

Research Article

Objectness Supervised Merging Algorithm for Color Image Segmentation

Haifeng Sima,¹ Aizhong Mi,¹ Zhiheng Wang,¹ and Youfeng Zou²

¹The School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China

²Henan Polytechnic University, Jiaozuo, China

Correspondence should be addressed to Aizhong Mi; miaizhong@hpu.edu.cn

Received 17 June 2016; Accepted 14 September 2016

Academic Editor: Wonjun Kim

Copyright © 2016 Haifeng Sima et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ideal color image segmentation needs both low-level cues and high-level semantic features. This paper proposes a two-hierarchy segmentation model based on merging homogeneous superpixels. First, a region growing strategy is designed for producing homogenous and compact superpixels in different partitions. Total variation smoothing features are adopted in the growing procedure for locating real boundaries. Before merging, we define a combined color-texture histogram feature for superpixels description and, meanwhile, a novel objectness feature is proposed to supervise the region merging procedure for reliable segmentation. Both color-texture histograms and objectness are computed to measure regional similarities between region pairs, and the mixed standard deviation of the union features is exploited to make stop criteria for merging process. Experimental results on the popular benchmark dataset demonstrate the better segmentation performance of the proposed model compared to other well-known segmentation algorithms.

1. Introduction

Color image segmentation is an important task in image analysis and understanding. It is widely used in many image applications as a critical step. The primary purpose of segmentation is to divide an image into meaningful and spatially connected regions (such as object and background) on the basis of diverse properties. Existing segmentation approaches can predominantly be divided into the following four categories: region-based methods, feature-based methods, boundary-based methods, and model-based methods. Most existing image segmentation methods consider only the low-level features [1–4] and it is difficult to obtain ideal segmentation results without high-level visual features. Therefore, many researchers aim to build a bridge between low-level features and visual cognitive behaviors of human. Most segmentation methods integrated high-level knowledge, which are learning cognitive information from underlying features, and they obtain the corresponding label of elements and textons of image. Researchers aim to find high-level summary tags or labels to represent all the underlying features. With all these elements and textons, the segmentation can employ affinity

computing to make decision of pixels labels of semantic objects and regions. This procedure employs knowledge-assisted multimedia analysis model to bridge the gap between semantics and low-level visual features [5, 6].

Many segmentation methods based on objects detection have shown their capability by utilizing low-level visual features [7, 8]. They integrated basic visual features to detect homogeneous regions for image segmentation. Even though these low-level features have shown favorable results, they are unbecoming in many complex scenes, such as texture, luminance, and overlapping. To deal with this problem, some researchers attempted to combine different features together [9]. However, most of them integrated these features in a simple way. Few studies evaluated the significance of object features to the overall segmentation. Besides, since most previous segmentation methods lack high-level knowledge about objects, it is difficult for them to discriminate semantic regions from cluttered images.

The objectness is a pixel-level characteristic obtained from training underlying characteristics, so we will combine it with color and texture to achieve better segmentation in this paper. The objectness is proposed by Alexe et al. [10]

as a novel high-level property estimated from multiple low-level features, and it is widely used for saliency computing, object detection, cosegmentation, object tracking, and so forth. It is designed for measuring the likelihood of a window containing a complete object in it at first. This measurement employs four cues in the calculation of objectness: multiscale saliency map, color contrast, edges density, and superpixels straddling. The four cues have been proved to be useful for detecting whole object from background. Jiang et al. [11] proposed a strategy for computing regional objectness by computing average objectness values from salient regions in slide windows. Through the above analysis, we find that the objectness is an effective feature to describe the integrity and semantic regions of objects in the image.

In order to tackle those issues remaining in existing segmentation methods, we developed a novel color image segmentation method based on merging superpixels supervised by regional objectness and underlying characteristics in this paper. Firstly, we adopt seeds growing strategy for superpixels computing. The seed pixels are selected from the blurring image from total variation (TV) diffusion image. Next, we introduce the growing method to expand these seed pixels to superpixels according to similarity of color and texture features at local areas. In the merging procedure, we define a combined feature consisting of color, texture, and objectness to measure the similarity between adjacent superpixels and implement merging procedure to form the segmentation results.

The main contributions of this paper are summarised as follows: (1) we develop efficacious superpixels extraction method to detect homogeneous regions with hybrid features, (2) we propose a combined histogram descriptor of color and texture feature for identifying semantic regions, and (3) we design a fast and less sampling objectness computing model and integrate it with low-level features for reasonable merging.

The rest of the paper is organized as follows: our segmentation method is shown in Section 2 including superpixels computing, objectness computing, combined feature definition of superpixels, and merging rules. The experiments on the Berkeley Segmentation Datasets are performed in Section 3 and the conclusion is made in Section 4. The segmentation process and intermediate results are shown in Figure 1.

2. Objectness Supervised Superpixels Merging Segmentation

In our method, the segmentation process consists of the following two stages. First, the input image is segmented into hundreds of uniform homogeneous superpixels with hybrid features including RGB channels and diffusion image. Second, we construct a descriptor for superpixels with invariant color-texture histograms as low-level features and define the objectness of all pixels based on sparse sampling of objectness windows as high-level features. Then, we develop the merging rules combined with two levels of features for reasonable segmentation. The detailed procedure of our method is as follows.

2.1. Compact Superpixels Computing. In the color image segmentation, to obtain homogeneous superpixels is an ideal preprocessing technique to simplify the segmentation task. A superpixel is defined as a connected homogeneous image region often called oversegmentation computed by certain strategies. An image is represented by hundreds of superpixels instead of many redundant pixels which can simplify the image representation and segmentation. There are many segmentation strategies for superpixels computing such as SLIC [12], N-cut [13], watershed [14], and Mean-Shift [15], and various strategies result in different property and quality of superpixels. Each method has its own advantages; however, all these methods did not perform well when encountering texture areas.

For segmentation task, the real boundary keeping and homogeneous superpixels extraction are critical for ideal final segmentation. Texture feature is consisted of group pixels that requires enough pixels to represent the texture patterns in superpixels. How to retain the superpixels boundaries surrounded complete texture is a difficult problem for superpixels computing algorithms. On the other hand, smaller scale superpixel is conducive to saving the real and accuracy boundaries. An important branch of superpixel extraction methods is to predefine segments number before segmentation and it is easy to control the running time and segmentation quality by adjusting segments number in these methods. It is also convenient to integrate multifeatures in these methods. The other data driven superpixels extraction methods are randomness in contrast. So we devolve our method based on the above analysis. We select seeds in the smooth area in the uniform grids of images and grow them to be superpixels with similarity computing. Implementation details are as follows.

2.1.1. Texture Smoothing. In order to enhance the descriptive ability of local regions, we detect texture information by integrating the diffusion feature and original channels into the region growing procedure. Perona and Malik proposed [16] an anisotropic diffusion filtering method for multiscale smoothing and edge detection; this diffusion process is a powerful image processing method which encourages intraregion smoothing in preference to smoothing across the boundaries. This filtering method tries to estimate homogeneous region of image with surface degradation and the edge strengths of local structure. To compute texture features, we adopt nonlinear diffusion to extract texture areas and enhance the edges simultaneously. It is easier to smooth regions and detect real boundary of image by TV diffusion model in texture area [17]. Therefore, the TV flow diffusion method is used for blurring local area of color image for extracting nonoverlapped superpixels. The implicit equation of nonlinear diffusion technique of TV is computed by

$$\frac{\partial I_c^i(x, y, t)}{\partial t} = \text{div} \left(g \left(\sum_{k=1}^3 |\nabla I_k| \right) |\nabla I^i| \right). \quad (1)$$

The initial value of I_c is single channel c of the original image in RGB space, and t is the iteration times. g is the diffusion function defined in [17]

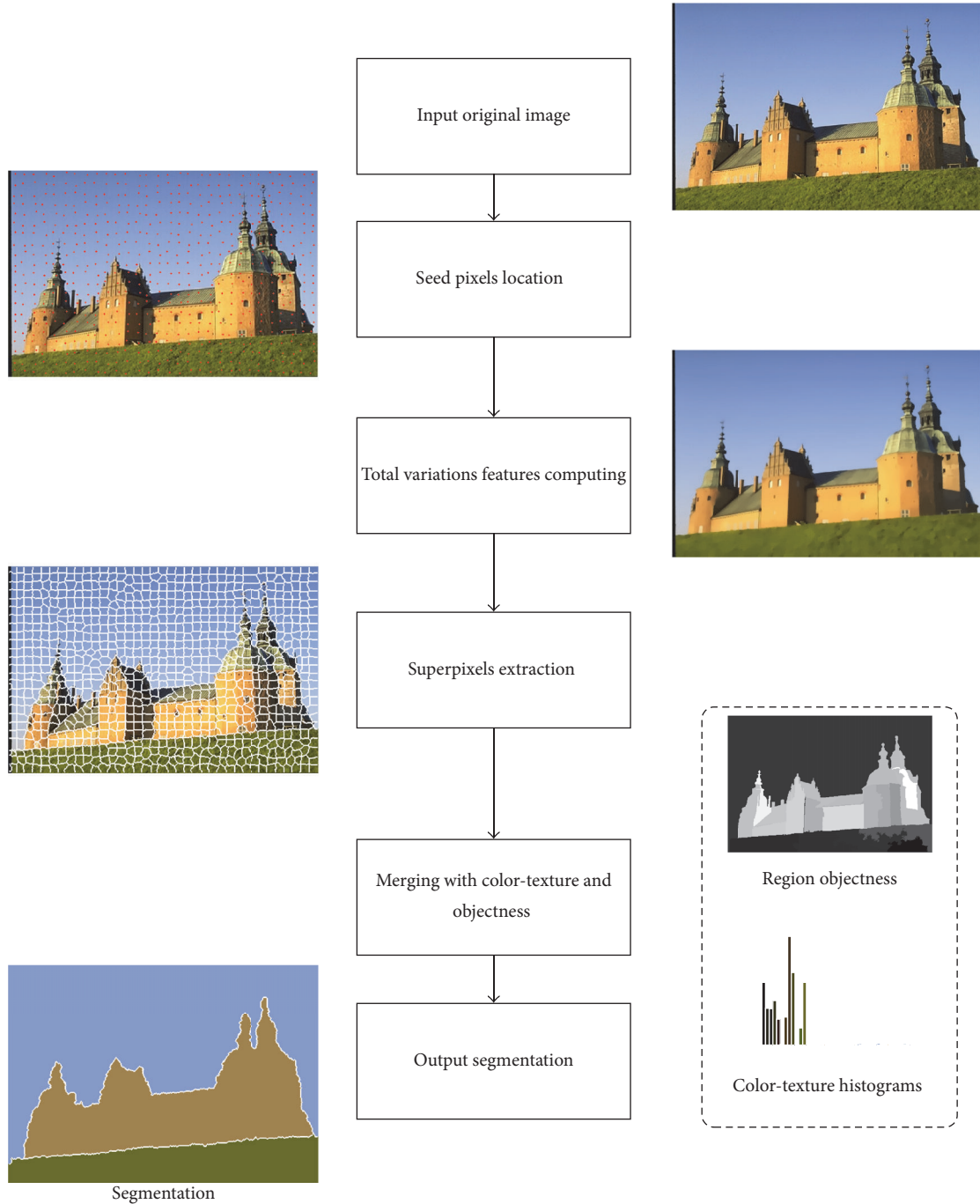


FIGURE 1: The segmentation process and intermediate results of our method.

$$g(|\nabla I_k|) = e^{-|\nabla I_k|/2\lambda^2}. \tag{2}$$

The iteration times rely on the superpixels scale for proper smoothing of local area. The reasonable iteration times of diffusion are in proportion to the radius of superpixels ≈ 13 in this paper.

2.1.2. Superpixels Growing Scheme. In this paper, we adopt a seed growing method to obtain compact superpixels with complete homogeneous pixels. The number of superpixels is preset in SLIC [12] and Turbo [18]. Seed pixels selection

method has been developed by choosing minimum gradients in divided uniform grids [12], which means to locate seed pixels at the smoothest areas of the grids. It is an effective method to find the optimal seed pixels for superpixels, so we employ the seed pixel extraction strategy and improve it by choosing seeds in the TV diffusion image as the foundation of growing superpixels.

The smoothed image is divided into a number of blocks and located gradient minimum point as the seed pixels. The color image after filtering provides a robust feature

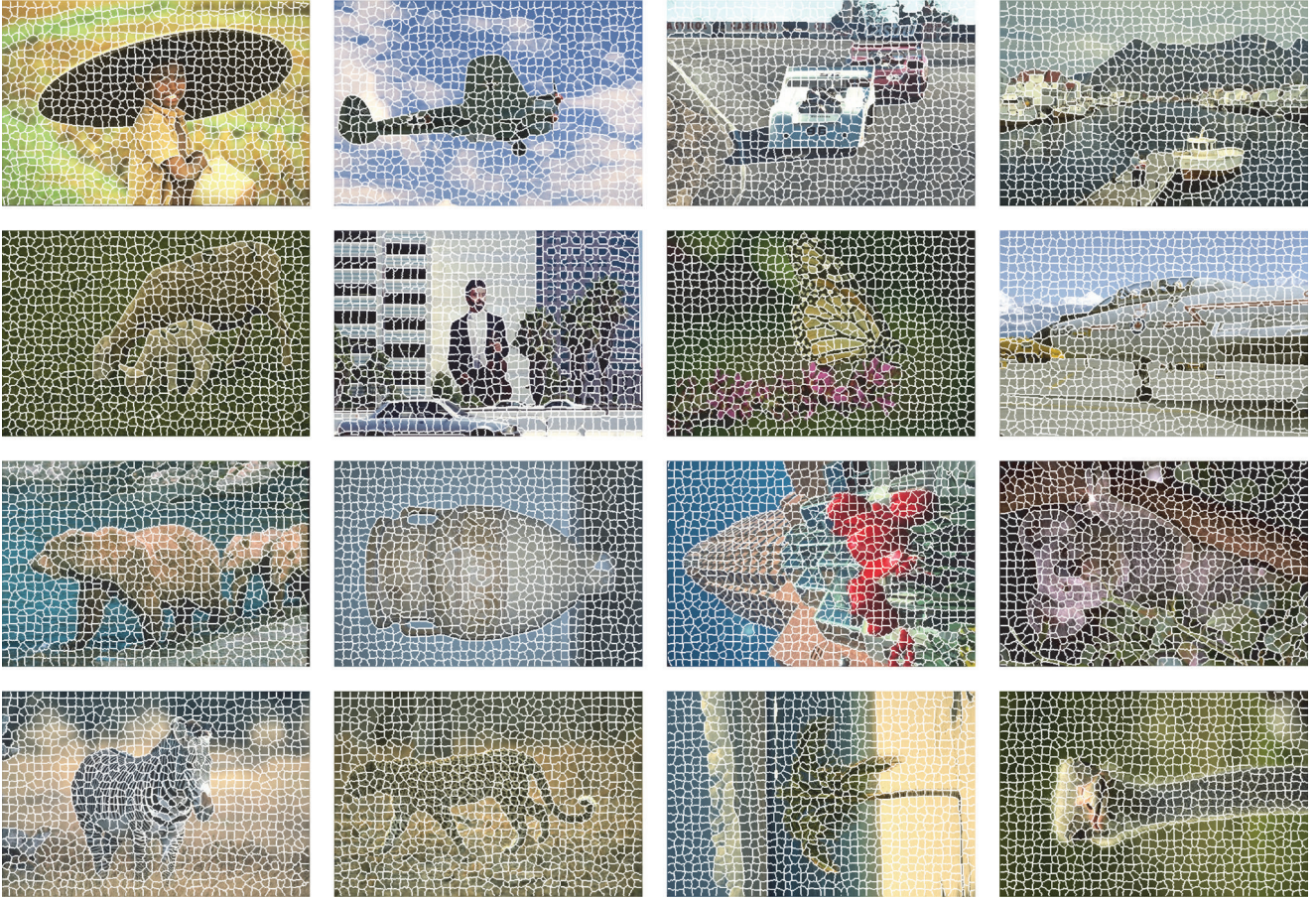


FIGURE 2: Some superpixels segmentation results of our method.

for computing homogeneous superpixels. We locate the initial seeds at the local minimum in gradient map of smoothed image in average grids at first. The grids setting method is in [12]. In the growing process, both original image and filtered image are integrated into a combined feature $F[R, G, B, R_F, G_F, B_F]$, and the similarity measurement between pixels X and Y , $\text{Sim}(X, Y)$ is calculated by Euclidean distance. The growing procedure is limited to local regions $(1.2r)$ for uniform superpixels, and r is the average radius of grids. Let s_1, s_2, \dots, s_i denote the initial seeds of the superpixels and A_i denote the growing area corresponding to s_i . The growing algorithm is thus outlined as follows:

- (1) Initialize the selected seed pixels as areas A_i , and assign a label p_i for each area A_i .
- (2) Compute average F_N of neighbor regions $(1.2r)$ F_N and F_i of A_i ; check the neighbor pixels of A_i , and if $\text{Sim}(N[i], F_i) < \text{Sim}(N[i], F_N)$, merge neighbor pixels $N[i]$ to A_i , until no pixels can be labeled.
- (3) Connect all scatter regions <20 pixels to adjacent and similar regions.

We have shown some superpixels of test images in Figure 2. Since precise boundary is necessary for successful work, we adopt a standard measure of boundary recall. The average performance of boundaries precision with regard

to superpixels numbers on BSD500 is shown in Figure 3; the smallest distance between ground truth and superpixels boundaries is 2 pixels in the experiment. We can see that more superpixels are conducive to more accurate boundaries, and 900 superpixels are big enough for merging segmentation in the next stage.

2.2. Combined Features and Superpixels Merging. Reliable region merging technique is critical for hierarchical segmentation. In this section, the similarity measurement of regions and corresponding stopping criteria are proposed based on the color-texture features and objectness of image. The merging process begins from the primitive superpixels of image until termination criterion is achieved, and the segmentation is finished.

To improve the segmentation accuracy and dependability, we define combined feature $\text{CF}(\text{His}, \text{Obj})$ to describe the oversegmentation regions. This feature integrates both color-texture information and objectness values. His is a kind of histograms containing color information and local color distribution feature of pixels. As high-level visual information, Obj is the objectness value or probability of a region belonging to an identifiable object.

2.2.1. Color-Texture Histograms. The number of colors that can be discriminated by eyes is limited, no more than 60

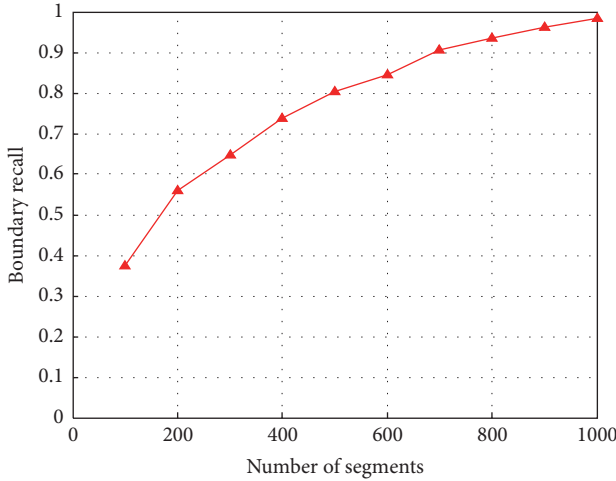


FIGURE 3: Boundary precision with regard to superpixel density.

kinds. Therefore, a quantized image is introduced to compute the color distribution of superpixels for segmentation and an image is quantized to 64 colors by the minimum variance method that leads to least distortion as well as reduces the computing time [19]. After quantization, the connection property of pixels in superpixels of same colors is computed as the texture features of superpixels. We assume Con as the connection relation between pixel C_p and its neighbor pixel $Neigh(j)$, and eight-neighbor pixels $Neigh(j = 0-7)$ are coded by $(2^{0-7} * Con_j)$ as local color pattern (LCP). If color difference C_d between $Neigh(j)$ and C_p is greater than C_{dth} (standard deviation of all pixels in quantized image), then $Con_j = 0$, otherwise $Con_j = 1$. After coding all pixels, sum up all LCP values of pixels with same color in the superpixel as color distribution weights for corresponding color bins. Thus, the composite color-texture histograms feature (CF) is computed by multiplying the probability of quantitative colors and pixel distribution in the superpixels. The bins of color-texture histograms are defined as follows:

$$\text{Bin}(i) = C_{\text{Bin}} \times C_{\text{LCP}}, \quad i = 1, 2, 3, \dots, 64. \quad (3)$$

2.2.2. Objectness Based on Sparse Sampling of Regions. As mentioned in [10], the objectness is an attribute of pixels computed with multicues for a set of sample slide windows, and the objectness map is biased towards rectangle intensity response. It seems that objectness has no ability to discriminate regions, but it provides a potential and useful value of all pixels in the rectangle. Jiang et al. [11] proposed a novel method to assign objectness value to every pixel based on the definition of objectness.

Inspired by the objectness computing method proposed by Jiang et al. [11], we propose an improved objectness computing method based on sparse sampling of slide windows for all pixels. We also employ four cues for estimation of objectness characteristics of all pixels in an image: the multiscale saliency, color contrast, edge density, and superpixels straddling. The multiscale saliency (MS cues) in the original paper [10] is calculated based on the residual spectrum of

image. This method highlights fewer regions of an image, and it is not suitable for segmentation. To expand the global significance of objectness, we use histogram-based contrast (HC) [20] salient map as the MS cues. The histogram-based contrast (HC) method is to define the saliency values for image pixels using color statistics of the input image. In the HC model, a pixel is defined by its color contrast to all other pixels in the image and sped up by sparse histograms. The saliency [20] value of pixel $I_{x,y}$ in image I is defined as

$$S(I_{(x,y)}) = \sum_{j=1}^n f_j D(c_{(x,y)}, c_j), \quad (4)$$

where $c_{(x,y)}$ is the color value of pixel $I_{(x,y)}$, n is the number of pixel colors after histogram quantization, and f_j is the statistic value of pixel color c_j that appears in image I . In order to reduce the noise caused by the randomness of histogram quantization, the saliency results are smoothed by the average saliency of the nearest $n/4$ neighbor colors.

The color contrast (CC) is a measure of the dissimilarity of a window to its immediate nearby surrounding area and the surroundings; edge density (ED) measures the density of edges near the window borders; superpixels straddling (SS) is a cue to estimate whether a window covers an object. These three cues are specifically described in [10]. With the above-mentioned four cues, the objectness score of test window w is defined by the Naive Bayes posterior probability in [10]

$$P_w(\text{obj} | \mathcal{E}) = \frac{p(\mathcal{E} | \text{obj}) p(\text{obj})}{p(\mathcal{E})}, \quad (5)$$

where \mathcal{E} is the combined cues set and $p(\text{obj})$ is the priors estimation value of the objectness. All the specific definitions of $p(\mathcal{E} | \text{obj})$, $p(\text{obj})$ and $p(\mathcal{E})$ are described in [10].

In the objectness computing methods [10, 11], the sampling windows are selected randomly by scales for calculation of objectness and 100000 windows are sampled for training. Then, they compute the other cues in \mathcal{E} with these windows.

In this paper, we try to sample less windows that most likely contain objects. Thus, presegment results can provide regions for locating object windows with higher probability. We calculate all the circumscribed rectangular windows of the presegmentation regions resulting from Mean-Shift ($h_s = 10$, $h_r = 10$, and $\text{minarea} = 100$) and put all these vertices of circumscribed rectangular windows as the candidates vertex of slide windows. Next, we select all the upper left vertices and lower right corner vertices to compose two sets, respectively: left-up (LU) and right-down (RD).

In order to find the most likely sampling windows including real objects or homogeneous semantic regions, we choose arbitrary vertex V_{LU} from LU to combine with all vertices in RD whose coordinate values are less than the coordinates of V_{LU} . These coupled vertices can provide more reasonable coordinates of sampling windows that contain objects. All the sample windows can be seen in Figure 4. We calculate all the objectness value of sampling windows by (5)

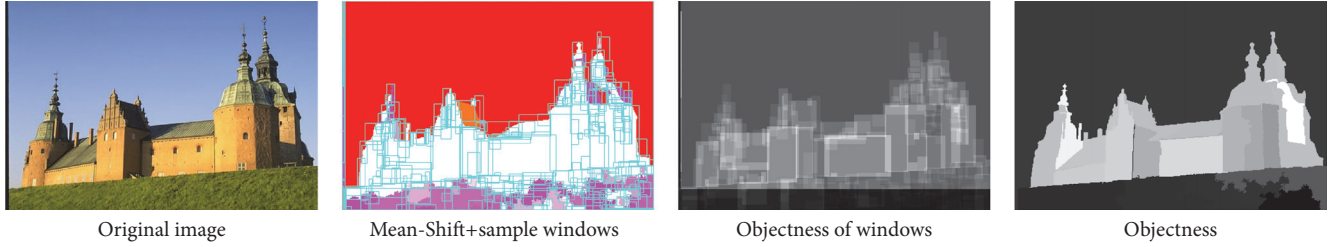


FIGURE 4: Objectness computing process.

and sum them by pixels locations as pixel-wise objectness with

$$O_p(x, y) = \sum_{\substack{w \in W \\ p(x, y) \in w}} P(w(x, y)), \quad (6)$$

where W is the sample windows set, w is the single window in W , and p is the score of sample window computed by (5). The pixel-wise objectness cannot represent the similarity of regions, so we compute the average objectness values of Mean-Shift segment regions as new objectness value of all pixels, and the objectness of superpixel is the average value of all pixels in it.

$$\text{Obj}_p(x, y) = \text{AVG}_{p \in R_i} O_p(x, y). \quad (7)$$

2.2.3. Merging Rules. The regions merging method and stop criteria are crucial for the segmentation results. We established adjacent table and similar matrix for all superpixels to carry out the merging procedure, and the metric of similarity between two regions can be calculated by the following formula:

$$\text{Dis}(X, Y) = \tau * (\chi(\text{His}(X), \text{His}(Y))) + (1 - \tau) * |\text{Obj}(X - Y)|. \quad (8)$$

The fixed weight assignment is not well adapted to obtain the real object regions by merging process, so we define alterable parameter τ using two types of features to regulate the similarity computing: (1) fixed weight assignment is not well adapted to obtain the real object regions by merging process; (2) when the feature change is greater, the weight should be set smaller in the similarity computing, otherwise it will lead to overmerging. Therefore, we define a balance parameter using two types of features to regulate the similarity computing:

$$\tau = \frac{\Delta(\text{Obj})}{R(\text{Obj})} \times \frac{\Delta(\text{His})}{2\mu + \sigma(\text{His}(\text{Superpixels}))}, \quad (9)$$

where Δ is the difference of given variables, R is the range of Obj , μ is the mean value of His features of superpixels, and σ is the standard deviation of His , Obj of current merged superpixels. Here, we set the threshold based on V to control the merging process. If the similarity between A and B is greater than V , the adjacent merges two regions A and B , updates adjacent relations and labels, and checks next regions

until no pair of regions satisfies the merging criterion. V is the combined standard deviation of all combined features (His , Obj) of current merged superpixels, and it is defined in

$$V = \tau * \sigma(\text{His}(\text{Superpixels})) + (1 - \tau) * \sigma(\text{Obj}(\text{Superpixels})). \quad (10)$$

The description of the merging strategy is displayed in Algorithm 1. Each time after merging, the combined feature $\text{CF}(\text{His}, \text{Obj})$ of the new region is updated as follows: firstly, the connection relations of all pixels are unchanged in the new region, then compute the statistics value for each bin of quantitative colors, and update of the histogram feature using formula (3) of the new region. Second, Obj feature of the new region is obtained by computing average objectness value of all the pixels in the new region.

3. Experimental Results

3.1. Dataset. The proposed segmentation model has been tested on Berkeley Segmentation Datasets [21, 22] and compared with human ground truth segmentation results. Specifically, the BSD dataset images were selected from the Corel Image Database in size of 481×321 . The Berkeley Dataset and Benchmark is widely used by the computer vision for two reasons: (1) it provides a large number of color images with complex scenes and (2), for each image, there is multiple ground truth segmentation which are provided for the numerical evaluation of the testing segmentation algorithms.

3.2. Visual Analysis of Segmentation Results. For a subjective visual comparison, we selected 5 segmentation results of test images and compared them with three classic algorithms (JSEG [23], CTM [24], and SAS [25]). The superpixels number of our method for comparing is set to be $K = 900$; the JSEG [23] algorithm needs three predefined parameters: quantized colors = 10, number of scales = 5, and merging parameter = 0.78; the results for CTM [24] is ($\lambda = 0.1$); the SAS [25] method is implemented with default parameters. From the results shown in Figure 4, it is clear that the proposed method has performed better in detecting complete objects from images (such as flowers, zebra, cheetah, person, and house in the test images). For all the five test images, our method has shown clear and complete contours of the primary targets. The SAS method is more accurate in homogeneous region detection than other methods, but it

Input:superpixels $G(L)$, adjacent table $\text{Sim}(G)$ of $G(L)$ **Output:**

The merged result.

- (1) For each N node in G , $N(i)$ is neighbor region of N
- (2) **if** $\text{Dis}(N, N(i)) < V$, **then**
- (3) merged $N(i)$ to N , $L = L - 1$,
- (4) change the region label of $N(i)$ to N ,
- (5) modified the histogram feature of two merged regions
- (6) update $G(L)$, $\text{Sim}(G)$ and V ;
- (7) **end if**
- (8) If no regions satisfied $\text{Dis}(N, N(i)) < V$, end the merge process, otherwise, return to (1);
- (9) Merge meaningless and small areas to adjacent regions
- (10) Return relabeled image.

ALGORITHM 1

perform less oversegmentation except in complex areas. CTM performs better in texture regions but lost in the completeness of the objects. JSEG is also a texture-oriented algorithm which has the same weakness as CTM. Thus, three other segmentation results have brought more over-segments and meaningless regions from the test results in Figure 5. The region number versus iteration times in the merging process of some test images is shown in Figure 6, and the region number versus parameter V in the merging process of some test images is shown in Figure 7.

In Figure 8, we have displayed fourteen segmentation results from test images. These results include different type of images, such as animals, buildings, person, vehicles, and planes with various texture features. The segmentation results show that the proposed growing-merging framework can segment the images into compact regions and restrain the over- or undersegmentation.

3.3. Quantitative Evaluation on BSD300. The comparison is based on the four quantitative performance measures: probabilistic rand index (PRI) [26], variation of information (VoI) [27], global consistency error (GCE) [21] and Boundary Displacement Error (BDE) [28]. The PRI Index compares the segmented result against a set of ground truth by evaluating the relationships between pairs of pixels as a function of variability in the ground truth set. VoI defines the distance between two types of segmentation as the average conditional entropy of one type of segmentation given the other, and GCE is a region-based evaluation method by computing consistency of segmented regions. BDE measures the average displacement error of boundary pixels between two boundaries images. Many researchers have evaluated these indexes to compare the performance of various methods in image segmentation. The test values of five example images on four algorithms are shown in Table 1, and the average performance on BSD300 [21] are shown in Table 2. With the above comparison strategy, our algorithm provided more approximate boundaries with manual expert segmentation results. The bins number of the combined histograms is critical parameter for segmentation, so we have take some test examples by varying bins in Figure 9 to verify 64 bins is more proper for accurate segmentation.

TABLE 1: The PRI, VoI, GCE, and BDE value of four test images.

	Index	Proposed	SAS	CTM	JSEG
12084	PRI	0.7848	0.76155	0.7051	0.7026
	VoI	2.4184	2.1261	3.6425	4.3114
	GCE	0.2256	0.1753	0.1658	0.1209
	BDE	7.7028	6.8194	7.4629	9.2061
130066	PRI	0.8209	0.7286	0.7710	0.7457
	VoI	1.4075	2.2986	2.6194	3.6813
	GCE	0.1416	0.3001	0.1117	0.1226
134052	BDE	9.4618	18.6371	10.2823	13.3104
	PRI	0.7630	0.6454	0.6303	0.5833
	VoI	1.2913	2.0665	2.6110	3.9948
15062	GCE	0.1020	0.2263	0.0833	0.1302
	BDE	6.1061	11.2896	9.4271	15.1976
	PRI	0.8882	0.8973	0.8673	0.8705
385028	VoI	1.5428	1.4820	2.1843	2.6231
	GCE	0.2303	0.1959	0.1386	0.1075
	BDE	8.7102	21.2930	12.2526	13.9926
	PRI	0.9152	9.1108	0.9031	0.9036
	VoI	1.5868	3.6813	1.8808	2.2501
	GCE	0.1436	0.2236	0.0888	0.1072
	BDE	9.1108	9.1016	8.6668	8.6987

TABLE 2: The average values of PRI, VoI, GCE, and BDE for BSD300.

	SAS	JSEG	CTM	Proposed
PRI	0.8246	0.7754	0.7263	0.8377
VOI	1.7724	2.3217	2.1010	1.773
GCE	0.1985	0.2017	0.2033	0.1964
BDE	12.6351	14.4072	11.9245	10.8311

3.4. Boundary Precision Comparison on BSD500. Estrada and Jepson [29] have proposed a thorough-quantitative-evaluation matching strategy to evaluate the segment accuracy for boundary extraction. The algorithms are evaluated using an efficient algorithm for computing precision and recall with regard to human ground truth boundaries. The



FIGURE 5: Comparison of four segmentation methods (proposed, SAS, CTM, and JSEG).

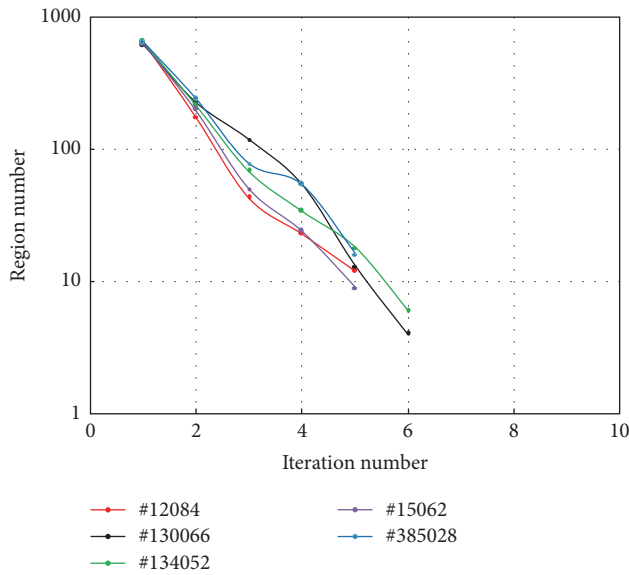


FIGURE 6: Merging curves of iteration-region numbers.

evaluation strategy of precision and recall is reasonable and equitable as measures of segmentation quality because

of not being biased in favor of over- or undersegmentation. An excellent segmentation algorithm should bring low segmented error and high boundary recall. It defines precision and recall to be proportional to the total number of unmatched pixels between two types of segmentation S_{source} and S_{target} , where S_{source} is the boundaries extracted from segmentation results by computer algorithms and S_{target} is the boundaries of human segmentation provided by BSD500 [22]. A boundary pixel is identified as true position when the smallest distance between the extracted and ground truth is less than a threshold ($\epsilon = 5$ pixels in this paper). The precision, recall, and F -measures are defined as follows:

$$\text{Precision} \{S_{source}, S_{target}\} = \frac{\text{Matched}(S_{source}, S_{target})}{|S_{source}|} \quad (11)$$

$$\text{Precision} \{S_{source}, S_{target}\} = \frac{\text{Matched}(S_{source}, S_{target})}{|S_{target}|}$$

The precision and recall measures provide a more reliable evaluation strategy based on two reasons: (1) it refined boundaries and avoided duplicate comparison of identical boundary pixels, and (2) it provides dynamic displacement parameters for comprehensive and flexible evaluation of segmentation performance. In this section, we displayed the

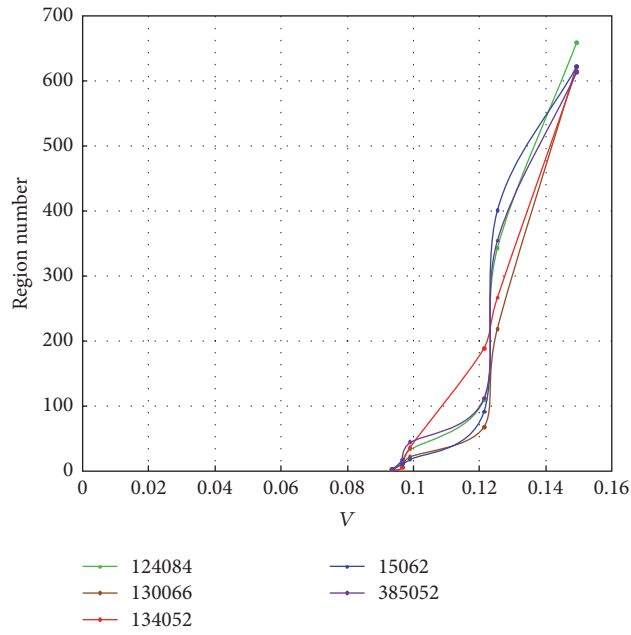


FIGURE 7: Merging curve of region number- V of test images in Figure 3.



FIGURE 8: Some segmentation results and objectness map.

boundaries precision and recall curves with five segmentation algorithms (MS [15], FH [2], JSEG [23], CTM [24], and SAS [25]) in Figure 10. The ROC curves fully characterize the performance in a direct comparison of the segmentations quality provided by different algorithms. Figure 10 shows that our method provides more closely boundaries of segmentations with the ground truth.

4. Conclusion

In this paper, we propose a novel hierarchical color image segmentation model based on region growing and merging. A proper growing procedure is designed to extract compact and complete superpixels with TV diffusion filtered image at first. Second, we integrate color texture and objectness information of superpixels into the merging procedure and

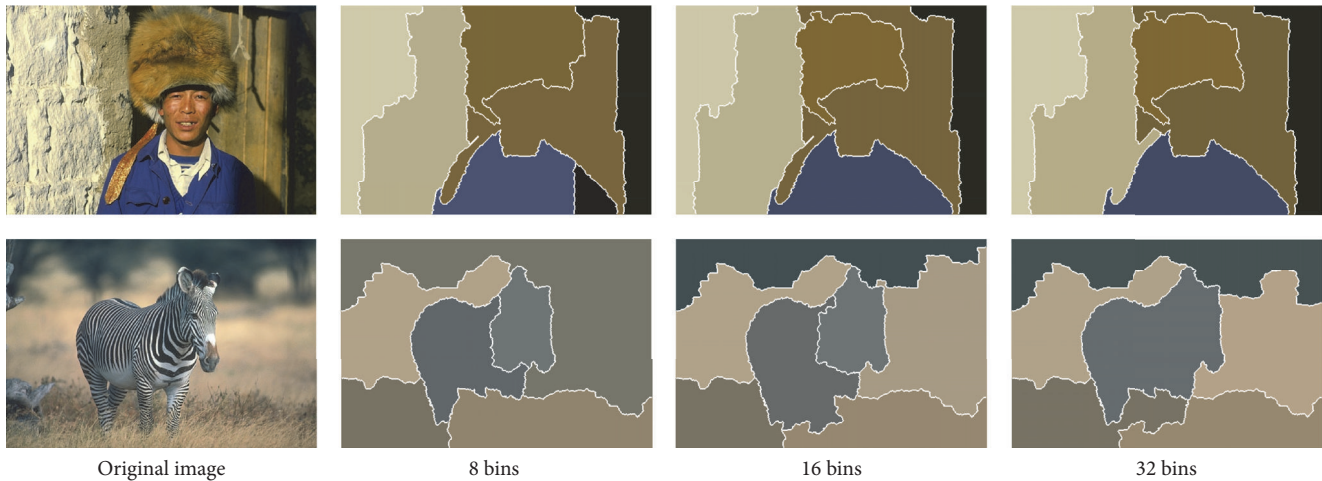


FIGURE 9: Some segmentation results of varying bins.

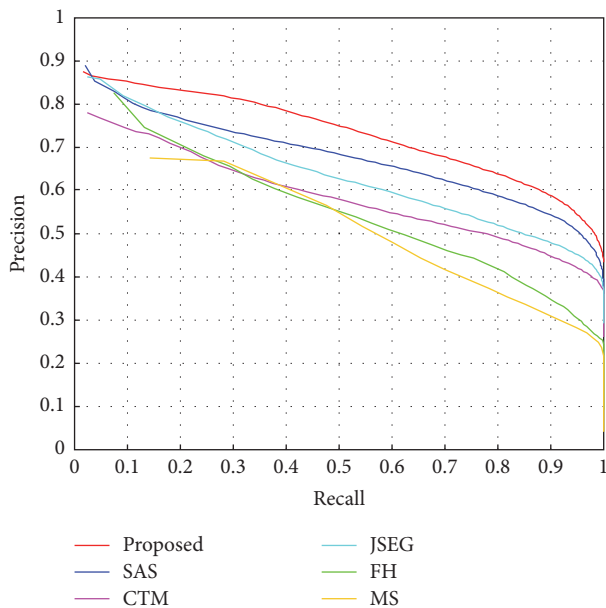


FIGURE 10: Boundaries of ROC curves on BSD500 when $\varepsilon = 5$.

develop effective merging rules for ideal segmentation. In addition, using a histogram-based color-texture distance metric, the merging strategy can obtain complete objects with smooth boundaries and locate the boundaries accurately. The subjective and objective experimental results for nature color images show a good segmentation performance and demonstrate a better discriminate ability with objectness feature for complex images of the proposed method. In the future, we will focus on the extension of objectness to different kinds of segmentation strategies.

Competing Interests

The authors declare that they have no competing interests.

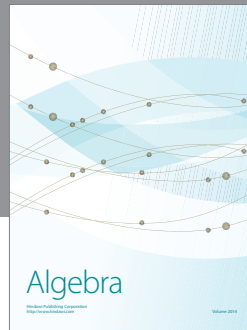
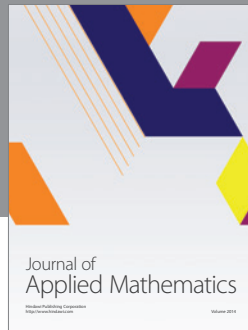
Acknowledgments

Haifeng Sima acknowledges the support of the Key Projects of National Natural Foundation, China (U1261206), National Natural Science Foundation of China (61572173 and 61602157), Science and Technology Planning Project of Henan Province, China (162102210062), the Key Scientific Research Funds of Henan Provincial Education Department for Higher School (15A520072), and Doctoral Foundation of Henan Polytechnic University (B2016-37).

References

- [1] J. Fan, D. K. Y. Yau, A. K. Elmagarmid, and W. G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 1454–1466, 2001.
- [2] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [3] M. Mignotte, "Segmentation by fusion of histogram-based K-means clusters indifferent colour spaces," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 780–787, 2008.
- [4] C. Li, C. Xu, C. Gui, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3243–3254, 2010.
- [5] S. Gould and X. He, "Scene understanding by labeling pixels," *Communications of the ACM*, vol. 57, no. 11, pp. 68–77, 2014.
- [6] T. Athanasiadis, P. Mylonas, Y. Avrithis, and S. Kollias, "Semantic image segmentation and object labeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 3, pp. 298–311, 2007.
- [7] S. Vicente, C. Rother, and V. Kolmogorov, "Object cosegmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 2217–2224, IEEE, Providence, RI, USA, June 2011.
- [8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 27th IEEE Conference on*

- Computer Vision and Pattern Recognition (CVPR '14)*, pp. 580–587, Columbus, Ohio, USA, June 2014.
- [9] W. Tao, Y. Zhou, L. Liu, K. Li, K. Sun, and Z. Zhang, “Spatial adjacent bag of features with multiple superpixels for object segmentation and classification,” *Information Sciences*, vol. 281, pp. 373–385, 2014.
- [10] B. Alexe, T. Deselaers, and V. Ferrari, “Measuring the objectness of image windows,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.
- [11] P. Jiang, H. Ling, J. Yu, and J. Peng, “Salient region detection by UFO: uniqueness, focusness and objectness,” in *Proceedings of the 14th IEEE International Conference on Computer Vision (ICCV '13)*, pp. 1976–1983, Sydney, Australia, December 2013.
- [12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “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.
- [13] 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.
- [14] L. Vincent and P. Soille, “Watersheds in digital spaces: an efficient algorithm based on immersion simulations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.
- [15] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [16] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [17] M. Rousson, T. Brox, and R. Deriche, “Active unsupervised texture segmentation on a diffusion based feature space,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 699–704, Madison, Wis, USA, June 2003.
- [18] A. Levinstein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, “TurboPixels: fast superpixels using geometric flows,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, 2009.
- [19] J.-P. Braquelaire and L. Brun, “Comparison and optimization of methods of color image quantization,” *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 1048–1052, 1997.
- [20] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, “Global contrast based salient region detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 409–416, June 2011.
- [21] 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 *Proceedings of the 8th IEEE International Conference on Computer Vision*, pp. 416–423, Vancouver, Canada, July 2001.
- [22] <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>.
- [23] Y. Deng and B. S. Manjunath, “Unsupervised segmentation of color-texture regions in images and video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, pp. 800–810, 2001.
- [24] 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.
- [25] Z. Li, X.-M. Wu, and S.-F. Chang, “Segmentation using superpixels: a bipartite graph partitioning approach,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 789–796, Providence, RI, USA, June 2012.
- [26] C. Pantofaru and M. Hebert, “A comparison of image segmentation algorithms,” Tech. Rep. CMU-RI-TR-05-40, Carnegie Mellon University, 2005.
- [27] M. Meila, “Comparing clusterings: an axiomatic view,” in *Proceedings of the 22nd ACM International Conference on Machine Learning (ICML '05)*, pp. 577–584, New York, NY, USA, 2005.
- [28] J. Freixenet, X. Munoz, D. Raba, J. Marti, and X. Cuff, “Yet another survey on image segmentation,” in *Computer Vision—ECCV 2002: 7th European Conference on Computer Vision Copenhagen, Denmark, May 28–31, 2002 Proceedings, Part III*, vol. 2352 of *Lecture Notes in Computer Science*, pp. 408–422, Springer, Berlin, Germany, 2002.
- [29] F. J. Estrada and A. D. Jepson, “Benchmarking image segmentation algorithms,” *International Journal of Computer Vision*, vol. 85, no. 2, pp. 167–181, 2009.




Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

