Segmentation by Fusion of Self-Adaptive SFCM Cluster in Multi-Color Space Components

Kun Chen kun_1949301@126.com
Student, Department of Information Mechanical and Electrical Engineering
Shang Hai Normal University
Shang Hai, 200234, China

Yan Ma ma-yan@shnu.edu.cn
Professor, Department of Information Mechanical and Electrical Engineering
Shang Hai Normal University
Shang Hai, 200234, China

Jun Liu liujunout@126.com
Student, Department of Information Mechanical and Electrical Engineering
Shang Hai Normal University
Shang Hai, 200234, China

Abstract

This paper proposes a new, simple, and efficient segmentation approach that could find diverse applications in pattern recognition as well as in computer vision, particularly in color image segmentation. First, we choose the best segmentation components among six different color spaces. Then, Histogram and SFCM techniques are applied for initialization of segmentation. Finally, we fuse the segmentation results and merge similar regions. Extensive experiments have been taken on Berkeley image database by using the proposed algorithm. The results show that, compared with some classical segmentation algorithms, such as Mean-Shift, FCR and CTM, etc, our method could yield reasonably good or better image partitioning, which illustrates practical value of the method.

Keywords: Color image Segmentation, Histogram, SFCM, Fusion, Multi-color Space Components

1. INTRODUCTION

Image segmentation is a popular technique for image processing. The purpose of image segmentation is to divide an image into regions that can be considered homogeneous with respect to a given criterion such as gray level, color or texture, etc [1-2]. Image segmentation is one of the most widely studied problems in image analysis and computer vision, and it is a significant step towards image understanding. Since color images carry much more color information which is important to human perception, with the rapid growing of computer processing ability, recently color image segmentation has became a hot research topic. It is widely applied in many areas such as: image compression, internet video transmission, medical image diagnosis and target tracking, etc. We should solve two problems for image segmentation: (1) choose the right color space; (2) select the appropriate segmentation strategy. Since the selection of color space depends on specific image and segmentation strategy, nowadays there is no color space can be suited for all color images [3].

Many methods have been proposed and studied in the last decades to solve the color image segmentation problem. Some researchers prefer to use more complicated feature selection procedures or more elaborate clustering techniques and then improve the final segmentation result by complex optimization method. Some segmentation techniques integrated with specific theory, method and means has emerged, such as segmentation based on lossy data compression [4-5], wavelet-domain hidden markov models [6], graph-based [7-8], Mean-Shift [9] and etc. Some researchers also use information fusion strategies to get better performance. They prefer to fuse the results associated with the simple method applied on different color spaces rather than to consider complex segmentation theory or model. Eg. Mignotte [10] proposed a method called FCR by fusion of multi-color spaces based on local histogram and K-means clusters. First, a simple clustering model based on local histogram has been proposed in [10], then, the model has been applied into RGB、HSV、YIQ、XYZ、LAB、LUV color spaces to achieve six segmentation results. Finally, six segmentation results have been fused to achieve the best segmentation results. It is a simple and effective method which makes use of the advantage of many different color spaces. But it also has some problems as follows: (1) The runtime of local histogram clustering modeling is too long. (2) The number of clusters is fixed, therefore, it can not meet the self-adaptive requirement for different images.

Learning from Mignotte's idea, we propose a novel, simple, efficient and self-adaptive method by fusion of multi-color space components. First, we choose six different color components elaborately through various experiments: Gray component, V(HSV) component, I(YIQ) component, Cr(YCbCr) component, B(LAB) component and U(LUV) component. Then we propose a peak-finding algorithm to determine cluster number of each component and initialize cluster centroid for SFCM clustering. Then, A clustering method is proposed to fuse six different segmentation results, where the cluster number is the mean of the above six cluster numbers. Finally, we propose region merging method to merge the previous segmentation results. The proposed method is tested on Berkeley natural image database. Extensive experiments show that, the method is simple, efficient, and robust to noise. Compared with FCR, our method can get better result and faster. Compared to the state-of-the-art segmentation methods recently proposed in the literature, our method performs competitively in terms of visual evaluations and quantitative performance measures.

2. INITIAL SEGMENTATION

FCM algorithm has been used as one of the most popular cluster techniques for image segmentation in computer vision and pattern recognition. It is developed by Dunn [11] in 1973 and improved by Bezdek[12] in 1981. Although FCM has been widely used in image segmentation domains, it still exists the following problems: (1) In terms of performance the algorithm depends on the initial cluster centroids; (2) The cluster number must be fixed before clustering; (3) High computational complexity; (4) No consideration of spatial information. Taking into account above problems, we use histogram technique to find initial cluster centroids and determine cluster number. We only cluster 1-D component of each color space in terms of computational complexity, therefore the method is simple and rapid. We use SFCM [13] to consider spatial information and achieve initial segmentation results. After initial segmentation, we achieve six different initial segmentation results from different color space components (Gray component, V component, I component, Cr component, B component, U component).

2.1 Peak Finding

How to determine initial cluster centroids has always been a problem of clustering. Good initial cluster centroids not only can yield better cluster results but also can make cluster faster. Selecting initial cluster centroids randomly is likely to lead the optimization of the algorithm's objective function to local extreme, therefore the accuracy of the cluster results will be affected. In this paper we utilize histogram technique to find cluster centroid. Here we take gray component as an example to propose the peak finding algorithm. In this way we also can obtain the peaks of other components. The procedure is as follows:

1) Quantize gray component into 0-255 intensity levels, count the frequency, and create the histogram. Let g(i) be the gray component histogram, x_i be the number of pixels associated with

ith intensity level in g(i).The histogram of gray component can be represented by the following equation:

$$g(i) = x_i, \quad 0 \le i \le 255 \tag{1}$$

2) Smooth histogram. Use 1D Gaussian filters with size of 1×5 for $g(i)$ to smooth twice, and the result of smoothing depends on Gaussian standard covariance σ_g. The histogram is more smoother with bigger σ_g. We have a new histogram $T_g(i)$ after smoothing.

3) Search for initial peaks. We search turning points on which gradient value varies from positive to negative. We take these turning points as initial peaks and get initial set of peaks P_1.

4) Remove small peaks. If the value of peak in set P_1 is less than threshold T_1, it is removed from P_1. So we have new set of peaks P_2.

5) Remove adjacent peaks and generate final peaks. If two peaks in P_2 are close enough, we think the gray values of the regions represented by the two peaks are similar. Therefore, we remove the smaller one while the distance between two peaks is less than threshold T_2. We get final set of peaks P_3.

2.2 Spatial FCM Clustering

The classical FCM algorithm is to assign pixels to each cluster by using fuzzy memberships. Let $X = (x_1, x_2, \cdots x_n)$ denotes an image with n pixels to be partitioned into c clusters, where x_i represents multispectral (features) data. The result of classification can be represented by a fuzzy membership degree matrix $U = \{\mu_{ik}\}$, where μ_{ik} represents the membership degree of kth pixel to ith cluster centroid. it is subject to the following constraints:

$$\mu_{ik} \in [0,1], \forall i,k \ ; \ \ 0 < \sum_k \mu_{ik} < n, \forall i \ ; \ \sum_i \mu_{ik} = 1, \forall k \tag{2}$$

FCM algorithm is an iterative optimization that minimizes the cost function defined as follows:

$$J = \sum_{k=1}^{n} \sum_{i=1}^{c} \mu_{ik}^{l} \left\| x_k - v_i \right\|^2, \tag{3}$$

Where $U = \{\mu_{ik}\}$ is the membership degree matrix according to Eq.(2), $V = \{v_1, v_2, \cdots v_c\}$ is the set of cluster centroids, $\|x_k - v_i\|$ represents the distance of pixel x_k to cluster centroid v_i, and we use Euclidean distance in initial segmentation. The parameter l controls the fuzziness of the resulting partition, and $l = 2$ is used in this study.

The membership functions μ_{ik} and the centroids V_i are updated iteratively as follows:

$$\mu_{ik} = \frac{\left\| x_k - v_i \right\|^{-2/(l-1)}}{\sum_{j=1}^{c} \left\| x_k - v_j \right\|^{-2/(l-1)}} \tag{4}$$

$$v_i = \frac{\sum_{k=1}^{n} \mu_{ik}{}^{l} x_k}{\sum_{k=1}^{n} \mu_{ik}{}^{l}} \tag{5}$$

The standard FCM algorithms is optimized when pixels close to their centroids are assigned high membership values, while those that are far away are assigned low values.

One of the problems of classical FCM algorithm in image segmentation is the lack of spatial information. Since image noise and artifacts often impair the performance of FCM segmentation, it would be attractive to incorporate spatial information into FCM. Chuang et al. [13] proposed a spatial FCM algorithm in which spatial information can be incorporated into fuzzy membership functions directly using

$$\mu'_{ik} = \frac{\mu_{ik}^p h_{ik}^q}{\sum_{j=1}^{c} \mu_{jk}^p h_{jk}^q} \tag{6}$$

Where p and q are two parameters controlling the respective contribution. The variable h_{ik} includes spatial information by

$$h_{ik} = \sum_{j \in N_k} \mu_{ij} \tag{7}$$

Where N_k denotes a local window centered around the image pixel k. The weighted μ_{ik} and the centroid v_i are updated as usual according to Eq. (4) and (5).

3. FUSION OF INITIAL SEGMENTATION RESULTS

We get six different initial segmentation results from six different color space components by using the method proposed in section 2. The cluster number of them is different, we record them as $K_i, 1 \le i \le 6$ (for example K_1 represent the cluster number of gray component, K_2 represent V component, etc). We use SFCM algorithm again to fuse above six results which with different cluster number and get a new result I_{fusion} after fusion.

3.1 Extract Feature Vector

For each initial segmentation result with $K_i (1 \le i \le 6)$ cluster number, considering the squared fixed-size ($N_W \times N_W$) neighborhood centered around the pixel. Let W_x represent the neighborhood of pixel location x.We calculate the normalized local histogram of the class labels for each pixel within W_x:

$$h(W_x) = (\frac{n_1}{N_w^2}, \frac{n_2}{N_w^2} \cdots \frac{n_{K_i - 1}}{N_w^2}, \frac{n_{K_i}}{N_w^2}) \tag{8}$$

Where $h(W_x)$ represent the feature vector of pixel location x in one of the six segmentation results, n_j denotes the number of pixels whose class labels are j within W_x. We do the same process toward six different segmentation result described above. After that, we get six feature vector location in the same place for each pixel. Then combine them in series and normalized. Finally, We get the fused local histogram of the class labels $h^*(W_x)$ with dimension $M = \sum_{i=1}^{6} K_i$, which is used as feature vector for input in the final clustering.

3.2 Fusion of Initial Segmentation by SFCM

We adopt SFCM algorithm (described in Section 2.2) again to partition $h^*(W_x)$ into N classes.

$$N = ceil(\sum_{i=1}^{6} K_i / 6) \tag{9}$$

Where ceil(A) represents round the elements of A to the nearest integers. We get segmentation result I_{fusion} by fusion, in which the distance between two feature vectors from local histogram of the class labels is calculated by Bhattacharya distance:

$$D_B[h_1^*, h_2^*] = \left(1 - \sum_{i=1}^{M} \sqrt{h_1^* \cdot h_2^*}\right)^{1/2} \quad (10)$$

Where h_1^*, h_2^* denote two normalized feature vectors from local histogram of the class labels, M denotes the dimension of feature vector.

4. REGION MERGING

Segmentations with clustering are often featured with numerous discrete small regions. The spatial connectivity between pixels in the same cluster could hardly be guaranteed. These minor regions on one hand preserves the image detail but on the other hand largely affects the segmentation quality. To generate reasonable segmentations, a simple and effective region merging strategy is necessary for this issue. In this paper, the region merging method is presented in LUV color space. The steps are as follows:

1、 Relabel regions after segmentation by 8-neighbors, which yields that not adjacent and color-homogeneous regions are marked with different labels.
2、 Search adjacent regions after relabeling.
3、 Calculate the mean value of L, U, V components for relabeled regions.
4、 Merge small regions. If the size of region is smaller than threshold T_3, it will be merged into its bigger adjacent region with the smallest Euclidean distance in LUV color space, and whose size is greater than T_3.
5、 Merge big regions. If the size of region is smaller than threshold $T_4 (T_3 << T_4)$, we calculate Euclidean distance in LUV color space with its adjacent regions whose size is greater than T_4. Search the smallest distance dc. If $dc < T_5$, the region will be merged into its adjacent bigger region with the smallest distance, and vice versa.

5. EXPERIMENT RESULTS AND ANALYSIS

The proposed algorithm is demonstrated on the computer Inter Core2 Duo CPU T6570 2.10GHz. We use Matlab R2011a to test the segmentation results on natural images in the Berkeley segmentation database[14], which also contains benchmark segmentation results obtained from human subjects. We have done numerous experiments which show that the results are best when the involved parameters σ_g, T_1, T_2 chosen to 3, 0.001S (S denotes the size of image),15, the window size $N_w \times N_w$ chosen to 5×5 and T_3, T_4, T_5 chosen to 0.003S, 0.05S, 50. We will analyze our algorithm from the following aspects: the choice of different color components, whether the algorithm is robust to noise and compare the algorithm with some state-of-art methods qualitatively and quantitatively.

The quantitative comparison is based on the following performance measures, namely a probabilistic measure called PRI [15,16] (higher probability is better) and three metrics VoI [17], GCE [14], and BDE [18] (lower distance is better). The qualitative meaning of these performance measures are recalled as follows.

1) PRI (Probabilistic Rand Index) counts the fraction of pairs of pixels whose labellings are consistent between the computed segmentation and the ground truth, averaging across multiple ground truth segmentations to account for scale variation in human perception.

2) VoI (Variation of Information) defines the distance between two segmentations as the average conditional entropy of one segmentation given the other, and thus roughly measures the amount of randomness in one segmentation which cannot be explained by the other.

3) GCE (Global Consistency Error) measures the extent to which one segmentation can be viewed as a refinement of the other. Segmentations which are related in this manner are

considered to be consistent, since they could represent the same natural image segmented at different scales.

4) BDE (Boundary Displacement Error) measures the average displacement error of boundary pixels between two segmented images. Particularly, it defines the error of one boundary pixel as the distance between the pixel and the closest pixel in the other boundary image.

5.1 Choice of Different Color Components

Extensive experiments show that the selection of different color components has important influence on the segmentation result. In order to compare with FCR [10] algorithm, we choose six components to fuse. Which six different color components are the best? We use self-adaptive histogram and SFCM clustering techniques to quantitatively test the components of HSV, YIQ, YCbCr, LAB, LUV color spaces and gray component on randomly chosen images.

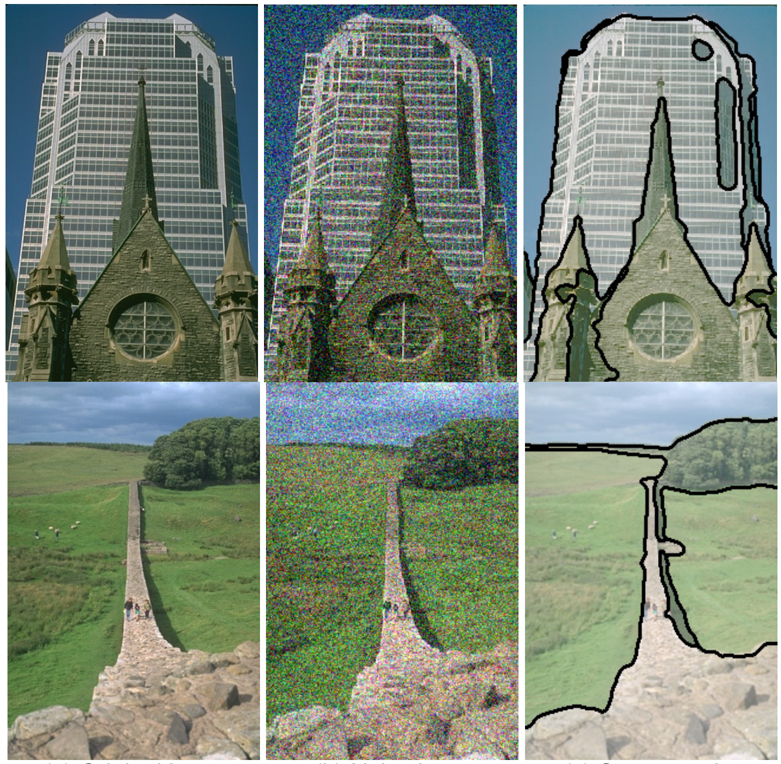
TABLE 1 shows the PRI, VoI, GCE and BDE performance of these 14 components on 100 randomly chosen images in the Berkeley segmentation database. Best performance of each measure is marked with bold. Second best is marked with underline. In PRI indice, V(HSV) component is best, Gray component is second best; In VoI indice, B component is best, I is second best; In BDE indice, Cr component is best, V component is second best. TABLE 1 also shows that some component is the best in one indice, but worse in other indices. Eg. A component has the best GCE indice, but PRI、BDE is worse. Therefore, we need to consider different performance measures of components together to select the best components. In the analysis, we choose Gray, V(HSV), I, Cr, B, U as six different components to fusion.

Component	PRI[15,16]	VoI[17]	GCE[14]	BDE[18]
Gray	<u>0.7045</u>	3.0394	0.3894	10.3919
H	0.6773	2.8031	0.3204	12.6409
S	0.6860	3.1189	0.3909	12.2657
V(HSV)	**0.7146**	3.0790	0.3953	<u>10.1278</u>
Y	0.6980	2.9322	0.3855	10.4577
I	0.6833	<u>2.6635</u>	0.3169	11.6272
Q	0.6511	2.9453	0.3485	11.9675
Cb	0.6782	2.9347	0.3612	10.2231
Cr	0.6776	2.9786	0.3565	**9.8885**
L	0.6958	2.8327	0.3282	10.7621
A	0.6190	2.6721	**0.2878**	14.6134
B	0.6767	**2.6045**	<u>0.2952</u>	11.5503
U	0.6568	2.7091	0.3013	10.7357
V(LUV)	0.6901	2.8074	0.3258	10.9033

TABLE 1: The Performance Measures of 14 Components.

5.2 Robust to Noise

According to [13],we know that SFCM algorithm is less sensitive to noise. Because the clustering of our algorithm is based on SFCM, we conclude that our algorithm may be robust to noise. In order to test it, we add the Gaussian noise (mean value is 0,variance is 0.03) to two randomly chosen images for segmentation. FIGURE 1 shows their original images, noise images and segmentation result by our proposed method. The result shows that even with Gaussian noise, we still can clearly get the correct part of the segmentation result, which proven the algorithm's robustness to noise. The reasons can be concluded as follows: First, our method use SFCM clustering which considering spatial information and can get better clustering results to noise image. Second, the proposed method adopts region merging technique after fusion of different segmentations, which can also effectively remove small noises.



(a) Original image (b) Noise image (c) Segmentation

FIGURE 1: Noise Image Segmentation

5.3 Comparison with State-of-the-art Methods

We test 300 images on Berkeley image database and compare our method with state-of-the-art methods such as: Mean-shift [9], NCuts [7], FH [8], CTM [4,5] and FCR [10].

FIGURE 2 shows the segmentation results of FCR, Mean-shift, CTM and our proposed method with 5 randomly chosen images. FIGURE 2(a) is original images. FIGURE 2(b) shows FCR segmentation results. FIGURE 2(c) is Mean-shift results. FIGURE 2(d) shows CTM results. FIGURE 2(e) is our proposed method. It is obvious that FCR and Mean-shift methods have over-segmentation problem in FIGURE 2. For certain images, these two methods can only yield small piece regions, and can't generate the right object, especially Mean-shift method. Our method can get better results which is close to human perception and has less over-segmentation problem.

TABLE 2 shows the mean value of performance measures over the 300 images of the Berkeley image database in different methods. Best performance of each measure is marked with bold. Second best is marked with underline. From TABLE 2 we can see that our method outperforms other methods for several different internal parameters, all the well-known segmentation algorithms presented in TABLE 2 in terms of PRI and BDE indices, second best in VoI indice and is obviously better than FCR in PRI, VoI and BDE indices.

TABLE 3 shows the average runtime of 100 randomly chosen images in the same platform. It is obvious that our method faster than FCR algorithm.

(a) Original Image

(b) FCR

(c) Mean-shift

(d) CTM

(e) Proposed Method

FIGURE 2: Comparison of FCR, Mean-shift, CTM and Our Method

Algorithms	**PRI[15,16]**	**VoI[17]**	**GCE[14]**	**BDE[18]**
Humans	0.8754	1.1040	0.0797	4.9940
FCR(K_1 =6, K_2 =6, k =0.13)[10]	<u>0.7842</u>	2.3925	0.2169	<u>9.2463</u>
CTM(η =0.1)[4,5]	0.7561	2.4640	**0.1767**	9.4211
CTM(η =0.2)[4,5]	0.7617	**2.0236**	<u>0.1877</u>	9.8962
Mean-shift[9]	0.7550	2.4770	0.2594	9.7001
NCuts[7]	0.7229	2.9329	0.2182	9.6038
FH[8]	0.7841	2.6647	0.1895	9.9497
Our Method	**0.7906**	<u>2.1395</u>	0.2218	**9.0652**

TABLE 2: Performance Measures Comparison to State-of-the-art Methods

Algorithms	**Our Method**	**FCR**
Runtime/s	138.732	317.859

TABLE 3: Runtime of Our Method and FCR.

6. CONCLUSION

This paper proposes a novel, simple, efficient and self-adaptive method by fusion of multi-color space components. Results show that the method provides good segmentation on a variety of color images. Histogram and SFCM cluster techniques are used in initial segmentation. The strategy not only can locate initial cluster centroids quickly but also can solve the problem of that clustering number is fixed. Then an effective fusion and region merging strategy is used to make segmentation result more close to human perception. The proposed method has been successfully applied on the Berkeley image database, and performs competitively among the recently reported state-of-the-art segmentation methods in terms of visual evaluations and quantitative performance measures. In our experiments, several limitations are found for the algorithm. One case is when the color of an image is too close, the segmentation result is bad. Another case is the algorithm only consider color information, do not consider other information such as texture. Future research work is on how to solve these problems and improve the results.

7. REFERENCES

[1] H. D. Cheng, X. H. Jiang, J. L. Wang. " Color Image Segmentation Based On Homogram Thresholding And Region Merging". Pattern Recognition, vol.36, pp. 373-393, 2002.

[2] H. D. Cheng, J. Li." Fuzzy Homogeneity And Scale-Space Approach To Color Image Segmentation". Pattern Recognition, vol.36, pp. 1545-1562, 2003.

[3] K. Y. Lin, J. H. Wu, L. H. Xu. " A Survey On Color Image Segmentation Techniques". Journal

of Image And Graphics, vol. 10, pp. 1-10, 2005.

[4] A. Y. Yang, J. Wright, S. Sastry, and Y. Ma. "Unsupervised Segmentation Of Natural Images Via Lossy Data Compression". Comput. Vis. Image Understand, vol. 110, pp. 212-225, 2008.

[5] Y. Ma, H Derksen, W. Hong, and J. Wright. "Segmentation Of Multivariate Mixed Data Via Lossy Coding And Compression". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 1546-1562, 2007.

[6] H. Choi, R. G. Baraniuk. "Multiscale Image Segmentation Using Wavelet-Domain Hidden Markov Models". IEEE Transactions on Image Processing, vol. 10, pp. 1309-1321, 2001.

[7] J. Shi and J. Malik. "Normalized Cuts and Image Segmentation". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 888-905, 2000.

[8] P. Felzenszwalb and D. Huttenlocher, "Efficient Graph-Based Image Segmentation". Int. J. Comput. Vis., vol. 59, pp. 167-181, 2004.

[9] D. Comanicu, P. Meer. "Mean shift: A Robust Approach Toward Feature Space Analysis". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 603-619, 2002.

[10] M. Mignotte. "Segmentation By Fusion Of Histogram-Based K-Means Clusters In Different Color Spaces". IEEE Transactions on image processing, vol. 17,pp. 780-787, 2008.

[11] J.C. Dunn. "A Fuzzy Relative Of The ISODATA Process And Its Use In Detecting Compact, Well Separated Cluster". Cybernetics, vol. 3, pp. 32–57, 1973.

[12] J. C. Bezdek, "Pattern Recognition With Fuzzy Objective Function Algorithms," in Plenum Press, New York, 1981.

[13] K. S. Chuang, H. L. Tzeng, S. Chen, J. Wu, T. J. Chen. "Fuzzy C-Means Clustering With Spatial Information For Image Segmentation". Computerized Medical Imaging and Graphics, vol. 30, pp. 9-15, 2006.

[14] D. Martin, C. Fowlkes, D. Tal, et al. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in Proc. of the 8th IEEE International Conference on Computer Vision, 2001, pp. 416-423.

[15] R. Unnikrishnan, C. Pantofaru, M. Hebert. "A measure for objective evaluation of image segmentation algorithms," in Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 34-41.

[16] R. Unnikrishnan, C. Pantofaru, M. Hebert. "Toward Objective Evaluation Of Image Segmentation Algorithms". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 929-944, 2007.

[17] M. Meila. "Comparing clusterings - An axiomatic view," in Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 577-584.

[18] J. Freixenet, X. Munoz, D. Raba, et al. "Yet another survey on image segmentation: region and boundary information integration," in Proceedings of European Conference on Computer Vision, 2002, pp. 408-422.