Proceedings of the 33rd Chinese Control Conference
July 28-30, 2014, Nanjing, China

Color moving object detection method based on automatic color clustering

LI Jing, WANG Junzheng, Mao Jiali

Key Laboratory of Complex System Intelligent Control and Decision, School of Automation, Beijing Institute of Technology, Beijing 100081
E-mail: bitljing@163.com

Abstract: Moving object detection is usually used in gray image, and the color image should be converted to gray images to achieve detection. However, the color object can't be described by simple gray information completely. In order to achieve automatic color moving object detection, the method based on automatic color clustering is especially proposed. Because the difference between adjacent pixels in an image is very small, the initial cluster center can be moved in the maximum distance. The proposed method can reduce the number of iterations and quicken convergence speed, meanwhile determine the number of color cluster and the location of the color object automatically. The proposed algorithm is validated in the actual system, experimental results show that the proposed method can automatically achieve color clustering for color image, while detect color objects.

Key Words: Color clustering; Color object; Moving object detection

1 Introduction

Image segmentation is a backbone step and a classical problem for most of the machine vision application. The traditional gray-scale segmentation can only use the intensity information. While color images segmentation contains more discriminate levels reaching up to approximately millions. Meanwhile, the ability of input hardware that can acquire full color and high resolution images in real time has further supported the requirement of effective algorithms in this area. As a result, computer vision has shifted more towards color input rather than gray-scale, and the problem of segmenting color images has attracted great attentions.

The most common methods for color image segmentation are based on histogram threshold and color clustering. The later method has a wider application than the former one because of its independence in color. The color clustering method includes two branches: supervised and unsupervised. The supervised algorithms are easy, but they might lose part of color information, while the unsupervised ones have a low error segmenting rate and a potential improvement in mission success rate[1,2].

In the clustering algorithm the validity analysis and algorithm complexity judgment of cluster is carried on only after determining the number of clusters. Means clustering is a classic unsupervised classification method which does not require training samples, and just iteratively perform classification algorithm. The most common means of clustering methods are K-mean, fuzzy C-Mean, these two methods are intuitive and easy to implement, but are unable to determine the number of clusters[3]. Lim etc. combined the histogram threshold and FCM for color image segmentation, trying to determine the number of clusters determined automatically and the center position, but there is no way to get rid of defects caused by the histogram[4]. Lin Kai-yan etc. adopted the hierarchical subtraction clustering to cluster the image in a number of color similar subset[5]. And chen etc. proposed a new fuzzy clustering algorithm to achieve a self-determined number of clusters, but which also increases the computational complexity[6].

According to H channel distribution characteristics of HSI space, the improved K-Mean algorithm is proposed, which can dynamically increase the number of clusters, and divide the number of clusters automatically. Considering the property of the adjacent pixels with smaller color difference, the initial cluster centers in K-Mean is scattered throughout the image color space, and moved with distance between all samples pixels and the most primitive cluster center, which can reduce the number of iterations and accelerate algorithm speed. The proposed method is also applied in actual color image clustering, object detection, and achieved good results.

2 K-Means algorithm

K-Means clustering algorithm is a simple and popular unsupervised learning algorithm in clustering. K-Means clustering algorithm has been shown to be very useful for a corpus of practical applications.

K-Means algorithm follows a simple and easy way to classify a given data set $\{x_1, x_2 \dots x_N\}$ through a certain number of clusters (assume k clusters) fixed a priori. The steps can be described as follows[7]:

Step 1. Define the cluster number k and initialize k centriods $\{c_j\}_{j=1}^{k}$ by an initializing method.

Step 2. Assign each data point to the nearest centriod.

Step 3. Update k new centriods as the mean of the data points that have been asigned as closest to them.

The above Step 2 and Step 3 are repeatedly implemented until a certain criterion is satisfied. Commonly, we may notice that the k centroids change their location step by step until no more changes are done.

The aim of this algorithm is to minimize an objective function. We often take the following squared error function as the objective function. It defined as

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\| x_i^{(j)} - c_j \right\|^2 , \qquad (1)$$

*This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61103157.

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centriod c_j , it is an indicator of the distance of the *n* data points from their respective cluster centriods.

3 The proposed adaptive clustering algorithm

3.1 Defining the distance in the Hue component

The proposed K-Means clustering algorithm is implemented in Hue component, so we should first redefine the distance of two points. The distribution of H in HSI space is shown in Fig 1.

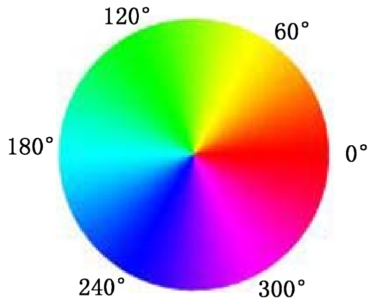


Fig. 1: The distribution of H in HSI

Considering the cyclic property of hue values, the distance between two Hue values H_1 and H_2 is:

$$d(H_1, H_2) = \begin{cases} |H_1 - H_2|, for |H_1 - H_2| \le \pi \\ 2\pi - |H_1 - H_2|, for |H_1 - H_2| > \pi \end{cases} \quad (2)$$

3.2 Initializing the centroids for the proposed K-Means

Initializing the centriods is the start and a key step of K-Means clustering algorithm. This algorithm starts by randomly choosing a point x as the first centriod. The point which is furthest from x is chosen as the second centriod. The distance of all points to the nearest of these two centriods is calculated. The point which is the furthest from its nearest centriod is chosen as the third centriod. By repeating choosing the furthest point from its nearest centriod, we can obtain K initial centriods. Katsavounidis et al. proved that this algorithm can reduce the computational complexity and achieves a better local minimum. In this paper, our algorithm only needs to initial 2 centriods, the min value of k.

3.3 Adaptive value of k in clustering

The problem of K-Means algorithm we want to solve in this paper is that we must have an a priori fixed clustering number k, apparently it is inconvenient to analyze the distribution of the colors to obtain prior information. In this paper, we present a technique of segmenting color image into k adaptive clusters with no prior information.

In Fig. 1, the attribute H is represented by the coordinate θ in the range of [0°, 360°]. As θ increases counterclockwise from 0° to 360°,the human visual perception of hue changes gradually from red, to yellow, green, cyan, blue, magenta, and back to red. The color function in the three primary colors behaves high similarity, while it behaves low similarity in the other three colors. So we can divide the hue circle into six zones, three main zones and three secondary zones. The discriminant function of main and secondary zones is as follows:

$$a = H / 30, b = H / 60$$

$$c = (a + b)\%2 = \begin{cases} 0, & main \quad zone \\ 1, & secondary \quad zone \end{cases} \quad (3)$$

where H is the hue value of the pixel, and a & b are the aliquot part of the division. This discriminate function is obtained from the rule in the Table 1. Then we set the acceptable cluster size for each hue zone. Here the cluster size means the distance in cluster between the farthest point and the centriod. The size in main hue zone is 30 and in secondary hue zone is 15.

Table 1: Main and secondary zone table

H	30---90		90---150		150---210		210---270		270---330		330---30	
	30-60	60-90	90-120	120-150	150-180	180-210	210-240	240-270	270-300	300-330	330-360	0-30
a	1	2	3	4	5	6	7	8	9	10	11	0
b	0	1	1	2	2	3	3	4	4	5	5	0
a+b	Odd		Even		Odd		Even		Odd		Even	
zone	Secondary		Main		Secondary		Main		Secondary		main	

Where H is the hue value of the pixel, and a & b are the aliquot part of the division

By judging whether the cluster centroid is in the main zone or in the secondary zone, and whether the farthest distance in cluster is larger than the fixed threshold 60 or 30; we can modify the value of k in K-Means iteration. If both of the judgments are true, then increase the number k, if not, remain the k, thus we can get an adaptive k by the above operation at last. By the above progress, the data set might be easily overestimated, so we have to do some correction. Firstly, we range the above clusters from low hue to high hue. In this paper, we define the distance between two adjacent clusters as d , the threshold between two adjacent clusters in the same main hue zone as d_{mm} , the threshold in the same secondary hue zone as d_{ss} and the threshold between the main hue zone and the secondary hue zone as d_{ms} .Then we correct the cluster number k as follows:

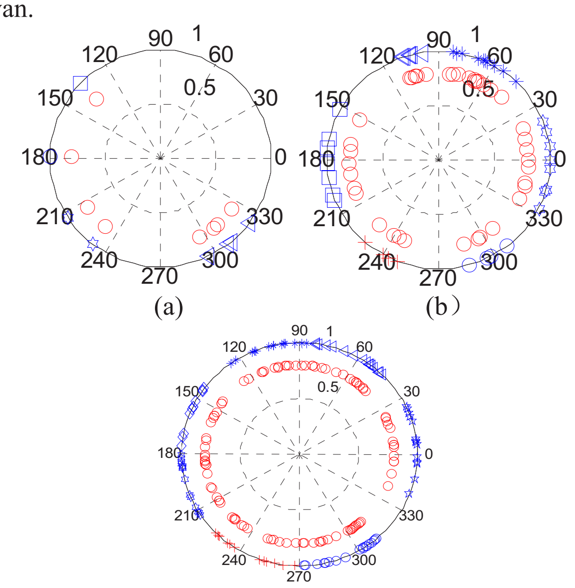
If the centriods are in the main hue zone and $d < d_{mm}$, or the centriods are in the secondary hue zone and $d < d_{ss}$, or the centriods are in the difference hue zone and $d < d_{ms}$, reduce the cluster number k in this iteration.

Thus we can obtain an adaptive k for K-Means algorithm in color segmentation.

4 Experimental results

4.1 Data experiment

The proposed algorithm was implemented in a matlab program and a C++ program with Open CV. From Fig. 1, it can be seen that the hue component in HSI color space is in the range of 0 to 360. So in first set of tests by matlab program, we used evenly distributed random data between [0,360] to simulate the hue component. The tests was conducted with three group of data mentioned above, with sample numbers of 8, 40, 100. All results were presented in polar coordinate shown in Fig. 3, attribute angle represented the value of hue, points in the unit circle represented the original samples, and the points in different shapes on the unit circle represented the samples after clustering, different shapes different clusters. In Fig. 2, 2(a) clustered 8 samples into 4 clusters: green, cyan, blue and magenta. While 2(b) clustered 40 samples into 6 clusters: red, yellow, green, cyan, blue and magenta, and 3(c) clustered 100 samples into 7 clusters: red, yellow, green, blue, magenta, and 2 clusters in cyan.



(a) (b）(c)

Fig. 2: Results of test by matlab with random evenly distributed data

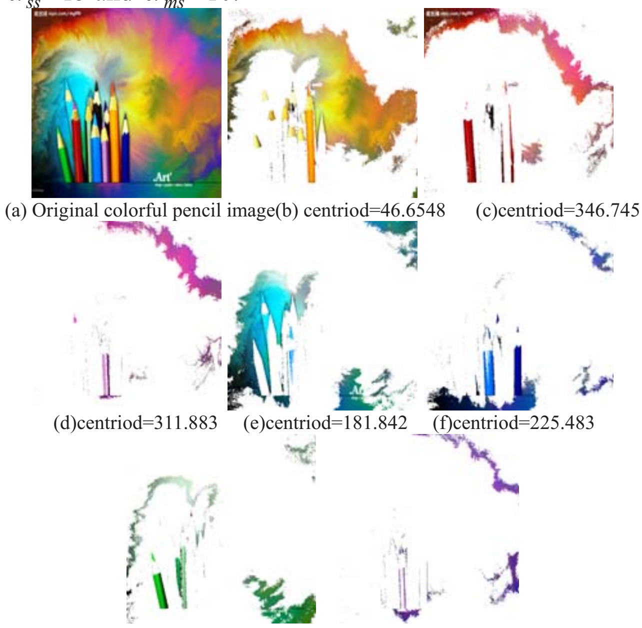
4.2 Image experiment

The second set of tests was conducted in C++ program with various images. Firstly, we used parameters of d_{mm} is 30, d_{ss} is 30 and d_{ms} is 30. In Fig. 3, the image of peppers was segmented into two clusters, with the first cluster centriod 357.177 and the second centriod 79.0142, representing red and green.

(a) Original (b)centrid=357.177 (c)centriod=79.0142

Fig. 3: Results of image peppers

In Fig. 4, the image of colorful pencils was segmented into seven clusters with the centroids labeled on each cluster image. Secondly, we used the parameters of d_{mm} =30, d_{ss} =15 and d_{ms} =10.



(a) Original colorful pencil image(b) centriod=46.6548 (c)centriod=346.745

(d)centriod=311.883 (e)centriod=181.842 (f)centriod=225.483

(g)centriod=128.818 (h) centriod =269.465

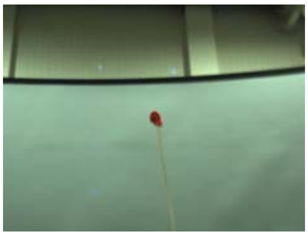
Fig. 4: Results of colorful pencils image

4.3 Visual servo tracking experiment

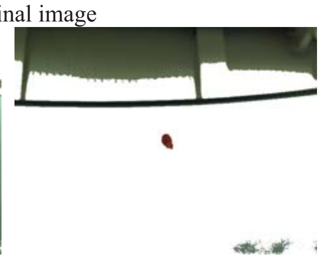
In order to verify the validity of the proposed method, CCD placed on a three-axis stabilized tracking platform is used to collect the object image, the object location is obtained by image processing platform and transferred to the controller of tracking platform. As shown in Fig. 5.

Fig. 5 Tracking platform and control system

Color clustering of object is shown in Fig. 6, considering the classification speed, the color number is divided into two categories, the color object is obviously be split out.



(a) Original image



(b) First result (c) Second result

Fig. 6 Color object clustering

Color object tracking result is shown in Fig 7. The image center is the axis center of CCD, and the tracked object is controlled in the center of the field of view. In the tracking process, the object is marked by blue point. And CCD moves to track the color object.

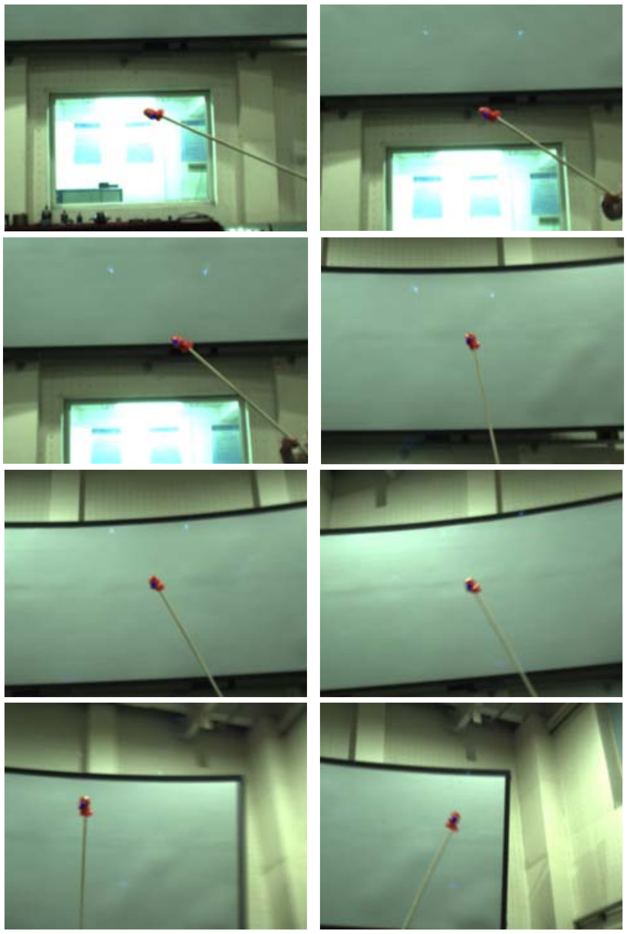


Fig. 7 Color object tracking results

In the experiment, the camera frame rate is 205fps, the real-time processing time of detection algorithm is less than 10ms/f, which can achieve real-time visual servo tracking control.

5 Conclusion

The improved K-Mean color image segmentation is proposed without increasing the computational complexity. According to the maximum distance of the center and classes, the initial centers is moved, which reduces the number of iterations and speed up the convergence of iteration. At the same time, according to the characteristics of HSI color space, the space is divided into main and secondary zone, and then the number of clustering is automatically adjusted according to the maximum distance in class. The proposed algorithm is validated in practical tracking system, the experimental results show that the proposed method can achieve color clustering, while detect color object automatically.

References

[1] Guo Dong, Ming Xie, Color clustering and learning for image segmentation based on neural networks, *IEEE transactions on neural networks*. 16(4): 925-936, 2005.

[2] 林开颜, 吴军辉, 徐立鸿, 彩色图像分割方法综述, *中国图像图形学报*, 2005,10(1):1-10.

[3] Xu, R., Wunsch, D. Survey of clustering algorithms, *IEEE Transactions on Neural Networks*, 2005, 16(3): 645–678

[4] Lim Y W, Lee S U. The color image segmentation algorithm based on the threshold and the fuzzy c-mean techniques. *Pattern Recognition*, 1990, 23(9):935-952.

[5] Lin K Y, Xu L H, Wu J H A fast fuzzy C-Mean clustering for color image segmentation. *Journal of Image and Graphics*, 2004, 9(2):159-163.

[6] Chen T Q, Lu Y. Color image segmentation-An innovative approach [J]. *Pattern Recognition*, 2002, 35(2):395-405.

[7] Francesco Camastra, Alessandro Verri, A Novel Kernel Method for Clustering, *IEEE Transactions on pattern analysis and machine intelligence*, 27(5): 801-805, 2005