

基于贪心算法的快速模糊连接度图像分割

周得水 葛洪伟

(江南大学物联网工程学院 江苏 无锡 214122)

摘要 提出一种基于 Dijkstra 的贪心算法来实现模糊连接度的快速计算方法。每次寻找全局最优解,并从两个方面减少迭代次数来加快运算速度。首先,通过统一根节点模糊连接度的方式,去掉原始 Dijkstra 算法的更新操作,每个像素点一次即可完成模糊连接度的计算。其次,每一次迭代可以实现多个像素点模糊连接度的计算。通过真实医学图像测试表明,与其他改进算法相比,其分割速度明显加快,且不影响其分割精度。

关键词 模糊连接度 图像分割 Dijkstra 算法 贪心算法

中图分类号 TP391 文献标识码 A DOI:10.3969/j.issn.1000-386x.2015.08.048

FAST FUZZY CONNECTEDNESS FOR IMAGE SEGMENTATION BASED ON GREEDY ALGORITHM

Zhou Deshui Ge Hongwei

(School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, Jiangsu, China)

Abstract In this paper, we present a Dijkstra-based greedy algorithm to realise the fast calculation method of fuzzy connectedness. It searches the global optimal solution every time and decreases the times of iteration from two aspects to speed up the computation. First, it removes the update operation of original Dijkstra's algorithm by the way of unifying the root node fuzzy connectedness, once per pixel point can the calculation of fuzzy connectedness be completed. Secondly, every iteration can realise the computation of fuzzy connectedness of multiple pixel points. It is showed by real medical image test that the new method is obviously faster in segmentation speed than other improved algorithms, and the accuracy of the segmentation doesn't be affected.

Keywords Fuzzy connectedness Image segmentation Dijkstra's algorithm Greedy algorithm

0 引言

图像分割是将图像划分为互不相交区域的过程,主要目的是从一副图像中提取出目标区域,方便下一步的图像处理与图像分析。在医学领域,图像分割是计算机辅助诊断的关键,由于图像自身以及各种人为和机器因素影响,使得医学图像中目标边缘不可避免具有模糊性和不均匀性^[1]。

自从 1965 年 Zaden 提出了模糊数学的概念之后,1979 年 Rosenfeld 将模糊数学里模糊连接度的概念引入到图像处理中,2003 年 Udupa 在此基础上将其应用于图像分割,并提出了模糊连接度和相对模糊连接度概念以及模糊连接理论的框架^[1],并进行了改进和扩充^[2,3],在医学图像的分割中取得了很好的分割效果^[4]。但是当处理大型数据集的时候,需要耗费很长的运算时间,一幅图大约需要消耗几个小时^[5],因而很多时候无法满足实际的需要。因为一幅图像中任意两点间路径有无数条,不可能穷尽其所有路径。在文献[1]中,提出使用动态规划的方式来实现模糊连接度的计算,但是仍然存在很多冗余工作。Nyul L. G^[6]等人提出一种方法去掉原始的动态规划算法中由于与生长点相邻的四邻域中有些点的模糊连接度还未算出导致

该路径连接度为零的冗余计算,并提出对于不同类型的分割对象采用不同的数据结构来计算亲和度。这种方法虽然能加快运算速度,但是增加亲和度计算的方式选择问题,选择不好同样影响分割效果和效率。Carvalho B. M 和潘建江等人^[7,11]提出采用贪心算法,可以加快模糊连接度的计算。每次将当前路径作为局部最优路径,但局部最优路径不一定是全局最优,需要通过后期更新来实现全局最优路径的计算。李雪丽等人^[9]通过先找最优路径,再计算模糊连接度的方式来简化模糊连接度的计算,但是在某些情况下会出现误算,得到的某个像素的模糊连接度比实际的要小。近年来,随着硬件技术的快速发展,Gammage C 等人^[10]采用多处理机系统,Y. Zhuge 等人^[12-15]使用 GPU 实现模糊连接度算法的并行加速,取得一定的效果但增加了硬件成本。

本文提出一种基于 Dijkstra 的快速贪心算法 ADFC (Advanced Dijkstra Fuzzy Connectedness)来实现模糊连接度计算的加速。ADFC 算法可以在一次迭代中完成所有当前种子点中模糊连接度相等点邻域点的模糊连接度计算,而且每次迭代都能找到全局最优路径,不需要更新,减少了迭代次数,加快运算速度。

收稿日期:2013-11-26。周得水,硕士生,主研领域:人工智能与模式识别。葛洪伟,教授。

1 模糊连接度理论

1.1 模糊邻近度、模糊亲和度

设 X 为集合, μ_p 称为集合 $X \times X$ 上的隶属函数, 则集合 X 上的二元模糊关系 ρ 定义为:

$$\rho = \{ (x, y) \in X \times X | ((x, y), \mu_p(x, y)) \} \text{ and } \mu_p: X \times X \rightarrow [0, 1] \quad (1)$$

设二维空间上的二元模糊关系 α 满足自反和对称性, 则称 α 为模糊空间元素的邻近关系。用其描述了两个空间元素位置的邻近度函数为^[1,2]:

$$\mu_\alpha(c, d) = \begin{cases} 1 & \|c - d\| \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

设 $S = (C, f)$ 是 (Z^n, α) 上的一个隶属度场, 若 C 上的一个二元模糊关系 κ 满足自反和对称性, 则称为 S 上模糊空间元素的亲和度。任意两像素间亲和度函数为^[2,3]:

$$\mu_\kappa(c, d) = \mu_\alpha(c, d) (w_1 \mu_\phi(c, d) + w_2 \mu_\psi(c, d)) \quad (3)$$

式中: $w_1 + w_2 = 1$, 其中 μ_ϕ 和 μ_ψ 分别为特征分量和均匀分量如下:

$$\mu_\phi(c, d) = e^{-((1/2) \times [(1/2) \times (f(c) + f(d)) - m_1] / s_1)^2} \quad (4)$$

$$\mu_\psi(c, d) = e^{-((1/2) \times [|f(c) - f(d)| - m_2] / s_2)^2} \quad (5)$$

式中 m_1, m_2 和 s_1, s_2 分别表示 $f(c) + f(d)$ 和 $|f(c) - f(d)|$ 的均值和方差, 取值来源于区域 A , 且其分割结果与区域 A 的选取有关。

1.2 路径连接度、模糊连接度

设 κ 为 S 上模糊空间元素亲和度, 在 κ 中从像素点 c 到像素点 d 一条路径 ρ_{cd} 定义为: 空间元素序列 $\langle c_1, c_2, \dots, c_m \rangle$ ($m \geq 2$), 其中 $c_i (i = 1 \dots m) \in C, c_1 = c, c_m = d$ 。

当然, S 中从点 c 到点 d 有许多条路径, 对于某一条从点 c 到点 d 的路径 $\rho_{cd} = \langle c_1, c_2, \dots, c_m \rangle$, 该路径的路径连接度为路上最弱的链接, 如下:

$$\mu_N(\rho_{cd}) = \min(\mu_\kappa(c_1, c_2), \dots, \mu_\kappa(c_{m-1}, c_m)) \quad (6)$$

从点 c 到点 d 的模糊连接度定义为从点 c 到点 d 的所有路径中路径连接度最大值, 记 C 中从 c 到 d 的所有路径为 P_{cd} , 则模糊连接度的隶属函数如下^[2]:

$$\mu_\kappa(c, d) = \max_{\rho_{cd} \in P_{cd}} [\mu_N(\rho_{cd})] \quad (7)$$

2 ADFC 算法

由式(6)和式(7)可知, 从种子点 s 经过当前种子点 c 到生长点 d 的标准模糊连接度基于动态规划的算法 GFC (General Fuzzy Connectedness), 如式(8)^[2], 其中生长点 d 满足 $N(d) = \{d | \mu_\kappa(c, d) = 1\}$ 。

$$\mu_\kappa(s, d) = \max_{c \in N(d)} [\min(\mu_\kappa(s, c), \mu_\kappa(c, d))] \quad (8)$$

基于 Dijkstra 的贪心算法 DFC^[7,11] (Dijkstra Fuzzy connectedness) 能降低算法复杂度。DFC 算法将(8)式中自上往下搜索变为自下往上搜索如式(9):

$$\begin{aligned} \mu_\kappa(s, d) &= \max_{d \in N(c)} [\min(\mu_\kappa(s, c), \mu_\kappa(c, d))] \\ &= \max[\mu_\kappa(s, c), \min_{d \in N(c)} (\mu_\kappa(c, d))] \\ &= \min[\mu_\kappa(s, c), \max_{d \in N(c)} (\mu_\kappa(c, d))] \end{aligned} \quad (9)$$

式(9)中像素 $d \in N(c)$ 为一变量, 因此对于某一确定的像素 d 式(9)可以简化为式(10)^[7,8]:

$$\mu_\kappa(s, d) = \min[\mu_\kappa(s, c), \mu_\kappa(c, d)] \quad (10)$$

在一副图像中, 由于同一个区域内像素点灰度相似性非常高, 因此同区域内部相邻点间的亲和度相等概率很大, 从而导致在同区域内很多像素点到种子点的模糊连接度相等。即存在多个当前种子点的集合 $M = \{c'_1, c'_2, \dots, c'_k\}$, 其中 $M \subseteq Q, Q = \{c_1, c_2, \dots, c_n\}$ 表示当前种子点队列, 且满足式(11)的情况:

$$\mu_\kappa(s, c'_1) = \mu_\kappa(s, c'_2) = \dots = \mu_\kappa(s, c'_k) \quad (11)$$

设集合 M 的邻域点即迭代生长点集合 D 如下:

$$\begin{aligned} D &= \{d_{c'_1}^1, \dots, d_{c'_1}^i, d_{c'_2}^1, \dots, d_{c'_2}^j, \dots, d_{c'_k}^1, \dots, d_{c'_k}^h\} \quad 1 \leq i, j, h \leq 4 \\ \{d_{c'_1}^1, \dots, d_{c'_1}^i\} &\in N(c'_1) \quad \{d_{c'_2}^1, \dots, d_{c'_2}^j\} \in N(c'_2), \dots, \{d_{c'_k}^1, \dots, d_{c'_k}^h\} \\ &\in N(c'_k) \end{aligned}$$

若当前种子点的邻域点不重叠, 如式(12), 则需用式(10)进行 k 次迭代完成集合 D 中生长点模糊连接度计算。

$$N(c'_1) \cap N(c'_2) \dots \cap N(c'_k) = \phi \quad (12)$$

若当前种子点的邻域点存在重叠, 如图1中第二次迭代时当前种子点集合为 $M' = \{C1, C2, C3, C4\}$, 则生长点集合 $D = \{D1, D2, D3, D4, D5, D6, D7, D8\}$ 其中, $D1, D3, D5, D7$ 四个生长点为 M' 的邻域重叠点。如用式(10), 则这四个点均需要进行两次迭代, 通过更新来计算这四点的模糊连接度。

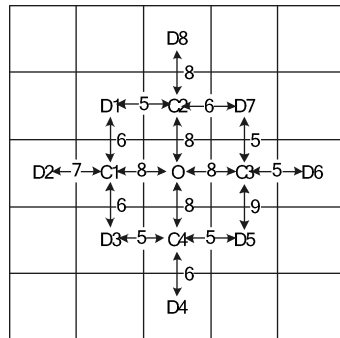


图1 模糊连接度算法图解

为减少上述两种情况下的迭代次数, 加快运算速度, 对于满足式(11)的当前种子点集合 M , 一次迭代完成集合 M 对应的生长点集合 D 的模糊连接度的计算。每次迭代都先在当前种子点集合 Q 中找出满足式(11)的集合 M , 当集合 M 中有邻域点存在重叠时, 直接选择亲和度最大的那条路径作为最优路径。采用式(13)一次迭代完成上述两种情况下模糊连接度的计算。

$$\mu_\kappa(s, d') = \min_{c \in M} [\mu_\kappa(s, c), \max_{d \in N(c)} (\mu_\kappa(c, d))] \quad (13)$$

其具体的算法如下:

输入: 图像 I , 种子点集合 S 以及任意两点间的亲和度 $\mu_\kappa(s, d)$, 其中 $s \in S, d \in I$ 。

输出: 每个像素点到种子点的模糊连接度 $F(c), c \in I$ 。

辅助数据结构: 数组 Q , 集合 M 。

1. $c \in I$, 令 $F(c) = 0$. 将 $s \in S$ 加入集合 Q , 令 $F(S) = 1$
2. while ($Q \neq \text{null}$) {
3. $M = \max(F(Q))$; // 取出 Q 中所有 F 最大的像素点
4. for ($c \in M$) {
5. $A_d = \max_{d \in N(c)} (\mu_\kappa(c, d))$;
// 找出集合 M 中像素重叠邻域点的最优路径
其中 $N(c) = \{c | \mu_\kappa(c, d) = 1\}$
6. for ($d \in N(c)$) {
 $FC_d = \min_{c \in M} (F(M), A_d)$;
// 计算集合 M 每个邻域点最优路径的连接度
7. if $F(d) = 0$, 说明该点模糊连接度还未算出,

8. 则令 $F(d) = FC_d$, 将赋值同时, 将 d 放入集合 Q 中}

在第 5 步, 当集合 M 中只有一个元素时, (5) 等价于 DFC 算法, 即式 (12)。当集合 M 中有多个元素时, 一个实例见图 1 所示。对图 1 在进行第二次迭代时当前种子点集合为 $M = \{C1, C2, C3, C4\}$, 其第二次迭代计算过程见表 1 所示。图 1 中每一个方格代表一个像素, 方格正中心字符为该像素编号, 其中 0 表示人工选择的种子点, 两个方格间双箭头上数字表示两像素间亲和度, 为表示方便将其值扩大了十倍。表 1 中 A1 表示 $\max(\mu_k(c, d))$, A2 表示 $f_c(M)$, A3 表示 ADFC 算法的计算得到的结果。A4 表示 GFC 算法得到的结果。

表 1 ADFC 算法对图 1 第二次迭代结果与 GFC 算法结果对比

	C1	C2	C3	C4	A1	A2	A3	A4
D1	6	5			6	8	6	6
D2	7				7	8	7	7
D3	6			5	6	8	6	6
O	8	8	8	8	8	8	8	
D7		6	5		6	8	6	6
D8		8			8	8	8	8
D5			9	5	9	8	8	8
D6			5		5	8	5	5
D4				6	6	8	6	6

从表 1 中可以看出 ADFC 算法和 GFC 算法计算得到的结果是一样的。

在文献[9]中提出通过每次迭代都先找与生长点的邻近点间具有最大亲和度的那条路径, 并将该路径当作最优路径的方式来计算模糊连接度, 简称 LFC, 能简化计算, 但是, 在图 2 所示的几种情况下计算出来的都是错误的模糊连接度, 且其值要比实际模糊连接度要小。

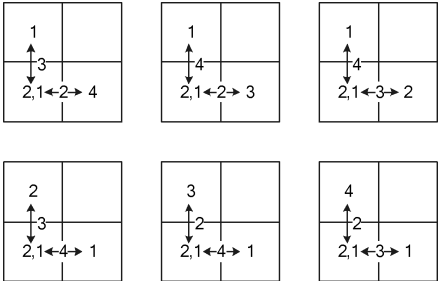


图 2 GFC 与 LFC 计算结果对比

在 ADFC 算法中, 出现当前种子点的邻域点间存在重叠的前提是当前种子点的模糊连接度是相等的, 通过这种统一根节点模糊连接度的方式, 能得到正确的模糊连接度, 见图 3 中 ADFC 算法得到的结果和 GFC 算法得到的结果一致。

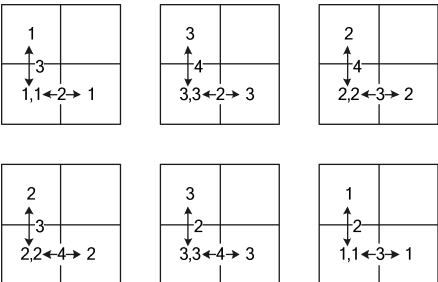


图 3 GFC 与 ADFC 计算结果对比

在图 2 和图 3 中一个方格代表一个像素, 方格中数字表示该点模糊连接度, 双箭头上数字表示两点间亲和度 (扩大了 10 倍)。其中左下角两个数字分别表示两种算法得到的模糊连接度。左边的数字均表示原始模糊连接度算法得到的结果。图 2 和图 3 中左下角右边的数字分别表示 LFC 算法和 ADFC 算法得到的模糊连接度结果。

2.1 实验环境及数据参数

为验证算法的有效性, 在 32 位 windows 7 系统, Inter Pentium i3, 2.20 GHz 处理器, 2 GB 内存的平台上用 MATLAB 7.11.0.584(R2010b), 采用来自宾夕法尼亚大学医学院真实人体腹部和胸部 CT 图进行两次实验测试。目标是分割出腹部的肝脏和胸部的呼吸系统 (肺和呼吸道), 其图片大小均为 512 × 512。在计算亲和度时为减少区域 A 的影响和区域 A 的选取工作, m 和 s 的取值均采用所取目标区域或背景区域种子点及其邻域点像素的均值和方差。且 w_1 取 0.8, w_2 取 0.2。

实验采用相对模糊连接度^[2]的方法来实现目标提取, 设 P_{obj} 表示目标区域像素点的集合, O 为目标种子点集合, B 为背景种子点集合, 则相对模糊连接度获取目标如式 (14) 所示:

$$P_{obj} = \{c | c \in C, \mu_k(O, c) > \mu_k(B, c)\} \quad (14)$$

为了检验分割精度, 设标准分割结果为 Φ_0 , 实验中得到的分割结果为 Φ 则实验中的识别率 P 的计算如式 (15):

$$P = \left(1 - \frac{|\Phi \otimes \Phi_0|}{|\Phi|}\right) \times 100\% \quad (15)$$

式中, $|\Phi \otimes \Phi_0|$ 为通过异或运算得出的实验分割结果与标准分割结果不符的像素点个数, $|\Phi|$ 表示整个图像的像素个数。

不同算法分割腹部的肝脏和胸部的呼吸系统 (肺和呼吸道) 分割耗时和分割精度见表 2 和表 3。表中 OT(s)、BT(s) 和 T(s) 分别表示计算目标模糊连接度、背景模糊连接度和计算两种模糊连接度之和所占用的 CPU 时间, 单位为秒。P(%) 为识别率。

表 2 不同算法分割腹部肝脏 CT 图的时间和精度对比

	GFC	LFC	DFC	ADFC
OT(s)	11 525.24	194.16	484.18	47.82
BT(s)	12 555.83	1025.78	478.64	47.67
T(s)	24 081.07	1219.94	962.82	95.49
P(%)	98.0125	97.4869	98.0125	98.0125

表 3 不同算法分割胸部呼吸系统 CT 图的时间和精度对比

	GFC	LFC	DFC	ADFC
OT(s)	8820.1	206.49	445.59	27.78
BT(s)	2395.6	1445.86	530.43	41.12
T(s)	11 215.7	1652.35	976.02	68.90
P(%)	97.3808	96.7793	97.3808	97.3808

不同算法分割腹部的肝脏和胸部的呼吸系统 (肺和呼吸道) 结果分别见图 4 和图 5, 每副图中六副小图分别为: (a) 待分割的图, 其中红色十字叉中心为选取的目标种子点, 绿色十字叉中心为选取的背景种子点, 图 4 中目标种子点的坐标为 (290, 127), 背景种子点坐标为 (413, 475), 图 5 中目标种子点的坐标为 (170, 279)、(252, 253) 和 (378, 282), 背景种子点坐标为 (277, 223); (b) 标准分割结果, 由医学专家分割所得; (c)、

Journal of Applied Mathematics, 2009, 39(1): 48-60.

- [9] 刘平, 刘晓曼, 朱永贵. 基于 K-SVD 字典学习的核磁共振图像重建方法[J]. 中国传媒大学学报:自然科学版, 2013, 20(4):132-140.
- [10] Wright J, Ma Y, Mairal J, et al. Sparse representation for computer vision and pattern recognition[J]. Proceedings of the IEEE, 2010, 98(6):1031-1044.
- [11] 余南南, 邱天爽, 毕峰, 等. 基于 K-SVD 的医学图像特征提取和融合[J]. 大连理工大学学报, 2012, 52(4): 219-226.
- [12] Hunter J E, Tugcu M, Wang X, et al. Exploiting sparse representations in very high-dimensional feature spaces obtained from patch-based processing[J]. Machine Vision and Applications, 2011, 22(3): 449-460.
- [13] 陈坤, 马燕, 刘俊. 一种自适应无监督彩色图像分割方法[J]. 小型微型计算机系统, 2013, 34(4):241-247.
- [14] 王娜, 郭敏. 结合差异控制的 FCM 方法在 MR 图像分割中的应用[J]. 计算机工程与应用, 2009, 45(34):212-214.
- [15] Rueda A, Malpica N, Romero E. Single-image super-resolution of brain MR images using overcomplete dictionaries[J]. Medical image analysis, 2013, 17(1):113-132.

(上接第 203 页)

(d)、(e)、(f) 分别表示 GFC、LFC、DFC 和 ADFC 算法的分割结果。

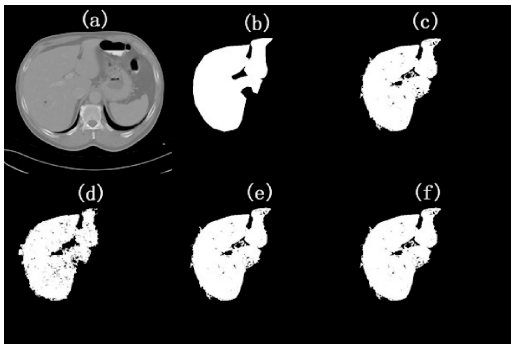


图 4 不同算法分割腹部肝脏结果对比

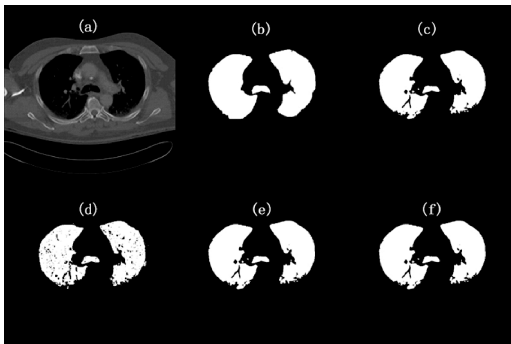


图 5 不同算法分割胸部呼吸系统结果对比

2.2 实验分析

从上述两个实验可以看出, GFC 用动态规划计算很耗时, 每次都需要几个小时, 令人无法忍受, DFC 用贪心算法计算明显能对 GFC 加速 7~8 倍^[7], LFC 算法虽然能一定程度上加快运算速度, 但是会出现误计算, 降低了分割效率。从时间和精度对比中可以看出本文提出的 ADFC 算法能在 DFC 的基础上加速 8~10 倍, 且不影响其分割效率。

3 结 语

ADFC 算法减少了 GFC 算法和 DFC 算法的冗余计算和迭代次数。可以在每次迭代同时计算多个像素点与种子点间的最优路径, 实现并行计算, 实验结果表明, 明显地加快了算法运行速度。且并行计算时, 由于保证了根节点模糊连接度相同, 因此可以直接先找最优路径, 每次找到的都是最优路径, 不会出现误计算的同时保证了识别率。如果同时将其与硬件结合, 将能更快地给模糊连接度算法加速。

参 考 文 献

- [1] Udupa J K, Samarasekera S. Fuzzy Connectedness and Object Definition: Theory, Algorithms and Applications in Image Segmentation[J]. Graphical Models and Image Processing, 1996, 58(3):246-261.
- [2] Udupa J K, Saha P K. Fuzzy Connectedness and Image Segmentation[J]. Proceedings of the IEEE, 2003, 91(10):1649-1669.
- [3] Clesielski K C, Udupa J K, Saha P K. Iterative relative fuzzy connectedness for multiple objects with multiple seeds[J]. Computer Visual and Image Understand, 2007, 107(3):160-182.
- [4] Zhou Yongxin, Jing Bai. Atlas-based fuzzy connectedness segmentation and intensity non-uniformity correction applied to brain MRI[J]. Bio-medical Engineering, 2007, 54(1):121-129.
- [5] Udupa J K, Wei L, Samarasekera S, et al. Multiple sclerosis lesion quantification using fuzzy connectedness principles[J]. Medical Imaging, 1997, 16(5):598-609.
- [6] Nyul L G, Falcao A X, Udupa J K. Fuzzy-connected 3D image segmentation at interactive speeds[J]. Graphical Models, 2002, 64(5):259-281.
- [7] Carvalho B M, Au C J, Erman G T, et al. Algorithms for fuzzy segmentation[J]. Pattern Analysis Application, 1999, 2(1):73-81.
- [8] Clesielski K C, Udupa J K, Falcao A X, et al. Fuzzy Connectedness Image Segmentation in Graph Cut Formulation: A Linear-Time Algorithm and a Comparative Analysis[J]. Mathematical Imaging and Vision, 2012, 44(3):375-398.
- [9] 李雪丽, 周果宏, 罗述谦. 用于血管图像分割的简化模糊连接算法[J]. 计算机辅助设计与图形学学报, 2003, 15(10):1224-1229.
- [10] Gammage C, Chaudhary V. On Optimization and Parallelization of Fuzzy Connected Segmentation for Medical Imaging[C]//Advanced Information Networking and Applications, 2006, 2(18):623-627.
- [11] 潘建江. 基于模糊连接度的图像分割及算法[J]. 软件学报, 2006, 16(1):67-76.
- [12] Ying Zhuge, Yong Cao, Miller R W. GPU accelerated fuzzy connected image segmentation by using CUDA[J]. Engineering in Medical and Biology Society, 2009:6341-6344.
- [13] Ying Zhuge, Yong Cao, Udupa J K, et al. Parallel fuzzy connected image segmentation on GPU[J]. Medical Physics, 2011, 38(7):4365-4371.
- [14] Ying Zhuge, Clesielski K C, Udupa J K, et al. GPU-based relative fuzzy connectedness image segmentation[J]. Medical Physics, 2013, 40(1):011903-1-10.
- [15] Ying Zhuge, Udupa J K, Clesielski K C, et al. GPU-based Iterative relative fuzzy connectedness image segmentation[J]. Medical Physics, 2013, 40(1):011903-1-10.