Image Segmentation Based on Prototypes Extraction and Merging of Clusters in Multiple Spaces

Xiaofeng Shi, Furao Shen* and Jinxi Zhao
The National Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, China
Email: cell000feng@gmail.com, frshen@nju.edu.cn, jxzhao@nju.edu.cn

***Abstract*—Pixel clustering is one of the basic methods for image segmentation. A critical problem of pixel clustering is how to measure the similarity between colors of the pixels on human visual perception. In this paper, we propose an adaptive clustering method for image segmentation, namely Prototypes Extraction and Merging (PEM) method. We first build a prototype network based on the Hebbian learning rule to represent the image. Then we use the Density Peaks Based Clustering (DPBC) method on the prototypes rather than the pixels for clustering in multiple color spaces, in which we choose a tiny Euclidean distance to approximate the similarity of the neighboring prototypes and find the density peaks. Finally we conduct the Multi-Space Merging (MSM) method to merge the color regions in multiple color spaces and get the final segmentation. In PEM the similarity measuring of colors is achieved adaptively according to the complexity and local contrast ratio of the image. Thus, it is extremely close to the discriminability of human visual system.**

I. Introduction

In the area of computer vision, automatic image segmentation is one of the most important research topics [1]. Image segmentation means dividing the image into several disjoint regions based on some features like color, texture, shape and edge. These features are supposed to be similar in the same regions but distinguishing among different regions. The segmentation of the color images is more useful than the grayscale images, and it is also much more complicated and challenging, for color images carry more information [2]. Automatic image segmentation can be applied in many areas, including object recognition, image retrieval, remote sensing image analysis, etc.

By the features used in segmentation, the segmentation methods can be categorized into three major approaches, i.e., feature-space-based clustering, spatial segmentation, and graph-based approaches [3]. Feature-space-based clustering algorithms segment the image by clustering the pixels based on the feature vectors, in which color feature and texture feature are usually used. Yan Y et al. [4] extracts color and texture features of the image and used K-means algorithm to cluster the pixels into several regions. But this kind of methods does not take account of the spatial information of the image and the pixels in disconnected regions might be clustered together if the feature vectors are similar. Spatial segmentation methods use the edge information as well as the spatial information to segment the images [5] [6], and a huge number of small fragmented regions are produced. Then these regions are merged together by some merging algorithms [6] [7]. Graph-based approaches synthesize both of the feature-space-based clustering and spatial segmentation methods. Ugarriza et al. [8] clustered the pixels based on color, then characterizes the texture present in each cluster and generates a final segmentation by utilizing an effective merging procedure considering the spatial and edge information.

No matter which of the methods mentioned above is used, color feature of the images is one of the most significant measurements to decide whether two pieces in the image should be merged into a region, for it is the most dominant feature and can easily be distinguished by humans eye. There are different kinds of color spaces to represent the pixel colors like RGB, HSV, HVC, XYZ and CIE L*a*b* color spaces. RGB is the most primary space for color image storage in computer. While HSV (or HVC) and CIE L*a*b* spaces are most frequently used for color clustering because they are most closely approximate the humans perception on color. Qixiang Ye et al. [9] uses the HVC color space and density-based clustering method for pixels clustering. Hanzi Wang and David Suter [10] applied the mean shift algorithm in the hue and intensity subspace of HSV for segmentation. Ugarriza et al. [8] exploits the information obtained from detecting edges in color images based on the CIE L*a*b* color space. The above methods use the Euclidean distance to measure the similarity of color vectors for clustering. However, in CIE L*a*b* color space as well as many other spaces, small Euclidean distance between two color points is proportional to the difference that human visual system perceives, but a large Euclidean distance has no meaning [11]. In Yan Y's work [4] Mallows distance is used for the similarity measuring in K-medoid Clustering algorithm. However, Mallows distance also has its shortcomings for measuring any two points in the color space, since large distance has no meaning and is incomparable for color. Hence, it is necessary to find a robust distance measure for color pixels clustering.

In years of study on clustering for image pixels, another critical problem is to select a suitable number of the clusters in an image. K-means and K-medoid algorithms are widely used in pixels clustering [4] [12] [13], but the parameter K needs to be decided previously. The colorful images usually needs more color regions to represent them than the relatively monotonous images. Therefore, the parameter K is sensitive for different

Corresponding author is Furao Shen.

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

images. Besides, for the high resolution photo, the number of the pixels is millions, which causes the unacceptable computation complexity. The adjusted self-organizing incremental neural network (SOINN) [14] is a real-time learning method that learns the input data points incrementally. It generates the prototype nodes corresponding to the inputs without any prior knowledge and can be used to build the topology of the input data. This feature of SOINN can be used to construct a network on the pixel distribution. Then we can cluster on the network prototypes rather than the entire pixels, which could be much more efficient. Besides, Alex et al. [15] proposed a clustering method by fast searching and finding of density peaks. The method can find the cluster centers rapidly based on the density distribution, but some efforts need to be done to choose the density peaks by human.

Inspired by the two learning algorithm, we proposed a real-time method for color pixels clustering based on a distance measuring technique that approximates the human vision perception on color, namely Prototypes Extraction and Merging (PEM) method. It adaptively chooses the number of color regions and segments the image automatically. Firstly, we extract the prototypes in the HSV color space by using SOINN. Then we categorize the prototypes by searching the density peaks in all of the RGB, HSV and CIE L*a*b* color spaces. When searching, we set the maximum search radius in a small value to guarantee that the Euclidean distance between two prototypes is proportional to the difference that human visual system perceives in that level. Finally, a merging method is proposed to combine the cluster regions in all of the three color spaces to reduce the number of segmented regions. The proposed method can preliminarily segment the image by color feature. By setting different parameters, the segmentation can be conducted in multi-scales. In the proposed method, only color features are considered. Other features like texture and spatial information can be used to merge or segment the regions in a further processing. Experiments are conducted to examine our method. From the results we find that the method is robust for various images with different categories. Even if only the color features are considered, we can still get some promising segmentation results.

II. The proposed method: Prototypes Extraction and Merging

The proposed segment method is actually a clustering method of the color pixels in an image. It overcomes the problem that the distance between two color points is hard to measure, and can categorize pixels into regions based on the humans sensitivity for colors. The criterion of whether two color points should cluster together and the number of the clusters is adaptively fixed by the contrast of the objects and the complexity of the colors in the input image. And the method also works on the images with high resolution.

Fig.1 shows the process of the PEM Clustering method. Before the image is segmented, a 5×5 Gaussian filter is used to reduce the noise in the image, because some noisy pixels might have very different color from the object pixels, which would lead to generate many useless tiny regions after the segmentation. After that, the Color Prototypes Extracting (CPE) algorithm is used to extract the representative color points from the input pixels. Next, the prototypes are clustered respectively in all of the RGB, HSV and CIE L*a*b* spaces by using the Density Peaks Based Clustering (DPBC) algorithm. During this, a small Euclidean distance radius is used to calculate the density of each prototype to make sure that the correlated nodes are similar in humans perception. As a consequence, a number of cluster regions are produced in each color space. To reduce the number of the regions, we merge the clusters which are correlated among different color spaces together by using the Multi-Space Merging (MSM) algorithm. In the learning system, we do not define a formula for color similarity measuring. Instead, we use the small Euclidean distance to measure the similar colors and produce small clusters. Then the clusters are merged adaptively in multiple spaces, during which similar regions join together and dissimilar regions are distinguished.

A. *Color Prototypes Extracting*

For a high-definition picture with millions of pixels, the computation complexity is unacceptable for clustering with methods like K-means in which the global information of the pixels is used. Therefore, a real-time color prototypes extracting (CPE) algorithm based on SOINN is proposed in this paper. We build a topological network to incrementally learn the input pixels one by one. Prototype nodes are generated and adjusted in the network to represent the pixels with similar colors. The topology of the prototypes can truly manifest the distribution of the color pixels of the image.

The neural network is built based on the HSV color space. We sample all the pixels of the image in a random sequence without replacement. Each pixel is represented by a three-dimensional vector (H, S, V). The pixels in the sequence are inputted to the system one by one to produce the prototypes. Firstly, the system receives two pixels and regards them as two initialized prototypes. Each of the prototypes has its weight and the distance threshold to absorb new pixels. Next, for each new coming pixel p, find its nearest prototype P_1 and the second nearest prototype P_2 in the network. Then there could be two kinds of cases.

Case 1. The pixel p is within the threshold distance of both P_1 and P_2.

This case suggests the pixel is very close to the winner and second winner prototypes, hence it can be absorbed by the winner prototypes, as it is shown in Fig. 2. We update the weight of P_1 and adjust the vector of P_1 by:

$$\overrightarrow{P_1} = \frac{W_{P_1} - 1}{W_{P_1}} \overrightarrow{P_1} + \frac{1}{W_{P_1}} \overrightarrow{p}$$

Where W_{P_1} is the weight of P_1. According to the Hebbian learning rule, both of P_1 and P_2 are activated by p, as a result, the connection between P_1 and P_2 should be stressed. If there is no edge between P_1 and P_2, we need to connect P_1 and P_2 with a new edge. Otherwise, refresh the age of the edge. For

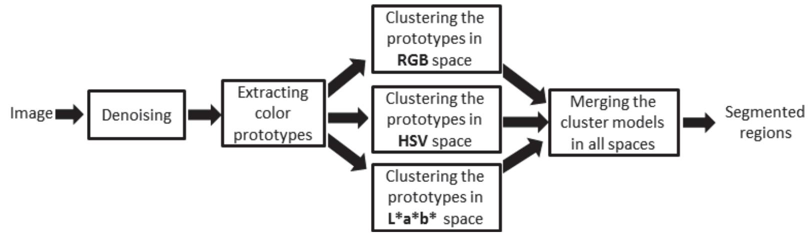

Fig. 1. The process of the proposed method.

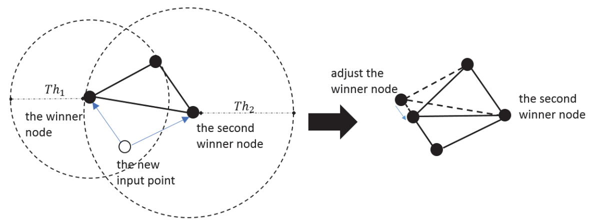

Fig. 2. If new input pixel p is within the threshold distance of both P_1 and P_2. It is absorbed by the winner node P_1.

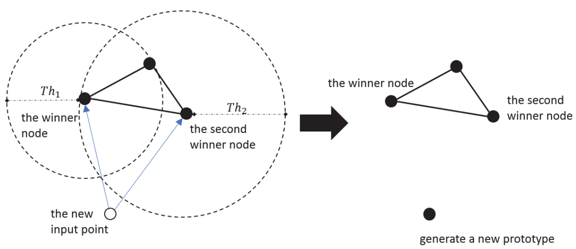

Fig. 3. If new input pixel p is beyond the threshold distance of either P_1 or P_2. A new prototype is generated.

both P_1 and P_2, search for all the other prototypes that are connected with them, and add the homologous edges by 1. If the age of one edge is too old, then remove it, because the correlation between these two prototypes is not close enough and out of time.

Case 2. The pixel p is beyond the threshold distance of either P_1 or P_2.

In this case, there is no prototype that has the similar color as p. Then p is regarded as a new prototype, as it is shown in Fig. 3.

Next, for both cases, update the distance threshold of $P_1(P_2)$ by using the maximum distance between $P_1(P_2)$ and the nodes that are connected with it.

For a natural image, there are always some noisy pixels that would produce isolated prototypes in the network. Hence we take a denoising technique: every time after several iterations are conducted, the isolated prototypes are removed from SOINN as noisy pixels, as it is shown in Fig. 4.

By this way, the construction of SOINN are formed as long as the pixels are inputted. Every prototype in the network is a small cluster of some pixels that are similar in color. Here the HSV color space is used to generate the prototype. Actually, other color spaces like RGB and CIE L*a*b* can also be used because the threshold to judge whether two pixels could join together is a quiet small Euclidean distance in any of the color spaces. Even its hard for human to tell the difference among the pixels in one cluster.

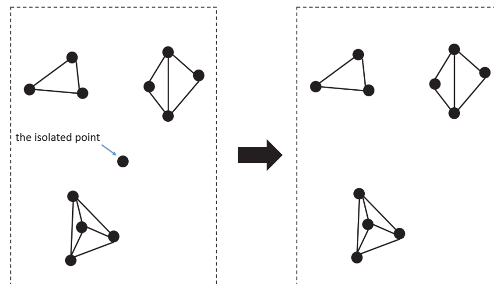

Fig. 4. The isolated prototypes are removed as noise.

The entire algorithm of CPE method is shown in Algorithm 1.

B. Clustering the Color Prototypes in Multiple Spaces Based on Density Peaks

No matter in which of the color spaces, the similarity of colors by human visual perception is not linearly dependent on the distance between color points. Especially for the dissimilar colors, it is meaningless to calculate the exact distance in space and compare to others. For example, we could be confused to tell whether red is more dissimilar to yellow or green. Moreover, a color measuring model is not suitable for different images. For another example, black and charcoal grey could be quiet similar to each other when compared with white in one image, while they could also be significantly different in another image with dark tone. Hence, it is hard to build a universal color measuring model. However, small Euclidean distance between two color points is proportional to the difference that human visual system perceives [11]. At this point, we could only use the Euclidean distance between neighboring color prototypes to approximate the similarity of the prototypes. Inspired by Rodriguez A's work [15], we propose the Density Peaks Based Clustering (DPBC) method to cluster the color prototypes by finding the density peaks. The density of each prototype is calculated in a small Euclidean distance threshold to ensure that all the density peaks are the representations of some very similar prototypes in human visual system.

Algorithm 1 Color Prototypes Extracting

Require: An array of the pixel points $PArray$ sampled from the original image without replacement in a random sequence. The threshold value of edge age $agemax$. The cycle size λ for noise reduction.

Ensure: A set of the learned prototype nodes P for the color space representation.

1: Initialize two prototype points

$$P_1 = PArray_1, P_2 = PArray_2$$

Add P_1 and P_2 to P; Initialize the threshold of P_1 and P_2.

$$Th_1 = dis(P_1, P_2), Th_2 = dis(P_1, P_2)$$

Where $dis(P_1, P_2)$ is the Euclidean distance. Initialize the weight and edge of P_1, P_2.

$$W_1 = 1, W_2 = 1, e_{1,2} = 0, a_{1,2} = 0$$

Initialize $E = \Phi$. Add $e_{1,2}$ to E.

2: Input a new pixel point q from PArray

$$S_1 = argmin_{p \in P} dis(p, q)$$

$$S_2 = argmin_{p \in P/\{S_1\}} dis(p, q)$$

3: If $dis(q, S_1) > Th_1$ or $dis(q, S_2) > Th_2$, then add q to P, initialize $W_q = 1$.

4: Else, $W_{S_1} = W_{S_1} + 1$. If there is no edge between S_1 and S_2, build edge connection e_{s_1,s_2} and add it to E, initialize the age a_{s_1,s_2} of e_{s_1,s_2} as 1; else update a_{s_1,s_2} as 1.

5: For all the edge connections e_{x,s_1} in E related to S_1, $a_{x,s_1} = a_{x,s_1} + 1$. If $a_{x,s_2} > agemax$, remove e_{x,s_1} from E.

6:

$$S_1 = S_1 + \frac{1}{W_{S_1}} \times (q - S_1)$$

7: End else

8:

$$Th_{S_1} = max_{n \in N_{S_1}} dis(n, S_1)$$

$$Th_{S_2} = max_{n \in N_{S_2}} dis(n, S_2)$$

Where N_{S_1} and N_{S_2} are the sets of the nodes that are connected with S_1 and S_2. If N_{S_1} or N_{S_2} is empty, then $Th_{S_1} = dis(S_1, S_2)$ or $Th_{S_1} = dis(S_1, S_2)$.

9: Noise reduction. For every λ times of iteration, remove the single prototype nodes from P. A node p_x is a single prototype node if it has no connection with any other nodes and

$$W_{p_x} < \sum_{i=1}^{n} \frac{W_i}{n}$$

where n is the number of the generated prototype nodes.

10: Repeat step 5 to 9 until all the pixel points in $PArray$ have been calculated.

Firstly, for each prototype p_i in the prototype set P that is generated by CPE, we calculate its density ρ_i by using the distance threshold parameter δ_0:

$$\rho_i = \sum_{p_j \in P} \chi \left(dis \left(p_i, p_j \right) - \delta_0 \right)$$

where $dis\left(p_i, p_j\right)$ means the Euclidean distance between p_i and p_j, $\chi(x) = 1$ if $x < 0$ and $\chi(x) = 0$ if $x > 0$. In the equation δ_0 is a threshold to judge whether p_i and p_j are similar in human visual perception and should be a small value.

Then for each node p_i, calculate the minimum distance δ_i between p_i and the other prototypes that have higher density than p_i, and record the winner prototype p_j as the Parent Node of p_i in a tree graph.

$$j = argmax_{\rho_j > \rho_i} dis \left(p_i, p_j \right)$$

$$\delta_i = dis(p_i, p_j)$$

If ρ_i is the maximum density, then $\delta_i = +\infty$.

Next we use δ_0 as the threshold to pick up the density peaks. If $\delta_i > \delta_0$, we think p_i as a cluster center point and add it to the center point set C.

In different from the original clustering algorithm (Rodriguez A, 2014), we do not use another density threshold ρ_0 to limit the center points for the denoising purpose. Because all the prototypes are the representations of the pixels in the image generated by CPE algorithm, in which the noisy pixels have already been removed. An isolated prototype may also represent the color feature of a small region in the image. Therefore an isolated prototype could also be the center of a cluster in which only a few or one prototype(s) are contained.

After all of the prototypes are processed, we could get a forest built up by the prototypes. The center points in C are the roots of the search trees and other nodes are associated with its Parent Node. Actually, every tree in the forest forms a cluster in the space. Then we could use the breadth-first searching algorithm (BFS) to search the forest and get all the clusters.

C. Merging the clusters in different color spaces

In the DPBC method, to make sure the Euclidean distances between prototypes could approximate the difference of colors, we use a small threshold value δ_0 to calculate the density and pick up the cluster centers. As a consequence, it produces a large number of cluster centers and the image is over segmented. But we know that a color point has different features like RGB(red, green, blue) values, HSV(hue, saturation, value) values and LAB (lightness, color-opponent dimensions) values. The pixel points in an image have different coordinate distributions in different feature spaces. Thus we could merge the segmented regions based on the clustering results in different color space. We propose the Multi-Space Merging(MSM) method, in which we use the three most representative spaces, RGB, HSV and CIE L*a*b*, to merge the clusters.

For any two prototypes in one cluster, we think there is a binary relationship between these two points. After we conduct DPBC in all of the three color spaces, we could get a number of prototype clusters. If we connect any two of the prototypes in one cluster with a binary relationship, a network composed of the prototypes are generated. The main idea of MSM is to find all the connected sub-graphs in the network graph. However, finding the connected sub-graphs directly is not robust. Because between-class distances of the clusters are quiet small in practice. Small overlap might also exist between dissimilar clusters, which would lead to the merging of these regions. Therefore, in MSM we use the region growing method to obtain the connected sub-graphs. When merging two cluster sets, to avoid the influence of noise, we introduce a concept of Color Overlap Rate between the two pieces of regions. The Color Overlap Rate r_{ij} between two cluster sets c_i and c_j can be defined by:

$$r_{ij} = \frac{size(c_i \cap c_j)}{min(size(c_i), size(c_j))}$$

Where $size(c)$ is the number of prototypes in cluster c. If and only if $r_{ij} > r_0$, we think region c_i and c_j are closely similar in color and combine them together, where r_0 is a parameter.

The whole MSM algorithm can be described as Algorithm 2:

Algorithm 2 Multi-Space Merging

Ensure: The merged region set: C

1: Use the DPBC in all of the RGB, HSV, CIE L*a*b* spaces and produce three cluster region sets. The number of prototypes for clustering is N_p.

$$C_i = \left\{ \left\{ p^i_{x_1}, p^i_{x_2}, ... \right\}, \left\{ p^i_{y_1}, p^i_{y_2}, ... \right\}, ... \right\} i = 1, 2, 3$$

2: Initialize set C by combining the cluster region sets together and calculate the total number of the prototype points N_C in C:

$$C = C_1 \cup C_2 \cup C_3, N_C = \sum_{c \in C} size(c)$$

3: While $N_C > N_p$ and N_C is still changing, do:

4: For any two clusters c_i and c_j in C, if $r_{ij} > r_0$, merge c_i and c_j by:

$$c_{ij} = c_i \cup c_j$$

Remove c_i and c_j from C and add c_{ij} into C. Update N_C by:

$$N_C = \sum_{c \in C} size(c)$$

5: end while

After the merging process, the number of segmented regions declined significantly. However, some isolated prototypes might still exist in the graph. These prototypes could also produce tiny regions and contribute to the complexity of the segmentation. Thus we put every isolated prototype into the cluster set which has the closest prototype to it. In fact, the elimination of the isolated prototypes just causes imperceptible changes in the segmented images.

D. Parameter determination

In the PEM clustering algorithm for image segmentation, several parameters are needed:

1. threshold value of edge age $agemax$ and the cycle size λ for noise reduction in CPE

2. the distance threshold parameters δ_0^{RGB} for the RGB space, δ_0^{HSV} for the HSV space and δ_0^{LAB} for the CIE L*a*b* space in DPBC

3. the minimum overlap rate r_0 in MSM.

According to the CPE algorithm, the parameter $agemax$ determines the number of edges in the topological graph of prototypes. When a prototype is evoked by a newly inputed pixel point, the ages of the edges that are associated with the winner prototype or the second winner prototype should be updated. We use this parameter to remove the edges that are too old. The endpoints of an old edge might be both evoked by an input data point once but never resonate to each other later. Hence, according to Hebbian rule, these inactive edges should be removed. On the other hand, removing edges causes isolated prototypes in the network. The parameter λ determines how often to remove the isolated prototypes for the purpose of de-noising. Therefore, λ is positively correlated to the number of prototypes in the network. In fact, the prototype network is generated adaptively. It reflects the distribution of the original data set. Therefore, the whole system is not sensitive to the two parameters. Usually, an image contains more than 10000 pixels. The CPE receives the pixels one by one, which means it needs more than 10000 rounds of iteration to build the network. Consequently, these two parameters could be set as tens or hundreds. In our experiments, we set $agemax = 200$ and $\lambda = 200$.

In DPBC, the parameters δ_0^{RGB}, δ_0^{HSV}, δ_0^{LAB} are the criterions of RGB, HSV and CIE L*a*b spaces, to estimate whether two color points are similar in human visual sense. If the Euclidean distance between two color points is smaller than any of the three parameters in the coordinate space, we regard the two colors as similar colors. Therefore, the parameters are determined according to human nature sense and could be regarded as constants. If the parameters are too large, dissimilar color prototypes might be categorized into one cluster. On the other hand, if the parameters are too small, a great number of cluster center points are produced in DPBC, which would result in over-segmentation. By the observation of the experiment results, we set $\delta_0^{RGB} = 0.04$ in the normalized RGB space, $\delta_0^{HSV} = 0.04$ in the normalized HSV space and $\delta_0^{LAB} = 6$ in the CIE L*a*b* space. The DPBC method is robust with the settings.

In MSM the minimum overlap rate r_0 is the parameter to estimate whether two clusters with overlap prototypes could be merged together. It is in the range of $(0, 1]$. By setting different values of the parameter, we can obtain hierarchical segmentation results in different region granularities. Hence

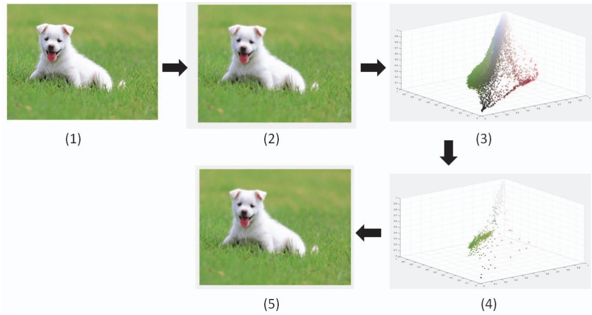

Fig. 5. The result of CPE: (1) the original image with the resolution of 531×407; (2) the preprocessed image; (3) the distribution in RGB space of 80000 pixels randomly extracted from (2); (4) the distribution in RGB space of the prototypes generated by CPE; (5) the image represented by 868 prototypes.

this parameter can be decided by users for the requirement of further segmentation and merging.

III. Experiments and results

We apply the proposed method on various natural color images. In Fig.5, we show the result of CPE. Fig.5(1) is the original image with the resolution of 531×407. Fig.5(2) is the result of preprocessed image with a 5×5 Gaussian filter. We extract about 80000 pixels from Fig.5(2) in a random sequence and the distribution of the pixels in the RGB space is shown in Fig.5(3). The pixels are subject to Gaussian distribution. The pixels in the main color kinds (like white and green block) are in the dense regions, while those in the minor color kinds (like red and black) are in the sparse regions. Then we apply CPE algorithm on the pixel sequence and the generated prototypes are shown in Fig.5(4). In Fig.5(4), 868 prototypes are produced and we can find out that the density distribution of the prototypes is very approximate to the original distribution. Then we replace every pixel in Fig.5(2) with the most similar prototype measured by Euclidean distance, and the produced image is shown in Fig.5(5). Comparing Fig.5(5) with Fig.5(2), we can hardly tell the difference between the two images by human vision, even though we use only 868 different prototypes to represent the image in Fig.5(5).

Then we apply DPBC on the prototypes for clustering in RGB, HSV and CIE L*a*b* spaces. The algorithm separately produces 86, 137, 45 center points in the three spaces. In Fig.6(b), (c), (d), we show the images represented by the center points in the three spaces.

Since we use a small Euclidean distance threshold to calculate the density of the prototypes in DPBC, a large number of center points are produced. Then we merge the regions of the three spaces by using MSM and we get the result Fig.6(e). In Fig.6(e), we set $r_0 = 0.05$, and the image is segmented into 10 color regions. From the result, we can see the dog is clearly segmented from the background, and the small parts of the dog (like the eyes, tongue, mouth, nose and ears) are also clearly segmented.

We apply the proposed method on more complex images from BSDS500. Fig.7 and Fig.8 show a group of results

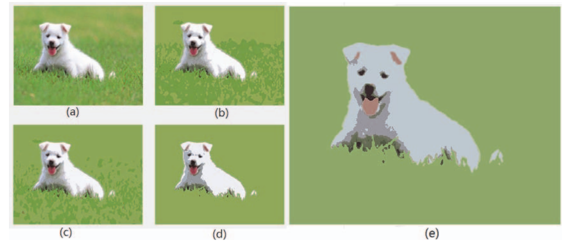

Fig. 6. The result of DPBC: (a) the image represented by 868 prototypes; (b) the image represented by 86 center points generated in RGB space; (c) the image represented by 137 center points generated in HSV space; (d) the image represented by 45 center points generated in CIE L*a*b* space; (e) the segmentation result after applying MSM: the image is segmented into 10 color regions.

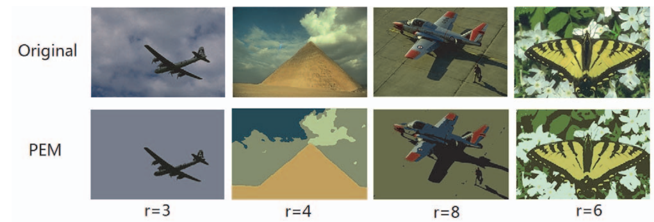

Fig. 7. The segmentation results of the relatively monotonous images in BSDS500.

where we set $r_0 = 0.25$. r is the number of color regions in the segmentation results, and it is determined automatically according to the complexity of the image. In Fig.7, the images are relatively monotonous with a few kinds of colors. Therefore, they are segmented into to a few color regions. On the other hand, in Fig.8, the images are more colorful and they are composed with more color regions. This results show that PEM could learn and segment the images adaptively without set the target number of regions.

And we also compared with Mean Shift algorithm, which is most widely used for clustering in color images. Many image segmentation methods [3] use Mean Shift algorithm to cluster the pixels into regions and then merge the regions. Fig.9 shows the results of the comparison, and we can find out that with the similar number of color regions, PEM performs better with more obvious boundary between different parts, especially those in similar color. Moreover, Mean Shift needs a sensitive

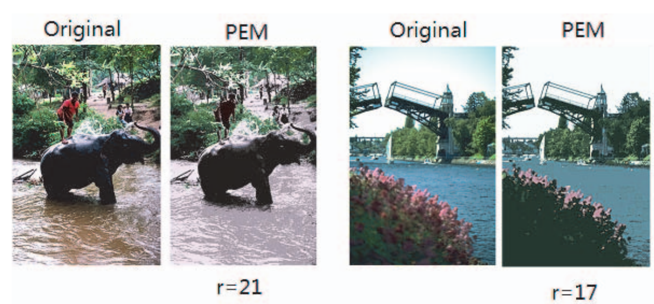

Fig. 8. The segmentation results of the more colorful images in BSDS500.

Fig. 9. The comparison results with Mean Shift algorithm. The figures in first row are the original images in BSDS500. The second row shows the results of clustering by using Mean Shift. The third row shows the results of PEM.

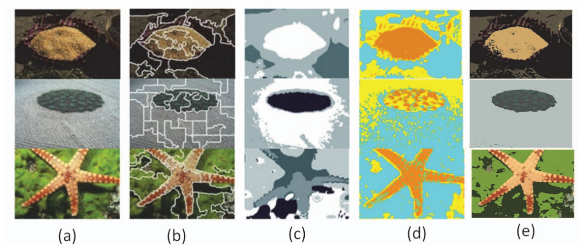

Fig. 10. Comparison with Edge flow, Perceptual color and K-medoid Clustering on BSDS500. (a) Original image, (b) Edge flow, (c) Perceptual color, (d) K-medoid Clustering, (e) PEM.

parameter band width threshold. To generate promising results, the parameter needs to be adjusted for different images. But in PEM, the number of color regions is determined adaptively. The parameter r_0 only controls the region granularities. For different images, we do not need to adjust r_0 if the region granularities is the same.

In Fig.10 and Fig.11, we make a comparison with other state-of-art methods that are reported in [4] and [8] works. We set the same parameter $r_0 = 0.25$ for PEM in all of these results (witch we suppose is the universal optimal parameter to produce the segmentation results with the same granularity as other methods do). In the results, some of the methods cause over segmented problem with too much meaningless color regions, while other methods sometimes cannot distinguish the different objects with similar colors. But our proposed method could clearly separate different objects without producing much over-segmented regions, and the results is very close to human visual discriminability.

IV. Conclusion

In this paper, we propose the PEM method to segment the image by investigating the distance measuring method between colors in human visual perception. We first use the CPE algorithm to extract prototypes of the pixels in an image according to the Hebb learning rule. Then we conduct the DPBC method for the prototype clustering in RGB, HSV and CIE L*a*b* color spaces. The clusters in the three spaces are merged together subsequently to reduces the number of regions. Consequently, the image is segmented based on the

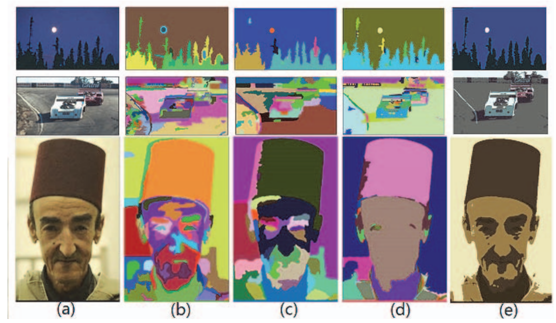

Fig. 11. Comparison with GRF, JSEG and GSEG on BSDS500. (a) Original image, (b) GRF, (c) JSEG, (d) GSEG, (e) PEM.

color features of the objects. The proposed method could segment the image adaptively without setting the number of regions. The experiments show PEM is robust to various images and is competitive against many other state-of-art methods.

Acknowledgment

This work is supported in part by the National Science Foundation of China under Grant Nos. (61375064, 61373001) and Jiangsu NSF grant (BK20131279).

References

[1] Pal N R, Pal S K. A review on image segmentation techniques[J]. Pattern recognition, 1993, 26(9): 1277-1294.
[2] Cheng H D, Jiang X H, Sun Y, et al. Color image segmentation: advances and prospects[J]. Pattern recognition, 2001, 34(12): 2259-2281.
[3] Tao W, Jin H, Zhang Y. Color image segmentation based on mean shift and normalized cuts[J]. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2007, 37(5): 1382-1389.
[4] Yan Y, Shen Y, Li S. Unsupervised color-texture image segmentation based on a new clustering method[C]//New Trends in Information and Service Science, 2009. NISS'09. International Conference on. IEEE, 2009: 784-787.
[5] Grau V, Mewes A U J, Alcaniz M, et al. Improved watershed transform for medical image segmentation using prior information[J]. Medical Imaging, IEEE Transactions on, 2004, 23(4): 447-458.
[6] Haris K, Efstratiadis S N, Maglaveras N, et al. Hybrid image segmentation using watersheds and fast region merging[J]. Image Processing, IEEE Transactions on, 1998, 7(12): 1684-1699.
[7] Makrogiannis S, Economou G, Fotopoulos S. A region dissimilarity relation that combines feature-space and spatial information for color image segmentation[J]. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2005, 35(1): 44-53.
[8] Ugarriza L G, Saber E, Vantaram S R, et al. Automatic image segmentation by dynamic region growth and multiresolution merging[J]. Image Processing, IEEE Transactions on, 2009, 18(10): 2275-2288.
[9] Ye Q, Gao W, Zeng W. Color image segmentation using density-based clustering[C]//Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. IEEE, 2003, 3: III-345-8 vol. 3.
[10] Wang H, Suter D. Color image segmentation using global information and local homogeneity[C]//Proceeding of 7th Conf. of Digital Image Computing: Techniques and Applications. 2003.
[11] Yoon K J, Kweon I S. Color image segmentation considering human sensitivity for color pattern variations[C]//Intelligent Systems and Advanced Manufacturing. International Society for Optics and Photonics, 2001: 269-278.
[12] Pappas T N. An adaptive clustering algorithm for image segmentation[J]. Signal Processing, IEEE Transactions on, 1992, 40(4): 901-914.

[13] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Ssstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods[J], IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, p. 2274 - 2282, May 2012.
[14] Shen F, Hasegawa O. A fast nearest neighbor classifier based on self-organizing incremental neural network[J]. Neural Networks, 2008, 21(10): 1537-1547.
[15] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.