

各位老师好，非常感谢老师们在百忙之中抽出时间来听我汇报GrabCut的论文复现工作。GrabCut是图像分割领域一个具有奠基性意义的工作，他向前继承了GraphCut文章的思想，并且成为后续图像分割领域的论文发展和创新的基石。而所谓图像分割指的就是根据灰度、颜色、纹理和形状等特征把图像划分成若干互不交迭的区域，并使这些特征在同一区域内呈现出相似性，而在不同区域间呈现出明显的差异性。

在grabcut前主要的图像分割方法有：

1.基于阈值的分割方法：基于图像的灰度特征来计算一个或多个灰度阈值，并将图像中每个像素的灰度值与阈值相比较，最后将像素根据比较结果分到合适的类别中。因此，该类方法最为关键的一步就是按照某个准则函数来求解最佳灰度阈值。在上图中magicWand采用的是方法是用户首先标记出前景的部分区域，然后通过计算该区域周围的像素与这个区域的相似程度来计算出剩余的前景区域，这种方法并不能完整的分割出物体，同时也没有很好的考虑到前景物体的边界。

2.基于边缘的分割方法：所谓边缘是指图像中两个不同区域的边界线上连续的像素点的集合，是图像局部特征不连续性的反映，体现了灰度、颜色、纹理等图像特性的突变。通常情况下，基于边缘的分割方法指的是基于灰度值的边缘检测，它是建立在边缘灰度值会呈现出阶跃型或屋顶型变化这一观测基础上的方法。

3.基于图论的分割方法：此类方法把图像分割问题与图的最小割（min cut）问题相关联。首先将图像映射为带权无向图 $G=\langle V, E \rangle$ ，图中每个节点 $N \in V$ 对应于图像中的每个像素，每条边 $\in E$ 连接着一对相邻的像素，边的权值表示了相邻像素之间在灰度、颜色或纹理方面的非负相似度。而对图像的一个分割 s 就是对图的一个剪切，被分割的每个区域 $C \in S$ 对应着图中的一个子图。而分割的最优原则就是使划分后的子图在内部保持相似度最大，而子图之间的相似度保持最小。

GraphCut和GrabCut就是基于图论的图像分割方法。

下面我将详细介绍一下GraphCut是如何通过基于图论和最小割算法的方法来对图像进行分割的。首先它对分割设计了一个总体的损失函数 $E(A)$ ，其中 A 表示一种可能的对于图像的分割， $E(A)$ 表示当前分割的代价。 $E(A)$ 由两部分组成， $R(A)$ 表示区域项region， $B(A)$ 表示边界项boundary， λ 是一个常数，表示区域项和边界项的权重比例，越大的 λ 则代表区域项所占的比重越大。其中 $R(A)$ 的物理意义表示为将像素 p 分配到前景或后景后，前景和后景对应的概率模型的损失和，他反应了每个像素在前后景模型的一个适应程度。而 $B(A)$ 的物理意义则是将前后景分割出来后其边界连续性的损失和，他表示在分割的边界出临界像素与其周围像素的连贯程度。整个graphcut的核心思想就是利用图论的方法去最小化我们的损失函数 $E(A)$ ，使得分割后的像素集合更好的符合前后景概率分布模型，同时也保持同一区域内图像像素的连贯性

介绍完了损失函数后一个核心的问题就是如何通过图论的方法构建无向图来表达我们的损失函数，以及为什么这种构建方法能够合理的表达我们的损失函数。

首先我来介绍如何通过构建无向图来表达我们的损失函数，图像上的每一个像素点会对应图中的每一个节点，除此之外图还包括两个额外的节点，一个是源点S，和汇点T，对应的就是右图中的Object terminal和Background terminal。

首先我们知道损失函数由两部分组成，边界项和区域项。这两个项对应到图中的边就分别是nlink和tlink，临端链接(neighbor link)和终端链接(terminal link)，而对于图中的每一个节点，他都有两个tlink，分别连向源点S和汇点T，和若干个nlink，nlink的数量取决于在它周围节点的个数，最多的情况一个节点有8个nlink，上下左右，左上、左下、右上、右下共8个。

知道了每个节点边的组成，下面我们来讲一讲如何设计边的权值。对于nlink，由于我们希望s-t割能够割开前景和后景的交界区域，而这个交界区域相邻像素一般有着比较大的欧式距离，因此nlink的值等于与它相邻节点的欧式距离的一个函数，也就是我们的损失函数中的边界项。

而对于节点的tlink，如果我们已经能够事先知道它属于背景或者前景，举个例子，一个像素我们已经事先它属于背景区域，那么它连接到汇点T的tlink的权值就会特别大，论文中将这个值设置为一个定值K，而连接到源点S的tlink的权值就会等于0，这样的设计能够使得我们的s-t割不会把一个已知是背景的点划分到前景，因为切到它连接汇点T的边的代价特别大。而对于我们事先并不知道这个点所属的区域，那么它们连接到源点S和汇点T的边的权值就等于其在对应的前后景概率分布模型的概率密度，也就是我们的损失函数的区域项。

最后我们来介绍一下tlink中K值是如何设计的，以及为什么要这样设计。常值K等于图中每一个节点的nlink的和的最大值，为什么要这样设计K呢，假设我们已经事先知道某一个节点属于背景，考虑一个最糟糕的情况，他周围的点与它的相似程度非常高但并不属于背景，这样的化如果把他从周围的点集切割出来，我们就需要将这个点所有的n-link都切割掉，这样所增加的损失就会特别大，但是如果我们通过将这个划分到背景后所减少的tlink损失始终会大于增加的nlink损失的话，这样我们的s-t就能正常的划分出那先已知前后景信息的节点。

介绍完了如何通过图像和损失函数的定义和参数构建图，下面我们从宏观的角度去解释一下为什么这种构建方法能够使得s-t割正确的代表我们之前所定义的损失函数。以上图的s-t割为例子，我们可以直观的得出这样的结论，一个s-t割对于所有节点来说，只会切割每个节点的两条tlink中的一条，而这些被所有被切割的tlink的总和就构成了我们的 $R(A)$ ，也就是损失函数的区域项。而对于n-link，一个s-t割只会切割恰巧位于边界两侧相邻节点的n-link，而它们的总和就构成了我们的 $B(A)$ ，也就是损失函数的边界项。因此我们通过最小割算法求取无数种s-t割中一个代价最小的s-t割也就等价于我们遍历所有可能的前后景划分方式A从而得到一个使得损失函数最小的划分方式，两个描述本质上是一样的。

介绍完如何通过图论的方法进行图像分割，下面我来讲解一下GraphCut的一个大致工作。GraphCut它主要用于单色图的物体分割问题，所谓的单色图就是一种只有若干种灰度颜色的图像。在GraphCut中，作者将 $R(A)$ 设计为直方图概率分布函数，也就是利用直方图模型来表示前后景的像素概率分布模型。而 $B(A)$ 即是相邻像素的一个与欧式距离有关的函数。设计完损失函数之后，GraphCut要求用户首先把部分的前景和背景图像手动标记出来，用以构建前后景的直方图模型，在通过构建图和最小割算法划分出未知区域的前背景。

相较于GraphCut单色图分割, GrabCut处理的则是彩色图前后景分割问题。不同与Graphcut的直方图模型, grabcut利用GMM (混合高斯模型) 来描述前后景的概率分布模型, 并且使用了迭代优化的方法来获取整体上最佳的分割。由于使用了迭代优化的方式, grabcut即使没有获得一个准确的前后景像素分布模型, 他也能通过逐步优化的方式去接近真实的前后景分布模型。因此, grabcut只需要用户通过一个矩形框大致的框出前景物体的范围即可, 这大大的提高了算法的用户友善程度。

下面我们具体介绍一下GrabCut中的损失函数是如何定义的。在第一个公式中， E 代表损失函数， U 和 V 分别代表损失函数的区域项和边界项。 α 表示目前的图像中像素标签，0表示背景，1表示前景， k 是像素在混合高斯模型中所属的多元高斯分布的下标， θ 代表混合高斯模型的参数，由均值和协方差组成， z 代表像素值。从第二个公式我们可以知道区域项等于图像中所有像素点在其对应混合高斯分布模型中的概率密度和。第三个公式具体给出了如何计算像素点的概率密度，它的值等于混合高斯分布的加权概率和取对数再取反，一个像素它属于某个混合高斯模型的概率越大，它对应的 D 值越小。在第四个公式中， C 代表划分前后景的边界两边的相邻像素的集合，边界项就等于这些边界上所有相邻点的欧式距离的平方乘以 β 取反再取对数再乘以 γ 。其中 γ 用来决定是边界项更重要还是区域项更重要。 β 是一个定值，它等于图像中所有相邻点的欧式距离的期望的倒数，这样的设计可以使得在分割算法在高对比度图像的边界和低对比度图像的边界都有着差不多的效果。从公式四中我们可以直观的看出，如果两个相邻像素的像素值完全相等，它们的 V 值就为最大，等于 γ 。

下面我们介绍Grabcut算法的初始化流程和迭代流程。初始化主要分为三个部分，首先用户手动的把前景的大致区域标记出来。其中trimapT的含义是，将一个图像划分为三个区域，确定的背景区域TB，确定的前景区域TF和不确定的区域TU，用户的矩形框外表示确定的背景区域，矩形框内表示不确定的区域。

对于TB区域内的像素，将标签置为0，而对于TU区域内的像素，将标签置为1。然后通过属于0，1标签的内的所有对应像素来初始化前后景混合高斯模型。

在迭代的过程中我们首先更新TU区域内每个像素的所属的高斯分布的 k 值，然后根据图像像素信息计算混合高斯分布的参数（均值和协方差），再接着第三步我们根据已知的混合高斯模型的参数和定义好的损失函数，构建图并且使用最小割算法获得最佳的 s - t 割来进行新的前后景划分，并且更新每个像素的标签。接下来回到第一步，通过新的前后景分割来计算新的 k 值和新的混合高斯模型参数和新的最小割，直到收敛，也即损失函数的下降趋势减缓或者停滞。

左边的是原图，中间的是最终的前背景混合高斯模型的分布情况，相同的颜色代表一个具有相同 k 值的像素点的集合。右边的是最终的分割结果。

这是一个分割的迭代过程中每一次迭代的效果图。其中右图表示图像中每个像素的标签，深绿色为确认的背景区域，浅绿色为生成的背景区域，红色为确认的背景区域。

这是我对Grabcut算法做的一些代码上的优化，来达到比较好的一个计算时间。主要包括两个方面，一个是参数复用，另一个是矩阵乘法优化

下面我们讨论一下如果将GrabCut中的混合高斯模型替换为快速颜色直方图会怎么样。首先我需要解释一下什么是快速颜色直方图，如果我们直接利用彩色图像中RGB的三个通道的所有组合来构建直方图，他会有256的三次方多种结果，SaliencyCut论文中提出了一种方法，通过降采样将256种通道强度中接近的值合并，从而降低成12种，并且再次剔除自然界中不常见的颜色情况，最终降低到83中颜色组合。中间的图即降采样后获得的图像，可以看出他还是比较完整的保留了图像的特征信息。

首先我们可以得出的结论是更换模型并不会对损失函数的边界项造成影响，因此grabcut最终划分的边界还是能够达到比较平滑的效果。对于区域项，可以确认的是如果我们已知正确的前后景概率分布模型的参数，那最终我们一定能正确的划分出前后景。如果用了直方图对前后景进行建模，由于直方图是一种无参估计，它会相比与GMM会需要更多的数据才能接近真实概率分布，而且对于错误的数据的容忍度更低。同时直方图的鲁棒性也更差，因为对于未知区域的一个像素点，如果它的像素值接近但是并不与背景中的像素值完全相等，它在直方图模型中的出现概率就为0，完全不会被认为是背景。而高斯分布对于接近分布均值的像素仍会给出较大的概率密度，而后者较大的值才是我们需要的正确的概率。

GrabCut论文发布后，由于其在图像分割问题中达到的显著的成效，后续有陆陆续续发布很多基于GrabCut的优化和创新的新的分割算法，比如LazySnapping, Saliency Cuts。其优化思路主要划分为三个方向，其中一个是基于改进损失函数的优化，另一种则是基于改进前后景模型的优化，还有就是基于改进算法运行复杂度和降低用户交互程度的优化。其中lazysnapping使用超点superpixel的概念，通过超点分割算法将分辨率较大的图像中 $m \times n$ 个像素点划分为若干个小区域，也即我们的superpixel，用这些超点来构建图并运行图分割算法，可以大大提高运行速率。而SaliencyCut论文中则提出了一个新的前后景模型Rcmap (Region Contrast map)，这种map的生成基于我们之前所说的快速颜色直方图。Saliencycut首先将图像分割为若干个区域，然后对于每个区域都计算它的颜色直方图。再对于每个区域，通过计算它与其它所有区域的直方图的差异性的加权和来代表当前区域的显著性程度，同时这个加权和还考虑到了空间距离这个因素，对于距离越远的区域，他的权重越小。最后再通过对Rcmap使用grapcut进行分割来获取最优的效果。