Fast Image Segmentation Based on K-Means Clustering with Histograms in HSV Color Space

Tse-Wei Chen [1], Yi-Ling Chen [2], Shao-Yi Chien [3]
Media IC and System Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University
BL-421, No. 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan
{[1]variant, [2]yipaul, [3]sychien}@media.ee.ntu.edu.tw

***Abstract*— A fast and efficient approach for color image segmentation is proposed. In this work, a new quantization technique for HSV color space is implemented to generate a color histogram and a gray histogram for K-Means clustering, which operates across different dimensions in HSV color space. Compared with the traditional K-Means clustering, the initialization of centroids and the number of cluster are automatically estimated in the proposed method. In addition, a filter for post-processing is introduced to effectively eliminate small spatial regions. Experiments show that the proposed segmentation algorithm achieves high computational speed, and salient regions of images can be effectively extracted. Moreover, the segmentation results are close to human perceptions.**

I. Introduction

Color image segmentation [1], whose purpose is to decompose an image into meaningful partitions, is widely applied in multimedia analysis. Depending on different applications, various kinds of techniques, including feature-space methods and spatial-domain methods, are used for color image segmentation. Feature-space methods, such as clustering, intend to classify pixels to different groups in a pre-defined color space, whereas spatial-domain methods, such as regions growing, manipulate pixels to form connected regions. Since feature-based methods and spatial-domain methods both have their strengths, algorithms which combine two methods are developed [2], [3]. Among them, K-Means clustering is often applied as an essential step in the segmentation process.

As a traditional clustering algorithm, K-Means is popular for its simplicity for implementation, and it is commonly applied for grouping pixels in images or video sequences. However, the quality of K-Means suffers from being confined to run with a fixed value of K rather than dynamically adjusted value of K [4]. There are solutions that run K-Means many times to find a suitable number of K [2], [5], but it is time-consuming. In addition, random initialization of centroids (cluster centers) makes the results different each time. Therefore, a fast and efficient algorithm for K-Means image segmentation is proposed to handle these problems.

On the other hand, choices of color space may have significant influences on the result of image segmentation. There are many kinds of color space, including RGB, YCbCr, YUV, HSV, CIE L*a*b*, and CIE L*u*v*. Although RGB, YCbCr, and YUV color spaces are commonly used in raw data and coding standards, they are not close to human perceptions. Besides, CIE color spaces are perceptually uniform but inconvenient since they require complicated computations. HSV (Hue, Saturation, Value) [6], which is shown to have better results for image segmentation than RGB color space [5], [7], is capable of emphasizing human visual perception in hues and has an easily invertible transform from RGB [8]. Nevertheless, to our best knowledge, clustering in HSV color space is still an open question since there is no appropriate criteria for separation or distance measurement of gray and color pixels. Based on these observations, a new method that combines gray and color histogram bins for clustering in HSV color space is developed in this paper.

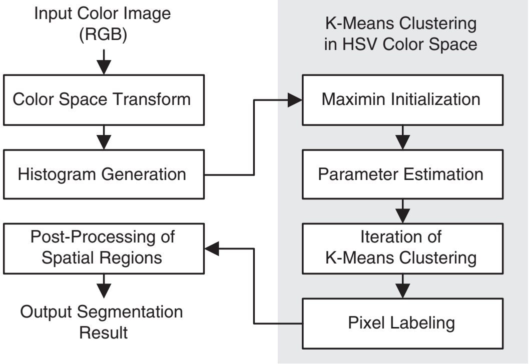

Input Color Image
(RGB)
Color Space Transform
Histogram Generation
Post-Processing of Spatial Regions
Output Segmentation Result
K-Means Clustering in HSV Color Space
Maximin Initialization
Parameter Estimation
Iteration of K-Means Clustering
Pixel Labeling

Fig. 1. Overview of the proposed algorithm.

In brief, the proposed technique has three main advantages for the segmentation process: one is to reduce the computation complexity of K-Means by using histogram quantization in HSV color space, another is to efficiently estimate the number of cluster of K-Means without testing each value of K; the other is to consider gray and color histogram bins together for the clustering procedure. Furthermore, the algorithm integrates a filter to efficiently alleviate over-segmentation, and the results of segmentation are close to human perception. Due to these advantages, it is not only useful for processing large amount of image data but also suitable for embedded systems which have few computational resources.

The paper is organized as follows. The proposed algorithm

978-1-4244-2295-1/08/$25.00 © 2008 IEEE

TABLE I

Correspondence of histogram bins and HSV indices.

Histogram bin	Range of parameter	Correspondence of index
$G(v)$	$v' = 0$ or $v' \in [1, 7]$ and $s' = 0$	$v = v'$
$B(h, s, v)$	$h' \in [0, 29]$ $s' \in [1, 7]$ $v' \in [1, 7]$	$h = h'$ $s = s'$ $v = v'$

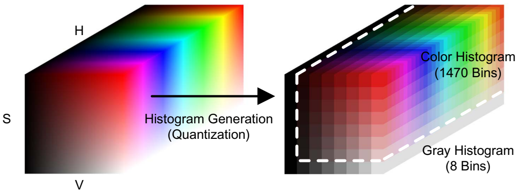


Fig. 2. Histogram generation process in HSV color space.

is first described in Sec. II. Next, in Sec. III, the experimental results will be shown. Finally, a short conclusion is given in Sec. IV.

II. Proposed Algorithm

An overview of the proposed algorithm can be illustrated in Fig. 1, the steps in which will be explained in the following sections.

A. Color Space Transform and Histogram Generation

HSV color space is chosen for the proposed algorithm, and the transform from RGB to HSV in [8] is adopted. Since the ranges of three dimensions in HSV color space are not the same (Hue: $[0, 360°]$, saturation: $[0, 1]$, and value: $[0, 1]$), a quantization process is performed to normalize the values in each dimension:

$$h' = \lfloor \text{Hue}/h_Q \rfloor , \tag{1}$$

$$s' = \lfloor \text{Saturation}/s_Q \rfloor , \tag{2}$$

$$v' = \lfloor \text{Value}/v_Q \rfloor , \tag{3}$$

where (h', s', v') denotes the quantization index, and the quantization parameters $h_Q = 12°$, $s_Q = 0.125$, $v_Q = 0.125$ are set empirically to accentuate the importance of hue and to save computational costs. Therefore, the HSV color space are divided into $30 \times 8 \times 8 = 1920$ partitions. However, the hue of pixels with low saturation is often meaningless since their color is close to gray, and it is suggested that color histogram bins and gray histogram bins should be separated for better color representation [5], [9]. $G(v)$ represents the number of pixel in the gray histogram bin with parameter v, whereas $B(h, s, v)$ represents the number of pixel in the color histogram bin with parameters (h, s, v). The correspondence of these parameters and the quantization indices (h', s', v') are summarized in Table I. Totally N_G gray histogram bins and N_B color histogram bins are generated, where $N_G = 8$ and $N_B = 30 \times 7 \times 7 = 1470$. The process of histogram generation and quantization is illustrated in Fig. 2.

B. Maximin Initialization and Parameter Estimation

In traditional K-Means algorithm, cluster number K is often specified by users, and the initial centroid positions are chosen randomly. In the proposed method, the parameters, including cluster number and the initial positions of centroids, are all estimated though the Maximin algorithm [3] in a systematic approach. Three steps of the proposed Maximin algorithm, which are used in the preliminary stage for K-Means, are briefly stated as follows:

Step A: From the color histogram bins and gray histogram bins, find the bin which has the maximum number of pixels to be the first centroid.

Step B: For each remaining histogram bin, calculate the min distance, which is the distance between it and its nearest centroid. Then the bin which has the maximum value of min distance is chosen as the next centroid.

Step C: Repeat the process until the number of centroid equals to K_{Max} or the maximum value of the distance in **Step B** is smaller than a predefined threshold Th_M. The threshold $K_{Max} = 10$ is set based on the assumption that there should be no more than 10 dominant color in one image for high-level image segmentation, and $Th_M = 25$ is set empirically according to the human perceptions of different color in HSV color space. The distance measurements of histogram bins and centroid vectors will be explained in Sec. II-C.

C. K-Means Clustering in HSV Color Space

The proposed K-Means clustering in HSV color space includes five steps, which are listed as follows:

Step 1: Estimate the parameters of K-Means, including suitable value of K and the position of K initial centroids from the Maximin algorithm in Sec. II-B.

Step 2: Two kinds of histogram bins will be clustered together in this step. For color histogram bins, since the hue dimension is circular (e.g. $0° = 360°$), the numerical boundary should be considered in the distance measurement and the process of centroid calculations. The distance measurement between a histogram bin vector $\boldsymbol{B}_i = (h_i, s_i, v_i)$ and a cluster centroid vector $\boldsymbol{C}_j^{(t)} = (h_j^{(t)}, s_j^{(t)}, v_j^{(t)})$ in the current iteration t is defined in the form of the Euclidean distance (2-Norm):

$$D^2(\boldsymbol{B}_i, \boldsymbol{C}_j^{(t)}) = D_h^2(h_i, h_j^{(t)}) + (s_i - s_j^{(t)})^2 + (v_i - v_j^{(t)})^2, \tag{4}$$

where

$$D_h^2(h_i, h_j^{(t)}) = \begin{cases} (\frac{360°}{h_Q} - |h_i - h_j^{(t)}|)^2, & \text{if } |h_i - h_j^{(t)}| > \frac{180°}{h_Q} \\ (h_i - h_j^{(t)})^2, & \text{otherwise.} \end{cases} , \tag{5}$$

Next, classify each color histogram bin to its nearest cluster centroid by the distance measurement. Thus the membership function of histogram bin $\boldsymbol{B}_i$ is defined by

$$\phi_{(j|\boldsymbol{B}_i)}^{(t)} = \begin{cases} 1, & \text{if } j = \arg\min_k D^2(\boldsymbol{B}_i, \boldsymbol{C}_k^{(t)}) \\ 0, & \text{otherwise.} \end{cases} . \tag{6}$$

On the other hand, for gray histogram bins, there is no hue information inside. Thus the distance measurement between a gray histogram bin $\boldsymbol{G}_i = (v_i)$ and a cluster centroid vector $\boldsymbol{C}_j^{(t)} = (h_j^{(t)}, s_j^{(t)}, v_j^{(t)})$ is different from (4):

$$D^2(\boldsymbol{G}_i, \boldsymbol{C}_j^{(t)}) = (s_j^{(t)})^2 + (v_i - v_j^{(t)})^2, \tag{7}$$

which means that the saturation values of gray histogram bins are all considered as zero, and the hue values can be arbitrary. Besides, the membership function of the gray histogram bin $\boldsymbol{G}_i$ is of the same form as (6), where $\boldsymbol{B}_i$ is replaced by $\boldsymbol{G}_i$.

Step 3: Recalculate and update K cluster centroids. Again, since the hue dimension is circular, the indices in the hue dimension should be considered not absolutely but relatively. An efficient method is introduced to calculate the relative hue index of the original hue index h_i to the centroid $\boldsymbol{C}_j^{(t)} = (h_j^{(t)}, s_j^{(t)}, v_j^{(t)})$:

$$\tilde{h}_{i,j}^{(t)} = \begin{cases} h_i - \frac{360^\circ}{h_Q}, & \text{if } |h_i - h_j^{(t)}| > \frac{180^\circ}{h_Q} \text{ and } h_j^{(t)} < \frac{180^\circ}{h_Q} \\ h_i + \frac{360^\circ}{h_Q}, & \text{if } |h_i - h_j^{(t)}| > \frac{180^\circ}{h_Q} \text{ and } h_j^{(t)} > \frac{180^\circ}{h_Q} \\ h_i, & \text{otherwise.} \end{cases}, \tag{8}$$

and the values in each dimension of all centroid vectors for the next iteration $\boldsymbol{C}_j^{(t+1)}$ are updated according to the following equations:

$$h_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} \tilde{h}_{i,j}^{(t)} \phi_{(j|\boldsymbol{B}_i)}^{(t)} B(\boldsymbol{B}_i)}{\sum_{i=1}^{N_B} \phi_{(j|\boldsymbol{B}_i)}^{(t)} B(\boldsymbol{B}_i)}, \tag{9}$$

$$s_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} s_i \phi_{(j|\boldsymbol{B}_i)}^{(t)} B(\boldsymbol{B}_i)}{\sum_{i=1}^{N_B} \phi_{(j|\boldsymbol{B}_i)}^{(t)} B(\boldsymbol{B}_i) + \sum_{i=1}^{N_G} \phi_{(j|\boldsymbol{G}_i)}^{(t)} G(\boldsymbol{G}_i)}, \tag{10}$$

$$v_j^{(t+1)} = \frac{\sum_{i=1}^{N_B} v_i \phi_{(j|\boldsymbol{B}_i)}^{(t)} B(\boldsymbol{B}_i) + \sum_{i=1}^{N_G} v_i \phi_{(j|\boldsymbol{G}_i)}^{(t)} G(\boldsymbol{G}_i)}{\sum_{i=1}^{N_B} \phi_{(j|\boldsymbol{B}_i)}^{(t)} B(\boldsymbol{B}_i) + \sum_{i=1}^{N_G} \phi_{(j|\boldsymbol{G}_i)}^{(t)} G(\boldsymbol{G}_i)}, \tag{11}$$

where $B(\boldsymbol{B}_i)$ denotes the number of pixels in the color histogram bin with histogram bin vector $\boldsymbol{B}_i$, and $G(\boldsymbol{G}_i)$ denotes the number of pixels in the color histogram bin with histogram bin vector $\boldsymbol{G}_i$. Note that the range of hue value of the new centroid should be normalized in the range of $[0, 360^\circ/h_Q)$.

Step 4: Check if the clustering process is converged according to the total distortion measurement, which is the summation of distances between each histogram bin and its nearest cluster centroid:

$$\begin{aligned} \Delta^{(t)} = & \sum_{i=1}^{N_B} \sum_{j=1}^{K} \phi_{(j|\boldsymbol{B}_i)}^{(t)} D^2(\boldsymbol{B}_i, \boldsymbol{C}_j^{(t)}) B(\boldsymbol{B}_i) \\ & + \sum_{i=1}^{N_G} \sum_{j=1}^{K} \phi_{(j|\boldsymbol{G}_i)}^{(t)} D^2(\boldsymbol{G}_i, \boldsymbol{C}_j^{(t)}) G(\boldsymbol{G}_i) \end{aligned}. \tag{12}$$

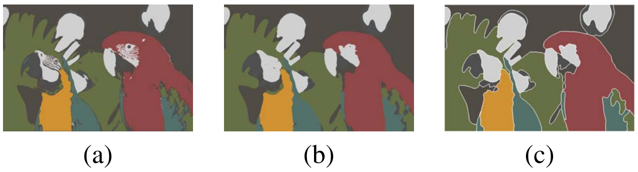

(a) (b) (c)

Fig. 3. Image *kodim23*: (a) labeled image, (b) filtered image, and (c) segmentation result.

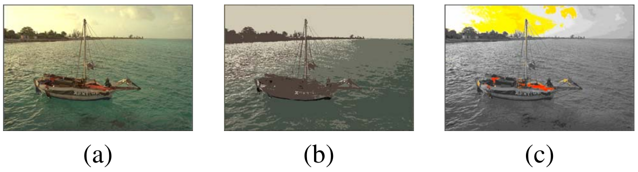

(a) (b) (c)

Fig. 4. Image *kodim06*: (a) the original image, (b) the clustering result of the proposed method , and (c) the clustering result in [5], where color and gray pixels are separated deterministically and marked.

When the difference of total distortion measurement $|\Delta^{(t+1)} - \Delta^{(t)}|$ is smaller than a predefined threshold or when the maximum iteration number is reached, the iteration is terminated. Otherwise, return to **Step 2** and t is incremented.

Step 5: Image pixels are labeled with the index of nearest centroid of their corresponding histogram bins. A labeled image $l(x, y)$ is obtained in this step, and K-Means clustering is finished. An example is shown in Fig. 3(a).

D. Post-Processing of Spatial Regions

To eliminate noise and unnecessary details of labeled images, an efficient statistical filter is introduced. A local histogram for a pixel on the position (x, y) in the labeled image is defined as

$$H(z|x, y) = \sum_{\substack{l(x',y') \in \mathcal{W}(x,y) \\ l(x',y') = z}} 1, \tag{13}$$

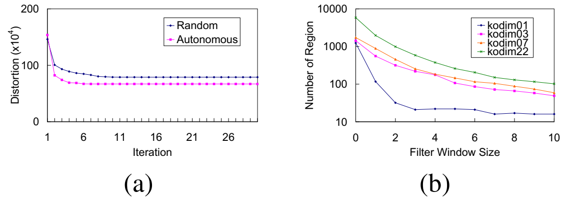
where $\mathcal{W}(x, y)$ is an $N \times N$ window centered in the spatial coordinate (x, y). Then the processed labeled image $\hat{l}(x, y)$ is obtained by using the following equation:

$$\hat{l}(x, y) = \arg\max_{z} H(z|x, y). \tag{14}$$

The purpose of this filter is to replace the pixel in the labeled image with the label with maximum number in a window. Afterwards, the spatial regions (8-connected) whose area is smaller than Th_A are merged to the neighboring region with the maximum area to avoid over-segmentation. Th_A is set to be 0.05% of the total number of pixels. Two examples in this stage are shown in Fig. 3(b)(c).

III. Experimental Results

The experiments, which contain four parts, are performed in Pentium-D 2.66GHz computer with 2GB memory. The first part is algorithm comparison. For 25 images of size 768×512, the average execution time of the proposed method is 0.29 second, and the method in [5] requires 1.20 second. Also, the



(a) (b)

Fig. 5. (a) Total distortion v.s. iteration of K-Means with image *kodim03*. (b) Region number v.s. filter window size with four images.

clustering result of these two methods are shown in Fig. 4, where the method in [5] separates color and gray pixels in HSV color space deterministically, and the proposed method clusters color and gray pixels together. It can be seen that the proposed method obtains better results than the method in [5] in terms of speed and robustness.

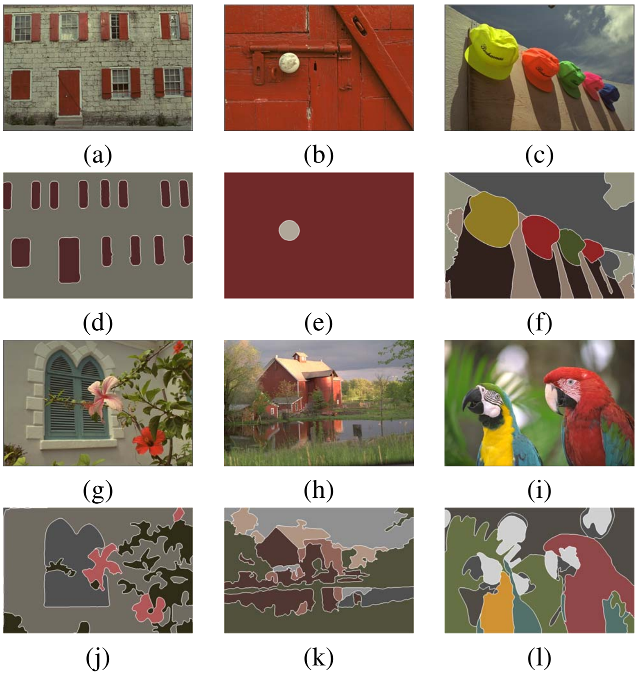
The second part is the functionality testing of the proposed K-Means clustering. To ensure K-Means operates properly in HSV color space with gray histogram bins and color histogram bins, the total distortion measurement in (12) is plotted with respect to iteration times. It is found that for all testing images, the distortion declines as the iteration time increases. Also, the Maximin initialization can obtain distortions similar to the random initialization: A plot of image *kodim03* is shown in Fig. 5(a), where the distortions of random initialization are average values in ten times.

The third part is the performance analysis for post-processing. To test if the proposed filter can effectively reduce the region number of the labeled image, the relation of region (8-connected) number and filter window size with four testing images are plotted in Fig. 5(b). It can be seen that the region number declines dramatically as the window size N increases.

The final part focuses on segmentation results and human perceptions. Six testing images are shown in Fig. 6, where the cluster number K, automatically selected for each image, is listed below. Additionally, the window size of filter for post-processing is set to $N = 7$. In Fig. 6(a)(d), the doors and the wall are segmented well; in Fig. 6(b)(e), the door knob and the background are distinguished; in Fig. 6(c)(f), five hats with different color are extracted; in Fig. 6(g)(j), the gray wall can be separated from the gray window; in Fig. 6(h)(k), the boundary of the sky and the house is clear; in Fig. 6(j)(l), colorful segments of parrots are partitioned. It can be seen that the segmentation results are similar to high-level perceptions of human visions, and the number of cluster is suitable for color representation in these testing images.

IV. Conclusions and Future Work

The characteristics of the proposed image segmentation algorithm include a new quantization method to generate gray and color histograms in HSV color space for K-Means clustering, where the cluster number is automatically estimated based on Maximin initialization. Moreover, the proposed method is both fast and efficient to extract regions with different colors in images, and the segmentation results are close to human perceptions. For future developments, spatial and texture information will be integrated into pixel features for clustering, and the segmented images will be used for region-based image retrieval.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 6. Original images: (a) *kodim01*, (b) *kodim02*, (c) *kodim03*, (g) *kodim07*, (h) *kodim22*, (i) *kodim23*. Segmentation results: (d) *kodim01* ($K = 2$), (e) *kodim02* ($K = 2$), (f) *kodim03* ($K = 7$), (j) *kodim07* ($K = 4$), (k) *kodim22* ($K = 6$), (l) *kodim23* ($K = 7$).

References

[1] L. Lucchese and S. Mitray, "Color image segmentation: A state-of-the-art survey," in *Proceedings of the Indian National Science Academy*, Mar. 2001, pp. 207–221, (Invited Paper).

[2] M. Luo, Y.-F. Ma, and H.-J. Zhang, "A spatial constrained K-Means approach to image segmentation," in *Proceedings of the Joint Conference of International Conference on Information, Communications and Signal Processing, and Pacific Rim Conference on Multimedia*, vol. 2, Dec. 2003, pp. 738–742.

[3] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Still image segmentation tools for content-based multimedia applications," *International Journal of pattern recognition and artificial intelligence*, vol. 18, no. 4, pp. 701–725, Jun. 2004.

[4] T. Elomaa and H. Koivistoinen, "On autonomous K-Means clustering," in *Proceedings of 15th International Symposium on Methodologies for Intelligent Systems*, May 2005, pp. 228–236.

[5] S. Sural, G. Qian, and S. Pramanik, "Segmentation and histogram generation using the HSV color space for image retrieval," in *Proceedings of IEEE International Conference on Image Processing*, Sep. 2002, pp. 589–592.

[6] J. R. Smith, "Color for image retrieval," in *Image Databases*. John Wiley & Sons, Inc., 2002, ch. 11, pp. 285–311.

[7] Z.-K. Huang and D.-H. Liu, "Segmentation of color image using EM algorithm in HSV color space," in *Proceedings of IEEE International Conference on Information Acquisition*, Jul. 2007, pp. 316–319.

[8] W. Chen, Y. Q. Shi, and G. Xuan, "Identifying computer grahics using HSV color model and statistical moments of characteristic functions," in *Proceedings of IEEE International Conference on Multimedia and Expo*, Jul. 2007, pp. 1123–1126.

[9] J. Smith and S.-F. Chang, "Single color extraction and image query," in *Proceedings of IEEE International Conference on Image Processing*, Oct. 1995, pp. 528–531.