Multimed Tools Appl (2016) 75:11417–11432
DOI 10.1007/s11042-015-2860-6

Histogram-based colour image fuzzy clustering algorithm

Hai-peng Chen[1,2] **· Xuan-Jing Shen**[1,2] **· Jian-Wu Long**[3]

Received: 18 January 2015 / Revised: 17 June 2015 / Accepted: 3 August 2015 /
Published online: 27 August 2015

Abstract In this work, a histogram-based colour image fuzzy clustering algorithm is proposed for addressing the problem of low efficiency due to computational complexity and poor clustering performance. Firstly, the presented scheme constructs the red, green and blue (short for RGB) component histograms of a given colour image, each of which is pre-processed to preserve their smoothness. Secondly, the proposed algorithm multi-thresholds each component histogram, using some dominant valleys identified from a fast peak-valley location scheme in each global histogram. Thirdly, a new histogram is reconstructed by applying a histogram merging scheme to the RGB three-component histograms, and multi-thresholding this new histogram again using some dominant valleys obtained from the fast peak-valley location scheme. Thus, the proposed approach can easily identify the initialisation condition of cluster centroids and centroid number. Finally, we construct a new dataset composed of some pre-segmented small regions using the WaterShed algorithm, and the FCM (Fuzzy C-Means) algorithm is executed on this dataset, instead of on pixels, in combination with the initial cluster centroids. Experimental results have demonstrated that the proposed algorithm is more efficient than the DSRPCL (Distance Sensitive Rival Penalised Competitive Learning) algorithm and the HTFCM (Histogram Thresholding Fuzzy C-Means) algorithm with respect to run times and PRI (Probability Rand Index) values.

Keywords Colour image segmentation · Histogram · Clustering · FCM algorithm

1 Introduction

Image segmentation is a fundamental problem in image processing, but it also represents a difficult research problem. Image segmentation is widely used in the fields of pattern recognition, computer

✉ Hai-peng Chen
chenhp@jlu.edu.cn

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China

[2] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China

[3] College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400045, China

vision, machine learning and medical image processing. Based on whether the processing image contains colour information, we can divide the image segmentation techniques into the grey image segmentation techniques and the colour image segmentation techniques. The family of grey image segmentation techniques is widely applied in processing text images, industrial images and medical images and includes the edge detection algorithm [2], the clustering algorithm [3], the thresholding segmentation algorithm [6, 7, 13, 18], and the partial differential equation based segmentation algorithm [5]. The family of colour image segmentation techniques is mainly used to process, for example, natural images and videos [16]. Common methods in colour image segmentation techniques include the interactive segmentation algorithm based on graph theory [4, 10, 11] and the clustering algorithm [8, 14, 19]. Recently, Sarkar etc. proposed a multilevel colour image segmentation algorithm based on minimum cross entropy and differential evolution and obtained efficient performance [12].

The interactive segmentation algorithm based on graph theory belongs to the class of supervised methods and is used mainly in cases requiring human interaction, such as editing photos. The clustering algorithm is an unsupervised method and can be used in colour image clustering and the video processing; the classical algorithm implementing this strategy is the FCM method [8, 9, 14, 19]. Because the FCM clustering algorithm is not restricted by the dimensionality of the sample data, it can process the one-dimensional grey-scale images, the three-dimensional colour images and the high-dimensional data. Together with the robustness of clustering performance, the FCM clustering algorithm has been extensively researched and has many applications [3, 8, 9, 14, 19]. Krinidis et al. proposed a robust fuzzy local information c-means clustering algorithm (FLICM) [3] in 2010, with a comparative analysis of the existing clustering algorithms, including the traditional FCM algorithm, the constrained FCM algorithm, the enhanced FCM algorithm and the fast generalised FCM algorithm. Additionally, the FLICM algorithm exhibits notably strong anti-noise performance. As the key parameter that affects the FCM clustering performance are the cluster number and the selection of the initial cluster centroids, which are set by users of the above clustering algorithms, the clustering performance shows no substantial improvement. To improve the FCM clustering performance, Mok et al. presented a robust adaptive clustering analysis method for automatic identification of clusters [9]. However, it is difficult to apply the method to image clustering with a large sample number. Ma et al., in 2006, proposed a cost-function approach to rival penalised competitive learning (DSRPCL) [8]. This method can automatically identify the clusters through a penalised competitive mechanism, but a large number of initial clusters are still needed to be set by the users, and in addition, the initial cluster centroids are obtained by a randomised approach. Yu et al. put forward an ant colony–fuzzy C-means hybrid algorithm (AFHA) [19]. First, the ant colony optimisation is used to obtain the initial cluster centroids; second, the traditional FCM algorithm is applied to conduct colour image clusters. However, this algorithm performs slowly in the ant colony optimisation step, which results in low efficiency [14]. On this basis, Tan et al. proposed a colour image segmentation using histogram thresholding-fuzzy C-means hybrid approach (HTFCM) [14]. This approach uses the histogram information to effectively improve the clustering efficiency and performance, but whether the obtained initial cluster centroids are reasonable, is basically influenced by three control parameters. In the cluster initialisation process, an unreasonable hard division method is employed in the algorithm, which results in much misclassification. Moreover, the pixels are taken as the sample data, leading to poor efficiency due to the large number of pixels.

Based on the above analysis, this paper proposes a histogram-based colour image fuzzy clustering algorithm (HCIFCM). This algorithm takes full advantage of the important piece of information that optimal segmentation threshold is located in the valley, and it can automatically determine the number of clusters and the initial cluster centroids accurately. To improve the clustering efficiency, the WaterShed algorithm [17] is used to perform image pre-segmentation. Making use of regions obtained through the pre-segmentation and not the original pixels as the clustering samples has improved the clustering efficiency greatly. To verify the effectiveness of the algorithm, many testing experiments are performed on the Berkeley image database [1], and a comparison between the DSRPCL algorithm [8] and HTFCM algorithm [14] is made in terms of the running time and the PRI metric. The experiments demonstrate that the clustering performance of our algorithm is more desirable.

The structure of this paper is organized as follows. In Section 2, the HTFCM algorithm is reviewed. In Section 3, the novel HCIFCM algorithm is proposed for addressing the problem of low efficiency due to computational complexity and poor clustering performance of the HTFCM algorithm. Section 4 gives the experimental results on some colour natural images. Finally, the conclusion is drawn in Section 5.

2 Motivation

In 2011, Tan et al. presented a colour image segmentation algorithm using histogram thresholding-fuzzy C-means hybrid approach (HTFCM algorithm). The basic steps of this algorithm are as follows.

Step.1 First, apply a peak search method to find all the peak sets (P_R, P_G and P_B) of the statistical RGB histogram in a colour image. The corresponding number of the peaks is denoted by n_R=|P_R|, n_G=|P_G| and n_B=|P_B|, respectively. The basic principle of the peak search method is to retain the candidate peak whose number of pixels is greater than the given threshold H.

Step.2 Merge the peaks obtained in **Step.1** to form candidate cluster centroids (i_R,i_G,i_B), where, $i_R \in P_R$, $i_G \in P_G$ and $i_B \in P_B$. Subsequently, assign every image pixel to the nearest cluster centroid and count the number of pixels in each cluster. Eliminate the cluster centroid, for which cluster, the number of pixels assigned is less than a given threshold V. The pixels in each eliminated cluster centroid are reassigned to the nearest cluster centroid. Finally, update each cluster centroid.

Step.3 Calculate the Euclidean distance D for any two cluster centroids. Merge the nearest cluster and update the cluster centroid if the minimum distance $D_{\min}<dc$, where dc is the given threshold. Stop the iteration unless $D_{\min}<dc$ is not satisfied. Then, the initial cluster centroids are obtained.

Step.4 All the pixels in the image will be taken as the sample data. To obtain the cluster results, we implement the FCM algorithm for the sample data, using the initial cluster centroids gained in the previous step.

From the basic steps of the HTFCM algorithm, we can see that whether the obtained initial cluster centroids are reasonable is mainly affected by three control parameters, H, $V$and dc. For example, in Step1, setting too small a threshold H will result in excessive candidate peaks being reserved, which increases the difficulty of subsequent processing; while if it is set too large, many useful peaks will be lost, which makes the final obtained initial cluster centroids unreasonable. The latter two parameters are critical, as well. These parameters are often selected based on a subjective experience that will directly affect the reasonableness of the initial cluster centroids. In addition, there are many other problems. For instance, after the peak is searched in Step1, the division method adopted by the HTFCM algorithm, which is to divide each pixel into the nearest cluster, is not reasonable. The division method takes the mean of two peaks as the segmentation threshold, while the optimal threshold should be in the valley between the two peaks. However, the mean is not in the valley, or it deviates far away in general. Therefore, the HTFCM algorithm may cause much misclassification. Moreover, all the pixels are taken as the sample data in the FCM clustering process, causing poor efficiency.

3 Method description

To address the problems above, we presented a histogram-based colour image fuzzy clustering algorithm (HCIFCM). The detailed process is presented in Fig. 1.

3.1 Histogram preprocessing

In a colour image, the value range of grayscale *i* is [0,255], and the component histograms of RGB - the three primary colours - are denoted by *R*(*i*), *G*(*i*) and *B*(*i*), for each component, respectively. Smoothing is made for each component histogram as follows,

$$H_S(i) = \frac{1}{2k+1}\sum_{x=i-k}^{i+k} S(x) \tag{1}$$

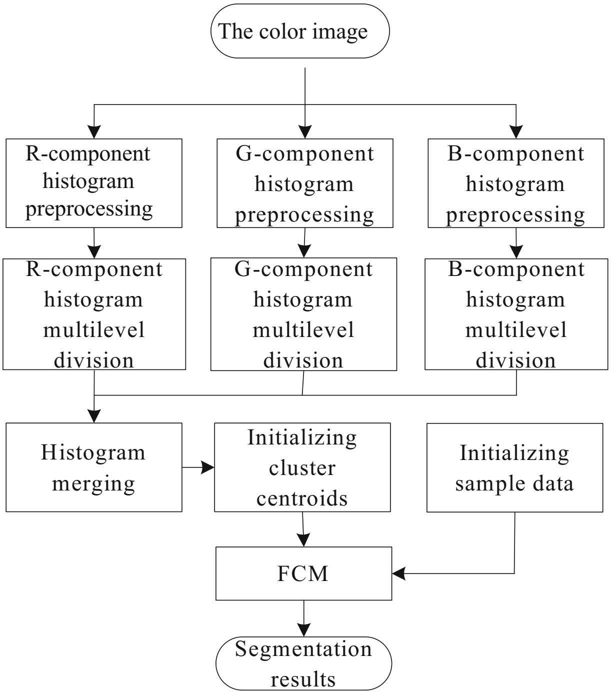

The color image
R-component histogram preprocessing
G-component histogram preprocessing
B-component histogram preprocessing
R-component histogram multilevel division
G-component histogram multilevel division
B-component histogram multilevel division
Histogram merging
Initializing cluster centroids
Initializing sample data
FCM
Segmentation results

Fig. 1 HCIFCM algorithm process

where k is a positive integer, and $S(x)$ is a component histogram, $S\in\{R,G,B\}$. For example, $H_R(i)$ denotes the smoothed R component histogram. The parameter k should not be too large; otherwise, it will cause excessive smoothness in the histogram [14]. Commonly, k can be set from 2 to 5, which will not only make the histogram achieve smoothness but also ensure that the image detail information is not lost too much [14]. Because k is small, many local extrema may still be exhibited in the smoothed histogram; therefore, we need to eliminate these to further improve the histogram waveform's smoothness and reduce the influence of the local extrema. This paper employs the following equation to search for all the local extrema in the histogram [14].

$$\begin{cases} PMax = \left\{ i \middle| H_S(i) > H_S(i-1) \wedge H_S(i) > H_S(i+1) \right\} \\ VMin \quad = \left\{ i \middle| H_S(i) < H_S(i-1) \wedge H_S(i) < H_S(i+1) \right\} \end{cases} \tag{2}$$

where $PMax$ is the maximum set and $VMin$ is the minimum set. To eliminate the extrema, the greater value between the two neighbours is used to replace the maximum, while the lower value is used to replace the minimum. That is,

$$\begin{cases} H_S(i) = \max\{H_S(i-1), H_S(i+1)\}, & \text{if} \;\; i\in PMax \\ H_S(i) = \min\{H_S(i-1), H_S(i+1)\}, & \text{if} \;\; i\in VMin \end{cases} \tag{3}$$

After the above pre-processing, the histogram waveform will be smoother, so that the peak and the valley can be positioned accurately.

3.2 Histogram multilevel division and mergering

In this study, the fast location method based on the grayscale wave transformation [18] was employed to obtain all the peaks and valleys in the whole histogram waveform. When the difference between a putative peak and its neighbouring valley is larger than the wave threshold T, we can firmly determine a peak or a valley. Without pre-processing for the histogram, different grayscale images need to be given different wave thresholds to achieve optimal segmentation results. The value of T ranges from 10 to 40. According to the experiment, we find that for certain images in other application fields, the value of T is larger or smaller. The reason is that, in this method, the grayscale wave change, from a row, a column or a straight line direction at any angle, is taken as a waveform that has no statistical properties and is easily disturbed by noise or other factors. This results in the sensitivity to threshold T.

However, the application of the fast location method in reference [18] to this paper is more reasonable and effective. On the one hand, the object being processed in this study is the image histogram with special statistical properties and little external noise or other factor influences. On the other hand, the histogram waveform is smoother after pre-processing. The above two properties can alleviate the sensitivity to the threshold T. Through experiments, we find that the peak and the valley can be located accurately as long as we set a small fixed wave threshold.

The valley obtained by the fast location method is denoted by $V_S(i) = \{i_1, i_2, \cdots, i_\iota, \cdots, i_{n_S}\}$, where n_S is the total number of valleys and n_S+1 is the total number of peaks. If n_S=0, that is $V_S(i)$=∅, the histogram contains only one peak, namely the unimodal distribution; otherwise, the histogram is called multimodal. We take the obtained valley as the threshold for dividing the histogram into n_S+1 categories. In this step, all the wave thresholds are denoted by T_1. The detailed division process is as follows.

$$\begin{cases} L_S(i) = 0, & if \quad n_S = 0 \\ L_S(i) = 0, & if \quad n_S > 0 \wedge i < i_\iota \wedge \iota = 1 \\ L_S(i) = n_S, & if \quad n_S > 0 \wedge i > i_\iota \wedge \iota = n_S \\ L_S(i) = \iota - 1, & if \quad n_S > 0 \wedge i > i_{\iota-1} \wedge i < i_\iota \wedge \iota > 1 \end{cases} \tag{4}$$

where $L_S(i)$ is a division for each component histogram.

According to each component histogram above, the number of peaks is obtained, denoted by n_R, n_G and n_B respectively. Next, we obtain the classification numbers by the histogram multilevel division method. These are C_R, C_G and C_B, where C_R=n_R+1, C_G=n_G+1 and C_B= n_B+1. And the number of all possible classification patterns is L (L=C_R×C_G×C_B). After dividing each component histogram, a merge procedure is needed to obtain a single partition of the pixels in the whole image. For any pixel in the colour image, we denote its value as (r,g,b), and the classification number which this pixel belongs to is defined as

$$L(r,g,b) = L_R(r) \times C_G \times C_B + L_G(g) \times C_B + L_B(b) \tag{5}$$

3.3 Initializing clustering centroids

After finishing the division for all the pixels, we count the number of the pixels contained in each class. Let $H(i)$ denote the number of pixels in Class i, where $i \in [0, L-1]$, then a new histogram $H(i)$ is obtained. For each $H(i)$, use the histogram multilevel division method to obtain a division $\mathfrak{R}(i)$, in which process, the fluctuation threshold T_2 is adopted. The length of the new histogram $H(i)$ commonly satisfies L<<255. Due to the certain number of pixels in the same image, the height of the new histogram is much higher than the original histogram, which results in the greater drop between the peak and the valley. Thus we can find that the ratio of two fluctuation thresholds is inversely proportional to the ratio of the size about the two corresponding histograms, that is $\frac{T_1}{T_2} \propto 1/\left(\frac{256}{L}\right)$. Therefore, the fluctuation threshold T_2 can be set as follows.

$$T_2 = \beta \frac{256}{L} \cdot T_1 \tag{6}$$

where β is the control parameter.

If the division $\mathfrak{R}(i)$ includes C classes, the image is initially divided into C classes. And the final classification number $\Upsilon(r,g,b)$ for each pixel is given by,

$$\Upsilon(r,g,b) = \mathfrak{R}(L(r,g,b)) \tag{7}$$

Next, we calculate the colour mean $\left(\overline{r}_i, \overline{g}_i, \overline{b}_i\right)$ of each class and take it as the initial cluster centroid, where $i \in [0, C-1]$.

3.4 Initializing sample data

In paper [14], the pixels are taken as the sample data for FCM clustering and the large number of pixels causes a very time consuming clustering process. However, replacing the pixels with the pre-segmentation regions as the new sample is an admissible solution. WaterShed is one of the typical pre-segmentation methods, as is the Lazy Snapping [4] method presented by Li et al. in the process of the interactive colour image segmentation, and the IRMLGC algorithm [10, 11] proposed by Peng. The latter two methods use the pre-segmentation regions as the nodes, to model the image and complete further segmentation. However, the WaterShed algorithm is based on the gradient information of the grayscale image, irrespective of the rich information on colour contained in the colour image. Therefore, the colour information is used in this study to calculate the gradient value to alleviate the over-segmentation and to make the pre-segmentation more accurate. The area of the segmented region is usually small, so we adopt the colour mean as a new sample set to depict each region, which significantly reduces the sample number, and then improves the FCM clustering efficiency.

3.5 FCM algorithm

In the FCM algorithm, the key is that we set the cluster number and the cluster centroids appropriately, using a common method - the random cluster centroids initialisation. However, it may not be able to achieve the ideal clustering result, even using a large number of iterations, especially for the image clustering. Because the sample size is very large, unreasonable initialisation would make the clustering process more time-consuming; in addition, the generation of clustering results has certain randomness. However, once the initial cluster centroids are more accurate, the FCM algorithm will be much effective. To improve the clustering performance of the FCM algorithm effectively, in our paper, the histogram information was used to initialise the cluster centroids.

3.6 Examples analysis

The implementation process of this algorithm is illustrated by the following example. Figure 2(a) is a Sperm image, and Fig. 3(a), (c) and (e) are the RGB component original histograms, respectively. From Fig. 3(c), it is obvious that, on the main peak, there are many local extrema, which will bring considerable interference in accurately searching the real peaks

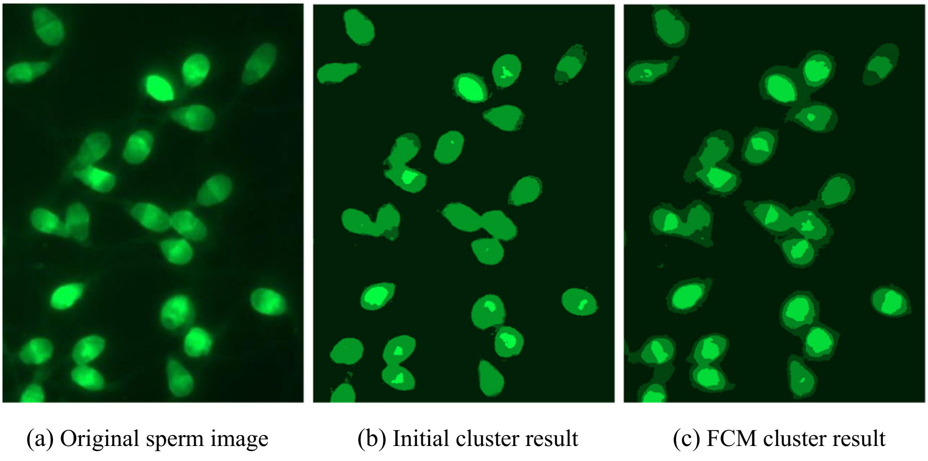
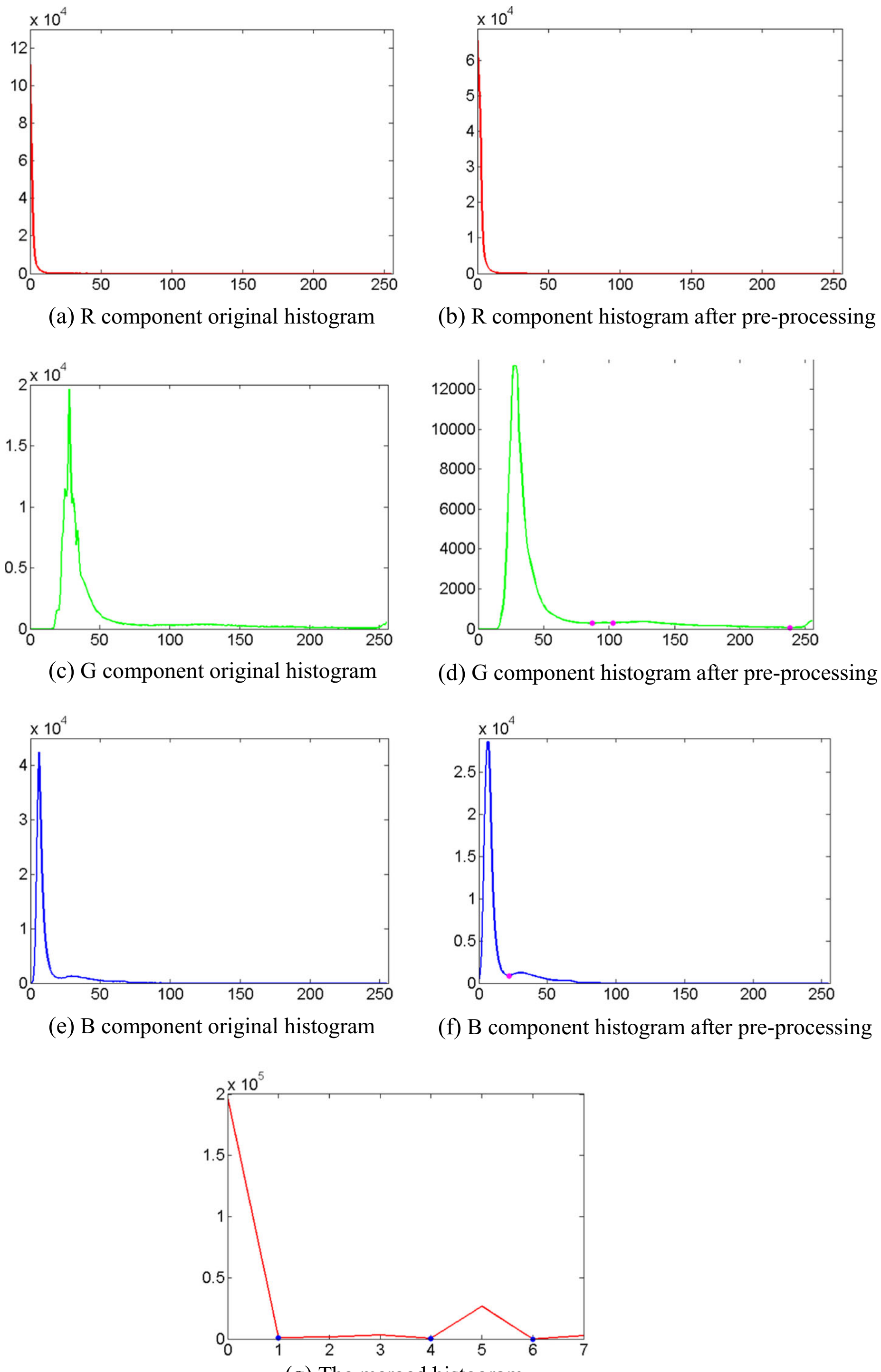

(a) Original sperm image (b) Initial cluster result (c) FCM cluster result

Fig. 2 Sperm image and its cluster result. **a** Original sperm image. **b** Initial cluster result. **c** FCM cluster result



(a) R component original histogram

(b) R component histogram after pre-processing

(c) G component original histogram

(d) G component histogram after pre-processing

(e) B component original histogram

(f) B component histogram after pre-processing

(g) The merged histogram

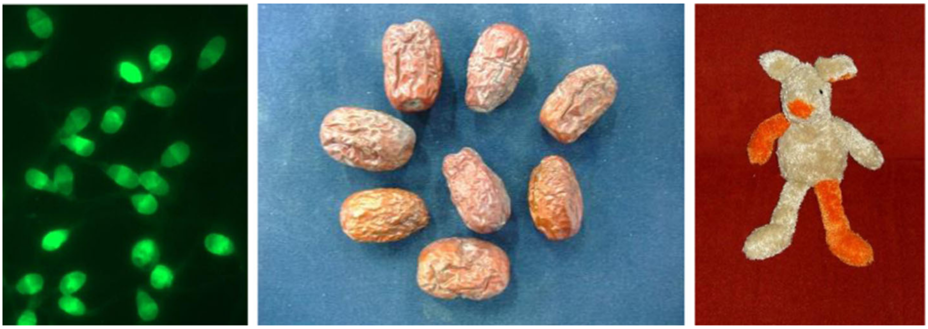
◀ **Fig. 3** Histograms. **a** R component original histogram. **b** R component histogram after pre-processing. **c** G component original histogram. **d** G component histogram after pre-processing. **e** B component original histogram. **f** B component histogram after pre-processing. **g** The merged histogram

and valleys. Figure 3(d), (b) and (f) are the results of the RGB component original histograms after pre-processing. From the comparison between Fig. 3(c) and (d), we can see that the histogram waveform is very smooth after pre-processing, which provides an advantage for positioning the peaks and valleys accurately. The valleys searched by the fast location method are marked by the pink points.

In Fig. 3(b), the histogram is a unimodal distribution; in Fig. 3(d), the histogram, with three valleys, is divided into four categories; while in Fig. 3(f), there is one valley, so the histogram is divided into two categories. We denote the above categories as C_R=1, C_G=4 and C_B=2, respectively. On merging the categories of the three components, eight (L=C_R×C_G×C_B=8) possible clustering centroids are obtained, and the merged histogram is shown in Fig. 3(g). Three valleys are found by using the fast location method shown as the blue points in Fig. 3(g). Next, the histogram is divided into four categories and four initial cluster centroids are finally obtained. The initial clustering results are more accurate and are obtained from the initial cluster centroids directly; these are illustrated in Fig. 2(b). Figure 2(c) is the clustering results obtained by FCM methods; here, we can see that every small object maintains complete contour, and even the dark edge regions show accurate clustering with a clear distinction from the background.

4 Experimental results and analysis

The testing environment selected in our simulation experiments is as follows: Intel Xeon 3.10 GHz, 4G memory, VS2008 and VC++ programming environment. The experiment includes two parts. In Experiment I, qualitative analysis and comparison is made between our algorithm and the HTFCM algorithm; while in Experiment II, we make quantitative analysis and comparison on the BSD300 image database between our algorithm, the HTFCM algorithm and the DSRPCL algorithm.



(a) Sperm (b) Date (c) Teddy

Fig. 4 Testing images for Experiment I. **a** Sperm. **b** Date. **c** Teddy

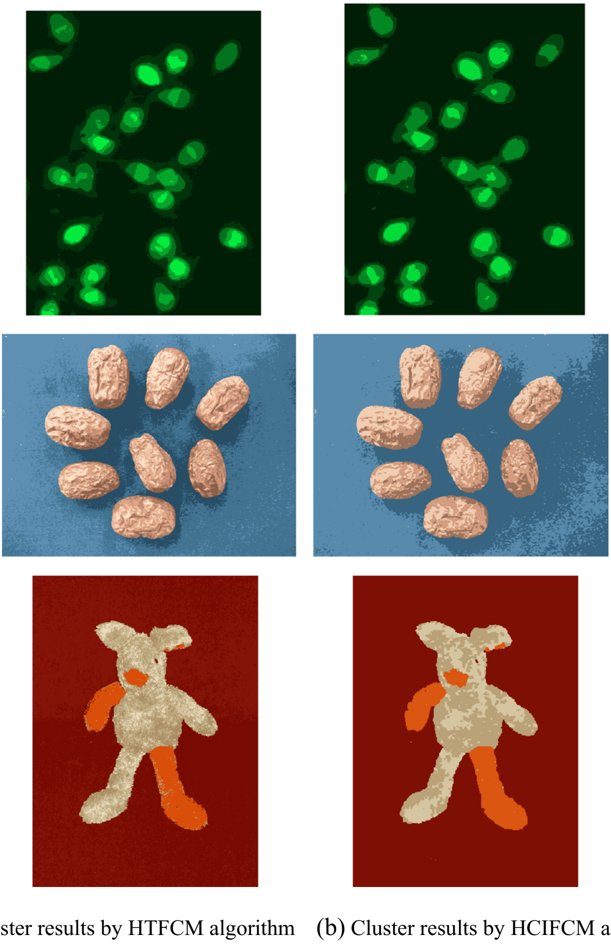

(a) Cluster results by HTFCM algorithm
(b) Cluster results by HCIFCM algorithm

Fig. 5 Experimental results I. **a** Cluster results by HTFCM algorithm. **b** Cluster results by HCIFCM algorithm

4.1 Experimental results I

For Experiment I, the selected testing images are shown in Fig. 4; these are the Sperm image, Date image and the Teddy image with sizes 422×552, 800×600 and 284×398, respectively. Figure 5 shows the cluster results of the testing images in Fig. 4 by HTFCM algorithm and HCIFCM algorithm. Table 1 depicts the performance comparison between the HTFCM algorithm and the HCIFCM algorithm from seven aspects, such as clustering number, number of iterations, time consumed for initialising cluster centroids, time consumed of FCM, total time consumed and ratio of total time consumed. From the table, we can see that compared with the HTFCM algorithm, the clustering time of the HCIFCM algorithm has greatly reduced, and the clustering results are more optimistic with respect to the visual effects. As to why our algorithm has a better performance on running time, there are two reasons: on the one hand, for the clustering centroids initialisation, the main processing object is the histograms, and not all the pixels, which accelerates the process; on the other hand, the WaterShed algorithm is used for the pre-segmentation, and instead of the pixels, the segmentation regions are taken as the

Table 1 Comparison of experimental results I

Image	Clustering approaches	Clustering number	Number of iterations	Time consumed for initialising cluster centroids (ms)	Time consumed of FCM (ms)	Total time consumed (ms)	Ratio of total time consumed
Sperm	HTFCM	5	17	297	3453	3750	1.2
	HCIFCM	4	19	62	3094	3156	
Date	HTFCM	9	19	1125	26672	27797	22.2
	HCIFCM	5	8	204	1046	1250	
Teddy	HTFCM	7	26	890	6532	7422	28.0
	HCIFCM	4	10	62	203	265	

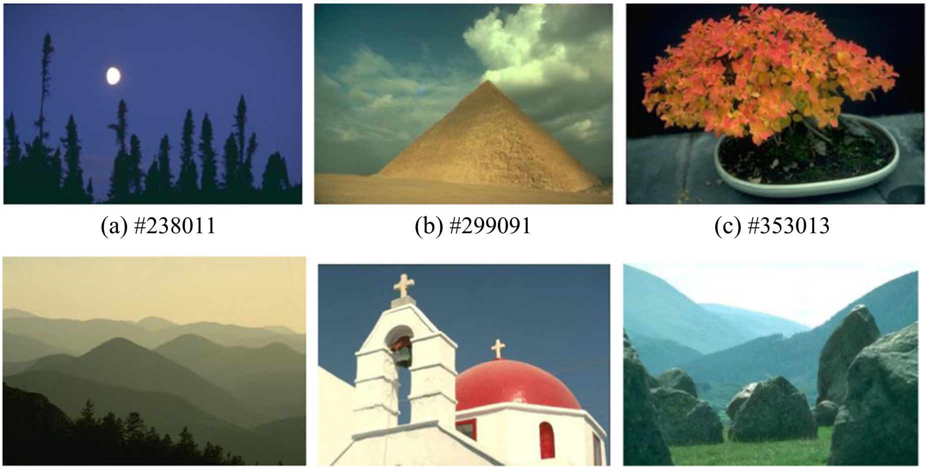
sample data, which greatly reduces the sample number and improves the performance of the FCM algorithm. The proposed method is superior to the HTFCM method in the aspects of efficiency, clustering numbers and iterations due to the accurate initial cluster centroids and robust histogram based multilevel thresholding framework.

4.2 Experimental results II

For Experiment II, we test 300 images derived from Berkeley database (BSD300) and part of the original images are shown in Fig. 6.

In the colour image segmentation, PRI (Probability Rand Index) is a widely used assessment metric [11, 15, 16] for comparing with manually labelled images, such as Ground Truth, to obtain the statistical ratio of correct segmented pixels. The PRI metric is given by,

$$PRI(S_{test},\{S_K\}) = \frac{1}{\binom{N}{2}} \sum_{i,j,i<j} \left[I\left(l_i^{S_{test}} = l_j^{S_{test}}\right) p\left(\hat{l}_i = \hat{l}_j\right) + I\left(l_i^{S_{test}} \neq l_j^{S_{test}}\right) p\left(\hat{l}_i \neq \hat{l}_j\right) \right] \tag{8}$$



(a) #238011 (b) #299091 (c) #353013

(d) #55067 (e) #118035 (f) #241004

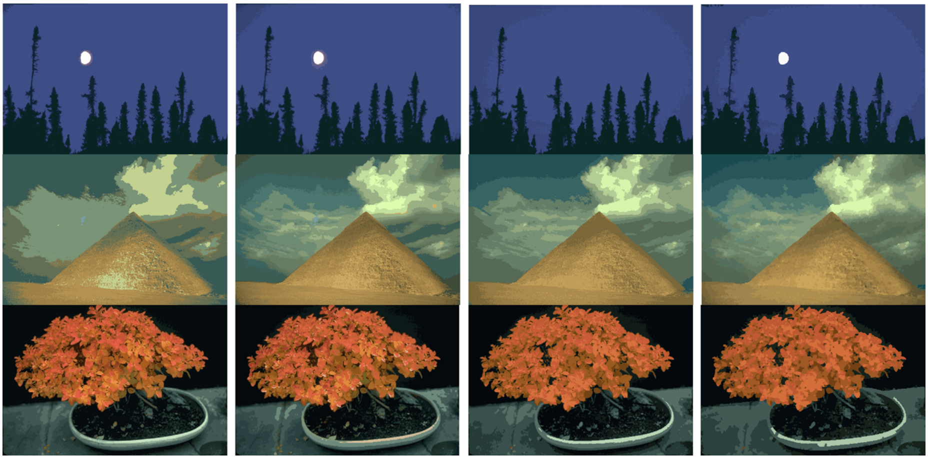
Fig. 6 Testing images for Experiment II. **a** #238011. **b** #299091. **c** #353013. **d** #55067. **e** #118035. **f** #241004

where N is the sample number and l_i is the clustering number of the pixel i. $I(\,)$ denotes a binary function, $I(1)=1$ and $I(0)=0$. S_{test} is the segmentation result and $\{S_K\}$ is the set of manually labelled images. $p\left(\hat{l}_i=\hat{l}_j\right)=\frac{1}{K}\sum_{k=1}^{K} I\left(l_i^{S_k}=l_j^{S_k}\right)$ and $p\left(\hat{l}_i=\hat{l}_j\right)+p\left(\hat{l}_i\neq\hat{l}_j\right)=1$.

According to the authors' suggestions from paper [8], in this experiment, we set $P=0.2$, and the learning rate α of the DSRPCL1 algorithm and that of the DSRPCL2 algorithm to 0.1 and 0.5, respectively; simultaneously, according to the suggestions from paper [14], we set $H=20$, $V=0.008N$ and $dc=28$ in the HTFCM algorithm, and in our algorithm, we set $T_1=10$ and $\beta=30$, both of which are obtained from experience. In addition, the initial clustering number k of the DSRPCL1 algorithm and the DSRPCL2 algorithm are both set to 30.

Figure 7 shows the clustering results of the three testing images (#238011, #299091 and #353013) in Fig. 6, which are respectively obtained by the HTFCM algorithm, the DSRPCL1 algorithm, the DSRPCL2 algorithm and the HCIFCM algorithm proposed by this paper. Compared with other algorithms, for the moon in image #238011, the clustering result of HCIFCM algorithm is more accurate, whether in the edge retention or integrity; for the pyramid in the image #299091, the misclassification of this method is much less, and additionally, the clustering result preserves the sky hierarchy, which is similar to the original image; for the leaves in image #353013, the clustering result is smoother, and the integrity is considerably better.

Table 2 illustrates the performance comparison between the DSRPCL1 algorithm, the DSRPCL2 algorithm, the HTFCM algorithm and the HCIFCM algorithm from six aspects, including the clustering number, the PRI metric, the time consumed for initialising cluster centroids, the time consumed of FCM clustering, the total time consumed and the ratio between the total time consumed. From the table, we can see that the PRI value of the HCIFCM algorithm is much higher and that the clustering results are more accurate. In terms of the time consumed for initialising the cluster centroids, our algorithm is quicker and more stable due to the statistical histograms which are taken as the processing object, instead of the image pixels as in the HTFCM algorithm. In terms of the total time consumed, our algorithm is



(a) DSRPCL1 (b) DSRPCL2 (c) HTFCM (d) HCIFCM

Fig. 7 Comparison of the experimental results. **a** DSRPCL1. **b** DSRPCL2. **c** HTFCM. **d** HCIFCM

Table 2 Comparison of the clustering performance

Images	Clustering approaches	Clustering number	PRI	Time consumed for initialising cluster centroids (ms)	Time consumed of FCM clustering (ms)	Total time consumed (ms)	Ratio between the total time consumed (*/HCIFCM)
#238011	DSRPCL1	7	0.193216	–	6909.47	6909.47	33.8
	DSRPCL2	12	0.108387	–	1592	1592	7.8
	HTFCM	4	0.240437	279.327	955.078	1234.4	6.0
	HCIFCM	5	0.245179	41.6475	162.828	204.476	1
#299091	DSRPCL1	8	0.054484	–	6915.14	6915.14	3.2
	DSRPCL2	30	0.234336	–	1588.45	1588.45	0.7
	HTFCM	12	0.035402	1735.63	9851.29	11586.9	5.3
	HCIFCM	27	0.235242	40.601	2136.31	2176.91	1
#353013	DSRPCL1	29	0.1048	–	6923.52	6923.52	17.0
	DSRPCL2	30	0.023599	–	1608.11	1608.11	3.9
	HTFCM	11	0.037422	3310.03	7062.42	10372.4	25.4
	HCIFCM	9	0.178989	40.4878	367.615	408.103	1
#55067	DSRPCL1	5	0.018462	–	3418.67	3418.67	7.6
	DSRPCL2	17	0.155097	–	1610.47	1610.47	3.6
	HTFCM	6	0.095784	1992.04	438.072	2430.11	5.4
	HCIFCM	13	0.248779	41.419	411.353	452.772	1
#118035	DSRPCL1	10	0.023849	–	6941.96	6941.96	13.0
	DSRPCL2	27	0.024993	–	1609.39	1609.39	3.0
	HTFCM	7	0.233942	3891.8	5908.57	9800.36	18.3
	HCIFCM	14	0.241762	41.4307	493.222	534.653	1
#241004	DSRPCL1	10	0.144042	–	6913.76	6913.76	4.1
	DSRPCL2	16	0.168368	–	1592.39	1592.39	0.9
	HTFCM	9	0.211435	1043.28	4001.74	5045.03	3.0
	HCIFCM	16	0.244882	41.0714	1651.05	1692.12	1

superior to the other three algorithms owing to the superpixel representation. Table 3 gives the average performance comparison on the BSD300 image database. Obviously, our algorithm has more advantages in the overall clustering performance.

Table 3 Mean comparison of clustering performance on BSD300 images

Clustering approaches	PRI Mean	Mean time consumed for initialising cluster centroids (ms)	Mean time consumed of FCM clustering (ms)	Total mean time consumed (ms)	Ratio between the mean and the total mean time consumed (*/HCIFCM)
DSRPCL1	0.112335	–	5850.57	5850.57	17.1
DSRPCL2	0.118709	–	1558.34	1558.34	4.4
HTFCM	0.135357	2836.1	5094.36	7930.46	16.6
HCIFCM	0.228656	41.2436	789.041	830.2846	1

5 Conclusions

In this work, we proposed a histogram-based colour image fuzzy clustering algorithm. This algorithm uses the histogram information in the colour image to initialise the cluster centroids. Next, the FCM clustering algorithm was applied to achieve the colour image segmentation. Compared with the DSRPCL algorithm and the HTFCM algorithm, our algorithm is more effective not only in the execution efficiency but also in the clustering performance. In the stage of cluster centroids initialisation, each component histogram, and not the pixels, is the main processing object. For the FCM clustering, the WaterShed algorithm is used for image pre-segmentation, and the segmented regions are taken as the new sample data, ultimately improving the efficiency of the algorithm greatly. In addition, a more efficient fast peak-valley location scheme is used to locate the valleys accurately, and the valleys are taken as the segmentation thresholds to segment the histogram, after which more reasonable initial cluster centroids can be obtained. Whether the initial cluster centroids obtained by the HTFCM algorithm are reasonable is basically affected by three control parameters, and there is also a degree of irrationality in the division process. For our algorithm, there are fewer parameters, and it is considerably easier to achieve parameter control. However, the parameter selection problem still restricts the clustering performance of our algorithm. Therefore, a non-parametric and automatic colour image clustering algorithm needs to be researched further.

References

1. Berkley Image Database. http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/
2. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8(6):679–698
3. Krinidis S, Chatzis V (2010) A robust fuzzy local information c-means clustering algorithm. IEEE Trans Image Process 19(5):1328–1337
4. Li Y, Sun J, Tang CK et al (2004) Lazy snapping. ACM Trans Graph 23(3):303–308
5. Li CM, Huang R, Ding ZH et al (2011) A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI. IEEE Trans Image Process 20(7):2007–2016
6. Long JW, Shen XJ, Chen HP (2012) Adaptive minimum error thresholding algorithm. Acta Autom Sin 38(7):1134–1144
7. Long JW, Shen XJ, Chen HP (2012) Interactive document images thresholding segmentation algorithm based on images regions. J Comput Res Dev 49(7):1420–1431
8. Ma JW, Wang TJ (2006) A cost-function approach to rival penalized competitive learning (RPCL). IEEE Trans Syst Man Cybern B Cybern 36(4):722–737
9. Mok PY, Huang HQ, Kwok YL et al (2012) A robust adaptive clustering analysis method for automatic identification of clusters. Pattern Recogn 45(8):3017–3033
10. Peng B, Zhang L, Zhang D et al (2011) Image segmentation by iterated region merging with localized graph cuts. Pattern Recogn 44(10–11):2527–2538
11. Peng B, Zhang L, Zhang D (2013) A survey of graph theoretical approaches to image segmentation. Pattern Recogn 46(3):1020–1038
12. Sarkar S, Das S, Chaudhuri SS (2015) A multilevel color image thresholding scheme based on minimum cross entropy and differential evolution[J]. Pattern Recogn Lett 54(1):27–35
13. Shen XJ, Long JW, Chen HP et al (2011) Otsu thresholding algorithm based on rebuilding and dimension reduction of the 3-Dimensional histogram. Acta Electron Sin 39(5):1108–1114
14. Tan KS, Isa NAM (2011) Color image segmentation using histogram thresholding-Fuzzy C-means hybrid approach. Pattern Recogn 44(1):1–15
15. Unnikrishnan R, Pantofru C, Hebert M (2007) Toward objective evaluation of image segmentation algorithms. IEEE Trans Pattern Anal Mach Intell 29(6):929–944
16. Vantaram SR, Saber E (2012) Survey of contemporary trends in color image segmentation. J Electron Imaging 21(4):1–28

17. Vincent L, Soille P (1991) Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans Pattern Anal Mach Intell 13(6):583–598
18. Wei W, Shen XJ, Qian QJ (2011) An adaptive thresholding algorithm based on grayscale wave transformation for industrial inspection images. Acta Autom Sins 37(8):944–953
19. Yu ZD, Au OC, Zou RB et al (2010) An adaptive unsupervised approach toward pixel clustering and color image segmentation. Pattern Recogn 43(5):1889–1906

Haipeng Chen graduated in 2000 from the Jilin University of Technology (now merged in Jilin University), China. He received the M.E. and Ph. D degree from the College of Computer Science and Technology, Jilin University. Then he worked in the College of Computer Science and Technology of Jilin University and is currently an assistant professor. His research interests include multimedia forensics, such as copy-move, splicing, recapture and CG forgery detection.

Xuanjing Shen received the B.E., M.E. and Ph. D degree from Harbin Institute of Technology, China. He became a lecturer in Harbin Institute of Technology and then joined Jilin University of Technology (now merged in Jilin University) in 1992. He is currently a professor and Ph.nstudent advisor in the College of Computer Science and Technology of Jilin University. He has wide research interests including image segment, multimedia forensics, and video understand technology and so on.

Jianwu Long received the M.E. and Ph. D degree from the College of Computer Science and Technology, Jilin University. His research interest covers image processing, computer vision, and pattern recognition.