

Graph Cut: (优化能量函数使其值达到最小)

(1) 能量函数: $E(L) = aR(L) + B(L)$

$R(L)$ 为区域项, $B(L)$ 为边界项

(2) 区域项: $R(L) = \sum_{p \in P} R_p(l_p)$ 其中 $R_p(l_p)$ 表示为像素 P 分配标签 l_p 的惩罚

目标惩罚: $R_p(1) = -\ln P(l_p | 'obj')$

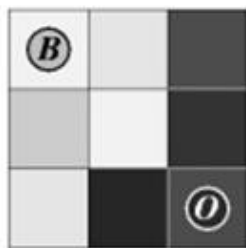
背景惩罚: $R_p(0) = -\ln P(l_p | 'bkg')$

如果全部的像素都被正确划分为目标或者背景, 那么这时候能量就是最小的。

(3) 边界项:

$$B(L) = \sum_{\{p,q\} \in N} B_{\langle p,q \rangle} \cdot \delta(l_p, l_q) \quad , \quad \delta(l_p, l_q) = \begin{cases} 0, & \text{if } l_p = l_q \\ 1, & \text{if } l_p \neq l_q \end{cases} \quad , \quad B_{\langle p,q \rangle} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right)$$

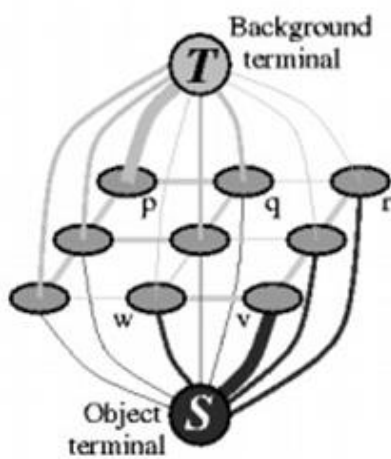
p 和 q 为邻域像素, 边界平滑项主要体现分割 L 的边界属性, $B_{\langle p,q \rangle}$ 可以解析为像素 p 和 q 之间不连续的惩罚。如果 p 和 q 越相似 (例如它们的灰度), 那么 $B_{\langle p,q \rangle}$ 越大, 如果他们非常不同, 那么 $B_{\langle p,q \rangle}$ 就接近于 0。如果两邻域像素差别很小, 那么它属于同一个目标或者同一背景的可能性就很大, 如果他们的差别很大, 那说明这两个像素很有可能处于目标和背景的边缘部分, 则被分割开的可能性比较大, 所以当两邻域像素差别越大, $B_{\langle p,q \rangle}$ 越小, 即能量越小。



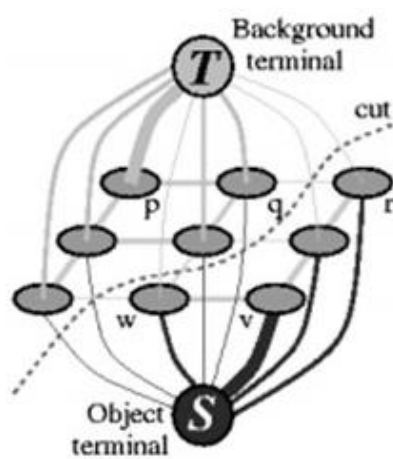
(a) Image with seeds.



(d) Segmentation results.



(b) Graph.



(c) Cut.

Grab Cut

(1) Graph Cut 的目标和背景的模型是灰度直方图，Grab Cut 取代为 RGB 三通道的混合高斯模型 GMM；

(2) Graph Cut 的能量最小化（分割）是一次达到的，而 Grab Cut 取代为一个不断进行分割估计和模型参数学习的交互迭代过程；

(3) Graph Cut 需要用户指定目标和背景的一些种子点，但是 Grab Cut 只需要提供背景区域的像素集就可以了。也就是说你只需要框选目标，那么在方框外的像素全部当成背景，这时候就可以对 GMM 进行建模和完成良好的分割了。即 Grab Cut 允许不完全的标注（incomplete labelling）。

高斯函数：

$$g(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp \left[-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right]$$

颜色模型

整个图像的 Gibbs 能量:

$$E(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}),$$

(1) U 就是区域项, 能量项的确定

$$U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n),$$

$$D(\alpha_n, k_n, \underline{\theta}, z_n) = -\log \pi(\alpha_n, k_n) + \frac{1}{2} \log \det \Sigma(\alpha_n, k_n) \\ + \frac{1}{2} [z_n - \mu(\alpha_n, k_n)]^T \Sigma(\alpha_n, k_n)^{-1} [z_n - \mu(\alpha_n, k_n)].$$

GMM 的参数 θ 就有三个: 每一个高斯分量的权重 π 、每个高斯分量的均值向量 μ (因为有 RGB 三个通道, 故为三个元素向量) 和协方差矩阵 Σ (因为有 RGB 三个通道, 故为 3x3 矩阵):

$$\underline{\theta} = \{\pi(\alpha, k), \mu(\alpha, k), \Sigma(\alpha, k), \alpha = 0, 1, k = 1 \dots K\},$$

描述目标的 GMM 和描述背景的 GMM 的这三个参数都需要学习确定

(2) 边界能量项:

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n) \in C} [\alpha_n \neq \alpha_m] \exp -\beta \|z_m - z_n\|^2.$$

本身有差别的像素 m 和 n, 它们的差 $\|z_m - z_n\|$ 还是比较低, 那么我们需要乘以一个比较大的 β 来放大这种差别, 而对于对比度高的图像, 那么也许本身属于同一目标的像素 m 和 n 的差 $\|z_m - z_n\|$ 还是比较高, 那么我们就需要乘以一个比较小的 β 来缩小这种差别。

迭代最小化

(1) 对每个像素分配 GMM 中的高斯分量 (例如像素 n 是目标像素, 那么把像素 n 的 RGB 值代入目标 GMM 中的每一个高斯分量中, 概率最大的那个就是最有可能生成 n 的, 也即像素 n 的第 k_n 个高斯分量):

$$k_n := \arg \min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$

(2) 对于给定的图像数据 \mathbf{z} ，学习优化 GMM 的参数（因为在步骤（1）中我们已经为每个像素归为哪个高斯分量做了归类，那么每个高斯模型就具有了一些像素样本集，这时候它的参数均值和协方差就可以通过这些像素样本的 RGB 值估计得到，而该高斯分量的权值可以通过属于该高斯分量的像素个数与总的像素个数的比值来确定。）：

$$\underline{\theta} := \arg \min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$

(3) 分割估计（通过 1 中分析的 Gibbs 能量项，建立一个图，并求出权值 t-link 和 n-link，然后通过 max flow/min cut 算法来进行分割）：

$$\min_{\{\alpha_n: n \in T_U\}} \min_{\mathbf{k}} E(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$

(4) 重复步骤（1）到（3），直到收敛。经过（3）的分割后，每个像素属于目标 GMM 还是背景 GMM 就变了，所以每个像素的 k_n 就变了，故 GMM 也变了，所以每次的迭代会交互地优化 GMM 模型和分割结果。另外，因为步骤（1）到（3）的过程都是能量递减的过程，所以可以保证迭代过程会收敛。

(5) 采用 border matting 对分割的边界进行平滑等等后期处理。

Snake 模型

一条可变形的参数曲线及相应的能量函数，以最小化能量目标函数为目标，控制参数曲线变形，具有最小能量的闭合曲线就是目标轮廓。

Snakes 模型的轮廓线承载了上层知识，而轮廓线与图像的匹配又融合了底层特征。这两项分别表示为 Snakes 模型中能量函数的内部力和图像力。

$$E_{total} = \int_s \left(\alpha \left| \frac{\partial}{\partial s} \bar{v} \right|^2 + \beta \left| \frac{\partial^2}{\partial s^2} \bar{v} \right|^2 + E_{ext}(\bar{v}(s)) \right) ds$$

其中第 1 项称为弹性能量是 v 的一阶导数的模，第 2 项称为弯曲能量，是 v 的二阶导数的模，第 3 项是外部能量（外部力），在基本 Snakes 模型中一般只取控制点或连线所在位置的图像局部特征例如梯度：图像力

$$E_{ext}(\bar{v}(s)) = P(\bar{v}(s)) = -|\nabla I(v)|^2$$

弹性能量和弯曲能量合称内部能量（内部力），用于控制轮廓线的弹性形变，起到保持轮廓连续性和平滑性的作用。而第三项代表外部能量，也被称为图像能量，表示变形曲线与图像局部特征吻合的情况。

最终对图像的分割转化为求解能量函数 $E_{total}(v)$ 极小化（最小化轮廓的能量）。在能量函数极小化过程中，弹性能量迅速把轮廓线压缩成一个光滑的圆，弯曲能量驱使轮廓线成为光滑曲线或直线，而图像力则使轮廓线向图像的高梯度位置靠拢。基本 Snakes 模型就是在这 3 个力的联合作用下工作的。