

交互式图像分割与实现

张磊

(清华大学土木工程系, 土博 14, 2014310053)

摘要: 图像分割是从输入图像中提取目标或感兴趣区域的过程, 是目标检测和识别过程中的重要步骤。由于目前的计算机人工智能还远未达到理解高层语义的水平, 全自动分割算法很难获得令人满意的结果。交互式图像分割相对于传统图像分割的最大的优点在于, 加入了用户主观的指导, 使得过去许多因为数据歧义而很难解决的问题变得定义清晰。因此, 人机交互式图像处理于近年兴起, 并得到越来越多的重视。本文研究实现的是一种针对彩色图像的交互式通用目标分割系统。首先, 自行设计图割算法中的区域惩罚项和边界惩罚项函数, 并在此基础上增加 k 均值聚类预处理以提升分割速度。然后, 采用基于特殊聚类预处理的区域增长算法对分割后的图像进行二次交互分割, 提高分割结果的准确率。其次, 用形态学开闭运算对图像分割结果进行二值滤波, 进一步提高分割准确率。最后在图像分割结束后, 系统还可以实现以下功能: 为目标更换背景、平移目标位置、对图像进行平滑处理。

关键词: 图像分割; 交互式; 图割; 区域生长

Application and Implementation of the Interactive Image Segmentation System

Zhang Lei

(Department of Civil Engineering, Tsinghua University, The first grade of Dr.in 2014, 2014310053)

Abstract: Segmenting foreground objects from natural images are a fundamental task in image understanding. But it is a challenging problem. A fully automatic algorithm cannot yet achieve satisfactory results. The difficulties lie in the high complexity in visual pattern modeling and the intrinsic ambiguity in visual pattern grouping. To make the problems solvable, one method is to construct interactive frameworks, and allow the user to specify the foreground objects and the background. The biggest advantage of interactive image segmentation, compared to tradition segmentation methods, is that with the guidance of the user, problems which previously were ambiguous in a semantic sense now become clearly defined and solvable. Therefore, interactive image processing was developed in recent years, and has been receiving more and more attention. In this paper, a kind of interactive general target segmentation system for color images is created. Firstly, we reset the region and boundary penalty functions in graph cut algorithm, and preprocess the data by using k -means clustering algorithm to enhance the segmentation speed. Sometimes, the initial segmentation result is not satisfactory. So the region growing algorithm based on a special cluster method is carried out in multiple interactions. In order to improve the accuracy of the segmentation results better, the segmentation results are processed by a morphological filter based on the open and close operations. Finally, after the success in image segmentation, the system can also implement the following functions: replacing the image background, changing the target location and smoothing the new image.

Key words: image segmentation; interactivity; Graph cut; region growing

1 引言

随着互联网技术的迅猛发展，人们早已走入信息社会，每时每刻产生的海量信息使人们渴望利用计算机来分析处理它们。其中大部分信息是图像图形信息，它相对于其他类型的信息来说具有信息量大、直观、形象、易于理解等优势。因此，数字图像分析被越来越多的学者所重视，正逐渐成为网络信息时代一个新的研究热点。而图像分析的前提是图像分割——特征提取、目标识别等的好坏都取决于图像分割的质量如何，所以说图像分割是图像分析、模式识别和计算机视觉中一项关键而基础的工作。人的眼睛在一幅复杂图像的任一点上只能识别几十种灰度级，但可以识别成千上万种颜色，所以许多情况下，单纯利用灰度信息无法从背景中提取出目标，还必须借助于色彩信息。因此彩色图像处理正受到人们越来越多的关注^[1]。

图像分割就是将目标对象从背景中分离出来，针对分割算法的研究已有几十年的历史，一直以来都受到人们的高度重视，而且是一个经典难题，发展至今仍然没有找到一个通用的解决方法。由于日常图像内容的复杂性，全自动的分割算法很难获得令人满意的结果。相比之下，采用交互式、半自动的分割方法，通过用户有限的交互行为，可以获取尽可能多的分割信息从而快速、准确地分割出目标对象。因此，人机交互式图像分割于近年兴起，并得到越来越多的重视。交互式图像处理最大的优点在于，由于加入了用户主观的指导，使得过去很多因为数据歧义而很难解决的问题变得定义清晰，再辅以具有针对性的算法，我们能够准确地解决在以前耗时长且所得结果粗糙的问题。综上所述，在新的时代背景下研究彩色图像的交互式分割算法具有重要意义。

2 图像分割的概念

图像分割就是依据图像的灰度、颜色、纹理和边缘等特征，把图像分成各自满足某种相似性准则或具有某种同质特征的连通区域的集合的过程，借助集合概念对图像分割比较严格的定义可描述如下：令集合 R 代表整个图像区域，对 R 的图像分割可以看作是将 R 分成 n 个满足下列条件的非空子集 R_1, R_2, \dots, R_n ^[2]。

1. $\bigcup_{i=1}^n R_i = R$ 。即分割成的所有子区域的并集能构成原来的区域 R
2. 对于所有的 i 和 j ， $i \neq j$ ，有 $R_i \cap R_j = \emptyset$ 。即分割成的各子区域互不重叠。
3. 对于 $i=1, 2, \dots, n$ ，有 $P(R_i) = \text{TRUE}$ 。即分割得到的属于同一区域的像素应具有某些相同的特性。
4. 对于 $i \neq j$ ，有 $P(R_i \cup R_j) = \text{FALSE}$ 。即分割得到的属于不同区域的像素应具有不同的特性。
5. 对于 $i=1, 2, \dots, n$ ， R_i 是连通的区域。即同一子区域内的像素应当是连通的。

其中， $P(R_i)$ 是定义在区域 R_i 内所有的点上的逻辑谓词， \emptyset 表示空集。上述五个条件依次表明了图像分割的完备性、独立性、单一性、互斥性和连通性。本文所指的图像分割是指将原图像分成两个集合：其中用户感兴趣的区域称为“目标”并用字母 O 表示，剩下的部分称为“背景”用字母 B 表示。

3 基于图割的分割算法

3.1 图割算法

1) 图像分割是一个像素标定的过程。假定图像 I 中所有像素组成的集合为 P , 其中任意一个像素 p ($p \in P$) 的类别归属用 A_p 来表示, A_p 只能标定为“obj” (“object”的缩写, 代表目标像素) 或“bkg” (“background”的缩写, 代表背景像素)。则图像的分割就可以用下面的向量 A 来表示:

$$A = (A_1, A_2, \dots, A_p, \dots, A_{|P|}) \quad (1)$$

2) 图像分割可以归结到一个能量函数的最小化问题。按照文献[3]中所言, 全局最优的分割方式对应着下面式(2)中的能量函数取值最小。

$$E(A) = \lambda \cdot R(A) + B(A) \quad (2)$$

其中:

$$R(A) = \sum_{p \in P} R_p(A_p) \quad (3)$$

$$B(A) = \sum_{\{p, q\} \in N} B_{\{p, q\}} \cdot \delta(A_p, A_q) \quad (4)$$

其中:

$$\delta(A_p, A_q) = \begin{cases} 1 & \text{if } A_p \neq A_q \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$E(A)$ 是能量函数, $R(A)$ 称为区域惩罚项, $B(A)$ 称为边界惩罚项, 系数 $\lambda \geq 0$, 表示区域惩罚项相对于边界惩罚项的权重因子。其中, $R(A)$ 是一个与待分割像素相关的区域性函数, 衡量的是基于所选用的概率模型, 对待分割像素 p 赋予标记 A_p 的惩罚。而 $B(A)$ 是对邻域系统中的相邻像素对 $\{p, q\}$ 之间连接性的惩罚, 且仅当 p 和 q 不属于同一类时才考虑这类惩罚。一般当 p 和 q 越相似时 $B_{\{p, q\}}$ 越大, 当 p 和 q 相似度越低时时 $B_{\{p, q\}}$ 越接近于零。

3) 利用 $R_p(A_p)$ 和 $B_{\{p, q\}}$ 将待分割图像构造成一个无向图 G 。文献[3]中证明图像的能量函数可以用图中的一个割来表示:

$$E(A) = \lambda \cdot R(A) + B(A) = \dots = \sum_{e \in C} \omega_e \quad (6)$$

其中, C 是无向图中的某条割, e 是割上的边, ω_e 是这些边上的权值。当 C 是无向图 G 的一个最小割时, 能量函数 $E(A)$ 也取得最小值, 因此将图像分割问题转化为图论中的最大流最小割问题。解决了图论的问题, 就能对应回图像分割的解决方案。

综上所述, 图割的过程可以由图 1 说明: 这是一个尺寸为 3×3 的图像, 图 1a 中 B 表示被用户选中的背景像素, O 表示目标像素; 图 1b 是利用 $R(A)$ 和 $B(A)$ 构建出来的无向图 G ; 然后在图 1c 中用最大流最小割的方法找到图 G 的最小割“cut”; 最后利用图 1c 中的结果将图中的节点对应回图像中的像素, 就能找到能量函数有全局最优解的一个分割方案。

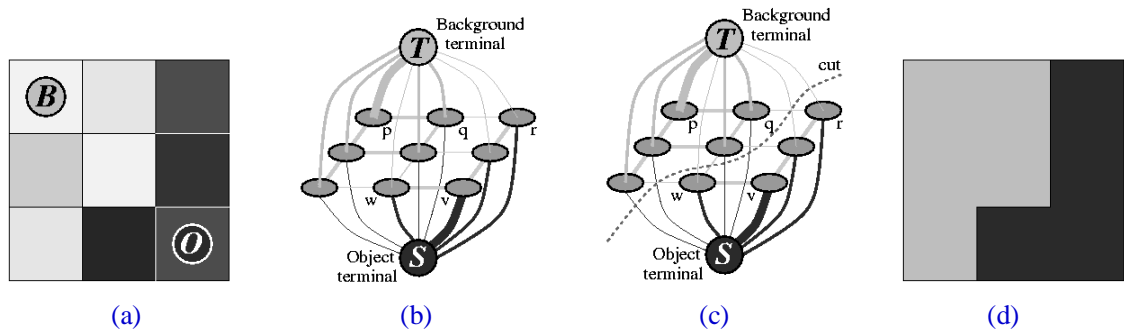


图 1 Graph Cut 算法求解过程

3.2 K 聚类算法

因为 Graph Cut 算法中建立的图考虑了邻域系统的权值，相当于考虑了图中像素的位置特征，所以只需要提取鼠标选定的目标像素点或背景像素点的颜色特征即可。在采用 C-KMC 算法取得其颜色特征时，用 RGB 空间的欧几里得距离作为分类标准^[4]。算法的输入是簇的数目 k 和 n 个像素点（每个点有三维特征，即像素的 RGB 颜色分量），输出是 k 个簇的中心（每个中心点也是有 RGB 的三维特征）。具体过程如下：

1. 选择 k 个初始化中心。如果 $n \leq k$ ，则直接进入步骤 4，不进行聚类；如果 $n > k$ ，则选择能将输入的 n 个像素点均分成 k 段的 k 个值作为初始值，并进入步骤 2。
2. 将样本集中的 n 个样本按照最小距离原则分配到最邻近的簇中。记样本集中第 $i(i=1,2,\dots,n)$ 个像素与第 $j(j=1,2,\dots,k)$ 个聚类中心的距离为 D_{ij} ，如公式(7)所示。

$$D_{ij} = \|I_i - I_j\| = \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2} \quad (7)$$

$$\text{其中,} \quad I = (r, g, b) \quad (8)$$

式(8)中的 I 是一个彩色空间的三维向量。遍历 k 个聚类中心，找到最小的 D_{ij} 值，将这第 i 个像素归入距离最短的簇中。按照这种方法将全部 n 个样本的归入最近邻的簇中。

3. 更新聚类中心。计算 k 个簇中的平均值向量，用此均值代替原来的聚类中心。
4. 反复执行第二步和第三步直至满足终止条件。本文的终止条件是没有新的聚类中心产生。

这样就得到了 k 个颜色特征值，这 k 个值将代表聚类之前的 n 个样本参与后续 Graph Cut 算法的分割运算。虽然聚类数目需要根据实验来确定，但一般来说满足 $k \ll n$ ，所以经过聚类预处理后可以提高算法的效率。

3.3 图割算法的改进

在传统 Graph Cut 算法中，用式(1)中的向量 A 来表示图像分割结果，但是这种 P 维向量和空间上二维的原图矩阵对应起来并不容易。为了方便地表示图像中某像素的类别，本文建立了一个“标志图”，记作 Img_Label 。它是一个尺寸与原图相同的灰度图像，且每个像素的灰度值只有两个取值：0 或 255，0 代表该位置对应于原图的像素属于背景；255 代表该位置对应于原图的像素属于目标。

因为建立的图 G 中边界惩罚 $B_{lp, q}$ 是邻域系统上边的权值，相当于考虑了图中像素的位置特征，所以区域惩罚项和边界惩罚项的公式只利用 RGB 颜色特征即可。又因为 Graph Cut 算法需要用户人工选取初始的目标像素和背景像素，而由于鼠标选点的连续性，紧密相邻的两个目标点（或背景点）的颜色和位置都

很接近，如果将它们都算作种子点去参与图割算法，将会做一些意义不大的重复工作，影响分割速度。所以本文采用基于 C-KMC 的 Graph Cut 算法，实验证明这种方法比传统方法在分割速度上提升了不少。

本算法的输入是像素值为全 0 的标志图 *Img_Label* 以及用户选取的目标（或背景）种子点，输出是代表分割结果的 *Img_Label*。具体分割过程可用下面的步骤描述。

1. 由用户选取一系列的目标种子点和背景种子点，然后然后用 C-KMC 算法分别对这两类种子点进行聚类，聚类的数目本文默认为 10。聚类结果记为集合 k_Obj （目标聚类中心）和 k_Back （背景聚类中心）。
2. 根据选取的种子点给图像构造一个无向图 G 。图像中每个像素 p 作为一个节点，再添加两个特殊的节点 S 和 T ，其中 S 是“obj”终端， T 是“bkg”终端。根据式(9)确定节点 p 在式(3)中的区域惩罚项 $R_p(A_p)$ ；再根据式(10)确定节点 p 在式(4)中的边界惩罚项 $B_{\{p,q\}}$ ：

$$\begin{cases} R_p(\text{"obj"}) = \lambda_1 / (1 + \min_{o \in k_Obj} \|I_p - I_o\|) \\ R_p(\text{"bkg"}) = \lambda_1 / (1 + \min_{b \in k_Back} \|I_p - I_b\|) \end{cases} \quad (9)$$

$$B_{\{p,q\}} = \lambda_2 / (1 + \|I_p - I_q\|) \quad (10)$$

其中， I 是彩色空间的三维向量；像素 $o \in k_Obj$ ，像素 $b \in k_Back$ ， q 是 p 的 4 邻域像素； λ_1 是一个比例系数， λ_2 也是一个比例系数，需要和 λ_1 互相配合用实验来确定，实际上，这两个系数的比值就相当于公式(2)中的系数 λ 。

3. 图 G 构建完成后，调用最大流算法找出全局最优解。
4. 将全局最优解用标志图 *Img_Label* 来表示，方便对分割得到的目标做后续处理。

总的来说，改进后的 Graph Cut 算法在对一般的图像进行分割时效果不错。但是当待分割图像比较复杂或者目标与背景颜色相似部分过多时，这种算法的效果也不太好。此时若适当增加 C-KMC 算法中聚类数目 k 可以改善这种情况，因为这样可以增加目标或背景种子点。

4 交互式图像分割系统设计

4.1 人机交互界面综述

人机交互界面的功能应包括原图像的显示、鼠标在原图像上划线来标定目标种子点和背景种子点、图像分割功能、分割后目标的显示以及对分割结果的后续操作。整个系统的 MFC 界面如下图 2。图上红色的标号并不属于系统界面，只是为了方便介绍各部分的功能。



图 2 分割系统的 MFC 界面

在上图中，标号为“Pic_1”和“Pic_2”的方形区域是 MFC 中的 Picture box（图片控件），结合 OpenCV 中的数据类型和函数，可以在控件中显示出彩色图像。标号为“0”到“7”的几个显示区域是 MFC 中的 Edit box（编辑框控件），只是将控件属性中的 Read Only 项设为 True，使它们只用来显示数据而不能由用户输入。标号为“In_1”也是一个 Edit box，但它是专门用来供用户输入的，输入的值是 K 均值聚类算法的聚类数目，而当没有输入时，系统默认聚类数取 10。另外一个输入部件是标号为“In_2”的 Combo box（组合框控件），下拉菜单中是三种背景颜色供选择，是专门为证件照换背景这个应用而设的。剩下的就是几个 Button box（按钮盒），其余的将在表 1 中简要介绍：

表 1 人机交互界面功能介绍

标号	显示内容	按钮名称	按钮功能
Pic_1	分割前-待分割图像； 分割后-配有新背景的目标图像	打开图像	打开一张图像，存入 IplImage* 结构指针中；初始化全部变量
Pic_2	分割各阶段的结果：目标用原图像素表示，背景用 纯蓝色代替	选择目标	将标志位 flag_Obj 置 1，使鼠标在 Pic_1 拖动画出的线为绿色
In_1	可输入数字，代表 K 均值聚类的数目；不输入时默 认为 10	选择背景	将标志位 flag_Back 置 1，使鼠标在 Pic_1 拖动画出的线为红色
In_2	证件照换背景时选择的背景颜色	图像分割	使用基于 C-KMC 的 Graph Cut 算法分割图像，在 Pic_2 显示结果
0	待分割图像的存储路径	二次交互	使用基于 C&P-KMC 的区域生长算法分割图像，在 Pic_2 显示结果
1	待分割图像的宽度，单位是像素	形态学滤波	使用形态学算法改善分割效果
2	待分割图像的高度，单位是像素	打开标准图像	选择一幅与待分割图像对应的标准分割图像进行准确率计算
3	鼠标画中的目标点的像素数目	更换背景	为分割后的目标换一个背景
4	鼠标画中的背景点的像素数目	油画效果	将换过背景的图像进行平滑处理
5	分割结果的查准率	新图像另存为	将被编辑过的图像保存起来
6	分割结果的查全率	退出	退出程序
7	分割程序所用的时间，单位是秒	取消	退出程序

4.2 分割算法的演示

由图 2 可知,真正执行图像分割算法的按钮只有 3 个,分别是“图像分割”、“二次交互”和“形态学滤波”。

首先,打开一幅图像,用鼠标分别选中目标像素点和背景像素点,就像图 3 中那样。然后按下按钮“图像分割”,屏幕上光标形状变为圆环状,表明程序正在进行中,当弹出一个消息对话框时,表明程序运行结束: Pic_2 中将会显示分割结果; Pic_1 中将会用蓝色的实线显示目标与背景之间的分界线;标号为 5 的编辑框中显示本次程序运行的时间;另外,为了方便进行二次交互,程序将目标点与背景点清零,编号为 3 和 4 的编辑框显示“0”。如图 4 所示,图像右侧有一些背景被误分作目标在 Pic_2 中显示了出来,所以这幅图像需要进行二次交互。

先按下按钮“选择背景”,表明鼠标画的点是背景,屏幕上会出现红色线段。然后在 Pic_1 中多余蓝色边界的位置处拖动鼠标画点。按下按钮“二次交互”,这次程序运行得很快,当弹出一个消息对话框时,表明程序运行结束。如图 5 所示,在男孩左边袖子上方的部位,目标图像较之图 4 得到了很大改善,但是依然有一些多余的零散的目标像素分布在背景中,此时,形态学滤波将派上用场。

直接按下“形态学滤波”按钮,当弹出一个消息对话框时,表明程序运行结束。如图 6 所示,原来零散分布的目标点都去除了,图像分割效果很好。至此,图像分割的部分就结束了。



图 3 目标和背景像素的选取图



图 4 初次分割结果



图 5 二次交互分割结果图

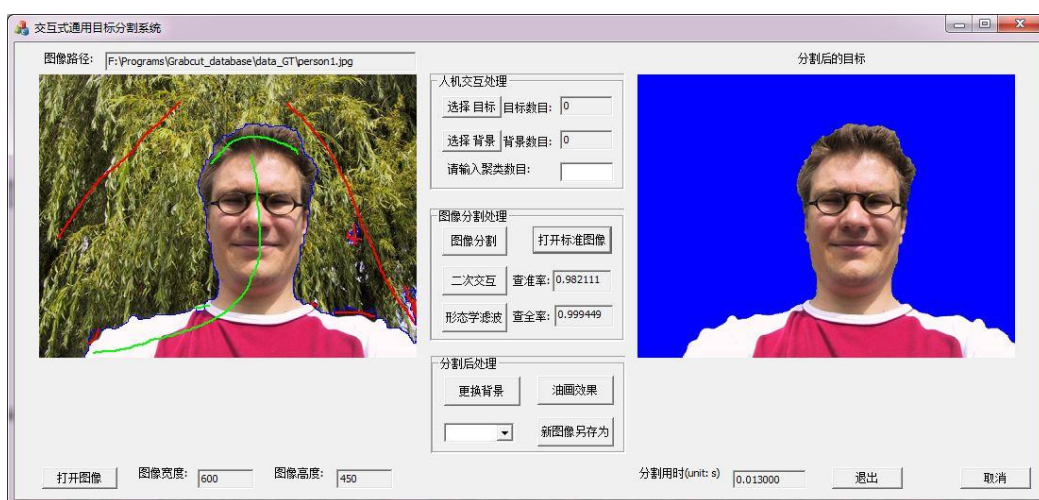










图 6 形态学滤波结果

4.3 图割算法参数的确定

Graph Cut 算法中有两个参数 λ_1 和 λ_2 需要结合试验来确定，在公式(10) 中， I 的取值范围是(0, 255)，所以 $\|I_p - I_q\|$ 的取值范围是(0, 255 $\sqrt{3}$)，为了使 $B_{I_p, q}$ 能在整数范围内，不妨设 $\lambda_2 = 10000$ ，然后通过改变 λ_1 的值，在不同图像上做实验，综合各个分割效果来确定其取值。实验结果如表 2 所示，可以看出： λ_1 越小，分割对区域越敏感，容易将细小连接断裂；而 λ_1 越大，分割对边缘越敏感，在背景颜色变化较多的图像中容易将一部分背景分到目标一类，综合考虑，本文中 λ_1 取 3000。

表 2 不同参数的分割结果

原图	$\lambda_1=1000$ 时分割结果	$\lambda_1=3000$ 时分割结果	$\lambda_1=5000$ 时分割结果
			
			

4. 4 对分割结果的交互操作

4. 4. 1 原图换背景与平移

为原图换背景以及平移是对分割结果的一种交互操作，将旧背景的 RGB 值换成新背景的 RGB 值即可达到更换图像背景或平移的目的。图 7 与图 8 分别是更换背景以及背景平移后的结果。

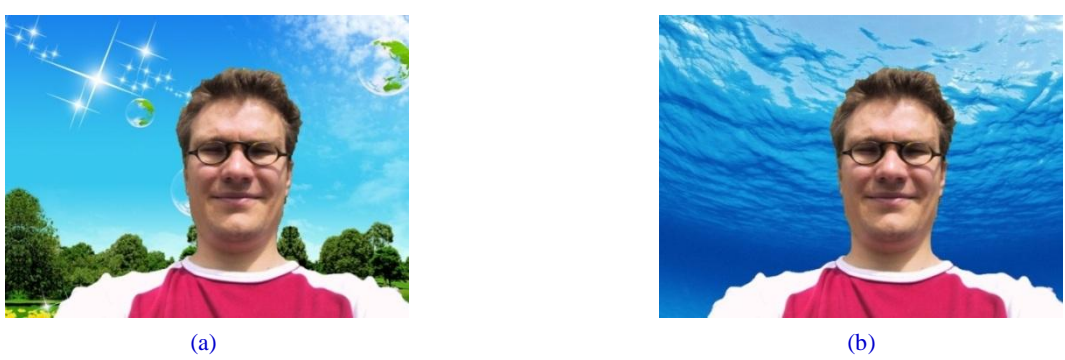


图 7 更换背景的结果



图 8 目标部分被平移的结果

4.4.2 图换平滑处理

本文采用 Mean Shift 算法进行彩色图像的平滑。把像素点以及相邻的像素点的 RGB 值取平均值作为该小区域的 RGB 代表值。处理的结果如图 9a 所示。对比平滑前后的两幅图像，可以看出在男孩的脸部和头发处能看到明显效果。对平滑结果再做一次平滑处理，如图 9b 所示，图像的“油画效果”更加明显。

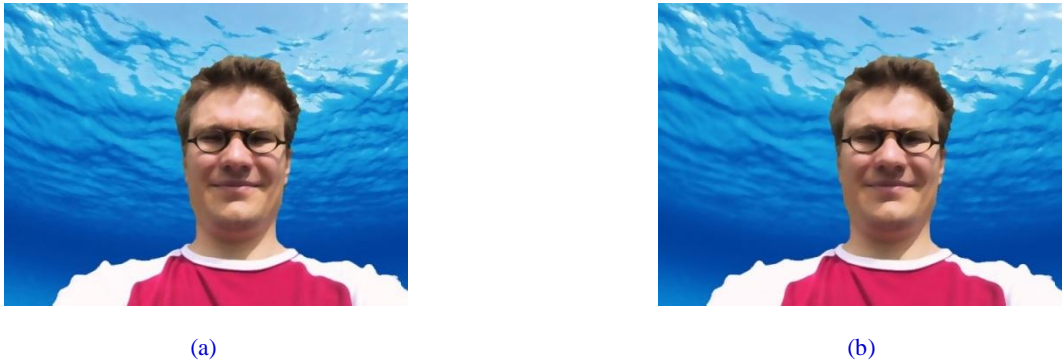











图 9 平滑处理结果

5 分割效果

对 3 幅标准图像进行分割实验，将分割结果列入表 3，分割时 K 均值聚类的参数 k 取默认值 10。表格第一列是图像的名称；第二列是用户标记区域，绿色线条表示目标，红色线条表示背景；第三列是初次交互采用基于 C-KMC 的 Graph Cut 算法得到的分割结果；第四列是采用基于 C&P-KMC 的区域增长算法得到的分割结果，有些图像需要进行多次交互才能得到满意结果；第五列是数据库中给出的分割结果。

表 3 本文分割结果与标准分割结果对比

图像	用户交互标记	初次分割结果	多次交互结果	标准分割结果
1				
2				
3				

从表 3 可以看出,对于目标背景相对简单的图像 1,初次交互分割得到的结果就已经比较令人满意了,此时进行二次交互主要是为了修饰细微部分。对于背景相对复杂的图像 2 或者目标区域本身具有一些孔洞的图像 3,第一次得到的结果里一些背景像素容易被错分成目标,这时二次交互甚至多次交互就是必须的了,并且可以看出经过多次交互后得到的分割结果明显优于前者。总体来说,分割效果还是不错的。

6 总结与展望

(1) 本文研究的是一种针对彩色图像的交互式分割方法。自行设计 Graph Cut 中的边的权值函数,并增加了 K 均值聚类预处理。

(2) 本文实现的是一个具有实用价值的图像分割软件。

(3) 对于一些目标与背景较为相似的复杂图像,初次分割的效果不好,分割达到满意效果需要的交互次数较多。

(4) 对于一些复杂的、大型的图像,分割时间较长。

(5) 可以选择更优的权值函数,获得更加的分割效果。

(6) 可以在算法上进一步完善,或者与其他算法相结合,达到更好的效果。

参考文献:

- [1] 林开颜, 吴军辉, 徐立鸿. 彩色图像分割方法综述[J]. 中国图象图形学报, 2005, 10(1) : 1-10.
- [2] 李俊山, 李旭辉. 数字图像处理[M]. 北京:清华大学出版社, 2007 : 167-168.
- [3] Boykov Y Y, Jolly M P. Interactive graph cuts for optimal boundary & region segmentation of objects in ND images[C]. Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE, 2001, 1: 105-112.
- [4] 吴永芳, 杨鑫, 徐敏, 张星. 基于 K 均值聚类的图割医学图像分割算法[J]. 计算机工程, 2011, 37(5) : 232-234.