

第 16 讲: 运动分割/跟踪

- 运动分割

- 主导运动分割
- 多运动分割
- 基于区域的运动分割
- 同时运动估计和分割

- 运动跟踪

- 基于图的时空分割
- Kanade-Lucas-Tomasi (KLT) 跟踪
- 均值漂移跟踪
- 粒子滤波跟踪
- 主动轮廓跟踪
- 二维网格跟踪

- 性能评估

主导运动分割

目标:分离主要的移动的前景或背景区域。

- 1) 计算整个区域的**主要2D平移**:

$$\begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} -I_x I_t \\ -I_y I_t \end{bmatrix}$$

如果主要运动不是平移，所估计的平移为主要运动的一阶逼近。

- 2) 将与估计的主要运动相对应的所有**像素标记**如下:
 - 使用所估计的主要运动参数对两幅**图像配准**。主要运动物体将在配准图像中显示为静止的，而图像的其他部分则不是。
 - 定义**矩阵条件数的倒数** $\lambda_{\min}/\lambda_{\max}$ 作为运动的可靠性度量。
 - 如果**归一化帧差较低且其运动可靠性高**，则像素被分类为**静止**。
- 迭代步骤1-2直到收敛。
- 在该区域内使用**仿射或透视模型**来改善运动估计。
- 排除以前计算的区域**，继续下一个主要对象。

基于时域连续性的主导运动

- 1) 将**内部表示图像**初始化为:

$$\bar{s}(\mathbf{x}, 0) = s(\mathbf{x}, 0)$$

- 2) 在上一帧的主要物体的支持区域 M_{k-1} 中计算当前内部表示帧和新帧之间的**主要运动参数**。
- 3) 使用计算出的主要运动参数，将帧k-1处的内部表示图像**扭曲**到新的帧。
- 4) 使用“**多分辨率变化检测**”算法检测配准图像之间的静止区域以获得 M_k 。
- 5) **更新**内部表示图像，
$$\bar{s}(\mathbf{x}, k) = (1 - \alpha)s(\mathbf{x}, k) + \alpha \text{ warp}(\bar{s}(\mathbf{x}, k - 1), s(\mathbf{x}, k)), \quad k = 1, \dots$$

然后转到步骤2。

多运动: 参数聚类

- **种子块选择:**

- 稠密运动估计 (分层, 3步Lucas-Kanade)
- 估计 $M \times M$ 块的仿射运动参数。
- 应用统计测试来消除被视为异常值的块。

- 种子块的**K均值聚类**: 给定 K , 初始化 $\bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2, \dots, \bar{\mathbf{A}}_K$ 为任意值。
- 对于每个种子块 $n \in \{1, 2, \dots, N\}$, 找到由 $k = \arg \min_s D(\mathbf{A}_n, \bar{\mathbf{A}}_s)$ 给出的 k , 其中 s 从集合 $\{1, 2, \dots, K\}$ 内选择。
- 将 S_k 定义为仿射参数向量最接近 $\bar{\mathbf{A}}_k, k = 1, \dots, K$ 的种子块集合。然后, 更新类均值:

$$\bar{\mathbf{A}}_k = \frac{\sum_{n \in S_k} \mathbf{A}_n}{\sum_{n \in S_k} 1}$$

- 重复, 直到类均值 $\bar{\mathbf{A}}_k$ 的**变化不超过预定值**。
- **运动分割**: 基于密集运动矢量与基于仿射模型的运动矢量之间的相似度, 将每个像素分类为最终运动类别之一。如果最小距离超过阈值, 则该像素不作标记。
- 根据DFD标准对所有“**未分配的像素**”进行分类。

多运动: 最大似然 (ML)

- 初始化 A_1, A_2, \dots, A_K .

- 为每个像素 \mathbf{x} 分配运动标记 $z(\mathbf{x})$:

$$z(\mathbf{x}) = \arg \min_k \left\| \mathbf{o}(\mathbf{x}) - \mathbf{O}(\mathbf{A}_k; \mathbf{x}) \right\|^2$$

其中 k 的值从集合 $\{1, 2, \dots, K\}$ 中选择。

- 更新 A_1, A_2, \dots, A_K 为:

$$\mathbf{A}_k = \arg \min_{\mathbf{A}} \sum_{\mathbf{x} \in Z_k} \left\| v(\mathbf{x}) - P(\mathbf{A}_k; \mathbf{x}) \right\|^2$$

这可以通过闭式实现:

$$\begin{bmatrix} x_1 & x_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & x_2 & 1 \end{bmatrix} \begin{bmatrix} a_{k,1} \\ a_{k,2} \\ a_{k,3} \\ a_{k,4} \\ a_{k,5} \\ a_{k,6} \end{bmatrix} = \begin{bmatrix} x_1 + v_1(\mathbf{x}) \\ x_2 + v_2(\mathbf{x}) \end{bmatrix}$$

- 重复步骤2和3, 直到类均值 \mathbf{A}_k 的变化不大于预定值为止。

最大后验概率 (MAP)

- 寻找可最大化下式的**标记场** $\mathbf{z}(\mathbf{x})$:

$$p(\mathbf{z} | \mathbf{o}) \propto p(\mathbf{o} | \mathbf{z}) p(\mathbf{z})$$

- 似然函数**:

$$p(\mathbf{o} | \mathbf{z}) = \frac{1}{(2\sigma^2)^{M/2}} \exp \left\{ - \sum_{\mathbf{x}_i} \frac{(\mathbf{o}(\mathbf{x}_i) - O(\mathbf{A}_{\mathbf{z}(\mathbf{x}_i)}; \mathbf{x}_i))^2}{2\sigma^2} \right\}$$

- 标记场的**先验概率密度函数**:

$$p(\mathbf{z}) = \frac{1}{Q} \sum_{\omega \in \Omega} \exp \left\{ - \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in N_{\mathbf{x}_i}} V_c(z(\mathbf{x}_i), z(\mathbf{x}_j)) \right\} \delta(\mathbf{z} - \omega) \quad Q = \sum_{\omega \in \Omega} \exp \{ - U(\omega) \}$$

- 整体代价函数**:

$$E(\mathbf{z}) = \frac{1}{2\sigma^2} \sum_{\mathbf{x}_i} (\mathbf{o}(\mathbf{x}_i) - O(\mathbf{A}_{\mathbf{z}(\mathbf{x}_i)}; \mathbf{x}_i))^2 + \sum_{\mathbf{x}_i} \sum_{\mathbf{x}_j \in N_{\mathbf{x}_i}} V_c(z(\mathbf{x}_i), z(\mathbf{x}_j))$$

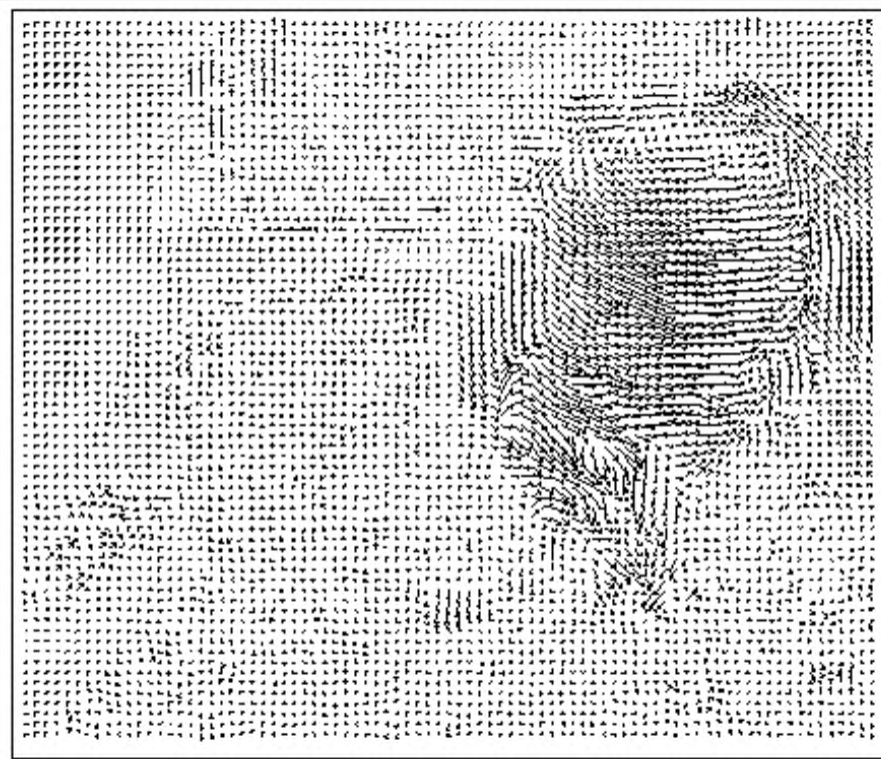
MAP 方法 (cont'd)

- 从光流矢量的**初始标记 $z(x)$** 开始。使用**最小二乘法**计算每个区域的模型运动参数。
- 在每个像素 x 处**更新分割标记**，如下所示：
 - **随机扰动**标记 $z(x_i)$ 。
 - 决定**是否接受或拒绝**基于 ΔE 的值的扰动（每个像素的更新仅取决于邻近像素的标记。）
- 处理所有像素一次后，使用新标记**重新估计**每个区域的**运动模型参数**。
- 如果满足停止条件，则退出，否则转到步骤2。

运动分割：示例

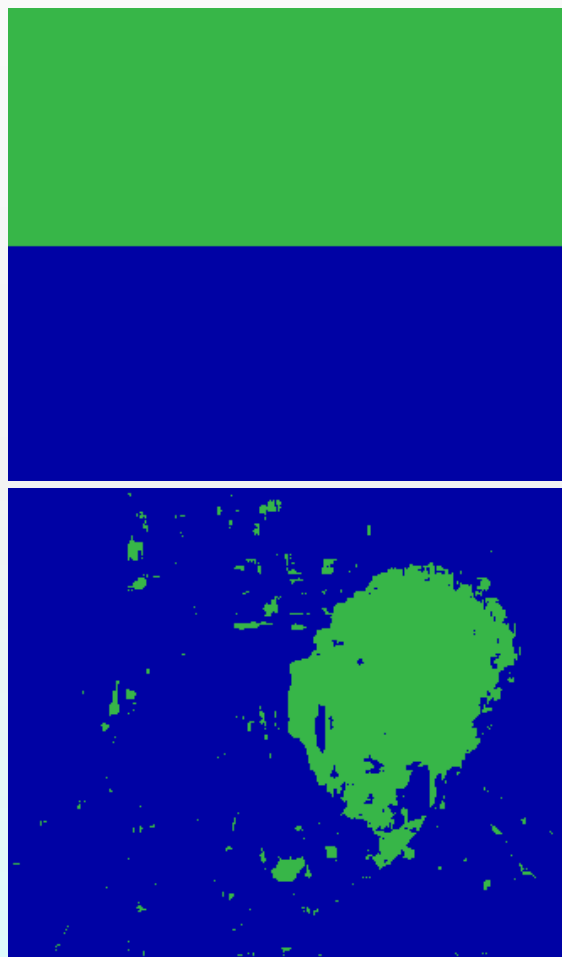


Mother and Daughter



稠密运动场

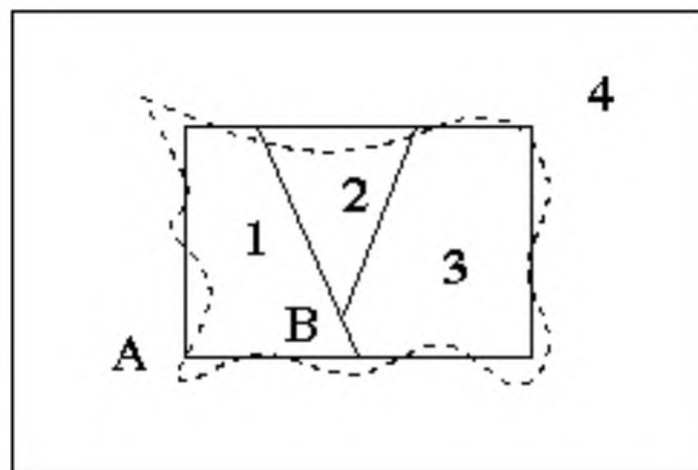
示例：参数聚类



初始条件 $K=2$

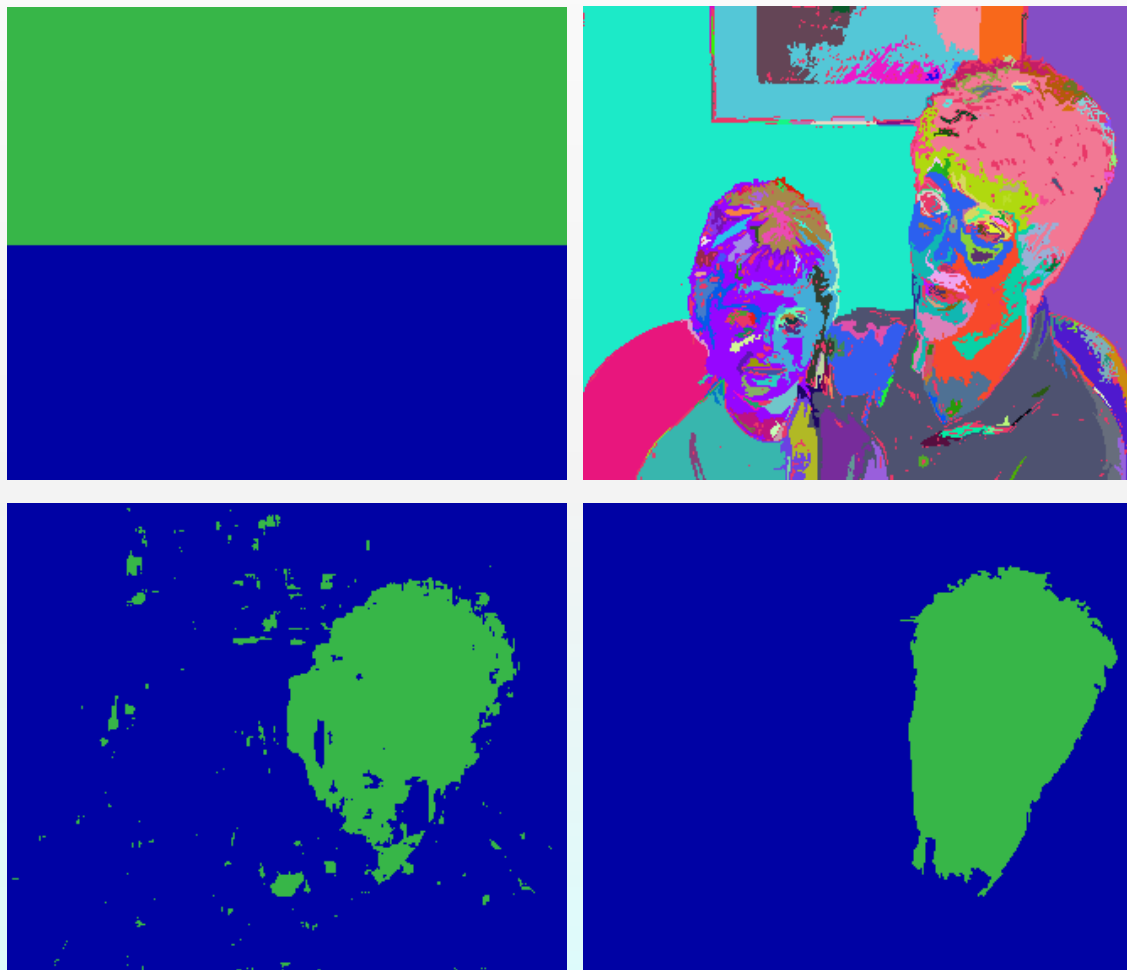
基于区域的运动分割

色彩和运动融合

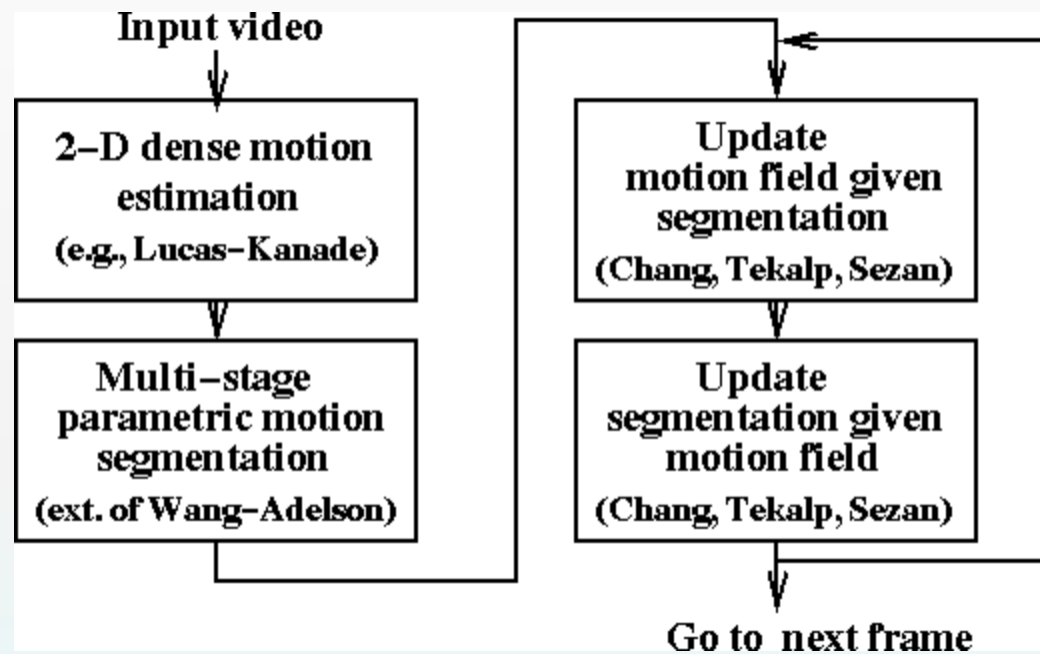


- 执行**基于像素的运动分割**（虚线）来确定运动**类的数量**，以及每个**类的参数模型**。
- 执行**颜色分割**以找到均质的颜色区域（实线）。
- 根据运动标准、DFD标准或它们的组合，将每个颜色区域**分配到各运动类别之一**。

基于区域的分割: 示例



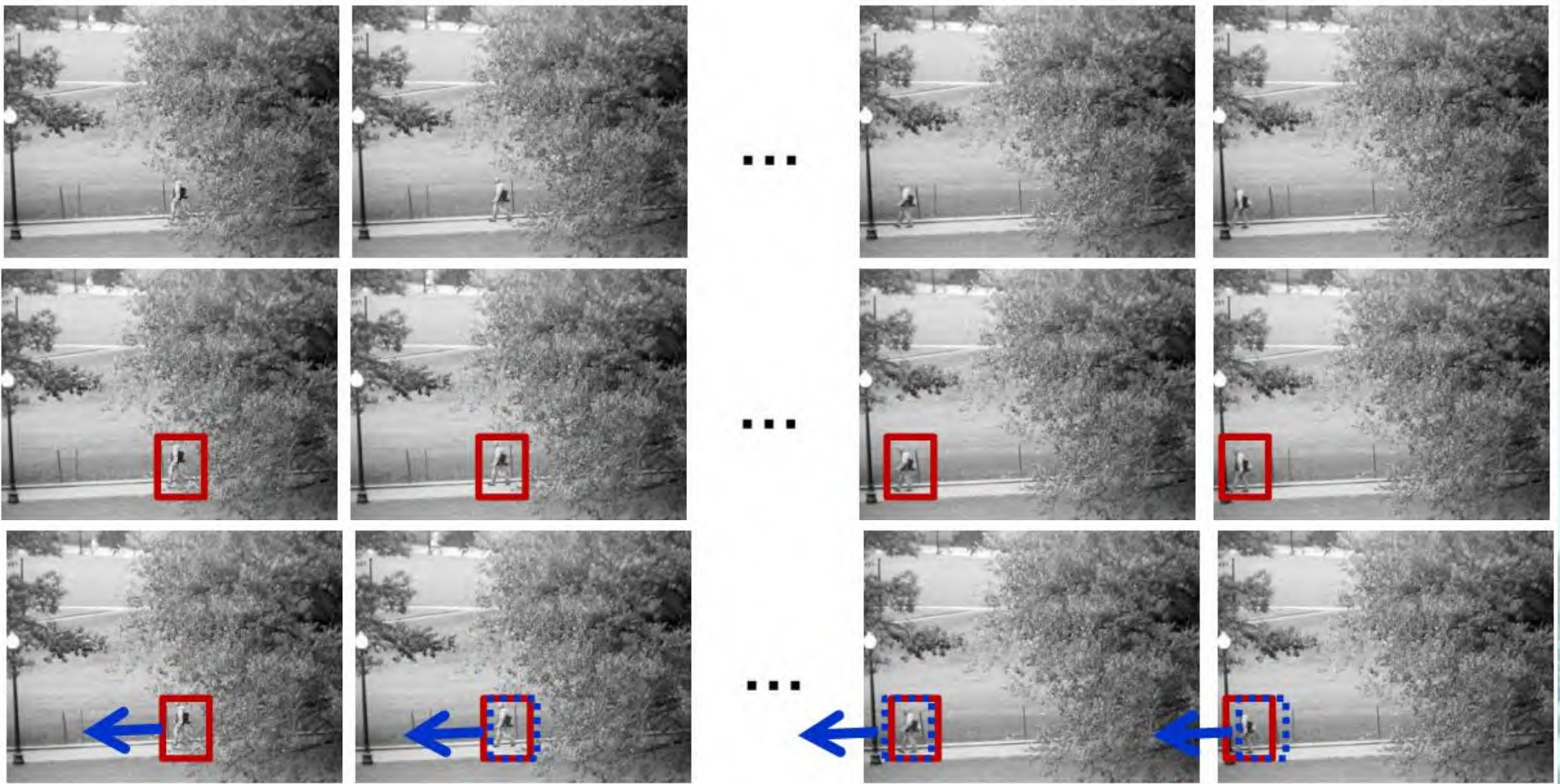
同时运动估计和分割



- 可以利用基于使用 **吉布斯先验知识的MAP** 标准进行更新。

运动跟踪

运动跟踪：一种**因果的时空视频分割**（计算属于单个或多个目标的在时域上相关联的区域）



基于图的时空分割和跟踪

- 生成**超体素**，为均质的时空区域。
- **基于图的分割**可以推广到通过**构建3D图**将视频分割成**超体素**。但此时不具有因果性，因为这需要用到整个视频序列。基于片段的处理方法可降低延迟以及内存的消耗。
- 具有**因果性基于图**的分割首先使用Felzenszwalb-Huttenlocher [Fel 04] 方法将帧 k 分割成超像素。然后，形成将帧 $k-1$ 中的区域（在前一步骤中计算）与帧 k 中的超像素**相连接的图**。连接边缘的权重取决于区域（帧 $k-1$ ）和连接超像素（帧 k ）中的像素数量，它们的平均颜色的差异以及它们的质心之间的距离。帧 k 的最终分割是通过**最小生成森林过程**来计算的。
- 基于图的视频体分割方法是**全自动**的，不需要初始化或初始目标检测；因此，可以在没有严格实时要求的一些跟踪应用之前作为**预处理**。

Kanade-Lucas-Tomasi (KLT) 跟踪

- **特征跟踪 – 跟踪好特征**
- 使用特征检测器（如**Harris角检测器**）在初始帧中选择一组“好特征点”。
- 使用具有基于关于**每个特征点的局部模板**（例如， 15×15 ）的平移或仿射运动模型的Lukas-Kanade (LK) 运动估计来为每个特征点找到帧到帧的对应矢量[Bak 04]。
- 如果可以找到LK运动估计的数值稳定解，则可以可靠地跟踪特征，这要求**矩阵M为适定的**，如果 $\min(\lambda_1, \lambda_2) \gg \lambda$ ，则相应的特征为好的特征，其中 λ_1 和 λ_2 是**M**的特征值， λ 为阈值。
- 对于每个特征点，**验证**在每帧内跟踪的**良好性**。可以去除一些特征（以消除那些被遮挡或不能被准确跟踪的特征），并且可以周期地（例如，每5帧）添加新的特征。

KLT 跟踪 (cont' d)

- **模板跟踪**：KLT框架可用于模板跟踪。
- 在第一帧中**提取对象的示例外观**作为模板，并且使用模板的参数扭曲在剩余帧中跟踪对象的示例外观。
- 其中较大挑战是在跟踪过程中**处理对象外观的变化**，包括3D运动，形状变化，照明变化，相机运动和遮挡。
- 解决方案：在当前模板位置**用新的图像更新每n个帧的模板**。
- 产生问题：“**漂移**”。每次更新模板时，会在时间上累积模板位置引入的小错误。
- 为避免漂移，可**保留多个模板**，包括初始模板。模板首先使用当前模板位置处的对象的图像进行更新，然后与初始模板对齐。

均值漂移模板跟踪

- 目标是由**颜色直方图 (pdf)** 来刻画, $h_l, l = 1, \dots, L$, 其中L是在 $N \times N$ 边框内的像素取值数。迭代计算当前帧中可最大化Bhattacharyya系数的位置 $\bar{\mathbf{x}}$:

$$\rho(\bar{\mathbf{x}}) = \sum_{l=1}^L \sqrt{p_l(\bar{\mathbf{x}}) h_l}$$

在目标模板的直方图和以当前帧中的位置 \mathbf{x} 为中心的边界框的直方图之间。

- 通过使用**均值漂移**算法计算出 **$\rho(\mathbf{x})$ 的最大值 (模式)**, 其包括三个主要步骤: **直方图计算**, **权重计算**和**查找新位置**。完整的算法由[Com 03]给出:
- 1. 初始化**: 在初始帧 (帧0) 中选择要跟踪的目标 (模板) 的 $N \times N$ 边界框。计算模板的L-bin颜色直方图 $h_l, l = 1, \dots, L$, 转到帧 $j = 1$ 。

均值漂移跟踪 (cont'd)

- 2. 设置**均值漂移迭代** $k = 0$ 。初始化边界框的中心 $\bar{\mathbf{x}}_k^{(j)}$ 。
- 3. 计算当前边界框的**颜色直方图** $\{p_l(\bar{\mathbf{x}}_k^{(j)})\}$ 。
- 4. 计算边界框内每个像素 $i = 1, \dots, N^2$ 的**权重值**:

$$w_i(\bar{\mathbf{x}}_k^{(j)}) = \sum_{l=1}^L \delta[s(\mathbf{x}_i) - l] \sqrt{\frac{h_l}{p_l(\bar{\mathbf{x}}_k^{(j)})}}$$

其中 $\delta[s(\mathbf{x}_i) - l]$ 为直方图 bin，对应于像素 $s(\mathbf{x}_i)$ 的颜色。

- 5. 使用均值漂移迭代方法**更新边界框的中心**

$$\bar{\mathbf{x}}_{k+1}^{(j)} = \frac{\sum_{i=1}^{N^2} \mathbf{x}_i w_i(\bar{\mathbf{x}}_k^{(j)})}{\sum_{i=1}^{N^2} w_i(\bar{\mathbf{x}}_k^{(j)})}$$

- 6. 如果 $\|\bar{\mathbf{x}}_{k+1}^{(j)} - \bar{\mathbf{x}}_k^{(j)}\| < \delta$, 增加帧计数 $j = j + 1$ 并转到步骤 2。
否则, 设置 $k = k + 1$ 并转到步骤 3 继续均值漂移迭代。

卡尔曼滤波器

- 可递归估计系统的状态
- 前提：系统模型和观察模型都是线性的，且其PDF都是高斯的

观察序列的状态变量线性表示：

$$\mathbf{y}(k) = \mathbf{H}(k)\mathbf{z}(k) + \mathbf{n}(k)$$

观察矩阵

零均值观察白噪声，其协方差矩阵 $\mathbf{R}(k)$

状态转移方程： $\mathbf{z}(k) = \mathbf{\Phi}(k, k-1)\mathbf{z}(k-1) + \mathbf{w}(k), \quad k = 1, 2, \dots, N$

状态转移矩阵

零均值白噪声，其协方差矩阵为 $\mathbf{Q}(k)$

卡尔曼滤波器

状态更新过程:

1. 预测，状态预测和误差矩阵预测
2. 更新，卡尔曼增益计算、状态更新和误差矩阵更新

状态预测方程: $\hat{\mathbf{z}}_b(k) = \mathbf{\Phi}(k, k-1)\hat{\mathbf{z}}_a(k-1)$

协方差预测方程: $\mathbf{P}_b(k) = \mathbf{\Phi}(k, k-1)\mathbf{P}_a(k-1)\mathbf{\Phi}^T(k, k-1) + \mathbf{Q}(k)$

Kalman增益方程: $\mathbf{K}(k) = \mathbf{P}_b(k)\mathbf{H}^T(k)[\mathbf{H}(k)\mathbf{P}_b(k)\mathbf{H}^T(k) + \mathbf{R}(k)]^{-1}$

状态更新方程: $\hat{\mathbf{z}}_a(k) = \hat{\mathbf{z}}_b(k) + \mathbf{K}(k)[\mathbf{y}(k) - \mathbf{H}(k)\hat{\mathbf{z}}_b(k)]$

协方差更新方程: $\mathbf{P}_a(k) = \mathbf{P}_b(k) - \mathbf{K}(k)\mathbf{H}(k)\mathbf{P}_b(k)$

粒子滤波器跟踪

- **粒子滤波**提供了一种鲁棒的跟踪框架，因为它同时考虑了多个状态假设。系统可以是非线性的，噪声也可以是非高斯的。
- **基本思想**是通过一个加权的样点集（粒子）来表示概率分布，以样本均值代替积分运算。

初始化:

- 选择一个边界形状，从有利于盒子中/附近像素的先验分布 $p(\mathbf{x}_0)$ 采样 N 个随机粒子 $\{\mathbf{x}_0^{(l)}\}_{l=1}^N$ 。设置权重 $\{W_0^{(l)} = 1/N\}_{l=1}^N$
- 对每一帧 $k = 0, 1, 2, \dots$

预测（密度扩散）：

- 给定动态模型，扩散每个粒子 $\mathbf{x}_{k+1}^{(l)} \sim p(\mathbf{x}_{k+1} | \mathbf{x}_k^{(l)})$, $l = 1, \dots, N$ 。

更新:

- 给定观察值 z_{k+1} , 基于作为 $p(z_{k+1} | \mathbf{x}_{k+1}^{(l)})$ 函数的似然函数 $L(z_{k+1} | \mathbf{x}_{k+1}^{(l)})$ ，为每个粒子 $\mathbf{x}_{k+1}^{(l)}$ 计算权重 $W_{k+1}^{(l)} \propto W_k^{(l)} L(z_{k+1} | \mathbf{x}_{k+1}^{(l)})$

粒子滤波跟踪 (cont'd)

- 归一化权重 $\hat{W}_{k+1}^{(l)} = \frac{W_{k+1}^{(l)}}{\sum_{l=1}^N W_{k+1}^{(l)}}$

- 状态估计 $\hat{\mathbf{x}}_{k+1}$ 为扩散粒子的加权平均:

$$\hat{\mathbf{x}}_{k+1} = \sum_{l=1}^N \hat{W}_{k+1}^{(l)} \mathbf{x}_{k+1}^{(l)}$$

- 为了抑制低权重的粒子, 估计粒子的有效数:

$$N_{\text{eff}} = \frac{1}{\sum_{l=1}^N (\hat{W}_{k+1}^{(l)})^2}$$

- 如果 $N_{\text{eff}} \leq N_{\text{thres}}$, 则进行重采样; 否则增加 k 并转到步骤 2

- 重采样:

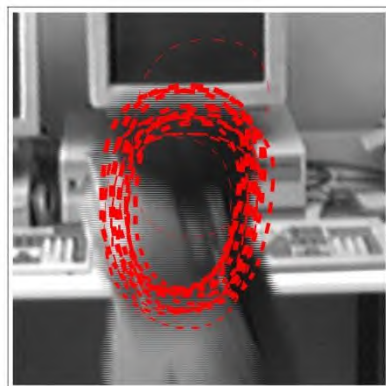
应用[Aru 02]中给出的重采样算法。

- 设定新的权重 $W_{k+1}^{(l)} = \hat{W}_{k+1}^{(l)} = 1/N, l = 1, \dots, N.$

主动轮廓跟踪

凝聚（条件密度传播）：在复杂背景下追踪曲线。因此，每个样本（粒子） $\{\mathbf{x}_k(l)\}_{l=1}^N$ 是具有不同位置和形状的曲线（而不是单个像素），其粗细程度与权重 $W_k^{(l)}$ 成比例，且这些曲线的加权平均值被计算作为状态估计。

每个粒子为一条简单轮廓



粒子的加权平均



Isard and Blake [Isa 98]

- **运动蛇形**：通过基于搜索的关于不同能量函数的最小化，将活动轮廓模型拟合到期望对象。轮廓被分成多个段，使得对于每个轮廓段将存在两个候选仿射模型，每个侧面一个。

运动蛇形算法

- 通过沿期望轮廓标记一组节点来指定一个**初始轮廓**，以定义一个近似的多边形，然后通过最小化帧内能量项来逼近期望物体的轮廓；
- 通过基于**局部曲率**，**颜色**和**运动矢量**选择一组特征节点，将蛇形线划分为**不重叠的段**[Fu 00]；
- 在每个节点处沿着**角等分线**对所选像素的轮廓内外估计运动矢量。
- 在下一帧中为每个轮廓段获取**两个预测位置**（基于使用轮廓内外的运动矢量的估计仿射模型）
- 基于**预测能量的最小化**，为每个轮廓段**选择这两个预测位置之一**
- 使用帧间和帧内能量项来**优化预测轮廓**，并转到步骤2开始处理下一帧。

二维网格跟踪

- **2D 网格**是将图像区域（分割）分割成多边形块的平面图形。块的顶点称为节点。
- 块通常是三角形或四边形的，分别形成**三角形或四边形网格**。当前帧中的三角/多边形块通过节点指向参考帧中的各个块的移动而变形，并使用称为纹理映射的参数模型将参考帧中每个片段内的纹理变形到当前帧上。



性能度量

- 可以将性能度量归类为**需要实值（GT）数据**和**不需要GT数据**的。
- 许多**在线分割数据集**提供**手工标记的GT分割**进行基准测试。用于开放的视频系统评估开发环境（ODViS）[Jay 02]允许用户对预先录制的视频生成GT数据。
- **不依赖GT数据**的度量方法通常评估区域内同质性，区域间差异以及结果的空间或时空一致性。
- 伯克利分割数据集和基准测试
- <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>
- 视觉跟踪基准测试
- <https://sites.google.com/site/trackerbenchmark/benchmarks/v10>

习题

- 大作业：Matlab Exercises 中任选一道题。
- 注：本章的“习题”实际为“Matlab练习”。