

基于测地距离的超像素分析算法

颜玉杰, 刘向阳

(河海大学 理学院, 江苏 南京 211100)

摘要:超像素分析指的是将数字图像细分为多个超像素的过程,旨在简化或改变图像的表示形式,使得图像更容易理解和分析。文章提出了一种基于测地距离的超像素分析算法,该算法采用引入代价函数的 Fast Marching 算法来计算像素点间的测地距离。将目标图像大致均匀地划分成 k 个初始长方形区域,在每个区域内选取局部密度最大的像素点作为种子点,再由种子点出发计算像素点间的测地距离,并根据测地距离对像素点进行标记,故而可以得到大小均衡,形状规整的超像素。该算法在计算测地距离时,充分考虑了像素点的颜色和位置特征,并且以小区域为单位计算测地距离不仅缩小了 Fast Marching 算法的搜索范围,加快了算法的运行速度,还可以使得某些像素点的测地距离被重复计算,便于选取最优值。该算法所得超像素的分割精度及规整度都取得了良好的效果。

关键词:超像素分析;超像素;Fast Marching 算法;局部密度;测地距离

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2022)02-0058-05

doi:10.3969/j.issn.1673-629X.2022.02.009

Superpixel Analysis Algorithm Based on Geodesic Distance

YAN Yu-jie, LIU Xiang-yang

(School of Science, Hohai University, Nanjing 211100, China)

Abstract: Superpixel analysis refers to the process of subdividing a digital image into multiple superpixels, which aims to simplify or change the representation of the image and make the image easier to understand and analyze. We propose a superpixel analysis algorithm based on geodesic distance, which uses Fast Marching algorithm with cost function to calculate the geodesic distance between pixels. The target image is roughly evenly divided into k initial rectangular regions. In each region, the pixel with the highest local density is selected as the seed point, and then the geodesic distance between the pixels is calculated from the seed point, and the pixels are marked according to the geodesic distance, so the size of the balanced and regular shape super-pixel can be obtained. When calculating the geodesic distance, the algorithm fully takes the color and location of pixels into account, and calculates the geodesic distance in a small area, which not only reduces the search scope of Fast Marching algorithm, speeds up the operation speed of the algorithm, but also makes the geodesic distance of some pixels be repeatedly calculated, which is convenient for selecting the optimal value. The segmentation accuracy and regularity of the superpixel obtained by this algorithm is effective.

Key words: superpixel analysis; superpixel; Fast Marching algorithm; local density; geodesic distance

0 引言

超像素是指具有相似纹理、颜色、亮度等特征的相邻像素构成的具有一定视觉意义的不规则像素块。图像分割是按照灰度、频谱、纹理等特性把图像空间划分成一定数量区域的过程,是近年来快速发展的一种图像预处理技术。相比于传统处理方法中的基本单元——像素,超像素更有利于局部特征的提取与结构信息的表达^[1],并且能够大幅度降低后续处理的计算复杂度,在计算机视觉领域尤其是图像分割中得到了广泛的应用。

近年来,很多学者提出了超像素分析算法。超像素生成算法大致可分为基于图论的算法和基于梯度下降的算法两类,具有代表性的基于图论的超像素分析算法有 Shi 等人提出的 Ncut (normalized cuts) 算法^[2-3],利用轮廓特征和纹理特征递归地进行图分割;Liu 等人提出了基于熵率算法,通过最大化目标函数以实现分割;Bergh 等人提出了基于能量驱动的 SEEDS 算法^[4];Chen 等实现了一种基于 NC 的线性谱聚类 (LSC) 算法;Gong 等基于差分进化 (DE) 提出差分进化超像素 (DES) 算法^[5]。对于基于梯度下降法,

收稿日期:2021-02-28

修回日期:2021-06-28

基金项目:国家自然科学基金项目(61001139)

作者简介:颜玉杰(1997-),女,硕士,研究方向为超像素分析;刘向阳,博士,副教授,研究方向为图像与视频分析、数据分析和机器学习。

已有的研究有 Vincent 等人提出的分水岭 (watersheds) 算法^[6], 是一种基于拓扑理论的数学形态学分割算法; Comaniciu 等人提出的 Mean-shift 算法, 是一个迭代模式搜索的过程; Achanta 等人提出的 SLIC (simple linear iterative clustering) 算法^[7], 是基于颜色和距离相似性进行超像素分割, 该算法思想简单, 可以生成大小均匀、形状规则的超像素; Zhao 等提出一种改进 SLIC 搜索策略的超像素快速线性迭代聚类 FLIC 算法^[8]; Ban 等提出一种高斯混合模型 (GMM) 的超像素算法^[9]。其中 SLIC 是目前最为经典、简单的算法, 该算法比现有算法更快, 有着更高的记忆效率, 展示了目前最优的边界贴合度。它是采用 k 均值聚类算法生成超像素, 在图像上大致均匀地选取 k 个像素点, 计算 k 个超像素内所有像素点的平均向量值, 重新得到 k 个聚类中心, 然后不断更新聚类中心, 直到收敛的过程。

该文提出了一种归属于第一类的基于测地距离的超像素分析算法, 该算法是定义初始划分区域内局部密度最大的像素点作为种子点, 然后由种子点出发计算测地距离, 完成超像素划分的过程。和传统算法相比, 该算法的优点在于: 生成超像素的形状紧凑度更高, 并且边界贴合度也更接近于 SLIC 算法。

1 图像测地距离与 Fast Marching 算法

1.1 图像测地距离

n 维空间中, 曲面上任意两点之间测地线的长度称之为测地距离。该文把测地距离应用于图像分析, 就是因为测地距离是计算空间中任意两点的曲面最短距离, 可以更好地描述联通的像素点的局部相似性。具体表达式如下^[10]:

$$D(z_i; M; \nabla I) = \min_{\{z_j | M(z_j=0)\}} d_{z_i z_j}$$

$$d_{z_i z_j} = \min_{\Gamma \in P_{z_i z_j}} \int_0^1 \sqrt{\|\Gamma'(s)\|^2 + \gamma^2 (\nabla I \cdot u)^2} ds \quad (1)$$

式中, I 表示输入图像, $d_{z_i z_j}$ 为空间上任意两点 z_i 、 z_j 的测地线的距离, D 为像素点 z_i 的测地距离, $P_{z_i z_j}$ 表示 z_i 、 z_j 两点间所有路径的集合, M 为二值掩码, $\|\Gamma'(s)\|: \mathfrak{R} \rightarrow \mathfrak{R}^n$ 表示这样的一条路径, 其中参数 $s \in [0, 1]$, $\Gamma'(s) = \partial \Gamma(s) / \partial s$, $u = \Gamma'(s) / \|\Gamma'(s)\|$ 。

测地距离的计算方法主要有两大类, 一类是基于图的方法, 将曲面看作是一个图, 把在图上计算最短距离扩展到计算曲面上的测地距离。另一类是基于样本的方法, 设定这里的曲面是离散可微的, 许多微分几何中的方法可以用来计算曲面上的测地距离。当前流行

的算法主要运用了两类偏微分方程模型:

(1) 程函方程 (双曲型方程):

$$\|\nabla u\| = 1 \quad (2)$$

(2) 热方程 (抛物型方程):

$$\frac{\partial u}{\partial t} = \Delta u \quad (3)$$

其中, 波传播问题可以转化成求解程函方程的问题, 而热扩散问题可以转化成求解热方程的问题。

1.2 Fast Marching 算法

Fast Marching 算法就是一种波传播方法^[11], 旨在通过求解正交网格离散化的程函方程获得测地距离的近似, 具体公式如下:

$$\|\nabla u(z_i, z_s)\| = C \text{ in } \Omega, C > 0$$

$$u = g(z_i, z_s) \text{ on } \Gamma \quad (4)$$

它是一种在矩形正交网格上以 $O(M \log M)$ 步为单位求解椭圆方程的数值算法, 其中 M 是网格点的总数, Ω 是 \mathfrak{R}^n 中的定义域, C 是给定的已知函数, 边界条件是 $u = g(z_i, z_s)$, 此处的边界是域 Ω 的边界, 可以是一个特定的曲线或者曲面。解这个程函方程的中心思想是使用数值方法中的逆向有限差分来近似拟合, 从而得到近似的数值解。

2 基于测地距离的超像素分析算法

2.1 初始化种子点

对于一幅大小为 $M * N$ 的图像 I , 图像中每个像素的颜色在 CIELAB 颜色空间 $[l \ a \ b]^T$ 中表示, 其取值范围是已知的, 而像素的位置 $[x \ y]^T$ 的取值范围随着图像的尺寸变化而变化, 所以每个像素点的五维向量表示为 $[x \ y \ l \ a \ b]^T$ 。如果简单定义像素点间的距离为 $xylab$ 空间中的五维欧氏距离将导致不同超像素大小的聚类行为不一致。对于较大的超像素, 空间距离远超过颜色距离, 因此空间距离会比颜色距离更重要, 这样会产生紧凑的超像素, 而对于较小的超像素, 情况刚好相反。为了权衡这两个距离, 该文添加了一个颜色权重 ω 。

计算初始区域内每个像素点的局部密度, 将局部密度最大值 ρ_{\max} 对应的像素点作为种子点, 目的是为了选取相对比较大的平滑区域的中心。2014 年 Alex Rodriguez 在新聚类算法 (DP 算法) 中提出了局部密度概念^[12]。对于局部密度这个指标的计算, 首先要计算出像素点集 $z = \{z_1, z_2, \dots, z_{M*N}\}$ 中任意两点间的欧氏距离:

$$d_{z_i z_j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \omega^2 * [(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2]} \quad (5)$$

再将像素点的局部密度定义为“数据集中到该点距离小于截断距离 d_c 的数据点个数”。该文采用高斯

核函数计算局部密度, 这样计算不同的数据点具有相同的局部密度值的概率会更小。即:

$$\rho_{z_i} = \sum_{j \neq i} \exp(-(d_{z_i}/d_c)^2) \quad (6)$$

其中, d_c 为截断距离, 是算法中的可变参数。

2.2 图像测地距离计算

由上述步骤选取的种子点出发, 该文基于引入代价函数的 Fast Marching 算法计算搜索范围内像素点间的测地距离。在处理一幅大图像时, Fast Marching 算法由多个初始点出发计算测地距离的时间代价太大, 因此该文设计了一种新的算法, 不再是从整体上计算一个图像多个起始点的测地距离, 而是缩小了每个种子点的搜索范围来计算测地距离, 最后将得到的距离矩阵进行整合。

2.2.1 算法代价函数

在一个二维规划空间中, 两点之间的最优路径满足最小代价要求, 即 $\min \int_s^i C dh$, 其中单位代价场 C 是一个各项异性场, 它与位置和方向都有关。假定有势函数 $u(z_i, z_s)$, z_s 表示种子点, $z_i (i \neq s)$ 表示其余像素点, 它满足 Eikonal 方程: $\|\nabla u(z_i, z_s)\| = C, C > 0$ 的最陡梯度下降轮廓线就是最优路径, 边界条件 $u = g(z_i, z_s) = 0$ 作为默认已知。考虑当 $C = 1$ 时, 方程的解就是起点到目标点的欧氏距离, 它可以简单地通过构造具有单位速度的扩展边界曲线得到。

设定在二维平面上的曲线是其法线方向向上的速度 v , 如果曲线经过种子点 z_s 的时间为 T , 那么在路径规划中最小代价求解问题的 Eikonal 方程通常写成:

$$\|T(z_i, z_s)\| v(z_i, z_s) = 1 \quad (7)$$

其中, $T(z_i, z_s)$ 为时间距离函数, 为拓展代价函数, 该算法定义的代价函数为:

$$\|T(z_i, z_s)\| = \frac{1}{v(z_i, z_s)} = d_{z_i} \quad (8)$$

通过上述方程再结合式(5)可以求解时间距离函数, 进而可通过 Fast Marching 算法求得种子点与其余像素点之间的测地距离。

2.2.2 测地距离的计算

对于任意一个种子节点 z_s , 利用 Fast Marching 算法计算像素点间的测地距离^[13]。初始化源点距离为 0, 其他顶点距离为 $\text{Inf}; d(z_s) = 0, d(z_l) = \text{Inf} (l \neq s)$, 设置源点状态为 Open, 其他顶点状态为 Far, 查找集合 Open 中距离值最小的顶点 z_i , 将其状态更新为 Dead, 将顶点 z_i 相邻状态为 Far 的顶点状态更新为 Open, 然后开始进行循环迭代, 查找顶点 z_i 相邻状态为 Open 的顶点 z_j , 遍历顶点 z_j 的一环邻域三角面片, 利用 Eikonal 方程计算顶点 z_j 的距离值 d , 记录 d_{\min} , 通过循环不断地更新顶点 z_j 的距离值 $d(z_j) = \min\{d(z_j), d_{\min}\}$, 重复上述过程, 就可以计算出每个种子节点 z_s 的测地距离 D_{z_s} 。

种子节点 z_s 的搜索范围变小, 会导致有些像素点的测地距离被重复计算, 对于某个像素点 z_a 的测地距离 D_1, D_2, \dots, D_q , 定义该像素点的测地距离为 $\min\{D_1, D_2, \dots, D_q\}$, (q 为该像素点被重复计算的次数)。如此, 就可以得到与原图像同等大小的 $M \times N$ 的距离矩阵。

2.3 超像素划分

本节主要提出了一种基于图标签的标记方法, 旨在通过此算法, 将已标记簇标签的局部密度最大的像素点的标签传递给所有剩余像素点, 迭代此过程直到所有像素点拥有相应的标签为止, 形成最终的超像素划分。使用该算法可以动态地将局部密度最大的像素点同时对剩余未分配标签的像素点进行标记。记种子点 z_s 的八邻域内的像素点集为 $\{z_{s_1}, z_{s_2}, \dots, z_{s_8}\}$, 从中找到未标记的且与种子点的测地距离最小的像素点 z_{s_i} , 那么 z_s 的标签就赋给像素点 z_{s_i} , 具体公式如下:

$$L(z_{s_i}) = \begin{cases} L(z_s), & \text{若种子点 } z_s \text{ 未标记且满足条件 } H \\ 0, & \text{其他} \end{cases} \quad (9)$$

其中, $L(z_s)$ 表示种子点 z_s 的标签, 约束条件为 $H: |D_{z_{s_i}} - D_{z_s}| < |D_{z_{s_i}} - D_{z_e}|, (r, e \in [1, 8], r \neq e)$, 其中 $|D_{z_{s_i}} - D_{z_e}|$ 表示像素点 z_{s_i}, z_s 之间测地距离差的绝对值^[14], 该公式可以判定当像素点自身已有簇标签的时候可以将簇标签 $L(z_s)$ 传递至未标记的像素点 $L(z_{s_i})$, 如此完成一次标记^[15]。重复上述步骤, 直到每个像素点都有对应的标签, 即完成超像素的划分。

2.4 流程图

该算法主要分为以下几个步骤进行, 首先是基于指定的 k 值将图像划分成大致均匀的长方形区域; 然后计算每个区域像素点的局部密度, 将局部密度最大的像素点作为种子点; 再从种子点出发, 计算像素点间的测地距离; 最后根据测地距离, 找到像素点的标签, 完成超像素划分。算法流程如图 1 所示。

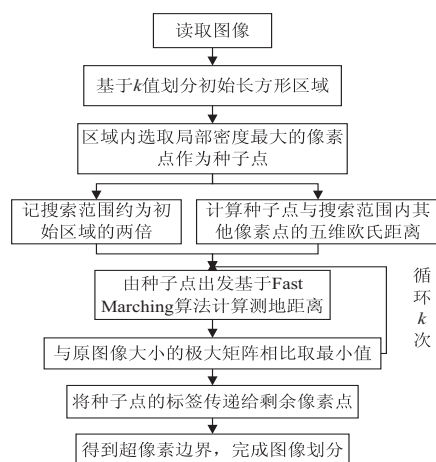


图 1 算法流程

3 实验结果及分析

为了验证文中超像素分析算法的性能,在 Berkeley Benchmark 标准数据集上选取 10 张有代表性的图片进行了对比实验,验证算法包括文中算法、SLIC^[7]、SEEDS^[4]、Watershed^[6]等。评价标准包括 ASA、SRC 等。实验中取数据集中所有两点间距离值的 1%~2%。

3.1 评价标准

利用可达分割精度(ASA)参数计算超像素分割算法的效率,并将超像素与地面真值分割的标签进行比较。ASA 参数定义为^[16]:

$$ASA = \frac{\sum \max |Z_k \cap G_v|}{\sum |G_v|} \quad (10)$$

考虑到超像素边界与人工标记边界存在误差,在人工标记边界的 24 邻域寻找超像素的标签。另外,紧密度是衡量超像素的重要评价指标,Giraud 在 2017 年

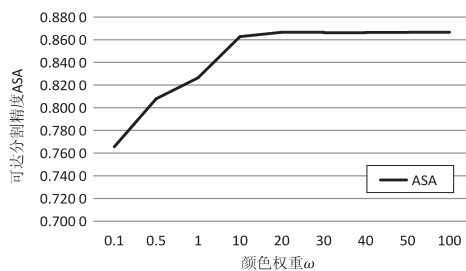
提出新的形状规律准则(SRC)以描述超像素的形状紧凑度^[17]。SRC 考虑了形状凸性、平衡再分配和轮廓光滑三种形状规则属性。形状规律性标准定义如下:

$$SRC = \sum_{Z_k} \frac{|Z_k|}{|I|} CR(Z_k) V_{xy}(Z_k) \quad (11)$$

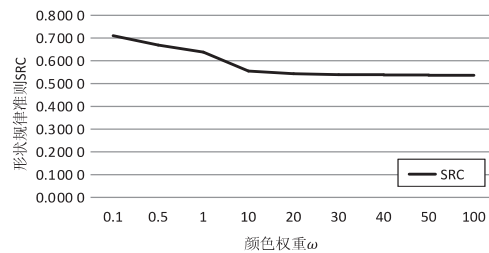
式中, $Z = \{Z_1, Z_2, \dots, Z_k\}$, Z_k 为第 k 个超像素, $CR(Z_k)$ 为圆形凸包覆盖率, $V_{xy}(Z_k)$ 为最小位置与最大位置方差比值。

3.2 算法参数分析

文中算法需设置的参数有超像素数量 k 和像素点的颜色权重 ω 。通过调整参数可以使得分割结果更加符合预期,对于每幅图的评价结果取算数平均数作为该算法在某超像素数量下的最终评价。为了便于评价,把超像素数量设置为 $k = 1\,000$,观察可达分割精度 ASA、形状规律准则 SRC 在 ω 改变时所发生的变化(见图 2)。



(a)不同 ω 下算法的 ASA 比较



(b)不同 ω 下算法的 SRC 比较

图 2 参数 ω 调整结果

由实验结果可知,当颜色权重 ω 设置过小时,可达分割精度 ASA 的值较低,会导致超像素的边界与图像边界贴合度不高,而超像素规整度 SRC 较高,为了权衡这两者,设置 $\omega = 0.5$,超像素分割的结果会更好。

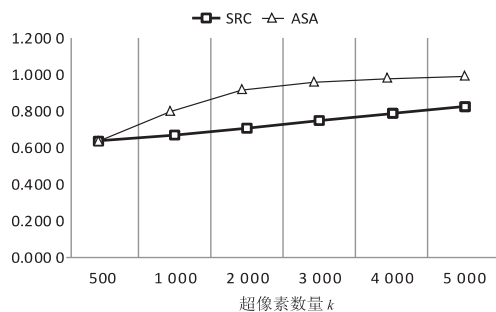


图 3 参数 k 调整结果

由图 3 可知,当超像素数量 k 设置的较大时,超像素的紧凑度与精确度都可以达到一个较高的值,对于处理更大的图像时,文中算法会取得一个更优的结果。

3.3 实验结果比较与分析

为了便于评价,超像素数量值统一设置为 1 000、2 000、3 000、4 000、5 000。对所有的分割结果,分别利用上述两种评价标准进行评价,对每幅图的评价标准取算数平均数作为该算法在某超像素数量下的最终评

价值。图 4、表 1 分别为 ASA、SRC 指标的比较结果。图 5 为 SLIC^[7]、SEEDS^[4]、Watershed^[6]和文中算法以 $k = 1\,000$ 、 $k = 3\,000$ 为示例划分图像的结果。

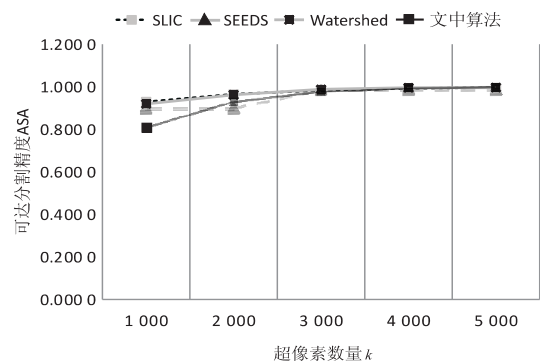


图 4 不同超像素算法的 ASA 对比

表 1 不同超像素算法的 SRC 对比

k	SLIC	SEEDS	Watershed	文中算法
1 000	0.447 3	0.381 0	0.382 9	0.669 0
2 000	0.578 7	0.381 0	0.438 8	0.706 4
3 000	0.685 6	0.531 5	0.459 0	0.748 5
4 000	0.755 9	0.531 5	0.489 3	0.785 9
5 000	0.803 2	0.531 5	0.521 2	0.819 1

由上述图表可以看出,文中算法生成的超像素规

整度明显优于其他算法,而可达分割精度 ASA 也与其他算法相接近。综合上述,文中算法生成的超像素整体效果较好,预计在处理更大的图像时,会取得一个更好的效果。

由图 5 的划分结果可以看出,文中算法与 SLIC 算法得到的超像素形状更为规整,对于图像中不平滑的区域,SLIC 算法所得超像素不能保持更好的形状规律准则,而文中算法却在保持分割精度的同时,使得所得超像素的规整度更优。

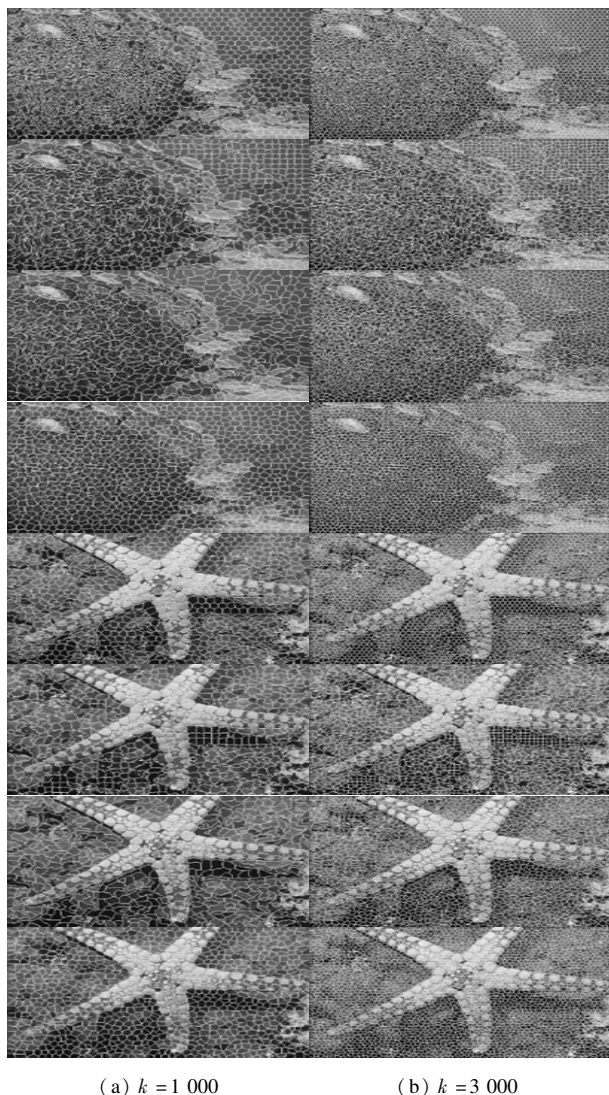


图 5 SLIC、Watershed、SEEDS、文中算法超像素分析结果(从上至下)

4 结束语

该文提出了一种基于测地距离的超像素分析算法,大量实验结果表明,算法的分割精度较好,当预期分割数较大时其 ASA 值优于 SLIC 算法,并且该算法生成的超像素形状规整度更好。综合来看,提出的基于测地距离的超像素分析算法的划分结果较好,且能稳定生成形状规整的超像素。接下来会进一步优化该

算法,加快算法的运行速度以及提升超像素的分割精度。

参考文献:

- [1] 张小凤,刘向阳. 基于图像超像素分析的图像分割方法[J]. 计算机技术与发展,2018,28(7):25-28.
- [2] SHI Jianbo, MALIK J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8):888-905.
- [3] 宋熙煜,周利莉,李中国,等. 图像分割中的超像素方法研究综述[J]. 中国图象图形学报,2015,20(5):599-608.
- [4] BERGH M V D, BOIX X, ROIG G, et al. SEEDS: superpixels extracted via energy-driven sampling[C]//Computer vision-ECCV 2012. Florence, Italy: Springer, 2012:13-26.
- [5] 罗学刚,吕俊瑞,彭真明. 超像素分割及评价的最新研究进展[J]. 激光与光电子学进展,2019,56(9):090005.
- [6] VINCENT L, SOILLE P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1991, 13(6):583-598.
- [7] ACHANTA R, SHAJI A, SMITH K, et al. SLIC superpixels compared to state-of-the-art superpixel methods[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(11):2274-2282.
- [8] ZHAO J, BO R, HOU Q, et al. FLIC: fast linear iterative clustering with active search[J]. Computational Visual Media, 2018, 4(4):333-348.
- [9] BAN Z H, LIU J G, CAO L. Superpixel segmentation using Gaussian mixture model[J]. IEEE Transactions on Image Processing, 2018, 27(8):4105-4117.
- [10] CRIMINISI A, SHARP T, BLAKE A. GeoS: geodesic image segmentation[J]. Lecture Notes in Computer Science, 2008, 5302(1):99-112.
- [11] YATZIV L, BARTESAGHI A, SAPIRO G. O(N) implementation of the fast marching algorithm[J]. Journal of Computational Physics, 2005, 212(2):393-399.
- [12] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191):1492-1496.
- [13] 于 晖,王永骥. 基于 Fast Marching 方法的多目标点路径规划的研究[J]. 计算技术与自动化,2015,10(3):11-15.
- [14] 谭乐怡,王守觉. 基于双重超像素集的快速路径相似度图像分割算法[J]. 自动化学报,2013,39(10):1653-1664.
- [15] 邹焕新,李美霖,曹 旭,等. 一种基于测地线距离的极化 SAR 图像快速超像素分割算法[J]. 雷达学报,2021,10(1):20-34.
- [16] 雷 涛,连 倩,加小红,等. 基于快速 SLIC 的图像超像素算法[J]. 计算机科学,2020,47(2):149-155.
- [17] GIRAUD R, TA V T, PAPADAKIS N. Evaluation framework of superpixel methods with a global regularity measure[J]. Journal of Electronic Imaging, 2017, 26(6):061603. 1-061603. 18.