

基于改进密度峰值聚类算法的图像分割

张力丹,王军锋

(西安理工大学 理学院,陕西 西安 710054)

摘要:聚类算法作为数据处理的一种技术,发展迅速且被广泛应用于各个领域。密度峰值聚类算法(clustering by fast search and find of density peaks,DPC)作为一种基于密度的聚类方法,可通过高效样本分配进行图像分割。然而DPC算法的聚类结果依赖于参数截断距离 d_c 的选取,为此提出一种基于信息熵的DPC算法以实现 d_c 的自适应选取。信息熵反映的信息大小与随机事件的概率呈负相关,随机事件发生的概率越大,提供的信息反而越少。因此信息熵可以体现出事件的不确定性,故可将使得信息熵最小的 d_c 作为DPC算法的最优参数。另外聚类算法的簇类个数 K 普遍难以确定,而DPC算法中的簇类中心通常由局部密度极大值点构成,改进算法则根据数字图像各区域内聚程度自适应确定 K 的选择阈值。为将DPC算法高效应用于图像分割,改进算法通过分块与合并的方式解决DPC算法时间复杂度较大的问题。经实验对比,改进算法具有更精确的聚类效果;在图像分割方面,改进算法能够更为精准地提取图像分割边缘并与GroundTruth更加吻合。

关键词:密度峰值聚类;图像分割;簇类合并;块处理;自适应截断距离

中图分类号:TP110.87

文献标识码:A

文章编号:1673-629X(2022)05-0047-06

doi:10.3969/j.issn.1673-629X.2022.05.008

Image Segmentation Based on Improved Density Peak Clustering Algorithm

ZHANG Li-dan, WANG Jun-feng

(School of Science, Xi'an University of Technology, Xi'an 710054, China)

Abstract: As a technology of data processing, clustering algorithm develops rapidly and is widely used in various fields. Clustering by fast search and find of density peaks (DPC) is a density-based clustering method that can perform image segmentation through efficient sample allocation. However, the clustering result of the DPC algorithm depends on the selection of the parameter cutoff distance d_c . For this reason, a DPC algorithm based on information entropy is proposed to realize the adaptive selection of d_c . The size of information reflected by information entropy is negatively related to the probability of random events. The greater the probability of random events, the less information provided. Therefore, the information entropy can reflect the uncertainty of the event, so the d_c that minimizes the information entropy can be used as the optimal parameter of the DPC algorithm. In addition, the number of clusters K of the clustering algorithm is generally difficult to determine, and the cluster center in the DPC algorithm is usually composed of local density maximum points. The improved algorithm adaptively determines the selection threshold of K according to the degree of cohesion of each area of the digital image. In order to efficiently apply the DPC algorithm to image segmentation, the improved algorithm solves the problem of large time complexity of the DPC algorithm by means of block and merge. After experimental comparison, the improved algorithm has a more accurate clustering effect. In terms of image segmentation, it can more accurately extract the edge of image segmentation and is more consistent with GroundTruth.

Key words: density peak clustering; image segmentation; cluster merging; block processing; adaptive cutoff distance

0 引言

聚类作为一种无监督的分类方法,能够用于无标签的数据对象的划分^[1]。作为数据处理的技术,聚类分析已经广泛应用于多个领域,例如计算机科学、数学、经济学等。聚类可分为划分聚类、层次聚类、基于

密度的聚类 and 基于网格的聚类等。K-means 作为最为经典且具代表性的划分聚类,需在给定簇类个数的前提下,经过反复迭代获取最优分组,但是该方法很难用于非球形聚类数据的无监督划分^[2]。FCM 聚类算法作为软聚类法的代表之一,需要构造各数据点归属

收稿日期:2021-05-11

修回日期:2021-09-13

基金项目:国家自然科学基金面上项目(61976176)

作者简介:张力丹(1996-),女,硕士研究生,研究方向为图像处理;王军锋,博士,副教授,研究方向为偏微分方程在图像中的应用。

于某一簇类的隶属度函数,而同一数据点可同时归属至不同的簇类中^[3]。DBSCAN 算法在限制 Eps 邻域及最小核心点 MinPts 下,通过建立了密度可达及直接密度可达一套机制将数据进行归类,但很难发现密度变化差异明显的簇类^[4]。

2014 年 Alex Rodriguez 发表于《Science》上的密度峰值聚类(DPC)算法^[5],其核心思想在于对簇类中心的刻画。该算法认为簇类中心同时具有两个特点:一是具有较大的密度,即簇类中心被密度较小的数据点包围;二是与其他局部密度更高的点的距离较大。该文将对 DPC 算法中的敏感参数截断距离 d_c 进行自适应选取,并将改进的 DPC 算法应用于图像分割。

1 密度峰值聚类算法

DPC 算法在含 n 个数据点的数据集中定义了各点 $i(i=1,2,\dots,n)$ 的局部密度 ρ_i 和局部距离 δ_i 。局部密度 ρ_i 的计算公式如下:

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \quad (1)$$

$$\chi(d_{ij} - d_c) = \begin{cases} 1, & d_{ij} < d_c \\ 0, & d_{ij} \geq d_c \end{cases} \quad (2)$$

式中, d_{ij} 为各数据点间的相似度; d_c 为截断距离,本质为簇类中心的邻域半径。上述公式表明,各数据点的局部密度为 d_c 邻域内所含数据点个数^[6]。截断距离 d_c 的选取基于各数据点的相似度,取值为升序排列的第 c 个数据点间的相似度 d_{ij} , c 的计算公式为:

$$c = \lfloor N \times P + 0.5 \rfloor \quad (3)$$

式中, N 为数据点间相似度总数; P 为截断距离 d_c 的取值阈值。因此,截断距离 d_c 仍需依赖人为选取的阈值 P 。各数据点的局部距离 δ_i 刻画了各数据点与密度更高点之间的距离,具体计算公式如下:

$$\delta_i = \begin{cases} \min(d_{ij}), & \rho_j < \max(\rho) \\ \max_j(d_{ij}), & \text{else} \end{cases} \quad (4)$$

上式表明,对于局部密度最大的数据点,其局部距离为该点与其他数据点间相似度的最大值^[7];其余数据点的局部距离则为该点与其他数据点间相似度的最小值。计算簇类中心的选择阈值 γ_i :

$$\gamma_i = f(\delta_i, \rho_i) \quad (5)$$

上式表明,阈值 γ_i 为关于局部密度 ρ_i 和局部距离 δ_i 的函数。文献[4]将前 K 个降序排列的 γ_i 所对应的数据点作为簇类中心。

2 自适应截断距离 d_c 的 DPC 算法

DPC 算法在计算局部密度 ρ_i 时采用基于 ε 近邻的方式,将数据点 x_i 的 d_c 邻域内包含的数据点个数作为该点局部密度。 d_c 的选取依赖于人为确定的阈值

P , P 按经验取值为 1% ~ 2%。DPC 算法核心在于计算各数据点的局部密度 ρ_i 和局部距离 δ_i ,并将局部密度 ρ_i 和局部距离 δ_i 的局部极值点作为局部峰值点,也就是簇类中心。其中局部距离 δ_i 的计算依赖于局部密度 ρ_i ,而局部密度 ρ_i 的计算又依赖于截断距离 d_c ,因此 d_c 的选取对密度峰值聚类结果的影响尤为重要^[8-10]。

然而不同的数据内部稀疏程度不一致,若仅靠经验值选取截断距离 d_c 势必导致错误归类。阈值 P 过大会导致截断距离 d_c 取值过大,容易将本不属于同一簇类的像素点合并;阈值 P 过小又会导致截断距离 d_c 取值过小,容易将本属于同一簇类的像素点分成多个簇类,因此基于阈值 P 的截断距离 d_c 并不可靠且影响了局部密度 ρ_i 的全局可靠性^[11]。

2.1 基于最小化信息熵的截断距离 d_c

DPC 算法中定义的局部密度 ρ_i 依赖于截断距离 d_c 以及数据点间的相似度 d_{ij} 。实际中,数据集在不同维度上的分量通常不在一个数量级上^[12-13],为此该文重新定义数据点间的相似度 d_{ij} 。计算数据集 X 的第 d 维分量的极差 A_d :

$$A_d = \max(I_d) - \min(I_d) \quad (6)$$

将数据点 x_i 和 x_j 间的相似度 d_{ij} 定义为:

$$d_{ij} = \frac{\sum_{d=1}^D \left(\frac{i_d}{A_d} \frac{j_d}{A_d} \right)^2}{\sqrt{\left(\sum_{d=1}^D \frac{i_d}{A_d} \right)^2 \left(\sum_{d=1}^D \frac{j_d}{A_d} \right)^2}} \quad (7)$$

式中, D 为数据维数。计算当前数据点 x_i 的 d_c 邻域内相似度均值 μd_{c-i} :

$$\begin{cases} \mu d_{c-i} = \frac{1}{N_{-i}} \sum_{j: d_{ij} \leq d_c} d_{pq} \\ N_{-i} = \sum_j \chi(d_{ij} - d_c) \\ \chi(d_{ij} - d_c) = \begin{cases} 1, & d_{ij} < d_c \\ 0, & d_{ij} \geq d_c \end{cases} \end{cases} \quad (8)$$

式中, p 和 q 为当前数据点 x_i 的 d_c 邻域内所含数据点; N_{-i} 为该点 d_c 邻域内所含数据点的个数。设数据集含有 n 个数据点,引入高斯核将当前数据点的局部密度 ρ_i 定义为:

$$\rho_i = \sum_{j=1}^n e^{-\frac{1}{2} \frac{(d_{ij} - \mu d_{c-i})^2}{(d_c - \mu d_{c-i})^2}}, \quad j \in \{j | d_{ij} < d_c\} \quad (9)$$

改进的局部密度增加了数据点 x_i 的 d_c 邻域内所含数据点的影响权值,使得同一簇类当中的数据点相似程度更紧密。各数据点的概率密度定义为:

$$P(x_i) = \frac{\rho_i}{Z}, \quad Z = \sum_{i=1}^n \rho_i \quad (10)$$

对于一个包含 n 个随机变量 x_1, x_2, \dots, x_n 的系统,

其信息熵为:

$$H(X) = - \sum_{i=1}^n P(x_i) \log(P(x_i)) \quad (11)$$

则基于公式(10)得到的数据集信息熵为:

$$H(X) = - \sum_{i=1}^n \frac{P_i}{Z} \log\left(\frac{P_i}{Z}\right) \quad (12)$$

由上式可知,改进方法后的信息熵中唯一变量为截断距离 d_c 。随机事件发生的概率越大提供的信息越少,信息熵体现出事件的不确定性。因此作为一种量化指标,信息熵越小对应的系统越稳定^[14-15]。

该文将使得信息熵最小的 d_c 值作为截断距离 d_c 的最优取值,具体如下:

$$d_c = \arg \min H(X) \quad (13)$$

2.2 基于改进 DPC 算法在图像分割上的应用

在 DPC 算法中,若局部密度 ρ_i 和局部距离 δ_i 越大,则对应的数据点 x_i 越有可能成为簇类中心。DPC 算法通过公式(5)将降序排列的前 K 个 γ_i 所对应的数据点作为簇类中心,这种选取方式本质上仍需人为选择。为了自适应地选择图像的簇类中心,定义簇类中心点选取的阈值 γ_{\min} 为:

$$\gamma_{\min} = E(\rho) \times E(\delta) \quad (14)$$

式中, $E(\rho)$ 和 $E(\delta)$ 分别为数据点的局部密度均值和局部距离均值。若当前数据点为簇类中心,则该数据点的局部密度 ρ_i 和局部距离 δ_i 应远大于其余数据点。将符合公式(14)的数据点作为初始的簇类中心。

计算各数据点的 ρ_i 和 δ_i 的乘积,并记作 γ_i :

$$\gamma_i = \rho_i \times \delta_i \quad (15)$$

在确定阈值之后,对初始判断的簇类中心进行深度筛选,最终确定的簇类中心集合 C 为:

$$C = \{c_i | \gamma_i > \gamma_{\min}, \delta_i > d_c, c_i \in X\} \quad (16)$$

上式表明, d_c 作为一个特殊的邻域半径,应当小于簇类中心之间的距离。因此,选择 γ_i 大于阈值 γ_{\min} 且 δ_i 大于 d_c 的对应点作为簇类中心。

由于 DPC 算法需要计算数据点之间的距离,以此来判断数据点之间的相似程度,因此对于一般图像而言,将像素点作为输入数据进行密度峰值聚类会耗时很久。为解决 DPC 聚类在处理较大数据集用时较久这一问题,采用并行分块处理图像数据的方式。将一般图像划分成 $M \times N$ 个块,对每一块依次使用改进的密度峰值聚类进行分割,将每一块的像素点作为输入数据,得到每一块的聚类中心(共计 k 个)。在此基础上,使用提出的改进的密度峰值聚类,重新对 $M \times N$ 个块中的 k 个聚类中心进行二次聚类,此次聚类的输入数据为 k 个聚类中心,输出数据为最终的 k 个聚类中心。经过两次聚类之后得到最终需要的聚类中心,最后对所有数据按照相似度进行归类,得到最终的分割结果。

基于改进 DPC 算法的图像分割流程如下所示。

算法 1: 基于改进算法的图像分割。

输入: 图像 I

将图像进行分块处理,共计 $M \times N$ 个块。

While $i \leq$ 最大迭代次数 $M \times N$

Run 第一次调用改进的 DPC 算法:

提取像素点特征: X, Y, R, G, B, LBP (分别为位置特征 X, Y , 颜色特征 R, G, B , LBP 纹理特征);

利用公式(7)计算像素点间的相似度 d_{ij} ;

利用公式(13)计算截断距离 d_c ;

使用公式(9)和公式(4)分别计算局部密度 ρ_i 、计算局部距离 δ_i ; 使用公式(14)计算簇类中心选择阈值 γ_{\min} ; 并使用公式(16)选取簇类中心;

遍历归类剩余像素点;

将属于同一类的像素点进行合并;

输出当前块的聚类结果;

End

End

二次调用改进的 DPC 算法,将第一次聚类中属于同一类的区域进行合并;

输出最终分割结果。

3 实验结果与分析

3.1 实验环境

对多个彩色数字图像数据集使用文中算法进行分割,并将文中算法与 K-means、FCM、DPC 算法和文献[5]中的算法进行对比,验证改进算法具有较高的分割精度,实验硬件为 i5-6500 CPU @ 3.20 GHz, RAM 4.00 GB, 64 位操作系统,实验软件为 Matlab R2016b。

3.2 图像分割评价指标

3.2.1 概率兰德指数(PRI)

设图像共包含 N 个像素点, S_g 为 GroundTruth, 使用不同算法得到的分割结果记为 S_{test} 。当 S_g 的第 i 个像素与 S_{test} 中第 j 个像素具有相同标签时,将该事件记作 A_{ij} , 该事件发生的概率为 p_{ij} 。PRI 指标取值范围为 $[0, 1]$, 数值越大算法分割精度越高。PRI 指标具体计算方式为:

$$\text{PRI}(S_{\text{test}}, S_g) = \frac{1}{N^2} \sum [A_{ij} p_{ij} + (1 - A_{ij})(1 - p_{ij})] \quad (17)$$

3.2.2 变换信息量指数(VOI)

计算分割结果 S_{test} 和 GroundTruth 图像 S_g 信息熵之和,并减去二者的联合信息熵,将计算结果作为 VOI 指标值。该指数代表相对变化的信息量,VOI 越小实验精度越高,且越接近 GroundTruth。VOI 指数具体计算方式为:

$$VOI(S_g, S_{\text{test}}) = H(S_g) + H(S_{\text{test}}) - 2H(S_g, S_{\text{test}}) \quad (18)$$

3.2.3 欠分割误差 (USE)

所谓的“欠分割”是指前景目标错误地划分到背景中,反之“过分割”提取到的前景中包含本属于背景中的某些区域。USE 指标能够衡量图像中的分割精度,对于包含 N 个像素点的数字图像,其在不同算法下得到的分割结果 S_{test} 与 S_g 分别由 nb 和 nb_{st} 个不相交的区域构成:

$$\{S_{\text{test}} \mid S_{\text{test}} = \bigcup_{i=1}^{nb} B_{i1} \cap B_{i2} \neq \emptyset \text{ 且 } i_1 \neq i_2\} \quad (19)$$

$$\{S_g \mid S_g = \bigcup_{i=1}^{nb_{\text{st}}} B_{i1}^{\text{st}} \cap B_{i2}^{\text{st}} \neq \emptyset \text{ 且 } i_1 \neq i_2\} \quad (20)$$

式中, B_i^{test} 和 B_j^g 分别为分割结果与 GroundTruth 的第 i 个和第 j 个非空区域,且各自区域包含的像素点个数为 npB_i^{test} 和 npB_j^g 。则 USE 指标计算方式为:

$$\text{USE} = \frac{1}{N} \left[\sum_{i=1}^{nb} \left(\sum_{npB_j^g: npB_i^{\text{test}} \cap npB_j^g > B} |npB_j^g| - N \right) \right] \quad (21)$$

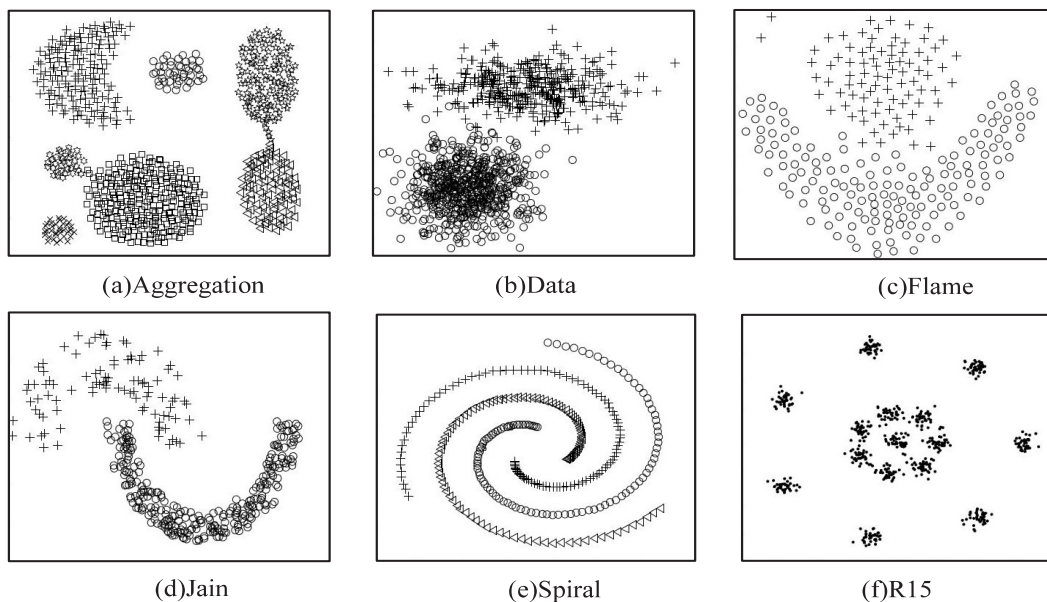


图 1 原始数据集分布

(2) 二维数据聚类效果。

为验证改进算法的有效性,分别使用 DPC 算法和改进算法进行实验,对比结果如图 2 所示。图 2 第一列为 DPC 算法结果,第二列为改进的 DPC 算法结果。其中将 DPC 算法中的截断距离阈值 P 均设置为 0.01,簇类个数与真实数据保持一致。改进算法的截断距离 d_c 均可自适应取得,簇类个数与真实数据保持一致。观察可得除了第四组数据集 Jain 以外,改进算法的聚类结果与原始分布更加吻合。

为了进一步验证文中算法的可靠性,选用精确度 (ACC)、F 测度 (FM) 和兰德系数 (RI) 三个有效性指标对算法结果进行量化分析,三个指标值越大,聚类效

果越好。表 2 给出了 6 个数据集在 DPC 算法和文中改进算法下的指标值,综合比较可知改进算法有效性指标值更佳。

3.3 结果分析

3.3.1 二维数据聚类效果

(1) 二维数据集。

选用 6 个不同数据集进行实验分析,表 1 给出了数据集的样本数和真实类别数。图 1 为原始数据分布。

表 1 不同数据集信息

数据集	样本数	维度	真实类数
Aggregation	788	2	7
Data	1 000	2	2
Flame	240	2	2
Jain	373	2	2
Spiral	312	2	3
R15	600	2	15

果越好。表 2 给出了 6 个数据集在 DPC 算法和文中改进算法下的指标值,综合比较可知改进算法有效性指标值更佳。

表 2 不同数据的有效性指标

数据集	DPC			改进算法		
	ACC	FM	RI	ACC	FM	RI
Aggregation	0.73	0.76	0.89	0.99	0.99	0.99
Data	0.99	0.98	0.98	1	1	1
Flame	0.78	0.67	0.66	1	1	1
Jain	0.92	0.87	0.85	0.86	0.79	0.75
Spiral	0.87	0.78	0.85	1	1	1
R15	0.99	0.98	0.99	0.99	0.99	0.99

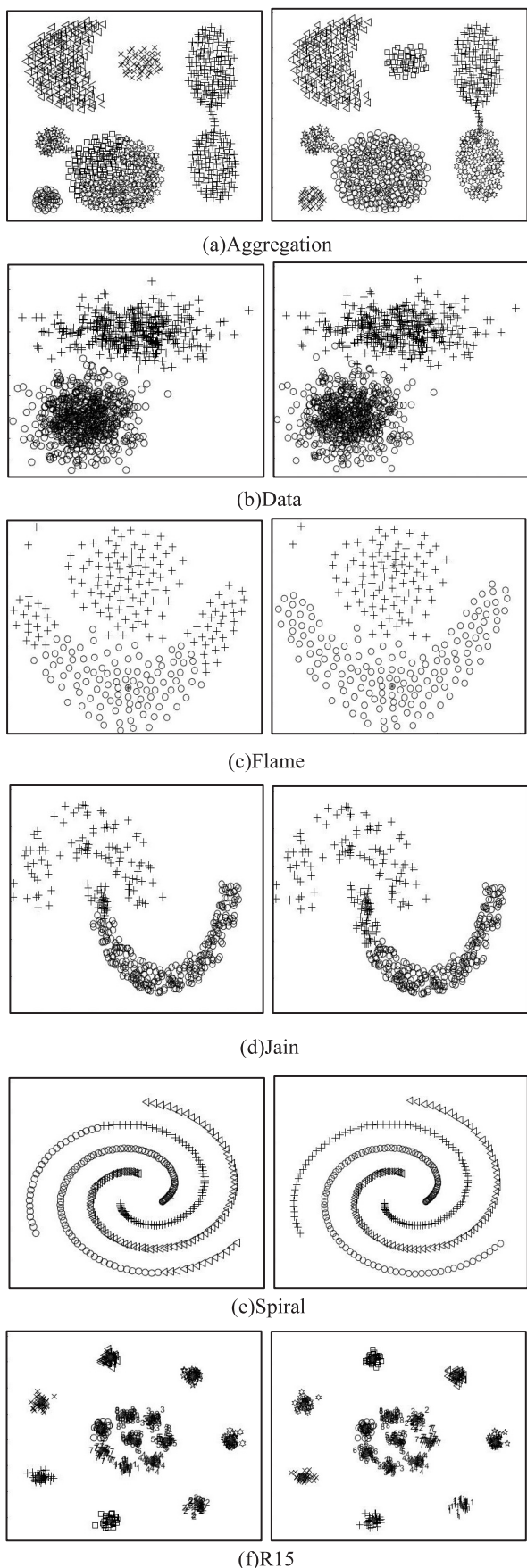


图2 原始数据集分布

3.3.2 数字图像分割效果

为验证文中算法能够较好地应用于图像分割,使

用改进算法对数字图像数据集进行分割。将文中算法与 K-means、FCM、DPC 和文献[5]中的算法进行对比,验证改进算法具有较高的分割精度,对比算法的分割结果如图3所示。K-means、FCM、DPC 和文献[5]中的算法需要确定簇类个数,且运行时间会随着类簇个数的增加而增加,而文中算法平均耗时为8秒,无论分割的簇类数有多大,运行时间基本不会发生改变,且文中算法的分割结果同 GroundTruth 更贴近,更能够清晰地分割图像边缘轮廓,将感兴趣的目标同背景分离。

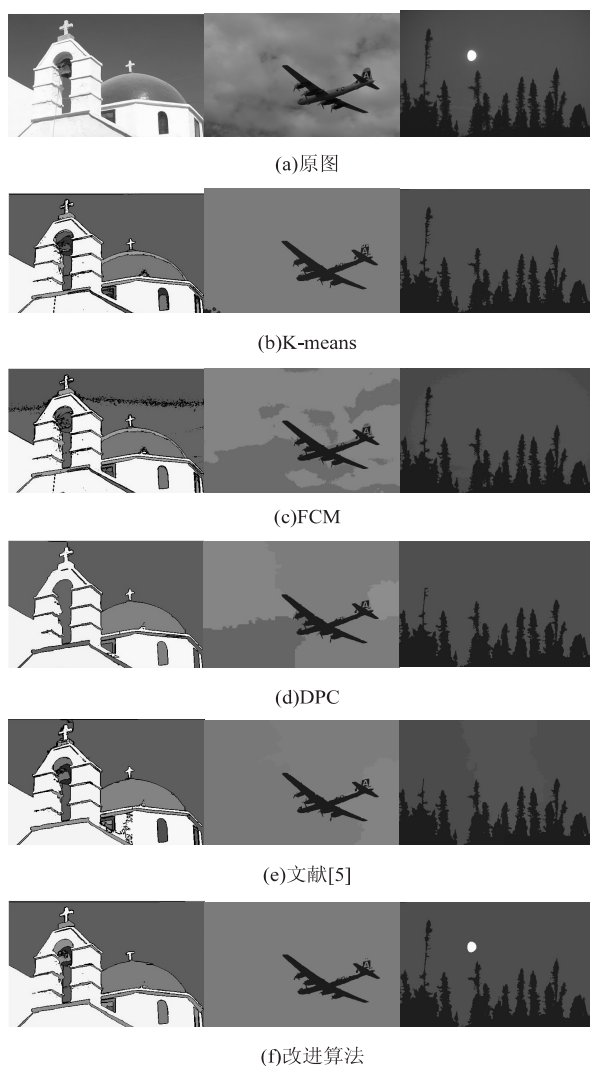


图3 不同算法分割结果

由图3可知,上述算法均能对数字图像进行分割。但在细节方面,K-means 和 FCM 算法会出现过分割的现象,即冗余区域过多,导致本应属于同一区域的像素点被冗余的分割线条划分开,DPC 算法和文献[5]中的算法错误分割较多。四种对比算法下的分割结果,均会导致颜色变化并不突兀的地方出现了过多冗余线条。而改进算法的分割得到图像轮廓与 GroundTruth 更加贴近,每个独立目标中的小区域很少,提取的轮廓界限也更加清晰可见。

作为较为经典的聚类算法, K-means 比原始的 DPC 算法在图像分割上具有更明显的分割精度, 然而无论是 K-means、FCM、DPC 还是文献[5]中的算法, 均无法合理地选取簇类个数, 这也是传统的聚类方法面临的统一难题。通过横向对比发现, 改进算法具有较强的稳定性, 从三个评价指标来看, 改进算法均具有较强的优越性, 无需确定任何参数, 便可自适应地进行较为理想的图像分割。计算图 3 中的 3 幅图像 PRI、VOI 和 USE 三个指标平均值, 指标数据如表 3 所示。

表 3 各指标比较

指标	K-means	FCM	DPC	文献[5]	文中算法
PRI	0.826 1	0.802 8	0.816 1	0.832 9	0.874 5
VOI	1.699 2	1.719 0	1.914 1	1.700 3	1.655 3
USE	0.462 5	0.476 4	0.493 8	0.459 1	0.406 5

4 结束语

针对 DPC 算法对参数敏感的问题, 对截断距离 d_c 进行自适应选取, 并将改进的算法应用于图像分割。为解决 DPC 算法难以处理较大图像的问题, 提出了分块并行的解决思路, 并根据图像各区域内聚程度提出自适应获取数字图像簇类个数的方法。实验表明, 该算法不但对二维数据有较好的聚类效果, 而且能够适用于图像分割。

参考文献:

- [1] LEI T, LIU P, JIA X. Automatic fuzzy clustering framework for image segmentation[J]. IEEE Transactions on Fuzzy Systems, 2019, 28(9): 1-15.
- [2] AGARWAL J, NAGPAL R, SEHGAL R. Crime analysis using K-Means clustering[J]. International Journal of Computer Applications, 2018, 83(4): 1-4.
- [3] QIAN X Z, HUI J. Density peak clustering optimized by adaptive aggregation strategy[J]. Journal of Computer Science and Exploration, 2020, 14(4): 712-720.
- [4] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.
- [5] 谢娟英, 高红超, 谢维信. K 近邻优化的密度峰值快速搜索聚类算法[J]. 中国科学: 信息科学, 2016, 46(2): 258-280.
- [6] 钱雪忠, 金 辉. 自适应聚合策略优化的密度峰值聚类算法[J]. 计算机科学与探索, 2020, 14(4): 712-720.
- [7] CHEN J, ZHENG H, LIN X, et al. A novel image segmentation method based on fast density clustering algorithm[J]. Engineering Applications of Artificial Intelligence, 2018, 73(8): 92-110.
- [8] 曾绍华, 唐文密, 詹林庆, 等. 基于自适应密度峰值聚类的野外紫色土彩色图像分割[J]. 农业工程学报, 2019, 35(19): 200-208.
- [9] 陈晋音, 何辉豪. 基于密度的聚类中心自动确定的混合属性数据聚类算法研究[J]. 自动化学报, 2015, 41(10): 1798-1813.
- [10] 孙 巍, 郭 敏. 基于 SLIC 与条件随机场的图像分割算法[J]. 计算机应用研究, 2015, 32(12): 3817-3820.
- [11] 蒋礼青, 张明新, 郑金龙, 等. 快速搜索与发现密度峰值聚类算法的优化研究[J]. 计算机应用研究, 2016, 33(11): 3251-3254.
- [12] 李 广, 冯 燕. 基于 SIFT 特征匹配与 K-均值聚类的运动目标检测[J]. 计算机应用, 2012, 32(10): 2824-2826.
- [13] LIU R, WANG H, YU X. Shared-nearest-neighbor-based clustering by fast search and find of density peaks[J]. Information Sciences, 2018, 450(7): 200-226.
- [14] JIANG D, ZANG W, SUN R, et al. Adaptive density peaks clustering based on K-nearest neighbor and Gini coefficient[J]. IEEE Access, 2020, 8(7): 900-917.
- [15] 刘安琪, 刘华勇, 王焕宝. 改进的 SLIC 超像素图像分割与合并算法[J]. 安徽建筑大学学报, 2020, 28(4): 39-46.
- [16] JIE Hu, LI Shen, GANG Sun. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). Washington D. C., USA: IEEE, 2018: 7132-7141.
- [17] GAO W, CAO B, SHAN S, et al. The CAS-PEAL large-scale chinese face database and baseline evaluations[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, 2008, 38(1): 149-161.

(上接第 46 页)

- [11] QIAN Sheng, LIU Hua, LIU Cheng, et al. Adaptive activation functions in convolutional neural networks[J]. Neurocomputing, 2018, 272: 204-212.
- [12] RAMACHANDRAN P, ZOPH B, LE Q V. Searching for Activation Functions[J]. arXiv:1710.05941, 2017.
- [13] 李国豪, 袁一帆, 贾晔烨, 等. 采用时空注意力机制的人脸微表情识别[J]. 中国图象图形学报, 2020, 25(11): 2380-2390.