New benchmark for image segmentation evaluation

Feng Ge*
Virginia Tech
Department of Electrical and Computer Engineering
Blacksburg, Virginia 24061

Song Wang
Tiecheng Liu
University of South Carolina
Department of Computer Science and Engineering
Columbia, South Carolina 29208
E-mail: songwang@engr.sc.edu

Abstract. *Image segmentation and its performance evaluation are very difficult but important problems in computer vision. A major challenge in segmentation evaluation comes from the fundamental conflict between generality and objectivity: For general-purpose segmentation, the ground truth and segmentation accuracy may not be well defined, while embedding the evaluation in a specific application, the evaluation results may not be extensible to other applications. We present a new benchmark to evaluate five different image segmentation methods according to their capability to separate a perceptually salient structure from the background with a relatively small number of segments. This way, we not only find a large variety of images that satisfy the requirement of good generality, but also construct ground-truth segmentations to achieve good objectivity. We also present a special strategy to address two important issues underlying this benchmark: (1) most image-segmentation methods are not developed to directly extract a single salient structure; (2) many real images have multiple salient structures. We apply this benchmark to evaluate and compare the performance of several state-of-the-art image segmentation methods, including the normalized-cut method, the watershed method, the efficient graph-based method, the mean-shift method, and the ratio-cut method.* © 2007 SPIE and IS&T. [DOI: 10.1117/1.2762250]

1 Introduction

By partitioning an image into a set of disjoint segments, image segmentation leads to more compact image representation. As the central step in computer vision and image understanding, image segmentation has been extensively investigated in the past decades, with a large number of image segmentation methods developed.[1–10] Reliable segmentation performance evaluation for quantitatively positioning the state-of-the-art in image segmentation is extremely important. In many prior works, segmentation performance is usually evaluated by subjectively or objectively judging several sample images.[11–16] Such evaluations on a small number of sample images lack statistical mean-

ings and may not be generalized to other images and applications. To address this problem, it has been agreed that a benchmark, which includes a large set of test images and some objective performance measures, is necessary for segmentation evaluation.[17]

For benchmark-based image segmentation evaluation,[17,18] we usually desire two important properties: *objectivity* and *generality*. Good objectivity means that all the test images in the benchmark should have an unambiguous ground-truth segmentation so that the segmentation evaluation can be conducted objectively. Good generality means that the test images in the benchmark should have a large variety so that the evaluation results can be extended to other images and applications. Unfortunately, there exists a well-known dilemma between objectivity and generality in benchmark-based segmentation evaluation: On the one hand, by collecting a large variety of test images that are not associated to specific applications, the ground-truth segmentation of many images may not be unambiguously and uniquely defined,[17] as illustrated in Fig. 1; on the other hand, by restricting the segmentation evaluation to a certain category of images and/or to a specific application (e.g., locating faces in photos), the evaluation results may not be applicable to other applications, although the well-defined ground-truth and segmentation-performance measures are available.

The goal of this paper is to develop a new image segmentation benchmark by seeking a better balance between objectivity and generality in evaluating image segmentation. Particularly, *we specify the goal of image segmentation as extracting a single salient object in the image*. In this formulation, the ground truth is a semantic feature of the object represented by a closed boundary that can be more easily and unambiguously constructed for many natural images, as shown in Fig. 1(b). By treating the salient object as the foreground *figure*, and the remaining portion as the *background*, such a formulation is usually referred to as *figure-ground* segmentation in the prior literature. For convenience, we will continue to use this terminology in this paper. However, it must be emphasized that the "background" in our test images, as discussed in detail later, has
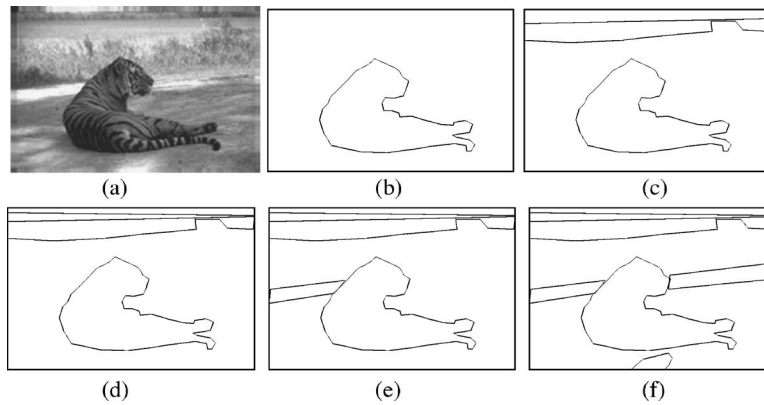
Fig. 1 The figure-ground segmentation is usually better defined than the general-purpose segmentation: (a) A sample image; (b) the unambiguous ground truth in the figure-ground segmentation; (c–f) four different ground-truth segmentations produced by different people in the general-purpose segmentation, i.e., partitioning the image into an unfixed number of segments.

a more general meaning than a trivial background region of homogenous intensity or uniform texture as assumed in prior figure-ground segmentation literatures. Actually, the background segment may contain many other objects. With this formulation, we include a large variety of test images, which guarantees the generality of the proposed benchmark.

However, we need to address two important issues in applying this benchmark to evaluate various general-purpose image segmentation methods. First, most available image segmentation methods are not specifically designed to extract a single salient object. Instead, they usually partition an image into more than two disjoint segments, as shown in Figs. 1(c)–1(f), and the segmentation accuracy is usually dependent on the number of resulting segments. In this paper, we develop a special strategy and propose a new concept of "upper-bound" performance to address this problem. With this strategy, a good image segmentation is expected to accurately separate the ground-truth salient object from the background while keeping the number of resulting segments small. Based on this strategy, all five segmentations shown in Fig. 1(b)–1(f) might be good segmentations because all of them separate the ground-truth foreground and background into different segments. Second, many real images contain multiple objects, and the salient object is not unambiguously defined. Although, in our benchmark, we collect only the images with unambiguous, most salient objects and some trivial objects that are much less salient in the background, we expect that the images with multiple equally salient objects can also be included and evaluated in this benchmark. In this paper, we address this problem based on the same special strategy for "upper-bound" performance and extend the goal of image segmentation to *separating a specified salient object from the background with a small number of segments*. While general-purpose image segmentation might be formulated in different ways in different applications, we believe that the capability to separate salient objects from the background would be a more general measure for evaluating its performance. Such a formulation of image segmentation has many applications in computer vision tasks, such as content-based image retrieval,[19] visual surveillance,[20] and

learning-based image segmentation algorithms.[21]

In the remainder of this paper, Section 2 briefly reviews the related work on image segmentation evaluation and summarizes the contribution of this paper. Section 3 introduces the benchmark construction. Section 4 briefly introduces the five image segmentation methods evaluated in this paper. Section 5 describes the performance measure we used in evaluation. Section 6 reports the evaluation results of the selected methods on our benchmark. A brief conclusion is given in Section 7.

2 Related Work and Our Contribution

A large number of literature on the image segmentation evaluation have been developed in the past few decades. Most previous works are focused on developing better ways to measure the accuracy/error of the segmentation. Some of them (e.g., Refs. 22–24) do not require ground-truth image segmentation as the reference. In these methods, the segmentation performance is usually measured by some contextual and perceptual properties, such as the homogeneity within the resulting segments and the inhomogeneity across neighboring segments. For example, in Ref. 22, the segmentation of an image sequence (video) is evaluated by checking the homogeneity of the resulting segments.

Most of the prior image segmentation evaluation methods, however, need a ground-truth segmentation of the considered image, and the performance is measured by calculating the discrepancy between the considered segmentation and the ground-truth segmentation.[11–15,25–32] Since the construction of the ground-truth segmentation for many real images is labor-intensive and sometimes not well or uniquely defined, most prior image segmentation methods are only tested on (1) some special classes of images used in special applications where the ground-truth segmentations are uniquely defined, (2) synthetic images where ground-truth segmentation is also well defined, and/or (3) a small set of real images.

For examples, in Ref. 25, a performance measure is developed to evaluate the medical image segmentation, where the test images are synthesized according to a medical imaging model. In Ref. 28, the segmentation of some special

medical images are evaluated with ground-truth segmentations extracted by multiple expert observers. The test data are 44 cardiac images and 30 skull images. The main goal of Ref. 28 is to investigate whether an automatic segmentation agrees with the observers' segmentation and whether the different observers' segmentations agree with each other. Goumeidane *et al.*[26] suggest a performance measure based on two distortion rates of the resulting segments to treat both underdetected and overdetected pixels. Experiments are conducted only on several simple binary synthetic images. Cavallaro *et al.*[27] develop a performance measure that combines both objective and perceptual errors and use it to evaluate the segmentation of a sequence of images with manually labeled segmentation. Freixenet *et al.*[30] propose a performance measure that combines boundary and region information and test several image segmentation algorithms on some synthetic data and several special classes of images in the USC-SIPI database. Everingham *et al.*[11] suggest to evaluate segmentation from different perspectives but avoid combining them into a single performance measure. In Ref. 11, six general-purpose segmentation algorithms are evaluated on 100 samples images of urban and rural outdoor scenes. Van Droogenbroeck and Barnich[31] propose a statistical measure to evaluate the performances of image segmentation against the ground-truth segmentation, without any experimental study. Motivated by the concept of phase-modulated signals, Paglieroni[32] develops a new performance measure for evaluating image segmentation against the ground truth. The experiment is conducted on one satellite image. Cardoso and Corte-Real[29] recently developed another measure to evaluate image segmentation results against a single ground-truth segmentation by combining perceptual and contextual information. The experiments are conducted in several sample images. Pal and Pal[33] and Zhang[15,34] provide surveys of some early image segmentation evaluation methods.

Different from these methods, this paper presents a benchmark for evaluating general-purpose image segmentation methods on a large variety of real images with well-defined objects as the ground truth. It is a continued study of our previous work[35] with a more thorough investigation of performance variation resulting from different parameter settings in 5 well-known, region-based segmentation methods. The work most related to ours is the Berkeley image segmentation benchmark.[17] The Berkeley benchmark contains more than 1000 various natural images. Since the ground-truth segmentation may not be well and uniquely defined, each test image in the Berkeley benchmark is manually segmented by a group of people. Given non-unique ground truths, this benchmark develops a global consistency error (GCE) and a local consistency error (LCE) for measuring the segmentation accuracy. These two measures tolerate unreasonable refinement of the ground truth, i.e., if the segmentation is a refined version of the ground truth, or vice versa, the segmentation error is zero. Therefore, trivial segmentations, where each segment only contains one pixel or the whole image is a single segment, always produce "perfect" 100% segmentation accuracy in this benchmark.

Different from the Berkeley benchmark, this paper constructs a single ground truth of the object representation for each test image by extracting a salient object from this image. We then *only* collect images with some identifiable salient object into the benchmark. The contributions of this paper can be summarized as

1. By formulating the goal of image segmentation as extracting a salient object from the image, a large variety of test images can be easily collected, and the manual construction for ground-truth segmentation can be easily performed. In this stage, we have collected 1023 test images and constructed their ground truths.
2. With a single defined ground truth that consists of only two segments, the segmentation performance measure can be more robustly defined and used. In this paper, we simply use the Jaccard coefficient[43] as the performance accuracy measure. In fact, many new measures developed in previous literature, as discussed above, may also be adapted and used in the proposed benchmark.
3. While image segmentation performance is highly dependent on the number of produced segments, we introduce a concept of "upper-bound" performance in this benchmark to better describe and address this problem. Furthermore, this new concept allows the inclusion of test images with multiple salient objects.
4. We apply the developed benchmark to evaluate five state-of-the-art image segmentation methods and obtain several insightful observations.

3 Test-Image Database Construction

As the first stage of the benchmark construction, we collected 1023 real natural images from Internet, digital photos, and some well-known image databases such as Corel. We carefully examined each image before including it into the database. A particular requirement is that each image contains a salient foreground object that is unambiguous in human visual perception. Object saliency, which is characterized by closure, proximity, and continuity, plays a very important role in perception and recognition of objects.[36] The ground-truth segmentation can be easily constructed by manually extracting the closed boundary of this salient object. To make this benchmark suitable for evaluating a large variety of image segmentation methods, color information is removed and all the images are unified to 256-bit grayscale images in PGM format, with a size in the range of 80×80 to 200×200.

We hired two computer science undergraduate students to build this test-image database. They use the following strategy to decide whether to include an image into the database. First, both of them look at the considered image and select the most salient object independently. Second, if both of them select the same object without any reservation, this image will be included in the database. Otherwise, if they choose different objects or either has reservations in determining the most salient object, this image will not be included. After one image is decided to be included into the database, they work together to construct a single ground-truth segmentation by extracting the closed boundary of the identified salient object.
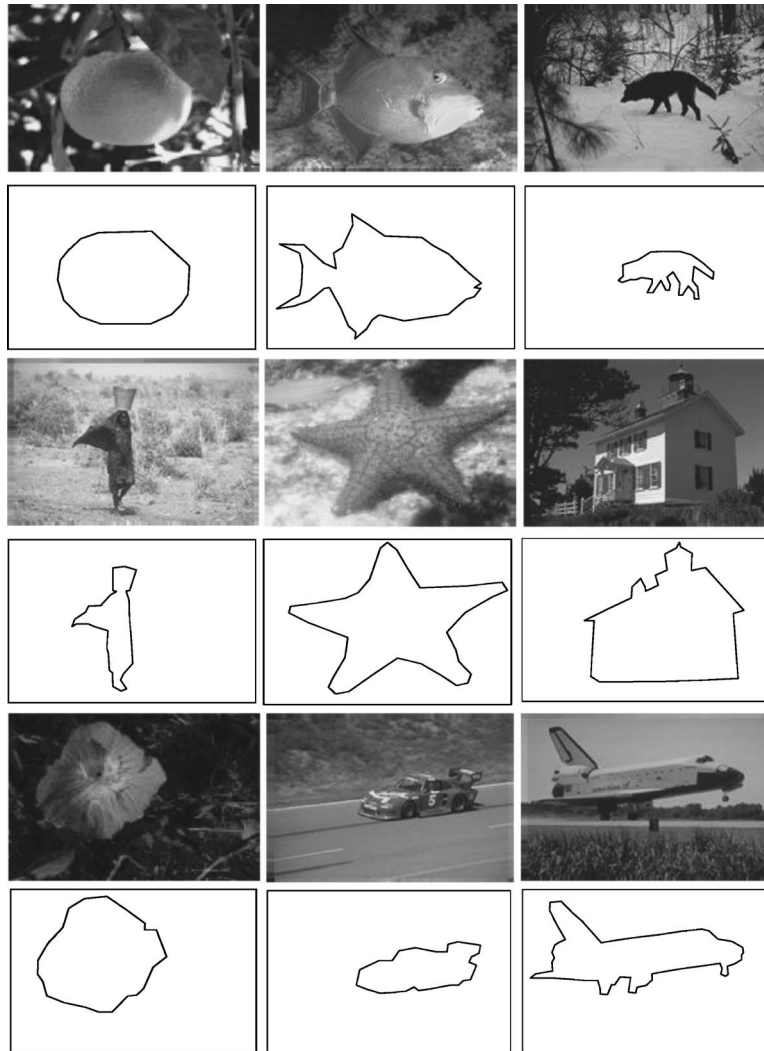
Fig. 2 Nine sample images in our image database and the ground truth produced manually.

Figure 2 demonstrates several sample images and their ground truths in the current image database. Note that we intentionally collect images with various foreground objects (such as human, animal, vehicle, building, etc.) and various backgrounds. Also note that, in the collected images, the salient object may not be the only object in the image, and the background may contain some objects that are not as perceptually salient as the foreground one. Certainly, the decision made by these two students may not always be psychophysically consistent with other people, i.e., some collected images, when presented to other viewers, may still result in a different foreground object. The only requirement is to pick one salient object and label it as the ground-truth foreground. We believe the ground truths constructed by these two students satisfy well this requirement.

4 Selected Image Segmentation Methods

Based on the above benchmark, we evaluate the following five image segmentation methods:

- Normalized-cut method (NC)[7] implemented by Shi and Malik.[37]

- Efficient graph-based method (EG)[5] implemented by Felzenszwalb and Huttenlocher.[38]
- Mean-shift method (MS)[2] implemented by Comaniciu and Meer.[39]
- Ratio-cut method (RC)[10] implemented by Wang and Siskind.[40]
- Watershed method (WS)[8] (Matlab implementation)

Sample image segmentations resulting from these methods are shown in Fig. 3. We chose these five methods based on three considerations: (1) They well represent different categories of image segmentation methods well; (2) all of them are relatively new methods and/or implementations that represent the current state-of-the-art of general-purpose image segmentation well; (3) the software implementations of these five methods are publicly available. In the following, we briefly overview these five methods.

4.1 Normalized-Cut Method (NC)

In NC, an image is modeled by a graph $G = (V, E)$, where V is a set of vertices corresponding to image pixels and E is a set of edges connecting neighboring pixels. The edge
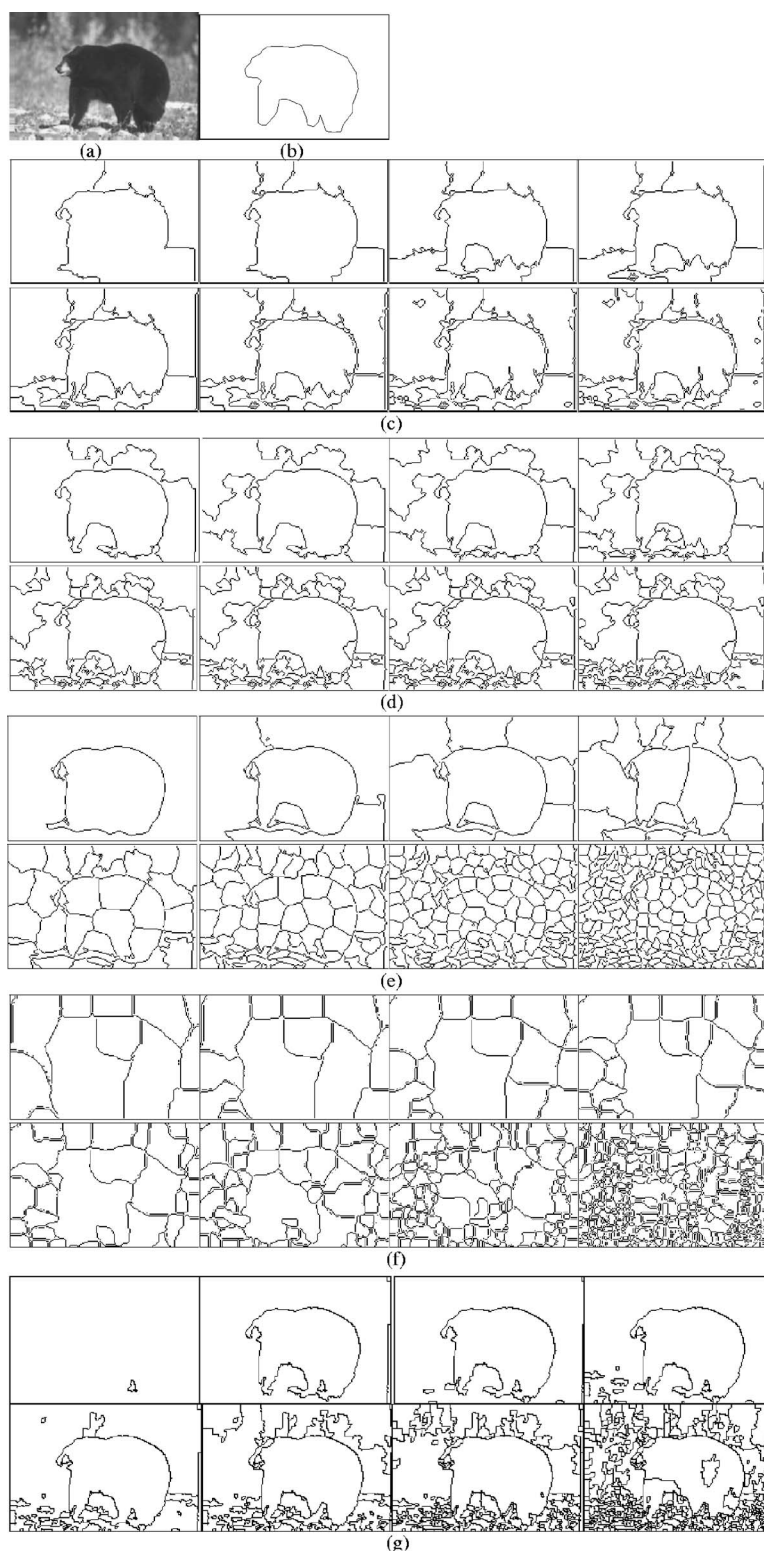
Fig. 3 Sample image segmentation results using the five selected methods at different parameter settings. (a) Original image; (b) ground truth; (c) EG segmentation. The eight results (from left to right, from top to bottom) are obtained by setting parameter S to 20%, 10%, 4%, 2%, 1%, 0.5%, 0.25%, and 0.125%, respectively. The segmentation parameters are explained in Section 6.2. (d) MS segmentation. Parameter S is the same as the one in (c). (e) NC segmentation. Parameter k is set to 2, 5, 10, 20, 40, 80, 160, and 320, respectively. (f) Watershed segmentation. The varying parameter is the Gaussian smoothing filter standard deviation 40, 35, 30, 25, 20, 15, 10, 5. (g) RC segmentation. The number of regions is set to 1, 2, 3, 4, 5, 6, 7, and 8, respectively. These parameter settings are obtained from the experimental study and will be discussed in Section 6.

weight $w(u,v)$ describes the affinity between two vertices u and v based on their intensity similarity and spatial proximity. Using this graph model, segmenting an image into two segments corresponds to a graph cut (A,B), where A and B are the vertices in two resulting subgraphs. In NC, the segmentation cost is defined by

$$\text{Ncut}(A,B) = \frac{\text{cut}(A,B)}{\text{assoc}(A,V)} + \frac{\text{cut}(A,B)}{\text{assoc}(B,V)}, \quad (1)$$

where $cut(A,B) = \Sigma_{u \in A, v \in B} w(u,v)$ is the cut cost of (A,B) and $assoc(A,V) = \Sigma_{u \in A, v \in V} w(u,v)$ is the association between A and V. NC segments the image by finding the cut (A,B) with the minimum cost in Eq. (1). Since this is a NP-complete problem, a spectral graph algorithm was developed to find an approximate solution. This algorithm can be easily repeated on the resulting subgraphs to get more segments. In the NC method, the most important parameter is the number of regions to be segmented. In our evaluation, we are going to vary this parameter to measure its performance.

4.2 Efficient Graph-Based Method (EG)

Similar to NC, EG[5,38] adopts a graph model and finds evidence of a boundary between two segments based on the intensity difference across the boundary and the intensity differences within each segment. However, the intensity difference within a segment is defined as the largest edge weight of the minimum spanning tree built from this segment, and the intensity difference across the boundary is defined as the minimum edge weight that connects these two segments. EG takes only $O(n \log n)$ computational time to segment an n-pixel image. In the adopted implementation,[38] there are three free parameters: a smoothing factor σ that is related to the Gaussian smoothing scales; a constant parameter K that controls how coarsely or finely an image is segmented; and a parameter S that constrains the minimum area of the resulting segments. Varying S usually results in a different number of segments. In our evaluation, we fix the smoothing factor σ to its default value and vary K and S to measure the segmentation performance.

4.3 Mean-Shift Method (MS)

MS[2,39] is a data-clustering method that searches for the local maximal density points and then groups all the data to the clusters defined by these maximal density points. When used for image segmentation, each pixel $\mathbf{x}_i$, $i=1,\ldots,n$, in the image is treated as an input data, and the density at point $\mathbf{x}$ is estimated by

$$\hat{f}(\mathbf{x}) = \frac{c}{nh^d} \sum_{i=1}^{n} K\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right),$$

where h is the bandwidth parameter, d is the data dimensionality, c is a normalization constant, and $K(\cdot)$ is the density estimation kernel. In the implementation of the mean-shift method,[39] the uniform kernel is used. To locate a local maximum of the density, an initial point $\mathbf{y}_1$ is selected and then successively updated by

$$\mathbf{y}_{j+1} = \frac{\sum_{i=1}^{n} \mathbf{x}_i K\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} K\left(\left\|\frac{\mathbf{y}_j - \mathbf{x}_i}{h}\right\|^2\right)}$$

until convergence. With these local maximal density points, the image can be segmented into regions by grouping each pixel to its corresponding local maximal density point. In the adopted implementation,[39] there are mainly three free parameters: the spatial bandwidth H_s; the range bandwidth H_r; and the minimum segment area S, which has the same meaning as the one in EG. Since all the test images in our benchmark are gray-level images, the range bandwidth H_r, which is mainly related to the color channels, is fixed to its default value. The bandwidth H_s determines the resolution in selecting the local maximal density points. In other words, H_s controls the number of resulting segments.

4.4 Ratio-Cut Method (RC)

RC[10,40] is another graph-based image segmentation method. As in NC, an image is modeled by a graph $G=(V,E)$ in RC, where V is a set of vertices corresponding to image pixels and E is a set of edges connecting neighboring pixels. Particularly, the 4-connectivity neighboring system is used in edge construction to make G a planar graph. The edge weight $w(u,v)$ is defined in a similar way to the ones defined in NC so that it describes the affinity between two vertices u and v based on their intensity similarity and spatial proximity. Then the image segmentation is formulated as finding a graph cut (A,B) to minimize the segmentation cost

$$\text{Rcut}(A,B) = \frac{\text{cut}(A,B)}{\text{assoc}(A,B)}, \quad (2)$$

where $cut(A,B)$ and $assoc(A,B)$ are defined in the same way as in NC. RC segments the image by finding the cut (A,B) with the minimum cost in Eq. (2). In Ref. 10, a polynomial-time algorithm is developed to find the minimum-cost ratio cut in a globally optimal fashion. Similar to NC, this algorithm can be repeated on the resulting subgraphs to get more segments. The most important parameter is the number of regions to be segmented. In our evaluation, we are going to vary this parameter to measure its performance.

4.5 Watershed Method (WS)

The Watershed method,[8,41] also called the watershed transform, is an image segmentation approach based on mathematical morphology. In geography, a watershed is the ridge that divides areas drained by different river systems. By viewing an image as a geological landscape, the watershed lines determine the boundaries that separate image regions. In the topographic representation of an image I, the numerical value (i.e., the gray tone) of each pixel stands for the elevation at this point. The watershed transform computes the catchment basins and ridge lines, with catchment basins corresponding to image regions and ridge lines relating to region boundaries. Methods for computing the watershed transform are discussed in detail in Refs. 8 and 42. In our evaluation, we use the watershed-transform function

of Matlab 7. However, the Matlab implementation of the watershed transform is very sensitive to image noise and usually produces over segmented regions. To solve this problem, we first smooth images with Gaussian smoothing filters of different scales before applying the watershed transform. By varying the parameter of Gaussian filters, we can segment an image into a target number of regions.

5 Performance Measure

To evaluate segmentation using this benchmark, the most desirable form of segmentation output is certainly a figure-ground-style segmentation, i.e., the image is partitioned into two segments, with one as the foreground and the other as the background. However, in most cases, the segmentation methods produce more than two regions. All the methods partition an image into a set of disjoint segments without labeling the foreground and background. Consequently, we develop a region-merging strategy so that they can be fairly evaluated in the benchmark.

Suppose the segments in an image I are $\{R_1, R_2, \ldots, R_n\}$. $R_i \cap R_j = \varnothing$ for $i \neq j$, and $\cup_{i=1}^{n} R_i = I$. In this case, the ground-truth foreground segment corresponds to a subset of the disjoint segments. To evaluate these methods in our benchmark, we apply a strategy to merge the segments and then use the merged region as the detected foreground object. For each segment R_i in an image, we count it into the foreground R if it has more than 50% overlap with the ground-truth foreground A in terms of the area, i.e.,

$$R = \bigcup_{i: \rho(R_i, A) > 0.5} R_i.$$

where

$$\rho(R_i, A) = \max \left\{ \frac{|R_i \cap A|}{|R_i|}, \frac{|R_i \cap A|}{|A|} \right\}.$$

An example of using this merging strategy for performance evaluation is illustrated in Fig. 4(c). Here the image Fig. 4(a) is segmented into 5 regions: $\{R_1, R_2, R_3, R_4, R_5\}$, and the ground truth A is given in Fig. 4(b). Considering the above criterion for merging image regions, only R_3 is selected in R.

Note that in the merging process, we find the best subset of image segments with the assumption that both the shape and the location of the ground-truth object A are known. In real applications like video surveillance and content-based image retrieval, the location and the shape of the objects are unknown. Therefore, a postprocess has to be developed to determine whether one specific segmented region should be selected as part of the "detected object."

Consider the problem we mention in Section 1—many real images contain multiple salient objects in which the most salient one may not be unambiguously defined from the human perception. From the above analysis, we can still include such images in the database and simply label all salient objects to construct the ground truths. Applying the same evaluation process on each ground truth one by one, we should get a similar result on each ground truth. However, in such a situation, some segmented regions may overlap with more than two ground truths. We have to analyze the overall performance on all ground truths. This complicates our analysis of the segmentation results. With our benchmark settings, we only consider the situation that one salient object exists in each image.

The basic performance measure we implement for this benchmark is the Jaccard coefficient,[43] which measures the region coincidence between the segmentation result and the ground truth. Specifically, let the region A be the ground-truth foreground object and the region R be the merged segments derived from the segmentation result using the region-merging strategy. We define the region-based segmentation accuracy as

$$P(R;A) = \frac{|R \cap A|}{|R \cup A|} = \frac{|R \cap A|}{|R| + |A| - |R \cap A|}, \tag{3}$$

where $|\cdot|$ is the operation of computing the region area. Different from the region-coincidence-based GCE and LCE measures used in the Berkeley benchmark, this measure has no bias to the segmentation that produces overly large or small number of segments. The numerator, $|R \cap A|$, measures how much the ground-truth object is detected. The denominator, $|R \cup A|$, is a normalization factor that normalizes the accuracy measure to the range of $[0, 1]$. With this normalization factor, the accuracy measure penalizes the error of detecting irrelevant regions as the foreground segments (false positives). This region-based measure is insensitive to small variations in the ground-truth construction and incorporates the accuracy and recall measurement into one unified function: This measure involves both false positives and false negatives. Figure 4 shows sample segmentation results and their segmentation accuracy using the proposed strategy.

Note that the segmentation accuracy mentioned above only provides an *upper bound* of the segmentation performance by assuming an ideal postprocess of region merging for applications without a priori known exact ground truth. For the extreme case where each pixel is partitioned as a segment, the upper-bound performance obtained is a meaningless value of 100%. This is a little similar to the GCE and LCE measures developed in the Berkeley benchmark. But the difference is that GCE and LCE also result in meaningless high accuracy when too few segments are produced, such as the case where the whole image is partitioned as a single segment. In this paper, we always set the segmentation parameters to produce a reasonably small number of segments when applying the strategy to merge the image regions. For simplicity, we always refer to "upper-bound performance" as "performance" in later sections when there is no confusion.

6 Evaluation Results

In this section, we empirically evaluate the performance of NC, EG, MS, WS, and RC on the proposed benchmark. We first show and compare the performance of these methods and the effects of their respective parameters. Then we show the relation between the performance and the number of segments in each method. We also reveal the correlation among these methods and investigate the performance by choosing the best segmentation method (out of these five methods) for each individual image.
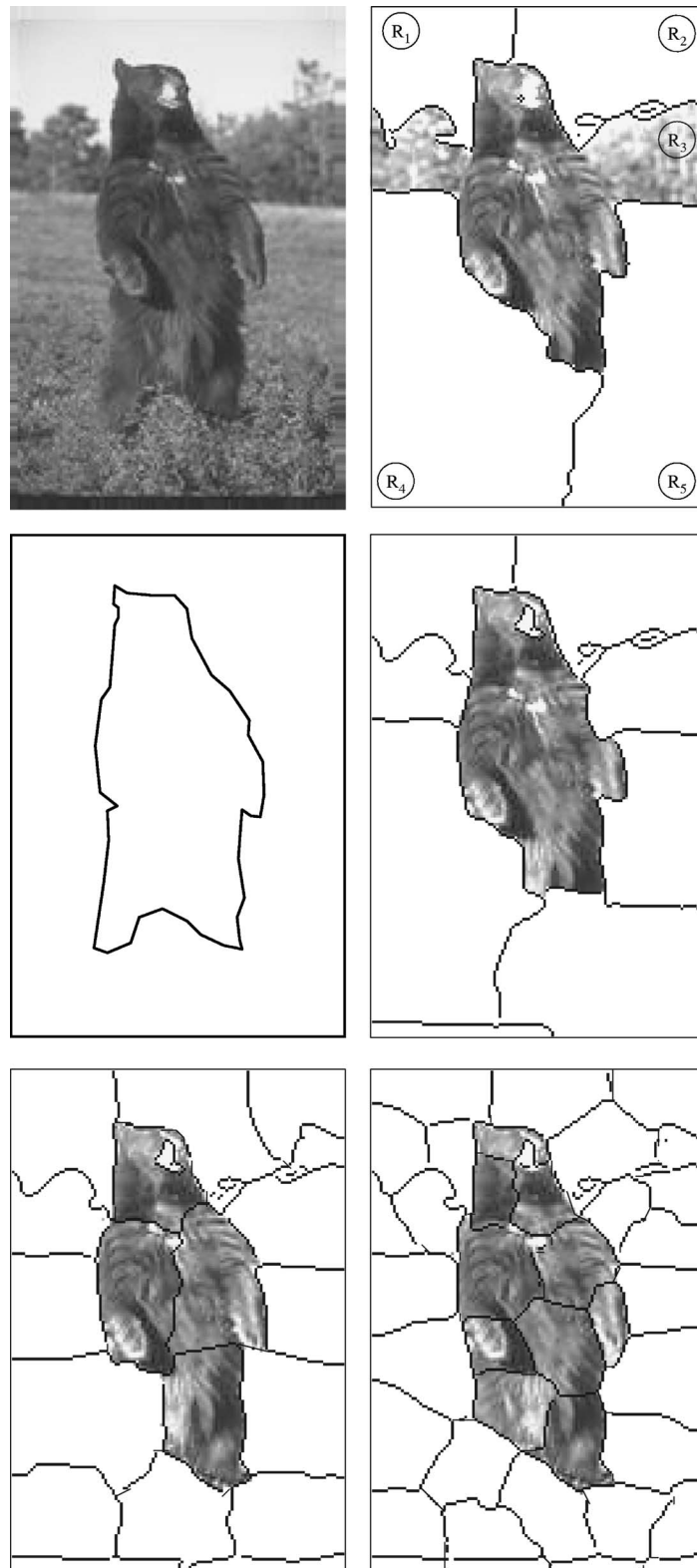
Fig. 4 Sample image segmentation results and their performance values. (a) Original image; (b) ground-truth segmentation; (c)–(f) segmentation results with different performances. In (c)–(f), the background regions are shown as white regions; the detected object regions are shown as the whitened original image; the boundaries of the regions are shown as black lines in the figure. The accuracy of segmentation results in (c)–(j) is 0.5, 0.7, 0.8, and 0.9, respectively.

Table 1 The average performance of EG (on all 1023 images) at different parameter settings. S_{min} indicates the minimal value corresponding to the case of allowing single-pixel segment.

Parameter K	Parameter S: The Minimum Allowed Segment Area Measured as the Percentage of the Image Area										
	50%	20%	10%	4%	2%	1%	0.5%	0.25%	0.125%	0.06%	S_{min}
100	0.26	0.33	0.44	0.58	0.68	0.76	0.82	0.85	0.87	0.89	0.90
300	0.26	0.33	0.45	0.60	0.68	0.74	0.77	0.79	0.80	0.80	0.81
500	0.26	0.35	0.46	0.60	0.66	0.70	0.72	0.73	0.74	0.74	0.75
1000	0.26	0.36	0.46	0.55	0.58	0.59	0.61	0.61	0.62	0.62	0.63

6.1 Performance Curve

In this section, we show the segmentation performance using a *cumulative-performance* histogram curve $p(x):[0,1] \rightarrow [0,1]$ (or *performance curve* in short), which describes the performance distribution on all 1023 images. In this curve, x represents the proportion of images, and $p(x)$ indicates the segmentation accuracy defined in Section 5. A specific point $(x, p(x))$ along this curve indicates that $100 \cdot x$ percent of the images are segmented with an accuracy lower than $p(x)$. Equivalently, this also means that $100 \cdot (1 - x)$ percent of the images produce segmentations with accuracy better than $p(x)$. Using a new segmentation method or a segmentation-parameter setting certainly will produce a new performance curve. Clearly, the higher a performance curve in the Cartesian coordinate system, the better the performance of the corresponding segmentation method and the parameter setting will be.

In this section, we also show the average performance $\bar{p}$ on all 1023 images in some tables. From the performance curve, the average performance can be derived by $\bar{p} = \int_0^1 p(x)dx$, and $p(0.5)$ is the median performance on all images. Clearly, the *cumulative-performance* histogram curve $p(x)$, $0 \le x \le 1$, describes the performance distribution on all images and therefore conveys more information than the average performance $\bar{p}$. The performance curve is continuous and monotonically nondecreasing. Two segmentation methods can have the same average performance but drastically different performance curves.

6.2 Segmentation Performance with Varied Parameter Settings

6.2.1 Efficient-graph method (EG)

The EG has two main parameters: K, which controls the splitting process of a segment, and S, which constrains the minimum area of each resulting segment. Table 1 shows that the parameter K affects the performance less than S does and the most appropriate value of K appears to be 100. For all tested values of K, the average performance $\bar{p}$ increases as the minimum region area S decreases. However, when S gets very small, $\bar{p}$ reaches a limit and cannot be improved any further. We can find in Table 1 that S is the dominant parameter in EG.

Figure 5(a) shows the performance curves resulting from varied S. The parameter K is fixed as 100. To make S invariant to the image size, we redefine S to be the ratio of the minimum allowed segment area to the total image area. We can clearly see the limit of $p(x)$ resulting from the decreased S. Even when we set S to be the minimal value S_{min} that allows single-pixel segments, $p(x) \not\equiv 1$ because the parameter K keeps an image from being overly segmented into individual pixels. In fact, we found that, when S is set to be the minimal value, the average number of produced segments in an image is around 500, which is too many for most applications. Figure 5(a) also shows that, when $S < 1\%$ (of the image area, as redefined above), the performance curve $p(x)$ moves up only marginally with the decreasing S. Table 1 and Fig. 5(a) suggest that an appropriate value of S is 1% and the performance curve $p(x)$ well approaches the limit when $S = 0.25\%$. In that case, the expected number of produced segments is 60.

6.2.2 Mean-shift method (MS)

The MS method has two main parameters: the level of resolution H_s and the minimum allowed segment area S. Similar to the EG, S is measured as the percentage of the image area. Table 2 shows the average performance $\bar{p}$ when setting different values for H_s and S. It indicates that the minimum allowed segment area S affects the performance $\bar{p}$ much more than H_s does. A better performance can usually be achieved when H_s is 1. Similar to EG, there exists a performance limit in MS because other parameters prevent an image to be segmented into individual pixels. Figure 5(b) shows the performance curves $p(x)$ with varied S and a fixed $H_s = 1$. Particularly, the performance curve $p(x)$ reaches the limit when the average number of produced segments is more than 2000. When the average number of produced segments gets over 24 (corresponding to $S < 1\%$), the performance curve moves up much more slowly. This suggests that, in this benchmark, it is appropriate to produce 10–100 segments when using MS, with a reasonable value of 40.

6.2.3 Normalized-cut method (NC)

In NC, we vary the parameter k, the target number of segments. The maximum possible value of k is the total num-
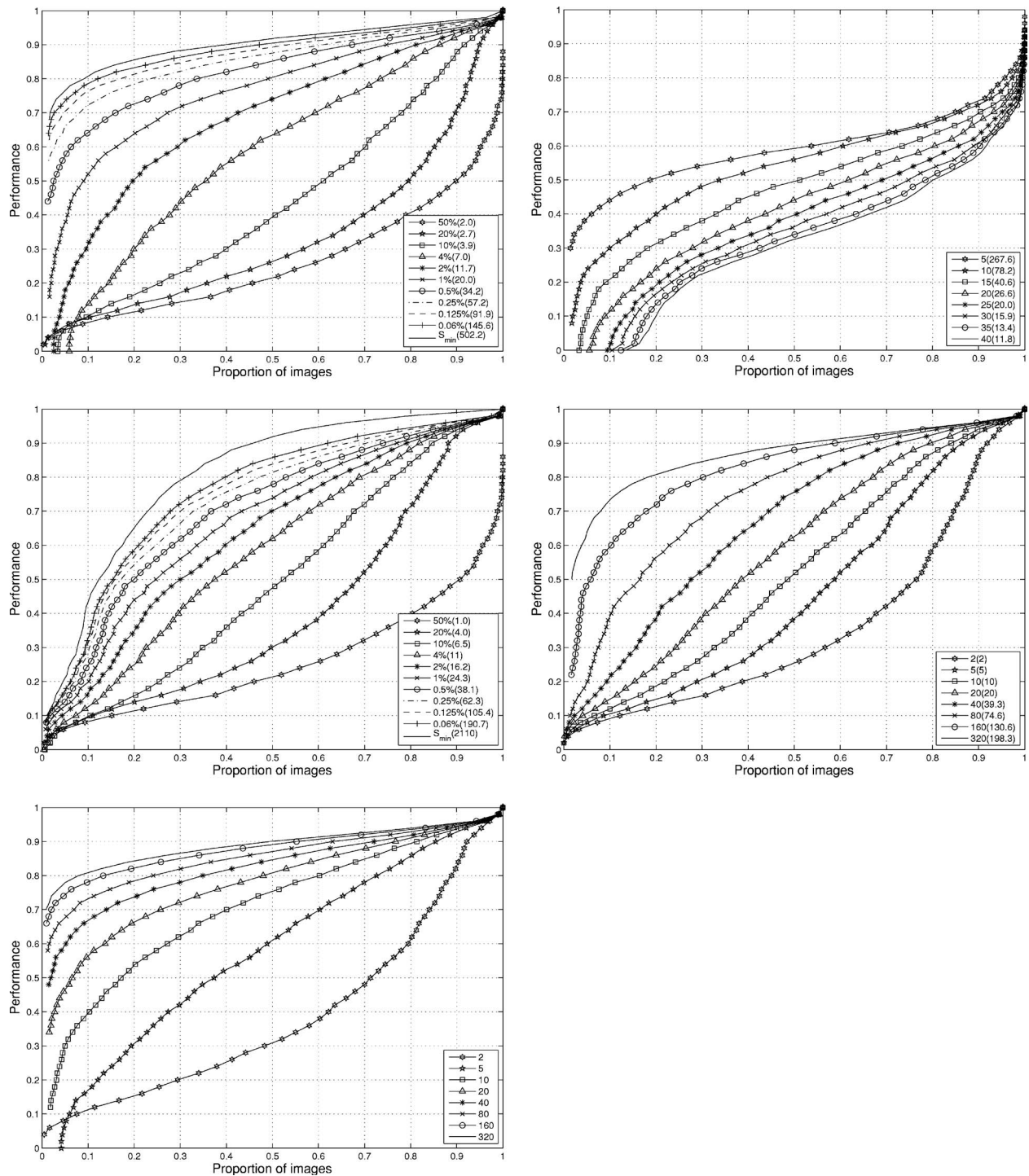
Fig. 5 The performance curves of the five image segmentation methods on the 1023 images in the database. (a) Efficient graph (EG); (b) mean-shift (MS); (c) normalized cut (NC); (d) watershed (WS); (e) ratio-cut (RC). In (a) and (b), the parameter is the minimum allowed segment area *S*. In (d), the parameter is σ in Gaussian filters. In (e), the varied parameter is the target number of regions. In (c) and in the parentheses of other subfigures, we show the average number of regions corresponding to the parameters.

ber of pixels; in that case $p(x) \equiv 1$, $x \in [0,1]$. As shown in Fig. 5(c), while the curve $p(x)$ moves up (not surprisingly) as k increases, it does not move up in a linear way in terms of the increase of k. The largest move-up of $p(x)$ happens when k increases from 2 to 5, and after that the move-up of

$p(x)$ is not substantial even if we increase k logarithmically. While a larger k improves the upper-bound performance $p(x)$, such an upper bound becomes more difficult to achieve because of the required postprocessing of region merging. Thus, we need to find an appropriate k by seeking

Table 2 The average performance of MS at different parameter settings. S_{min} indicates the minimal value corresponding to the case of allowing single-pixel segment.

Parameter H_s	Parameter S: The Minimum Region Area (measured as the percentage of the image area)										
	50%	20%	10%	4%	2%	1%	0.5%	0.25%	0.125%	0.06%	S_{min}
1	0.26	0.40	0.50	0.58	0.63	0.67	0.70	0.73	0.74	0.76	0.81
3	0.26	0.40	0.50	0.59	0.63	0.67	0.70	0.72	0.74	0.75	0.77
7	0.26	0.40	0.50	0.60	0.65	0.69	0.71	0.72	0.73	0.74	0.76
10	0.26	0.41	0.51	0.61	0.66	0.69	0.71	0.72	0.72	0.73	0.76

a compromise. From the experimental results shown in Fig. 5(c), we suggest selecting k to be less than 160, with 40 being the expected value when using NC on this benchmark.

6.2.4 Watershed method (WS)

The watershed transform usually leads to over segmentation of images due to image noise and other local irregularities. To overcome this problem, researchers have proposed many strategies such as region merging,[44] marker-controlled watershed segmentation,[8,41] hierarchical segmentation,[45] and multiscale segmentation.[46] In our evaluation, we use the Matlab function of watershed transform. To achieve segmentations with different number of segments, we adopt a strategy that is similar to that of the multiscale segmentation.[46] Before the watershed transform, each image is smoothed using a Gaussian filter of different scales. This preprocessing suppresses image noise and reduces the number of segments produced by the watershed transform. In the Gaussian filters, we vary the filter size N and the standard variation σ. Particularly, we set $N=\lfloor \frac{8}{5}\sigma \rfloor +1$; and Fig. 5(d) shows the performance of the WS with different Gaussian smoothing filters.

6.2.5 Ratio-cut method (RC)

The ratio-cut package[40] contains several parameters. In our experiment, we first use the default parameters to get a segmentation that is usually an oversegmentation of the input image. In this process, the ratio-cut algorithm iteratively partitions a segment into two subsegments until the ratio-cut cost is larger than a given threshold. The allowed range for this threshold is 0–765, and the default value of this threshold is 735. In this package, an iterative region-merging algorithm is developed to reduce the number of segments; the merging criterion is as defined in Eq. (2). The varied parameter for the ratio cut in our experiment is the target number of segments for merging. Note that, in practice, we may not get the target number of segments through merging in some images where the initial number of segments is smaller than the target number. In our experiment, we vary this target number in the range of 2–320 and find that the actual obtained average number of segments on all 1023 images are correspondingly varied in the range 2–102, as shown in Fig. 5(e).

6.3 Performance Comparison
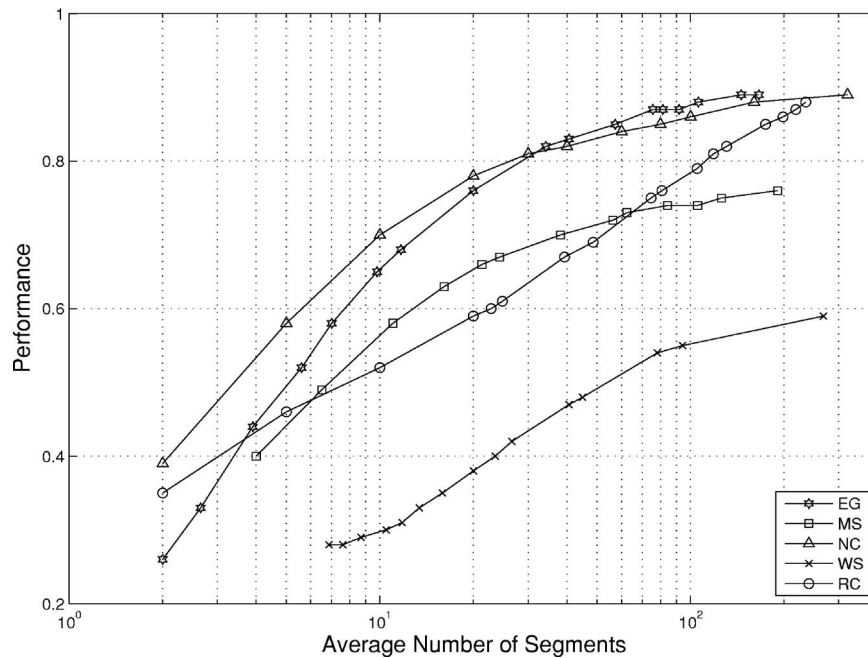
6.3.1 Comparison of the average performance

We first compare the average performance of different image segmentation methods. To make a fair comparison, we compare the average performance when images are segmented into the same number of segments. Table 3 shows the average performance $\bar{p}$ of these methods in terms of the number of produced segments. For EG and MS, the number of produced segments are controlled by the parameter S, i.e., the minimum allowed segment area. Therefore, we continuously vary S to achieve segmentation with a different number of segments. For NC, we directly control the number of segments for each image. For WS, we vary the Gaussian smoothing filter to achieve the target number of regions. For RC, we vary the target number of regions for each image.

From Table 3, we can see that, in the proposed benchmark, the average performance of the four methods (EG, MS, NC, RC) is saliently better than WS for all selected numbers of segments. The performance of the EG, MS, NC, and RC are very close, although the EG and NC methods are slightly better than MS and RC in performance. For all five methods, the average performance increases with the increase in image segments. However, as mentioned above, this performance shown here is an upper-bound one: With the increase of the resulting segments, this upper-bound performance becomes much more difficult to reach

Table 3 Comparison of the average performance of five image segmentation methods.

Number of Segments	EG	MS	NC	WS	RC
2	0.28	0.32	0.39	0.25	0.35
5	0.52	0.46	0.58	0.28	0.46
10	0.65	0.57	0.70	0.30	0.52
20	0.76	0.66	0.78	0.38	0.59
40	0.83	0.71	0.82	0.47	0.67
80	0.87	0.73	0.85	0.54	0.76

Fig. 6 The average performance of the five image segmentation methods in terms of the number of segments.

through region merging. From this perspective, the upper-bound performance derived from oversegmentation (more than 100 segments) is largely meaningless.

6.3.2 *Performance vs. the number of segments*

To further investigate the relation between the performance and the number of segments, we evaluated the average performance of the methods when different number of segments are produced. Figure 6 shows the trend of the average performance with the increase of the number of segments. We have two observations here: (1) A trade-off exists between the number of segments and the segmentation performance. Although the average performance is always monotonically increasing in all five methods, their increase speeds decrease when the number of segments gets big. For example, when the number of segments increases from 100 to 300 in the NC method, the average performance increases by only less than 0.03. Such an increase is almost meaningless, since an increase of 200 segments makes the postprocessing of region merging much more difficult and, therefore, this 0.03 increase of the upper-bound performance may not be achieved at all in practice. (2) There exists a performance limitation in some segmentation methods. Theoretically, when each pixel is partitioned as a segment, a perfect performance of 1.0 is achieved. However, most image segmentation methods do not allow such trivial segmentation. For example, in EG, even when the minimum allowed segment is set to be 1 pixel, the parameter K prevents each pixel from being partitioned as a separate segment.

In Fig. 6, we can also see that the NC method is slightly better than the other methods when fewer than 30 segments are produced. When 30–100 segments are produced, the performance of the EG and NC is close and is better than the performance of the other methods. As mentioned above,

we usually have little interest when more than 100 segmentations are produced. Table 3 and Fig. 6 also suggest the appropriate choices for segmentation parameters. It shows that, for the images in this benchmark, EG, MS, and NC all reach reasonably good performance when images are segmented into no more than 80 segments. The experimental results show that the appropriate range of the number of segments is 10–100. Particularly, around 40 segments are expected to be the target for EG, MS, and NC. The performance curve of the RC appears to be more linear, and an appropriate number of segments is 100.

From Table 3 and Fig. 6, we can surely draw the conclusion that the segmentation problem defined in this paper, i.e., separating one specified salient object from the background when partitioning an image into a relatively small number of segments, is far from solved with the state-of-the-art segmentation methods. Note that the performances discussed above are still the upper bounds that are usually difficult to reach in real applications. Also, remember that these 1023 images are carefully examined beforehand so that the human visual system is able to unambiguously extract the single ground-truth foreground object. From this perspective, we can see that there is still a long way to go to solve the general-purpose segmentation, where the ground truth may not be well defined.

6.3.3 *Comparison of winning cases*

To better compare the relative performance of different image segmentation methods, we also count the number of images on which one method outperforms the others. For example, if NC achieves the best performance on an image I_j, we consider NC the winner on I_j. For each method, we choose the best parameter setting from all the parameters we tested. We then count the number of winning images of each segmentation method and show the result in Table 4.

Table 4 The number of winning times of each method in terms of the number of produced segments.

Number of Segments	EG	MS	NC	WS	RC
2	102	163	498	92	168
5	292	189	325	67	150
10	258	197	379	18	171
20	321	205	296	12	189
40	381	206	223	5	208
80	416	213	186	1	207

From Tables 3 and 4, we can see that when fewer than 10 segments are produced, NC wins the most times among the five methods, especially at when 2 segments are performed; then NC performs better on almost half of the images (498). When targeting for more than 20 segments, EG wins more times than the other methods. Since the average performance of EG is very close to that of NC, it indicates that EG may win only marginally on most images. WS wins the fewest times and has the worst performance. Basically, for EG, MS, NC, and RC, there is no strong evidence (based on Tables 3 and 4) showing that one specific method is apparently superior to the others. In fact, their average performances are very similar when the number of segments is 80.

Several other reasons prohibit us from ranking the five segmentation methods: (1) Most performances listed here are estimations of upper bounds. Whether we can reach or approach the upper bound largely depends on specific applications; (2) many methods are not especially developed for the figure-ground-style segmentation formulated in this paper. Their performance may still be significantly improved if they are tuned to the figure-ground segmentation.

6.4 *Combination of image segmentation methods*

Besides evaluating and comparing the performance of individual image segmentation methods, it is also important to know whether and how these methods are statistically related. If these five methods can complement each other, then it would be worthwhile to further investigate ways to boost the performance by selecting the best segmentation method for different images. The experimental results can provide useful information regarding how much the combination of segmentation methods can improve performance. Such information can be used by learning-based segmentation algorithms.[21]

To better understand the correlation of these methods, we pick the best method (out of the five test methods) for each individual image and investigate the performance. We call this virtual method the "combined method" and its performance the "combined performance." This combined performance indicates the best performance we can get by "ideally" combining these five methods. Therefore, it is the upper-bound performance of these five methods. Note that this "combined method" is not a real method and cannot be

implemented in practice because it requires the ideal selection of the best method for each image. In this paper, we introduce the concept of the "combined method" only for the purpose of investigating the upper-bound performance of these five methods.

In combining the five methods, we specify the number of segments. Figures 7(a)–7(e) show the performance of this combined method when the resulting segments are 2, 5, 10, 20, and 40, respectively. At a low number of segments ($\leq$20), we can see that the performance of the combined method is clearly better than the other methods; at large number of segments (e.g., 40), it is only slightly better than the best method, the efficient graph-based method (EG). This may indicate that the learning-based segmentation algorithm should consider not only image features,[21] but also the number of image segments to achieve the best image segmentation results.

7 Discussions and Conclusions

In this paper, we presented a new benchmark for evaluating image segmentation. In this benchmark, image segmentation is evaluated according to its capability of separating a specified salient structure from the background with a relatively small number of segments. We find a large variety of images that satisfy the requirements of this benchmark to guarantee the generality and construct ground-truth segmentations to guarantee the objectivity. We develop a new strategy and a new concept of "upper-bound" performance to address the problems that many images may contain multiple salient structures and that many image segmentation methods may partition an image into more than two segments. Currently, we have collected 1023 natural images for this benchmark. In this paper, we applied this benchmark to evaluate the performance of five state-of-the-art image segmentation methods: the efficient graph-based method (EG), the mean-shift method (MS), the normalized-cut method (NC), the watershed method (WS), and the ratio-cut method (RC). We got the following observations and conclusions from the experiments.

1. Among the methods and the implementations we tested in the evaluation, WS has the worst performance using the proposed benchmark. It is clearly shown in Fig. 6 that WS is worse than the other methods, especially when the number of segments is over larger than 2. One possible reason is that the watershed transform is very sensitive to image noise so that many small segments are produced. An optimized watershed method may improve the performance.

2. When the number of image segments is less 20, NC outperforms the other methods. It is clearly shown in Fig. 6 and in Table 4. When the images are segmented into two regions, RC achieves the second-best performance. However, its performance quickly reduces to the second worst (only better than WS) when the number of segments is larger than 5. The performance of EG increases very quickly when the number of segments is getting large. It reaches the second-best position when the number of segments is 5. Among these five evaluated methods, MS achieves
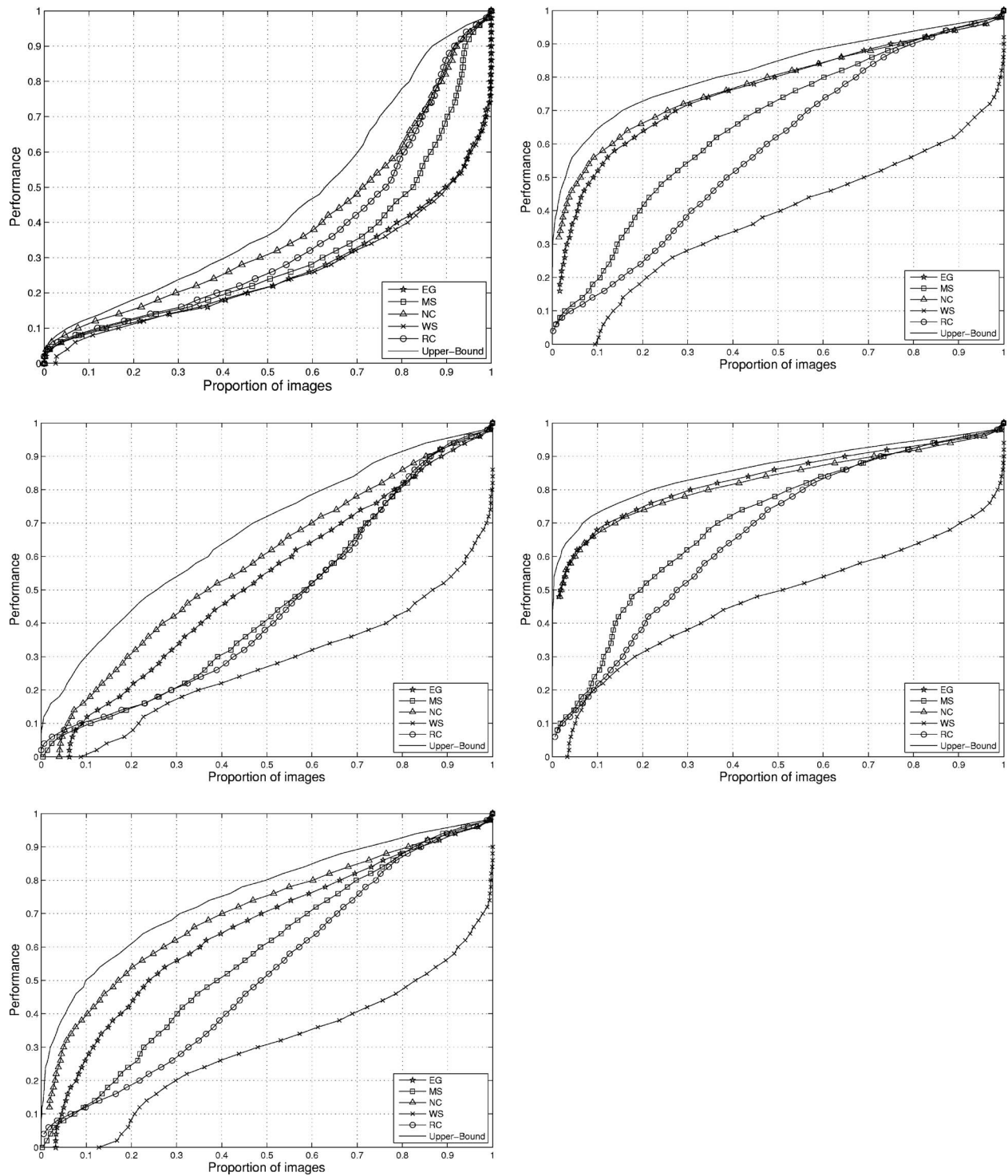
Fig. 7 The performance curves of each segmentation method and of the combined method when the average number of produced segments is 2, 5, 10, 20, and 40, respectively.

the median performance at different number of segments.

3. When the number of image segments is larger than 20, the performances of NC and EG are very close, with EG marginally better. As EG splits one segment into two according to the intensity differences across the boundary, it is better at locating object boundaries when the number of segments is getting large. How-

ever, an oversegmentation to too many regions will introduce the difficulty in merging image regions.

4. The experiments also show how the combination of different image segmentation method may improve the performance. As shown in Fig. 7, the "combined" method is superior to all the individual methods. However, the performance gained by combining methods diminishes when the number of segments is

getting large. When the images are segmented into a small number of regions (≤ 10), it is worthwhile to investigate the strategy to combine different image segmentation methods.

5. Our experiments show that general-purpose segmentation is still a problem far from well solved. Based on our observation, a segmentation with a performance of 70% or above can capture the most of the object (see Fig. 4 for reference). If we consider 70% as a performance threshold above which the object can be considered as "detected," then the percentages of the objects that can be detected are 15%, 40%, 60%, 75%, and 88% when the images are segmented into 2, 5, 10, 20, and 40 regions, respectively. Without a merging process, all these methods cannot detect objects very well. We recommend the segment number to be set at 10 so that the objects can be detected from a majority of images.

This benchmark provides a new perspective to quantitatively evaluate image segmentation methods. It reflects the performance in some real applications. For example, in the content-based retrieval of image databases, it is very helpful to detect and extract the salient objects in the images. A large amount of the images in real image databases contain only one salient object (e.g., animal, plant, flower, building), and our performance evaluation reveals the object detection performance when different segmentation methods are applied to these images. In the application of video surveillance, the foreground object (car, human, etc.) needs to be segmented from the background (street, road, etc.). Our evaluation results can also, to some extent, reflect the performance of separating the foreground objects from background without resorting to motion and other domain specific features.

We make this benchmark publicly available to other researchers[*] and hope it will promote research on image segmentation and segmentation evaluation.

Acknowledgments

References

1. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Color- and texture-based image segmentation using EM and its application to image querying and classification," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(8), 1026–1038 (2002).
2. D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), 603–619 (2002).
3. Y. Gdalyahu, D. Weinshall, and M. Werman, "Stochastic image segmentation by typical cuts," *Proc. IEEE Conf. Comp. Vis. Patt. Recog.*, pp. 596–601 (1999).
4. R. M. Haralick and L. G. Shapiro, "Survey: Image segmentation techniques," *Comput. Vis. Graph. Image Process.* **29**, 100–132 (1985).
5. D. H. P. Felzenszwalb, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.* **59**(2), 167–181 (2004).
6. J. Sethian, *Level Set Methods and Fast Marching Methods*, Cambridge Univ. Press, Cambridge (1999).
7. J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000).
8. L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithms based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(6), 583–598 (1991).
9. H. Nguyen, M. Worring, and R. Boomgaard, "Watersnakes: Energy-driven watershed segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(3), 330–342 (2003).
10. S. Wang and J. M. Siskind, "Image segmentation with ratio cut," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(6), 675–690 (2003).
11. M. Everingham, H. Muller, and B. Thomas, "Evaluating image segmentation algorithms using the Pareto front," *Proc. Eur. Conf. Comp. Vis.*, pp. 34–48 (2002).
12. B. McCane, "On the evaluation of image segmentation algorithms," in *Digital Image Computing: Techniques and Applications*, pp. 455–461 Auckland, New Zealand (1997).
13. C. W. Shaffrey, I. H. Jermyn, and N. G. Kingsbury, "Psychovisual evaluation of image segmentation algorithms," in *Advanced Concepts for Intelligent Vision Systems*, pp. 1–7, Ghent, Belgium (2002).
14. H. Zhang, J. E. Fritts, and S. A. Goldman, "An entropy-based objective segmentation evaluation method for image segmentation." in *SPIE Storage and Retrieval Methods and Applications for Multimedia*, Bellingham, MA, pp. 38–49 (2004).
15. Y. Zhang, "A survey of evaluation methods for image segmentation," *Pattern Recogn.* **29**(8), 1335–1346 (1996).
16. M. D. Levine and A. M. Nazif, "Dynamic measurement of computer generated image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **7**(2), 155–164 (1985).
17. 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," *Proc. Int. Conf. Comp. Vis.*, 2, 416–425 (2001).
18. F. J. Estrada and A. D. Jepson, "Quantitative evaluation of a novel image segmentation algorithm," *Proc. IEEE Conf. Comput. Vis. Patt. Recog.*, Vol. 2, 20–26 (2005).
19. J. P. Eakins and M. E. Graham, "Content-based image retrieval: A report to the JISC Technology Applications Program." Newcastle, Inst. for Image Data Research, Univ. of Northumbria (1999).
20. P. J. Withagen, "Object detection and segmentation for visual surveillance," PhD dissertation, Proefschrift Universiteit van Amsterdam, The Netherlands, ISBN: 9789090202822 (2006).
21. Y. Xia, D. Feng, R. Zhao, and M. Petrou, "Learning-based algorithm selection for image segmentation," *Pattern Recogn. Lett.* **26**(8), 1059–1068 (2005).
22. P. L. Correia and F. Pereira, "Stand-alone objective segmentation quality evaluation," *J. Appl. Signal Process.* **4**, 389–400 (2002).
23. M. Borsotti, P. Campadelli, and R. Schettini, "Quantitative evaluation of color image segmentation results," *Pattern Recogn. Lett.* **19**, 741–747 (1998).
24. J. Liu and Y.-H. Yang, "Multiresolution color image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(7), 689–700 (1994).
25. W. J. Niessen, C. J. Bouma, K. L. Vincken, and M. A. Viergever, "Error metrics for quantitative evaluation of medical image segmentation," *Theor. Found. Comp. Vis.*, pp. 275–284 (1998).
26. A. B. Goumeidane, M. Khamadja, B. Belaroussi, H. Benoit-Cattin, and C. Odet, "New discrepancy measures for segmentation evaluation," *Int. Conf. Image Process.*, II, 411–414 (2003).
27. A. Cavallaro, E. D. Gelasca, and T. Ebrahimi, "Objective evaluation of segmentation quality using spatio-temporal context," *Int. Conf. Image Process.*, III, 301–304 (2002).
28. V. Chalana and Y. Kim, "A methodology for evaluation of boundary detection algorithms on medical images," *IEEE Trans. Med. Imaging* **16**(6), 642–652 (1997).
29. J. S. Cardoso and L. Corte-Real, "Toward a generic evaluation of image segmentation," *IEEE Trans. Image Process.* **14**(11), 1773–1782 (2005).
30. J. Freixenet, X. Muoz, D. Raba, J. Mart, and X. Cuf, "Yet another survey on image segmentation: Region and boundary information integration," *Eur. Conf. Comput. Vis.*, pp. 408–422 (2002).
31. M. Van Droogenbroeck and O. Barnich, "Design of statistical measures for the assessment of image segmentation schemes," *Int. Conf. Comput. Anal. Images Patt.*, pp. 280–287 (2005).
32. D. W. Paglieroni, "Design considerations for image segmentation quality assessment measures," *Pattern Recogn.* **37**(8), 1607–1617 (2004).
33. N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recogn.* **26**, 1277–1294 (1993).
34. Y. J. Zhang, "Evaluation and comparison of different segmentation algorithms," *Pattern Recogn. Lett.* **18**(10), 963–974 (1997).
35. F. Ge, S. Wang, and T. Liu, "Image-segmentation evaluation from the perspective of salient object extraction," *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, I, 1146–1153 (2006).

---

[*]The whole benchmark, including all the images, ground-truth segmentations, and the source codes for evaluations, can be found at http://www.cse.sc.edu/$\sim$songwang/document/correspondence-2.0.tgz.

36. S. Wang, T. Kubota, J. Siskind, and J. Wang, "Salient closed boundary extraction with ratio contour," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(4), 546–561 (2005).
37. T. Cour, S. Yu, and J. Shi, "Normalized cut image segmenation source code." http://www.cis.upenn.edu/~jshi/software/.
38. P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based segmentation source code." http://people.cs.uchicago.edu/~pff/segment/.
39. D. Comaniciu and P. Meer, "Mean-shift image segmenation and edge detection source code." http://www.caip.rutgers.edu/ riul/research/code/EDISON/index.html.
40. S. Wang and J. M. Siskind, "Ratio-cut software." http://www.cse.sc.edu/~songwang/.
41. F. Meyer and S. Beucher, "Morphological segmentation," *J. Visual Commun. Image Represent* **1**(1), 21–46 (1990).
42. P. Soille, *Morphological Image Analysis: Principles and Applications* 2nd ed., Springer-Verlag, New York (2003).
43. T. Cox and M. Cox, *Multidimensional Scaling*, 2nd ed., Chapman & Hall/CRC Press, Boca Raton, FL (2000).
44. S. Beucher and F. Meyer, "The morphological approach of segmentation: The watershed transformation." in *Mathematical Morphology in Image Processing*, Marcel Dekker, New York (1992).
45. L. Najman and M. Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(12), 1163–1173 (1996).
46. P. Salembier, "Morphological multiscale segmentation for image coding," *Signal Process.* **38**(3), 359–386 (1994).

Feng Ge received his BS degree in engineering mechanics from Tsinghua University, Beijing, 2002 and received his MS degree in computer science and engineering from the University of South Carolina in 2006. Now he is a graduate student with the Department of Electrical and Computer Engineering at Virginia Tech, Blacksburg, VA.

Song Wang received his PhD degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign in 2002. From 1998 to 2002, he worked as a research assistant in the Image Formation and Processing Group of the Beckman Institute, University of Illinois at Urbana-Champaign. Since August 2002, he has been an assistant professor in the Department of Computer Science and Engineering at the University of South Carolina. His research interests include computer vision, medical image processing, and machine learning.

Tiecheng Liu received his PhD degree in computer science from Columbia University in 2003. Since 2003, has been with the department of computer science and engineering at the University of South Carolina. He is now an assistant professor. His main research interests include computer vision, image and video processing, multimedia, and advanced learning technologies. He published over 30 refereed papers in the area of computer vision and multimedia technology and has served as committee member for IEEE MCBAR'04, CIVR'05, and other conferences. He is a member of IEEE and ACM.