状态,  $y_k$  为观测值,  $u_k$  为过程噪声,  $v_k$  为观测噪声。

在目标跟踪中,  $x$  代表候选样本的状态 (位置、尺度等信息), 它的标签  $y$  代表该候选是否为目标。

## □ 目标跟踪需要解决的问题

- 估计后验概率密度  $p(x_k | y_{1:k})$



# 贝叶斯跟踪框架

- 贝叶斯滤波将状态估计视为概率推理过程，即利用贝叶斯求解后验概率密度

假设已知状态变量的初始概率密度函数：  $p(x_0|y_0) = p(x_0)$ ，则估计后验概率密度  $p(x_k|y_{1:k})$  可以通过预测和更新两步递推得到。

预测：  $p(x_k|y_{1:k-1}) = \int \boxed{p(x_k|x_{k-1})} p(x_{k-1}|y_{1:k-1}) dx_{k-1}$

状态转移概率

更新：  $p(x_k|y_{1:k}) = \frac{\boxed{p(y_k|x_k)} p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}$

观测似然度

卡尔曼滤波通过线性、高斯假设条件可得到上述过程解析解。

粒子滤波算法通过数值逼近方法求近似解。



# 卡尔曼滤波跟踪算法

## □ 卡尔曼滤波基本思想：

- 以**最小均方误差**为最佳估计准则，采用**信号**与**噪声**的状态空间模型，利用前一时刻的**估计值**和当前时刻的**观测值**来更新对**状态变量**的估计，求出当前时刻的估计值
- 卡尔曼滤波融合估计和观测的结果，利用两者的不确定性来得到更加准确的估计

## □ 卡尔曼滤波的重要假设：

- 被建模的系统是**线性**的： $k$ 时刻的系统状态可以用某个矩阵与 $(k - 1)$ 时刻的系统状态的乘积表示。
- 影响测量的噪声属于**高斯分布**的**白噪声**，噪声与时间不相关，且只用**均值**和**协方差**就可以准确地建模。

# 卡尔曼滤波跟踪算法

在目标跟踪中,  $x$  代表候选样本的状态 (位置、尺度等信息), 它的标签  $y$  代表该候选是否为目标。可以看出, 卡尔曼滤波引入了较强的线性假设 (即  $\Phi, H$ )

## □ 卡尔曼滤波递推过程

(1) 建立空间模型 (线性假设) :

$$\begin{aligned} x_{k+1} &= \Phi_k x_k + u_k \\ y_{k+1} &= H_{k+1} x_{k+1} + v_{k+1} \end{aligned}$$

带噪声  $u, v$  的  
状态方程  
观测方程

(2) 设置初始化条件,  $k=0$ 时:

$$\begin{aligned} \hat{x}_0 &= E\{x_0\} \\ C_0 &= Var\{x_0\} \end{aligned}$$

(3) 预测:  $\hat{x}_{k+1}^- = \Phi_k \hat{x}_k$

(4) 计算预测误差协方差:  $C_{k+1}^- = \Phi_k C_k \Phi_k^T + Q_k$

(5) 计算卡尔曼增益:  $K_{k+1} = C_{k+1}^- H_{k+1}^T (H_{k+1} C_{k+1}^- H_{k+1}^T + R_{k+1})^{-1}$





# 卡尔曼滤波跟踪算法

## □ 卡尔曼滤波递推过程

(6) 更新:  $\hat{x}_{k+1} = \hat{x}_{k+1}^- + K_{k+1}(y_{k+1} - H_{k+1}\hat{x}_{k+1}^-)$

(7) 计算估计误差协方差:  $C_{k+1} = (I - K_{k+1}H_{k+1})C_{k+1}^-$

(8) 计算下一帧, 令  $k = k + 1$ , 重复预测和更新过程。

## □ 回顾卡尔曼递推的前提假设:

- 观测噪声和模型噪声均服从高斯分布;
- 状态转移函数和观测函数都是线性的;
- 处理过程中各项均是时不变的;

以上假设严重限制了卡尔曼滤波跟踪在实际应用中的效果。



# 粒子滤波跟踪算法

## □ 什么是粒子滤波？

- 粒子滤波 (PF: Particle Filter)的思想基于蒙特卡洛方法 (Monte Carlo methods)，利用粒子集来表示概率，可以用在任何形式的状态空间模型上。
- 其核心思想是通过从后验概率中抽取的随机状态粒子来表达其分布，是一种顺序重要性采样法 (Sequential Importance Sampling)。
- 粒子滤波的实质是根据一定规则(采样函数)采样一些随机粒子(样本)，观测粒子的相似度(似然)来确定粒子的权重，并利用粒子和权值来近似地表示后验概率。
- 可适用于非线性、非高斯系统，克服了卡尔曼滤波的缺点
- 弱点：粒子滤波的计算复杂度高
  - ✓ 随着计算机处理能力的不断增强，早期限制粒子滤波应用的硬件运算能力等障碍正逐渐消失。

# 粒子滤波跟踪算法

## □ 粒子滤波基本思想

假设能从**状态的后验概率分布**  $p(x_{0:k}|y_{1:k})$  中独立获取  $N$  个样本  $\{x_{0:k}^{(i)}\}_{i=1}^N$ ，则状态的后验概率分布可以通过如下离散近似表达式逼近，即使用  $N$  个样本通过蒙特卡洛的方法近似得到后验概率密度：

$$\hat{p}(x_{0:k}|y_{1:k}) = \frac{1}{N} \sum_{i=1}^N \delta(x_{0:k} - x_{0:k}^{(i)})$$

通过引入已知的、容易采样的概率密度分布  $q(x_{0:k}|y_{1:k})$  并从其中采样粒子。此时状态的后验概率可通过加权和的形式逼近：

$$\hat{p}(x_{0:k}|y_{1:k}) = \sum_{i=1}^N w_k(x_{0:k}^{(i)}) \delta(x_{0:k} - x_{0:k}^{(i)})$$

在重要性概率密度函数中采样粒子，将采样的结果加权逼近后验概率

$$w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^N \tilde{w}_k^{(j)} \quad \text{其中} \quad \tilde{w}_k^{(i)} = \frac{p(y_{1:k}|x_{0:k}^{(i)})p(x_{0:k}^{(i)})}{q(x_{0:k}^{(i)}|y_{1:k})}$$

观测似然度

# 粒子滤波跟踪算法

## □ 粒子滤波基本思想

引入序贯重要性采样，采用递推的形式计算重要性权值，减少计算量：

$$q\left(x_{0:k}^{(i)}|y_{1:k}\right)=q\left(x_{0:k-1}^{(i)}|y_{1:k-1}\right) \cdot q\left(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k}\right)$$

假设系统符合马尔可夫过程，观测变量相互独立，权值递推公式如下：

$$\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{\overbrace{p\left(y_k|x_k^{(i)}\right)p\left(x_k^{(i)}|x_{k-1}^{(i)}\right)}^{\text{状态转移概率}}}{\underbrace{q\left(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k}\right)}_{\text{观测似然度}}}$$

随着时间增加，可能发生粒子权值的退化现象，即某一权值趋于1而其余的权值趋于0，权值的退化减少了可用有效样本数度量，在适当情况下重采样粒子。

$$N_{\text{eff}} = \left( \sum_{i=1}^N \left( w_k^{(i)} \right)^2 \right)^{-1}$$

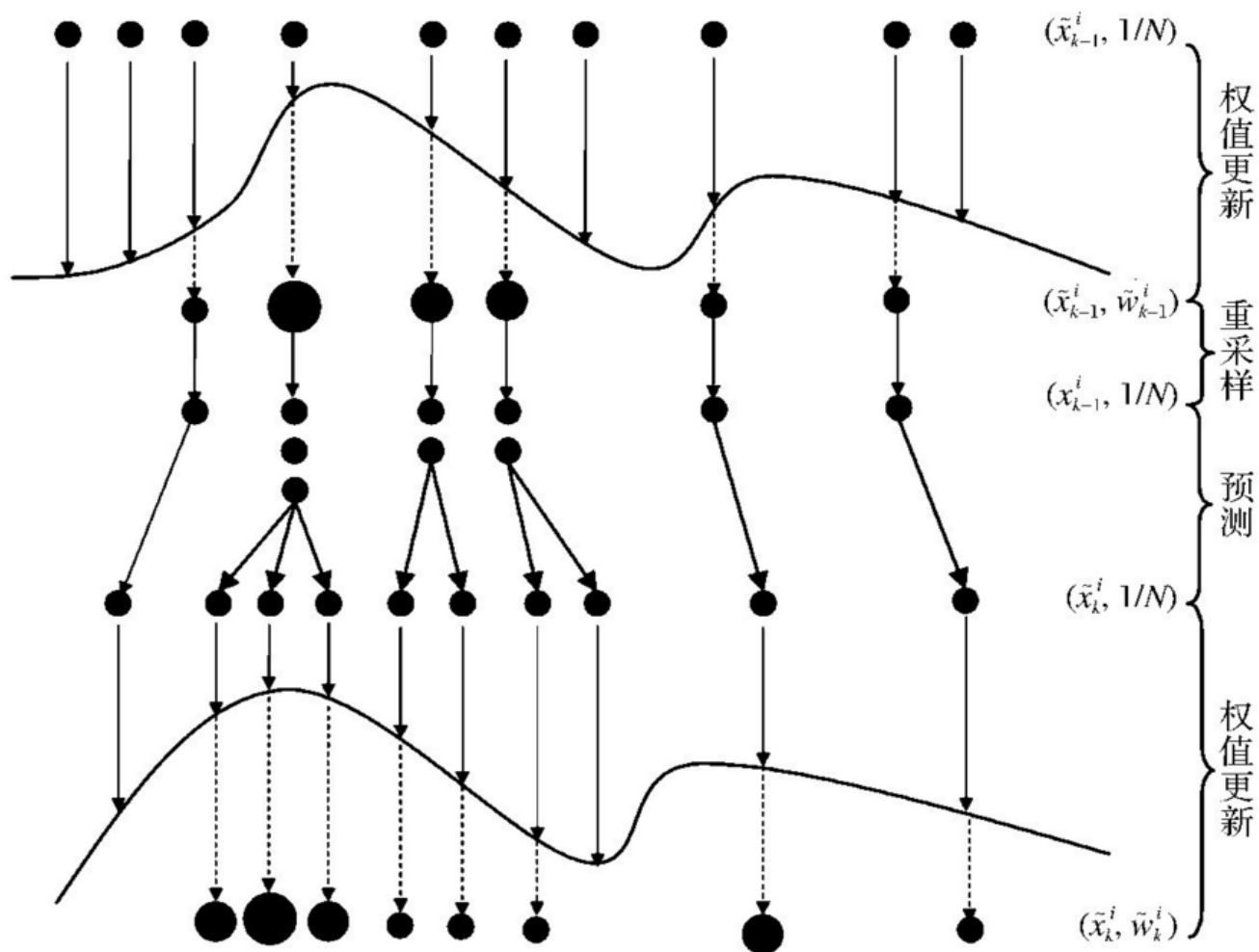


# 粒子滤波跟踪算法

## □ 粒子滤波跟踪时的特征选择

- 对目标状态的传播进行“假设”之后，需要利用得到的  $t$  时刻的 **观测量** 对其进行验证。因此“系统状态转移-系统观测”可以理解成“**假设-验证**”。
- 所谓 **观测量**，最直观的是指视频图像，可以使灰度图像也可以是处理后的特征量，如颜色、轮廓特征等。
- 使用观测量对系统状态转移的结果进行 **验证**，实际上是一个 **相似性度量** 的过程。
- 由于每个粒子代表目标状态的一个可能性，则系统观测的目的就是使实际情况相近的粒子获得的权值大一些，与实际情况相差较大的粒子获得的权重小一些。

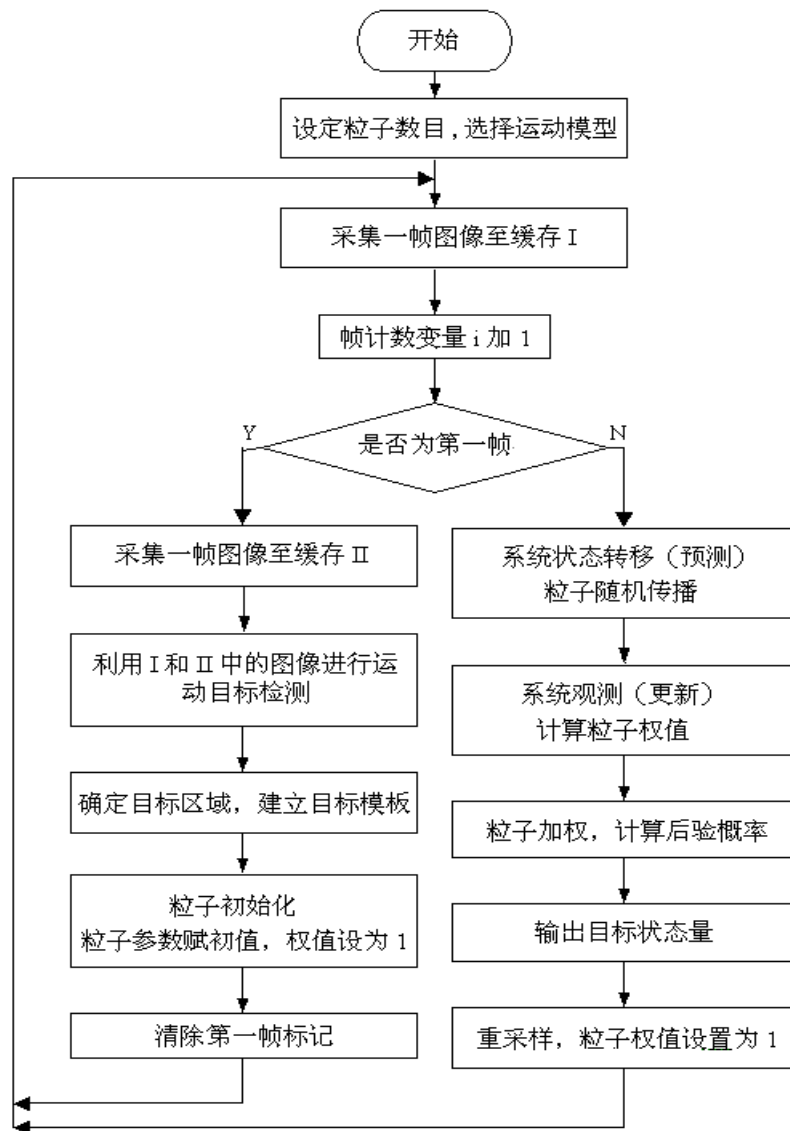
# 粒子滤波跟踪算法





# 粒子滤波跟踪算法

## 粒子滤波流程



# 粒子滤波跟踪算法

## □ 粒子滤波流程

在目标跟踪中，每个粒子代表一个图像块，它们的状态  $x$  代表该图像块的位置、尺寸等信息。它们的标签  $y$  (如0或1) 代表是否是目标

(1) 初始化:

确定粒子数目，选择运动模型，确定目标初始位置。

(2) 序贯重要性采样:

实际中，通常认为相邻帧目标运动是平滑的。当前帧中，我们在上一帧目标位置处按照一定模型采样（如高斯模型、均匀采样等）

根据  $x_k^{(i)} \sim q(x_k^{(i)} | x_{0:k-1}^{(i)}, y_{1:k})$  采样  $N$  个新粒子  $\{x_k^{(i)}\}_{i=1}^N$ 。

计算粒子权值:  $\tilde{w}_k^{(i)} = w_{k-1}^{(i)} \frac{p(y_k | x_k^{(i)}) p(x_k^{(i)} | x_{k-1}^{(i)})}{q(x_k^{(i)} | x_{0:k-1}^{(i)}, y_{1:k})}$

归一化粒子权值:  $w_k^{(i)} = \tilde{w}_k^{(i)} / \sum_{j=1}^N \tilde{w}_k^{(j)}$

实际中，粒子权重主要依赖外观模型衡量观测似然度。观测模型估计该粒子和第一帧目标的相似度。观测模型可采用直方图相似性等方案

# 粒子滤波跟踪算法

## □ 粒子滤波流程

(3) 重采样：

计算有效粒子个数:  $N_{\text{eff}} = \left( \sum_{i=1}^N \left( w_k^{(i)} \right)^2 \right)^{-1}$

如果  $N_{\text{eff}} < N_{\text{th}}$ ，则增加有效粒子个数，删减无效粒子。

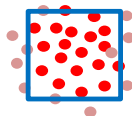
(4) 状态估计:  $\hat{x}_k = \sum_{i=1}^N w_k^{(i)} x_k^{(i)}$

根据每个粒子 (图像块) 的重要程度 (权重), 将他们的状态 (图像块位置, 尺寸) 进行加权求和。

# 粒子滤波跟踪算法

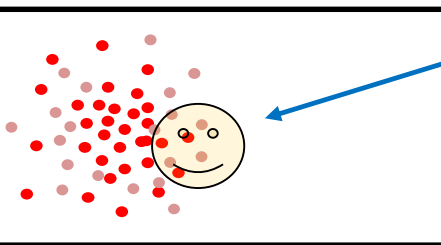
## □ 粒子滤波跟踪结果示例

第  $k-1$  帧



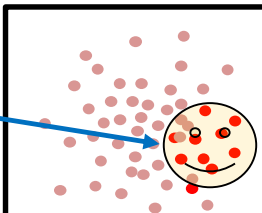
初始帧或上一帧，目标位置及粒子分布情况

第  $k$  帧

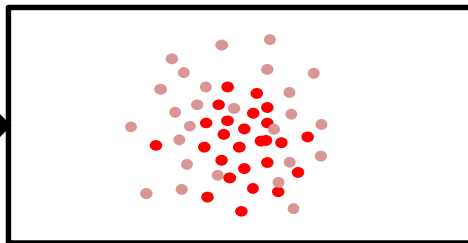


下一帧中，目标发生运动。我们先在上一帧位置处采样大量粒子。

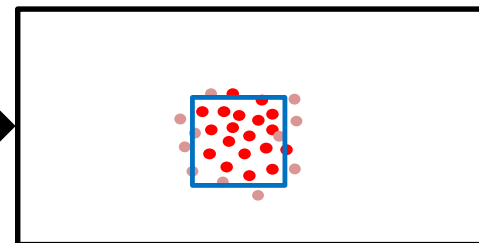
计算粒子权重，  
红色粒子权重  
更大



粒子重采样



估计目标新的状态



# 粒子滤波跟踪算法

## □ 粒子滤波跟踪的实际场景下跟踪结果



粒子滤波



# 粒子滤波跟踪算法

## □ 粒子滤波小结

- 粒子滤波核心思想是利用状态空间的一组带权值的随机样本逼近状态变量的概率密度函数，其显著优点是**多假设性**，不受先验分布及状态转移模型的限制，方便地对目标状态进行表达。
- 该方法需要用**大量的样本数量**才能很好地近似系统的后验概率密度，算法的复杂度较高。
- 此外，相似背景，遮挡，外观变化等会对粒子滤波的性能产生较大影响。





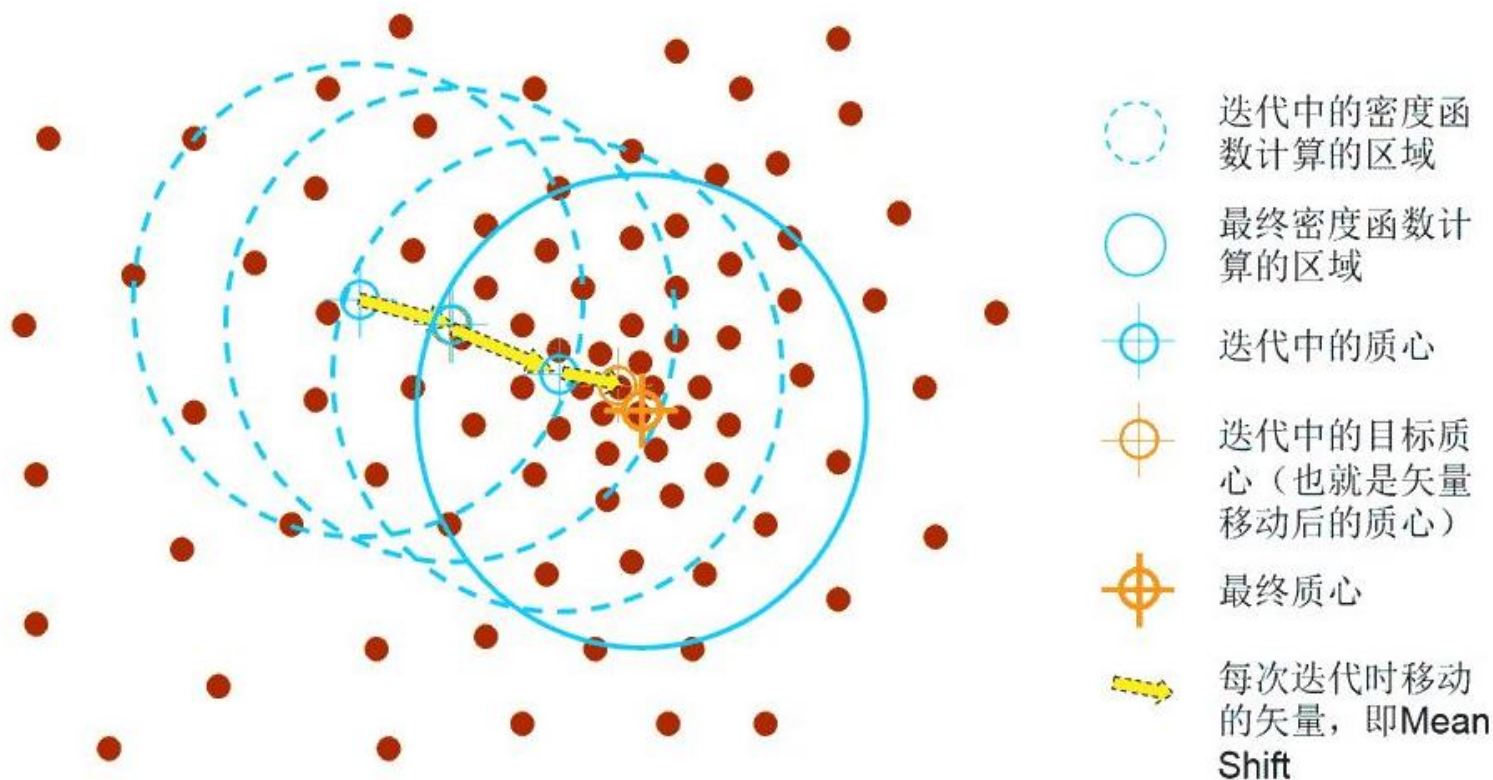
# 均值漂移算法

## □ 均值漂移 (MeanShift) 简介:

- MeanShift 算法是 Fukunaga 于 1975 年提出的。
- 直到 1995 年, Yizong Cheng 针对离  $x$  越近的采样点对  $x$  周围的统计特性越有效, 定义了一族核函数。同时他认为所有的样本点重要性不同, 设定了一个权重系数, 扩大了 MeanShift 的使用范围。
- MeanShift 基本思想: 利用概率密度的梯度爬升来寻找局部最优。

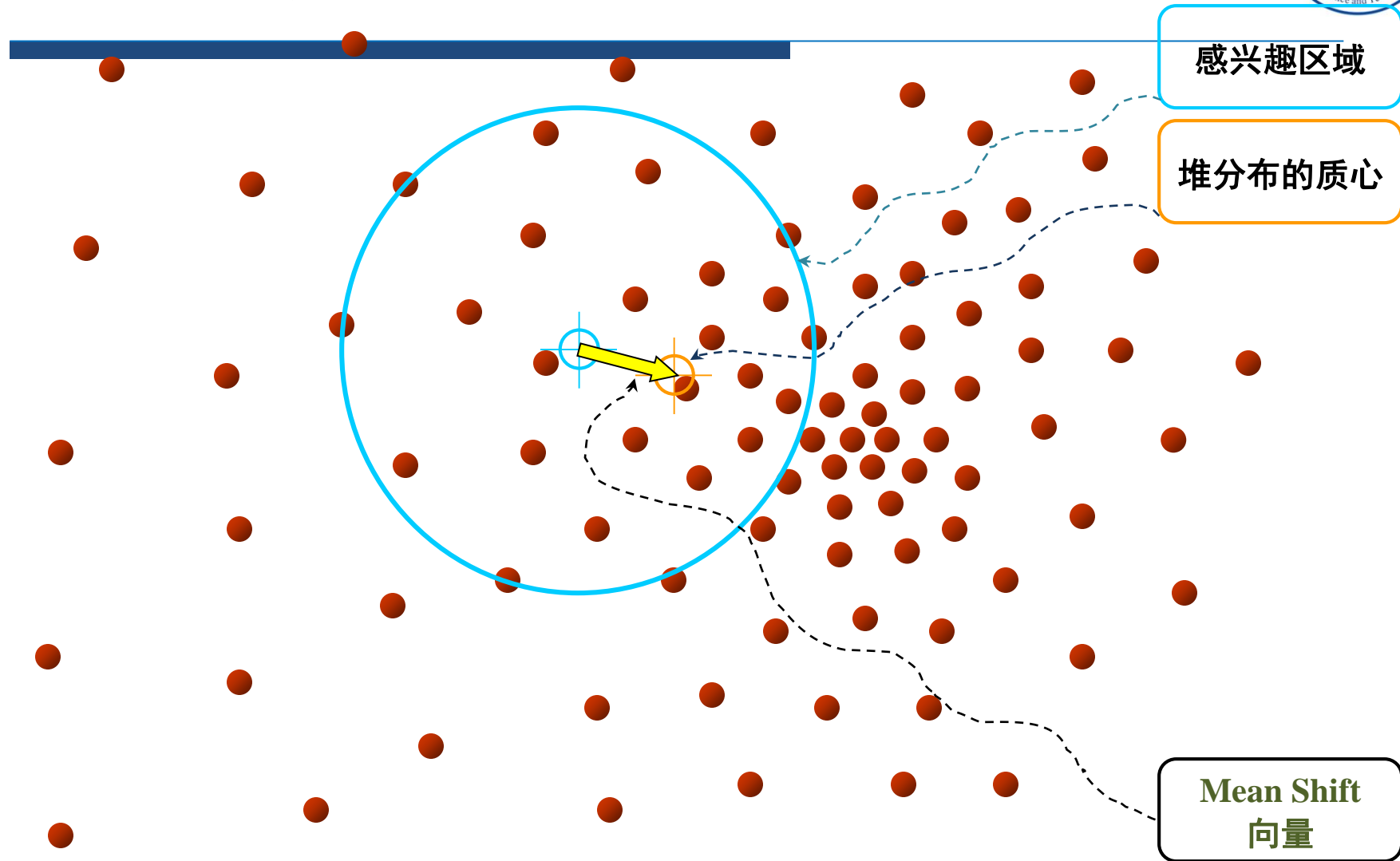
# 均值漂移算法过程

- MeanShift 向量逐步漂移到局部密度最大点并停止，达到跟踪目的：



目的：找出最密集的区域

# 直观图示



目标：找到密度最大的区域  
桌面上相同的弹球的分布

# 直观图示

感兴趣区域

堆分布的质心

Mean Shift  
向量

目标：找到密度最大的区域  
桌面上相同的弹球的分布

# 直观图示

感兴趣区域

堆分布的质心

Mean Shift  
向量

目标：找到密度最大的区域  
桌面上相同的弹球的分布

# 直观图示

感兴趣区域

堆分布的质心

Mean Shift  
向量

目标：找到密度最大的区域  
桌面上相同的弹球的分布



# 直观图示

感兴趣区域

堆分布的质心

Mean Shift  
向量

目标：找到密度最大的区域  
桌面上相同的弹球的分布

# 直观图示

感兴趣区域

堆分布的质心

Mean Shift  
向量

目标：找到密度最大的区域  
桌面上相同的弹球的分布

# 直观图示

感兴趣区域

堆分布的质心

目标：找到密度最大的区域  
桌面上相同的弹球的分布

# MeanShift 算法

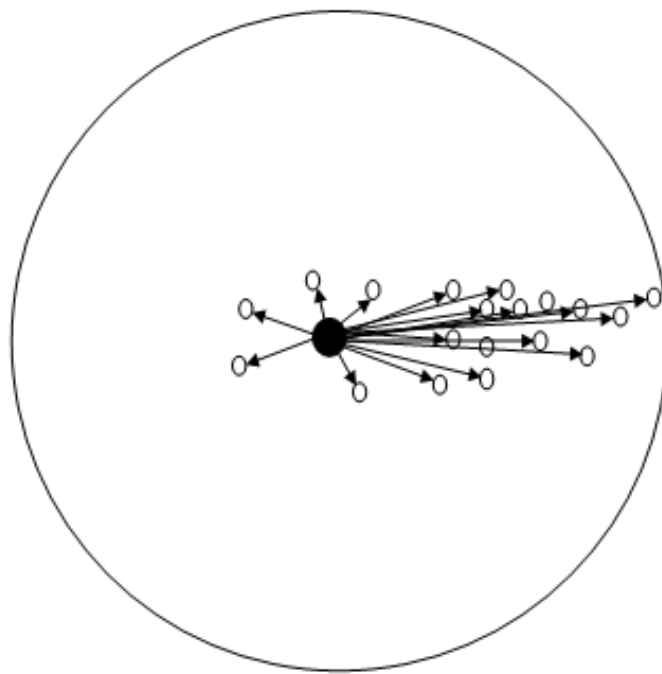
## □ 均值漂移 (MeanShift) 原理:

给定  $d$  维空间中的  $n$  个样本点  $x_i$  ( $i = 1, \dots, n$ ), 在  $x$  点的 MeanShift 向量的基本形式定义为:

$$M_h(x) = \left( \frac{1}{k} \sum_{x_i \in S_h} x_i \right) - x = \frac{1}{k} \sum_{x_i \in S_h} (x_i - x)$$

其中,  $S_h$  是一个半径为  $h$  的高维球区域,  $k$  表示  $n$  个样本点中有  $k$  个点落入区域  $S_h$  中。

直观地, MeanShift 向量表示区域中  $k$  个样本点相对于点  $x$  求偏移向量再平均。该向量指向概率密度梯度的方向。

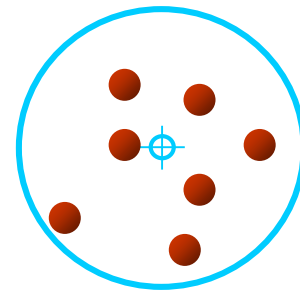


# 扩展的 MeanShift

## □ 引入两个参数

- 核函数
- 权重

$$M_h(x) = \frac{1}{k} \sum_{x_i \in S_h} (x_i - x)$$



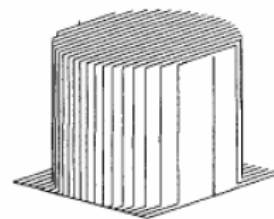
## □ 核函数

- 定义:  $x \in R^D$ ,  $\|x\|^2 = x^T x$ 。若函数  $K(x)$  存在一个剖面函数:  $k: [0, \infty) \rightarrow R$ , 即  $K(x) = k(\|x\|^2)$ , 并且  $k(r)$  满足:

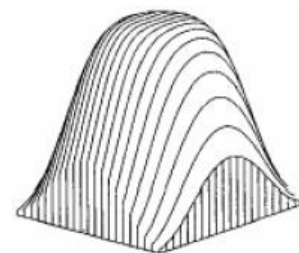
- ✓ 非负的
- ✓ 非增的
- ✓ 分段连续的, 且  $\int_0^\infty k(r) dr < \infty$

### ■ 常用核函数

- ✓ 单位均匀核函数
- ✓ 单位高斯核函数



$$K(x) = \begin{cases} 1, & \text{if } \|x\| < 1 \\ 0, & \text{if } \|x\| \geq 1 \end{cases}$$



$$K(x) = e^{-\|x\|^2}$$



# 扩展的 MeanShift

Mean Shift 扩展形式：

$$m(x) = \frac{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) w(x_i)(x_i - x)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) w(x_i)}$$

简化运算：

$$m(x) = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) w(x_i)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) w(x_i)} - x$$

Mean Shift向量的方向和样本点的概率密度函数的梯度方向是一致的



# 核密度梯度估计

□ 给定 $D$ 维空间 $n$ 个样本点 $x_i$ ,  $f(x)$ 的核密度估计(也称parzen估计)为

$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_H(x_i - x)$ 。对核函数进行简化并引入权重, 可得:

$$\hat{f}(x) = \frac{\sum_{i=1}^n k\left(\left\|\frac{x_i - x}{h}\right\|^2\right) w(x_i)}{h^d \sum_{i=1}^n w(x_i)}$$

假设: 所有的样本点  
从同一个函数采集

□ 概率密度核函数 $f(x)$ 的梯度估计 $\nabla \hat{f}(x)$ 为

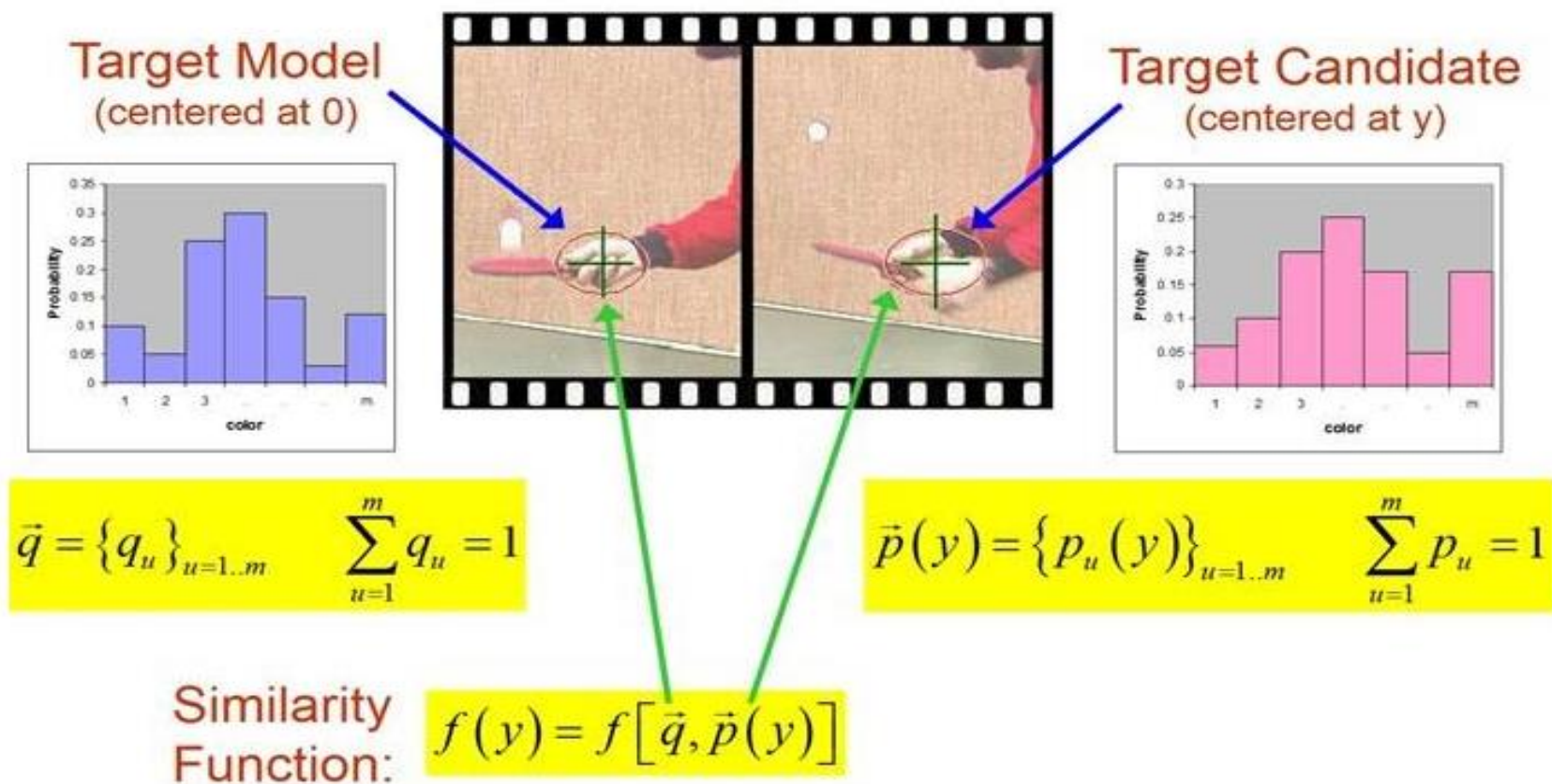
$$\nabla f(x) \approx \nabla \hat{f}(x) = \frac{2 \sum_{i=1}^n (x_i - x) k' \left( \left\| \frac{x_i - x}{h} \right\|^2 \right) w(x_i)}{h^{2+d} \sum_{i=1}^n w(x_i)}$$

$$= \frac{2}{h^2} \left[ \frac{\sum_{i=1}^n g\left(\left\|\frac{x_i - x}{h}\right\|^2\right) w(x_i)}{h^d \sum_{i=1}^n w(x_i)} \right] \left[ \frac{\sum_{i=1}^n g\left(\left\|\frac{x_i - x}{h}\right\|^2\right) w(x_i) x_i}{\sum_{i=1}^n g\left(\left\|\frac{x_i - x}{h}\right\|^2\right) w(x_i)} - x \right]$$

$$\begin{aligned} G(x) &= g(\|x\|^2) \\ g(x) &= -k'(x) \end{aligned}$$

# 均值漂移目标跟踪算法 (TPAMI-03)

## □ 均值漂移 (Mean Shift) 跟踪 (生成模型) <sup>[1]</sup>:



- Comaniciu D, Ramesh V, Meer P. Kernel-based object tracking[J]. IEEE Transactions on pattern analysis and machine intelligence, 2003, 25(5): 564-577.



# 均值漂移目标跟踪算法

核函数。图像  
块的中心像素  
的权重更大

## □ 目标核函数直方图

$$\hat{q}_u = C \sum_{i=1}^n k(\|x_i\|^2) \delta[b(x_i) - u]$$

统计直方图

其中  $C = 1 / \sum_{i=1}^n k(\|x_i\|^2)$  为归一化常数,  $k$  表示核函数(在核估计中通常是平滑作用), 目标区域共  $n$  个像素点  $\{x_i\}_{i=1, \dots, n}$ , 该区域颜色分布离散成  $m$  级,  $b(x_i)$  表示像素点  $x_i$  的量化值。

## □ 候选（待跟踪目标）核函数直方图

$$\hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u]$$

其中  $C_h = 1 / \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right)$ , 候选目标区域  $\{x_i\}_{i=1, \dots, n_h}$ , 该区域的中心位置为  $y$ ,  $h$  表示核函数  $k$  的窗宽。其余变量物理意义同上。

# 均值漂移目标跟踪算法

- 通过Bhattacharyya系数度量候选样本 (中心为  $y$  的图像块) 和目图像标核函数直方图的相似度:

$$\hat{\rho}(y) = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u}$$

- Taylor 展开, 将  $\hat{p}_u(y)$  代入并化简得到:

$$\hat{\rho}(y) \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k \left( \left\| \frac{y - x_i}{h} \right\|^2 \right)$$

$$\text{其中 } w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \delta[b(x_i) - u]$$

$y_0$  是MeanShift迭代的起点位置,  
在跟踪中, 通常是目标在上一帧的位置

由于第一项为常量, 因此我们最大化  $\hat{\rho}(y)$ , 本质上寻找新的质心  $y$  使得候选区域和模板的相似程度最大化

# 均值漂移目标跟踪算法

- MeanShift 方法求解目标的新位置  $y_1$ :

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)} \quad \text{其中 } g(x) = -k'(x)$$

- 目标跟踪中通常使用 Epanechnikov 作为核函数，简单地讲，即计算空间任意一点到中心位置的欧氏距离。
- 在跟踪问题中，核函数起到了位置加权的作用，即更加相信中心位置像素点的信息。在单位均匀核的情况下， $g(x)$ 是常数，上式可以简化为：

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i}$$



# 均值漂移目标跟踪算法

## □ MeanShift 实现目标跟踪流程：

1. 在当前帧，计算候选目标的特征 (候选图像块的颜色直方图)

2. 计算候选目标与初始目标的相似度：  $\hat{\rho}(y_0) = \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u}$

3. 计算权值  $\{w\}_{i=1,2,\dots,m}$

4. 利用 MeanShift 算法，计算目标新位置：

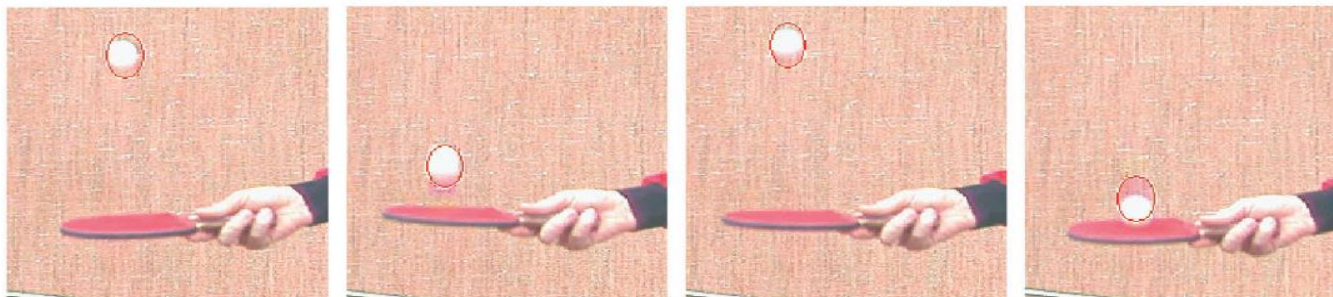
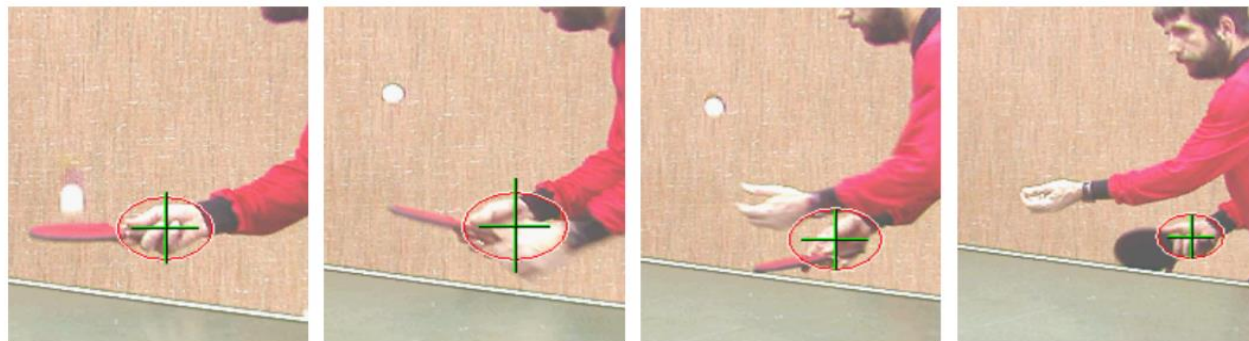
$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y_0 - x_i}{h}\right\|^2\right)}$$

5. 若  $\|y_1 - y_0\| < \varepsilon$ ，则停止，否则  $y_0 \leftarrow y_1$  转步骤 3

限制条件：新目标中心需位于原目标中心附近

# 均值漂移目标跟踪算法

## □ 实验结果示例：





# 均值漂移目标跟踪算法

## □ MeanShift 跟踪算法小结：

### □ 优点：

- 无参密度估计的方法，计算量小，能满足实时性的要求
- 基于核函数直方图的建模对目标的旋转形变鲁棒性好

### □ 缺点：

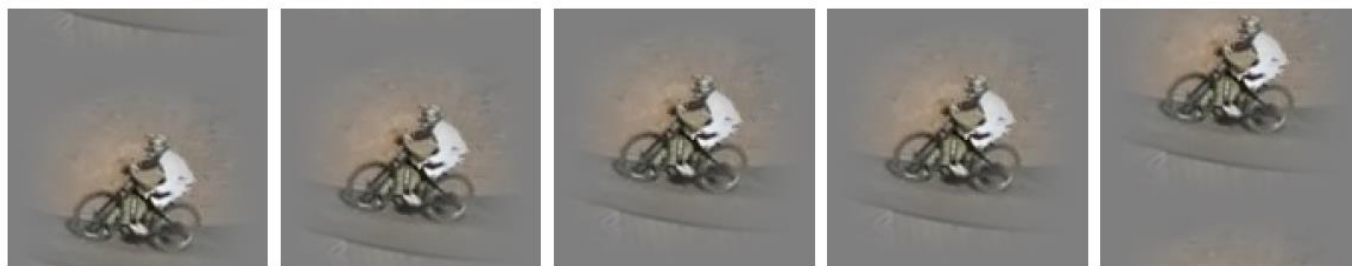
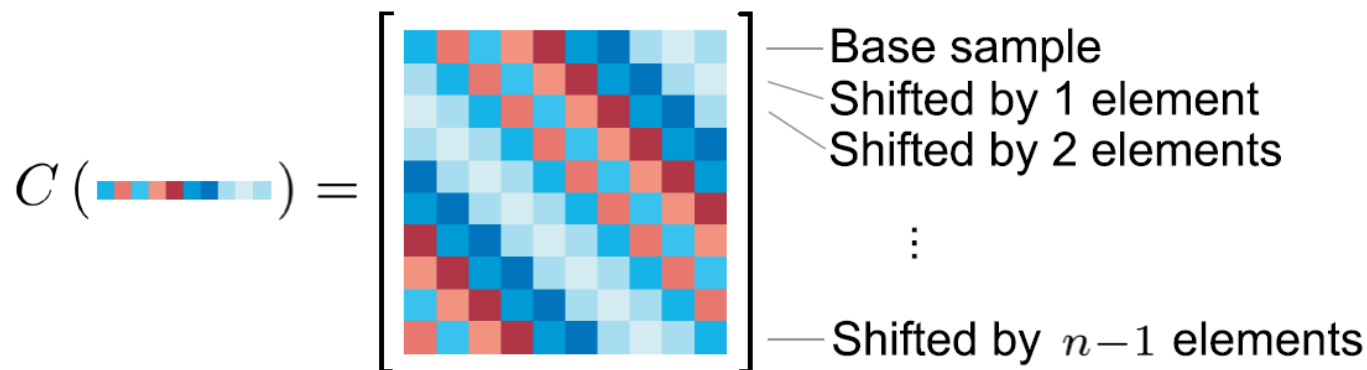
- 仅采用颜色直方图，易受具有相似颜色的背景物体干扰
- 缺乏必要的模板更新过程
- 目标运动速度快或严重遮挡时，跟踪易发生漂移

# 相关滤波器跟踪算法 (TPAMI-15)

## □ 相关滤波器 (Correlation Filter, CF) : 判别式跟踪算法

- 通过对感兴趣区域的水平和垂直方向圆周移位, 可以获取大量的训练样本。通过训练这些样本获取滤波器用于跟踪。

## □ 训练样本示例:



+30

+15

Base sample

-15

-30





# 相关滤波器跟踪算法

- 岭回归的目标在于学习映射函数  $f(z) = w^T z$  来最小化如下的均方误差

$$\min_w \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2$$

- 岭回归具有如下的**闭合解**<sup>[4]</sup>:

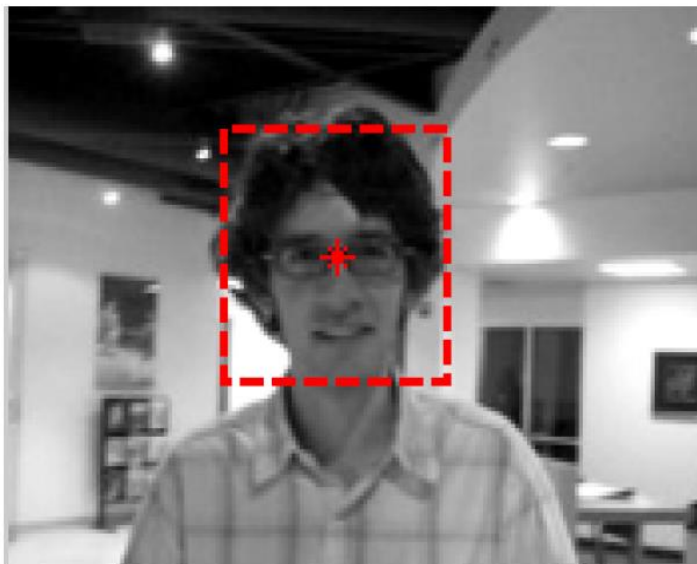
$$w = (X^T X + \lambda I)^{-1} X^T y$$

- 其中  $X$  为样本矩阵，每一行为一个样本  $x_i$
- 上述闭合解的**复杂度**在于**矩阵求逆**。
- 相关滤波器 (Correlation Filter)，源于岭回归，通过**频域求解以及循环矩阵的性质**巧妙避免上述矩阵求逆过程。



# 相关滤波器跟踪算法

## □ 训练样本示例：



- 样本的标签在 $[0, 1]$ 范围内，通过高斯函数由移位距离确定。
  - 标签较大的样本，由于其循环移位很小，视为正样本 (绿色样本)。
  - 标签较小的样本，由于其循环移位很大，视为负样本 (红色样本)。

# 相关滤波器跟踪算法

□ 通过将单张图片特征进行循环移位，数据矩阵  $X$  有如下形式：

$$X = C(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_n \\ x_n & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_n & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_2 & x_3 & x_4 & \cdots & x_1 \end{bmatrix}$$

通过构造上述的数据矩阵，目的在于极大加速岭回归的求解。

$$\mathbf{w} = (X^T X + \lambda I)^{-1} X^T \mathbf{y}.$$



# 相关滤波器跟踪算法

$$w = (X^T X + \lambda I)^{-1} X^T y$$

我们将上述的闭合解转化成复数域, 如下:

$$w = (X^H X + \lambda I)^{-1} X^H y$$

其中 $H$ 代表共轭转置,  $*$ 代表复共轭, 即 $X^H = (X^*)^T$

此外, 由于矩阵 $X$ 是循环的, 可以通过离散傅里叶变换(DFT)对角化:

$$X = F \text{diag}(\hat{x}) F^H$$

其中 $\hat{\cdot}$ 表示傅里叶变化, 即 $\hat{x} = \mathcal{F}(x)$

$F$ 为DFT矩阵, 即 $\mathcal{F}(z) = \frac{1}{\sqrt{n}} F z$



# 相关滤波器跟踪算法

基于  $w = (X^T X + \lambda I)^{-1} X^T y$  和  $X = F \text{diag}(\hat{x}) F^H$

有了如上的准备工作，我们可以逐步得到岭回归的频域闭合解：

$$X^H X = F \text{diag}(\hat{x}^*) F^H F \text{diag}(\hat{x}) F^H$$

$$X^H X = F \text{diag}(\hat{x}^*) \text{diag}(\hat{x}) F^H$$

$$X^H X = F \text{diag}(\hat{x}^* \odot \hat{x}) F^H$$

$$w = (F \text{diag}(\hat{x}^* \odot \hat{x}) F^H + \lambda F^H I F)^{-1} X^H y$$

$$= (F \text{diag}(\hat{x}^* \odot \hat{x} + \lambda)^{-1} F^H) X^H y$$

$$= F \text{diag}(\hat{x}^* \odot \hat{x} + \lambda)^{-1} F^H F \text{diag}(\hat{x}^*) F^H y$$

$$= F \text{diag} \left( \frac{\hat{x}^*}{\hat{x}^* \odot \hat{x} + \lambda} \right) F^H y$$



# 相关滤波器跟踪算法

$$Fw = \text{diag} \left( \frac{\hat{x}^*}{\hat{x}^* \odot \hat{x} + \lambda} \right) Fy$$

$$\hat{w} = \text{diag} \left( \frac{\hat{x}^*}{\hat{x}^* \odot \hat{x} + \lambda} \right) \hat{y}$$

由于  $\text{diag}(\cdot)$  矩阵和向量间的矩阵乘法可替代成元素间乘法 (element-wise product), 上式化简成如下表达:

$$\hat{w} = \frac{\hat{x}^* \odot \hat{y}}{\hat{x}^* \odot \hat{x} + \lambda}$$



完全避免了矩阵求逆,  
元素间的乘积极其高效

# 相关滤波器跟踪算法

## □ 检测过程：

- 当相关滤波器学习完成后。给出一个新的数据矩阵  $Z$ ，我们可以通过如下预测其回归值 (response map)：

时域  $r(w, Z) = Zw$  或频域结果  $\hat{r}(w, Z) = \hat{z} \odot \hat{w}$

- **左式：**我们依旧可以在时域进行求解，其中  $Z$  (大写) 是基样本  $z$  的循环移位矩阵。该矩阵和滤波器  $w$  进行相乘即可得到搜索样本的预测值。
- **右式：**我们同样可以在频域进行快速求解，从而避免手工地对搜索样本  $z$  进行循环移位以得到  $Z$ 。注意，右式求得的响应值是频域的，需要进行反变换回到时域。