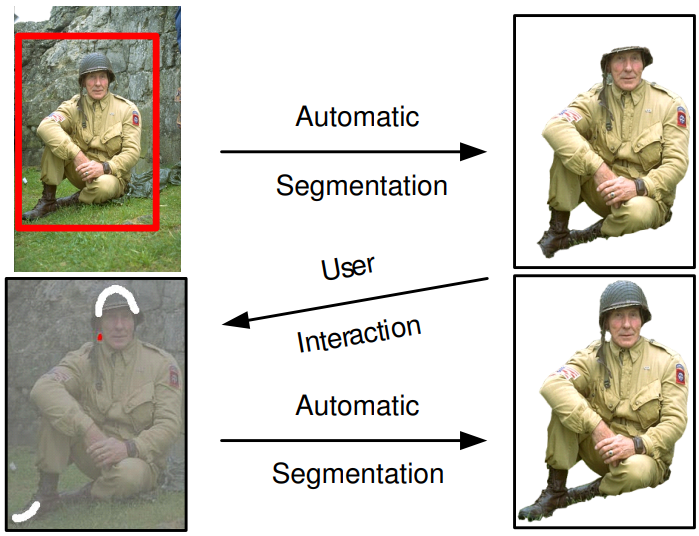
实验二：GrabCut静态图像分割方法

一、实验介绍

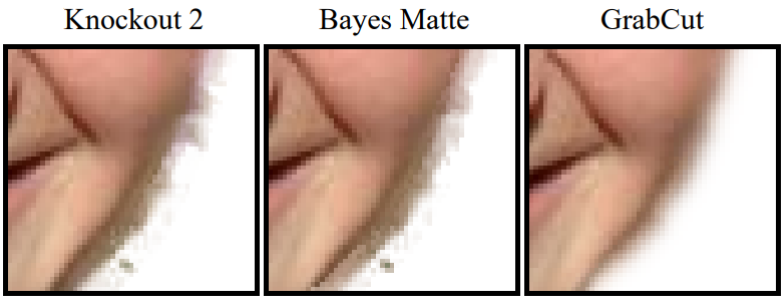
1. GrabCut利用迭代的graph cut进行交互式前景提取。能够实现对静态图像高效的、交互的前景背景分割，对于图像编辑有非常重要的现实意义。下图所示为GrabCut实现的抠图效果。用户只需要在物体外围拉一个包住目标物体的矩形框，就可以实现较好的分割：

2. 当前景和背景颜色比较相似的时候，相似的部分可能难以分割出来。这时候可以利用前/背景画刷进行交互，达到更理想的抠图效果。



3. GrabCut的Border Matting功能可使分割边缘自然而平滑，下图是和其他两种Matting算法效

果的对比：



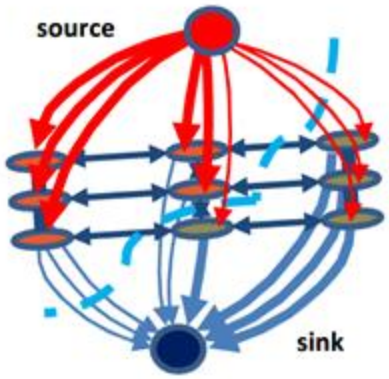
二、算法原理

(1) 图像分割（graph cut）

一幅灰度图像由一组灰度值来表示，记为 $z=\{z_1,z_2,...,z_N\}$ 。分割任务可以看作为每个像素点预测一个 α 值，0代表背景，1代表前景。一幅图像的前景背景灰度分布（灰度直方图）可以由参数 $\theta=\{h(z;\alpha),\alpha=0,1\}$ 来描述，h表示灰度直方图。分割任务就是给定模型 θ 和图像数据 z ，预测 α 。

(2) 基于能量最小准则进行分割

定义一个能量函数，其最小值对应理想的分割。直观来说，该能量函数应该受前景背景灰度直方图以及 α 值的分布影响。其数学表达形式为 $E(\alpha,\theta,z)=U(\alpha,\theta,z)+V(\alpha,z)$ 。



其中，数据项U用来评估给定参数 θ ， α 分布对数据z的拟合程度，定义为 $U(\alpha,\theta,z)=\sum_n -logh(z_n;\alpha_n)$

平滑项写为， $V(\alpha,z)=\gamma\sum_{(m,n)\in C} dis(m,n)^{-1}[\alpha_n\neq\alpha_m]exp\{-\beta(z_m-z_n)^2\}$ 。C表示相邻像素的集合。在实际应用中，定义水平、垂直、斜角相邻的像素（8连通方式）能够获得好的结果。

$$\beta = \left(2\left\langle (z_m - z_n)^2 \right\rangle\right)^{-1},$$

where $\langle \cdot \rangle$ denotes expectation over an image sample.

现在就得到了整个能量模型。分割任务可以被估计为一个全局最小化问题：$\hat{\alpha} = argmin_{\alpha} E(\alpha, \theta)$ 。接下来，运用标准的最小化cut算法获取最小值，就可以实现分割前景背景的目的。

一张二维图像可以看做一个MRF随机场，在确定每个像素（随机场中的顶点）Label值以后，对其分割可以直接调用 mincut（从图论的角度验证等价于maxflow）,网上有现成的库可以调用： http://vision.csd.uwo.ca/code/ 中 Max-flow/min-cut 或者 Planar Graph Cut。Python 相关库参考： http://pmneila.github.io/PyMaxflow/tutorial.html#getting-started。

(3) 从graph cut到GrabCut

首先，通过用高斯混合模型GMM (Gaussian Mixture Model) 代替灰度直方图，单色图像模型被颜色模型取代；关于高斯混合模型，可以参考：

维基百科：https://en.wikipedia.org/wiki/Mixture_model#Gaussian_mixture_model，漫谈Gaussian Mixture Model：https://blog.pluskid.org/?p=39，斯坦福公开课：http://open.163.com/newview/movie/free?pid=M6SGF6VB4&mid=M6SGKK6L3。

其次，只进行一次的最小化分割估计算法被一个在估计和参数学习中可选的、可靠迭代的过程代替；

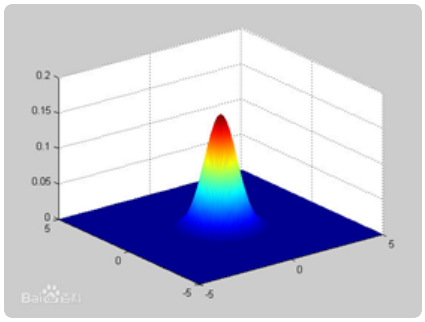
第三，对于交互用户不需要提供完整的标记——对于用户只需指定背景，这可以通过在对象周围放置一个矩形框实现。

(4) 高斯混合模型（GMM）

混合模型：一个可以用来表示在总体分布中含有K个子分布的概率模型。

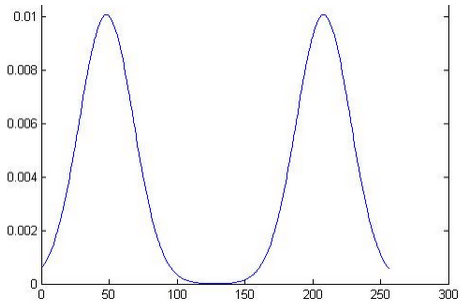
单高斯模型：对于多维样本数据X，高斯分布的概率密度函数为：

$P(x|\theta) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\{-(x-\mu)^T \Sigma^{-1}(x-\mu)/2\}$ ，其中， μ 为数据均值，Σ 为协方差，D为数据维度。二维高斯分布的图像为，



高斯混合模型：可以看作是由K个单高斯模型组合而成的模型，其概率分布为 $P(x|\theta)=\sum_{k=1}^{K} w_k\phi(x|\theta_k)$ 。

其中，参数 $\theta_k=(\mu_k,\sigma_k,w_k)$ ，是每个子模型的期望、方差，在混合模型中发生的概率， ϕ 表示子模型的概率密度函数。一个混合模型可以使用任何概率分布拟合，使用高斯分布是因为高斯分布具备很好的数学性质以及良好的计算性能。如下图所示，样本数据无法用单一高斯模型来较好的描述，这种情况下混合高斯分布就是很好的选择。



混合高斯模型的求解方法：混合高斯模型的似然函数为 $logL(\theta)=\sum_{j=1}^{N} logP(x_j|\theta)=\sum_{j=1}^{N} log(\sum_{k=1}^{K} w_k\phi(x_j|\theta))$ ，无法使用最大似然法求出最优参数，因为对每个观测数据点来说，事先不知道它是属于哪个子分布的。直接求导无法计算，需要通过迭代的方法求解。

(5) EM算法

EM算法是一种迭代算法，用于含有隐变量的概率模型参数的最大似然估计。

每次迭代包含两个步骤：

(1) E-step：求期望 $E(\gamma_{jk}|X,\theta)$ 对所有的 $j=1,2,...,N$ 。

$\gamma_{jk}=\frac{w_k\phi(x_j|\theta_k)}{\sum_{k=1}^{K} w_k\phi(x_j|\theta_k)}$ ，表示第个数据来自子模型的可能性。

(2) M-step：求极大，计算新一轮迭代的模型参数。

$$\mu_k = \frac{\sum_j^N (\gamma_{jk} x_j)}{\sum_j^N \gamma_{jk}}, k = 1, 2, \ldots, K$$

$$\Sigma_k = \frac{\sum_j^N \gamma_{jk}(x_j - \mu_k)(x_j - \mu_k)^T}{\sum_j^N \gamma_{jk}}, k = 1, 2, \ldots, K \quad \text{（用这一轮更新后的 } \mu_k \text{ ）}$$

$$\alpha_k = \frac{\sum_{j=1}^N \gamma_{jk}}{N}, k = 1, 2, \ldots, K$$

（3） 重复1、2，直至收敛。

至此，我们就找到了高斯混合模型的参数。需要注意的是，EM 算法具备收敛性，但并不保证找到全局最大值，有可能找到局部最大值。解决方法是初始化几次不同的参数进行迭代，取结果最好的那次。

（6）GMM参数初始化

高斯混合模型参数初始化的方法是，首先对数据用kmeans进行聚类，聚类的数量就是高斯分布的数量，然后对聚类后的每个cluster计算均值和方差作为高斯分布参数的初始估计值。

kmeans算法：python相关参考：https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html，Matlab相关参考：https://ww2.mathworks.cn/help/stats/kmeans.html。

三、实验流程

（1）用户通过选择矩形框选择初始trimap（即rgb子图像）。矩形外部像素标记为背景，内部像素标记为未知。

（2）计算机创建初始分割，未知像素归为前景类，背景像素归为背景类。

（3）为初始前景和背景创建GMM（kmeans聚类，对每一类计算高斯分布的均值和方差）。

（4）前景类中的每个像素被分配给前景GMM中最可能的高斯分量，背景类进行相同操作。

（5）根据上一步的分配的像素集更新GMM（抛弃原来GMM,创建新的GMM）。

（6）创建图，执行min cut 算法产生新的像素分类。

（7）重复执行4-6步直到收敛。

四、实验要求

1） 本次实验不限编程语言，可以使用Python， MATLAB 等语言（推荐使用MATLAB）。

2） 实验禁止直接调用 grabcut 函数。

3） 实现grabcut算法。

4） 请将以下文件打包发送至 algorithm_2022@126.com

a) 实验代码

b) 实验报告

c) 实验结果

文件名如： 张三 _ 第五次实验 .zip

邮件主题： 张三， PB00000000 ， 第五次实验

五、参考文献

[1] ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. Grabcut– interactive foreground extraction using iterated graph cut. In Proc. of ACM SIGGRAPH, 2004,309—314.

[2] BOYKOV, Y., AND JOLLY, M.–P. Interactive graph cuts for optimal boundary and region segmentation of objects in n–d images. In ICCV, 2001, 105—112.