

Robust Image Segmentation Using Contour-guided Color Palettes

Xiang Fu¹, Chien-Yi Wang¹, Chen Chen¹, Changhu Wang^{2*}, C.-C. Jay Kuo¹

¹University of Southern California, Los Angeles, California, USA

²Microsoft Research, Beijing, China

{xiangfu, chienyiw, chen80}@usc.edu, chw@microsoft.com, cckuo@sipi.usc.edu

Abstract

The contour-guided color palette (CCP) ¹ is proposed for robust image segmentation. It efficiently integrates contour and color cues of an image. To find representative colors of an image, color samples along long contours between regions, similar in spirit to machine learning methodology that focus on samples near decision boundaries, are collected followed by the mean-shift (MS) algorithm in the sampled color space to achieve an image-dependent color palette. This color palette provides a preliminary segmentation in the spatial domain, which is further fine-tuned by post-processing techniques such as leakage avoidance, fake boundary removal, and small region mergence. Segmentation performances of CCP and MS are compared and analyzed. While CCP offers an acceptable standalone segmentation result, it can be further integrated into the framework of layered spectral segmentation to produce a more robust segmentation. The superior performance of CCP-based segmentation algorithm is demonstrated by experiments on the Berkeley Segmentation Dataset.

1. Introduction

Automatic image segmentation is a fundamental problem in computer vision. It plays an important role in diverse applications, such as object detection, scene parsing, and image retrieval. It partitions an image into a small number of disjointed coherent regions with low-level features, with the goal of minimizing intra-variance and maximizing inter-variance among regions. It is desired that the segmentation result is close to human semantic understanding and not sensitive to parameter setting and/or image content.

To segment an image, pixel (or superpixel) grouping in the spatial and spectral domains were performed in the literature. Typically, spatial-domain pixel grouping is guided by contours [2, 7] while spectral-domain pixel grouping is

achieved by clustering in a color space [17, 4, 5, 8, 11, 12, 18]. Thus, contours and colors are two widely used features in image segmentation, yet each of them has its own limitations. For example, contours are not reliable if they are short and fragmented. They might fail to separate two regions if parts of their common boundaries are blurred and/or with a low contrast. The color feature is not effective to handle regions with textures or gradual color transition, leading to over-segmentation. One common challenge in these methods is the selection of proper parameters, such as the color clustering bandwidth. In general, these optimal parameters are image dependent and difficult to determine. In this work, we integrate contour and color cues under one unified framework, and propose the contour-guided color palette (CCP) for robust image segmentation. That is, it has only one key parameter and its performance is stable when the parameter lies in a suitable range.

The basic idea of CCP is described as follows. To find representative colors of a given image, we collect color samples from both sides of long contours, and conduct the mean-shift (MS) algorithm [4] in the sampled color space to define an image-specific color palette. This scheme reduces color complexity of the original image, yet keeps a sufficient number of representative colors to separate distinctive regions and yield a preliminary segmentation. This result is further refined by post-processing techniques in the spatial domain, which leads to a robust standalone segmentation. The CCP result can be applied to any superpixel-based segmentation algorithm by replacing the over-segmentation layer, such as mean-shift (MS) [4], Felzenszwalb and Huttenlocher's graph-based (FH) [8], and SLIC [1] superpixels. Furthermore, it can be integrated into the layered spectral segmentation framework, such as multi-layer spectral segmentation (MLSS) [11] and segmentation by aggregating superpixels (SAS) [12], and used as a coarse layer in this context for a more robust segmentation. The superior performance of CCP-based segmentation algorithms are demonstrated in the experiments on the Berkeley Segmentation Dataset (BSDS) [13].

The rest of this paper is organized as follows. Related

*Corresponding Author

¹The MATLAB code of CCP Segmentation can be downloaded at https://github.com/fuxiang87/MCL_CCP.

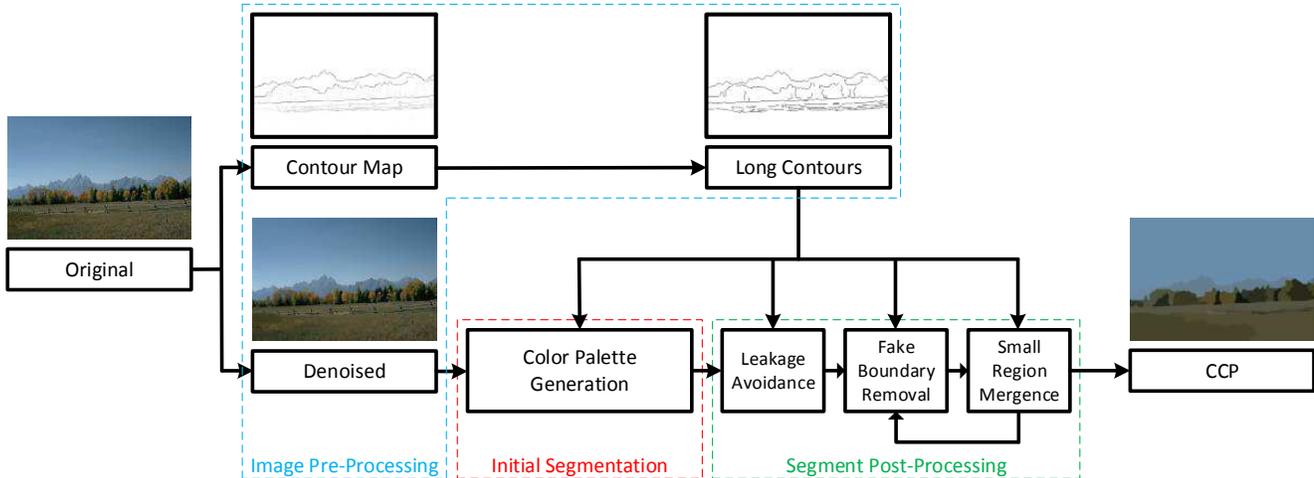


Figure 1. The block diagram of the proposed contour-guided color palette (CCP) method.

work is reviewed in Sec. 2. The CCP method is described in detail in Sec. 3. The advantages of CCP over MS are analyzed in Sec. 4. Then, the integration of CCP with layered spectral segmentation is introduced in Sec. 5. Experimental results are shown in Sec. 6. Finally, concluding remarks are given in Sec. 7.

2. Related Work

According to the studies on human visual perception [10, 3], people pay more attention to dissimilarities between two regions and lean to group similar regions in appearance, which correspond to the contour (1D) and regional (2D) cues, respectively. Both of the two cues are needed for a better image segmentation. Regional cues are contributed by color and texture. Most image segmentation algorithms can be classified into two categories, *i.e.*, region-based and contour-based methods.

Region-based methods find the similarity among spatially connected pixels and group them together using surface properties such as luminance and color. Representative approaches include watershed [17], k-means, mean-shift (MS) [4], normalized cuts (NCut) [5, 18], Felzenszwalb and Huttenlocher’s graph-based (FH) [8], multi-layer spectral segmentation (MLSS) [11], and segmentation by aggregating superpixels (SAS) [12]. However, these methods might neglect obvious discontinuities between two regions. To overcome this limitation, contour-based methods, such as gPb-OwT-UCM [2], and saliency driven total variation (SDTV) [7] were developed to find connected regions blocked by detected contours. However, it is still challenging to detect closed contours in low-contrast or blurred regions for segmentation.

One can combine region and contour cues to overcome their individual limitations, and several ideas were introduced in [9, 16]. For example, one can take the contour cue as a post-processing step to correct region-based segmen-

tation results or treat the contour as a barrier in an affinity measure.

In this work, a new method called CCP is proposed to effectively integrate contour and color cues. Unlike the other methods, we take the contour cue as guidance to form an image-dependent color palette. It reduces color complexity of the original image, yet keeps a sufficient number of representative colors to separate distinctive regions. The CCP method is detailed in the following section.

3. Contour-guided Color Palette Method

3.1. System Overview

The basic idea of the CCP method can be simply stated as follows. Long contours play an important role in image segmentation since they provide useful spatial-domain information in region partition. However, they may not form a closed region due to weak boundaries in some parts, leading to the leakage problem. To assist contour-guided segmentation, we use the color information as an auxiliary cue. That is, we collect color samples along both sides of each long contour and perform clustering in the sampled color space for color quantization. Once color is quantized, we get a number of closed regions with long contours as their boundaries. This initial segmentation can be further refined by post-processing techniques in the spatial domain. Fig. 1 shows the block diagram of the CCP method, which mainly consists of three modules: (1) image pre-processing, (2) contour-guided color palette generation for an initial segmentation, and (3) segment post-processing.

The image pre-processing module includes denoising and contour extraction. There are many standard algorithms to select for this module. In our implementation, we adopt the bilateral filtering scheme [19] for denoising. And, we apply the structured edge detection [6] method to the original input image to obtain a contour map with pixel value in-

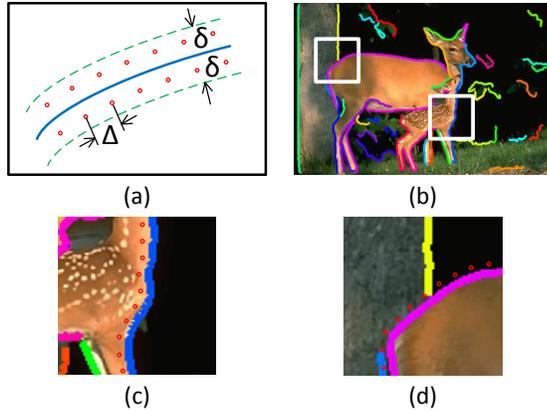


Figure 2. Illustration of banded region of interest (B-ROI) for color palette generation. (a) shows a B-ROI with a bandwidth of 2δ . Pixels are sampled from both sides of the contour with a uniform stepsize Δ . (b) is an example image overlaid with detected long contours. Long contours are indicated by different colors. (c) and (d) are the zoom-in of two local B-ROI's of (b). In (c), the colors of the pixels labeled in red along one side of the B-ROI look similar without an obvious jump. In (d), the colors of the pixels labeled in red change with one large jump.

dicating the probability of being a contour point. Then, long contours are selected from the contour map. The contour information will be used to generate the desired color palette, based on which the initial segmentation result is directly obtained (module 2), followed by an effective post-processing step (module 3). Both modules 2 and 3 are guided by the contour information.

3.2. Color Palette Generation

The well-known color clustering algorithms, such as k-means and MS [4] clustering, consider the color distribution of all pixels or superpixels in an image. However, not all pixels and their associated colors are equally important for the segmentation purpose as illustrated by the following two examples. First, the strong color-varying pixels inside a texture region (*e.g.*, a large number of flowers in a garden in Fig. 3(a)), where the complexity of the color representation increases, are actually of less importance. Second, the pixels of similar colors inside a homogeneous region (*e.g.*, a large near-white building in Fig. 3(e)), that gives many redundant color samples in the color space, are also not that important. A relatively minor variation in these images (*e.g.*, the flower density and the wall size, respectively) will affect the color-based segmentation. Generally speaking, these algorithms are sensitive to their parameter settings, and it is challenging to automatically find a good parameter set for an arbitrary image.

To develop a robust segmentation algorithm, we attempt to reduce the influence of color variations in an image by selecting a set of representative colors. To achieve this,

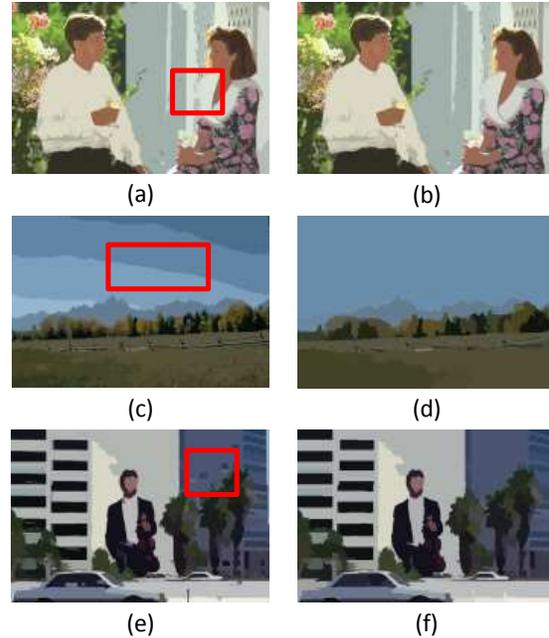


Figure 3. Comparison of CCP segmentation results before and after post-processing. Please focus on the squared regions in red: (a) shows a long contour straddled by two regions with similar colors; (c) shows the fake boundary in the sky due to gradual color transition; (e) shows small segments in the background building region. (b), (d), and (f) are the post-processed results of (a), (c), and (e), respectively.

we focus on key regions and obtain color samples accordingly. For image segmentation, one would like to have large segments and ignore small ones. Since large segments are enclosed by long contours, we can define a banded region of interest (B-ROI) for each long contour and its neighborhood. The B-ROI is centered at the contour location with a bandwidth of 2δ as shown in Fig. 2(a). After obtaining the B-ROI, we sample pixels from both sides of the contour with a uniform stepsize Δ and have their colors in the Lab color space [21] to form a set of representative colors. In the implementation, we used the structured edge detection algorithm [6] to extract the contour and set $\delta = 2$ and $\Delta = 1$ pixels, respectively.

We observe two typical cases for pixels along one side of the B-ROI for different images. First, the color remains about the same or changes gradually without an obvious jump. Second, the color changes with one or several large jumps, yet each interval between two jumps does have a similar color. These two cases are shown in Figs. 2(b)-(d), where Fig. 2(b) is an illustrative image overlaid with long contours while Figs. 2(c) and (d) provide the zoom-in images of two local regions of Fig. 2(b) and correspond to the two cases, respectively. For Case 1, as shown in Fig. 2(c), color samples can be further reduced to their average color. For Case 2, we need to select multiple color samples, each

of which represents the color in one interval. As shown in Fig. 2(d), the B-ROI goes through the deer body at one side and two background regions at the other side. In this case, we need to split the color samples into two groups and each group is represented by its average color.

Further color simplification required by Case 1 and 2 can be achieved by MS clustering with bandwidth parameter h_r in the spectral domain. Another is to adopt MS clustering with different bandwidth parameters for sampled colors located in different regions of the image. Since the object of interest is usually in the central region while the background is in the boundary region of an image, we adopt two bandwidth parameters, *i.e.*, a smaller one and a larger one (h_{rc}, h_{rb}), for sampled colors in the central and boundary regions, respectively. The final representative color set is called the color palette of the input image. Then, a color-quantized image can be obtained by replacing the color of each pixel with its most similar color in the color palette. In this way, an initial segmentation result is obtained.

3.3. Segment Post-Processing

Three post-processing techniques are proposed to better the segmentation result: 1) leakage avoidance by contours, 2) fake boundary removal, and 3) small region mergence, as illustrated in Fig. 3.

The first problem arises when there is a long contour straddled by two regions with similar colors. One such example is given in Fig. 3(a), where the white fence and the white collar are close in color but separated by a long contour. After color quantization, the fence is mingled with the collar to yield complicated patterns and, as a result, these two regions are blurred. This is known as the leakage problem. To avoid this, we check the regions along each side of the contour in the B-ROI. After color quantization, even if both sides of the contour are quantized into the same color, they are still separated by the long contour.

The second problem occurs when there is a smooth color transition over a large region. For example, the sky color in Fig. 3(c) changes smoothly and it is split into multiple regions due to color quantization. This fake boundary can be removed by checking the common boundary of adjacent regions. We consider the ratio of the length of the common boundary and the minimum perimeter of the two regions, which indicates the relative significance of the common boundary. If the common boundary is significant and not overlaid much with detected long contours, these two regions will be merged. By this criterion, isolated regions have a high priority to be merged when there is not a long contour around them.

The third problem occurs in the textured area such as the background building with small windows in Fig. 3(e). They are merged to the closest “effective neighbors” for simplicity. This can be implemented by merging a small region to

its neighbor region of a similar color but without a contour in between.

Since region aggregation is irreversible, we need to pay special attention to the order of fake boundary removal and small region mergence. In the beginning, region sizes are relatively small. The small region mergence process might merge two similar regions in the dark or blurred area, leading to the leakage problem. However, the fake boundary removal process does not have this side effect. For this reason, we conduct fake boundary removal before small region removal. Fake boundary removal and small region mergence can be conducted iteratively to achieve better performance. We conduct the iteration twice in the implementation. The post-processed results of Figs. 3(a), (c) and (e) are shown in Figs. 3(b), (d) and (f), respectively.

4. Comparison of MS and CCP

In Fig. 4, we compare in detail the segmentation results of mean-shift (MS) method and our CCP method for three typical images, denoted as #1, #2 and #3 from the left to the right. For MS, we select three best spectral bandwidth (BW) parameters from all the odd numbers between 5 and 25, resulting in 7 (small), 13 (medium) and 19 (large); for CCP, we select the spectral BW parameters as $h_r = 5$ (small), $(h_{rc}, h_{rb}) = (5, 7)$ (medium) and $h_r = 7$ (large) in the color palette generation process. The spatial BW parameter is set to 7 in all MS results while no spatial BW parameter is required by CCP. We can see that CCP provides simplified segmentation results, which are more consistent with human perception and can serve as standalone solutions. In contrast, MS gives highly over-segmented images that are not acceptable to human eyes even if the spectral BW parameter is large enough. Similar conclusions were drawn from all the 300 images in the Berkeley Segmentation Dataset (BSDS) [13]. It is no doubt that CCP visually outperforms MS by a significant margin.

Selection of proper spectral and spatial BW parameters for MS is actually a challenging task. The quality of the MS segmentation result is sensitive to these two parameters. They are not only image dependent but also region dependent. To the best of our knowledge, there is no automatic mechanism to select good BW parameters. For comparison, CCP only demands one BW parameter, and its results are stable over a range of BW values as illustrated in Fig. 4. This avoids the huge burden of performance fine-tuning.

To explain the superior performance of CCP over MS, we list the representative color numbers and the boundary F-measures (harmonic mean of precision and recall, defined in [14]) under three spectral BW parameters for the three images in Table 1. We also provide the average results of the entire BSDS300 dataset. A smaller spectral BW parameter usually generates more representative colors as illustrated by the numbers in the same column in the upper half of the

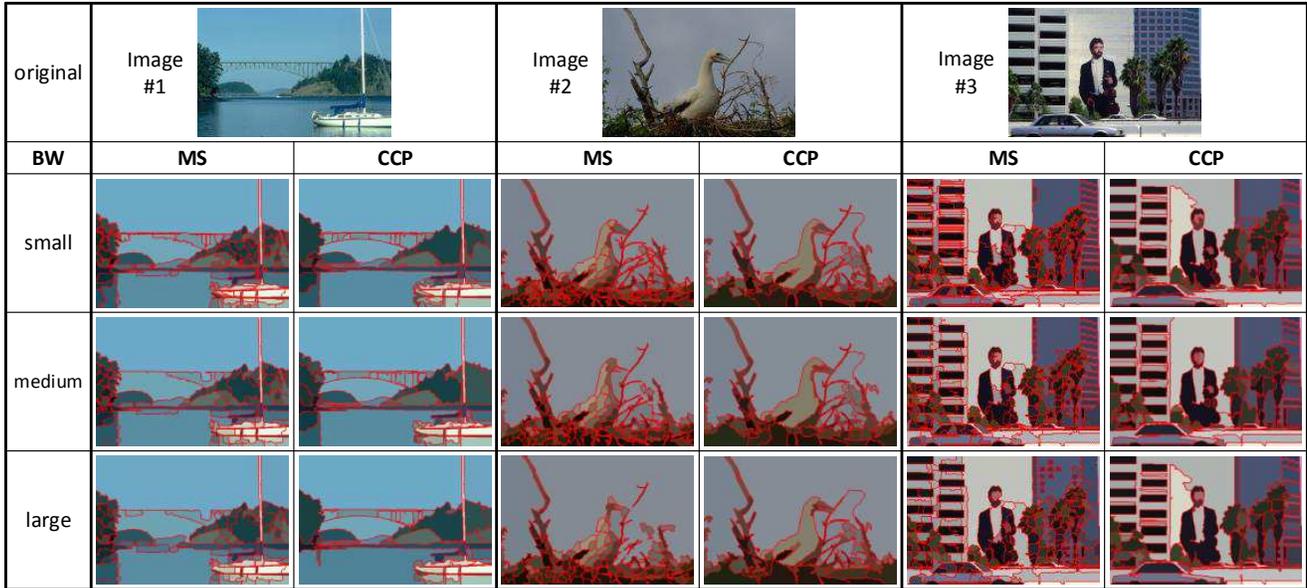


Figure 4. Comparisons of segmentation results by MS and CCP for three typical images, with different spectral BW parameters.

table. The corresponding number of representative colors of CCP is significantly less than those of MS, although the three BW parameters of CCP are smaller than their counterparts of MS. In addition, CCP can achieve much better boundary F-measures than MS, as shown in the lower half of the table. This indicates a better boundary adherence with respect to human ground-truth boundaries. These two comparisons show the power of color sampling along the contours adopted by CCP. Through color sampling, we can eliminate color samples in regions of little significance and merge these regions with other important ones, as shown in the mountain, branch and tree regions of image #1, #2 and #3, respectively. Furthermore, because CCP yields fewer color samples in the color space, we can adopt a smaller BW parameter without increasing the number of representative colors too much. In this way, CCP can reduce the risk to make two colors along a significant contour get mixed, and thus avoid a severe leakage problem which usually occurs in the MS method. For comparisons, please look at the boundaries between the bridge and the sky, those between the bird and the sky, and those between the face and the building in Fig. 4.

For a segmented image, we count the number of pixels for a specific representative color and sort the color index according to the number of associated pixels in a descending order. Then, we plot the cumulative normalized histogram as a function of the representative color index, as shown in Fig. 5. The curve reaches 100% when all representative colors are used. Let us use Image #1 as an example. The blue, green and red curves are obtained using large, medium, and small spectral BW parameters. The three curves of CCP reach 100% at color index #54, #68 and

Table 1. Comparisons of the numbers of representative colors (upper) and the boundary F-measures (lower) by MS and CCP under three BW parameters for the three typical images. We also provide the average results of the entire BSDS300 dataset.

BW	MS				CCP			
	#1	#2	#3	All	#1	#2	#3	All
s.	116	116	217	192	66	83	118	81
m.	100	117	218	178	68	85	122	77
l.	98	113	215	173	54	75	107	63
s.	0.70	0.67	0.72	0.59	0.75	0.75	0.78	0.68
m.	0.69	0.64	0.75	0.60	0.75	0.75	0.80	0.68
l.	0.68	0.60	0.76	0.60	0.74	0.74	0.78	0.68

#66 while those of MS reach 100% at color index #98, #100 and #116, respectively, as indicated by the data in Table 1. Meanwhile, CCP can achieve around 10% higher boundary F-measures than MS. There are a few dominant colors in simple images such as Images #1 and #2, which can be caught by both CCP and MS. CCP reaches a higher percentage than MS with these dominant colors. Image #3 is more complicated in its content and more representative colors are needed. In all three cases, along with better boundary adherence, the CCP curves are closer to the upper-left corner of the figure than MS. This indicates that CCP can use fewer colors to represent a larger region of an image and provide a more simplified result. Similar conclusions were drawn from all 300 images in the BSDS dataset.

5. Layered Affinity Models using CCP

Spectral segmentation has received a lot of attention in recent years due to its impressive performance [5, 18]. It begins with a graph representation of a given image, where

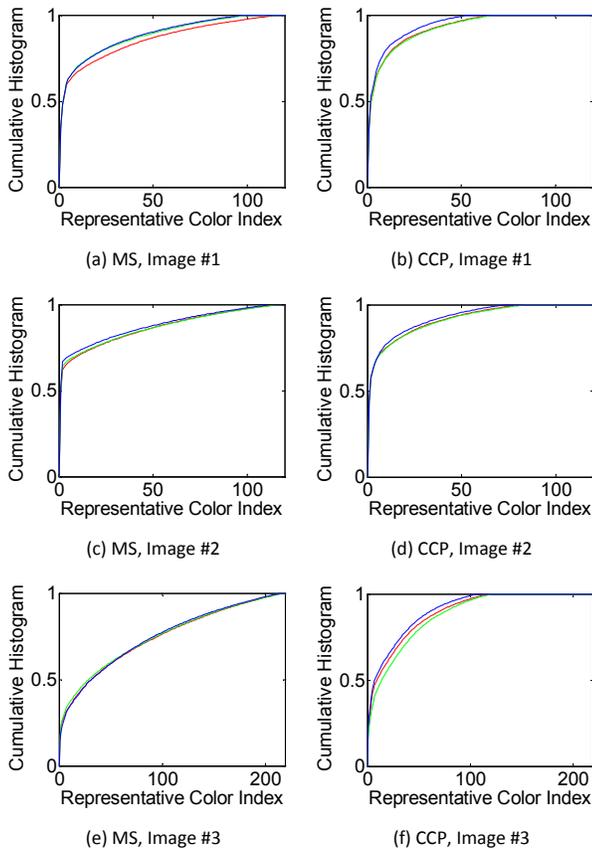


Figure 5. Plots of the cumulative histogram versus representative color indices for MS and CCP on three typical images, where the blue, green and red curves are obtained using large, medium and small spectral BW parameters.

each pixel is a node. Then, a sparse affinity matrix is created to measure the similarity between nearby nodes, while ignoring the connection among distant nodes even if they are in the same homogeneous region; say, two distant nodes in the same sky region. The simplification of sparse affinity matrices often leads to over-segmentation. To overcome this problem, a layered affinity model was introduced to allow more connections, such as the full pairwise affinity in MLSS [11] and the bipartite graph partitioning in SAS [12]. These methods share one common idea, namely, building a graph model consisting of multiple layers. The finest one is the pixel layer as constructed by the standard spectral segmentation method. Then, one can add a couple of coarse layers on top of the pixel layer, where each coarse layer uses superpixels as its nodes and defines an affinity matrix accordingly. Typically, these superpixel layers are constructed using the MS [4] and the FH graph-based [8] methods. Finally, nodes between different layers are connected by an across-affinity matrix. Although these methods share the same basic idea, they differ in the details of the layered

affinity matrix implementation.

To further improve the segmentation result of CCP, we can leverage the two layered affinity models proposed in MLSS and SAS. The integrated methods are called CCP-LAM (where LAM denotes “layer-affinity by MLSS”) and CCP-LAS (where LAS denotes “layer-affinity by SAS”), respectively. CCP-LAM and CCP-LAS can be easily obtained by replacing the superpixel layers in MLSS and SAS, respectively, by the CCP segmentations as described in Sec. 3, with the pixel layer kept as the finest layer.

It was observed in [11] and [12] that the final image segmentation result can benefit from the diversity of multiple coarse layers. Following this line of thought, we create multiple CCP segmentations by varying bandwidth parameter h_r or (h_{rc}, h_{rb}) of the MS algorithm in the color palette generation process, which has been discussed in Sec. 3.2.

6. Experimental Results

In this section, we evaluate the performance of three CCP segmentation results by consider three parameter settings: 1) CCP-1, $h_r = 5$; 2) CCP-2, $(h_{rc}, h_{rb}) = (5, 7)$; and 3) CCP-3, $h_r = 7$. Furthermore, we take CCP-1, CCP-2 and CCP-3 as three coarse layers in the context of spectral segmentation with two layered affinity models (*i.e.*, LAM and LAS) to result in CCP-LAM and CCP-LAS methods. To achieve the optimal performance of CCP-LAM and CCP-LAS, we follow the procedure stated in [11, 12] to manually select the best segment number using the LAM or LAS graph.

We compare the performance of CCP-1, CCP-2, CCP-3, CCP-LAM and CCP-LAS with several benchmarking methods on the Berkeley Segmentation Dataset (BSDS) [13] in Table 2. The BSDS benchmark consists of 300 color images of size 481×321 pixels displayed either horizontally or vertically, and several hand-labeled segmentations were collected from different human subjects for each image. The benchmarking methods include NCut [18], MNCut [5], MS [4], FH [8], SDTV [7], RIS-HL [22], MLSS [11], and SAS [12]. Their numbers are taken from [7, 11, 12, 22].

As shown in Table 2, five performance metrics (*e.g.*, [7, 11, 12, 22]) are used for quantitative evaluation. They are: 1) Segmentation Covering (Cov) [2], which measures the region-wise covering of the ground truth by a segmentation; 2) Probabilistic Rand Index (PRI) [20], which counts the likelihood of pixel pairs whose labels are consistent between a segmentation and the ground truth; 3) Variation of Information (VoI) [15], which measures the amount of randomness in one segmentation that cannot be contained by the other; 4) Global Consistency Error (GCE) [13], which measures the extent to which one segmentation can be viewed as a refinement of the other; 5) Boundary Displacement Error (BDE) [9], which measures the average displacement error of boundary pixels between two seg-

mented images. The segmentation result is better if Cov and PRI are larger while the other three criteria (VoI, GCE and BDE) are smaller. The best and the second best results in Table 2 are highlighted in red and blue, respectively.

We draw the following conclusions from Table 2. First, CCP-LAM and CCP-LAS achieved the best performance in terms of all five metrics by a large margin. Second, all the three CCP methods had outstanding performance in the GCE and BDE metrics. This means that CCP yields an excellent segmentation with better boundary adherence and less displacement error with respect to the ground truth. It is worthwhile to emphasize that no image-dependent parameter was used in CCP-1, CCP-2 and CCP-3. The same parameter setting is applied to all the images. In contrast, a set of experiments were run in all other benchmarking methods, and the best result for each image was selected and used in performance computation.

Table 2. Performance comparison of several segmentation methods on the BSDS300 Dataset, where the best two results are highlighted in red (best) and blue (second best).

Algorithm	Cov	PRI	VoI	GCE	BDE
NCut [18]	0.44	0.7242	2.9061	0.2232	17.15
MNCut [5]	0.44	0.7559	2.4701	0.1925	15.10
MS [4]	0.54	0.7958	1.9725	0.1888	14.41
FH [8]	0.51	0.7139	3.3949	0.1746	16.67
SDTV [7]	0.57	0.7758	1.8165	0.1768	16.24
RIS-HL [22]	0.59	0.8137	1.8232	0.1805	13.07
MLSS [11]	0.53	0.8146	1.8545	0.1809	12.21
SAS [12]	0.62	0.8319	1.6849	0.1779	11.29
CCP-1	0.47	0.7900	2.8502	0.1046	11.26
CCP-2	0.48	0.7932	2.7835	0.1077	11.17
CCP-3	0.53	0.8014	2.4723	0.1270	11.29
CCP-LAM	0.68	0.8404	1.5715	0.1635	10.20
CCP-LAS	0.68	0.8442	1.5871	0.1582	10.46

Furthermore, Fig. 6 shows the segmentation results of ten images by MLSS, SAS, CCP-LAM and CCP-LAS for visual comparison. Again, CCP-LAM and CCP-LAS produced significantly better and meaningful segmentation results over MLSS and SAS in terms of visual appearance.

7. Conclusions

The contour-guided color palette (CCP) was proposed for robust image segmentation. This method effectively integrated the contour and color cues of an image, reduced its color complexity, and kept a sufficient number of distinctive colors to achieve the desired segmentation task. Based on the image-specific color palette, a preliminary segmentation was obtained and it was further fine-tuned by post-processing techniques. The CCP method produced an acceptable standalone segmentation result, which could be further integrated with layered affinity models for spectral segmentation. The superior performance of the proposed

CCP-LAM and CCP-LAS methods over existing state-of-the-art methods was demonstrated by extensive experimental results.

References

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2274–2282, 2012.
- [2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.
- [3] V. Bruce, M. A. Georgeson, and P. R. Green. *Visual Perception: Physiology, Psychology and Ecology*. Psychology Press, 4th edition, 2003.
- [4] D. Comaniciu and P. Meer. Mean Shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, May 2002.
- [5] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR’05*, pages 1124–1131, 2005.
- [6] P. Dollr and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV’13*, pages 1841–1848, 2013.
- [7] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof. Saliency driven total variation segmentation. In *ICCV’09*, pages 817–824, 2009.
- [8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, Sept. 2004.
- [9] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. In *ECCV’02*, pages 408–422, 2002.
- [10] J. J. Gibson. *The perception of the visual world*. Houghton Mifflin, 1950.
- [11] T. H. Kim, K. M. Lee, and S. U. Lee. Learning full pairwise affinities for spectral segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(7):1690–1703, July 2013.
- [12] Z. Li, X.-M. Wu, and S.-F. Chang. Segmentation Using Superpixels: A bipartite graph partitioning approach. In *CVPR’12*, pages 789–796, 2012.
- [13] 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 *ICCV’01*, pages 416–423, 2001.
- [14] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):530–549, May 2004.
- [15] M. Meilă. Comparing Clusterings: An axiomatic view. In *ICML’05*, pages 577–584, 2005.
- [16] X. Muñoz, J. Freixenet, X. Cufí, and J. Martí. Strategies for image segmentation combining region and boundary information. *Pattern Recogn. Lett.*, 24(1-3):375–392, Jan. 2003.
- [17] J. B. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundam. Inf.*, 41(1,2):187–228, Apr. 2000.

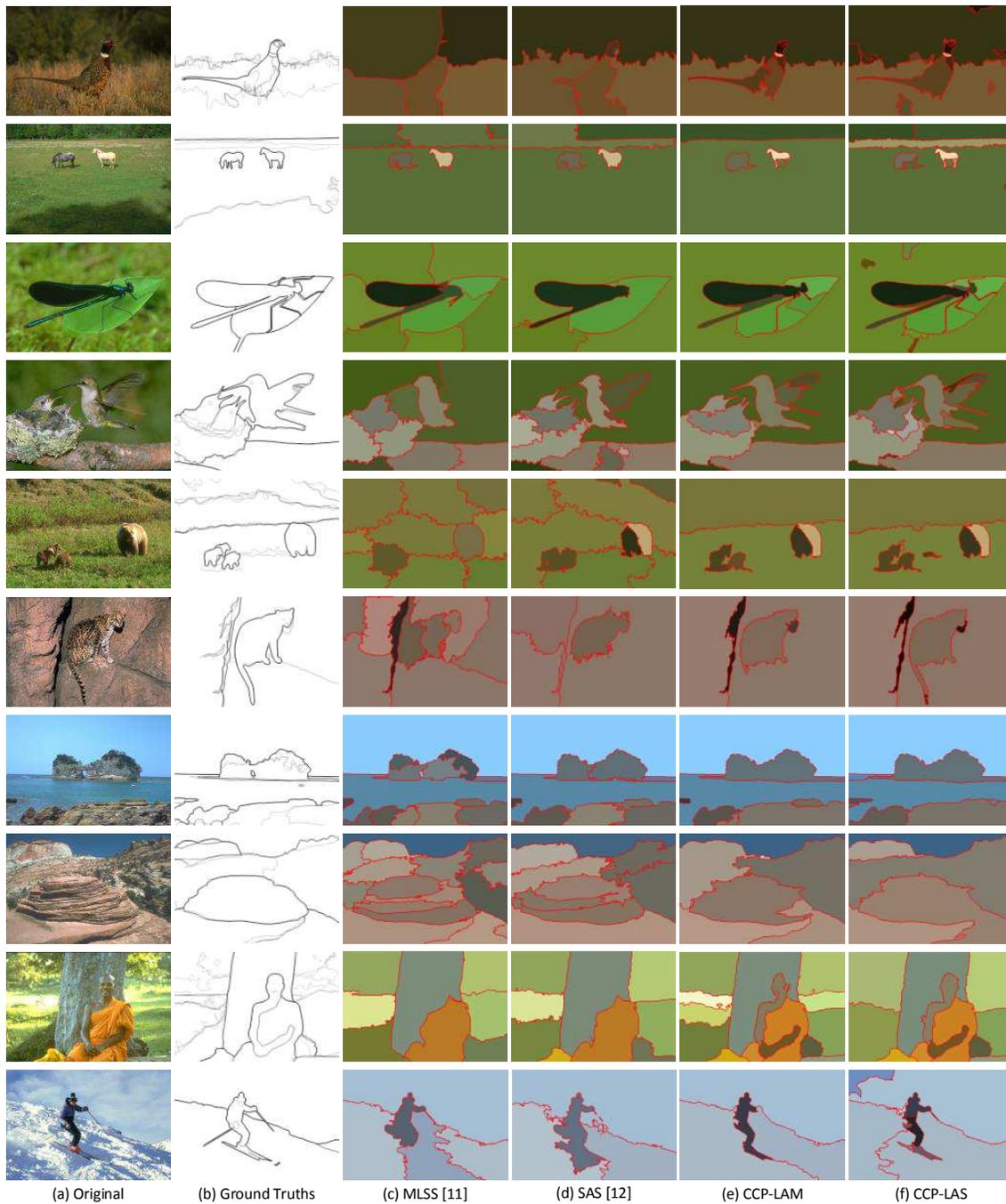


Figure 6. Visual comparisons of segmentation results of CCP-LAM and CCP-LAS against two state-of-the-art methods MLSS and SAS. For more result comparisons, please refer to the supplementary material.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, Aug. 2000.

[19] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *ICCV '98*, pages 839–846, 1998.

[20] R. Unnikrishnan, C. Pantofaru, and M. Hebert. Toward objective evaluation of image segmentation algorithms. *IEEE*

Trans. Pattern Anal. Mach. Intell., 29(6):929–944, June 2007.

[21] Wikipedia. Lab color space - Wikipedia, the free encyclopedia, 2015 (accessed April-2015).

[22] J. Wu, J. Zhu, and Z. Tu. Reverse image segmentation: A high-level solution to a low-level task. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.