A New Benchmark for Image-Segmentation Evaluation

Feng Ge, Song Wang, and Tiecheng Liu
Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29208
{gef|songwang|tiecheng}@cse.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. In this paper, we present a new benchmark to evaluate five different image segmentation methods according to their capability of separating 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: (a) most image-segmentation methods are not developed to directly extract a single salient structure; (b) 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.

Keywords: Image-Segmentation Evaluation, Figure-Ground Segmentation, Performance Upper Bound Analysis.

1. Introduction

By partitioning an image into a set of disjoint segments to represent image structures, image segmentation leads to more compact image representations and bridges the gap between the low-level and the higher-level structures. As the central step in computer vision and image understanding, image segmentation has been extensively investigated in the past decades, with the development of a large number of image-segmentation methods [10, 22, 9, 14, 16, 1, 2, 23, 13, 28]. However, general-purpose image segmentation is still an unsolved problem; we still lack reliable ways in performance evaluation for quantitatively positioning the state of the art of image segmentation. In many prior works, segmentation performance is usually evaluated by subjectively or objectively judging on several sample images [19, 31, 12, 6, 15, 20, 46]. Such evaluations on a small number of sample images lack statistical meanings 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 [11].

For benchmark-based image-segmentation evaluation [11, 5], 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 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 [11], 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 the objectivity and the generality in evaluating image segmentation. Particularly, *we specify the goal of image segmentation as extracting the single most salient structure in the image.* In this formulation, the ground truth is a segment with 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 structure as the foreground *figure*, and the remaining portion as the *background*, such a formulation is usually referred to as *figure-ground* segmentation in prior literatures. 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 a more general meaning than a triv-

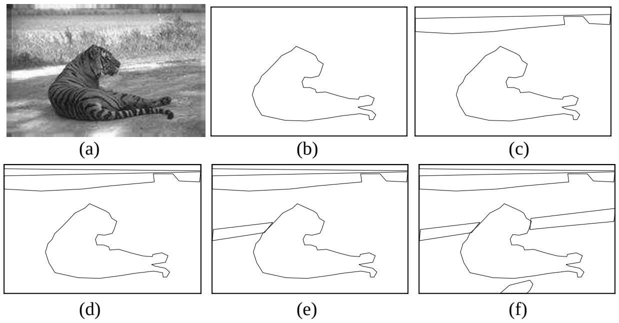


Figure 1: The figure-ground segmentation is usually better defined than the general-purpose segmentation: (a) a sample image. (b) The unambiguous ground truth in the figure-ground segmentation. (c-f) Four different ground-truth segmentations produced by different people in the general-purpose segmentation, i.e., partitioning the image into an unfixed number of segments.

ial background region of homogenous intensity or uniform texture as assumed in prior figure-ground segmentation literatures. Actually the background segment may contain many other structures. 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 structure. Instead, they usually partition an image into more than two disjoint segments, as shown in Fig. 1(c-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 structure from the background while keeping the number of resulting segments to be small. Based on this strategy, all the five segmentations shown in Fig. 1(b-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 structures, and the salient structure is not unambiguously defined. Although in our benchmark, we collect only the images with an unambiguous salient structure, we expect that the images with multiple salient structures 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 structure from the background with a small number of segments*. While the general-purpose image segmentation might be formulated in different ways in different applications, we believe that the capability of separating the salient structures from background would be a more general measure for evaluating its performance. Such a formulation of image segmentation also has many applications in computer-vision tasks, such as content-based image retrieval.

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 six 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

There has been a large number of literatures on the image-segmentation evaluation developed in the past decades. Most of previous works are focused on developing better ways to measure the accuracy/error of the segmentation. Some of them [47, 37, 36] do not require the 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 [36], 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 [19, 31, 12, 6, 15, 20, 33, 39, 34, 35, 42, 43, 44, 40]. Since the construction of the ground-truth segmentation for many real images is labor intensive and sometimes not well or uniquely defined, most of prior image-segmentation methods are only tested on: (a) some special classes of images used in special applications where the ground-truth segmentations are uniquely defined, (b) synthetic images where ground-truth segmentation is also well defined, and/or (c) a small set of real images.

For examples, in [33], a performance measure is developed to evaluate the medical image segmentation, where the test images are synthesized according to a medical-imaging model. In [39], 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 the work [39] is to investigate whether an automatic segmentation agree with the observers' segmentation and whether the different observers' segmentations agree with each other. Goumeidane et al [34] suggest a performance measure based on two distortion rates of the resulting segments to treat both under-detected and over-detected pixels. Experiments are conducted only on several simple binary synthetic images. Cavallaro et al [35] 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 labelled segmentation. Freixenet et al [42] 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 [6] suggest to evaluate segmentation from different perspectives but avoid combining them into a single performance measure. In [6], six general-purpose segmentation algorithms are evaluated on 100 samples images of urban and rural outdoor scenes. Droogenbroeck and Barnich [43] 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 [44] 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 [40] recently develop 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 [45] and Zhang [31, 41] provide surveys of some early image-segmentation evaluation methods.

Different from these above methods, this paper presents a benchmark for evaluating general-purpose image segmentation method on a large variety of real images. The work most related to ours is the Berkeley image-segmentation benchmark [11]. 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. Without any special guidance, such manual segmentations reflect the general human perception and therefore, different people usually construct different manual segmentation on the same image. Particularly, different people may partition an image into different number of segments, as illustrated in Fig. 1. The Berkeley benchmark collects all different manual segmentations of an image as the ground-truth segmentation, i.e., the ground-truth segmentation is non-unique. While this benchmark achieves good generality, it has some problems on the evaluation objectivity. 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, in this paper, a single ground-truth segmentation is constructed for each test image by extracting a salient structure from this image. We then *only* collect images with some identifiable salient structure into the benchmark. This way, it is easier to construct the ground-truth segmentation and define the segmentation accuracy while the evaluation generality can still be well kept with the large variety of the collected images. Particularly, with a single ground-truth segmentation, the proposed benchmark avoids the problem of tolerating unreasonable refinement in the evaluation measures as in the Berkeley benchmark. The contributions of this paper can be summarized as

1. By formulating the goal of image segmentation as extracting a salient structure 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-truth segmentations.

2. With a single defined ground truth segmentation that only consists of two segments, the segmentation performance measure can be more robustly defined and used. In this paper, we simply use the Jaccard coefficient [48] as the performance accuracy measure. In fact, many new measures developed in previous literatures, 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 structures.

4. We apply the develop 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 structure that is unambiguous in human visual perception. This way, the ground-truth segmentation can be easily constructed by manually extracting the closed boundary of this salient structure. 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 gray-scale 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 structure independently. Second, if both of them select the same structure without any reservation, this image will be included into the database. Otherwise, if they choose different structures or any one of them has reservations in determining the most salient structure, 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 structure.

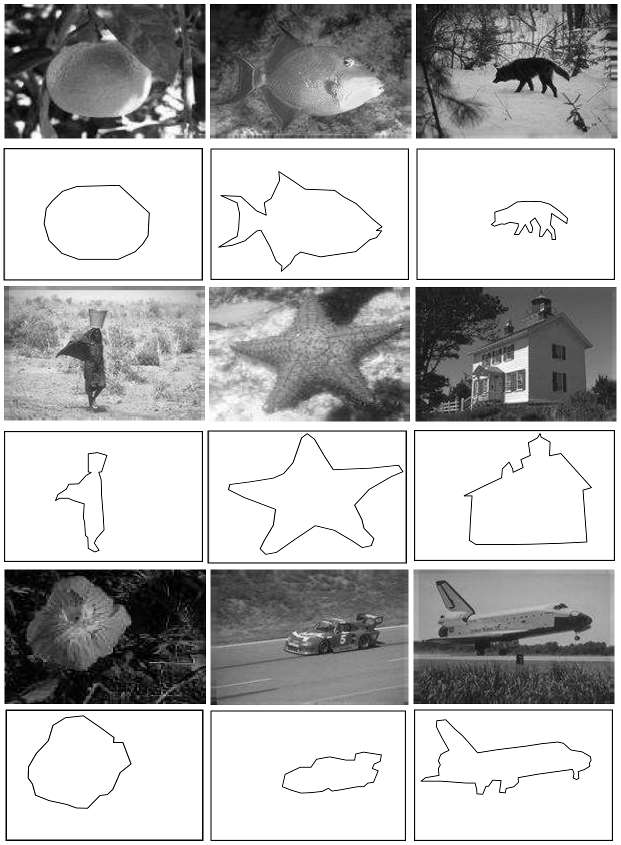


Figure 2: Nine sample images in our image database and the ground truth produced manually.

Figure 2 demonstrates several sample images and their ground-truth segmentations in the current image database. Note that we intentionally collect images with various foreground structures(such as human, animal, vehicle, building, etc.) and various backgrounds. Also note that, in the collected images, the salient structure may not be the only structure in the image, and the background may contain some structures 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 structure. In Section 5, we will develop a special strategy to handle this problem. With this special strategy, an image with multiple salient structures can still be evaluated. The only requirement is to pick one salient structure and label it as the ground-truth foreground. We believe the ground truths constructed by these two students well satisfy this requirement.

4. Selected Image-Segmentation Methods

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

- Normalized-cut method (NC) [16] implemented by Shi and Malik [4].
- Efficient graph-based method (EG) [13] implemented by Felzenszwalb and Huttenlocher [8].
- Mean-shift method (MS) [2] implemented by Comaniciu and Meer [3].
- Watershed method (WS) [22] (Matlab implmentation).
- Ratio-cut method (RC) [28] implemented by Wang et al. [29].

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

Normalized-cut method (NC)[16, 4]. In NC, an image is modelled 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 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

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

where $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$ is the cut cost of (A, B) and $assoc(A, V) = \sum_{u \in A, v \in V} w(u, v)$ is the association between A and V. NC segments the image by finding the cut (A, B) with the minimum cost (1). Since this is a NP-complete problem, a spectral-graph algorithm was developed to find an approximate solution. This algorithm can be easily

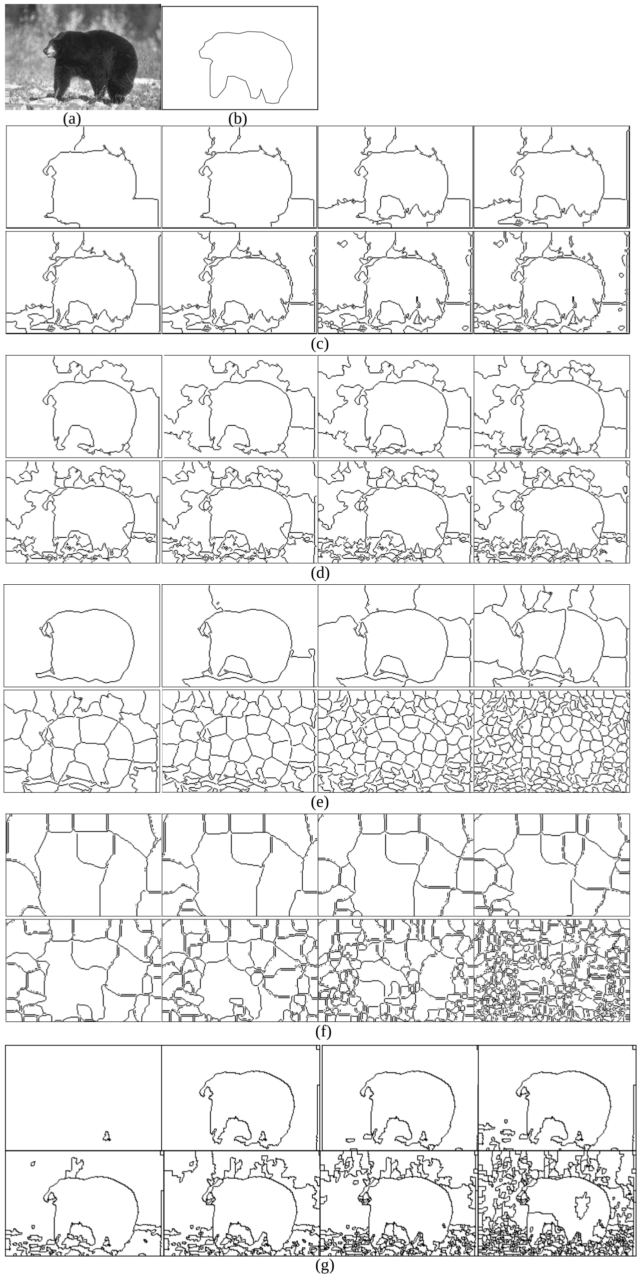


Figure 3: Sample image segmentation results using the six selected methods at different parameter settings. (a) Original image. (b) Ground truth. (c) EG segmentation. The eight results (from left to right, from top to bottom) are obtained by setting parameter S to 20%, 10%, 4%, 2%, 1%, 0.5%, 0.25%, and 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, 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.

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.

Efficient graph-based method (EG) [13, 8]. Similar to NC, EG adopts a graph model and finds the evidence of a boundary between two segments based on the intensity differences 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 [8], 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 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.

Mean-shift method (MS) [2, 3]. MS 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 $\mathbf{x}_i, i = 1, ..., n$ in the image is treated as an input data and the density at point $\mathbf{x}$ is estimated by

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

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 [3], 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 [3], 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 to 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.

Ratio-cut method (RC) [28, 29]. RC is another graph-based image-segmentation method. Like in NC, an image is modelled 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 similar way to the ones defined in NC os 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 minimizes the segmentation cost

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

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 (2). In [28], 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.

Watershed method (WS) [25, 22].

Watershed method, also called 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 images 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 [22, 30]. In our evaluation, we use the watershed-transform function of Matlab 7. However, the Matlab implementation of the watershed transform is very sensitivity 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 labelling 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 $\{R_1, R_2, \ldots, R_n\}$. $R_i \cap R_j = \emptyset$ 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 percent 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.

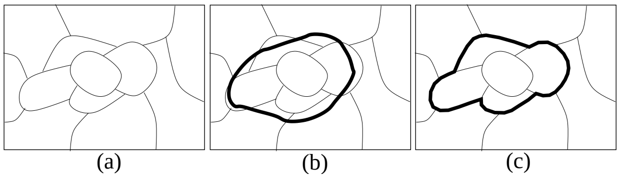

(a) (b) (c)

Figure 4: An illustration of evaluating an image segmentation result in the proposed benchmark: (a) an image-segmentation result; (b) the boundary of the ground-truth segmentation (the thick curve) overlapped on the segmentation result; (c) the figure-ground segmentation (the thick curve) derived using the proposed region-merging strategy.

Note that in the merging process, we find the best subset of image segments with the assumption that the ground-truth foreground object A is known. But in real applications, the foreground is not known beforehand. In this sense, by assuming that an ideal merging post-process always exists, the evaluation based on this strategy in fact represents an upper bound performance.

This strategy is particularly useful in addressing another important problem mentioned in Section 1 — Many real images contain multiple salient structures in which the most salient one may not be unambiguously defined from the human perception. Using this strategy, we can still include such images into the database and simply label one salient structure to construct the ground truth. The basic assumption underlying this evaluation strategy is that a good segmentation method should be able to detect a specified salient structure in an image even if this image contains multiple salient structures.

The basic performance measure we implement for this benchmark is Jaccard coefficient [48], which measures the region coincidence between the segmentation result and the ground truth. Specifically, let the region A be the ground-truth foreground structure 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 segmentations that produces overly large or small number of segments. The numerator, $|R \cap A|$, measures how much the ground-truth structure is detected. The denominator, $|A \cup R|$, is a normalization factor which 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. Fig. 5 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 postprocessing step of region merging. Note that this upper-bound performance may not be achieved or even approached in real applications, where the ground truth is not *a priori* known. In general, the upper-bound performance calculated using this strategy is useful only when the total number of segments, n, is small. For the extreme case where each pixel is partitioned as a segment, the upper-bound performance obtained is a meaningless value of 100 percent. 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 fewer 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 correlation among these methods and investigate the performance by choosing the best segmentation method (out of these five methods) for each individual image.

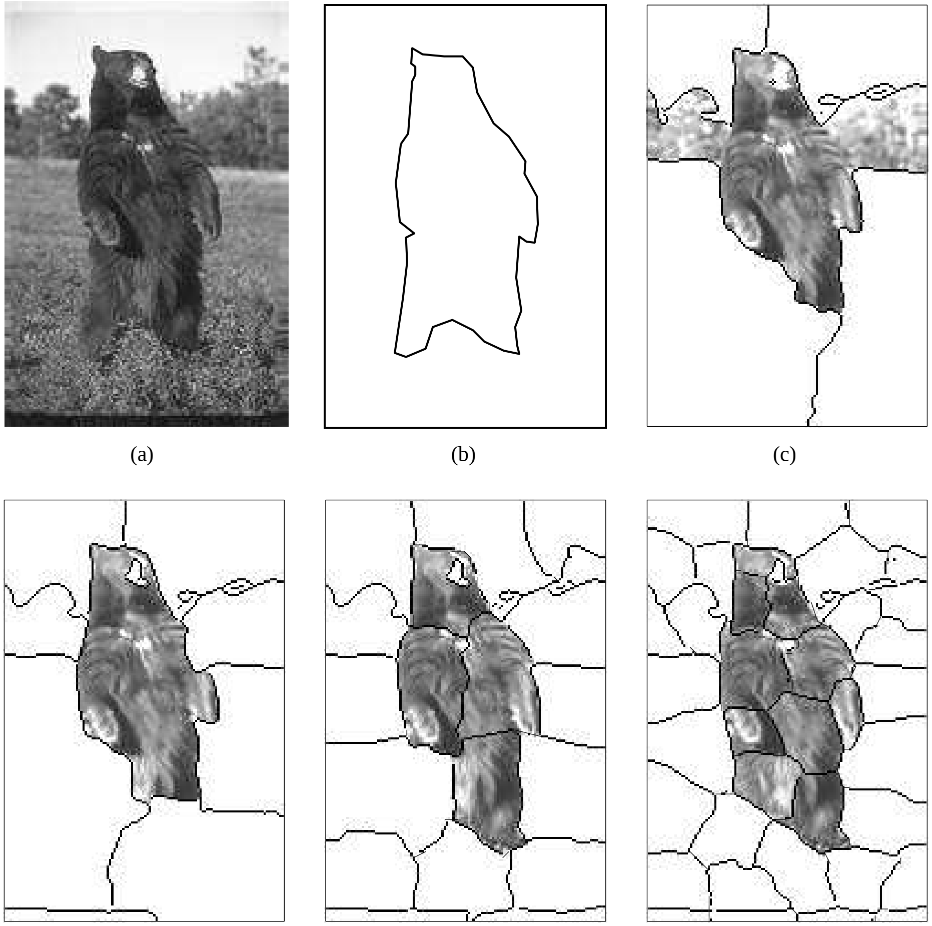
6.1 Performance Curve

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 Sections 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.

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 $p(x)$, $0 \leq x \leq 1$, 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 non-decreasing. Two segmentation methods can have the same average performance but drastically different performance curves.

6.2 Segmentation Performance with Varied Parameter Settings

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.

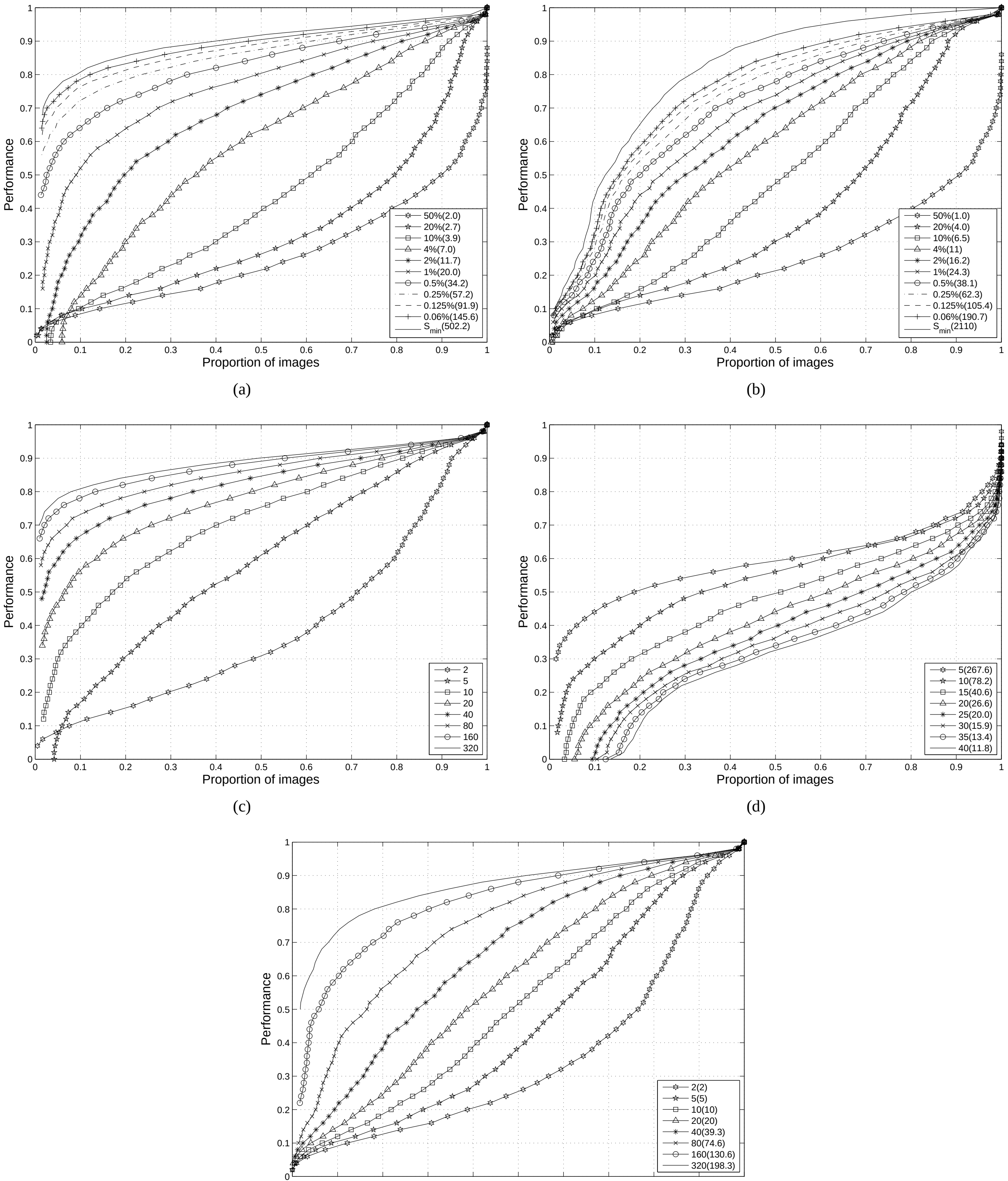

(a) (b) (c)

(d) (e) (f)

Figure 5: 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.

Figure 6 (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 6 (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 of S. Table 1 and Figure 6 (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.

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. 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 6 (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



(a)

(b)

(c)

(d)

(e)

Figure 6: 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 parenthesis of other subfigures, we show the average number of regions corresponding to the parameters.

	Parameter S: The minimum allowed segment area measured as the percentage of the image area										
Parameter K	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

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 S: The minimum region area (measured as the percentage of the image area)										
Parameter H_s	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

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.

of 40.

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 number of pixels; in that case $p(x) \equiv 1, x \in [0, 1]$. As shown in Fig. 6(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 post-processing of region merging. Thus we need to find an appropriate k by seeking a compromise. From the experimental results shown in Fig. 6 (c), we suggest selecting k to be less than 160, with 40 being the expected value when using NC on this benchmark.

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 [24], marker-controlled watershed segmentation [25, 22], hierarchical segmentation [27], and multi-scale segmentation [26]. 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 multi-scale segmentation [26]: 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. 6 (d) shows the performance of the WS with different Gaussian smoothing filters.

Ratio-cut method (RC). The ratio-cut package [29] contains several parameters. In our experiment, we first use the default parameters to get a segmentation that is usually an over segmentation of the input image. In this process, the ratio cut algorithm iteratively partitions a segment into two sub-segments 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 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 1030 images are correspondingly varied in the range $2 - 102$, as shown in Fig. 6 (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 different number of segments. For NC, we directly control the number of segments for each image. For MS, we vary the Gaussian smoothing filter to achieve the target number of regions. For RC, we vary the variable of target number of regions for each image.

Number of segments	EG	MS	NC	WS	RC
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

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

From Tables 3, we can see that, in the proposed benchmark, the average performance of the four methods (EG, MS, NC, RC) are saliently better than WS for all the selected number 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 these five methods, the average performance increases with the increase of 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 through region merging. From this perspective, the upper-bound performance derived from over-segmentation (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 7 shows the trend of the average performance with the increase of the number of segments. We have two observations here: (a) 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 NC method, the average performance only increases by 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. (b) 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. 7, we can also see that the NC method is slightly better than the other methods when less than 30 segments are produced. When $30 - 100$ segments are produced, the performance of the EG and NC are close, and are better than the performance of the other methods. As mentioned above, we usually have little interest when more than 100 segmentation are produced. Table 3 and Fig. 7 also suggest the appropriate choices of the 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. 7, we can surely draw the conclusion that the segmentation problem defined in this paper, i.e., separating one specified salient structure 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 in above are still the upper bounds that are usually difficult to reach in real applications. Also be reminded that these 1023 images are carefully examined beforehand so that the human visual system is able to unambiguously extract the single ground-truth foreground structure. 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.

From Tables 3 and 4, we can see that when less than 10 segments are produced, NC wins the most times among the five methods. When targeting for more than 10 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 least times and has the worst performance. Basically, for EG, MS, NC, and RC, there is no strong evidence (based

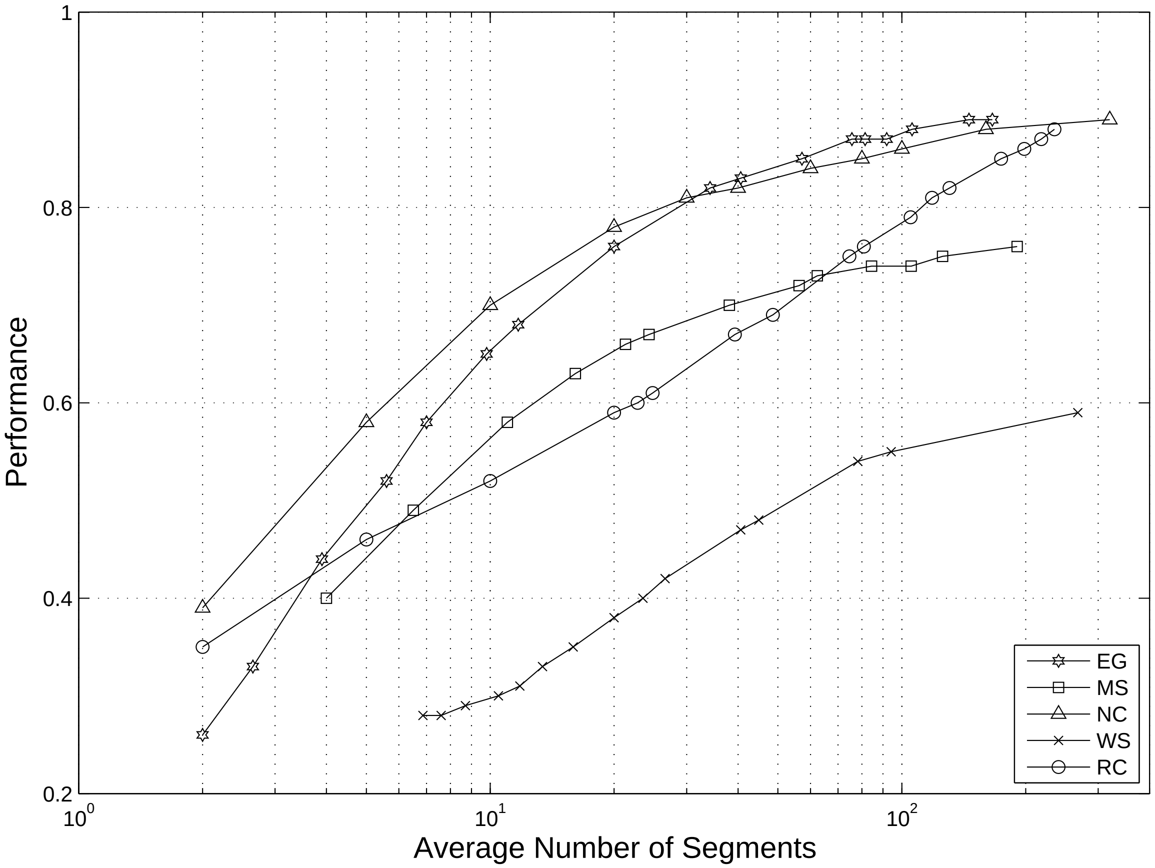


Figure 7: The average performance of the five image segmentation methods in terms of the number of segments.

Number of segments	EG	MS	NC	WS	RC
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

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

on Table 3 and Table 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: (a) Most performances listed here are estimations of upper bounds. Whether we can reach or approach the upper bound largely depends on specific applications; (b) Many methods are not especially developed for 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 method, 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 for researchers to further investigate ways to boost the performance by combining them. 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 as the "combined method" and its performance as 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" for the only purpose of investigating the upper bound performance of these five methods.

In combining the five methods, we specify the the number of segments. Figure 8(a-d) shows the performance of this combined method when the resulting segments are 5, 10, 20, and 40, respectively. We can see that the performance of this "ideal" combined method is not much better than that of each individual method. Especially when the number of segments gets big, e.g., 40, the performance gain by combining the methods is marginal. These results indicates that the combination of different image-segmentation methods does not substantially boost the the segmentation performance.

7. Contribution 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.

2. The performances of the EG, MS, NC, and RC are close. When less than 10 segments are produced, NC is slightly better than the other methods. When more than 10 segments are produced, the performance of the EG is marginally better than that of the other methods. But the performance differences among them are very small, and there is no obvious winner.

3. The experimental results provide useful information of parameter selection in these image segmentation methods. For all the five methods, the target number of segments should be in the range of $10 - 100$, with 40 being a typical number for EG, MS, and NC, and 100 being the expected number for RC.

4. There is no strong evidence showing that these segmentation methods complement each other. Therefore, a combination of them may not significantly boost the performance.

This benchmark provides a new perspective to quantitatively evaluate image-segmentation methods. However, our experiments show that general-purpose segmentation is still far from a solved problem even with the state-of-the-art methods. We make this benchmark available to other researchers [1] and hope it will help evaluate new image-segmentation methods.

Acknowledgements

This work was funded, in part, by NSF-EIA-0312861 and a grant from the University of South Carolina Research and Productivity Scholarship Fund.

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 Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[2] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[3] D. Comanicu and P. Meer. Mean-shift image segmenation and edge detection source code. http://www.caip.rutgers.edu/riul/research/code/EDISON/index.html.

[4] T. Cour, S. Yu, and J. Shi. Normalized cut image segmenation source code. http:// www.cis.upenn.edu/∼jshi/software/.

[5] F. J. Estrada and A. D. Jepson. Quantitative evaluation of a novel image segmentation algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 20–26, 2005.

[6] M. Everingham, H. Muller, and B. Thomas. Evaluating image segmentation algorithms using the pareto front. In *European Conference on Computer Vision*, pages 34–48, 2002.

[7] D. Fan. Level-set image segmenation source code. http://www.cs.wisc.edu/∼fan/LevelSet/.

[8] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based segmentation source code. http://people.cs.uchicago.edu/∼pff/segment/.

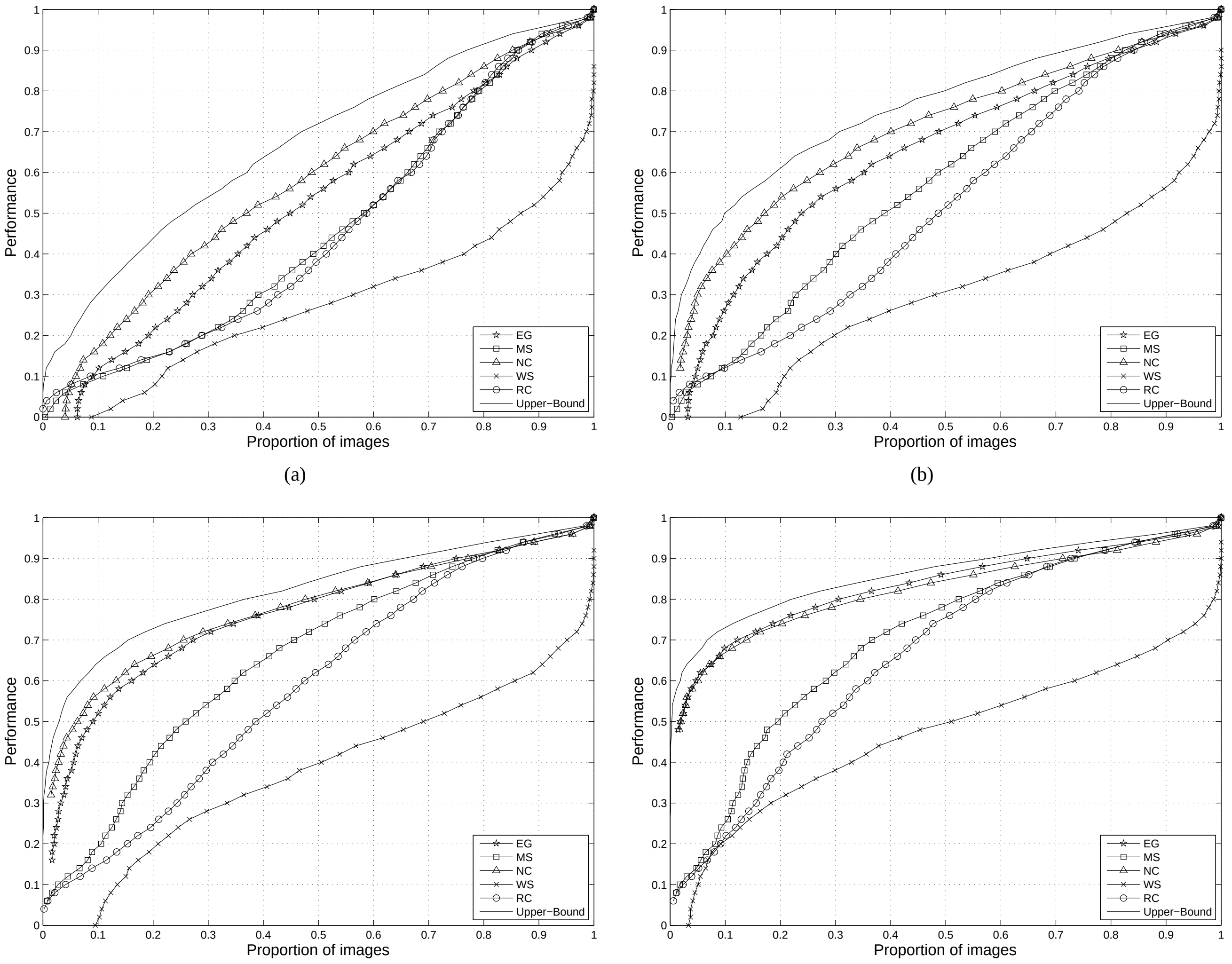
[9] Y. Gdalyahu, D. Weinshall, and M. Werman. Stochastic image segmentation by typical cuts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 596–601, 1999.

[10] R. M. Haralick and L. G. Shapiro. Survey: Image segmentation techniques. *Computer Vision Graphics Image Process*, 29:100–132, 1985.

[11] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision*, volume 2, pages 416–425, 2001.

---

[1] The whole benchmark, including all the images, ground-truth segmentations, and the source codes for evaluations, can be found from `http://www.cse.sc.edu/∼songwang/document/segmentation-benchmark.tgz`.



(a)
(b)
(c)
(d)

Figure 8: The performance curves of each segmentation method and of the combined method when the average number of produced segments is 5, 10, 20, and 40, respectively.

[12] B. McCane. On the evaluation of image segmentation algorithms. In *Digital Image Computing: Techniques and Applications*, pages 455–461, 1997.

[13] D. H. P. Felzenszwalb. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[14] J. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge, U.K.: Cambridge Univ. Press,, 1999.

[15] C. W. Shaffrey, I. H. Jermyn, and N. G. Kingsbury. Psychovisual evaluation of image segmentation algorithms. In *Advanced Concepts for Intelligent Vision Systems*, pages 1–7, 2002.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[17] S. Wang, T. Kubota, J. Siskind, and J. Wang. Salient closed boundary extraction with ratio contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):546–561, 2005.

[18] S. Wang, T. Kubota, and J. M. Siskind. Ratio-contour source code. http://www.cse.sc.edu/∼songwang/.

[19] L. Yang, F. Albregtsen, T. Lonnestad, and P. Grottum. Psychovisual evaluation of image segmentation algorithms. In *Lecture Notes in Computer Science*, pages 759–765, 1995.

[20] 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*, pages 38–49, 2004.

[21] Y. Zhang. A survey of evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.

[22] L. Vincent and P. Soille. Watersheds in Digital Spaces: An Efficient Algorithms Based on Immersion Simulations. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(6):583–598, 1991.

[23] H. Nguyen, M. Worring, and R. Boomgaard. Watersnakes: Energy-driven watershed segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):330–342, 2003.

[24] S. Beucher, F. Meyer. The Morphological Approach of Segmentation: The Watershed Transformation. Mathematical Morphology in Image Processing, New York:Marcel dekker, 1992.

[25] F. Meyer, S. Beucher. Morphological Segmentation. Journal of Visual Communication and Image Representation, 1(1):21-46, 1990.

[26] P. Salembier. Morphological Multiscale Segmentation for Image Coding. Signal Processing, 38(3):359-386, 1994.

[27] L. Najman, M. Schmitt. Geodesic Saliency of Watershed Contours and Hierarchical Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(12):1163–1173, 1996.

[28] S. Wang, J. M. Siskind. Image Segmentation with Ratio Cut. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):675–690, 2003.

[29] S. Wang, J. M. Siskind. Ratio-cut software. http://www.cse.sc.edu/ ∼songwang/.

[30] P. Soille. Morphological Image Analysis: Principles and Applications. 2nd ed., New York:Springer-Verlag, 2003.

[31] Y. Zhang. A survey of evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346, 1996.

[32] K. L. Vincken, A. S. E. Koster, C. N. de Graaf, and M. A. Viergever Model-based Evaluation of Image Segmentation Methods. *Theoretical Foundations of Computer Vision*, 299–311, 1998.

[33] W.J. Niessen, C.J. Bouma, K.L. Vincken, and M.A. Viergever. Error Metrics for Quantitative Evaluation of Medical Image Segmentation. *Theoretical Foundations of Computer Vision*, 275–284, 1998.

[34] A.B. Goumeidane, M. Khamadja, B.Belaroussi, H. Benoit-Cattin, and C. Odet, New discrepancy measures for segmentation evaluation. *International Conference on Image Processing*,II:411–414, 2003.

[35] A. Cavallaro, E.D. Gelasca, and T. Ebrahimi. Objective evaluation of segmentation quality using spatio-temporal context *International Conference on Image Processing*,III:301–304, 2002.

[36] P.L. Correia, and F. Pereira. Stand-Alone Objective Segmentation Quality Evaluation *Journal on Applied Signal Processing*,4:389–400, 2002.

[37] M. Borsotti, Pl. Campadelli, and R. Schettini. Quantitative Evaluation of Color Image Segmentation Results *Pattern Recognition Letters*,19:741–747, 1998.

[38] Ramn Romn-Roldn, Juan Francisco Gmez-Lopera, Chakir Atae-Allah, Jos Martnez-Aroza, Pedro Luis Luque-Escamilla A measure of quality for evaluating methods of segmentation and edge detection *Pattern Recognition*, 34(5):969-980,2001.

[39] V. Chalana, and Y. Kim. A methodology for evaluation of boundary detection algorithms on medical images. *IEEE Transactions of Medical Imaging*, 16(6):642-652,1997.

[40] J. S. Cardoso, and L. Corte-Real. Toward a Generic Evaluation of Image Segmentation *IEEE Transactions on Image Processing*, 14(11):1773-1782,2005.

[41] Y.J. Zhang. Evaluation and Comparison of Different Segmentation Algorithms. *Pattern Recognition Letters*, 18(10):963-974, 1997.

[42] J. Freixenet, X. Muoz, D. Raba, J. Mart, and X. Cuf. Yet Another Survey on Image Segmentation: Region and Boundary Information Integration. *European Conference on Computer Vision*, 2002.

[43] M. Van Droogenbroeck, and O. Barnich. Design of Statistical Measures for the Assessment of Image Segmentation Schemes. *International Conference on Computer Analysis of Images and Patterns*, 280-287, 2005.

[44] D.W. Paglieroni. Design considerations for image segmentation quality assessment measures. *Pattern Recognition*, 37(8):1607–1617, 2004.

[45] N. R. Pal, and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 1993.

[46] M.D. Levine, and A.M. Nazif. Dynamic measurement of computer generated image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2):155–164, 1985.

[47] J. Liu and Y.-H. Yang. Multiresolution color image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(7):689–700, 1994.

[48] T. Cox and M. Cox Multidimensional Scaling. 2nd ed. Chapman & Hall/(CRC), 2000.