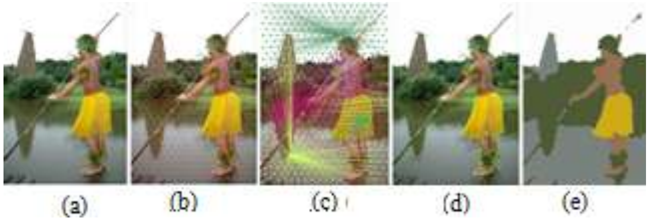

Figure 1: The procedure of SLICAP. (a) Original image; (b) SLIC superpixels; (c) Cluster superpixels using AP. Clustering center data points represent "exemplar", where different colors mean different clusters; (d) Boundary result of SLICAP; (e) Region result of SLICAP.

2. Methods

2.1 Analytical Framework of SLICAP Algorithm

The proposed SLICAP method is formulated by a combination of the SLIC superpixels algorithm and the AP clustering algorithm. Specifically, the superpixels are generated by SLIC firstly. Then a similarity matrix is constructed. And finally, superpixels are clustered by using the AP algorithm with the similarity matrix. Through a practical example of image segmentation, we show the analytical framework of the SLICAP method in Fig. 1.

SLIC has a primary parameter that controls the number of superpixels. An example of using the SLIC superpixel method to generate superpixels is shown in Fig. 1(b). Here we set the number of superpixels K as 600. The advantage of the SLIC method is that it provides a similarity matrix for AP clustering with low computational complexity. Besides, it well adheres to image boundaries [3].

The superpixels are then clustered by AP. The advantage of AP is that the number of exemplars does not have to be specified beforehand. Instead, an appropriate number of exemplars emerges from the message passing method [6] and only depends on the input exemplar preferences. It is more suitable for unsupervised segmentation than the K-means clustering. In Fig. 1(c), the superpixels are merged in five regions automatically, and each region has a center (so-called “exemplar” in AP). The boundary is yielded between different parts, as illustrated in Fig. 1(d). The resulting segmented regions are delineated in Fig. 1(e), where the color of each region is the mean of the corresponding superpixels. We see that the oar is not continuous. This is due to that the number of superpixels is not sufficiently large. On the other hand, increased number of superpixles needs higher complexity. We thus seek a tradeoff between the segmentation performance and its complexity.

2.2 Similarity Matrix Construction

In this subsection, we construct a similarity matrix in CIELAB color space that keeps consistent with the human visual perception. This CIELAB color space is based on the human visual system. It includes some colors that our physical world can not recreate. With the SLIC algorithm, we calculate the mean vector $[L\,a\,b]^T$ of all the superpixels, where L represents brightness, and a and b represent the change from red to green and from blue to yellow, respectively. For the purpose of comparison, three similarity matrices are designed as follows:

similarity A

$$s(i,k) = -[w_L(L_i - L_k)^2 + w_a(a_i - a_k)^2 + w_b(b_i - b_k)^2] \quad (1)$$

similarity B

$$s(i,k) = 1 - \exp[\frac{w_L(L_i - L_k)^2}{\varepsilon_L^2} + \frac{w_a(a_i - a_k)^2}{\varepsilon_a^2} + \frac{w_b(b_i - b_k)^2}{\varepsilon_b^2}] \quad (2)$$

similarity C

$$s(i,k) = -\exp\{-[\frac{w_L(L_i - L_k)^2}{\varepsilon_L^2} + \frac{w_a(a_i - a_k)^2}{\varepsilon_a^2} + \frac{w_b(b_i - b_k)^2}{\varepsilon_b^2}]^{-1}\} \quad (3)$$

$$s(i,i) = colorradius \times mean(s') \quad (4)$$

where i and k denote the indices of superpixels, and $s(i,k)$ denotes the element in the ith row and the kth column of a similarity matrix. The similarity $s(i,k)$ means the preference that data point i is chosen as an exemplar [6]. Besides, w_L, w_a, w_b are the weights of the three channels. They keep balance so as to be consistent with human perception. ε is the standard deviation of color distribution of superpixels. s' remains the off-diagonal elements of s. The quantity colorradius adjusts the number of clusters, and if its value is low, the number of targets would increase, which leads to more detailed segmentation results. The default value of colorradius is set as 20.

We see that the Euclidean distance is applied to similarity A. On the other hand, similarity B and similarity C include the standard deviations of the color distribution and take the exponential form. We will find in the experiment section that the frame based on the Euclidean distance (i.e., similarity A) delivers better performance for the AP clustering algorithm than the other two similarities. Also, the figures in Fig. 1 are produced by adopting similarity A.

The AP algorithm takes a collection of real-valued similarities between superpixels as an input. The similarity matrix of AP means that, in terms of Euclidean distance, two superpixels in a similarity matrix are more similar if their distance is more close to zero. Otherwise, they are more dissimilar if the value is more far from zero.

3. Experiments

All the experiments are conducted in the same running environment of computer, in which CPU is Intel(R) core(TM) 2, 2.13 GHz With 2G memory. Experiment platform and software are Linux 3.2.0-67-generic and MATLAB 7.14.0 (R2012a), respectively. The segmentation results of images are assessed by the boundary-based and region-based criteria.

We compare our algorithm with a classical methods, i.e., normalized cuts [8] (Ncuts), as well as SLIC-K-means (SLICKM). SLICKM replaces the AP clustering with K-means [9]. Likewise, we use the Euclidean distance and the CIELAB color space in SLICKM. In our experiment, the related parameters are set as follows. $colorradius$

1) SLICAP: We set the number of superpixels K as 600, the weight factor m between color and spatial differences as 20, w_L, w_a, w_b and $colorradius$ as 3, 10, 10, and 20, respectively. The superpixels are clustered by AP with default parameters.

2) Ncuts: The number of blocks is equal to 30 for the best performance.

3) SLICKM: K and m are same with SLICAP. The setting of the number of segmentation sections follows "Nseg.txt" in [10]. Specifically, if the segmentation number is set as N in "Nseg.txt", then the clustering number of K-means in SLICKM is limited in a interval near N and

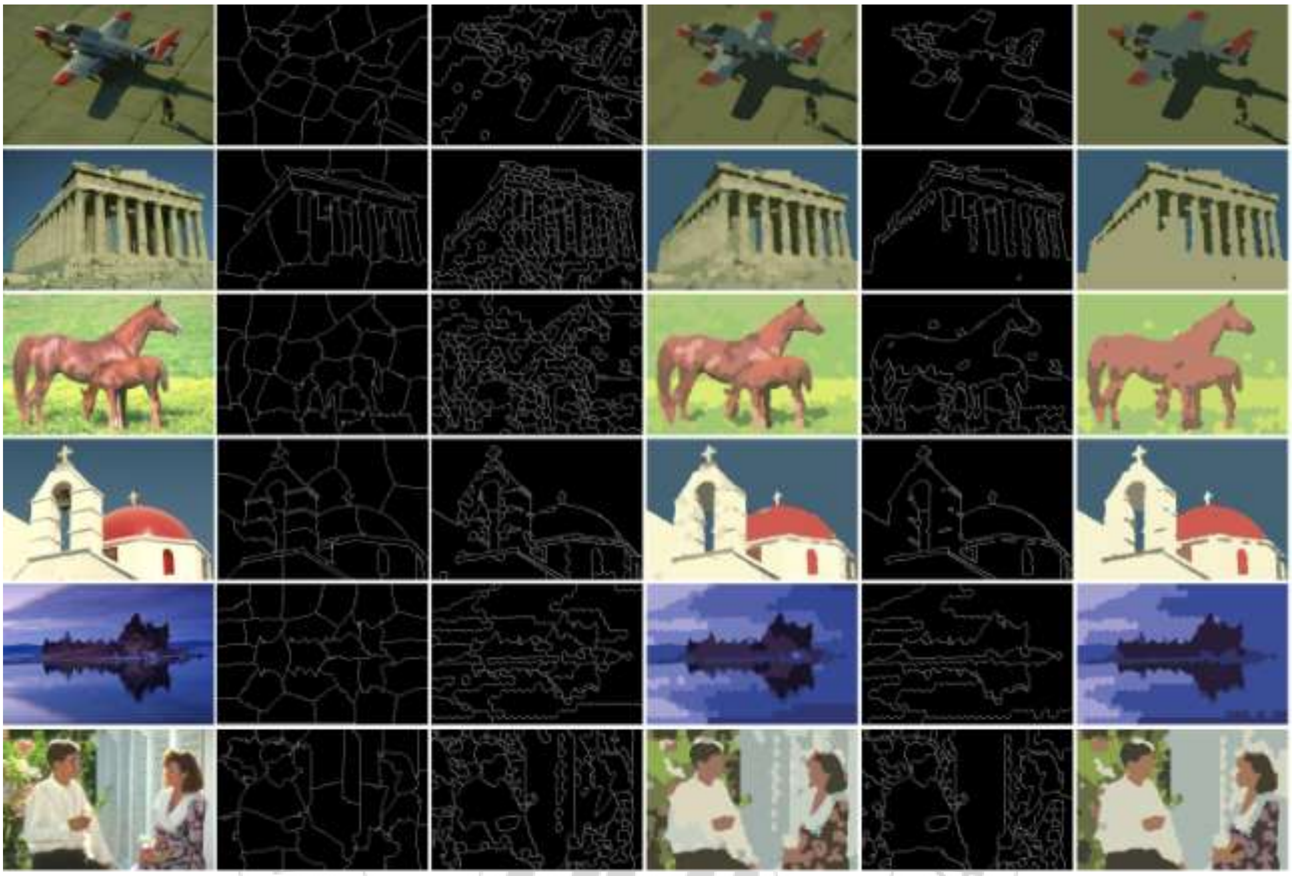

Figure 2: Segmentation examples on the Berkeley Segmentation Database. (a) Input image; (b) Ncuts; (c) Boundary result of SLICKM (average); (d) Mean color region result of SLICKM (average); (e) Boundary result of SLICAP (using similarity A); (f) Mean color region result of SLICAP (using similarity A)

chosen randomly within this interval. SLICKM is performed 200 times on the whole dataset and the best result is shown in the experiments.

3.1 Database

The image segmentation algorithms are evaluated on the Berkeley Segmentation Database (BSD) [11], which consists of 300 natural images. In order to obtain a fair assessment of the results from the superpixels-based image segmentation, 100 pictures of smaller number of targets from BSD are randomly selected to construct a sub-database. Besides, BSD offers a benchmark that produces a score for an algorithm, which will be discussed in the following section.

3.2 Boundary and Region Quantitative Evaluations

In order to compare the competing solutions, boundary and region quantitative evaluations are used. For boundary quantitative evaluation, the BSD [12] Precision-Recall framework is employed, where "Precision" and "Recall" are calculated and then used to get the F-measure. For region quantitative evaluation, the following measures are used: Probabilistic Rand Index (PRI) [13] [14], Variation of Information (VoI) [15] [16], and Boundary Displacement Error (BDE) [17] [18]. PRI, a variant of the Rand Index, counts the number of pixel pairs whose labels in the segmentation result are consistent with those in the ground truth. VoI was introduced for the purpose of clustering comparison. BDE measures the average displacement of the region boundaries between the segmentation result and the ground truth. In short, a segmentation result is better if it has a higher PRI, a lower VoI, and a lower BDE.

4. Results and Discussion

Some segmentation examples are shown in Fig. 2, where we adopt the optimal dataset scale (ODS) instead of the optimal scale per image (OIS). Comparing Fig. 2(b) and (c) with (e), we see that SLICAP well adheres to object boundaries and consists with human perception. It is observed from Fig. 2(d) and (f) that SLICAP produces a more appropriate number of targets automatically. The reason is that the appropriate number of exemplars is obtained by using the AP algorithm. So SLICAP is a suitable algorithm for unsupervised segmentation.

4.1 Performance Evaluation

The boundary performance evaluation based on the F-measure of the above mentioned methods is reported in Table 1. We see that the F-measure of SLICAP using similarity A exceeds 0.65, suggesting that SLICAP well matches object boundaries. Although the performance of SLICAP using similarity B is not as outstanding as that of SLICAP using similarity A, it outperforms Ncuts and SLICKM. In addition, the range of similarity matrix of SLICAP (similarity C) is lower than others, which may inflect its performance. Note that, in this paper, we use the "hard" boundary representation as the segmentation criterion instead of the "soft" boundary representation. Therefore, the results of obtained boundaries are not optimized in terms of the benchmark of BSD.

Table 1: Boundary performance evaluation based on the F-measure of SLICAP against other methods on BSD

Method	Mean cost time	Mean cost time for clustering
Ncuts	91.3105	—
SLICKM (average)	11.1789	0.2008
SLICAP (similarity A)	21.4791	7.7685

The region performance evaluation based on PRI, VoI, and BDE is shown in Table 2. In terms of PRI, SLICAP using similarity A is close with SLICAP using similarity B, and they are better than the other methods. In terms of VoI and BDE, SLICAP using similarity A outperforms the other segmentation algorithms consistently. Compared with SLICAP using similarity A, SLICAP using similarity B demonstrates competitive performance. As a result, we see that the Euclidean distance is more appropriate for the AP clustering in this framework.

Table 2: Region performance evaluation based on PRI, VoI, and BDE of SLICAP against other methods on BSD

Method	F-measure
Ncuts	0.5893
SLICKM (average)	0.5831
SLICKM (best of 200)	0.6312
SLICAP (similarity A)	**0.6570**
SLICAP (similarity B)	0.6313
SLICAP (similarity C)	0.5988

4.1 Running Time

The mean cost time of the three methods for per image on BSD is shown in Table 3. Since the clustering procedure is not required for Ncuts, there is a dash at the corresponding position. The cost time of SLICAP (similarity B) and SLICAP (similarity C) is close with that of SLICAP(similarity A), and the cost time of SLICKM (best of 200) is close with that of SLICKM (average). They are thus not listed in Table 3.

In Table 3, we see that the mean cost time of SLICAP (similarity A) for clustering per image is more than that of SLICKM (average). However, thinking about that SLICKM (best of 200) needs to be run two hundred times, the total time consumed by SLICKM (best of 200) is actually much more than that of SLICAP (similarity A). On average, SLICAP takes 21.48 seconds to segment an image of size 481*321, where 10 seconds are for SLIC and only 7.8 seconds for the AP clustering. The SLICAP method could be implemented in real-time if using C language programming for a practical application (producing superpixels is less than half second if using SLIC executable file in Windows).

We point out that the settings of the parameters in SLICAP would affect its running time, such as the maximum number of iterations, the threshold of convergence value and the damping factor.

Table 3: Cost time of the three methods for each image.

Method	PRI	VoI	BDE
Ncuts	0.7801	3.0475	12.7841
SLICKM (average)	0.7875	3.0528	12.8173
SLICKM (best of 200)	0.8006	2.5377	11.4315
SLICAP (similarity A)	**0.8147**	**2.1108**	**9.9034**
SLICAP (similarity B)	**0.8155**	2.4241	10.6973
SLICAP (similarity C)	0.7807	2.5358	12.4449

5. Conclusion

We propose a novel approach based on superpixel to image segmentation. This approach builds a similarity matrix after using the SLIC superpixel algorithm, and then merges these superpixels into several regions by the AP clustering algorithm with the similarity matrix. The results of the experiment on BSD show that it performs very well both in boundary-based and region-based assessments. Moreover, the number of targets is determined automatically. On the other hand, this method uses only color information and does not exploit the texture and spatial information of the image. We are currently studying how to utilize texture or spatial information to improve segmentation performance.

References

[1] X. Ren and J. Malik, "Learning a classification model for segmentation," in Proc. IEEE Conf. International Conference on Computer Vision (ICCV), pp. 10-17, 2003.

[2] Z. Ren and G. Shakhnarovich, "Image segmentation by cascaded region agglomeration," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 2011-2018, 2013.

[3] R. Achanta, A. Shaji, Smith K, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, 2012.

[4] C. Y. Hsu and J. J. Ding, "Efficient image segmentation algorithm using SLIC superpixels and boundary-focused region merging," in Proc. IEEE Conf. Information, Communications and Signal Processing , pp. 1-5, 2013.

[5] C. Y. Ren and I. Reid, "gSLIC: A real-time implementation of SLIC superpixel segmentation,"

University of Oxford, Department of Engineering, Technical Report, 2011.

[6] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," Science, vol. 315, no. 5814, pp. 972-976, 2007.

[7] B. J. Frey and D. Dueck, "Mixture modeling by affinity propagation," in Advances in Neural Information Processing Systems, vol. 18, pp. 379, 2006.

[8] J. Shi and J. Malik, "Normalized cuts and image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp. 888-905, 2000.

[9] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881-892, 2002.

[10] Z. Li, X. M. Wu, and S. F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) , pp. 789-796, 2012.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in Proc. IEEE Conf. International Conference on Computer Vision (ICCV), vol. 2, pp. 416-423, 2001.

[12] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pp. 898-916, 2011.

[13] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 929-944, 2007.

[14] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," Computer Vision and Image Understanding, vol. 110, no. 2, pp. 212-225, 2008.

[15] J. Pont-Tuset and F. Marques, "Measures and meta-measures for the supervised evaluation of image segmentation," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), pp. 2131-2138, 2013.

[16] M. Meilă, "Comparing clusterings: An axiomatic view," in Proc. International Conference on Machine Learning, pp. 577-584, 2005.

[17] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," Computer Vision and Image Understanding, vol. 110, no. 2, pp. 260-280, 2008.

[18] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary information integration," in Proc. European Conference on Computer Vision (ECCV), pp. 408-422, 2002.