

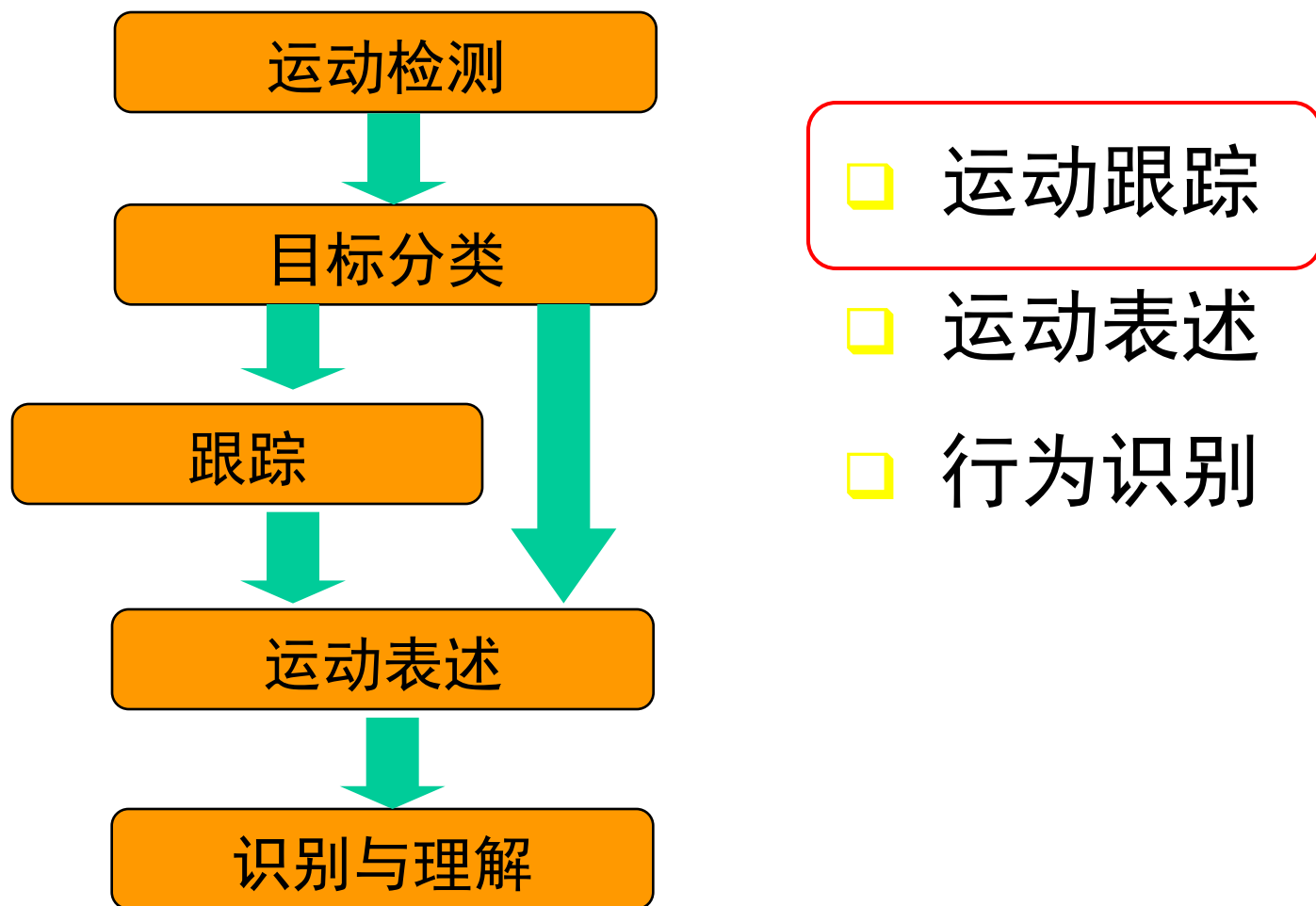
运动跟踪

董秋雷

中国科学院自动化研究所

qldong@nlpr.ia.ac.cn

运动分析的一般流程



什么是跟踪(Tracking)?

- 所谓跟踪，就是指在图像序列中持续地估计出感兴趣的运动目标所在区域（位置），形成运动目标的运动轨迹；有时还需要估计出运动目标的某些运动参数（比如速度、加速度等）。

跟踪问题分类

- 场景中运动目标的数目： 单运动目标 vs. 多运动目标
 - 在多目标跟踪过程中，必须考虑到多个目标在场景中会互相遮挡(Occlusion)，合并(Merge)，分离(Split)等情况。
 - 多目标跟踪中的数据关联问题(Data Association)。



跟踪问题分类

- 摄像机的数目： 单摄像机 vs. 多摄像机
 - 使用多摄像机有望解决因相互遮挡导致的运动目标丢失问题，但多摄像机的信息融合是一个关键性问题。
- 摄像机是否运动： 摄像机静止 vs. 摄像机运动
 - 摄像机的运动形式，一种是摄像机支架固定，但摄像机可以偏转(Pan)，俯仰(Tilt)以及缩放(Zoom)；另一种是摄像机装在某个移动的载体上，如移动的车辆或飞行的飞机。
 - 摄像机的运动增加了运动目标检测的难度。

跟踪问题分类

- 场景中运动目标的类型： 刚体 vs. 非刚体
 - 交通车辆—刚体； 人—非刚体。
- 传感器的种类： 可见光图像 vs. 红外图像
 - 白天使用可见光图像；晚上使用红外图像。

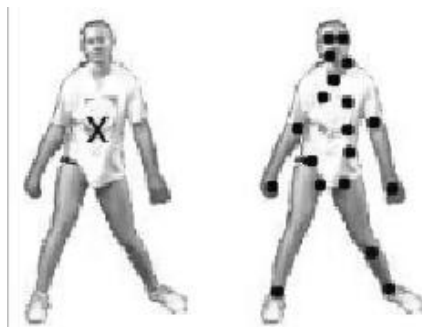
- ❑ **运动目标的表示方法**
- ❑ **具体介绍常用的跟踪方法**
- ❑ **多目标跟踪的数据关联问题**
- ❑ **多摄像机跟踪**

运动目标的表示方法

- ❑ 基于点的跟踪
- ❑ 基于区域的跟踪
- ❑ 基于轮廓的跟踪
- ❑ 基于模型的跟踪

基于点的跟踪

- 质心或一组特征点集

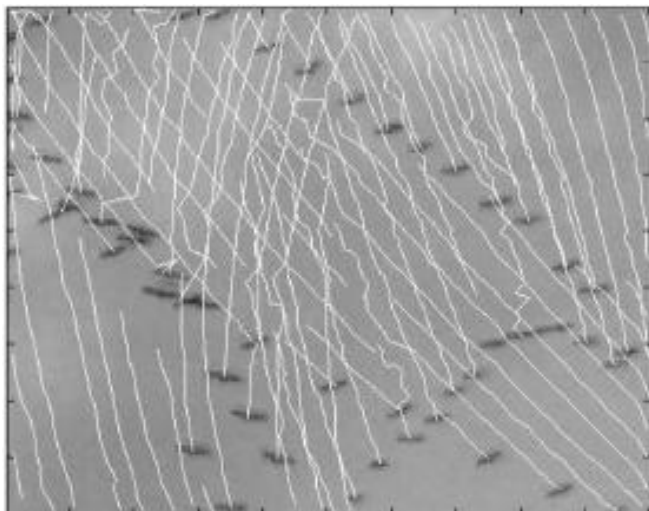


- 运动轮廓的角点



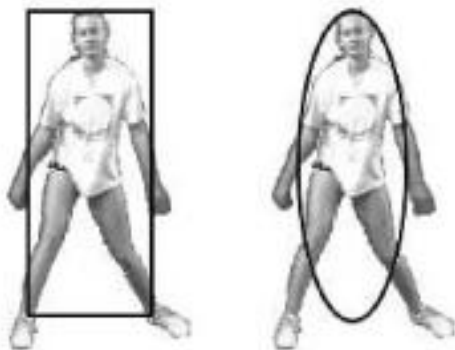
基于点的跟踪

- 将运动目标表示成点比较适合跟踪那些在图像上所占区域比较小的目标
- 做简单刚体运动的目标（特征点集）



基于区域的跟踪

- 将运动目标用比较简单的几何形状表示，比如矩形或椭圆等

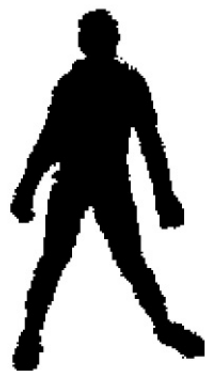


- 适合于表示简单的刚体或非刚体运动目标。
- 相较于后面要介绍的活动轮廓等表示方法精度较差。

基于轮廓的跟踪



Contour

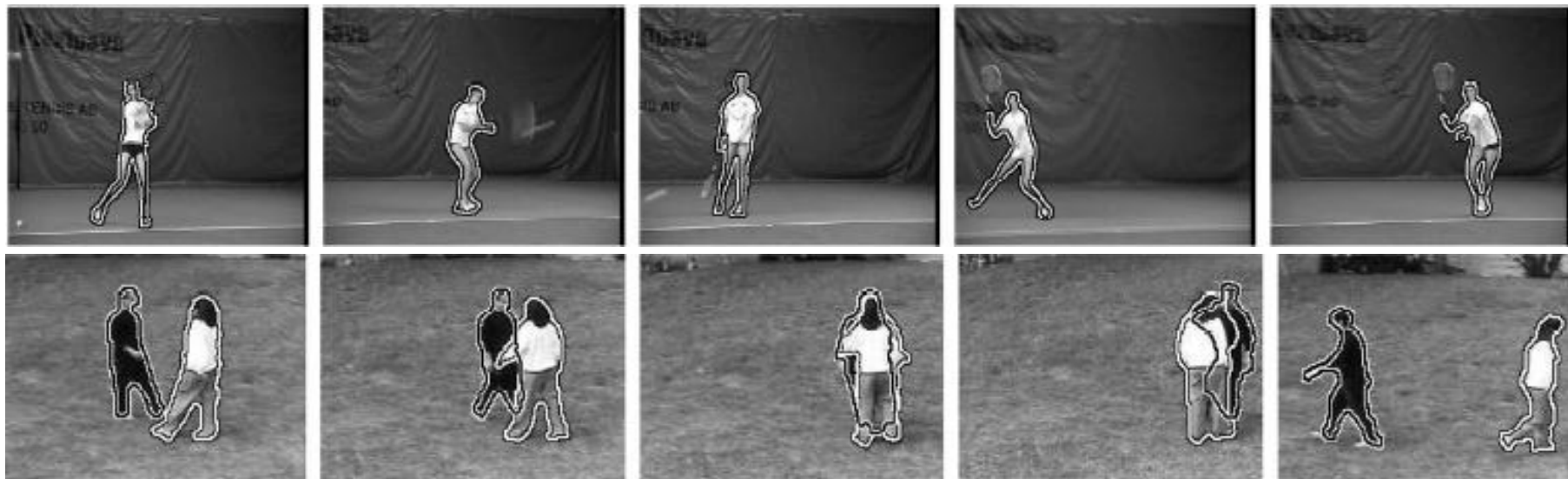


Silhouette

- ❑ Contour表示运动目标的外部轮廓
- ❑ Silhouette表示运动目标外部轮廓内的区域
- ❑ 适用于表示复杂的非刚体运动目标

基于轮廓的跟踪

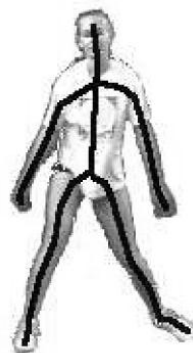
- 主动轮廓 Active Contour
- 利用封闭的曲线轮廓来表示运动目标，并且该轮廓能够自动连续地更新



* YILMAZ, A., LI, X., AND SHAH, M. 2004. Contour based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans. Patt. Analy. Mach. Intell. 26, 11, 1531–1536.

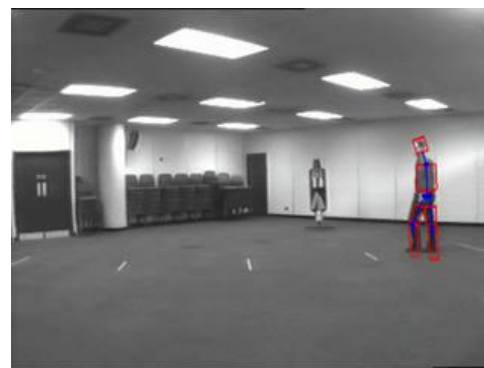
基于模型的跟踪

□ 二维形状模型



Skeletal Articulated Model

□ 立体模型 Volumetric Model



运动目标的表示方法

- ❑ 基于点的跟踪
- ❑ 基于区域的跟踪
- ❑ 基于轮廓的跟踪
- ❑ 基于模型的跟踪

由
简
到
繁



- 采用上述的哪种方法来表示运动目标和不同的应用场合、运动目标的运动特性、以及对跟踪算法的精度要求等密切相关。

- ❑ 运动目标的表示方法
- ❑ 具体介绍常用的跟踪方法
- ❑ 多目标跟踪的数据关联问题
- ❑ 多摄像机跟踪

跟踪问题的两种解决思路*

- 自底向上 (Bottom-up) 的处理方法
 - 又称为数据驱动 (Data-driven) 的方法, 不依赖于先验知识
- 自顶向下 (Top-down) 的处理方法
 - 又称为模型驱动 (Model-driven) 的方法, 依赖于所构建的模型或先验知识

* Comanniciu D, Ramesh V, Meer P. **Kernel-based object tracking**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(5): 564~577

跟踪问题的两种解决思路*

□ 自底向上 (Bottom-up) 的处理方法

→ 模板匹配 (Template Match)

→ 均值漂移 (Mean Shift)

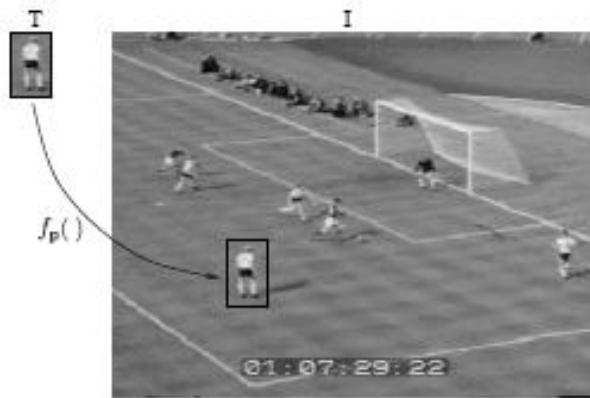
□ 自顶向下 (Top-down) 的处理方法

→ 卡尔曼滤波器 (Kalman Filter)

→ 粒子滤波器 (Particle Filter)

自底向上 (Bottom-up) 的处理方法

❑ 模板匹配法 (Template Matching)



- 在上一帧图像中目标位置（或模板 T 位置）为： (x, y)

- 在当前帧搜寻位置

$$(x', y') = (x + dx, y + dy)$$

使得

$$\arg \max_{dx, dy} \text{cov}(T(x, y), I(d + dx, y + dy))$$

- ❖ 概念上相对比较简单
- ❖ 进行穷尽的搜索计算量非常大

□ 在前一帧目标位置附近的一个小区域内搜索

- ❖ FIEGUTH, P. AND TERZOPOULOS, D. 1997. **Color-based tracking of heads and other mobile objects at video frame rates.** In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 21–27

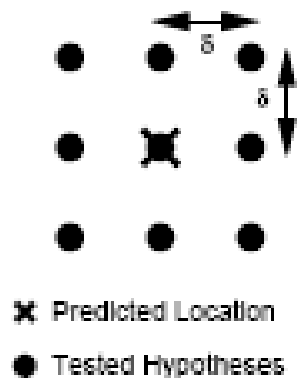


Figure 2. The nine lattice points which determine the local hypotheses to be tested for each object at each frame.

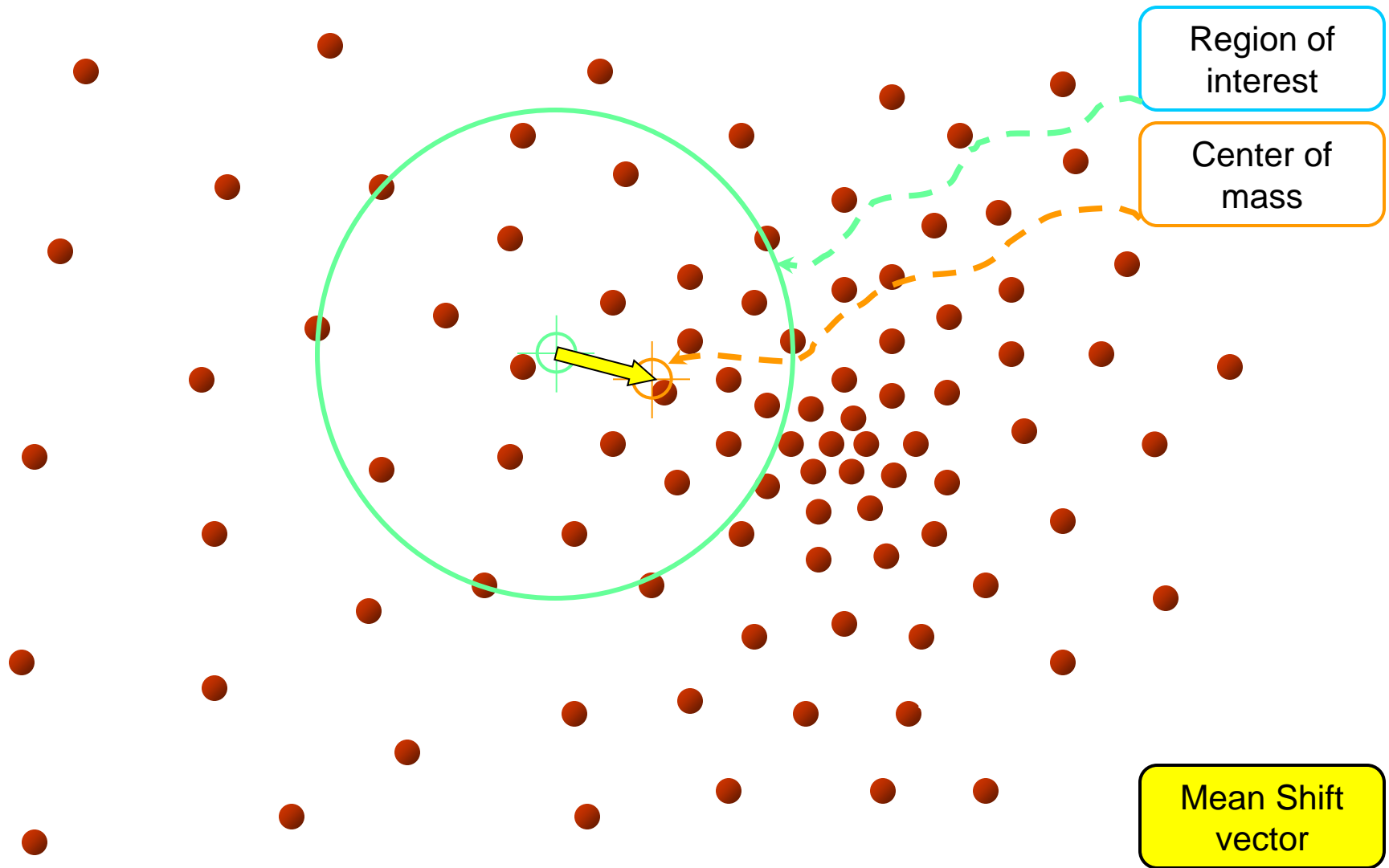
□ 使用模板匹配的快速算法

- ❖ SCHWEITZER, H., BELL, J. W., AND WU, F. 2002. **Very fast template matching.** In European Conference on Computer Vision (ECCV). 358–372.

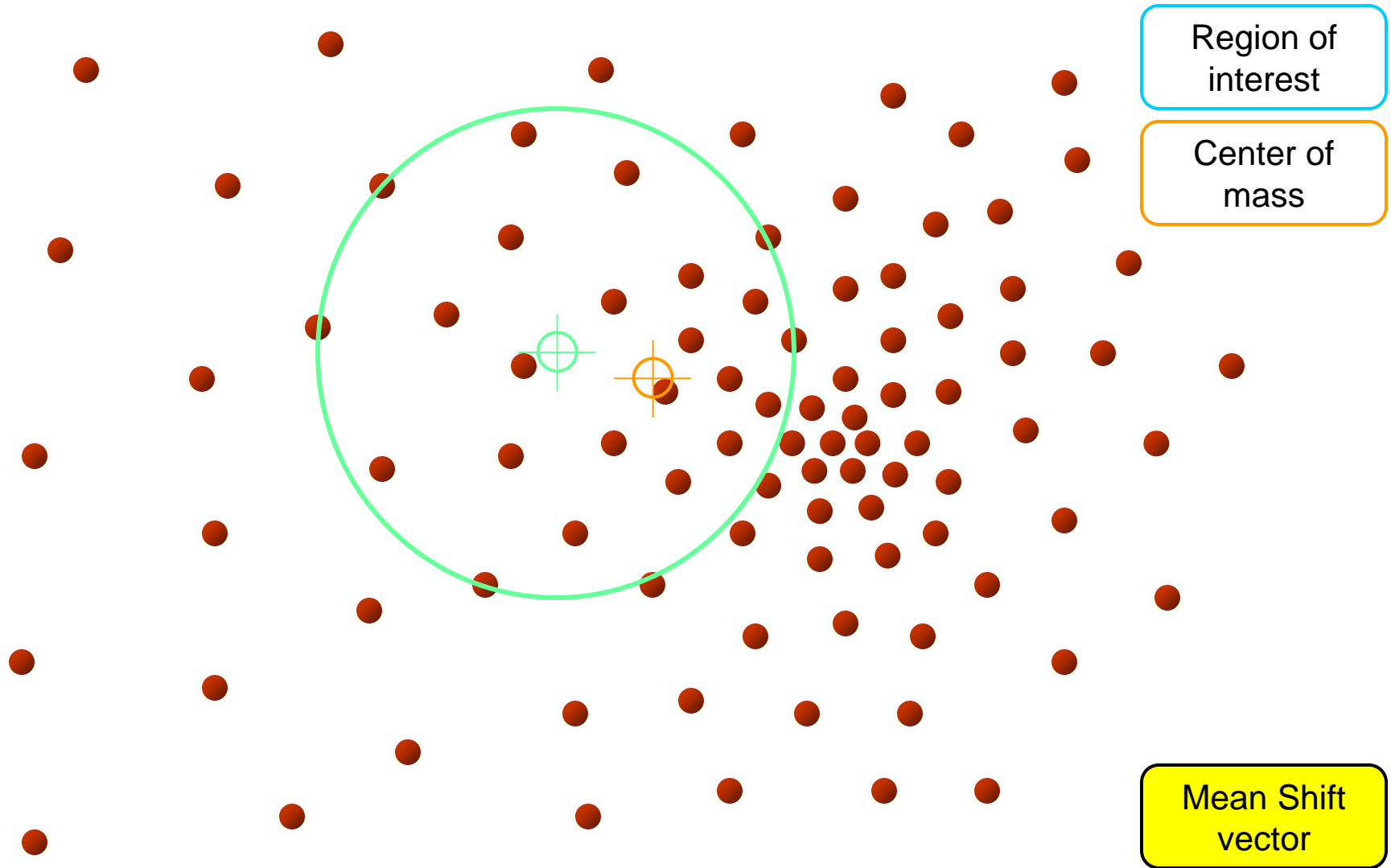
自底向上 (Bottom-up) 的处理方法

- 均值漂移法 (Mean Shift)
- 原理: 在图像分割中已经做过介绍
- 核心思想: 找到概率密度梯度为零的采样点, 并以此作为特征空间聚类的模式点.

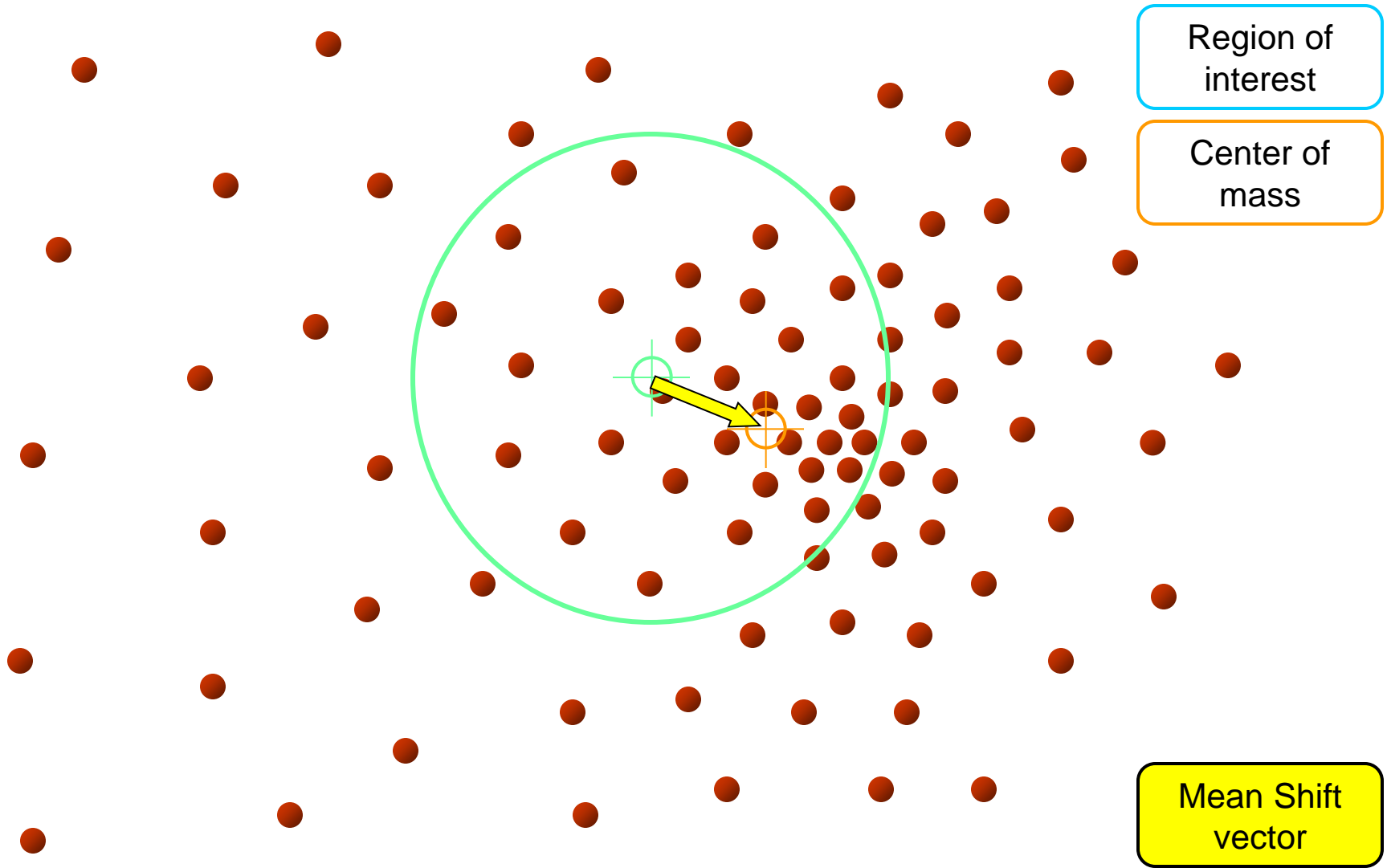
均值漂移Intuitive Description



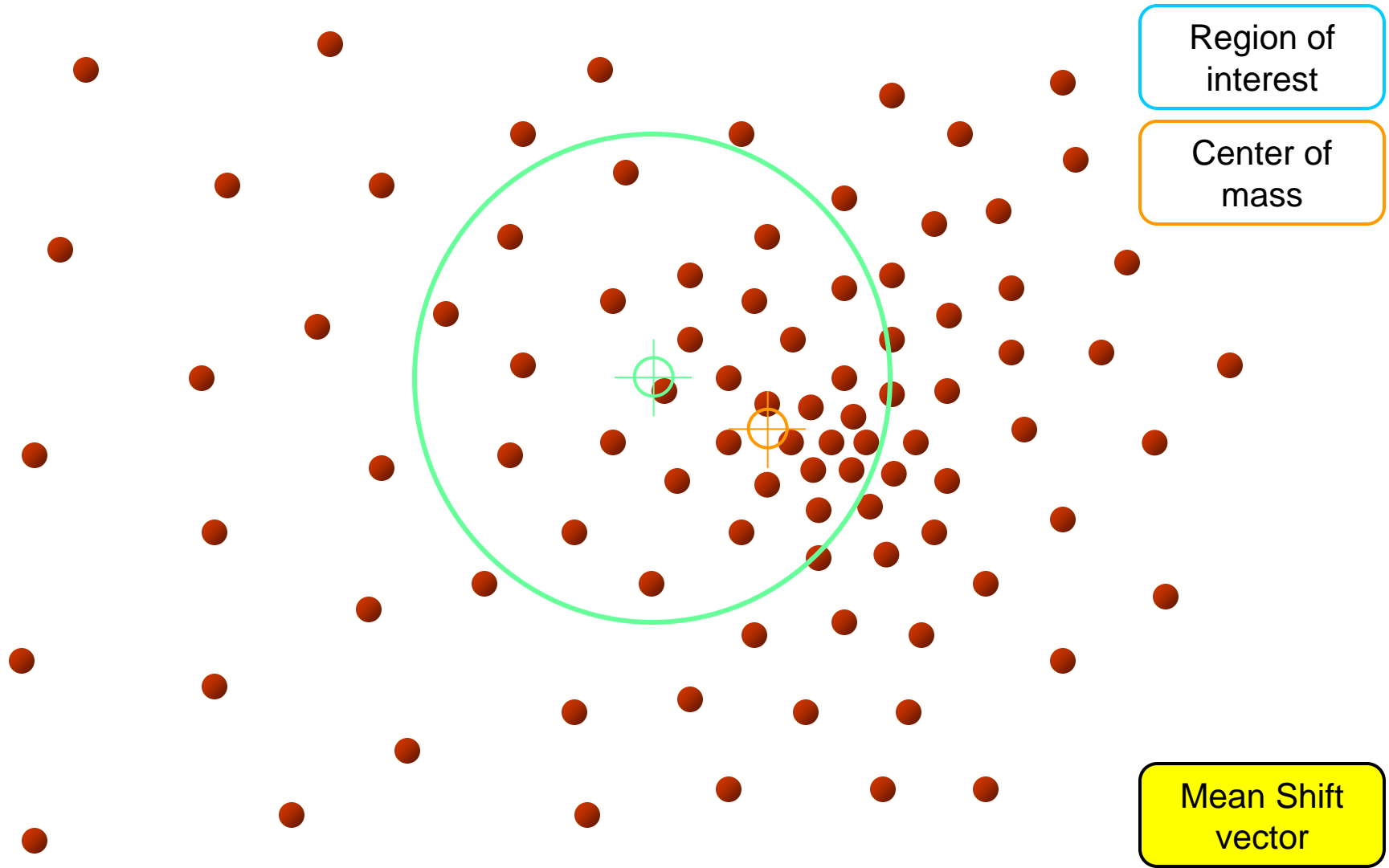
均值漂移Intuitive Description



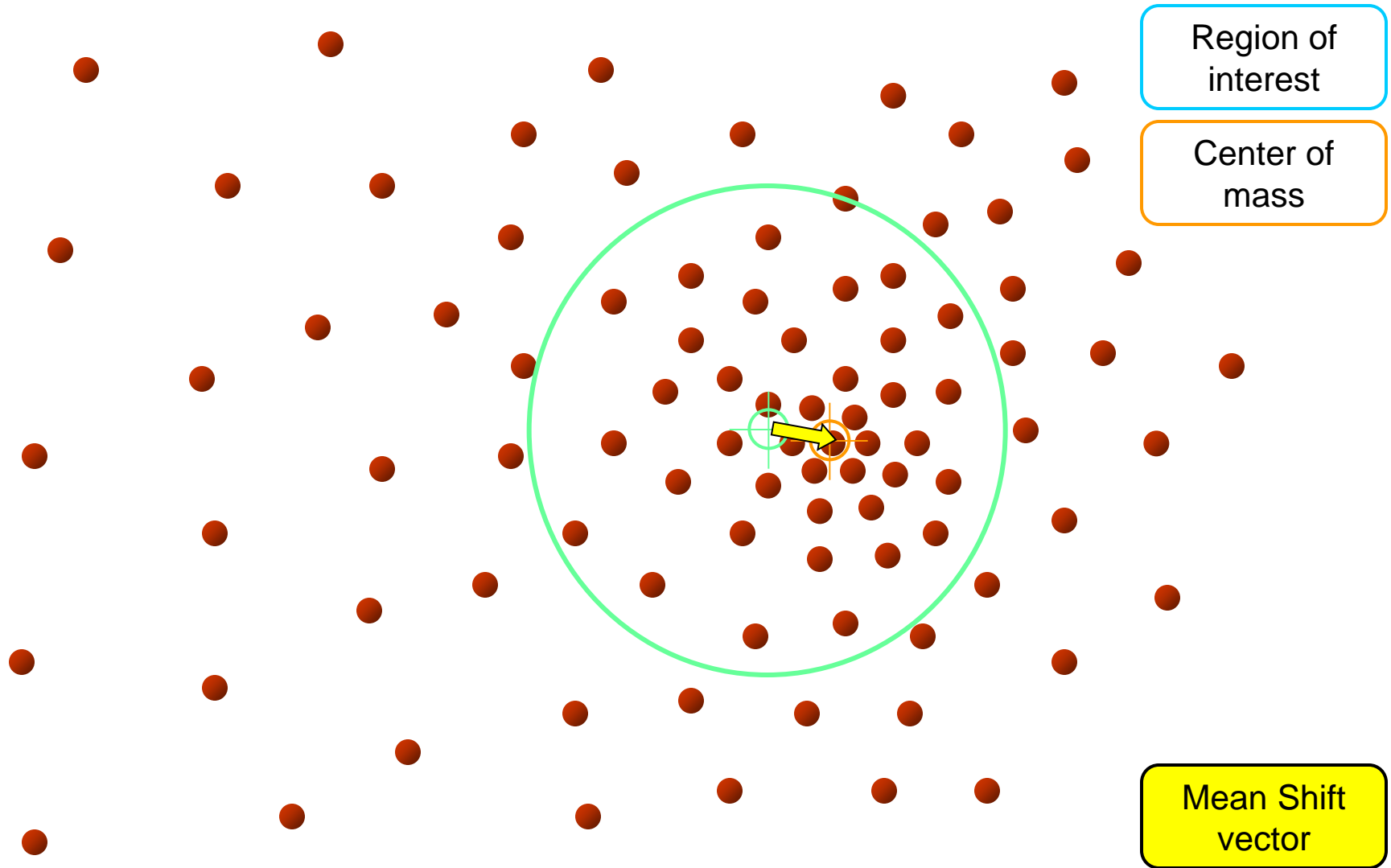
均值漂移 Intuitive Description



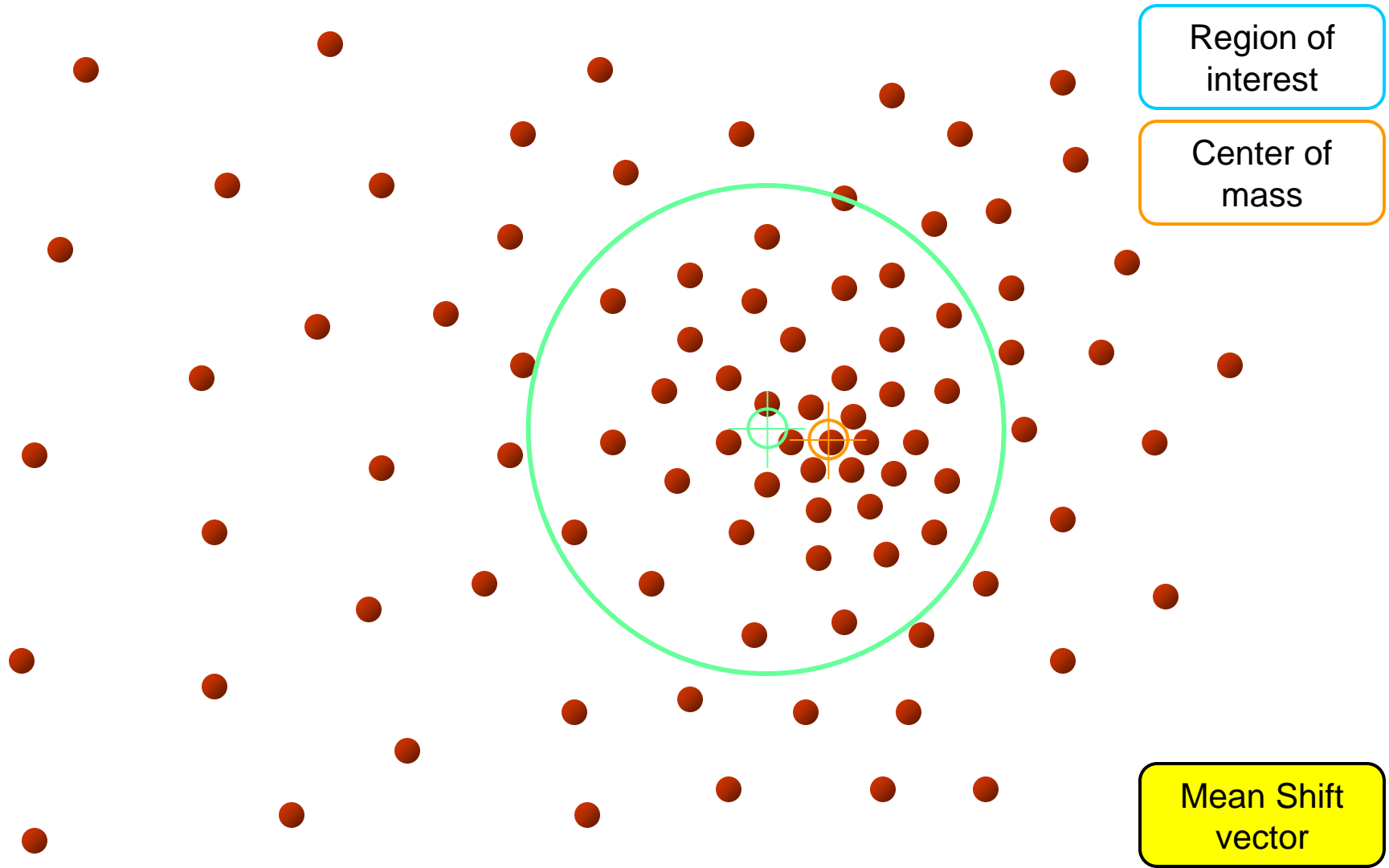
均值漂移Intuitive Description



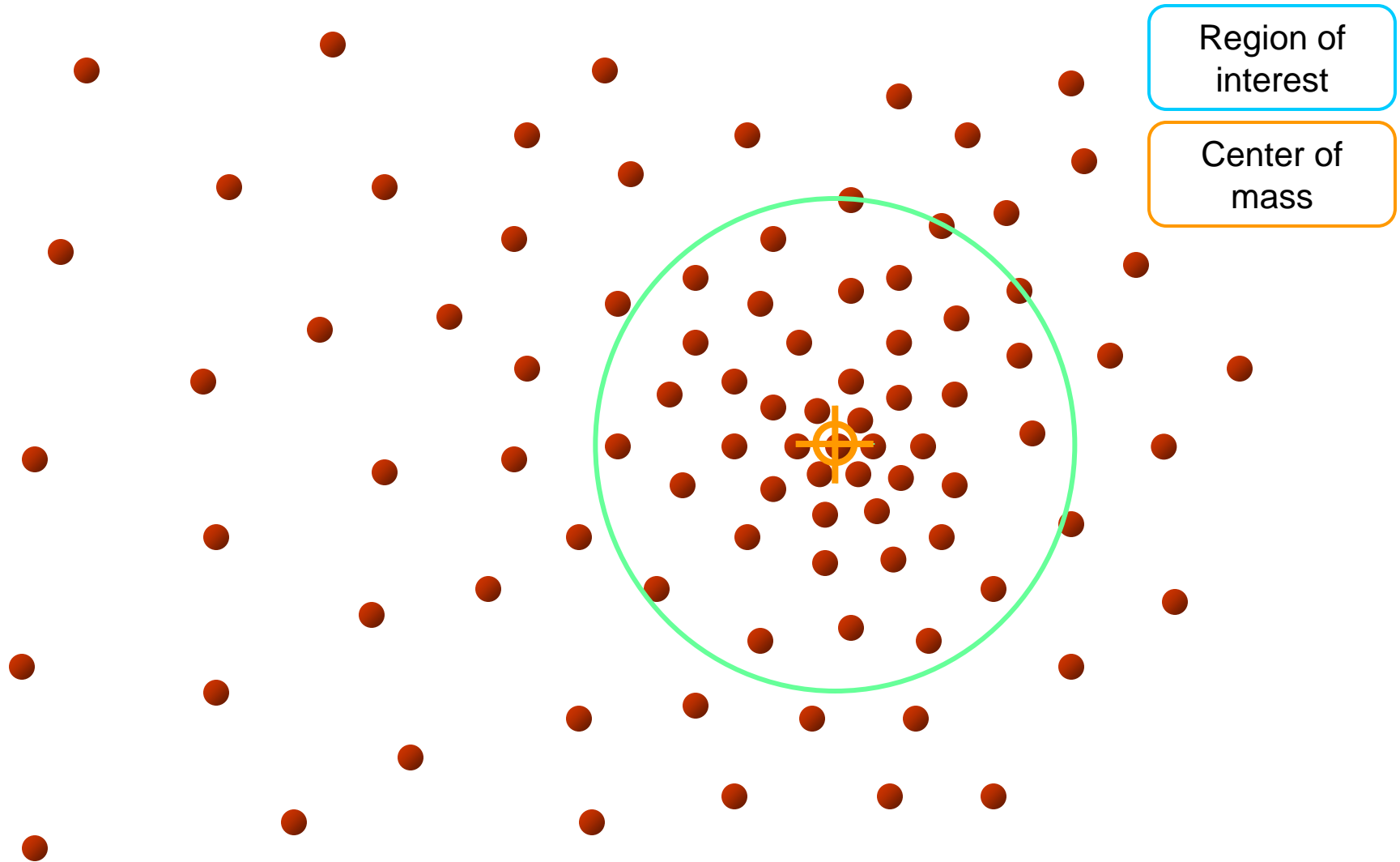
均值漂移 Intuitive Description



均值漂移 Intuitive Description



均值漂移Intuitive Description



Mean Shift跟踪算法

- 利用目标的灰度（或彩色）分布来描述这个物体, 假设目标中心位于 x_0 , 则该物体可以表示为

$$\hat{q}_u = C \sum_{i=1}^n k \left(\left\| \frac{x_i^s - x_0}{h} \right\|^2 \right) \delta[b(x_i^s) - u]$$

- 候选的目标可以描述为

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

- 目标跟踪过程转化为寻找令 $\hat{p}_u(y)$ 与 \hat{q}_u 相似度最高的解的过程。

Mean Shift跟踪算法

- 相似度量: $\hat{\rho}_u(y) \equiv \rho[p(y), q] = \sum_{u=1}^m \sqrt{p_u(y) \hat{q}_u}$

- 泰勒展开:

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p(y_0) q_u} + \frac{1}{2} \sum_{u=1}^m p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}}$$

→
$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{p(y_0) q_u} + \frac{C_h}{2} \sum_{i=1}^n w_i k \left(\left\| \frac{y - x_i}{h} \right\|^2 \right)$$

其中 $w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}}$

自底向上 (Bottom-up) 的处理方法

□均值漂移法 (Mean Shift)

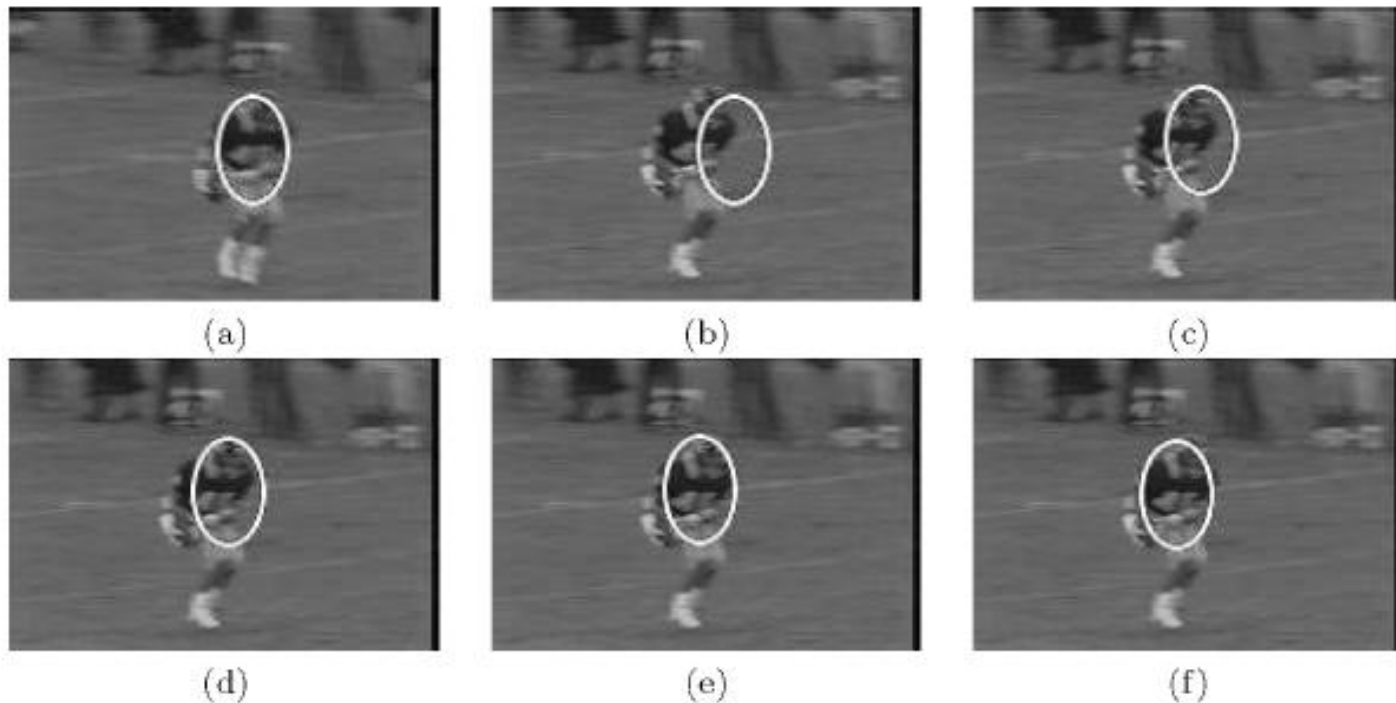


Fig. 12. Mean-shift tracking iterations. (a) estimated object location at time $t-1$, (b) frame at time t with initial location estimate using the previous object position, (c), (d), (e) location update using mean-shift iterations, (f) final object position at time t .

自底向上 (Bottom-up) 的处理方法

□ 均值漂移法 (Mean Shift) —— 通常只需要经过4—5次迭代

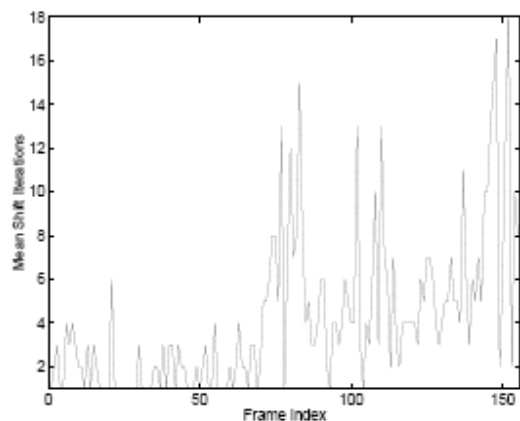


Figure 2: The number of mean shift iterations function of the frame index for the *Football* sequence. The mean number of iterations is 4.19 per frame.

- ❖ Comanniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(5): 564~577

自顶向下 (Top-down) 的处理方法

- 通过建立状态空间模型，把跟踪问题表示为动态系统的状态估计问题。
- 思路是将目标跟踪问题转换为在贝叶斯 (Bayesian) 理论框架下已知目标状态的先验概率，在获得的测量值 (Measurement) 后不断求解目标状态的后验概率的过程。

状态空间模型 (State Space Model)

□ 动态系统由状态转移方程和观测方程组成。

□ 状态转移方程：

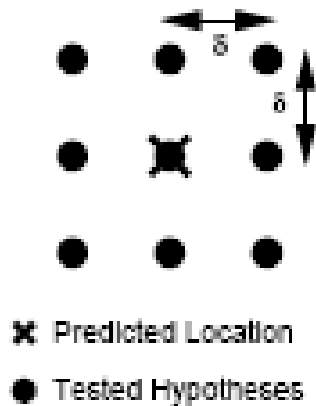
$$x_k = f_k(x_{k-1}, v_{k-1})$$

f : 在很多跟踪问题中可能是非线性的

x_k, x_{k-1} : 当前时刻和前一时刻的状态

v_{k-1} : 系统噪声，通常是非高斯的

状态空间模型 (State Space Model)



$$x_k = x_{k-1} + v_{k-1}$$

Figure 2. The nine lattice points which determine the local hypotheses to be tested for each object at each frame.

状态空间模型 (State Space Model)

□ 观测方程

$$y_k = h_k(x_k, n_k)$$

h : 在很多跟踪问题中可能是非线性的

y_k : 测量值

x_k : 状态

n_k : 测量噪声, 通常是非高斯的

贝叶斯递归滤波器 (Recursive Filters)

- 状态转移方程: $x_k = f_k(x_{k-1}, v_{k-1}) \leftrightarrow p(x_k | x_{k-1})$
- 测量方程: $y_k = h_k(x_k, n_k) \leftrightarrow p(y_k | x_k)$
- 已知信息: $y_{1:K} = (y_1, \dots, y_K), \quad p(x_0)$
- 要求的是: $p(x_k | y_{1:k})$

贝叶斯递归滤波器 (Recursive Filters)

□ 步骤:

■ 预测—测量—更新 (Predict—Measurement—Update)

■ 预测: $p(x_{k-1}|y_{1:k-1}) \rightarrow p(x_k|y_{1:k-1})$

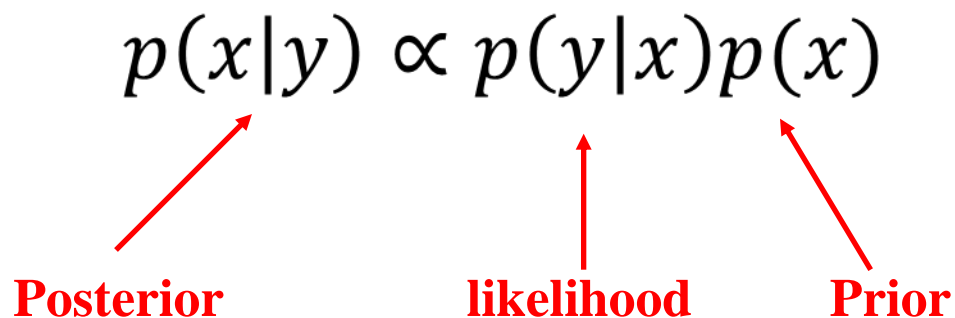
■ 更新: $p(x_k|y_{1:k}), y_k \rightarrow p(x_k|y_{1:k})$

测量



预测—测量—更新

□ 复习一下贝叶斯准则 (Bayes Rule) :

$$p(x|y) \propto p(y|x)p(x)$$


Posterior **likelihood** **Prior**

预测—测量—更新

□ 预测: (using Chapman-Kolmogoroff equation)

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

□ 更新: (using Bayes Rule)

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

$$p(x_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_k|y_{1:k-1})dx_k$$

- 上述推导给出了理论上解的形式，实际中往往无法得到解析表达
- 在特定假设下可以获得最优解——基于线性、高斯假设
 - 卡尔曼滤波器
- 对任意概率密度函数——概率密度函数多峰情况
 - 粒子滤波器

R. E. Kalman



- Born 1930 in Hungary
- Studied at MIT / Columbia
- Developed filter in 1960/61

卡尔曼滤波器 (Kalman Filter)

基本假设：

□ 后验概率分布 $p(x_{k-1}|y_{1:k-1})$ 为高斯分布

□ 动态系统是线性的

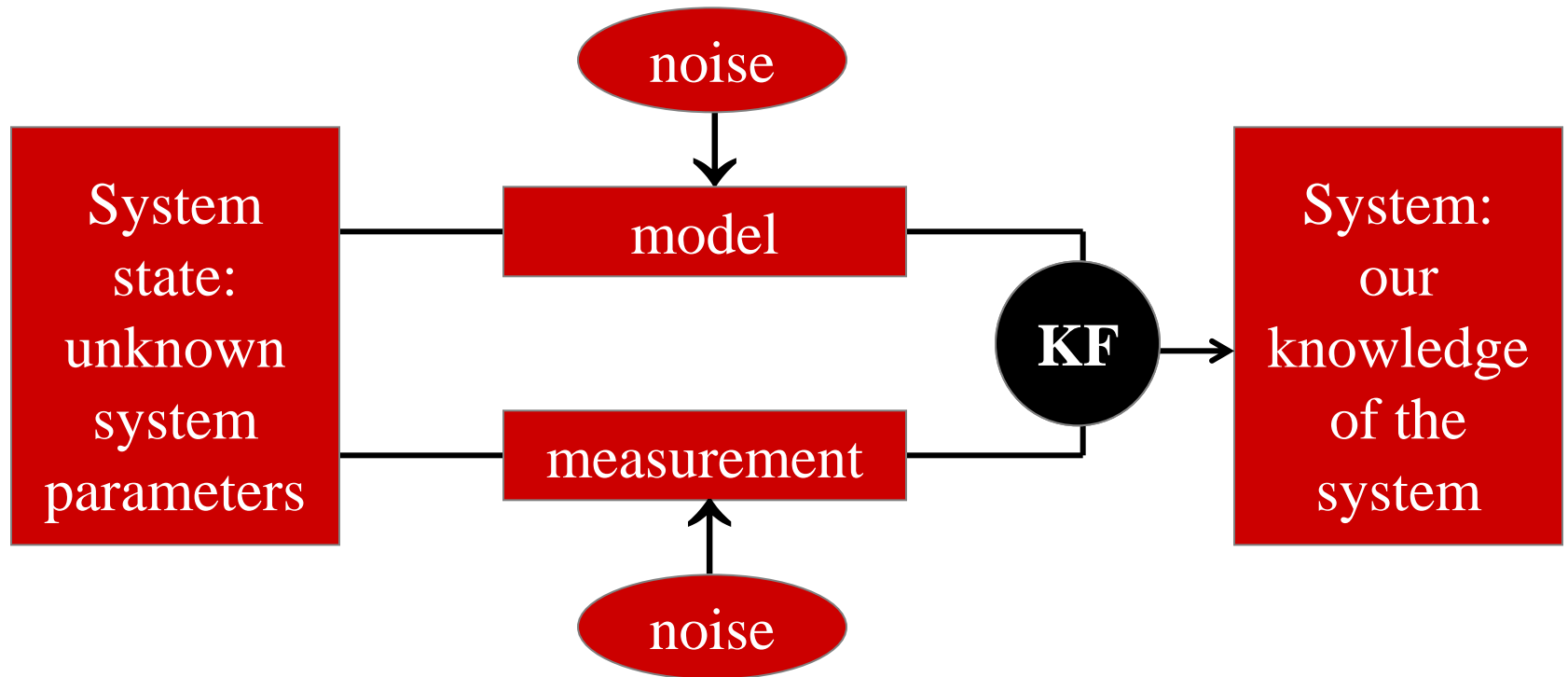
$$x_k = Ax_{k-1} + Bu_{k-1} + q_{k-1}$$

$$y_k = Hx_k + r_k$$

□ 系统噪声和测量噪声都是高斯分布的，协方差矩阵分别为 Q_{k-1} 和 R_k 。

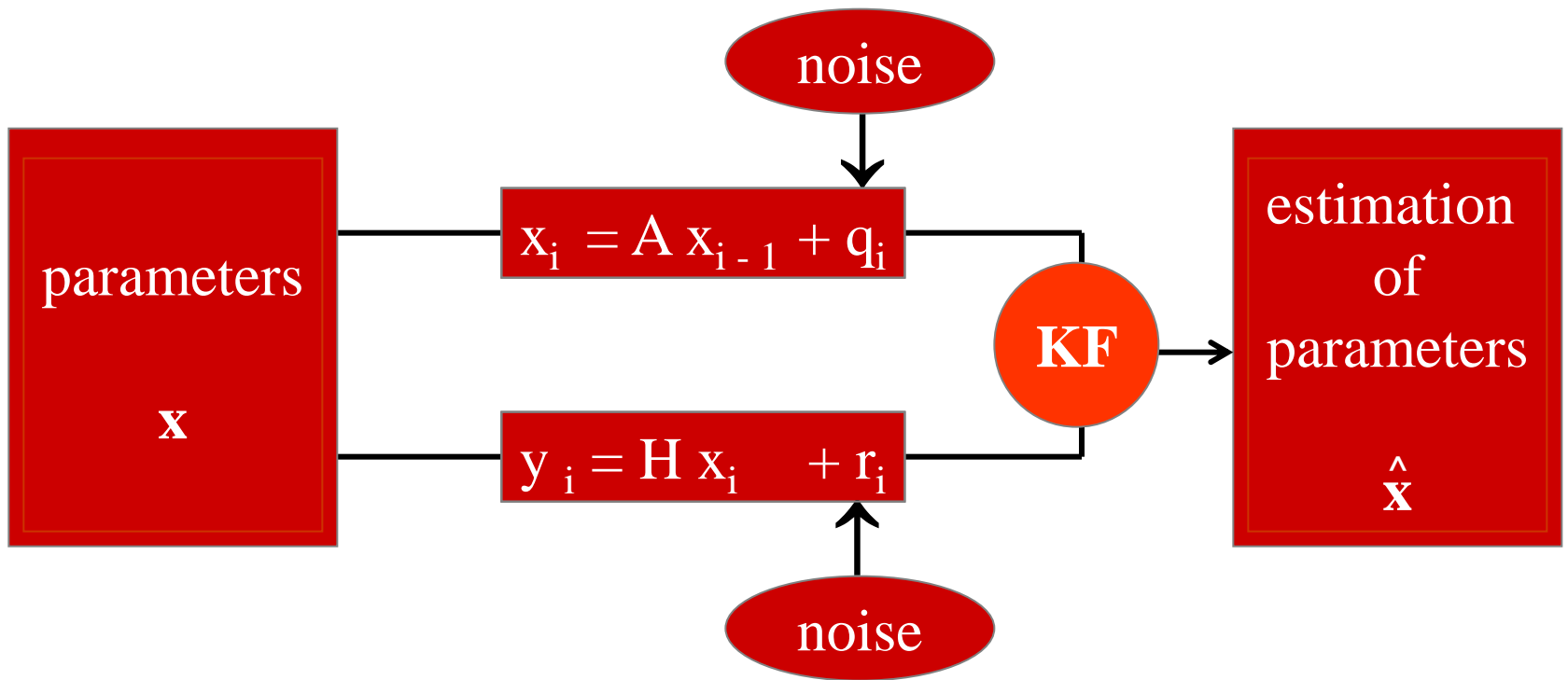
Kalman filter – KF

When and where?



KF description

using vectors and matrices



Noise

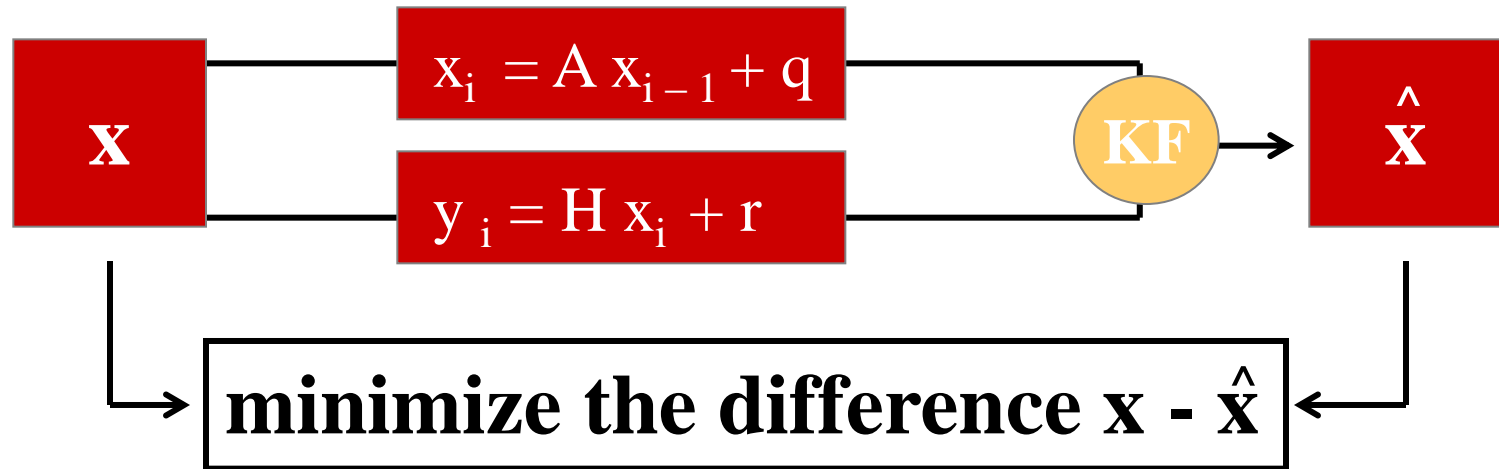
Noise: e Gaussian $\Rightarrow E(e^2) = \sigma^2$

Noise covariance matrix

$$P = E(ee^T) = \begin{pmatrix} E(e_1e_1) & E(e_1e_2) & \cdots \\ E(e_2e_1) & E(e_2e_2) & \\ \vdots & & \ddots \end{pmatrix}$$

- System noise: $\mathbf{x}_i = \mathbf{A} \mathbf{x}_{i-1} + \mathbf{q}_i \Rightarrow \mathbf{Q} = E(\mathbf{q}\mathbf{q}^T)$
- Measurement noise: $\mathbf{y}_i = \mathbf{H} \mathbf{x}_i + \mathbf{r}_i$
 $\Rightarrow \mathbf{R} = E(\mathbf{r}\mathbf{r}^T)$

KF algorithm



- Prediction: $\hat{\mathbf{x}}_i^- = \mathbf{A} \hat{\mathbf{x}}_{i-1} + \mathbf{q}$
- Correction: $\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i^- + \mathbf{K} (\mathbf{y}_i - \mathbf{H} \hat{\mathbf{x}}_i^-)$

Kalman gain matrix

Kalman gain matrix

- Minimize the expected error

$$\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}} ; \mathbf{P} = E(\mathbf{e}\mathbf{e}^T) = \begin{pmatrix} E(\mathbf{e}_1\mathbf{e}_1) & E(\mathbf{e}_1\mathbf{e}_2) & \cdots \\ E(\mathbf{e}_2\mathbf{e}_1) & E(\mathbf{e}_2\mathbf{e}_2) & \\ \vdots & & \ddots \end{pmatrix} \Rightarrow \frac{\partial \mathbf{P}_{ij}}{\partial \mathbf{K}} = 0$$

- It is easy to show

$$\mathbf{K} = \mathbf{P}^- \mathbf{H}^T (\mathbf{H} \mathbf{P}^- \mathbf{H}^T + \mathbf{R})^{-1},$$

$$\text{where } \mathbf{P}_i^- = \mathbf{A} \mathbf{P}_{i-1} \mathbf{A}^T + \mathbf{Q}$$

- Limits:

- system noise \ll measurement noise $\Rightarrow \hat{\mathbf{x}}_i = \hat{\mathbf{x}}_i^-$
- system noise \gg measurement noise $\Rightarrow \hat{\mathbf{x}}_i = \mathbf{H}^{-1} \mathbf{y}_i$

卡尔曼滤波器—时间更新和状态更新

- 时间更新

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$

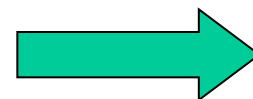
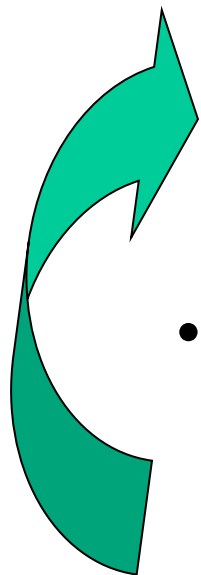
$$P_k^- = AP_{k-1}A^T + Q$$

- 状态更新

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - H\hat{x}_k^-)$$

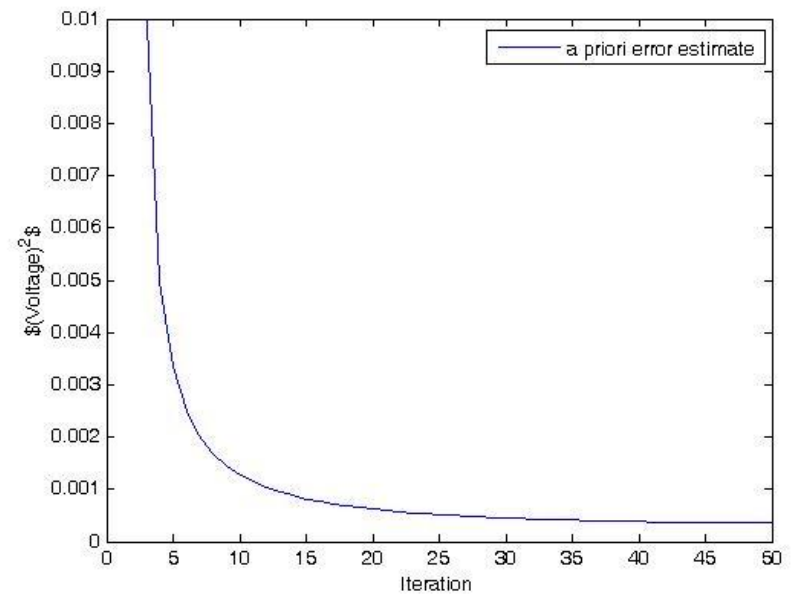
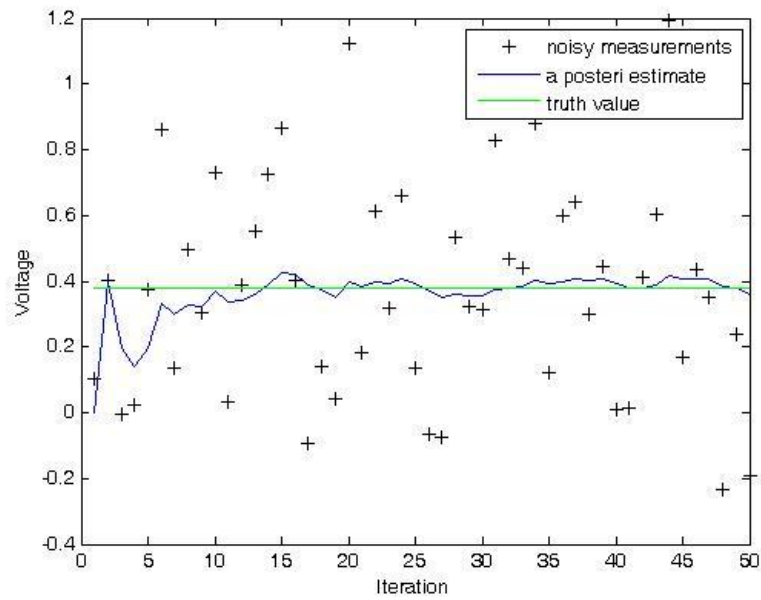
$$P_k = (I - K_k H)P_k^-$$



实例

- 测量电压：假设我们可以测量这个常数的幅值，但观测幅值中掺入了幅值均方根为0.1伏的白噪声。
- 方程描述：
- $x_k = Ax_{k-1} + Bu_{k-1} + q_{k-1} = x_{k-1} + q_{k-1}$
- $y_k = Hx_k + r_k = x_k + r_k$
- 过程的状态不随时间变化， $A = 1$ ；没有控制输入， $u = 0$ ；包含噪声的观测值是状态变量的直接体现， $H = 1$ 。

实例



卡尔曼滤波器的扩展

- Extended Kalman Filter (EKF)

- Unscented Kalman Filter (UKF)

- 同样基于高斯分布的假设；
- 状态转移方程和测量方程为非线性函数；
- 沿用Kalman Filter的框架；
- 将非线性函数局部线性化。

❖ 为处理更为一般的概率密度函数（比如，多峰情况），我们需要适应性更强的方法——粒子滤波器。

粒子滤波器 (Particle Filter)

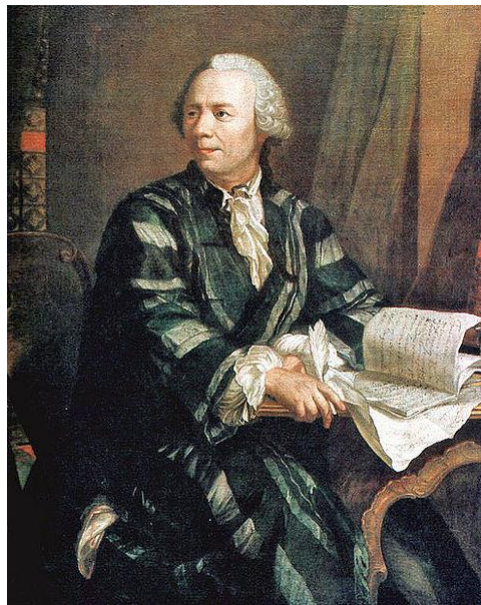
□ 粒子滤波器的一系列别名：

- Condensation Algorithms
- Sequential sampling-importance re-sampling (SIR)
- Bootstrap Filtering
- Interacting particle approximations
- Survival of the fittest
-

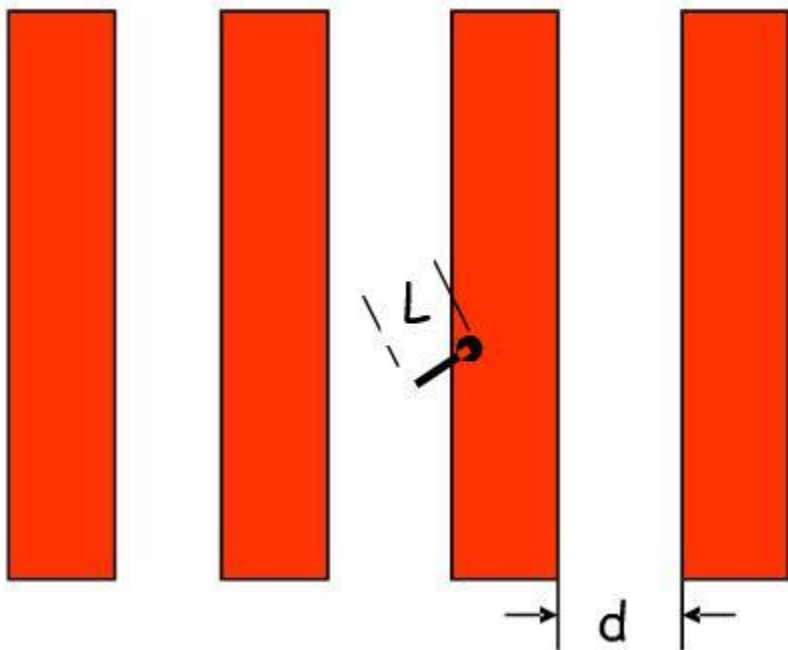
两个问题

- π 是多少? 怎么求?
- 3.14159265358979323846264338327950288419716939937510582097944
592307816406286208998628034825342117067982148086513282306764
709384460955058223172535940812848111745028410270193852110955
596446229489549303819644288109756659334461284756482337867831
652712019091456485669234603486104543266482133936072602491412
737245870066063155881748815209209628292540917153643678925903
600113305305488204665213841469519415116094330572703657595919
530921861173819326117931051185480744623799627495673518857527
248912279381830119491298336733624406566430860213949463952247
371907021798609437027705392171762931767523846748184676694051
320005681271452635608277857713427577896091736371787214684409
012249534301465495853710507922796892589235420199561121290219
608640344181598136297747713099605187072113499837297804995105
973173281609631859502445945534690830264252230825334468503526
193118817101000313783875288658753320838142061717766914730359
825349042875546873115956286388235378201667315642315632318742
3187323128423128323658323

两个问题



投针实验

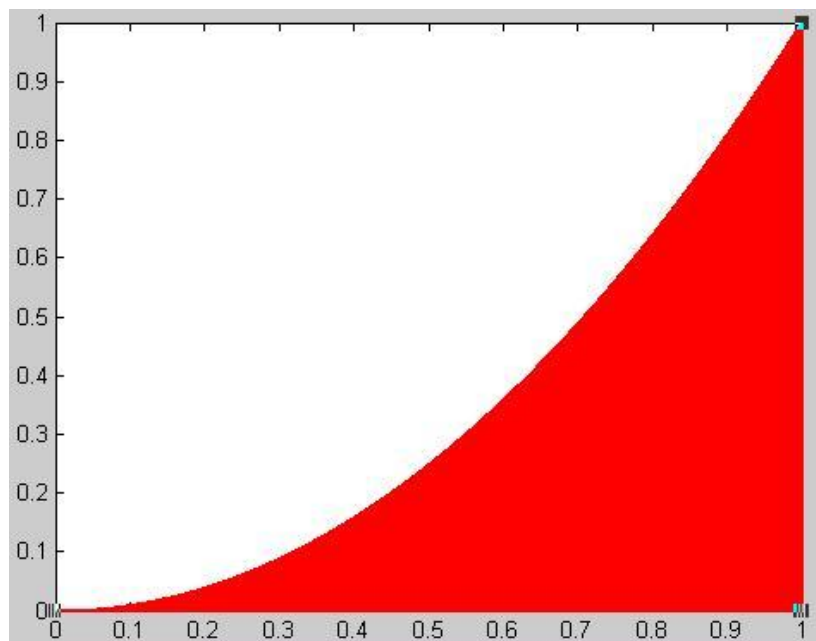


试验者	时间(年)	针长	投针次数	相交次数	π 的估计值
Wolf	1850	0.80	5000	2532	3.15956
Smith	1855	0.60	3204	1218	3.15665
Fox	1884	0.75	1030	489	3.15951
Lazzarini	1925	0.83	3408	1808	3.14159292

Monte Carlo方法

- 研究问题：某随机事件A出现的概率。
- 基本思想：通过某种“实验”的方法，得出A事件出现的频率，以此估计出A事件出现的概率。

Monte Carlo方法



- 随机生成1000000个点，结果为0.3328。

历史

- First attempts – simulations of growing polymers
 - M. N. Rosenbluth and A.W. Rosenbluth, “Monte Carlo calculation of the average extension of molecular chains,” *Journal of Chemical Physics*, vol. 23, no. 2, pp. 356–359, 1956.
- First application in signal processing - 1993
 - N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proceedings-F*, vol. 140, no. 2, pp. 107–113, 1993.
- Books
 - A. Doucet, N. de Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*, Springer, 2001.
 - B. Ristic, S. Arulampalam, N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House Publishers, 2004.
- Tutorials
 - M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

粒子滤波器 (Particle Filter)

- 基于贝叶斯准则的序贯蒙特卡罗算法 (Sequential Monte Carlo)
- 通过寻找一组随机样本对任意概率密度函数进行近似，也即：用一组离散样本点来近似连续的概率密度函数。

Monte Carlo方法

- 例子：估计如下积分值（ p 为高斯分布函数）

$$v = \int_{-\infty}^{+\infty} x^2 p(x) dx$$

- Monte Carlo 方法：

1. 模拟 M 个随机变量（符合高斯分布）

$$x^{(m)} \sim N(0, \sigma^2)$$

2. 计算均值

$$v = \frac{1}{M} \sum_{m=1}^M (x^{(m)})^2$$

重要性采样 (Importance sampling)

- 计算积分

$$E(f(x)) = \int_{-\infty}^{+\infty} f(x)p(x)dx = \int_{-\infty}^{+\infty} f(x)\frac{p(x)}{\pi(x)}\pi(x)dx$$

- Monte Carlo方法
- 1. 模拟M个随机变量 $(\pi(x))$
 $x^{(m)} \sim \pi(x)$

- 2. 计算均值

$$E(f(x)) \approx \frac{1}{M} \sum_1^M f(x^{(m)}) \underbrace{\frac{p(x^{(m)})}{\pi(x^{(m)})}}_{\omega^{(m)}}$$

重要性采样

- 计算积分

$$E(g_t(x_{0:t})) = \int g_t(x_{0:t}) p(x_{0:t} | y_{1:t}) dx_{0:t}$$

$$E(g_t(x_{0:t})) = \int g_t(x_{0:t}) \frac{p(x_{0:t} | y_{1:t})}{q(x_{0:t} | y_{1:t})} q(x_{0:t} | y_{1:t}) dx_{0:t}$$

- Monte Carlo 方法:

1. 模拟N个随机变量 (q(x)) —

2. 计算均值

$$E(g_t(x_{0:t})) = \frac{1/N \sum_{i=1}^N g_t(x_{0:t}^{(i)}) \omega_t(x_{0:t}^{(i)})}{1/N \sum_{i=1}^N \omega_t(x_{0:t}^{(i)})}$$

$$= \sum_{i=1}^N g_t(x_{0:t}^{(i)}) \tilde{\omega}_t(x_{0:t}^{(i)})$$

$$\hat{p}(x_{0:t} | y_{1:t}) = \sum_{i=1}^N \tilde{\omega}_t^{(i)} \delta_{x_{0:t}^{(i)}}(dx_{0:t})$$

Sequential importance sampling

$$p(x_{0:t}) = p(x_0) \prod_{j=1}^t p(x_j | x_{j-1}), \quad p(y_{1:t} | x_{0:t}) = \prod_{j=1}^t p(y_j | x_j)$$

1. Particle generation $x_k^{(m)} = p(x_k | x_{k-1})$

2a. Weight computation $\omega_k^{*(m)} = \omega_{k-1}^{*(m)} p(y_k | x_k^{(m)})$

2b. Weight normalization $\omega_k^{(m)} = \frac{\omega_k^{*(m)}}{\sum_{m=1}^M \omega_k^{*(m)}}$

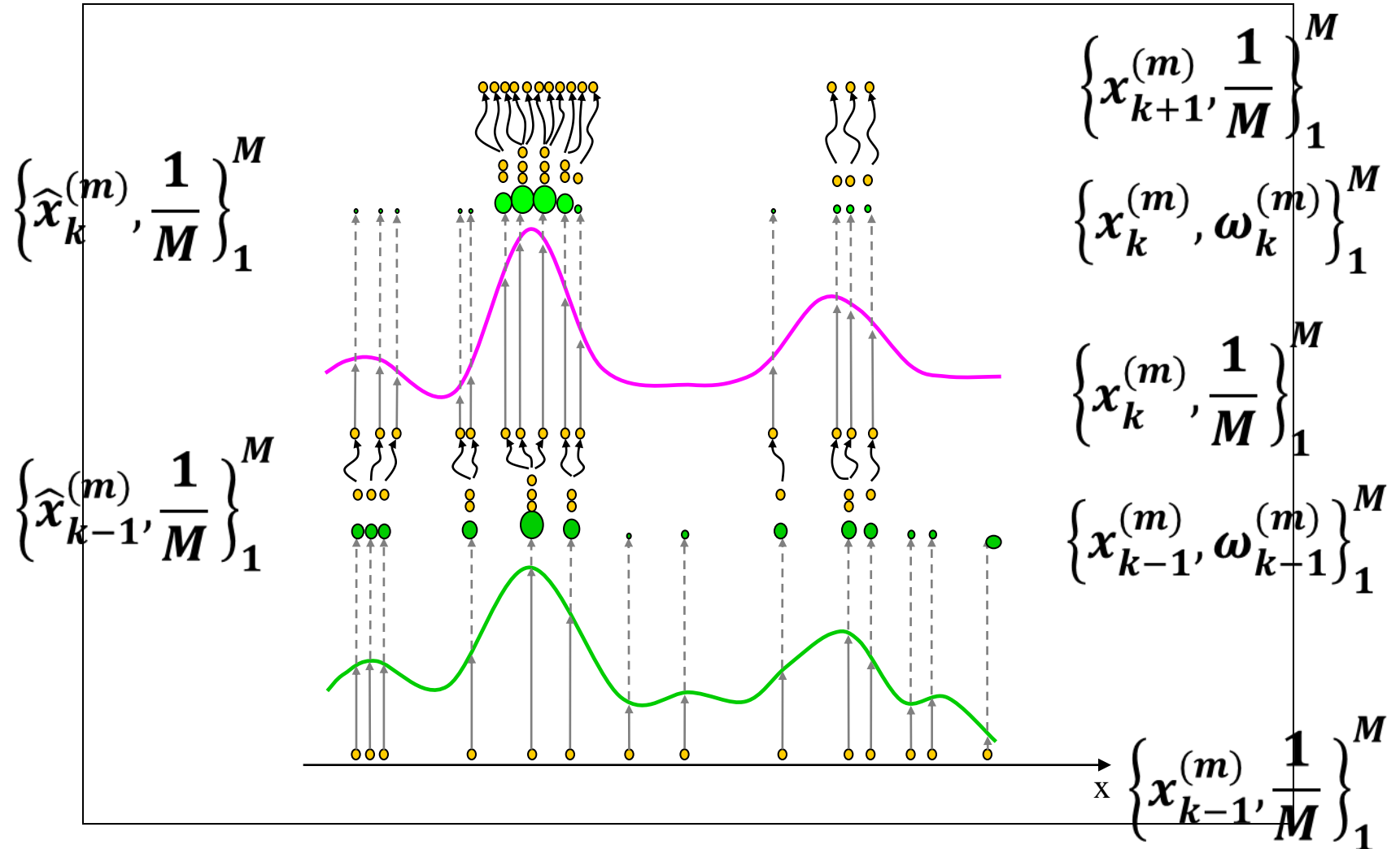
3. Estimate computation $E(g(x_k | y_{1:k})) = \sum_{m=1}^M g(x_k^{(m)}) \omega_k^{(m)}$

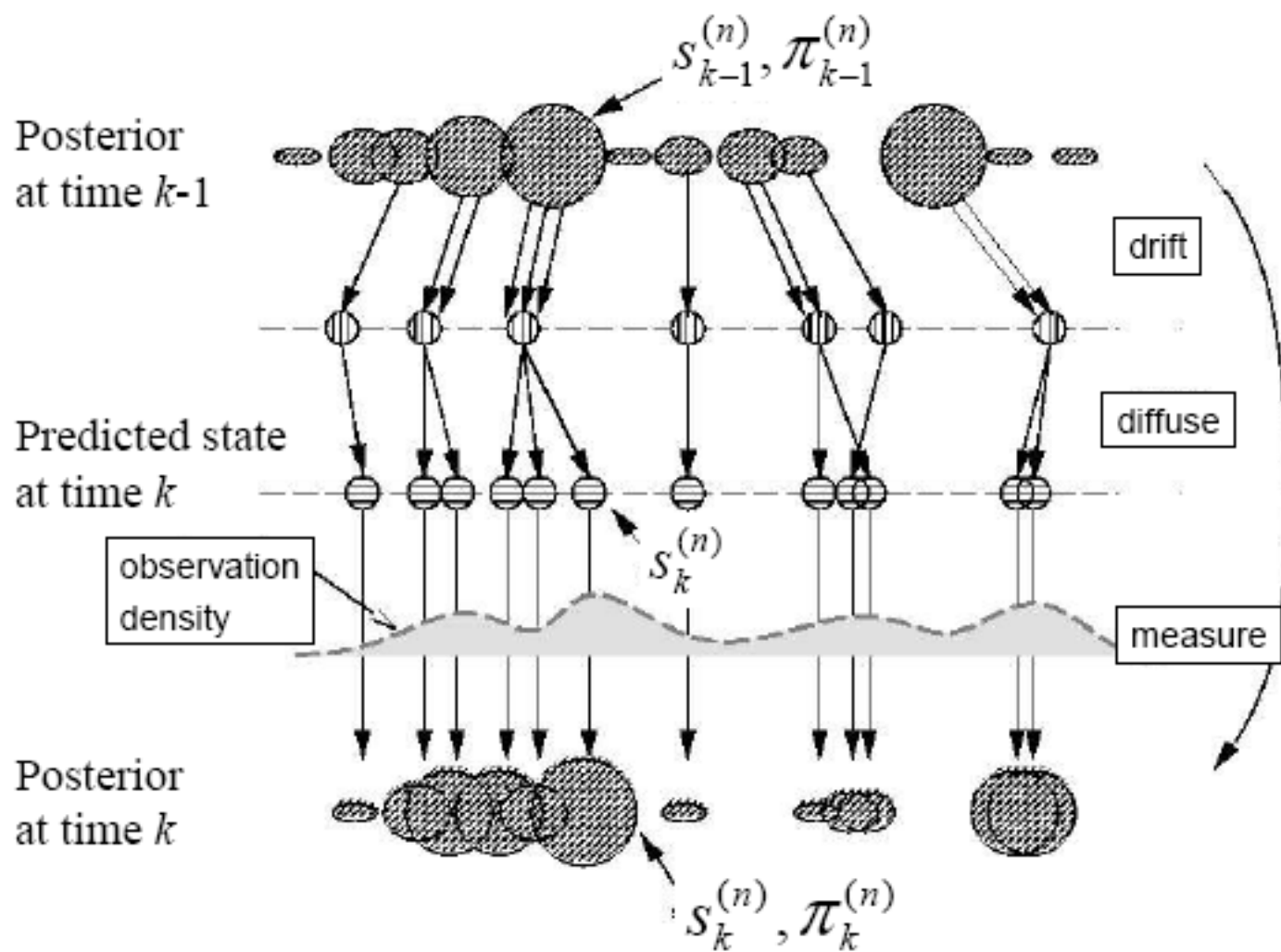
重采样Resampling

- 重采样

$$\left\{ \hat{x}_k^{(m)}, \frac{1}{M} \right\}_{m=1}^M \sim \left\{ x_k^{(m)}, \omega_k^{(m)} \right\}_{m=1}^M$$

Resampling





粒子滤波器 (Particle Filter)

优点

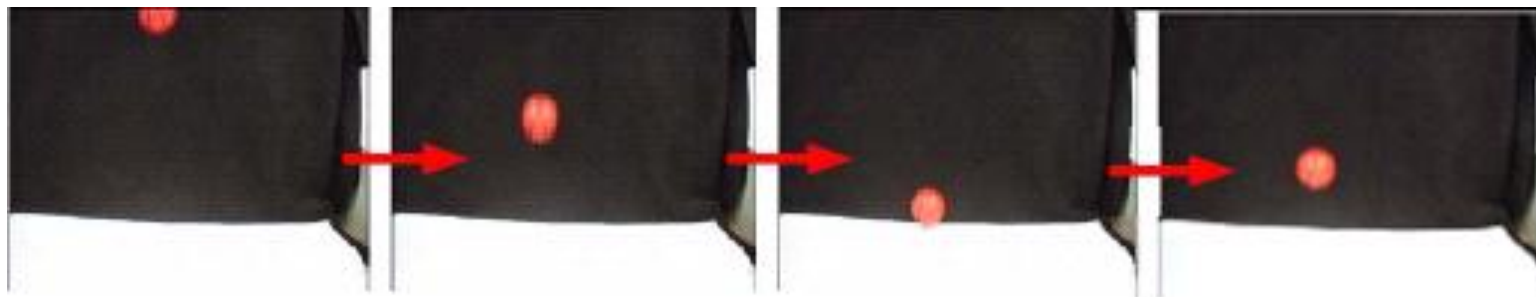
1. 通过随机采样近似复杂的、无法解析表达的概率密度函数
2. 适应性广

不足：

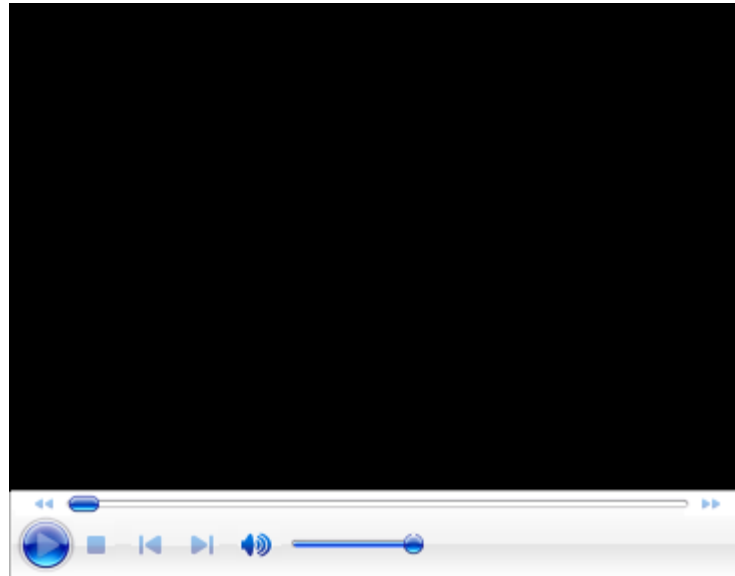
1. 高计算复杂度
2. 粒子数目
3. 重要性密度函数的选择

实验 —— Bouncing Ball

Kalman Filter vs. Particle Filter 性能比较



Kalman Filter

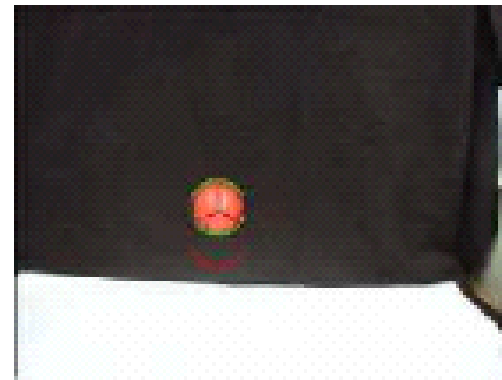


The estimated position from the Kalman filter (red) is compared against the actual ground truth position (green).

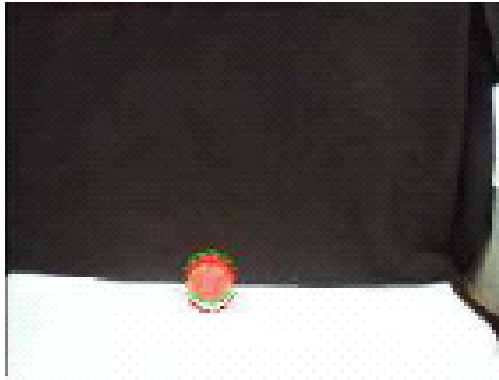
Kalman Filter

Kalman filter Failures:

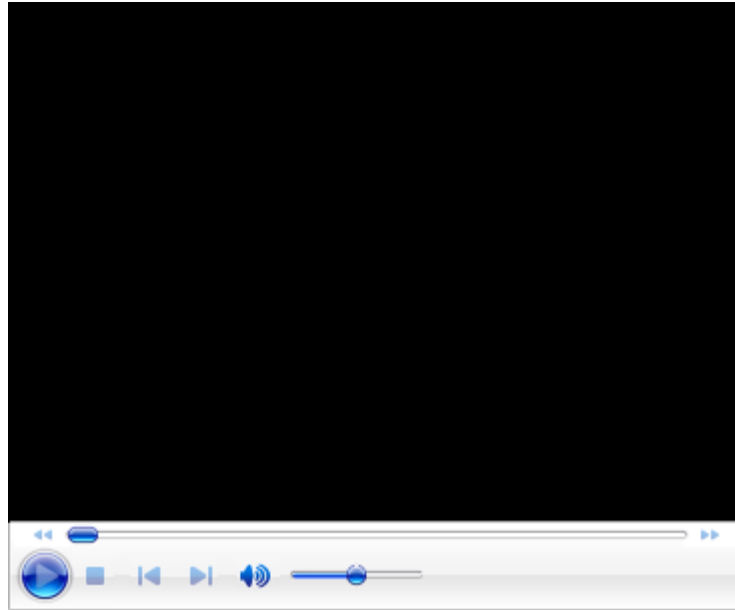
14: BOUNCE OVERSHOOT 16: SLOW CATCH UP



59: GRAVITY PULLS DOWN AT REST



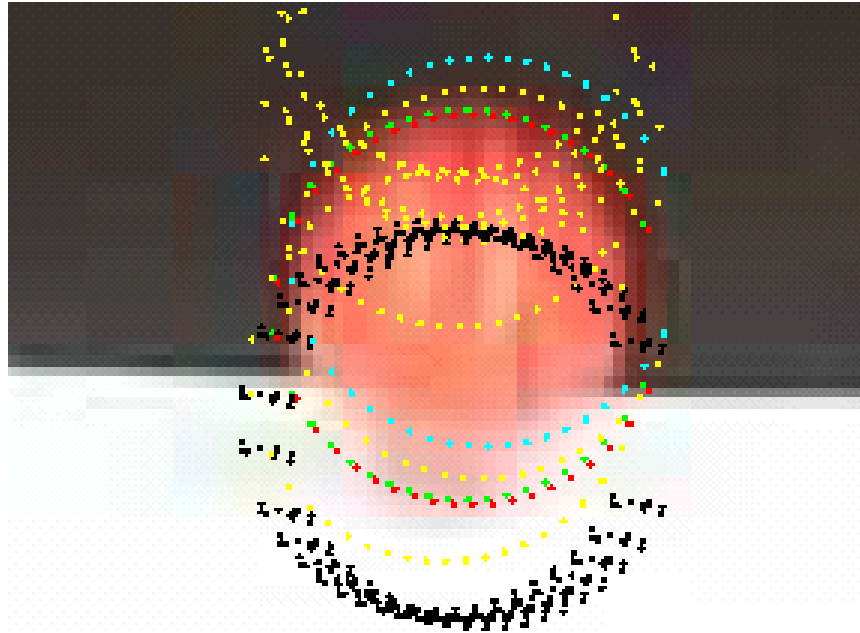
Particle Filter



The predicted position from the particle filter (red) is compared against the actual ground truth position (green).

Particle Filter

EXAMPLE OF SAMPLING EFFECTS



Red:final estimate Green:data
Yellow:BOUNCE Blue:STOP Black:FALL

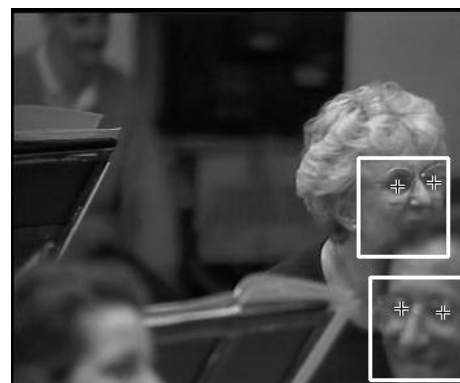
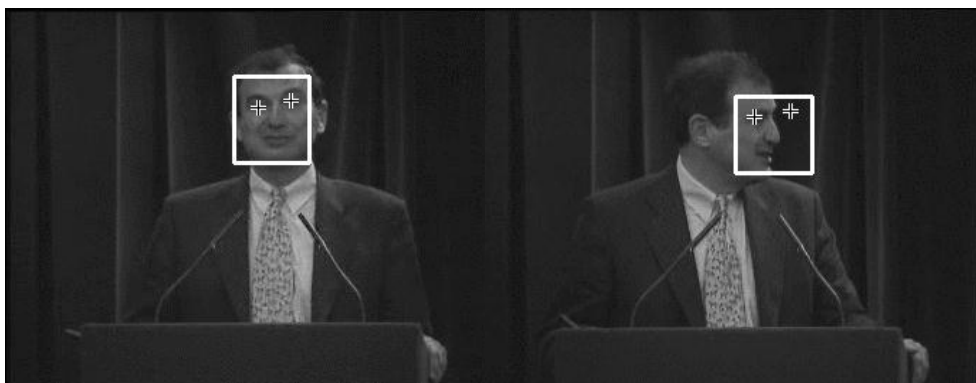
运动跟踪中的一些难点问题

- 运动跟踪所面临的主要难点可以归结为对跟踪算法在三个方面的要求，即对算法的鲁棒性、准确性和快速性的要求。

运动跟踪中的一些难点问题

□ 鲁棒性要求跟踪算法能够在各种环境条件下实现对运动目标持续稳定的跟踪。影响视觉跟踪算法鲁棒性的主要原因在于：

- 被跟踪运动目标的姿态变化；
- 运动目标所处环境的变化（比如，光照条件等）；
- 部分遮挡引起的运动目标不规则变形和全部遮挡引起的运动目标的暂时消失。



运动跟踪中的一些难点问题

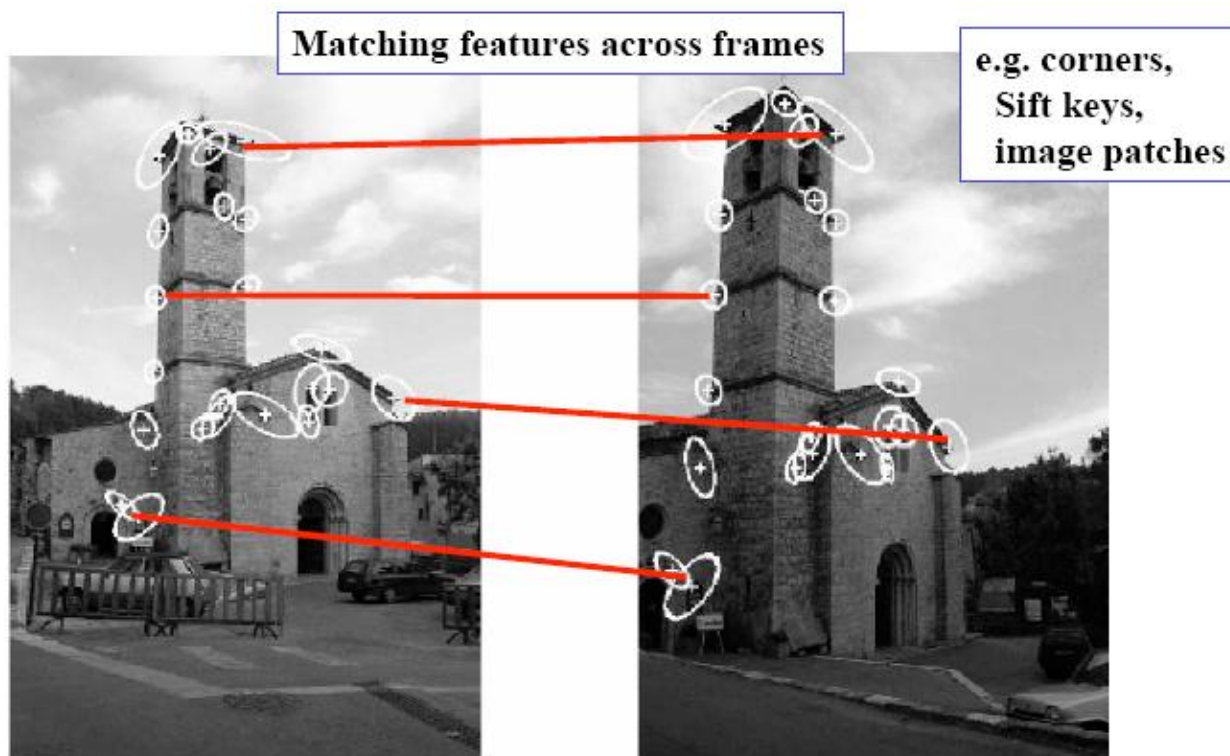
- 准确性的要求包括两个方面：
 - 对运动目标检测的准确性；
 - 对运动目标分割的准确性；
- 快速性的要求在于：在保证所要求的跟踪精度的前提下实现实时地跟踪。

- ❑ 运动目标的表示方法
- ❑ 具体介绍常用的跟踪方法
- ❑ 多目标跟踪的数据关联问题
- ❑ 多摄像机跟踪

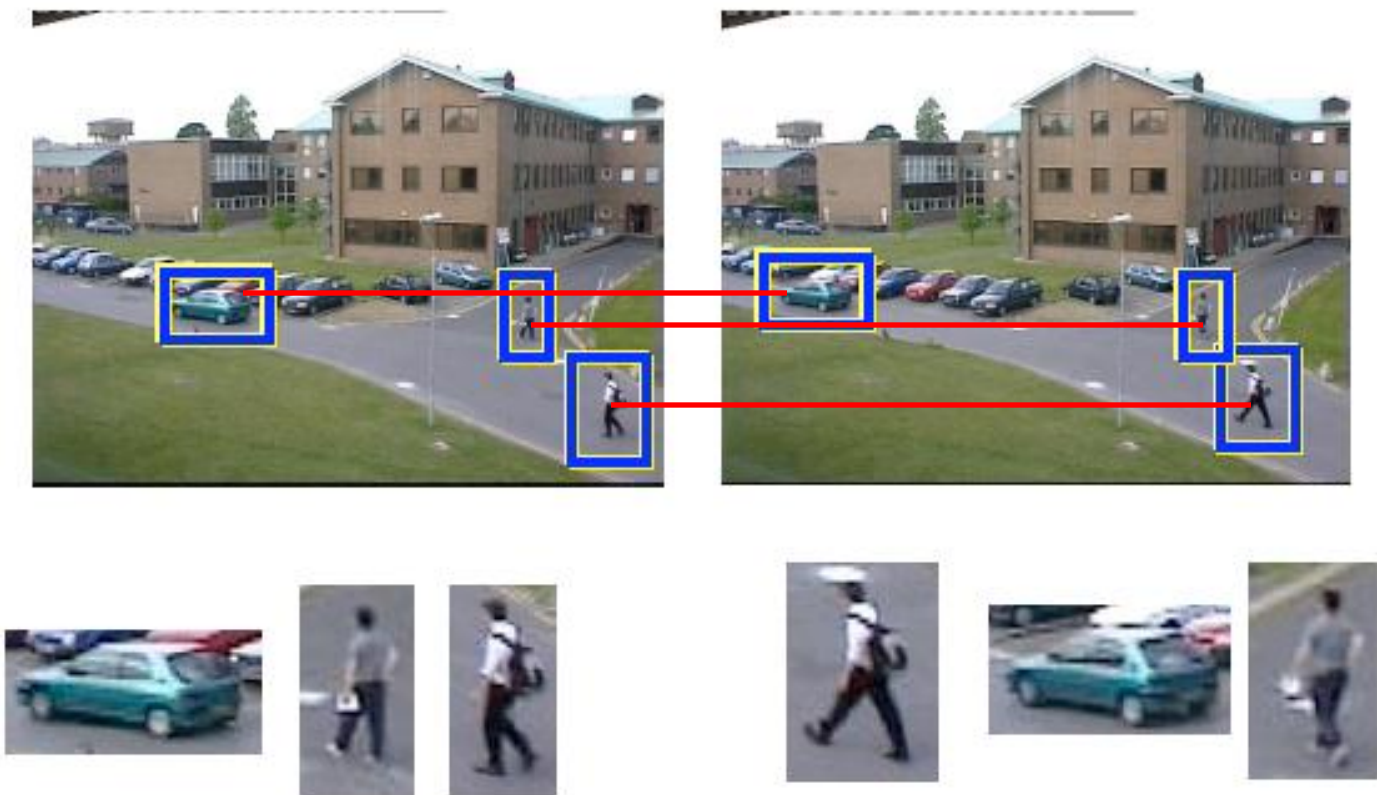
多目标跟踪中的数据关联

□ 数据关联 — Data Association

□ 类似于在两幅图像中找对应点 (Correspondence) 的问题



多目标跟踪中的数据关联



多目标跟踪中的数据关联

□ 贝叶斯递归滤波器

■ 预测—测量—更新

■ 预测: $p(x_{k-1}|y_{1:k-1}) \rightarrow p(x_k|y_{1:k-1})$

■ 更新: $p(x_k|y_{1:k}), y_k \rightarrow p(x_k|y_{1:k})$

测量



- ❑ 运动目标的表示方法
- ❑ 具体介绍常用的跟踪方法
- ❑ 多目标跟踪的数据关联问题
- ❑ 多摄像机跟踪

多摄像机跟踪

- 在跟踪过程中需要使用多个摄像机主要基于以下的两个原因：
 - 可以通过选择最好的视点来解决遮挡问题；应该说，解决遮挡问题最有实际意义的潜在方法应该是基于多摄像机的跟踪系统。
 - 可以扩大跟踪的有效区域，单个摄像机的视野范围毕竟是有限的。

多摄像机跟踪 一个例子

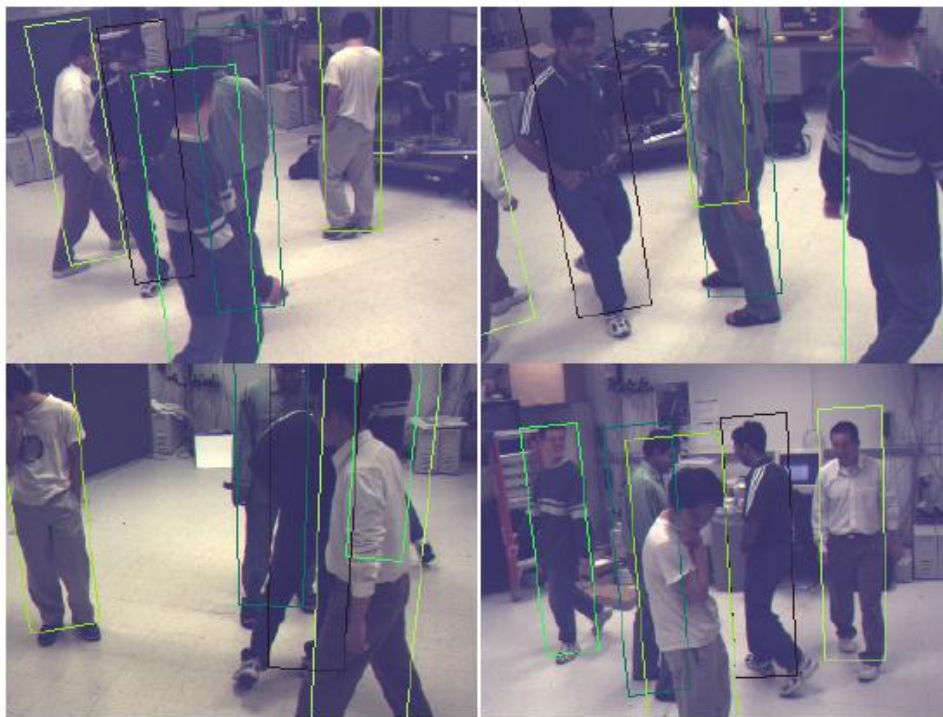
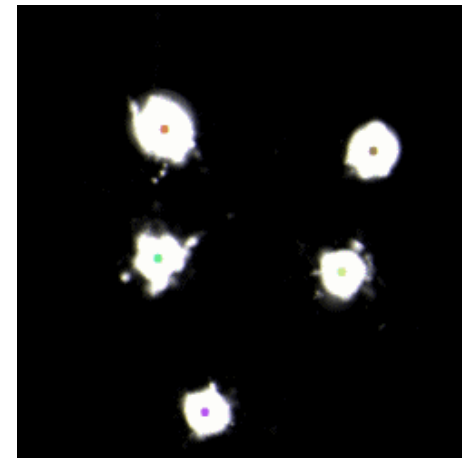


Fig. 1. Four images from a 6-perspective sequence at a particular time instant. The boxes show the positions found by the algorithm.

MITTAL, A. AND DAVIS, L. M2 tracker: A multiview approach to segmenting and tracking people in a cluttered scene. IJCV, 2003 51, 3, 189–203

多摄像机跟踪 一个例子



SLAM

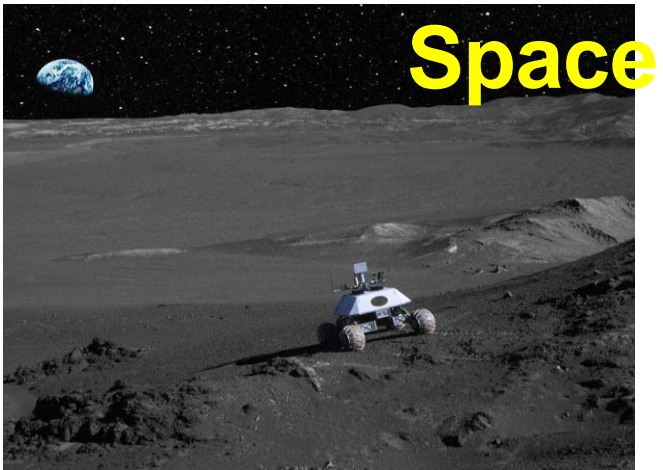
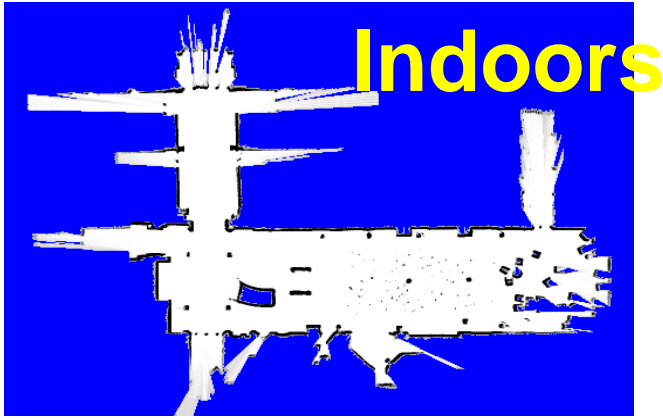
Introduction

- What is SLAM?
 - Simultaneous Localization and Map-Building
 - SLAM is a technique used by robots and autonomous vehicles to build up a map within an unknown environment while at the same time keeping track of their current position.

Introduction



Introduction



Introduction

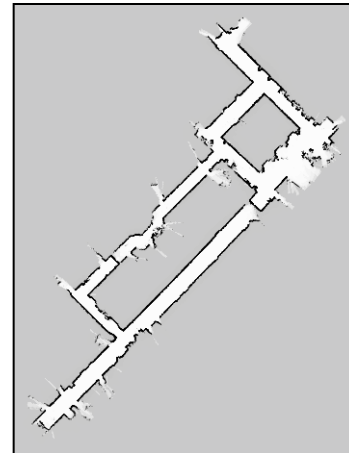
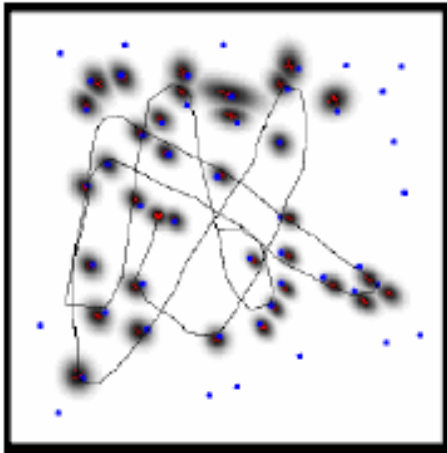
- The SLAM problem:
 - Given:
 - The robot's controls
 - The observations of nearby features
 - Estimate:
 - Localization (estimated pose)
 - Map of the environment

Map

- What is a map?
 - collection of some features
- What is a feature?
 - Point
 - Line segment
 - Patch
 - ...

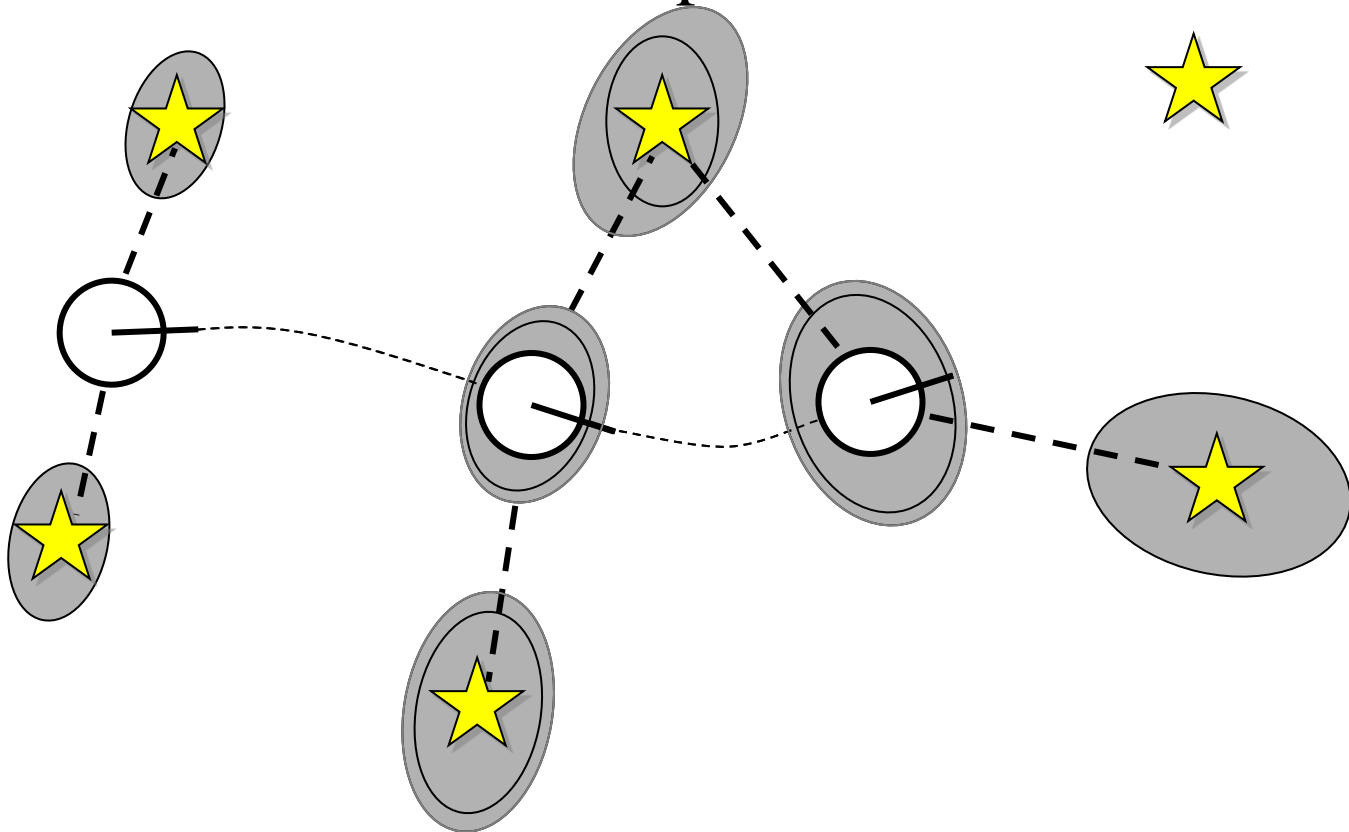
Map

- How to represent a map?
 - Landmark-based map representation
 - Grid-based map representation



Difficulties

- Chicken and egg problem
 - Camera location and map are both unknown



Difficulties

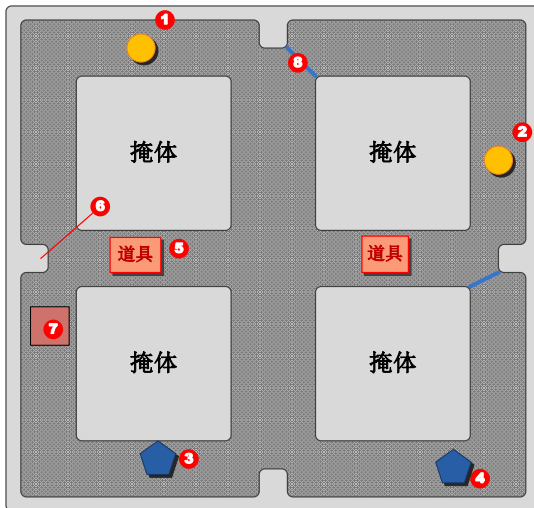
- In the real world, the mapping between observations and landmarks is unknown.
- The small error will be quickly accumulated over time steps.
- Different environment
 - Static
 - dynamic
- Uncertainties

Solution to SLAM

- State is Augmented with Pose and Map estimate
- Each time a landmark is observed
 - If the landmark/feature is new, add it to the map;
 - Otherwise update all the feature in the map;
- As more landmarks are observed, the correlation between landmarks grows.

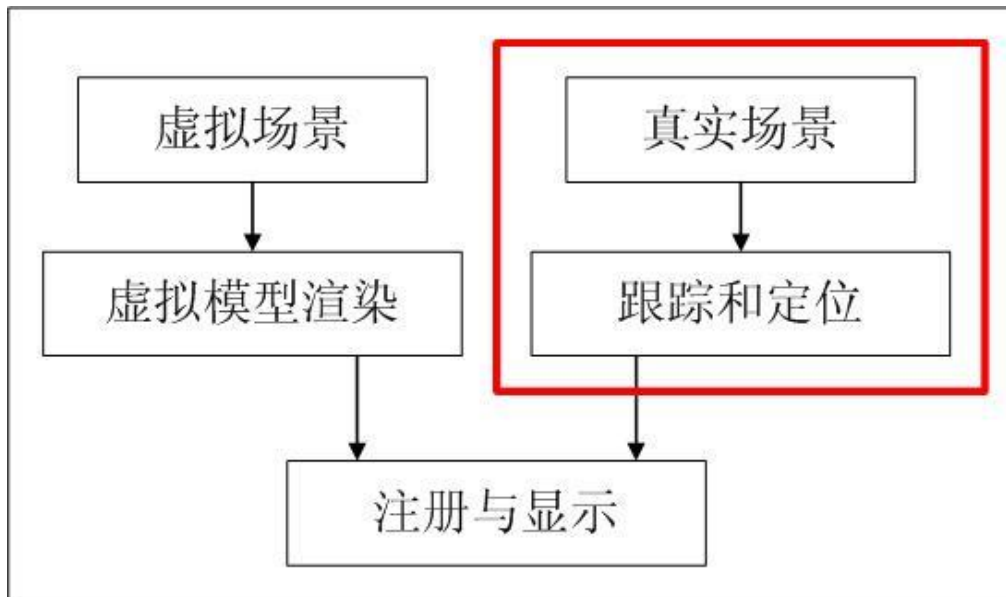
Experiments

- Background: pervasive game



Experiments

- Framechart and HMD



Discussion

- SLAM is a hard problem which has been not yet fully solved
 - Probabilistic methods which take account of sensor and process model error tend to work well;
 - Effective algorithms should be robust to poor data associations;
 - Real-time operation limits complexity of algorithms.

推荐阅读

□ 综述

- ❖ A. Yilmaz, O. Javed and M. Shah **Object Tracking: A Survey** ACM Journal of Computing Surveys, Vol. 38, No. 4, 2006

□ 卡尔曼滤波器

- ❖ Greg Welch, Gary Bishop **An Introduction to the Kalman Filter** TR 95-041

□ 粒子滤波器

- ❖ Michael Isard, Andrew Blake **CONDENSATION – conditional density propagation for visual tracking** Int. J. Computer Vision, 29, 1, 5--28, (1998)
- ❖ http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/ISARD1/condensation.html

□ 核方法

- ❖ Comanniciu D, Ramesh V, Meer P. Kernel-based object tracking. IEEE PAMI, 2003, 25(5): 564~577

谢谢