

第 15 讲: 图像/视频分割

- 图像分割

- 基于像素的分割
 - 阈值, K均值, 均值漂移
- 区域分割
 - 贝叶斯, 基于图的方法
- 轮廓分割
 - 主动轮廓模型

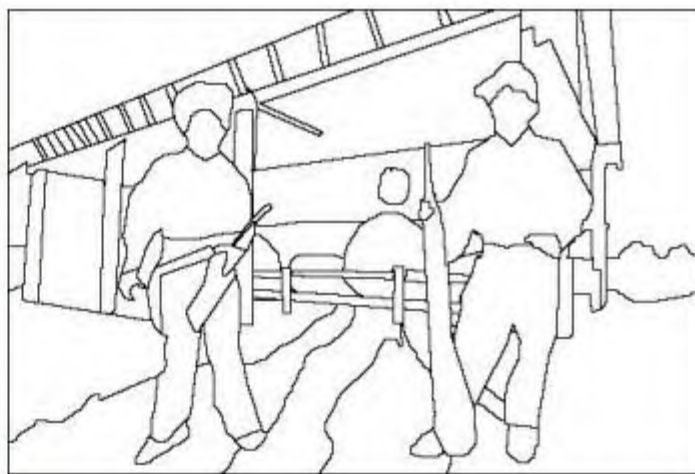
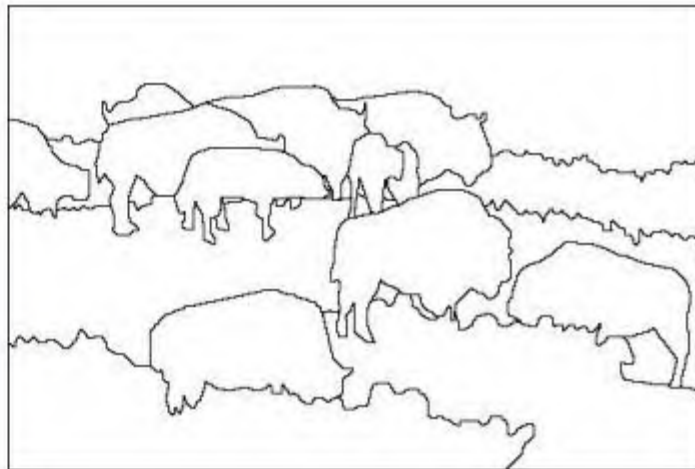
- 变化检测

- 镜头边界检测
- 背景消除

- 视频抠图

- 颜色键控
- 转描机技术

分割



简介

- **视频分割**：是指将图像（帧）或视频划分为在某些特征空间中为均质的空间、时间或时空区域。
 - **时间分割**：基于帧的相似性将视频分割成镜头
 - **空间分割**：将每个视频帧划分成均质区域，这可以通过基于颜色相似度（帧内图像分割）对每帧单独分割来实现。
 - **时空分割**：产生时间上相连的空间分割或物体轨迹。例如通过帧间分割（基于颜色和帧之间的运动相似度）或颜色和运动跟踪来实现。
- 与任何分割问题一样，有效的视频分割需要适当的**特征选择**和适当的**距离度量**。

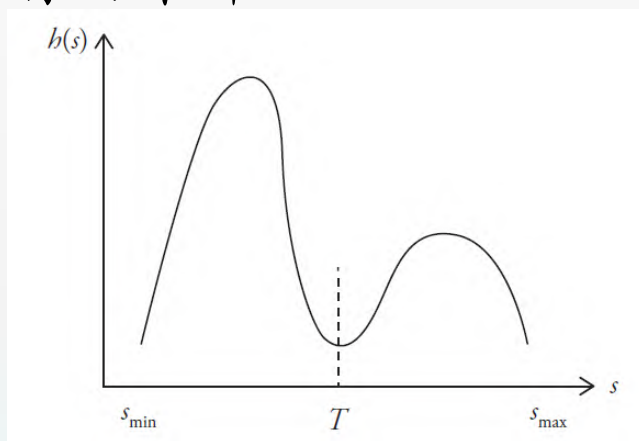
简介(cont'd)

影响特定分割方法选择的因素包括：

- **分割精度**：如果使用分割来提高压缩效率或码率控制，则计算出的边界和实际对象边界之间是否对齐可能不是很大的问题。另一方面，对于基于对象的视频制作/编辑或形状相似性匹配，估计的边界和实际的对象边界应该完美对齐，甚至单个像素误差可能也是不可容忍的。
- **内容复杂性**：复杂性取决于相机运动量，颜色和纹理均匀性，对象之间的对比度，运动平滑度，进入和离开场景的物体，物体形状规律性，切换频率和特殊效果等。显然，更复杂场景需要更复杂的分割算法，例如，检测镜头切换要比检测擦除或淡化效果更容易。
- **实时性能**：如果必须实时进行分割，例如在视频会议中进行码率控制，则必须使用全自动算法。另一方面，对于离线应用，人们可以采用半自动交互算法，如视频编辑，索引或离线视频编码，或获得语义上有意义的分割。

图像分割：阈值

- **双峰直方图**：考虑在深色背景上由明亮物体组成的图像 $s(x_1, x_2)$ ，它的直方图将有两个峰。



- **二值化**：基于灰度信息从背景分割对象的直观方法是选择分离这两个主要模式（峰值）的阈值 T 。

$$z(x_1, x_2) = \begin{cases} 1 & \text{if } s(x_1, x_2) > T \\ 0 & \text{其他} \end{cases}$$

- 阈值技术可以分为两大类：**全局**和**局部**。

阈值 (cont'd)

- **Otsu 阈值** 可将两类分离，使得**类内方差最小**，**类间方差最大**。
- 作为阈值 k 的函数的类内方差 σ_w^2 由两个类 σ_1^2 和 σ_2^2 的方差的概率加权和来定义，由：

$$\sigma_w^2(k) = P_1(k) \sigma_1^2 + P_2(k) \sigma_2^2$$

其中 $P_1(k) = \sum_{i=0}^k p(i)$ 是两类被阈值 k 分开后第 1 类的概率， $p(i)$ 表示归一化直方图，而 $P_2(k) = 1 - P_1(k)$ 。

- Otsu 说明，应尽量**减少类内方差**，相当于**最大化类间方差**。

$$\sigma_B^2(k) = \sigma^2 - \sigma_w^2(k) = P_1(k)P_2(k)[\mu_1(k) - \mu_2(k)]^2$$

其中 σ^2 是组合方差，类均值 $\mu_1(k)$ 与 $\mu_2(k)$ 为：

$$\mu_1(k) = \sum_{i=0}^k \frac{i p(i)}{P_1(k)} \quad \mu_2(k) = \sum_{i=k+1}^{L-1} \frac{i p(i)}{P_2(k)}$$

阈值 (cont'd)

- **Otsu 阈值算法 [Ots 79]**

- 计算归一化直方图 $p(i)$, $i = 0, \dots, L-1$, 共 L 级。
- 遍历所有可能的阈值 $k = 0, \dots, L-1$.
 - 计算 $P_1(k), P_2(k)$, 与 $\mu_1(k), \mu_2(k)$
 - 计算 $\sigma_B^2(k) = P_1(k)P_2(k)[\mu_1(k) - \mu_2(k)]^2$
- Otsu阈值由 $k^* = \arg \max_{0 \leq k \leq L-1} \sigma_B^2(k)$ 给出。如果最大值不是唯一的, 则 k^* 由 k 个最大值的平均值给出。
- 如果直方图具有 **$K > 2$** 个显著模式 (峰值), 则需要 **$K-1$ 阈值**将像素分类为 K 个分割。将Otsu方法扩展到多级阈值处理的情况称为**多Otsu方法**[Lia 01]。
- 随着**模式数量的增加**, 确定可靠的阈值变得更加困难。

聚类：K均值

- 将每个样本分配给**具有最近聚类平均值的类**。
- 令： $\mathbf{x}=(x_1, x_2)$ 表示像素的坐标， $s(\mathbf{x})$ 表示其灰度级。K-means方法旨在最小化性能指标：

$$J = \sum_{l=1}^K \sum_{\mathbf{x} \in \Lambda_l^{(i)}} \|s(\mathbf{x}) - \mu_l^{(i+1)}\|^2$$

其中 $\Lambda_l^{(i)}$ 表示第 i 次迭代后分配给聚类 l 的样本集， μ_l 表示第 l 个簇的平均值。指数 J 测量每个样本距它们各自类的距离的总和。

- **K-means算法**通常收敛到索引 J 的**局部最小值**，因此不同的初始化可能导致不同的分割结果。
- K均值聚类中最大的挑战是**确定合适的类数**，通常假定是已知的。可以尝试不同的 K 值，直到达到所需的聚类质量。

K均值 (cont'd)

K均值算法可归纳如下:

- 任意选择 **K 个初始聚类均值**: $\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_K^{(1)}$.
- 在第*i*次迭代中将**每个像素 \mathbf{x}** , 按照下式**分配到 K 个聚类中的一个**:

$$\mathbf{x} \in \Lambda_j^{(i)} \quad \text{if} \quad \|s(\mathbf{x}) - \mu_j^{(i)}\| < \|s(\mathbf{x}) - \mu_l^{(i)}\| \quad \text{for all } l = 1, 2, \dots, K, \quad l \neq j$$

其中 $\Lambda_l^{(i)}$ 表示聚类中心为 $\mu_l^{(i)}$ 的样值集合。即将每个样值分配到聚类均值离其最近的类中。

- 更新聚类的均值** $\mu_l^{(i+1)}$ 作为所有 $\Lambda_l^{(i)}$ 中样值的均值, $l = 1, 2, \dots, K$

$$\mu_l^{(i+1)} = \frac{1}{N_l} \sum_{\mathbf{x} \in \Lambda_l^{(i)}} s(\mathbf{x}) \quad l = 1, 2, \dots, K$$

其中 N_l 为 $\Lambda_l^{(i)}$ 中样值的数量。

- 如果 $\mu_l^{(i+1)} = \mu_l^{(i)}$, 对所有的 $l = 1, 2, \dots, K$, 算法**已经收敛**, 则迭代结束。否则转到步骤 2。

聚类: 均值漂移 (MS)

- **MS** 是一种**模式发现算法**，它将每个像素分配到离其最近聚类模式的类。与 K 均值不同，MS **不用假定类数已知**。
- 对于每个数据点 x_j
 - 取一个**窗口**，其大小为 h ，包含 n 个样值，且以数据点 x_j 为中心。
 - 计算**加权均值** (重心或质心)，其中使用核 $g(\cdot)$ 计算窗口内各数据的权重。

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

- **移动窗口中心到新的均值**并重复该过程直到收敛。
- MS 为**加权均值**与**核中心**之间的差值: $m(x_j) - x_j$ 。

均值漂移(cont'd)

- 合并以上三个步骤，假设**高斯核** $g(x) = e^{-x}$ ，每个数据点 x_j 核 y_j (模式)的中心可**迭代计算**：

$$y_j^{(k+1)} = \frac{\sum_{i=1}^n x_i e^{-\left\| \frac{y_j^{(k)} - x_i}{h} \right\|^2}}{\sum_{i=1}^n e^{-\left\| \frac{y_j^{(k)} - x_i}{h} \right\|^2}}, \quad k = 1, 2, \dots$$

其中 $y_{j(0)} = x_j$ 为**窗的初始中心** (以数据点为中心)。

- 通常，均值漂移在**每个数据点**上运行，有时则会在特征空间中均匀选择出的点上运行[Com 02]。
- 算法的稳定点**是密度函数的模式。与同一模式相关联的所有点属于同一个类。
- MS算法是**自适应梯度下降法**，保证收敛到具有零梯度分布的点（模式）[Com 02]。如果数据点稀疏，则MS步长很大，而步长在接近模式处较小。

贝叶斯分割方法

- **贝叶斯方法**可以被视为具有**空间连通性约束**的聚类。
- **分割场 z 的估计**被定义为给定观察图像 g **最大化后验pdf $p(z|g)$** 的估计。使用贝叶斯规则，

$$p(z|g) \approx p(g|z)p(z)$$

- **类条件pdf**: 给定分割标记 z 的数据 g 的条件pdf $p(g|z)$ ，将分割标记与数据相关联。
- Derin和Elliott [Der 87]将每个图像**区域的强度**建模为**常数**，由标量 $\mu_l, l = 1, 2, \dots, K$ 表示。也就是说，平均向量 μ 的元素达到 K 个不同的值， $\mu_l, l = 1, 2, \dots, K$ 。然后，条件pdf可以表示为：

$$p(g|z) \approx e^{-\sum_x \frac{(g(x) - \mu_{z(x)})^2}{2\sigma_z^2}}$$

其中 $z(x) = l$ 表示将像素 x 分配到区域 $l = 1, 2, \dots, K$ 。

- 对 $p(g|z)$ 的最大化即为**最大似然 (ML)** 图像分割。

贝叶斯分割方法 (cont'd)

- **标记的先验pdf**: $p(\mathbf{z})$ 是先验的pdf, 表示对**分割的期望**, 也就是对分割标记**施加空间连接约束**。

$$p(\mathbf{z}) = \frac{1}{Q} \sum_{\omega \in \Omega} e^{-U(\mathbf{z}/T)} \delta(\mathbf{z} - \omega)$$

其中 Ω 表示 \mathbf{z} 的有限样本空间, $\delta(\cdot)$ 是Dirac函数, T 是**温度参数**, 归一化常数 $Q = \sum_{\omega \in \Omega} e^{-U(\omega)}$ 称为分区函数。 $U(\mathbf{z})$ 被称为由 $U(\mathbf{z}) = \sum_{C \in \mathcal{C}} V_C(\mathbf{z})$ 定义的**吉布斯势**, 其中 \mathcal{C} 是所有团的集合, V_C 是个体团势函数。

- **单个像素团的势**反映了不同标记的先验概率:

$$V_C(z(\mathbf{x})) = \alpha_l \text{ if } z(\mathbf{x}) = l \text{ and } \mathbf{x} \in C \text{ for } l = 1, 2, \dots, K$$

α_l 越小, 为区域 l 的可能性就越大, $l = 1, 2, \dots, K$ 。

- 分割的空间连通性可以通过**两像素团的势**来施加:

$$V_C(z(\mathbf{x}_i), z(\mathbf{x}_j)) = \begin{cases} -\beta & \text{if } z(\mathbf{x}_i) = z(\mathbf{x}_j) \text{ and } \mathbf{x}_i, \mathbf{x}_j \in C \\ \beta & \text{if } z(\mathbf{x}_i) \neq z(\mathbf{x}_j) \text{ and } \mathbf{x}_i, \mathbf{x}_j \in C \end{cases}$$

其中 β 为正。值 β 越大, 平滑度约束越强。

贝叶斯分割方法 (cont'd)

- 标记的后验pdf: 结合条件pdf和先验pdf,

$$p(\mathbf{z}|\mathbf{g}) = e^{-\sum_{\mathbf{x}} \frac{1}{2\sigma_{\xi}^2} [g(\mathbf{x}) - \mu_{\mathbf{z}(\mathbf{x})}]^2 - \sum_{C \in \mathcal{C}} V_C(\mathbf{z})}$$

- 最大化后验pdf以找到 $\mu_l, l = 1, 2, \dots, K$ 的估计和所需的分割标记 \mathbf{z} 。注意, 最大化上式是相当于最小化代价函数:

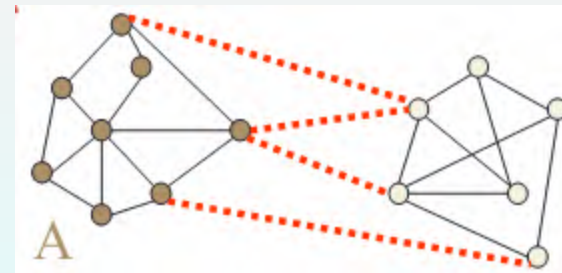
$$E(\mathbf{z}) = \sum_{\mathbf{x}} \frac{1}{2\sigma_{\xi}^2} [g(\mathbf{x}) - \mu_{\mathbf{z}(\mathbf{x})}]^2 + \sum_{C \in \mathcal{C}} V_C(\mathbf{z})$$

- MAP分割问题的解决方案可以通过Monte-Carlo方法获得, 例如模拟退火, 以达到全局最优。然而, 经常会采用次优的贪婪方法, 虽然达不到最优化, 但可以减少计算复杂度。
- 贪婪算法的解: 是一个迭代过程, 即在两个步骤之间进行迭代:
 - 给定 $\mu_{\mathbf{z}(\mathbf{x})}$ 通过最小化 $E(\mathbf{z})$ 估计 \mathbf{z} : 从类均值的一些初始估计开始, 并将每个像素分配给K类中的一个。
 - 使用最新估计的标签 $\mathbf{z}(\mathbf{x})$ 更新类均值 $\mu_{\mathbf{z}(\mathbf{x})}$ 。
- 特殊情况 - K-means: 关闭空间平滑度约束, 即忽略第二项。

基于图的分割方法

- 令 $G = (V, E, W)$ 表示一个图，其中 V 是节点集合， E 是连接各节点的边集合（连接）， W 是边缘关联矩阵。一对节点 i 和 j 通过具有权重 $w(i, j) = w(j, i) > 0$ 的边连接，它们之间的关联性（不相似性）的度量。节点可以是单个像素或超像素。
- 可以通过去除连接两个部分的边将图形分割成两个不相交的集合 A 和 $B = V - A$ 。
- 两组之间的不相似度可以计算为去除边后的总权重，这被称为图论中的图割。

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} w(i, j)$$



- 归一化图割 (Ncut) 准则:

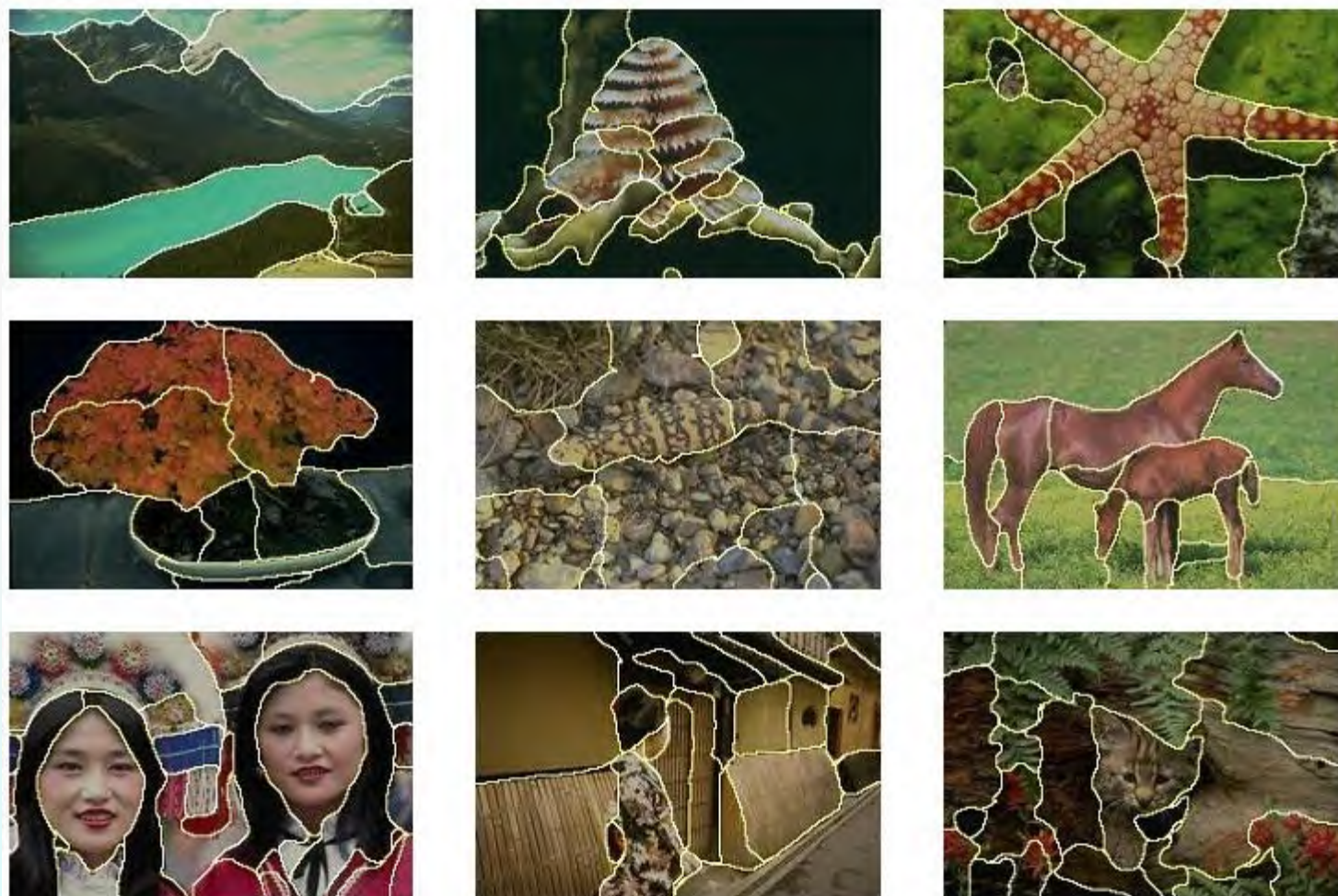
$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)}$$

其中 $\text{assoc}(A, V)$ 和 $\text{assoc}(B, V)$ 分别表示从 A 和 B 中的节点到图中所有节点的总连接数，它有利于出现相等大小的区域。

归一化图割

- 当图为**非平面**时，寻找全局最优解是**NP难问题**，此时可采用近似方法。
- Shi和Malik [Shi 00]: 令 $d_i = \sum_j w(i, j)$ 表示从节点 i 到所有其他节点的**总关联度**， \mathbf{D} 是具有对角项 d_i 的 $N \times N$ 对角矩阵， \mathbf{W} 是 $N \times N$ 矩阵，其元素为 $w(i, j)$ ， \mathbf{x} 是 $N \times 1$ 指示向量，如果节点 $i \in A$ ，则 $x_i = 1$ ，否则为 $x_i = -1$ 。Shi 与 Malik 证明：
$$\min_{\mathbf{x}} \text{Ncut}(A, B) = \min_{\mathbf{y}} \frac{\mathbf{y}^T (\mathbf{D} - \mathbf{W}) \mathbf{y}}{\mathbf{y}^T \mathbf{D} \mathbf{y}}$$
其条件 $y_i \in \{1, -b\}$ ，其中 $\mathbf{y} = (\mathbf{1} + \mathbf{x}) - b(\mathbf{1} - \mathbf{x})$ ，而 $b = \frac{\sum_{x_i > 0} d_i}{\sum_{x_i < 0} d_i}$
- 如果将 \mathbf{y} 松弛而取实数值，则可以通过求解**广义特征值问题** $(\mathbf{D} - \mathbf{W}) \mathbf{y} = \lambda \mathbf{D} \mathbf{y}$ 来实现最小化，其中 $(\mathbf{D} - \mathbf{W})$ 被称为拉普拉斯矩阵。该解由**第二最小特征向量**给出。由于解 \mathbf{y} 是实值，必须选择阈值来估计指示向量 \mathbf{x} 。一个常见的过程是在1个均匀间隔的可能的分割阈值上进行搜索以获得最小的 $\text{Ncut}(A, B)$ 。

归一化图割实验结果



主动轮廓

- **主动轮廓模型**（也称为蛇形）是通过最小化能量函数拟合到对象/分割边界的参数平面曲线。
- Amini等人 [Ami 90]使用**离散动态规划**进行优化，其数值解更稳定，并允许包含硬约束。
- 用户在期望对象的边界上指定N个节点， $\mathbf{x}_i = (x_{1i}, x_{2i})$ ， $i = 1, 2, \dots, N$ 。初始对象轮廓是通过将这些节点与直线连接获得的轮廓段的并集，其数学表达式：

$$C(s) = (s-i)\mathbf{x}_{i-1} + (s-i+1)\mathbf{x}_i, \quad i-1 \leq s \leq i$$

- 然后通过**最小化**作为三个能量项的加权和的**总能量函数**将初始轮廓**拟合**到期望的物体边界上。

$$E = \alpha_{\text{int}}E_{\text{int}} + \alpha_{\text{image}}E_{\text{image}} + \alpha_{\text{est}}E_{\text{est}}$$

其中 E_{int} 表示提高曲线的规律性（平滑度）的内力， E_{image} 是将轮廓推向高梯度边缘的图像力， E_{est} 有助于结合倾向于特定形状的期望形状的先验知识。

主动轮廓 (cont'd)

- 我们将总能量表达为**局部轮廓段上能量项 E_i 的总和**:

$$E_{\text{snake}} = \sum_{i=1}^N E_i = \alpha_{\text{int}} \sum_{i=1}^N E_{\text{int},i} + \alpha_{\text{image}} \sum_{i=1}^N E_{\text{image},i} + \alpha_{\text{ext}} \sum_{i=1}^N E_{\text{ext},i}$$

其中每个轮廓段可最多连接三个节点 $\mathbf{x}_i, \mathbf{x}_{i-1}, \mathbf{x}_{i-2}$ 。对于闭合轮廓，我们假设节点1连接到节点N；因此，N个节点有N个轮廓段。

- 通过**离散多级决策过程（动态规划）**搜索最小值，其中在每个阶段 n ，最小化：

$$E^{(n)} = \sum_{i=1}^n E_i, \quad n = 1, \dots, N$$

可以以**递归形式**执行：

$$\min_j \{E^{(i-1)}(j, k) + E_i(j, k, l)\}$$

其中 j, k 和 l 分别表示节点 $\mathbf{x}_{n-2}, \mathbf{x}_{n-1}$, 和 \mathbf{x}_n 周围的所有可能的搜索点（扰动）的索引。在最后阶段发现最小的蛇形能量，其中 $E_{\text{snake}} = E^{(N)}$ 。然后，可以通过在每个可产生最小蛇形能量节点处对搜索点回溯来计算最佳轮廓。

场景变化检测

- 什么是**场景变化**?
 - **镜头切换** (突然变化)
 - **场景的渐变**
 - 淡出/入 (图像变暗或相反)
 - 溶解 (从图像1到图像2的渐变过渡)
 - 擦除 (从图像1 到图像2 的从右向左的转换)
- 特殊领域的**特定效果**
 - 体育广播

场景变化检测 (cont'd)

- **像素差法**：如果预定数量的像素表现出大于阈值的差异，则可以声明“切换”。更鲁棒的方法可以是将每个帧划分为矩形块，独立地计算每个块的统计量，例如均值和方差，然后根据设定的阈值检查具有改变统计量的块的计数。这种方法对摄像机运动和噪声/压缩损伤比较敏感。
- **直方图法**：对于每个帧 k ，我们计算 n 个直条的颜色直方图， $h_k(i), i=1, \dots, n$ 。存在各种度量和测试方法来量化直方图的相似性/不相似性。这些包括直方图交叉测量，卡方检验，Kolmogorov-Smirnov检验。虽然这是有效的检测切换和淡化的方法，但无论强度还是直方图差异都不能有效地区分擦除和相机运动，例如镜头摇动和缩放。这些特殊效果的检测需要将直方图差异和相机运动估计进行组合的方法。
- **压缩域方法**：DC图像。

运动物体检测：帧差

- **帧差 (FD)** 图像示出了当前帧 $s_k(\mathbf{x})$ 和模型帧 $M_k(\mathbf{x})$ 之间的逐像素差，即：

$$FD_k(\mathbf{x}) = s_k(\mathbf{x}) - M_k(\mathbf{x})$$

- 假设摄像机静止，并且在 $M_k(\mathbf{x})$ 考虑照明变化，我们可以通过将**FD阈值化**来区分由于真实运动引起的还是由于噪声所引起的帧差：

$$z_k(\mathbf{x}) = \begin{cases} 1 & \text{if } |FD_k(\mathbf{x})| > T \\ 0 & \text{otherwise} \end{cases}$$

- **模型帧选择**

- **连续帧差**：模型帧是前一帧。
- **固定参考**：模型帧被指定并保持不变。
- **滤波模型帧**

$$M_k(\mathbf{x}) = \begin{cases} (1 - \alpha)s_k(\mathbf{x}) + \alpha M_{k-1}(\mathbf{x}) & k = 1, \dots \\ s_0(\mathbf{x}) & k = 0 \end{cases}$$

$0 < \alpha < 1$ 为学习率。

多分辨率帧差

- 构造一个**高斯金字塔**，每帧表示为多个分辨率。从最低分辨率水平开始处理。

– 对于当前分辨率下的每个像素，计算**归一化帧差**：

$$FDN_k(\mathbf{x}) = \frac{\sum_{\mathbf{x} \in N} |S_k(\mathbf{x}) - M_k(\mathbf{x})| |\nabla M_k(\mathbf{x})|}{\sum_{\mathbf{x} \in N} |\nabla M_k(\mathbf{x})|^2 + c}$$

其中 N 表示像素 \mathbf{x} 的局部邻域， $\nabla M_k(\mathbf{x})$ 表示像素 \mathbf{x} 处图像亮度的梯度。

c 是为避免数字不稳定而引入的常数。

- 如果**归一化帧差较大** (表明该像素为**运动像素**)，则将该像素在上一个分辨率级别的归一化帧差替换为新的值。否则，保留上个分辨率级别得到的值。
- 对所有的分辨率级别，重复上述步骤直到最高分辨率。
- 在最高分辨率级别，对归一化运动检测函数进行**阈值化**。

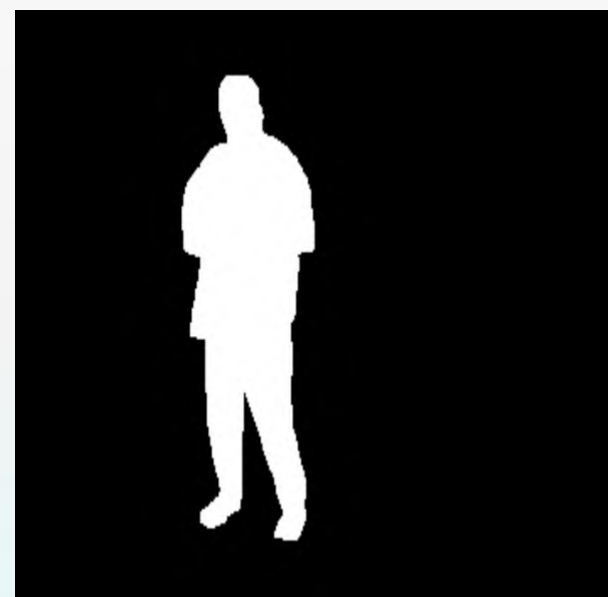
背景消除：示例



模型 (参考) 图像



当前帧



前景区域

自适应背景建模

- 对复杂背景建模的流行方法是将**每个像素**视为**时间像素过程**，使得像素强度的分布由**K维混合高斯建模**，其中K在3和5之间变化。
- **给定状态k的像素强度分布**（由于时间纹理或噪声和小的照明变化）是具有均值 μ_k 和方差 σ_k^2 的高斯分布。参数 ω_k , $k = 1, \dots, K$ 表示由状态k生成像素的先验概率，其中： $\sum_{k=1}^K \omega_k = 1$
- 每个高斯分布的**状态数K**，**均值 μ_k 和方差 σ_k^2** 以及**每个高斯的先验概率 ω_k 都是未知数**，且必须从观察到的像素数据估计出来。
- 基于自适应混合高斯分布的背景建模具有**几个优点**，包括：i) **每个像素使用不同的判决阈值**，ii) **阈值随时间变化**，iii) **多个背景模型可以共存**（以处理短暂的遮挡）。
- 需要**初始化阶段**，学习过程可能很慢。

混合高斯算法

- 对每帧 t 中的每个像素 \mathbf{x} , 我们确定生成像素 \mathbf{x} 的状态 k , 使得如果它在均值 μ_k 的 $2.5\sigma_k$ 范围内, 则可确定产生像素 \mathbf{x} 的状态为 k 。然后, 第 k 个高斯的均值和方差被更新为 [Sta 99]:

$$\begin{aligned}\mu_{k,t} &= (1-\rho)\mu_{k,t-1} + \rho s_t(\mathbf{x}) \\ \sigma_{k,t}^2 &= (1-\rho)\sigma_{k,t-1}^2 + \rho (s_t(\mathbf{x}) - \mu_{k,t})^2 \\ \rho &= \alpha N(s_t(\mathbf{x}) | \mu_{k,t}, \sigma_{k,t}^2) \\ \omega_{k,t} &= (1-\alpha)\omega_{k,t-1} + \alpha M_{k,t}\end{aligned}$$

其中 $N(s_t(\mathbf{x}) | \mu_{k,t}, \sigma_{k,t}^2)$ 表示高斯分布, 均值为 $\mu_{k,t}$ 方差为 $\sigma_{k,t}^2$, 对于匹配的高斯分布, $M_{k,t} = 1$, 否则为 0。

- 如果 $s_t(\mathbf{x})$ 的值与任何现有的高斯分布不匹配, 则用具有大方差、均值为 $s_t(\mathbf{x})$ 且低概率 $m_{k,t}$ 的新分布去替代可能性最小的分布。
- 具有最大先验概率和最小方差的高斯模型属于背景模型。因此, 高斯分布被按照 $\omega_{k,t}/\sigma_{k,t}$ 的大小从最高到最低进行排序, 第一个 B 被选为背景模型, 其中: $B = \arg \min_{i=1}^b \omega_i > T$

视频抠图：色度键控

- **色度键控**，也称为**蓝屏抠图**，是一种视频捕获技术，其中每个前景物体都在专门的工作室中单独记录，背景使用某种色彩，例如蓝色。
- 提取前景对象的问题成为**色彩分割问题**。



前景对象



组合

视频抠图: 转描机技术

- 良好的约束设置下, **背景消除**或**运动分割**能够得到**语义上**有意义的对象。
- 无约束环境中, 需要**用户交互来定义语义对象**, 因为只有用户可以知道在特定上下文中有意义的内容。
- 用户在几个关键帧中利用**交互标记感兴趣对象**的轮廓。
- 通过**自动运动跟踪**计算所有其他帧中的对象轮廓。

转描机技术是一种动画家用来逐帧追踪真实运动的动画技术, 这种技术最早是把预先拍好的电影投放到毛玻璃上, 然后动画家将其描绘下来。

