

多阈值优化的交互式医学图像分割方法

兰红

(江西理工大学信息工程学院 赣州 341000)

摘要 交互式图像分割方法对边界模糊的医学图像进行分割时通常需要用户标记较多的初始种子或进行二次交互,这给用户带来不便。针对此问题,提出一种简化标记的多阈值优化交互式分割算法,该算法在 GrowCut 交互式算法基础上通过引入图像灰度直方图的多个阈值自动生成初始种子模板,并利用改进的细胞自动机迭代算法实现图像分割。算法简化了用户操作,提高了分割精度。算法应用于临床肝脏图像和牙菌斑图像分割,显示了良好的分割效果。

关键词 交互式,多阈值,灰度直方图,细胞自动机,医学图像分割

中图分类号 TP391 文献标识码 A

Interactive Medical Image Segmentation Algorithm Optimized by Multi-thresholds

LAN Hong

(School of Information and Technology, Jiangxi University of Science and Technology, Ganzhou 341000, China)

Abstract Interactive image segmentation methods usually require users to provide much more initial seeds or more than once interactive when they are used for medical image segmentation with fuzzy boundaries. This paper presented an optimized interactive image segmentation algorithm with multi-thresholds technology. The proposed algorithm is based on GrowCut algorithm and improved in two aspects; one is automatically generating initial seeds templates by image gray histogram with multi-thresholds, and the other is raising iterative efficiency by improved cellular automaton iterative algorithm. Compared with GrowCut algorithm, the proposed algorithm simplifies the user interactive operations and improves the segmentation accuracy. Experimental results on clinical plaque and liver image segmentations demonstrate the sound performances of the proposed algorithm.

Keywords Interactive, Multi-thresholds, Histogram, Cellular automata, Medical image segmentation

交互式图像分割的研究目的就是通过尽可能简便而直观的人机交互,快速准确地提取出图像中用户感兴趣的前景物体。它可以弥补全自动分割由于缺少先验知识或用户交互等因素造成的分割不精确等问题^[1],正成为越来越流行的研究热点。

交互式分割因其标记方法的不同可以分为两类,一类是基于边界的局部方法,如智能剪刀(Intelligent Scissors)^[1]算法、Snake^[2]算法和水平集(LevelSet)^[3]算法等。另一类是基于种子点的全局方法,如 Lazy Snapping^[4]算法、图割(Graph-Cut)^[5]算法和 GrabCut^[6]算法等。基于边界的方法以图像前景的边界为出发点,要求用户使用一条曲线大致标明边界位置,然后根据图像的特征对曲线进行优化,进而提取出前景物体边界^[1-3]。基于种子点的方法不需要用户标记整个前景物体的轮廓,只需在前景和(或)背景区域内粗略地标记一些种子点,系统根据这些种子点利用后台的全局优化算法提取出目标边界^[4-6]。

细胞自动机(Cellular Automata CA)^[7]是由 Ulam 和 Von Neumann 于 1966 年提出的一个在时间和空间上离散的非线性动力学模型,它能够通过简单的行为规则实现复杂计算。基于细胞自动机的图像处理自被提出以来已广泛应用于各种应用领域的动态系统,包括图像去噪和边缘检测等^[8]。

2006 年 Vladimir 与 Vadim 提出了基于细胞自动机的交互式图像分割算法,也称 GrowCut 方法^[9],该方法具有较强的鲁棒性,可以作用于任意维图像,有能力完成较难分割的任务。然而 GrowCut 算法依赖于用户的初始种子标记,种子点过多或标记不准确直接影响算法的迭代速度和分割效果。为此本文提出一种基于多阈值优化的 GrowCut (Multi-Thresholds GrowCut, MTGC)算法,以简化用户操作,提高 GrowCut 算法的鲁棒性和分割精度。

1 GrowCut 算法介绍

1.1 GrowCut 算法的基本思想

GrowCut 算法^[9]的核心思想是利用细胞自动机原理迭代计算细胞及其邻域之间的强度关系,据此改变细胞状态,实现前景目标的分割。

数字图像可以看成是 $k \times m$ 像素的二维矩阵,因而 GrowCut 算法将其视作一个具有特殊结构状态的细胞自动机。一个细胞自动机可以定义为一个三元组 $A = (S, N, \delta)$,其中 S 代表非空状态集, N 是一个邻域系统, $\delta: S^N \rightarrow S$ 是局部转换函数,该函数可根据细胞邻域的第 t 次迭代状态值确定该细胞第 $t+1$ 次迭代时的状态。该算法在数字图像应用中,设 p 为图像像素, P 为像素集集合, $p \in P \subseteq Z^n$,则该图像

到稿日期:2012-11-07 返修日期:2013-02-25 本文受江西省教育厅科技项目(GJJ11465)资助。

兰红(1969—),女,博士生,副教授,主要研究方向为图像处理、模式识别, E-mail:lanhong69@163.com。

对应的细胞自动机状态集和邻域系统分别为 $S(p)$ 和 $N(p)$, 其中 $S(p)$ 也是一个三元组, 定义为:

$$S(p) = (l_p, \theta_p, \vec{C}_p) \quad (1)$$

式中, l_p 为细胞标号; θ_p 为细胞强度, 取 $[0, 1]$ 之间的数值; \vec{C}_p 为细胞的特征向量, 取图像的基本信息, 如颜色、灰度等。当用户通过标记前景背景定义初始种子开始分割时, 种子细胞标号 l_p 相应设定, 细胞强度 θ_p 设为种子像素的强度值, 这样就设置了细胞自动机的初始状态。

细胞状态的修改通过邻域模型迭代实现。GrowCut 算法中邻域模型采用 3×3 的 Moore 型邻域实现, Moore 型邻域模型定义如下:

$$N(p) = \{q \in Z^n : \|p - q\|_\infty = \max_{i=1, n} |p_i - q_i| = 1\} \quad (2)$$

其邻域距离通常采用无穷范数求解。GrowCut 算法中邻域距离定义为细胞 p 和邻域细胞 q 的特征向量 \vec{C}_p 和 \vec{C}_q 之间的差。在迭代过程的每个时刻, 如果邻域细胞 q 的攻击强度大于防御细胞 p 的强度, 防御细胞被攻破, 细胞 p 的标号和强度都被改变。这样局部竞争的结果就是最强的细胞占领了邻域场, 逐渐扩展到整个图像; 迭代计算持续直至收敛到一个稳定结构, 细胞状态不再改变。

1.2 GrowCut 的初始种子模板设计

GrowCut 算法是基于种子点的多标记算法, 要求用户首先标记出目标前景和背景, 如图 1(a)^[9] 所示, 红色点线标记前景, 蓝色点线标记背景。

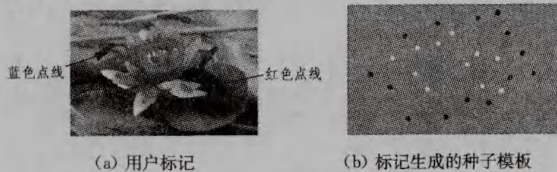


图 1 GrowCut 算法的交互式初始化

根据用户标记的前景和背景, GrowCut 算法设计了一个与原图像大小相等的包含表示前景、背景和无关区域 3 个整数值种子模板图。如图 1(b) 所示, 其中白点表示前景, 标记 $l_p = 1$; 黑点表示背景, 标记 $l_p = -1$; 其余区域为无标记区域, $l_p = 0$ 。模板中白点和黑点的选取依据用户标记的像素点坐标。

1.3 GrowCut 算法存在的不足

GrowCut 算法中的目标分割依赖于初始种子点的设定。用户标记的种子点太少或者标记不准确, 都会导致分割结果出现较大误差。虽然 GrowCut 算法提供分割过程中的二次交互, 用户可以再增加新的标记点, 但增加之后, 算法还要重新再生成种子模板, 这样不仅增加用户的负担, 还影响了分割速度。图 2、图 3 是采用 GrowCut 算法分别定义不同种子点分割局部肝脏图像的效果图。

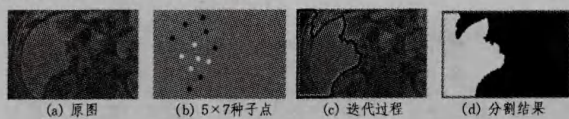


图 2 5 个前景点 7 个背景点的肝脏分割

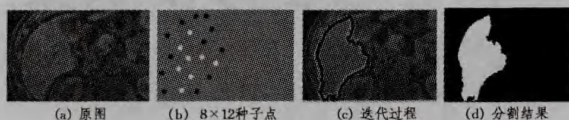


图 3 8 个前景点 12 个背景点的肝脏分割

图 2 中根据用户标记生成了 5 个前景点 7 个背景点的种子模板, 可以看出边界处的分割效果不好。图 3 中用户在边界处增加了种子点, 定义了 8 个前景点 12 个背景点, 分割效果比图 2 好, 但底部边缘的分割还是不理想, 若要得到精确分割, 还需用户再增加种子点。

对比图 2(a) 和图 3(a) 的标记可以看出, 种子点的数量和标记位置直接影响分割结果。特别是对于像素相似且距离很接近的前景和背景, 如果要准确分割出目标边缘, 就需要用户很仔细地标记初始前景点和背景点, 给用户带来不便。对此引入图像自身特性, 结合像素的颜色和亮度等信息, 自动生成标记种子点, 以简化用户的操作。

2 多阈值优化的 GrowCut 图像分割算法 (MTGC)

2.1 简化的初始种子设定

针对 GrowCut 中存在的初始种子设置需要用户较大工作量的问题, 本文借鉴文献 [6] 中 GrabCut 的交互方法, 提出一种改进的初始种子标记方法: 首先使用矩形选取框工具包围目标区域, 将此区域作为“背景”, 图 4(a)~图 4(c) 显示了矩形框的操作过程, 用户只需分别点击目标区域的左上角和右下角, 就可以形成如图 4(c) 所示的矩形框闭合区域; 再在包围区域内采用同样的矩形框标记“前景”目标, 如图 4(d) 所示。利用矩形选取框工具标记前景和背景对普通用户而言操作更为便捷, 也更容易掌握。

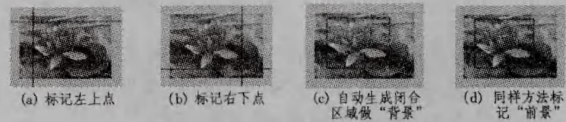


图 4 简化的前景、背景标记

2.2 多阈值优化的初始种子模板自动生成

采用矩形框标记“前景”和“背景”在构建如图 2(b) 所示的初始种子模板时, 需要选择具有代表性的合适的像素点作为细胞自动机中初始标号为 1 和 -1 的细胞。在此引入图像多阈值方法进行处理, 根据图像灰度直方图中的局部阈值进行分类, 求取每一类的均值点作为每种颜色像素的代表, 自动生成初始种子模板。

首先定义一个三元图 $T = \{T_B, T_U, T_F\}$ 表示图像区域, T_B 、 T_F 分别代表用户标记的背景区域和前景区域, T_U 为未标记的剩余区域。则利用灰度直方图自动生成初始种子模板的步骤如下:

(1) 输入图像 I , 绘制图像 I 的直方图。直方图区间为 $[0, L]$ ($L=255$), 如图 5 所示, 红色星线表示归一化的直方图分布, 其中横轴代表灰度值 ($gray_level$), 纵轴代表像素个数 ($number$)。

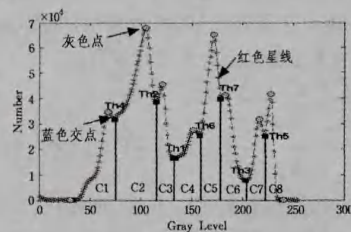


图 5 灰度直方图的多阈值

(2) 计算直方图中的极大值点。遍历直方图区间 $[1, L-1]$ 中每一点 $gray_level(i)$, 比较该点与其左右邻点 $gray_level$

$(i-1)$ 和 $gray_level(i+1)$ 的个数值,如果该点的个数值比其左右两点值都大,则标记为极大值点,图5中用灰色圆点表示。

(3)计算直方图中的极小值点。与求取极大值点方法相同,不同的是如果该点的像素个数值比其左右两点值都小,则标记为极小值点,图5中用蓝色方点表示。

(4)根据极大极小值点所对应的像素灰度值求取多个阈值 Th_1, Th_2, \dots, Th_n ,生成 $n+1$ 个像素类,方法如下:

a)在极大值点集合中选取两个最大值,比较两个最大值之间的极小值,选取最小的极小值作为最小值,以该最小值所对应的像素灰度值作为第一个阈值 Th_1 ;该阈值将所有像素分为 C_1, C_2 两类,并形成两个像素区间;

b)在 $[0, Th_1)$ 和 $(Th_1, 255]$ 区间内分别求取两个最大值之间的最小值,选取最小值对应的像素灰度值得到 Th_2 和 Th_3 ,两个类分裂成4个类;

c)再分别在 $[0, Th_2), (Th_2, Th_1), (Th_1, Th_3), (Th_3, 255]$ 区间内用同样方法求取阈值 $Th_4 \sim Th_7$,若子区间内不存在阈值则不再搜索此区间,4个类分裂成8个类;

d)根据 $Th_1 \sim Th_7$ 再划分子区间求取局部阈值,为了避免过细分割,设定最大分类数 C_{max} ,当子区间数超过 C_{max} 或者不再有新阈值生成时停止求取局部阈值。

(5)根据阈值标记像素区间 C_1, C_2, \dots, C_{n+1} ,计算每个区间内的像素分布的均值 v_1, v_2, \dots, v_{n+1} ,以此作为像素区间内像素值的代表。

(6)分别计算用户定义的蓝色框背景区域 T_B 和红色区域 T_F 的边界值,并选取 T_B 和 T_F 区域内像素为 v_1, v_2, \dots, v_{n+1} 的像素点坐标分别作为背景种子和前景种子点。

(7)根据种子点自动生成初始种子模板,初始标号为0,背景标号为-1,前景标号为1,其流程如图6所示。

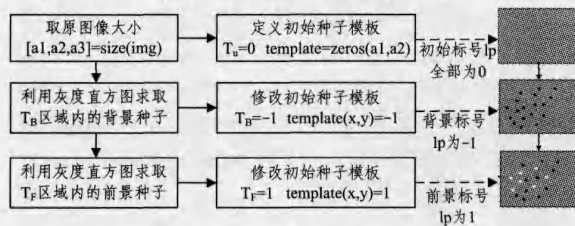


图6 初始种子模板生成流程

基于灰度直方图的初始种子模板简化了用户的操作,融合了图像像素信息。按照局部阈值进行分类,选取局部阈值区间的均值作为前景和背景种子可以使初始种子模板拥有更多信息,为后续迭代算法的快速分割提供基础。

2.3 迭代过程优化

GrowCut算法依据细胞自动机的工作原理进行迭代。根据式(1)的三元组定义和式(2)的邻域模型,细胞 p 对其邻域细胞 q 的攻击强度通过以下公式进行计算:

$$\theta_q' = g(\|\vec{C}_p - \vec{C}_q\|_2) \cdot \theta_q \quad (3)$$

其中 $g(x) = 1 - \frac{x}{\max\|\vec{C}\|}$ 是取值在 $[0, 1]$ 之间的单调下降函数。

每一次迭代过程中比较 θ_q' 和 θ_p 的大小,如果 $\theta_q' > \theta_p$,则将邻域细胞 q 的标号和强度赋给 p ,然后进入下一轮迭代。

GrowCut算法在迭代过程中遍历图像域 X 中的每个细胞 p ,计算细胞 p 及其邻域细胞 q 的强度值并比较大小。邻域通常取 3×3 邻域,即每个细胞 p 要和其周围8个邻域细胞

进行比较。这对于高精度复杂图像来说需要耗费较多的时间。为此本文对GrowCut的迭代过程进行了优化,即在计算邻域细胞 q 的强度之前,增加对 q 标号值的判断。因为初始种子模板中标记为0的部分属于非前景又非背景的无关区域,如果邻域细胞 q 当前时刻的标号为0,表明该邻域细胞 q 不会对细胞 p 产生“攻击”,则无需比较该邻域细胞 q 对细胞 p 的攻击强度,转而继续判断 p 的下一个邻域细胞。优化后的算法有助于减少算法的迭代次数,提高运行速度。

2.4 算法实现过程

总结上面的介绍,MTGC算法的实现过程就是首先由用户利用简单的矩形框标记前景和背景区域,算法通过求取图像多阈值自动生成初始种子模板,然后利用改进的细胞自动机迭代算法修改种子细胞标号,最终实现目标图像的分割。算法的具体流程如图7所示。

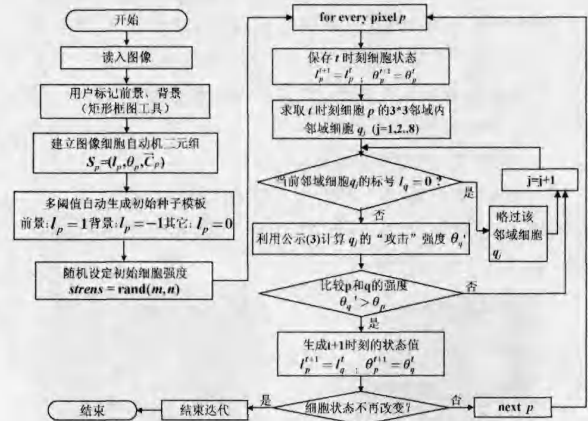


图7 MTGC算法流程

由于计算过程是根据图像像素映射的细胞进行扫描,保证了算法的收敛性,并且 g 是取值范围在 $[0, 1]$ 的单调递减函数,因此每个细胞强度的增长是有界的。

3 实验验证与效果分析

为了验证本文算法的有效性,算法在Window XP操作系统平台上,采用Matlab工具和VC相结合实现。实验数据来源于临床研究的图像数据库。以牙齿斑口腔图像和腹部MRI肝脏图像分割为例,与目前流行的交互式分割算法GrabCut和原有GrowCut算法进行了对比。其中GrabCut是基于图割的交互式图像分割方法,利用能量最小化函数迭代实现分割,其初始种子定义采用一个矩形框作为初始分割背景,前景提取利用高斯混合模型根据颜色分类建模迭代完成。3种算法的分割效果分别如图8和图9所示。

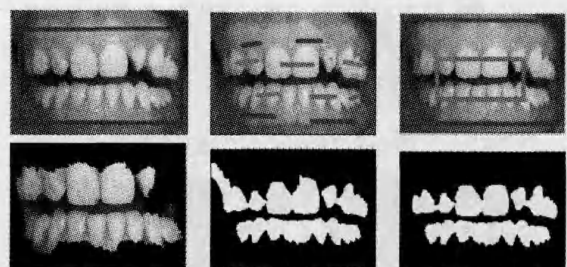


图8 牙齿斑口腔图像3种算法分割效果比较

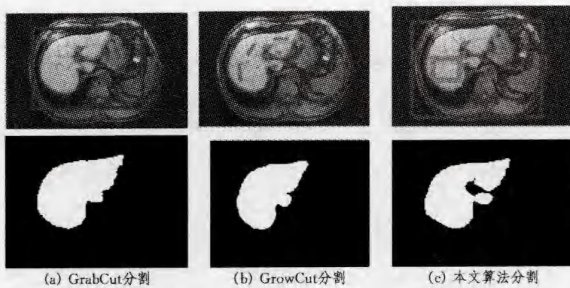


图9 肝脏图像分割3种算法分割效果比较

图8(a)和图9(a)是采用GrabCut算法分割的效果图,图8(a)中GrabCut分割无法区分出每颗牙,且上下牙之间的部分也作为前景提取了出来;图9(a)的分割轮廓不够清晰。图8(b)和图9(b)是采用GrowCut方法分割的结果,由于初始种子定义的不同,导致分割效果出现误差,如图8(b)中牙齿图像的左上部、图9(b)中肝脏中间的沟回分割都不够精确。图8(c)和图9(c)是采用本文提出的改进方法进行分割的结果,其灰度直方图和种子模板分别如图10和图11所示。由于利用灰度直方图自动生成种子模板,使得用户标记区域内的像素点均可以根据其类别进行标记,从而得到较好的分割效果。

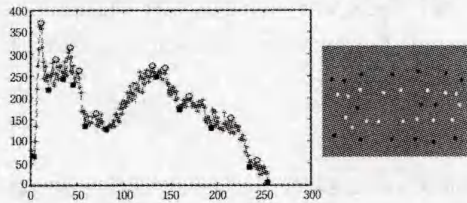


图10 牙齿斑口腔图像的多阈值直方图和自动种子模板

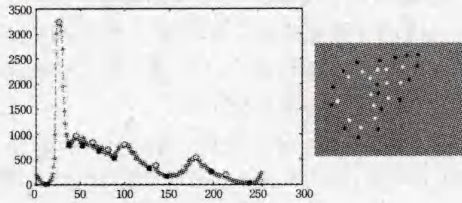


图11 肝脏图像的多阈值直方图和自动种子模板

以上实验结果表明,利用图像的多阈值直方图自动生成

初始种子模板可以达到和手工设置种子模板相一致的效果,但却节省了人工设置种子点的多次交互操作,且利用改进的迭代算法可以达到更好的分割结果,提高了算法的鲁棒性。

结束语 本文针对GrowCut算法中初始种子标记要求用户较多工作量的问题,提出了简化标记、自动生成初始种子模板的多阈值优化细胞自动机图像分割方法。算法结合了图像的像素信息,简化了用户的操作,提高了算法的分割精度,对边界模糊的医学图像分割具有良好的效果。与自动分割方法相比,交互式图像分割可以融合用户的先验知识,具有更高的研究价值和更广的应用领域,不过交互式图像分割的研究目的是通过尽可能简便而直观的人机交互,快速而准确地提取出图像中具有高层语义的前景物体。因此,尽可能使用户通过简明的交互方式对高分辨率图像进行准确分割是未来进一步深化研究的重点。

参考文献

- [1] Mortensen E N, Barrett W A. Interactive segmentation with intelligent scissors[J]. Graphical Models and Image Processing, 1998, 60(5): 349-384
- [2] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models[J]. International Journal of Computer Vision, 1988, 1(4): 321-331
- [3] Osher S, Fedkiw R. Level Set Methods and Dynamic Implicit Surfaces[M]. Springer-Verlag, New York, 2002
- [4] Li Yin, Sun Jian, Tang C K, et al. Lazy Snapping [J]. ACM Transaction on Graphics, 2004, 24(3): 303-308
- [5] Boykov Y, Jolly M P. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images[C]//IEEE International Conference on Computer Vision, 2001: 102-115
- [6] Rother C, Kolmogorov V, Blake A. Grabcut-Interactive Foreground Extraction using Iterated Graph Cuts[J]. ACM Transaction on Graphics, 2004, 24(3): 309-314
- [7] Von Neumann J. Theory of Self-Reproducing Automata [M]. Theory of Self-Reproducing Automate, 1966
- [8] Thomas C D. Evolution of Cellular Automata for Image Processing[D]. University of Birmingham, April 2000
- [9] Vezhnevets V, Konouchine V. GrowCut-interactive multi-label ND image segmentation by cellular automata[C]//Proceeding of Graphicon, 2006: 231-234

(上接第292页)

- [8] Chen L, Zhou J, Wang D, et al. Parallel Image Registration with OpenMP and MPI [C]//International Conference on Computational Intelligence and Software Engineering, 2009: 1-4
- [9] 李锦, 钟诚. 异构机群上高效可扩展的 Motif 发现并行算法[J]. 计算机科学, 2012, 39(3): 279-282
- [10] Deng Y, Manjunath B S, Shin H. Color image segmentation [C]//IEEE Conference on Computer Vision and Pattern Recognition, 1999: 446-451
- [11] Deng Y, Manjunath B S. Unsupervised segmentation of color-texture regions in images and video [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(8): 800-810
- [12] Jing F, Li M, Zhang H J, et al. An efficient and effective region-based image retrieval framework [J]. IEEE Transactions on Image Processing, 2004, 13(5): 699-709
- [13] Adamek T, O'Connor N. Efficient contour-based shape representation and matching [C]//Proc. 5th ACM SIGMM Int. Workshop on Multimedia Information Retrieval, 2003: 138-143
- [14] Sun K, Yang Y Y, et al. Image restoration using piecewise iterative

curve fitting and texture synthesis [C]//International Conference on Intelligent Computing, 2008, 2: 1056-1063

- [15] Chen C H, Wang P S P. Handbook of pattern recognition and computer vision [M]. World Scientific Publishing Co., 1999: 218-222
- [16] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer [C]//ACM SIGGRAPH, 2001: 341-346
- [17] Liang L, Liu C, Xu Y Q, et al. Real-time texture synthesis by patch-based sampling [J]. ACM Transactions on Graphics, 2001, 20(3): 127-150
- [18] Quinn M J. Parallel Programming in C with MPI and OpenMP [M]. McGraw-Hill Companies, Inc., 2004
- [19] 都志辉. 高性能计算并行编程技术—MPI 并行程序设计[M]. 北京:清华大学出版社, 2001
- [20] Guo C L, Ma Q, Zhang L M. Spatio-Temporal Saliency Detection using Phase Spectrum of Quaternion Fourier Transform [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2008: 1-8