Superpixel Based Hierarchical Segmentation for Color Image

Chong Wu††, Member, Le Zhang††, Houwang Zhang†††, and Hong Yan†, Nonmembers

SUMMARY In this letter, we propose a hierarchical segmentation (HS) method for color images, which can not only maintain the segmentation accuracy, but also ensure a good speed. In our method, HS adopts the fuzzy simple linear iterative clustering (Fuzzy SLIC) to obtain an over-segmentation result. Then, HS uses the fast fuzzy C-means clustering (FFCM) to produce the rough segmentation result based on superpixels. Finally, HS takes the non-iterative K-means clustering using priority queue (KPQ) to refine the segmentation result. In the validation experiments, we tested our method and compared it with state-of-the-art image segmentation methods on the Berkeley (BSD500) benchmark under different types of noise. The experiment results show that our method outperforms state-of-the-art techniques in terms of accuracy, speed and robustness.

key words: image segmentation, robustness, superpixel

1. Introduction

Image segmentation plays an important role in computer vision and image processing. It can be further used in some vision tasks, such as object recognition and classification. The main task of image segmentation is to divide an image into disjoint regions, in each of which pixels are similar in color and other properties [1].

Image segmentation methods can be divided into two main categories: (1) supervised and (2) unsupervised techniques. In this letter, we focus on the latter. Most unsupervised methods are based on clustering techniques because they are superior to other unsupervised methods in terms of speed and accuracy [2].

Among clustering methods, fuzzy C-means (FCM) clustering [3] method is widely used in image segmentation. FCM can retain more original image information in contrast to hard clustering methods, because it incorporates the idea of partial membership to represent the relationship between pixels and clusters. However, the original FCM has two drawbacks which may result in poor segmentation performance sometimes. The first one is that the image segmentation result obtained by FCM is sensitive to noise due to the lack of local spatial information and the other one is high time-consuming caused by the repeated distance computation between clustering centroids and pixels.

In recent years, a variety of FCM based approaches have been developed, like FLICM [4], FGFCM [5], FRFCM [6]. However, most of them have the same shortcomings as we state above, and most of them are pixel-wise, which leads to the high computational complexity. Although some superpixel-based clustering methods have been proposed [7], the problems of robustness and speed still exist among them.

To solve the above problems, we propose a hierarchical segmentation (HS) method based on fuzzy simple linear iterative clustering (Fuzzy SLIC) [8], fast fuzzy C-means clustering (FFCM) [7] and non-iterative K-means clustering using priority queue (KPQ). This is an expanded and improved version of our previous work [2]. Compared to our previous work, it has two main improvements: (1) it can be applied to multi-object segmentation; (2) KPQ is proposed and FFCM is simplified to improve the segmentation and computational performance. It performs continuous hierarchical processes from the initial over-segmentation to the final refined segmentation. It can take the advantage of different levels of segmentation techniques. Firstly, Fuzzy SLIC will return a robust over-segmentation result. Then, FFCM will combine the superpixels and output a rough segmentation result. Last, KPQ will be conducted to refine the segmentation result. All three steps have a low complexity so that HS has a good computational efficiency. To validate the proposed method, we tested and compared it with five state-of-the-art methods: FCM [9], FLICM [4], FGFCM [5], FRFCM [6], and SFFCM [7] on the Berkeley (BSD500) benchmark [10] under noise-free and noisy environments. The contributions of this letter include:

(1) A method fusing different levels of segmentation techniques has been proposed.
(2) We developed a non-iterative K-means clustering to produce smooth and continuous boundaries.
(3) We have developed a fast and robust image segmentation method.

2. A Brief Review of Fuzzy Simple Linear Iterative Clustering

Superpixel generation method has become a useful pre-processing tool for image processing since the concept of superpixel was proposed [11]. To solve the robustness problem
of most superpixel generation methods, fuzzy simple linear iterative clustering (Fuzzy SLIC) has been developed [8], which is based on a local spatial fuzzy C-means clustering and dynamic fuzzy superpixels. By default, two parameters should be set. They are \( n \), the desired number of superpixels, and \( C \), the compactness coefficient used to control the superpixel shape. In our method, we set \( n = 60 \) and \( C = 17 \) to balance segmentation accuracy and computational performance. Due to the page limitation, please refer to [8] for the details of Fuzzy SLIC.

3. Simplified Fast Fuzzy C-Means Clustering

When traditional fuzzy C-means clustering (FCM) is used in image segmentation, it computes the distance on pixel-wise for many times until the membership partition matrix converges. It is time-consuming because of a large number of pixels in an image. To speed up this process, we do the clustering on superpixel-wise rather than pixel-wise. In addition, to further speed up the clustering process, we adopt fast fuzzy C-means clustering (FFCM) [7]. To reduce the complexity, we simplify the objective function of FFCM as follow.

\[
J = \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ij}^m ||s_i - c_j||^2, \quad s_i = \frac{1}{|R_i|} \sum_{p \in R_i} x_p, 1 \leq i \leq n
\]

(1)

where \( m \) is the number of clusters, \( h \) is the fuzzy partition matrix exponent, \( u_{ij} \) is the membership degree, \( c_j \) is the \( j \)th cluster centroid, \( n \) is the number of superpixels, \( R_i \) denotes the set of pixels in the \( i \)th superpixel, \( |R_i| \) is the number of the pixels in \( R_i \), \( x_p \) denotes the color value of pixel \( p \) within \( R_i \), hence, \( s_i \) is the mean value of color of pixels in the \( i \)th superpixel.

The local optimum of Eq. (1) can be obtained by updating \( c_j \) and \( u_{ij} \) as follows,

\[
c_j = \frac{\sum_{i=1}^{n} u_{ij}^m s_i}{\sum_{i=1}^{n} u_{ij}^m}, \quad u_{ij} = \left( \sum_{k=1}^{m} ||s_i - c_k||^{2/(h-1)} \right)^{-1}
\]

(2)

Compared to original FFCM update Eq. (3), Eq. (2) reduces the computation of \( u_{ij} \). This simplification is based on an assumption that the number of pixels in each superpixel is roughly equal and Fuzzy SLIC is the key to make the assumption true. Because the superpixels generated by Fuzzy SLIC have a regular size even in high noisy environment.

\[
c_j = \frac{\sum_{i=1}^{n} u_{ij}^m s_i}{\sum_{i=1}^{n} u_{ij}^m}, \quad u_{ij} = \frac{||s_i - c_j||^{2/(h-1)}}{\sum_{k=1}^{n} ||s_i - c_k||^{2/(h-1)}}
\]

(3)

The pseudo code of FFCM is shown in Algorithm 1.

4. Non-Iterative K-Means Clustering Using Priority Queue

To produce smooth and continuous boundaries, we propose a non-iterative K-means clustering using priority queue (KPQ) to refine the segmentation result obtained by FFCM. Using KPQ can achieve a fast and non-iterative K-means clustering. In KPQ, the ith superpixel candidate with the shortest distance \( D_{ij} \) to the \( j \)th cluster centroid will be popped from the priority queue (PQ). The metric to measure distance is calculated as follow:

\[
d_{ij} = \frac{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}{(l_{max} - l_{min})^2 + (a_{max} - a_{min})^2 + (b_{max} - b_{min})^2}
\]

\[
d_{ij} = \frac{(x_i - x_j)^2 + (y_i - y_j)^2}{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}
\]

\[
D_{ij} = d_{ij}^c + d^t_{ij} + C_c
\]

where, \( C_c \) is a normalized factor used to weight the relative

Algorithm 1 Fast Fuzzy C-means Clustering

Require: Superpixel image \( I \), label matrix \( L \), number of clusters \( m \), number of superpixels \( n \);
Ensure: Label matrix \( L_e \);
1: Set \( h = 2 \);
2: Set \( g' \) (convergence condition) = 1e-5;
3: Set \( itr \) (max iteration times) = 100;
4: Compute \( s_i \) as Eq. (1);
5: Initialize randomly the membership partition matrix \( U^{(0)} \) according to \( I \);
6: for \( b = 1 \) to \( itr \) do
7: \quad Update the clustering centroids \( c \) using Eq. (2);
8: \quad Update the membership partition matrix \( U^{(b)} \) using Eq. (3);
9: \quad if \( b > 1 \) then
10: \quad \quad If \( max(U^{(b-1)} - U^{(b)}) < g' \) then break;
11: \quad end if
12: \quad end if
13: \quad end for
14: Compute label matrix \( L_e \);
15: return \( L_e \);

Algorithm 2 Non-iterative K-means Clustering using Priority Queue

Require: Superpixel image \( I \), label matrix \( L \), number of clusters \( m \), number of superpixels \( n \);
Ensure: Segmentation result matrix \( L' \);
1: Initialize \( Label[k] = 0 \);
2: Compute \( s = [s_1, \ldots, s_n] \) using Eq. (1);
3: Use label matrix \( L \) to compute the initial cluster centers \( C = [c_1, \ldots, c_m] \) using Eq. (2);
4: for \( b = 1 \rightarrow m \) do
5: \quad Compute the distance \( D_{ik}, \ldots, D_{ih} \) using Eq. (4);
6: \quad Find the two smallest distance \( D_{ik}, D_{ik+1} \);
7: \quad Element \( e_i \) ← \([k, b, 0] \);
8: \quad Element \( e_i \) ← \([k, b, 0] \);
9: \quad Push \( e_i \) and \( e_i+1 \) on priority queue \( Q \);
10: end for
11: while \( Q \) is not empty do
12: \quad Pop \( Q \) to get \( e; k = e_k; \)
13: \quad if Label[k] = 0 then
14: \quad b = e_b;
15: \quad Label[k] = b;
16: \quad Compute the distance \( D_{ik}, \ldots, D_{ih} \) using Eq. (4);
17: \quad Create array \( E = [c_1, \ldots, c_m] \);
18: \quad for \( a = 1 \rightarrow m \) do
19: \quad \quad if Label[a] = 0 then
20: \quad \quad \quad \( e_a = [k, b, D_{ik}] \);
21: \quad \quad \quad Push \( e_a \) on \( Q \);
22: \quad \quad end if
23: \quad end for
24: \quad Update the cluster centroids \( C \);
25: \quad end if
26: end while
27: Compute segmentation result matrix \( L' \);
28: return \( L' \);
importance between color distance $d_{c,i,j}$ and spatial distance $d_{s,i,j}$ in the CIELAB color space instead of RGB color space. In our method, we set $C_c = 0.03$. The values of $d_{c,i,j}$ and $d_{s,i,j}$ are normalized into $[0,1]$, which use the Euclidean distance to map the spatial distance and color distance in the CIELAB color space.

The pseudo code of KPQ is as Algorithm 2 shows. After executing FFCM, we obtain $m$ initial cluster centroids $c = [c_1, \ldots, c_m]$, where $c_j = [l_i, a_i, b_i, x_i, y_i]^{T}$, and $[l_i, a_i, b_i]^{T}$ is the color information, $[x_i, y_i]^{T}$ is the spatial information. In KPQ algorithm, each cluster centroid is assigned to two initial seeds (superpixels) which are nearest to the cluster centroid. Beginning with these $2m$ initial seeds, the algorithm adopts a priority queue to decide which superpixel will be added into a cluster (object). Unlabeled superpixels are pushed in the priority queue. Popping the queue provides the superpixel that has the shortest distance $D$ to a centroid. Once the new superpixel is added to a cluster, the cluster centroid will be updated. Hence, the whole process of updating centroids is in a single iteration. An overview of our proposed method HS is shown in Fig. 1.

5. Experiments and Results

To evaluate the performance of our method, we compared it with five comparative clustering-based methods: FCM [9], FLICM [4], FGFCM [5], FRFCM [6], and SFFCM [7] on the Berkeley Segmentation Dataset (BSD500) [10]. The experiments are conducted on Mac with OSX 10.14, Intel Core i5 2.3GHz 2 cores CPU, and 8 GB RAM.

The parameters of each method are set to the default. Four performance measures: the probabilistic rand index (PRI), the variation of information (VI), the global consistency error (GCE), and the boundary displacement error (BDE) are used to compare the quality of image segmentation results like [2]. If the segmentation result is more similar to the ground truth, PRI will be larger, but VI, GCE and BDE will be smaller.

Table 1 Comparison results obtained by all methods in binary object segmentation. The best values are in bold face and the second best values are in italic.

| Algorithms | BDE | PRI | Vol | GCE |
|------------|-----|-----|-----|-----|
| noise-free |     |     |     |     |
| FCM        | 14.41 | 0.69 | 3.25 | 0.46 |
| FGCMM      | 14.49 | 0.67 | 3.39 | 0.48 |
| FLICM      | 14.32 | 0.68 | 3.36 | 0.48 |
| FRFCM      | 13.80 | 0.71 | 2.87 | 0.39 |
| SFFCM      | 13.27 | 0.74 | 2.23 | 0.28 |
| HS         | 14.03 | 0.76 | 2.39 | 0.26 |
| gaussian   |     |     |     |     |
| FCM        | 16.51 | 0.66 | 3.74 | 0.53 |
| FGCMM      | 16.41 | 0.66 | 3.66 | 0.53 |
| FLICM      | 16.43 | 0.67 | 3.49 | 0.50 |
| FRFCM      | 16.56 | 0.66 | 3.64 | 0.51 |
| SFFCM      | 18.46 | 0.71 | 2.52 | 0.33 |
| HS         | 15.92 | 0.74 | 2.67 | 0.30 |
| multiplicative |     |     |     |     |
| FCM        | 16.51 | 0.66 | 3.74 | 0.53 |
| FGCMM      | 16.41 | 0.66 | 3.66 | 0.53 |
| FLICM      | 16.43 | 0.67 | 3.49 | 0.50 |
| FRFCM      | 16.56 | 0.66 | 3.64 | 0.51 |
| SFFCM      | 18.46 | 0.71 | 2.52 | 0.33 |
| HS         | 15.92 | 0.74 | 2.67 | 0.30 |
| salt & pepper |     |     |     |     |
| FCM        | 16.48 | 0.63 | 3.44 | 0.50 |
| FGCMM      | 16.44 | 0.65 | 4.02 | 0.58 |
| FLICM      | 16.55 | 0.63 | 3.32 | 0.45 |
| FRFCM      | 14.77 | 0.68 | 3.14 | 0.42 |
| SFFCM      | 21.80 | 0.69 | 2.77 | 0.38 |
| HS         | 15.09 | 0.75 | 2.49 | 0.28 |
in terms of BDE and our HS achieves the best BDE in most cases. The reason why superpixel-based methods perform worse in terms of BDE in binary object segmentation is that superpixel has a property called regularity which is contradictory to boundary accuracy. Regular superpixels provide a direct and easy way to analyze the neighborhood information of the image content. However, a more regular shape means a lower boundary accuracy. It is a trade-off. When superpixels are combined together, boundary error will be accumulated. There are more superpixels combined in binary object segmentation than in multi-object segmentation, and this is why superpixel-based methods achieve worse BDE in binary object segmentation. Figure 2 shows the average execution time of all methods. The speed of HS is at the fastest level. It is faster than SFFCM which is called one of the extremely fast segmentation methods and very close to the FGFCM. With the increase of number of objects, the execution time of HS is reduced while the fastest FGFCM is increased. Figure 3 shows the visual comparison results obtained by all methods with salt and pepper noise. We can see that except our method, all methods are unable to retain the boundary information and preserve the structure from the image under a noisy environment.

6. Conclusion

In this letter, we propose a hierarchical segmentation (HS) method. In the validation experiments, our method outperforms state-of-the-art methods for both noise-free and noisy images. One of the most significant advantages of our method is that it is faster than or has comparable speed with some extremely fast state-of-the-art methods like SFFCM and FGFCM.

Acknowledgments

This work was supported by the Hong Kong Research Grants Council [Project C1007-15G] and City University of Hong Kong [Projects 7005230 and 9610460].

References

[1] H. Zhang and Y. Zhu, “KSLIC: K-mediods clustering based simple linear iterative clustering,” Chinese Conference on Pattern Recognition and Computer Vision, vol.11858, pp.519–529, Springer, 2019.
[2] C. Wu, L. Zhang, H. Zhang, and H. Yan, “Improved superpixel-based fast fuzzy C-means clustering for image segmentation,” IEEE International Conference on Image Processing, pp.1455–1459, IEEE, 2019.
[3] J.C. Dunn, “A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters,” Cybernetics and Systems, vol.3, no.3, pp.32–57, 1973.
[4] S. Krinidis and V. Chatzis, “A robust fuzzy local information C-means clustering algorithm,” IEEE Trans. Image Process., vol.19, no.5, pp.1328–1337, 2010.
[5] W. Cai, S. Chen, and D. Zhang, “Fast and robust fuzzy C-means clustering algorithms incorporating local information for image segmentation,” Pattern Recognition, vol.40, no.3, pp.825–838, 2007.
[6] T. Lei, X. Jia, Y. Zhang, L. He, H. Meng, and A.K. Nandi, “Significantly fast and robust fuzzy C-means clustering algorithm based on morphological reconstruction and membership filtering,” IEEE Trans. Fuzzy Syst., vol.26, no.5, pp.3027–3041, 2018.
[7] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, and A.K. Nandi, “Superpixel-based fast fuzzy C-means clustering for color image segmentation,” IEEE Trans. Fuzzy Syst., vol.27, no.9, pp.1753–1766, 2016.
[8] C. Wu, L. Zhang, H. Zhang, and H. Yan, “Fuzzy SLIC: Fuzzy simple linear iterative clustering,” arXiv preprint arXiv:1812.10932, 2018.
[9] N.R. Pal and J.C. Bezdek, “On cluster validity for the fuzzy C-means model,” IEEE Trans. Fuzzy Syst., vol.3, no.3, pp.370–379, 1995.
[10] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol.33, no.5, pp.898–916, 2010.
[11] X. Ren and J. Malik, “Learning a classification model for segmentation,” IEEE International Conference on Computer Vision, pp.10–17, IEEE, 2003.