Grab cut

Graph Cut

每个像素，都分配一个标签 $L \in \{0, 1\}$，代表 fore/background

定义能量函数，评价分割效果 $E(L) = \lambda R(L) + B(L)$

R Region　B Boundary

obj　bgd

每个像素都视作一个节点，加上源和宿 S 和 T

节点间的边 n-link，和 S,T 的边为 t-link

权值分配：

$\langle p, q \rangle$　$B_{\langle p,q \rangle}$

$B_{\langle p,q \rangle} \propto exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \frac{1}{dist(p,q)}$　p, q 差越小，权越大

越像 bgd，边权越小

$\langle p, S \rangle$ { $\lambda R_p(bgd)$ 不确定；K p为前景；0 p为背景

$\langle p, T \rangle$ { $\lambda R_p(obj)$ 不确定；0 p为前景；K 背景

$R_p(obj) = -\log p(I_p | O)$ aka：越像 obj，$\langle p, T \rangle$ 边权越大

建立直方图，归一化为概率密度，就得出概率

直方图由种子点获得

而 $K = 1 + \max_p \sum_{邻域} B_{\langle p,q \rangle}$，相当于整个图像上与周围像素差最小的一个极大的边权

几乎不会被分割

最小割，先割的是权小的边

可以证明，按如上方法分配权重等价于能量最小

在添加新 obj 点时，对边权重修改的小 trick

如果直接按上文的方法改，就会造成网络流量的变化

边权加一个常数，不影响结果，还能提高算

Grab cut

$E(\alpha, k, \theta, z) = U(\alpha, k, \theta, z) + V(\alpha, z)$

对 fore 和 back 都建立 GMM，(K=5)

对于每个 pixel，有 $\alpha \in \{0, 1\}$（0: back, 1: fore），$k \in \{1, \dots K\}$（属于哪一 Gauss分量）

对于每个 GMM，都有 π（各分量权重），μ，Σ（covar）→ 统一为参数 θ

μ：3元素；Σ：3×3矩阵（RGB）

两个GMM直接按划定好的区域，学习参数，就可以得出概率了，作为 t-link权值

矩形范围内都划为 PR_FG，外面都是 BG

PR_FG、FG → fore GMM

PR_BG、BG → back GMM

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

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

对 nlink，像素差越大，越应被分到，能量越小

在RGB中，用欧氏距离衡量像素差。

为了适应低对比度图像，乘上 β 来放大差异高，同理

$$V(\alpha, z) = \gamma \sum (\alpha_n \neq \alpha_m) \cdot \exp(-\beta \|z_n - z_m\|^2)$$

γ 为 empirical 值，=50

迭代最小

$z \to k$

① 根据初始化的GMM，预测每个 pixel 属于哪个类（概率最大）

② GMM根据图像和①学习 θ　$z, k \to \theta$

③ 分割估计　$z, \theta \to C$

多次迭代后，每个pixel的归属变了，k_n 也变了，GMM也变了，

所以每次迭代都会优化 Gmm

Border Matting 用于后期平滑。

GMM初始化

用 kMeans 对输入像素聚类 (k=5)

然后 进行一波拟合（输入 → 聚类标签）

更新 π, μ, Σ

时间复杂度——怎么算

KMeans 不跟，均值，最近，迭代，收敛

为什么不用直方图而用GMM？

① 为了处理彩色，彩色直方图的构成空间本来就很稀疏

② 单纯的直方图只包含了像素亚度信息

采用GMM可以浓缩数据

为什么只要计算4个方向？

不会重复，每个点只需处理 180° 的邻边即可

问题——α哪去了？　本身就是二值化的，可归纳为是否和S,T点相连

为什么不用迭代？

KMeans，初始化 GMM参数
GMM预测出每个pixel的标签
再拿pixel和标签去训练GMM

将init 和训练后的比较

KMeans为GMM提供3良好的初始化，而不是EM (costly)
如果不初始化也能跑，但区域块R就没用了，对fore / back的判断不太可靠