

结合 CS-LBP 纹理特征的快速图割算法

刘毅^{1,2} 黄兵² 孙怀江¹ 夏德深¹

(南京理工大学计算机科学与技术学院 南京 210094)¹ (南京审计学院信息与科学学院 南京 210029)²

摘要 图割算法是目前最有效的交互式图像分割方法之一。针对当前景和背景颜色相似时容易发生分割错误并产生 shrinking bias 现象,以及基于像素的计算导致交互效率不高的问题,提出一种结合纹理特征的改进算法。该算法首先利用 Mean Shift 算法对图像进行预分割,构建区域邻接图,然后用累计直方图、CS-LBP 纹理描述子对每个区域进行颜色和纹理特征的提取,通过在能量函数中引入纹理约束项以及局部自适应的正则化参数,有效改善了分割效果和 shrinking bias 现象。实验结果表明,本算法交互效率得到了提高,分割结果更加精确。

关键词 图割, GrabCut, 均值漂移, 累积直方图, 中心对称局部二值模式

中图分类号 TP391.4 **文献标识码** A

Fast Image Segmentation Algorithm Combining CS-LBP Texture Features

LIU Yi^{1,2} HUANG Bing² SUN Huai-jiang¹ XIA De-shen¹

(School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China)¹

(School of Information and Science, Nanjing Audit University, Nanjing 210029, China)²

Abstract Graph cuts algorithm is one of the most effective interactive image segmentation methods. But it is prone to produce segmentation errors and shrinking bias phenomena when the color of foreground and background is similar and its interaction efficiency is not high due to pixel-based calculation. To improve these problems, an algorithm combining CS-LBP texture features was proposed in this paper. First the mean shift algorithm is applied to pre-segment the original image into regions to construct region adjacency graph. Then cumulative histogram and CS-LBP texture descriptor are used to extract color and texture features from each region. A new term of texture constraint is added to the energy function and local adaptive regularization parameter is used. So the segmentation effect and shrinking bias phenomenon are improved efficiently. The experiments show that interactivity efficiency and segmentation accuracy are improved.

Keywords Graph cuts, GrabCut, Mean shift, Cumulative histogram, CS-LBP

1 引言

在图像分析过程中,人们往往需要将感兴趣的前景对象从静态图像或视频中分割提取出来,才能做进一步的处理。由于前景对象的定义具有一定的主观性,因此全自动分割算法往往通用性较差,交互式半自动分割方法通过有限的用户交互引入先验信息,有效地引导分割过程,具有更强的实用性。由 Boykov 等人提出的 Graph Cuts^[1,2]算法,是目前最有效的交互式图像分割方法之一,具有全局最优、区域与边界特征相结合等很多优良的特性。但传统的图割算法只利用了图像的颜色特征,当前景和背景颜色相似时,分割效果较差,而且此时能量函数主要由边界项决定,容易发生 shrinking bias 现象^[3],不利于细长边界的分割。另外,图割算法是基于像素的,为提高算法的效率,Mean Shift^[4,5]算法、分水岭^[6,7]算法被用来构造超像素^[8]简化计算。

本文利用 Mean Shift 算法对图像进行预分割,通过区域

邻接图的构造,将图像表示成基于区域的,提高了算法效率。通过提取分割区域的颜色特征和 CS-LBP^[9]纹理特征,将颜色与纹理特征结合起来,采用局部自适应的正则化参数,改善了分割效果和 shrinking bias 现象。实验结果表明,算法效率得到了提高,分割效果和 shrinking bias 现象得到了明显改善。

2 图割算法模型

图像分割问题可以看成对图像中各像素进行二元(前景/背景)标号的过程。图割算法模型的核心思想在于构造一个包含区域和边界信息的标号能量函数,为求解能量函数的最小值,将图像映射为对应的带权图,通过网络流理论的应用,最终转化为求解带权图的最大流/最小割问题。标号能量函数一般定义为:

$$E(l) = R(l) + \lambda B(l) = \sum_{p \in P} R_p(l_p) + \lambda \sum_{(p,q) \in N} B_{(p,q)}(l_p, l_q) \quad (1)$$

式中,区域项 $R(l)$ 用来衡量标号 l 和用户通过交互指定的前景/背景观察数据的一致性;边界项 $B(l)$ 起到光滑的作用,约

到稿日期:2012-07-23 返修日期:2012-11-24 本文受基金项目中国数字化虚拟人切片图像分割研究(60805003)资助。

刘毅(1979—),男,博士生,讲师,主要研究方向为图像处理、多媒体信息检索, E-mail: s_worm@163.com;黄兵(1972—),男,博士,副教授,主要研究方向为模糊粗糙集理论及应用;孙怀江(1968—),男,博士,研究员,博士生导师,CCF 高级会员,主要研究方向为证据理论、信任管理、模式识别;夏德深(1941—),男,博士,教授,博士生导师,CCF 高级会员,主要研究方向为图像处理、卫星遥感、模式识别。

束相邻像素应具有相同的标号值, λ 起到调节作用。

由式(1)可以发现,当前景和背景的颜色很相似时,区域项的值将也很接近,能量函数的值主要取决于边界项,此时用户通过交互引入的先验信息作用有限,从而导致错误的分割,而且当边界项起主导作用时,能量函数的最小化对细长边界的分割将产生影响,容易导致 shrinking bias 现象。因此,当颜色特征不能区分时,必须考虑重要的纹理特征,将颜色与纹理结合起来。

3 结合 CS-LBP 纹理特征的快速图割算法

针对 Graph Cuts 算法存在的基于像素导致的效率不高、特征单一、存在 shrinking bias 现象 3 个主要问题,本文通过 Mean Shift 预分割,构造区域邻接图,将颜色特征与纹理特征结合起来,计算局部自适应的正则化参数,并逐一进行改进,下面将具体讨论改进的思路。

3.1 Mean Shift 预分割

由于 Mean Shift 算法与分水岭算法相比产生过分割的现象较少,能够较好保持图像的结构,因此选取 Mean Shift 算法对图像进行预分割是较好的选择,本文采用了文献[10]所描述的方法。经过预分割后,得到 n 个区域块 $R = \{R_i | i=1, \dots, n\}$ 。为提高 Graph Cuts 算法的效率,将每个区域 R_i 抽象成一个超级像素,建立区域邻接图 $G=(V, E)$, V 由区域块 R_i 抽象而成, E 由相邻区域构成,如图 1 所示,并将用户标记的前景/背景像素所在的区域映射到 V 中相应的顶点作为邻接图的前景种子集合 $\{\theta^f\}$ 、背景种子集合 $\{\theta^b\}$ 。

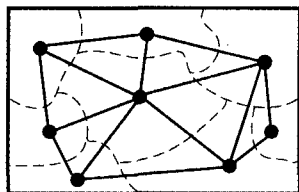


图 1 Mean Shift 预分割后构建的区域邻接图

3.2 颜色与纹理特征的提取

彩色直方图作为一种统计特征描述算子可用来描述区域中颜色的分布,对 Mean Shift 预分割获取的区域 R_i 进行颜色特征的提取。为降低直方图的维数,将 RGB 3 个分量分别均匀量化为 16 个等级,但是这种简单的量化存在一个严重的问题,会把两种相近的颜色量化到同一个区间,也可能量化到不同的区间。累积直方图^[11]较好地解决了这一问题,不仅消除了普通直方图中常见的零值,也克服了普通直方图量化过粗带来的误差放大问题。可以通过 $H(k) = \sum_{i=0}^k \frac{n_k}{N}$, $k=0, 1, \dots, 15$ 得到归一化的累积直方图(n_k 为区域 R_i 具有量化值为 k 的像素个数, N 为区域 R_i 像素的总个数)。分别提取区域 P 和 Q 的 RGB 3 个分量的累积直方图,通过式(2)计算它们的相似距离。

$$d_c(P, Q) = \sqrt{\sum_{i=0}^{15} \left[\frac{(H_P^R(i) - H_Q^R(i))^2 + (H_P^G(i) - H_Q^G(i))^2 + (H_P^B(i) - H_Q^B(i))^2}{3} \right]} \quad (2)$$

同样,对区域 R_i 进行纹理特征的提取,局部二值模式(LBP)^[12]纹理描述子是目前对于二维图像简单高效的局部纹理描述子之一。然而,LBP 提取的特征维数通常很高(256

维),且对平坦区域不具有鲁棒性^[13],CS-LBP^[9](Center-Symmetric Local Binary Pattern)采用中心对称的思想,对图像 4 个方向的变化进行编码,将特征维数降低到 16 维,如图 2 所示。

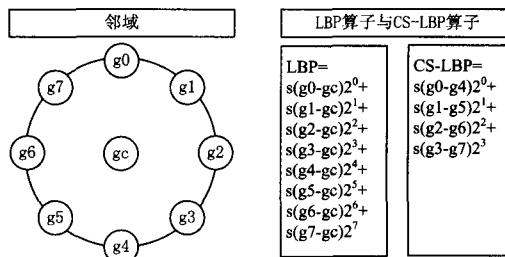


图 2 LBP 与 CS-LBP 原理

为加强对平坦区域的鲁棒性,CS-LBP 对灰度的变化设置了一个阈值 T ,通用的 CS-LBP 描述子可以表示为:

$$CS-LBP_{R,N,T}(x, y) = \sum_{i=0}^{(N/2)-1} s(n_i - n_{i+(N/2)}) 2^i$$

$$s(x) = \begin{cases} 1, & x > T \\ 0, & \text{otherwise} \end{cases}$$

式中, R 为邻域半径, N 为像素个数,本文采用常用的 3×3 邻域,根据文献[9]的建议,阈值 T 取灰度范围的 1%。类似颜色直方图的处理,本文对提取的 CS-LBP 纹理谱同样计算累积直方图。

3.3 区域项的构造

当前景和背景的颜色特征相似时,仅依靠颜色特征将很难区分,因此必须将颜色特征与纹理特征结合起来,构造新的区域项:

$$R(l) = -\ln(\rho P_C(v_i | l_i) + (1-\rho) P_T(v_i | l_i)) \quad (3)$$

式中, v_i 为区域邻接图的第 i 个顶点, $l_i \in \{0, 1\}$ 为前景/背景标号, $P_C(v_i | l_i)$ 为颜色约束项, $P_T(v_i | l_i)$ 为纹理约束项, ρ 调整两者的比重。要计算颜色和纹理约束项,必须计算前景与背景之间的颜色/纹理的相似程度。分别计算顶点 v_i 与前景和背景种子在颜色和纹理上的最短相似距离(参见式(2)),即: $D_i^{FC} = \min_k(d_c(v_i, \theta_k^f))$, $D_i^{BC} = \min_k(d_c(v_i, \theta_k^b))$, $D_i^{FT} = \min_k(d_T(v_i, \theta_k^f))$, $D_i^{BT} = \min_k(d_T(v_i, \theta_k^b))$, 越相似距离值越小,因此可以定义:

$$P_C(v_i | l_i) = \begin{cases} \frac{D_i^{FC}}{D_i^{FC} + D_i^{BC}}, & \text{if } l_i = 1 \text{ and } D_i^{FC} + D_i^{BC} \neq 0 \\ \frac{D_i^{BC}}{D_i^{FC} + D_i^{BC}}, & \text{if } l_i = 0 \text{ and } D_i^{FC} + D_i^{BC} \neq 0 \\ 0.5, & \text{if } D_i^{FC} + D_i^{BC} = 0 \end{cases}$$

$$P_T(v_i | l_i) = \begin{cases} \frac{D_i^{FT}}{D_i^{FT} + D_i^{BT}}, & \text{if } l_i = 1 \text{ and } D_i^{FT} + D_i^{BT} \neq 0 \\ \frac{D_i^{BT}}{D_i^{FT} + D_i^{BT}}, & \text{if } l_i = 0 \text{ and } D_i^{FT} + D_i^{BT} \neq 0 \\ 0.5, & \text{if } D_i^{FT} + D_i^{BT} = 0 \end{cases}$$

由式(3)可知, ρ 的作用是确保当前景和背景颜色相似时,纹理约束起的作用较大,当纹理相似时,颜色约束起的作用较大,因此定义:

$$\rho = \begin{cases} \frac{D_i^{FC} + D_i^{BC}}{D_i^{FC} + D_i^{BC} + D_i^{FT} + D_i^{BT}}, & \text{if } D_i^{FC} + D_i^{BC} + D_i^{FT} + D_i^{BT} \neq 0 \\ 0.5, & \text{otherwise} \end{cases}$$

3.4 边界项的构造

边界项由相邻顶点之间的颜色距离决定,定义:

$$B_{(p,q)}(l_p, l_q) = \begin{cases} 0, & \text{if } l_p = l_q \\ \exp\left(-\frac{d_c(v_p, v_q)^2}{\sigma^2}\right), & \text{if } l_p \neq l_q \end{cases} \quad (4)$$

式中, σ 为邻接像素的距离差异均值。

为进一步缓解 shrinking bias 的影响, Vicente 等^[3]在图割算法完成初步分割后, 需要用户提供额外的提示信息, 利用 Dijkstra 算法计算最短路径实现细长边界的连通。Candemir 等^[4]则采取局部自适应的正则化参数来降低边界处的惩罚, 提高细长边界出现的概率, 但需要用 Canny 算子进行边缘检测。基于文献^[14]的思想, 由于前景像素之间的差异较小, 而前景与背景像素间的差别较大, 它们之间会出现边界, 因此, 降低分属于前景和背景像素之间的惩罚, 从而得到带局部自适应正则化参数的边界项, 由式(1)能量函数构造为:

$$E(l) = \sum_{p \in P} R_p(l_p) + \sum_{(p,q) \in N} (1 - |P(v_p|1) - P(v_q|1)|) \lambda B_{(p,q)}(l_p, l_q) \quad (5)$$

将式(3)、式(4)代入式(5)即可得到修改后的能量函数。

4 实验与结果分析

为验证本文算法的正确性和有效性, 采用实验平台为: Microsoft Visual Studio 2010, CPU: Intel i520, RAM: 2GB, 通过实验取 $\lambda=1$, 利用 Berkeley 分割图库、GrabCut^[15]提供的标准测试图像, 分别从分割效果、交互效率两方面分析本文算法的性能。

4.1 分割效果

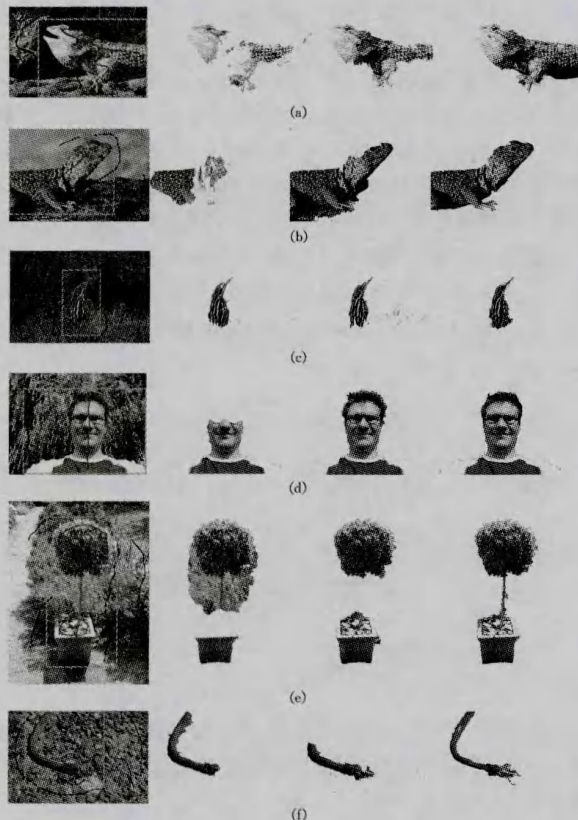


图3 本文算法与 GrabCut、文献^[1]算法分割效果对比(从左到右依次为: 用户交互标记图像(红色-前景、蓝色-背景、绿色-GrabCut 矩形标记框)、GrabCut 分割结果、文献^[1]分割结果、本文算法分割结果)

将本文算法与 GrabCut 算法及文献^[1]算法的分割效果进行对比, 如图 3 所示。从分割效果可以发现, 本文算法对图像纹理部分分割效果的改善非常明显, 如图 3(a)~图 3(d)所示, 对 shrinking bias 现象也有所改善, 如图 3(e)中的树干、图 3(f)中蛇的身体和昆虫的脚。另外, 本文算法的分割结果中也出现了一些图像细节的丢失, 这是由 Mean Shift 算法预分割对部分图像结构细节的破坏造成的。

4.2 交互效率

本文算法的耗时由 Mean Shift 算法预分割的时间和图割算法分割的时间构成, 由于预分割只要进行一次, 而分割过程可能需要用户多次交互对分割结果进行修正, 需要多次运行图割算法, 因此, 图割算法的运行效率对交互过程的影响较大。GrabCut 算法需要迭代分割确定 GMM 参数, 而交互过程不再进行迭代, 以初次分割计算交互时间。3 种算法的交互效率比较结果如表 1 所列, 可见本文算法由于在区域邻接图上进行计算, 交互效率有了大幅提升。

表 1 交互效率对比

图像编号	GrabCut 算法(ms)	文献 ^[1] 算法(ms)	本文算法(ms)
图 3(a)	2012	1514	780
图 3(b)	3713	2777	1279
图 3(c)	2574	2340	1201
图 3(d)	2512	1778	968
图 3(e)	7317	5273	2356
图 3(f)	2637	2668	1264

结束语 本文将颜色特征和纹理特征结合起来, 构造局部自适应的正则化参数, 较好地避免了前景和背景颜色相似时图割算法容易产生分割错误并导致 shrinking bias 的现象。通过采用区域邻接图, 缩小了 s-t 网络流图的规模, 改善了图割算法基于像素计算交互效率不高的问题。实验结果表明本文算法在分割效果、交互效率方面均有较大的改善。但由于 Mean Shift 算法预分割对部分图像结构细节的破坏, 导致分割结果中也出现了部分图像细节的丢失, 如何解决这一问题, 值得我们下一步重点考虑。

参考文献

- [1] Boykov Y, Jolly M P. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images[C]//Proceedings of Computer Vision, 2001(1):105-112
- [2] Boykov Y, Funka-Lea G. Graph cuts and efficient N-D image segmentation[J]. International Journal of Computer Vision, 2006, 70(2):109-131
- [3] Vicente S, Kolmogorov V, Rother C. Graph cut based image segmentation with connectivity priors [C]//Proceedings of Computer Vision and Pattern Recognition, 2008:1-8
- [4] 韩守东, 赵勇, 陶文兵, 等. 基于高斯超像素的快速 Graph Cuts 图像分割方法[J]. 自动化学报, 2011, 37(1):11-20
- [5] 刘技, 康晓东, 贾富仓. 基于图割与均值漂移算法的脊椎骨自动分割[J]. 计算机应用, 2011, 31(3):760-762
- [6] Peng Bo, Zhang Lei, Zhang D, et al. Image segmentation by iterated region merging with localized graph cuts[J]. Pattern Recognition, 2011, 44(10/11):2527-2538
- [7] 徐秋平, 郭敏, 王亚荣. 基于分水岭变换和图割的彩色图像快速分割[J]. 计算机工程, 2009, 35(19):210-212

(下转第 314 页)

高度,大大减少了计算复杂度,说明该算法是一种较好的算法。

表 2 3种计算表面高度算法运行时间的比较

法线图大小	本文算法	迭代算法	函数计算法
32 * 32	0.20s	0.18	0.23s
64 * 64	0.70s	0.84s	0.73s
128 * 128	2.42s	3.87	3.4s
256 * 256	8.20s	12.36s	11.23s

3.2 算法结果验证

在上述相同的 PC 上验证高度图的正确性,实验采用能体现高度效果的浮雕贴图技术,即利用一张标准的表面法线图和用算法恢复的高度图,在光照情况下产生凹凸更加明显的效果,此外也采用了没有高度信息的法线贴图技术作为对比。该验证实验利用 OpenGL 接口通过 CG 语言编程实现。浮雕贴图技术的实现是基于 Fabio 提出的二分搜索方法^[11],它的核心思想是除了利用表面法向量之外,还利用高度信息改变位移实现凹凸效果^[12]。图 8 就是使用表面法向量图(见图 6)和本文算法生成的高度图(见图 7(e))实现的凹凸效果,它的凹凸效果明显强于仅仅用法线贴图的凹凸纹理映射(见图 9),尤其是离光照比较远的部分。

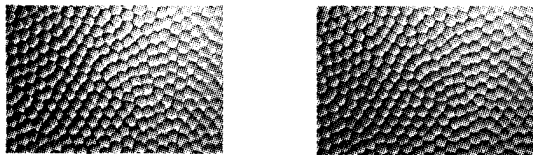


图 8 利用表面法线图和本文算法生成的高度图的凹凸效果
图 9 只利用表面法线图的凹凸效果

结束语 本文提出了一种从物体表面法向量估计高度信息的新颖算法,为了实现它,首先根据表面法线图计算出高度差值,并且利用高度差值图来保存物体表面像素与像素之间的高度差,然后,选定极坐标积分的半径以及步进角度,分别对每个像素进行极坐标积分运算。实验结果表明,本文的算法具有较高的准确度,生成的高度图完全可以用来实现浮雕贴图技术。由于算法计算量较大,本文方法仍然不适合实时渲染高度信息,只能对有需求的物体模型事先做好高度图,这是本算法需要改进之处。

(上接第 302 页)

[8] Veksler O, Boykov Y, Mehrani P. Superpixels and Supervoxels in an Energy Optimization Framework[C]//Proceedings of European Conference on Computer Vision (ECCV'10). 2010: 211-224

[9] Heikkilä M, Pietikainen M, Schmid C. Description of Interest Regions with Center-Symmetric Local Binary Patterns [C]//Proceedings of ICVGIP. 2006:58-69

[10] Comaniciu D, Meer P. Mean Shift: A Robust Approach Toward Feature Space Analysis[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(5): 603-619

[11] Stricker M A, Orengo M. Similarity of Color Images[C]//Proceedings of Storage and Retrieval for Image and Video Databases (SPIE). 1995:381-392

参考文献

[1] Blinn J F. Simulation of wrinkled surfaces[C]//Proceedings of SIGGRAPH. 1978:286-292

[2] Zhang R, Tsai P-S, Cryer J E, et al. Shape from shading: A survey[J]. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1999, 8(21): 690-706

[3] Feng Qiao-sheng, Wang Yun-qiong, Tian Jie. Low Dimensional and Easily Interactive Method of 3D Shape from Shading[C]//International Conference on Wireless Communications, Networking and Mobile Computing. 2010

[4] Ren Guo-quan, Li Wen-zhao, Chen Liang, et al. Application of Shape from Shading Algorithm in Wear Debris 3D Surface Shape Recovery[C]//International Conference on Measuring Technology and Mechatronics Automation. 2010: 1:586-589

[5] Smith G D J, Bors A G. Height Estimation From Vector Fields of Surface Normals[C]//Digital Signal Processing. 2002: 1031-1034

[6] 王松,李著文,于金辉.利用拓片恢复汉画像的浮雕效果[J].计算机辅助设计与图形学学报,2011,23(5):784-789

[7] Zatzarinni R, Tal A, Shamir A. Relief Analysis and Extraction [J]. ACM Transactions on Graphics, 2009, 5(28)

[8] Wang Jian-guo, Wang Xiao-tong, Xu Xiao-gang. The Algorithm of Extraction Base Surface from Mesh[C]//International Conference on Information, Electronic and Computer Science. 2010: 1467-1470

[9] Tasdizen T, Whitaker R, Burchard P, et al. Geometric Surface Processing via Normal Maps [J]. Transactions on Graphics, 2003, 4(22): 1012-1033

[10] Kelly A R, Hancock E R. Surface Height Recovery From Surface Normals Using Manifold Embedding [C]//International Conference on Image Processing. 2004:2107-2110

[11] Policarpo, Fabio. Relief Mapping in a Pixel Shader Using Binary Search[EB/OL]. http://www.Paralelo.com.br/A_rquivos/ReliefMapping.pdf, 2012-07-04

[12] Miyazaki R, Harada K. Creating the Displacement Mapped Low-Level Mesh and its Application for CG Software[J]. International Journal of Image and Graphics, 2010, 3(10): 467-480

[12] Ojala T, Pietikainen M, Harwood D. A comparative study of texture measures with classification based on feature distributions [J]. Pattern Recognition, 1996, 1(29): 51-59

[13] Heikkilä M, Pietikainen M. A texture based method for modeling the background and detecting moving objects [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(4): 657-662

[14] Candemir S, Akgül Y S. Adaptive Regularization Parameter for Graph Cut Segmentation [C]//Proceedings of ICIAR. 2010 (6111): 117-126

[15] Rother C, Kolmogorov V, Blake A. Grabcut: Interactive foreground extraction using iterated graph cuts[J]. Proceedings of SIGGRAPH, 2004, 23(3): 309-314