Computer Vision and Image Understanding 113 (2009) 126–149

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

An iterative multi-scale tensor voting scheme for perceptual grouping of natural shapes in cluttered backgrounds

Leandro Loss [a], George Bebis [a,*], Mircea Nicolescu [a], Alexei Skurikhin [b]

[a] Computer Vision Laboratory, University of Nevada, Department of Computer Science and Engineering, 171 Reno, NV 89557, USA
[b] MS D436, Space and Remote Sensing Group, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 1 July 2007
Accepted 29 July 2008
Available online 26 August 2008

Keywords:
Segmentation
Boundary detection
Grouping
Object detection
Tensor voting

ABSTRACT

Grouping processes, which "organize" a given data by eliminating the irrelevant items and sorting the rest into groups, each corresponding to a particular object, can provide reliable pre-processed information to higher level computer vision functions, such as object detection and recognition. In this paper, we consider the problem of grouping oriented segments in highly cluttered images. In this context, we have developed a general and powerful method based on an iterative, multiscale tensor voting approach. Segments are represented as second-order tensors and communicate with each other through a voting scheme that incorporates the Gestalt principles of visual perception. The key idea of our approach is removing background segments conservatively on an iterative fashion, using multi-scale analysis, and re-voting on the retained segments. We have performed extensive experiments to evaluate the strengths and weaknesses of our approach using both synthetic and real images from publicly available datasets including the Williams and Thornber's fruit-texture dataset [L. Williams, Fruit and texture images. Available from: <http://www.cs.unm.edu/~williams/saliency.html>, 2008 (last viewed in July 2008)] and the Berkeley segmentation dataset [C.F.P. Arbelaez, D. Martin, The berkeley segmentation dataset and benchmark. Available from: <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>, 2008 (last viewed in July 2008)]. Our results and comparisons indicate that the proposed method improves segmentation results considerably, especially under severe background clutter. In particular, we show that using the iterative multiscale tensor voting approach to post-process the posterior probability map, produced by segmentation methods, improves boundary detection results in 84% of the gray-scale test images in the Berkeley segmentation benchmark.

1. Introduction

Perceptual grouping (or organization) can be defined as the ability to detect organized structures or patterns in the presence of missing and noisy information. It has been proven to be of fundamental importance in computer vision, providing reliable pre-processed information to higher level functions, such as object detection and recognition. Indeed, many low-level vision methods, such as edge labeling [3], rely on perfect segmentation and connectivity, producing undesired results when these assumptions are not valid. Other methods, like shape from contour [4], rely on connected edges, and can benefit from the removal of noise (i.e., erroneous segments). Pattern recognition approaches, such as [5], also rely on connected edges, and usually fail when the edge image is very fragmented. Besides, the complexity of such schemes is directly proportional to the number of distinct primitives in the input. Still, the amount of noise is in general directly proportional to the computational cost of finding true objects in a scene. By using global perceptual organization cues on connecting fragmented edge images can alleviate many of these problems.

Although perceptual grouping ability is present in different biological systems (e.g. visual [6] and auditory [7]), in computer vision it has been simulated using empirical evidence based primarily on research performed by the Gestalt psychologists [8]. Determining organized structures from a given set of points or edges can be a very difficult task, as the actual measurement of compatibility within a sub-set is not well defined. The Gestalt psychologists are considered the first to address the issues of perceptual grouping. Several laws of how grouping might work inside the human mind have been formulated, although their computational implementation turns out to be non-trivial as they lead to conflicting interpretations.

Considering inputs in the form of edges, the Gestalt laws most relevant to computer vision have been related to proximity and good continuation, usually represented in one expression called *saliency*. Conversion of the saliency measure to a prior probability is commonly done, allowing the perceptual grouping problem to be approached using probabilistic techniques [9–11]. Quite

* Corresponding author. Fax: +1 775 784 1877.
E-mail addresses: loss@cse.unr.edu (L. Loss), bebis@cse.unr.edu (G. Bebis), mircea@cse.unr.edu (M. Nicolescu), alexei@lanl.gov (A. Skurikhin).

1077-3142/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2008.07.011

frequently, perceptual grouping has also been tackled as an optimization problem, where the best or most perceptive configuration emerges after searching [3,12–14]. Yet another way of dealing with perceptual grouping is to consider each pixel or edgel as a node in a graph and use a pairwise saliency measure as the strength of the edges of the graph [15–17]. A brief review of representative approaches is presented in Section 2.

The use of a voting process for salient feature inference from sparse and noisy data was introduced by Guy and Medioni [18] and then formalized into a unified tensor voting framework in [19]. Tensor voting represents input data as tensors and interrelates them through voting fields built from a saliency function that incorporates the Gestalt laws of proximity and good continuation. The methodology has been used in 2D for curve and junction detection and for figure completion in [20] and [21]. It has also been applied in 3D for dense reconstruction from stereo [22] or multiple views [23], and for tracking [24]. Examples of higher dimensional voting include the 4D frameworks for epipolar geometry estimation [25] and motion analysis [26], the 8D method for the estimation of the fundamental matrix [27], and the *N*D approach for image repairing [28].

In this paper, we propose a new approach for perceptual grouping of oriented segments in highly cluttered images based on tensor voting. Similar problems have been considered in other studies including [10,15,14]. Specifically, we have developed an iterative tensor voting scheme that removes noisy segments using multi-scale analysis, and re-votes on the retained segments. The proposed approach has been motivated by two observations: (i) structures should reach a maximum saliency when all segments that support them do so and there are no more segments to be added, and (ii) non-salient segments do not exhibit consistent stability over multiple scales.

This paper aims at showing that this process results in better quality segmentations, specially under severe background clutter. In contrast to traditional tensor voting approaches, that use hard thresholding and single-scale analysis, our method removes noisy segments conservatively according to their behavior across a range of scales. Then, it applies re-voting on the remaining segments to estimate their saliency more reliably. It is worth mentioning that multi-scale tensor voting approaches have been proposed before in the literature [29,30]. The main objective of these approaches, however, was to determine an optimal scale for processing. In contrast, our approach performs analysis over the entire range of scales. Moreover, iterative tensor voting schemes have been adopted in [31,32] in order to compute saliency more reliably. However, these iterative scheme differ from the one proposed here in that the role of their iterations was to strengthen salient structures enough to allow a single threshold to segment out clutter; our scheme, on the other hand, removes clutter iteratively.

We have performed extensive experiments and comparisons to test our approach using both synthetic and real images. First, we experimented with a dataset introduced by Williams and Thornber (WT) [1,10]. Although containing real object contours, we consider this a synthetic dataset due to the artificial way the images were created. To make this dataset more challenging and the experiments more complete, we have augmented WT's dataset by incorporating images containing multiple objects having different sizes and incomplete boundaries. The synthetic dataset provides important insight on the method's strengths, allowing us to study special cases that would be difficult to isolate in real, natural images. Second, we experimented with real images from the Berkeley segmentation dataset [2,33] and compared our results to five other methods that are among the top performers for this dataset. The objective of these experiments is to demonstrate the effectiveness of our method, as well as its limitation in real scenarios. Our results indicate that the proposed scheme improves segmentation results considerably, especially under severe background clutter. It is worth mentioning that using the iterative, multiscale tensor voting scheme to post-process the posterior probability maps produced by segmentation methods, improves boundary detection in 84% of the grayscale test images in the Berkeley segmentation dataset. An earlier version of this work, involving detection of single objects with closed boundaries in synthetic images, has appeared in [34].

The rest of the paper is organized as follows: Section 2 provides a review of representative perceptual grouping approaches. Section 3 summarizes the tensor voting framework and discusses the main challenges in applying it for perceptual grouping. Section 4 presents the new approach and provides a number of examples to illustrate the main ideas. Section 5 describes the datasets used in our experiments and our evaluation methodology. Section 6 presents our experimental results and comparisons. Finally, conclusion and directions for future work are presented in Section 7.

2. Perceptual grouping review

Perceptual grouping has been used in computer vision in different contexts and for different applications. We review below a number of representative approaches.

Gestalt principles such as collinearity, co-curvilinearity and simplicity are noted to be important for perceptual grouping by Lowe [12]. Ahuja and Tuceryan [9] were among the first to introduce a method for clustering and grouping of sets of points based on an underlying perceptual pattern. Proximity and good continuation were used as compatibility measures by Dolan and Weiss [3] to the development of a hierarchical grouping approach. Grouping is performed by Mohan and Nevatia [35] based on models of the desired features which are previously computed according to the contents of the scene. In a later work [36], the same authors develop a grouping method based explicitly on symmetries, performing the connectivity steps locally.

Ullman [37] deals with grouping of edge fragments as an optimization problem which suggests that the smoothest line joining every pair of fragments should minimize the integral of the square of the curvature. Although there is clearly a intuitive idea behind this approach, one can note that elliptical curves, for example, cannot be constructed by joining only a pair of circular arcs. Also, as Guy and Medioni noted [38], this scheme cannot be promptly generalized to a set of three or more edge fragments, and does not allow for outliers. The tensor voting framework used in this work is in essence an extension of the idea above where otherwise a curve may be formed (and/or approximated) by joining an unlimited number of (possibly) short circular arcs, and outliers are dealt naturally. Parent and Zucker [39] proposed a relaxation labeling scheme that utilizes local kernels incorporating co-circularity measures used to estimate tangent and curvature. Very similar kernels are used in the tensor voting framework, but applied in a different way. A saliency measure is proposed by Ullman and Shashua [15] to guide the grouping process and eliminate erroneous features in the image. Their scheme tends to give preference to long curves with low total curvature.

Hérault and Horaud [14] tackled the problem of segmenting oriented edges into figure and ground as a quadratic programming problem, solved by simulated annealing. Saliency was defined as a function of proximity, contrast, co-circularity and smoothness. An optimization step searches for the configuration of image edgels that leads to the highest interactivity between elements while minimizing an objective function which has two terms, one that accounts for the total saliency of the edgel configuration, and another one that prevents trivial solutions, such as all edgels selected. The latter one, although it is said to be related to the signal-to-noise ratio (SNR), was not explained how to be computed

and, in practice, it is very sensitive. Sarkar and Boyer [16] make use of a saliency measure that includes, in addition to proximity and good continuation, parallelism and perpendicularity in order to assess man-made land development from aerial images. Clustering is done by computing the eigen-decomposition of an affinity matrix composed of pairwise saliency measures.

Recently, Williams and Thornber [10] have proposed a probabilistic approach based on *Closed Random Walks* (CRWs). In their approach, saliency was defined relatively to the number of times an edge is visited by a particle in a random walk. The main restriction assumed in their work is that the movement has to start and finish on the same edge. This reduces the number of paths to consider along with the complexity of the problem, however, it imposes a restriction that is not practical. For example, objects in real images are not expected to be closed or well formed, due to occlusions and pre-processing artifacts. Their technique was compared to five other methods in the literature and found to outperform them. Mahamud et al. [11] generalized the CRW technique to deal with multiple salient contours, but still closed.

Summarizing the main features of the methods above and contrasting them to the tensor voting framework, it is interesting to note that virtually all of them use local operators to infer a more global structure. Also, many of them are inherently iterative, relying on optimization techniques (e.g., relaxation or minimization), which are sensitive on initialization and are subject to instabilities. The main difference among these methods is in the choice of the compatibility measures employed or the function to be minimized.

3. Perceptual grouping using tensor voting

3.1. Tensor voting framework

In the framework proposed by Medioni et al. [19], input data are encoded as elementary tensors. Support information (including proximity and smoothness) is propagated from tensor to tensor by vote casting. Tensors that lie on salient features (i.e., curves in 2D, or curves and surfaces in 3D) strongly support each other and deform according to the prevailing orientation, producing generic tensors. Each such tensor encodes the local orientation of features, given by the tensor orientation, and their saliency, given by the tensor shape and size. Features can then be extracted by examining the tensors resulting from voting.

Fig. 1 illustrates the voting process for the extraction of salient curves from a noisy set of 2D points. The input points (Fig. 1a) are initially encoded as ball tensors, equivalent to circles in 2D, as shown in Fig. 1b. The voting process allows tensors to propagate their position information in a neighborhood, such that, (i) tensors that lie on a salient curve strongly reinforce each other and deform according to the prevailing orientation (normal to the curve), and (ii) isolated tensors receive little support, as they do not correspond to any underlying salient curve, and therefore can be identified as noise (see Fig. 1c).

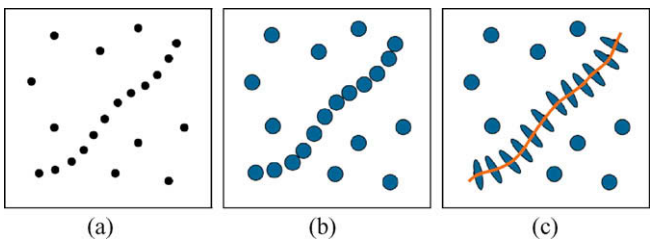


Fig. 1. Tensor voting example: (a) input points, (b) ball tensor encoding, (c) deformation of tensors reveals the salient curve.

(1) *Tensor representation and voting.* In 2D, a generic tensor can be visualized as an ellipse. It is described by a 2×2 eigensystem, where eigenvectors e_1, e_2 give the ellipsoid orientation and eigenvalues λ_1, λ_2 (with $\lambda_1 \geqslant \lambda_2$ give its shape and size. The tensor is represented as a matrix S:

$$S = \lambda_1 \cdot e_1 e_1^{\mathrm{T}} + \lambda_2 \cdot e_2 e_2^{\mathrm{T}} \tag{1}$$

There are two types of features in 2D—curves and points (junctions)—that correspond to two elementary tensors. A curve element can be intuitively encoded as a *stick tensor* where one dimension dominates (i.e., along the curve normal), while the length of the stick represents the curve saliency (i.e., confidence in this knowledge). A point element appears as a *ball tensor* where no dimension dominates, showing no preference for any particular orientation.

Input tokens are encoded as such elementary tensors. A point element is encoded as a ball tensor, with e_1, e_2 being any orthonormal basis, while $\lambda_1 = \lambda_2 = 1$. A curve element is encoded as a stick tensor, with e_1 being normal to the curve, while $\lambda_1 = 1$ and $\lambda_2 = 0$. Tokens communicate through a voting process, where each token casts a vote at each token in its neighborhood. The size and shape of this neighborhood, and the vote strength and orientation are encapsulated in predefined voting fields (kernels), one for each feature type—there is a stick voting field and a ball voting field in the 2D case. Revisiting the example in Fig. 1, note that the input was encoded as ball tensors. However, if some orientation information is initially known (e.g., from edge detection), the input can be simply encoded using stick tensors.

At each receiving site, the collected votes are combined through simple tensor addition, producing generic tensors that reflect the saliency and orientation of the underlying salient features. Local features can be extracted by examining the properties of a generic tensor, which can be decomposed in its stick and ball components:

$$S = (\lambda_1 - \lambda_2) \cdot e_1 e_1^{\mathrm{T}} + \lambda_2 \cdot (e_1 e_1^{\mathrm{T}} + e_2 e_2^{\mathrm{T}}) \tag{2}$$

Each type of feature can be characterized as: (a) *Curve*—saliency is $(\lambda_1 - \lambda_2)$ and orientation is e_1, and (b) *Point*—saliency is λ_2 with no preferred orientation. After voting, curve elements can be identified as they have a large curve saliency $\lambda_1 - \lambda_2$ (appear as elongated tensors), junction points have a large point saliency λ_2 and no preferred orientation (appear as large ball tensors), while noisy points have low point saliency. Therefore, the voting process infers curves and junctions simultaneously, while at the same time identifying outliers, that is, tokens with little support. The method is robust to considerable amounts of outlier noise and does not depend on critical thresholds, the only free parameter being the scale factor σ which defines the voting fields.

(2) *Vote generation.* The *vote strength* $\mathrm{VS}(\vec{d})$ decays with the distance $|\vec{d}|$ between voter and recipient, and with the curvature ρ:

$$VS(\vec{d}) = \exp\left(-\frac{|\vec{d}|^2 + c \cdot \rho^2}{\sigma^2}\right) \tag{3}$$

where c is a constant regulating the relative effects of distance and curvature. The *vote orientation* corresponds to the smoothest local continuation from voter to recipient (see Fig. 2). A tensor P with locally known curve information, illustrated by curve normal $\vec{N}_p$, casts a vote at its neighbor Q. The vote orientation is chosen to ensure a smooth curve continuation through a circular arc from voter P to recipient Q. To propagate the curve normal $\vec{N}$ thus obtained, the vote $V_{\mathrm{stick}}(\vec{d})$ sent from P to Q is encoded as a tensor according to:

$$V_{\mathrm{stick}}(\vec{d}) = VS(\vec{d}) \cdot \vec{N}\vec{N}^{\mathrm{T}} \tag{4}$$

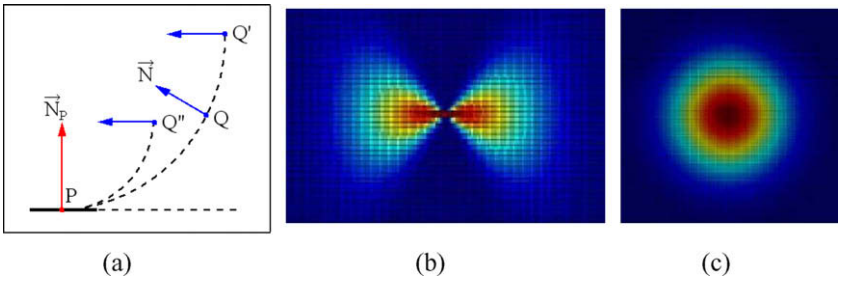


Fig. 2. Vote generation in 2D: (a) decay function used by tensor voting framework, (b) stick voting field, and (c) ball voting field.

It should be noted that, the vote strength at Q' and Q'' is smaller than at Q due to the fact that Q' is farther away and Q'' corresponds to a higher curvature than Q. Fig. 2b shows the 2D stick field, with its color-coded strength. When the voter is a ball tensor, with no information known locally, the vote is generated by rotating a stick vote in the 2D plane and integrating all contributions according to Eq. 5. The 2D ball field is shown in Fig. 2c.

$$V_{\text{ball}}(\vec{d}) = \int_0^{2\pi} R_\theta V_{\text{stick}}(R_\theta^{-1}\vec{d})R_\theta^{\mathrm{T}}\,\mathrm{d}\theta \tag{5}$$

Table 1 shows a summary of the geometric features that appear in a 2D space and their representation as *elementary ID* tensors, where n and t represent the normal and tangent vector respectively. From a *generic* 2D tensor that results after voting, the geometric features are extracted as shown in Table 2. The framework can be readily extended to higher dimensions, for example, in 3D the features are points, curves or surfaces, corresponding to ball, plate, or stick tensors, all expressed as 3×3 eigen-systems.

The space complexity of the voting process is $\mathbf{O}(n)$, where n is the input size (i.e., total number of candidate tokens). The average time complexity is $\mathbf{O}(mn)$ where m is the average number of candidate tokens in the neighborhood. Therefore, in contrast to other voting techniques, such as the Hough Transform, both time and space complexities of the tensor voting methodology are *independent* of the dimensionality of the desired feature.

3.2. Grouping using tensor voting

Although the tensor voting framework has only one free parameter, the scale σ, several other issues must be considered carefully when employing it for perceptual grouping and segmentation. The voting dimensionality, the features to be used as tokens, and the encoding of the input tokens are important issues that need consideration.

The voting dimensionality is determined by the number of features used to represent the problem, and influences directly the performance and quality of the results. Ideally, the fewest number of features with maximal representation capability is desired, as it is more likely to produce the best results within the shortest time. This raises the issue of what features to use. First, the features chosen must be in the Euclidean space, or at least be scaled to, so that the decay function which establishes the vote strength is a valid one, as suggested in [40]. Pixel coordinates, edgel orientation and gradient, are examples of features commonly used for raster images or their edge-based counterparts.

Table 1
Elementary tensors in 2-D

Feature	λ_1	λ_2	e_1e_2	Tensor
Point	1	1	Any orthonormal basis	Ball
Curve	1	0	n t	Stick

Table 2
Elementary tensors in 2-D

Feature	Saliency	Normal	Tangent
Point	λ_2	None	None
Curve	$\lambda_1 - \lambda_2$	e1	e_2

Token encoding has considerable impact on the performance of tensor voting. It was mentioned earlier that an input token can be initialized either as a ball or a stick tensor in 2D. The benefits of using stick tensors instead of ball tensors in 2D, can be easily understood by comparing the voting fields of Fig. 2b and c. Stick voting fields cover smaller regions and, in general, require fewer vote castings than ball voting fields, allowing faster computations. Although this choice is not extremely critical in the voting results, stick encoding allows the introduction of prior knowledge in terms of the tokens' preferred direction (e.g., edgel orientation) and should be used whenever it is possible.

In the case of edges, one can choose among several different tensor representations as shown in Fig. 3. One way would be assigning a ball tensor to each pixel of the edge contour as shown in Fig. 3b. Alternatively, one could assign a stick tensor to each pixel with position and orientation determined the pixel and its adjacent neighbors (see Fig. 3c). The main disadvantage of the above representations is that they lead to a large number of tensors, increasing computational requirements. Alternatively, one could choose a subset of representative pixels along the edge contour and initialize them as ball or stick tensors (see Fig. 3d). This would lead to a more economical representation and lower computational requirements.

We have adopted this last approach in our study. Using the middle and/or end pixels along the edge contour can yield good support for short edge segments, however, this choice would not work well for long edge segments since the distance between tokens plays an important role in the voting process. Here, we propose re-sampling the edge contour into a number of equidistant points using a fixed sampling step. Then, we initialize the tensor voting framework by encoding sampled points as stick tensors with position and orientation determined by the position and gradient information of the sampled points.

Another issue that needs consideration is the selection of the scale parameter σ. In [40], it was found that tensor voting has low sensitivity with respect to σ. However, finding the appropriate σ value might not be easy in practice. It is well known that small scales capture local structures while large scales capture global configurations. In a real scenario, it is unlikely that we would have any *a-priori* information about the size of objects in the scene, making the choice of σ a "trial-and-error" process. In general, the choice of the scale parameter will vary from application to application, or even worse, from image to image.

Analyzing information at a single scale can compromise or make hard the detection of structures with different sizes. This sit-

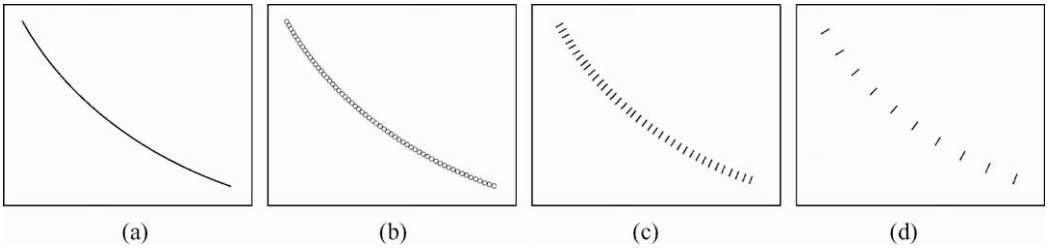


Fig. 3. Various tensor initializations using edge contours: (a) edge contour, (b) each pixel on the edge contour could be considered to be a token and initialized as a ball tensor, (c) each pixel on the edge contour could be considered to be a token and initialized as a stick tensor tangent to the curve, (d) a subset of the edge pixels, obtained through subsampling, could be considered to be tokens and initialized as stick tensors tangent to the contour.

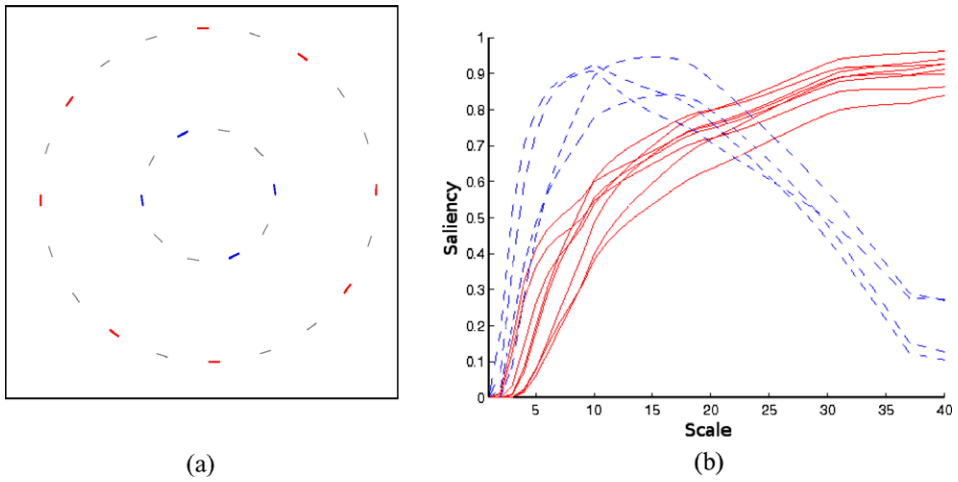


Fig. 4. (a) Two circles with different sizes and few segments highlighted, (b) normalized saliency curves corresponding to the segments selected (dashed for smaller circle). The saliency of the smaller circle increases until the voting neighborhood contains all of its segments. After this point, it is surpassed by the saliency of the larger circle, which keeps increasing until it reaches its own maximum.

uation can be illustrated using an image containing two similar figures, one smaller than the other, as shown in Fig. 4. To help visualization, we have plotted "Scale versus Saliency" curves, thereafter called *saliency curves.* Specifically, a saliency curve is computed by voting in different scales and computing the saliency of each segment in each scale. We then normalize the saliency curves according to the average saliency of all segments in the image in order to prevent a monotonically increasing curve. This is due to the fact that, as the voting neighborhood increases, segment saliency also increases simply because new segments are considered.

As the voting neighborhood increases, the smaller circle starts becoming more salient since more of its segments are considered in the voting process. Its saliency maximum is reached when the voting neighborhood contains all its segments, (i.e., at around $\sigma = 10$). After this point, not having any more segments to strengthen its saliency, the smaller circle starts"losing" saliency for the larger one, which becomes more salient as more of its segments are included in the voting neighborhood. Once the larger circle reaches its maximum saliency, at around $\sigma = 35$, its saliency curves stabilize since there are no more segments to consider beyond this scale.

Another important issue when segmenting a figure from the background is the choice of a threshold for filtering out non-figure segments. It is reasonable to expect that if the saliency values of the figure are quite higher than those of the background, then it would be easy to find a threshold value that separates them completely. Fig. 5 shows a simple example where we consider a well-formed circle surrounded by random noise at SNR = 70%. By applying tensor voting and observing its saliency histogram shown in Fig. 5b, it becomes evident that by eliminating segments with a saliency value below a threshold T = 45%, all noisy segments are filtered out while all figure segments are preserved (see Fig. 5c).

However, this is hardly the case in practice. Let us consider the image shown in Fig. 6a. Applying tensor voting to the original image and plotting the corresponding saliency curves (Fig. 6b) (only curves that overlap are shown) and saliency histogram (Fig. 6c), we can easily conclude that there is no threshold value able to provide a perfect figure-background segmentation. Although the saliency histogram shown in Fig. 6c corresponds to one, high scale, the same happens at different scales as well. Moreover, even if we were able to choose an optimal threshold in some way, the number of misclassified segments would be unavoidably large as shown in Fig. 6d–f.

4. Iterative multi-scale tensor voting

The example of Fig. 6 illustrates that a high threshold value could eliminate parts of the figure while a low threshold value could preserve too many background segments, leading to poor segmentation results in both cases. Aiming at eliminating the largest number of background segments while preserving as many figure ones as possible, we have developed an iterative tensor voting scheme based on multi-scale analysis and re-voting. The key idea is conservatively removing segments from the image in an iterative fashion, and applying re-voting on the remaining segments to estimate saliency information more reliably. Improvements in figure segmentation come from two facts: (i) after each iteration, low sal-

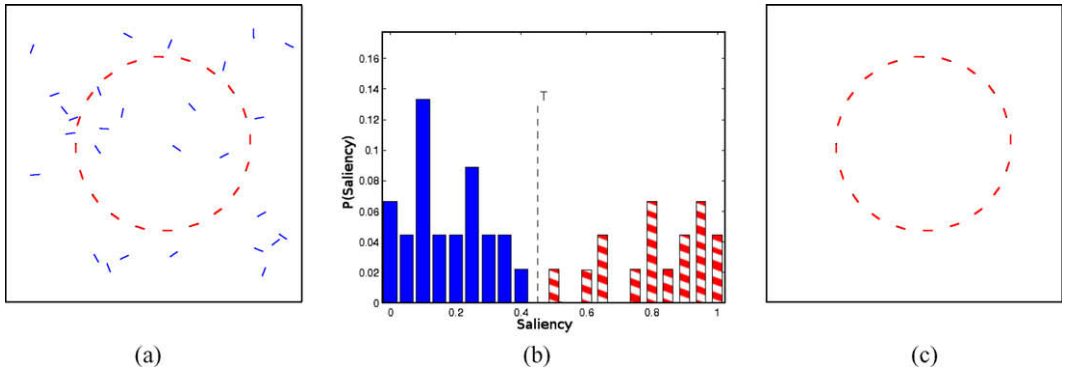


Fig. 5. A simple example where figure and background can be separated easily using a single threshold: (a) original image, (b) saliency histogram (striped for figure) and the optimal threshold T, (c) resulting segmentation.

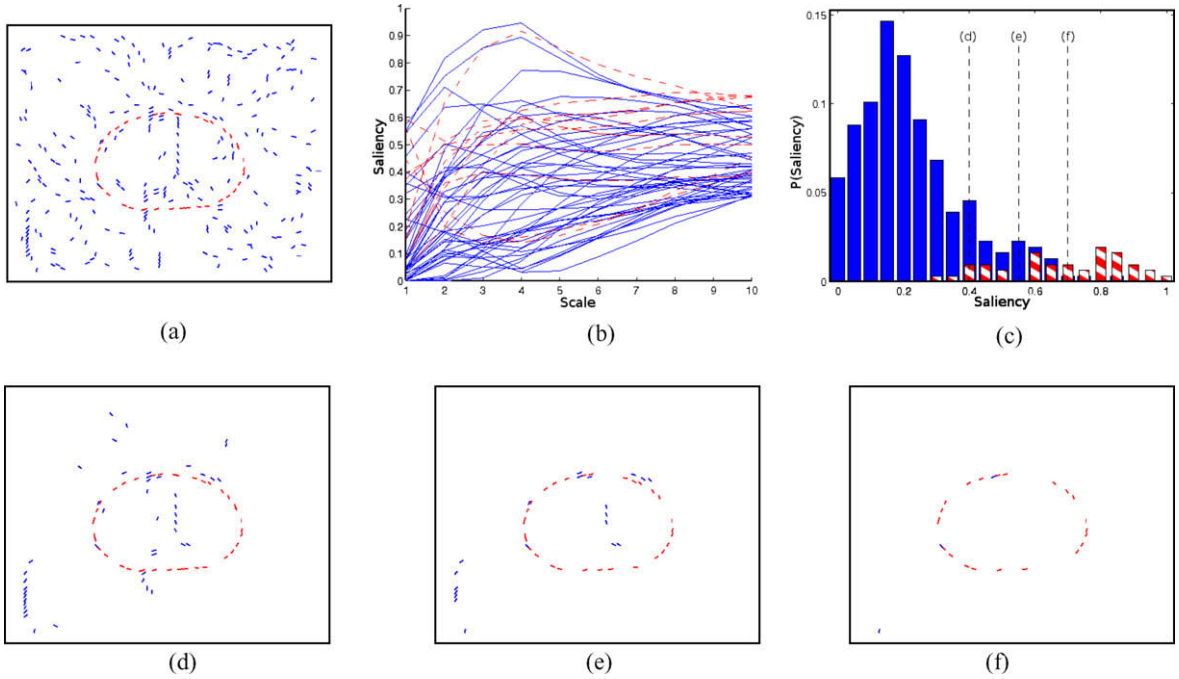


Fig. 6. An image with SNR = 15% processed by different threshold values. A unique, fixed threshold value (T) cannot produce a good segmentation at any scale, (a) original image, (b) overlapping saliency curves corresponding to segments of the figure (dashed) and the background, (c) saliency histogram (striped for figure) and three threshold choices: (d) $T = 40\%$, (e) $T = 55\%$, and (f) $T = 70\%$.

iency segments are filtered out and, (ii) after the subsequent re-voting steps, background segments get less and less support. Fig. 7 illustrates this idea using the example shown in Fig. 6. As more and more background segments are eliminated, the saliency difference between figure and background segments becomes more and more pronounced.

From an implementation point of view, the conservative elimination of low saliency segments is performed by applying a low threshold T_s, which, in most cases, removes background segments only. In the next iteration, a new saliency map is obtained using re-voting, without considering the eliminated segments this time. After re-voting, the threshold value is increased to adapt to the strengthening of figure saliency due to the elimination of background segments. In practice, we slightly increase T_s after each re-voting session by a fixed amount ΔT_S.

Multi-scale analysis is incorporated to this scheme by voting in a number of scales and thresholding according to the behavior of saliency in these scales. The key idea is that non-salient segments do not exhibit consistent stability over multiple scales, an idea motivated by scale-space theory [41]. Specifically, the saliency curve of a segment is computed by voting indifferent scales and computing the saliency of that segment in each scale. Segments are then eliminated if they do not present any significant saliency peaks across a range of scales. This will preserves salient segments of any size. Algorithmically, this is implemented by counting the number of scales that the saliency curve stays above the threshold T_s. If this number does not exceed another threshold T_σ, then we consider that the corresponding segment does not have strong saliency and it is eliminated. Fig. 8 illustrates this procedure. As mentioned in the previous section, we normalize the saliency curves according to the average saliency of all the segments in the image.

Below, we present the pseudo-code of the iterative, multiscale tensor voting scheme. The input to the algorithm are the number of iterations I, number of scales K, and the size of the input image (i.e., width W_{img} and height H_{img}). ΔT_s is the amount by which T_s is incremented in each iteration to account for stronger saliencies due to the formation of more organized structures as clutter is eliminated (see Fig. 9).

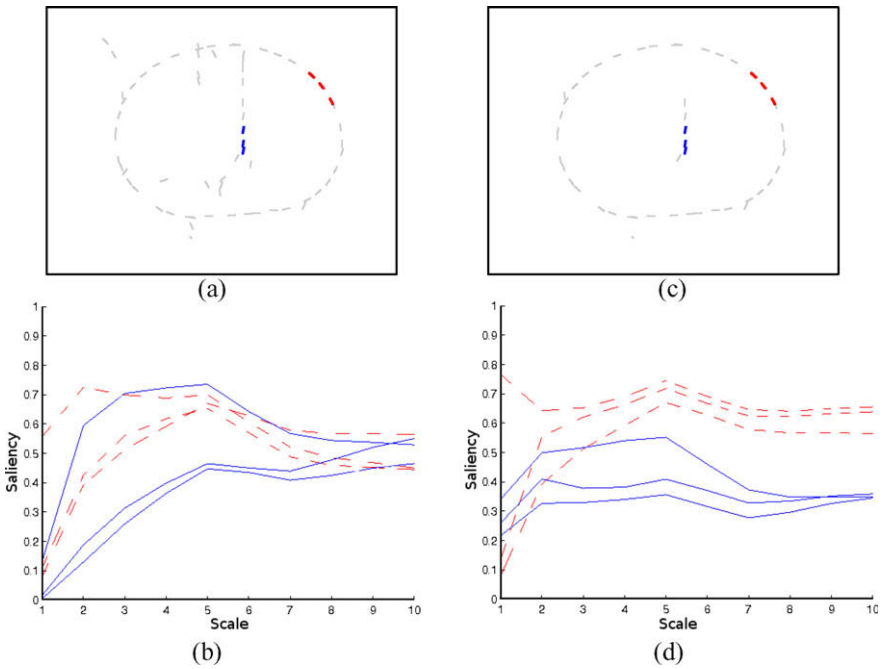


Fig. 7. Conservative elimination of segments improves discrimination between figure and background segments after re-voting: (a) image with a few segments selected from ground and figure, (b) saliency curves (dashed for figure) for selected segments showing overlap in various scales, (c) image after conservative thresholding which eliminates some spurious segments, (d) saliency curves (dashed for figure) after re-voting showing better separation between figure and background segments.

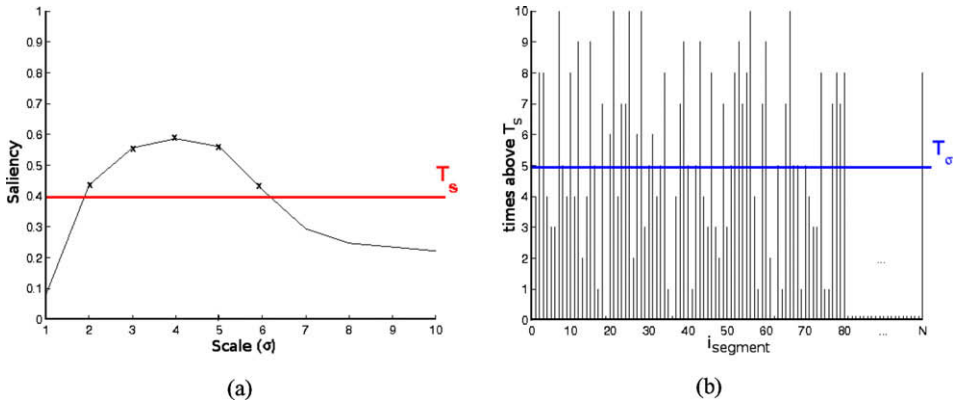


Fig. 8. Illustration of thresholds T_s and T_σ: (a) the number of times a saliency curve is above T_s is computed, (b) segments whose saliency curves do not reach a number of times more than T_σ are eliminated.

1. Initialize I, K, T_s, T_σ and ΔT_s
2. Set i $\leftarrow$0, m $\leftarrow$ max $\{H_{img}, W_{img}\}$, and $\sigma_j \leftarrow \frac{j \times m}{K}, j = 1, 2, \ldots, K$
3. While i less than I:
   3.1. Apply tensor voting at scales $\sigma \leftarrow \sigma_1, \sigma_2, \ldots, \sigma_K$
   3.2. Eliminate segments with saliency below T_s more than T_σ times
   3.3. $T_S \leftarrow T_S + \Delta T_s$
   3.4. $i \leftarrow i + 1$

The iterative multiscale voting scheme can be implemented efficiently without requiring to compute the votes from in a brute-force manner at each iteration or at each scale. Specifically, the votes at iteration i can be computed from the votes at iteration $i-1$ by simply subtracting the votes cast at iteration $i-1$ by the low saliency segments eliminated at iteration i. Similarly, the votes at a given scale σ_j can be computed from the votes at the immediate lower scale σ_{j-1}. Since the voting neighborhood increases as the scale increases, we need to compute and add only votes corresponding to segments that lie in area corresponding to the difference between the two neighborhoods.

The complexity of the iterative scheme is asymptotically the same to the complexity of the original tensor voting scheme at a fixed scale. Specifically, let us assume that there are N segments in the image and M of them are contained in the voting neighborhood for a given fixed scale; then, the complexity of voting is $O(NM)$ or $O(N^2)$ since $M = O(N)$. In the case of iterative voting, we perform I iterations and vote at K different scales in each iteration. The complexity of voting at each scale σ_j is $O(NM_j)$ where $j = 1, 2, \ldots, K$ and M_j is the number of segments contained in the difference of the neighborhoods corresponding to σ_j and σ_{j-1}. Since $M_j = O(N)$, and $K = O(1)$, the complexity at each iteration would be $O(N^2)$. The overall complexity would be $O(N^2)$ since $I = O(1)$.

Fig. 9 shows the behavior of figure (dashed) and background saliency curves during different iterations of the proposed approach. The input image has SNR = 15% (i.e., about 7 times more

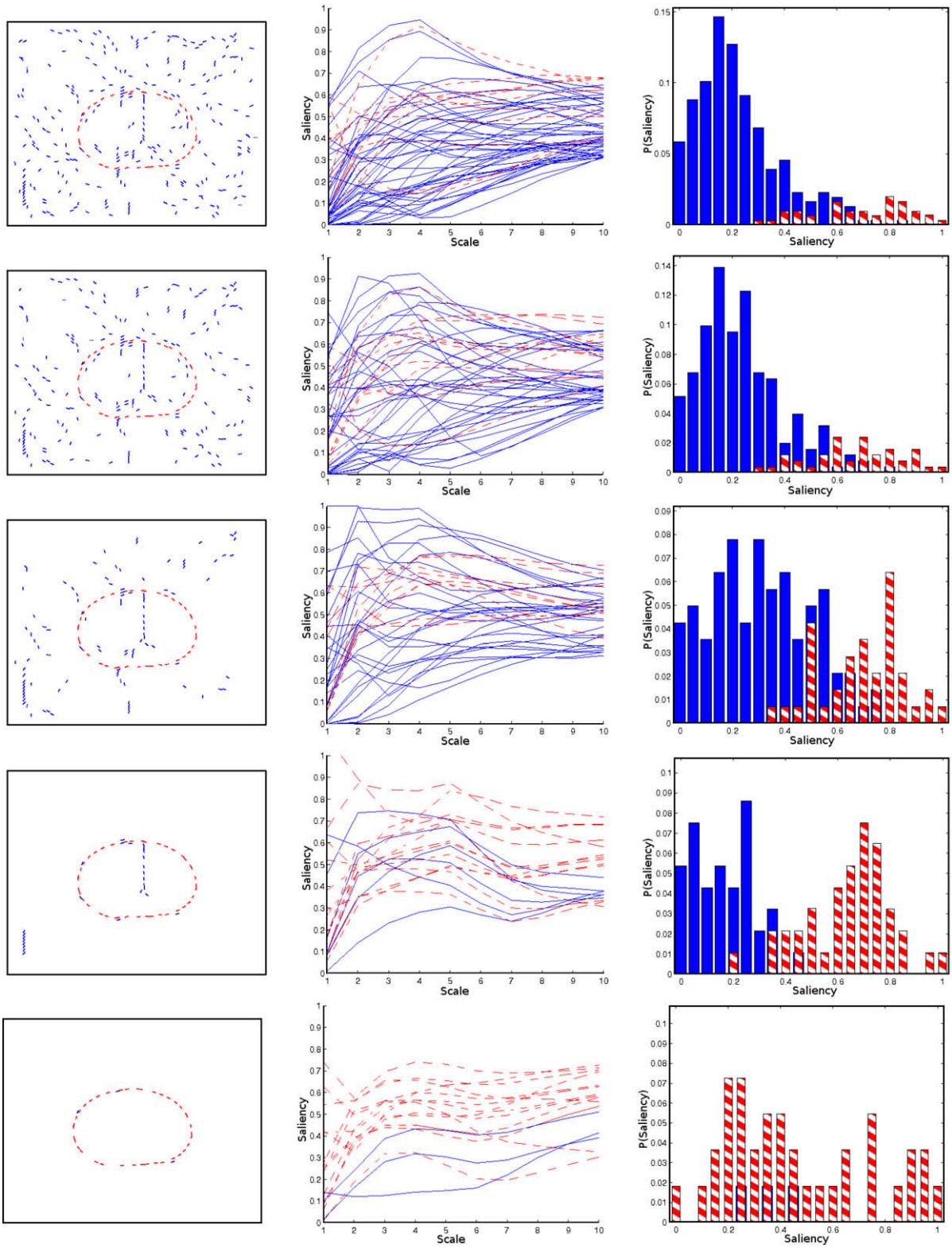


Fig. 9. Image with 15% SNR processed by our iterative, multi-scale tensor voting scheme. By conservatively eliminating low saliency segments, the saliency difference between figure (dashed) and background segments becomes more and more pronounced. Each column shows: (i) resulting image, (ii) saliency curves of segments in the ambiguity region, and (iii) saliency histogram at the highest scale. By row: First—Original image. Second—Resulting image using $T_s = 10\%$. Third—Resulting image using $T_s = 20\%$. Fourth—Resulting image using $T_s = 30\%$. Fifth—Resulting image using $T_s = 40\%$.

background segments than figure ones). The threshold value T_s goes from 10 up to 40 with a $\Delta T_s = 10\%$. The voting was performed with a σ ranging from 1 (5% of image size) to 20 (100% of image size). It should be mentioned that we experimented with different ΔT_s values or numbers of scales, however, we did not notice significant differences in our results except when using a rather big ΔT_s value or a rather small number of scales. The improvements over using the naive approach (i.e., fixed threshold and single

scale—see Fig. 6) are remarkable. A quantitative comparison can reveal the benefits of the proposed scheme. In Fig. 6, using $T = 55\%$ (Fig. 6e), 10 out of 40 figure segments were eliminated (FN rate equal to 25%) and 19 out of 270 ground segments were not filtered out (FP rate equal to 7%). In contrast, our methodology eliminated 2 out of 40 figure segments (FN rate equal to 5%) and did not filter out 3 out of 270 ground segments (FP rate equal to 1%).

5. Datasets and evaluation methodology

We have divided our experiments in two parts. In the first part, we have performed a series of experiments using synthetic images based on the set of fruit and texture sampled silhouettes used in [10]. The objective of this set of experiments is to consider different figure-ground configurations in order to get important insight on the method's strengths, allowing us to study special cases that would be difficult to isolate in real, natural images. The second part reports test results on the Berkeley segmentation dataset and benchmark [33]. The objective of this set of experiments is to demonstrate the effectiveness of our method, as well as its limitation in real scenarios.

Part I of our experiments was performed with synthetic images created from a pair of sampled silhouettes belonging to a fruit or a vegetable (thereafter called figure) and textured background (thereafter called background). Nine figure silhouettes were re-scaled to an absolute size of 32×32 and placed in the middle of nine 64×64 re-scaled background windows. We have experimented with five different SNR values in order to reduce the number of figure segments proportionally to the number of background segments. Further details regarding this benchmark can be found in [10]. The images used to build the benchmark are shown in Fig. 10. Fig. 11 shows some examples of benchmark images for different SNRs.

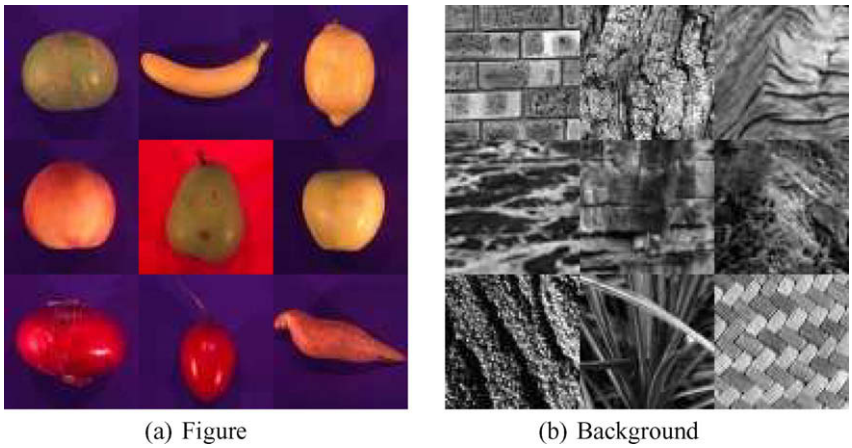
This set of images offers a good synthetic dataset for experimentation and comparison purposes. It is composed of real objects in real backgrounds which is more challenging than images containing a random background which is typically used. Nevertheless, since the objects have always a closed contour and placed in the same position and scale, this dataset lacks realistic characteristics that would make it more challenging.

We have augmented WT's dataset by using the same objects and backgrounds, however, we have incorporated new characteristics in order to make it more realistic. In particular, we have created more test images by varying the number of figures and their size, and by removing parts of their boundary, opening their silhouette. Fig. 12 shows some examples from the extended benchmark. Table 3 summarizes the different datasets used in our experiments. Note that for the datasets with more than one figure, only one SNR was used since the number of background segments in WT's was limited. It is worth mentioning that in Williams and Thornber's evaluations [10], different algorithms were tested by comparing the set of N most salient segments returned by each algorithm, where N is the number of foreground segments. Our algorithm makes a decision on each segment without assuming knowledge of N.

In this part, quantitative evaluations and comparisons with other methods were performed using Receiver Operational Characteristic (ROC) curves (i.e., False Positives (FP) versus False Negatives (FN) plots). A FN is a figure segment detected as background while a FP is a background segment detected as figure. For each dataset, the ROC curves are average ROC curves over all the images in the dataset. In order to allow a direct comparison with WT's method [10], we also show SNR vs FP and SNR vs FN plots.

We have also performed additional experiments using the Berkeley Segmentation dataset and benchmark [2], [33]. In order to evaluate the contribution of our method in real boundary detection and segmentation scenarios, we used our method to post-process the Boundary Posterior Probability (BPP) map produced by five different segmentation methods from the Berkeley segmentation benchmark: Brightness Gradient (BG), Gradient Magnitude (GM), Multi-Scale Gradient Magnitude (MGM), Texture Gradient (TG), and Brightness/Texture Gradients (BTG). Thresholding the BPP map yields a set of boundaries in an image. The output of our method is a new BPP map which is computed by counting the number of iterations each pixel survived the elimination process. The longer a pixel is conserved, the higher is its probability to belong to an organized structure in the image. For evaluation, we used the gray-scale test images and the corresponding BPP maps from the Berkeley segmentation benchmark. Pixels in the BPP map were encoded as tensors whose size was given by the BPP intensity and direction by the normal to the edge direction crossing the pixel.

To quantify boundary detection results, we used Precision-Recall Curves (PRCs) like in the Berkeley segmentation benchmark. PRCs reflect the trade-off between true boundary pixels detected and non-boundary pixels detected at a given threshold. It should be mentioned, however, that all comparisons in the Berkeley benchmark were carried out using the F-measure [42], which is a weighted harmonic mean of precision (P) and recall (R): $F = PR/(\alpha R + (1 - \alpha)P)$ where (a) is a weight. The value of α was set to .5



(a) Figure (b) Background

Fig. 10. Images used to build the benchmark (publicly available at [1]).

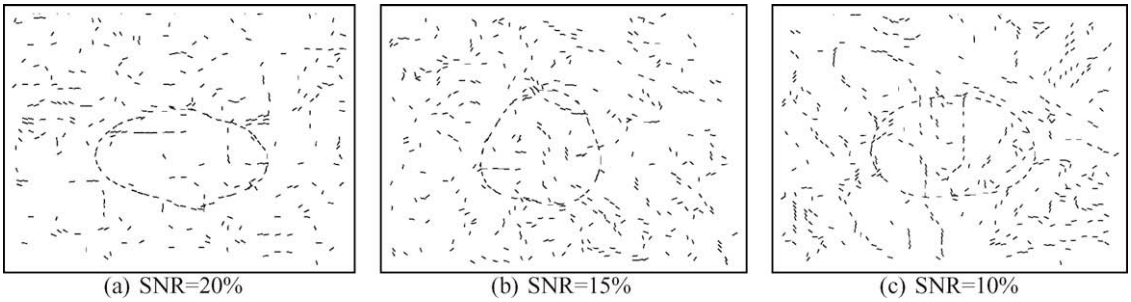


Fig. 11. Examples of benchmark images from [10] at different SNRs.

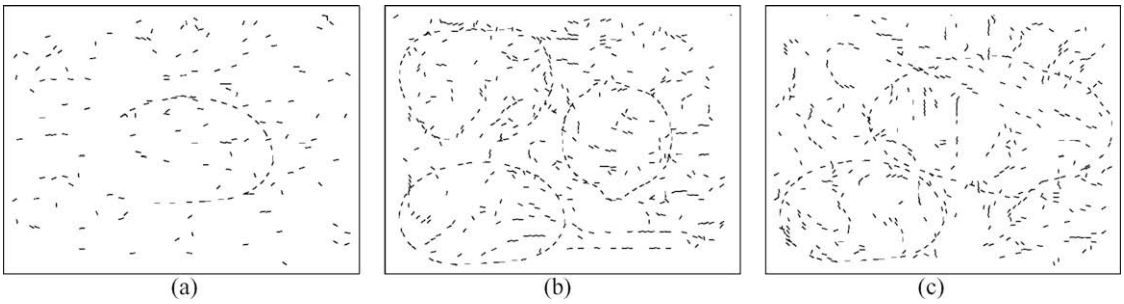


Fig. 12. Examples from the extended benchmark: (a) open figure contour, (b) multiple figures, (c) multiple instances of the same figure with different sizes.

Table 3
Different datasets built from 9 objects, 9 backgrounds and 5 SNRs

Dataset	Images	Characteristics	SNR
Single figure	405	One object, one background	25–5%
Incomplete figure contour	405	One object, one background	25–5%
Multiple figures	1458	Two or three objects, one background	25%
Figures with different size	1458	Two objects, one background	25%

in [33] which is usually called the *equal regime*. Different values of (α) allow for different regimes (e.g., *high precision regime* for $\alpha > .5$, or *high recall regime* for $\alpha < .5$).

To avoid any bias towards a specific regime and evaluate overall performance more objectively, we have also computed the Area Above the precision-recall curve (AAC) in our experiments. The use of AAC's dual, the Area Under a Curve (AUC), has been investigated in other studies (e.g., [43]), suggesting that AUC is a better measure for evaluating overall performance instead of using a single measurement on the curve. In our case, our objective is minimizing AAC in order to improve both precision and recall rates.

A BPP map can be visualized as an image whose pixel intensity encodes the probability that a pixel lies on a boundary. The higher the pixel intensity, the higher the probability that the pixel lies on a boundary. Fig. 13b–f show the BPP map computed by each of these methods for the images in Fig. 13a. The ground truth obtained by five human subjects is shown in Fig. 13f. All five methods above have been previously evaluated on the Berkeley dataset and represent some of the top performers. The BPP maps, specific results and ranking information for each method are publicly available from the Berkeley benchmark website [2].

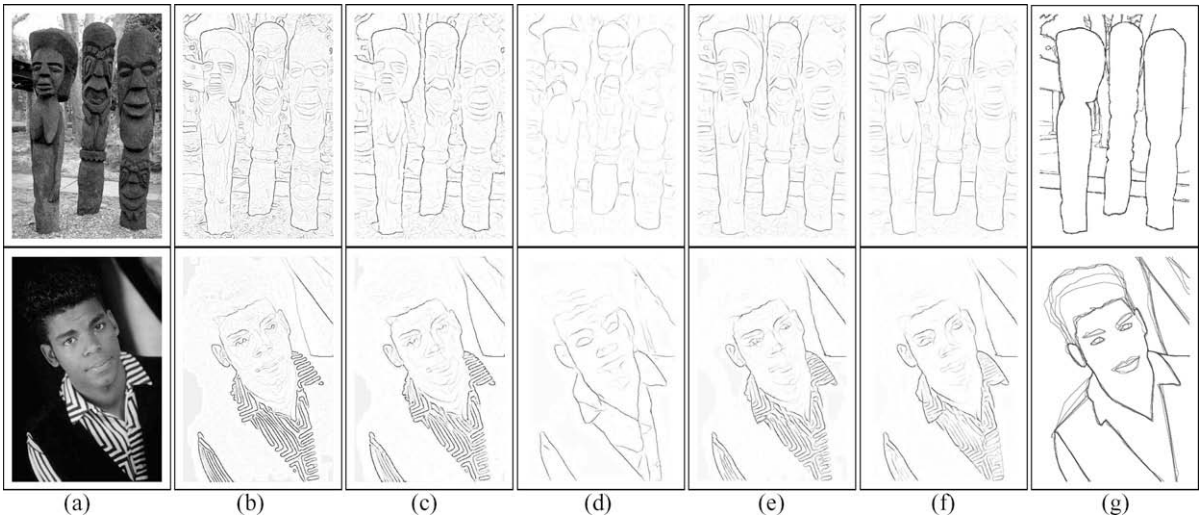


Fig. 13. The BPP map computed by the methods tested in our study: (a) original image, (b) GM BPP map, (c) MGM BPP map, (d) TG BPP map, (e) BG BPP map, (f) BGT BPP map, (g) ground truth.

6. Experimental results and comparisons

6.1. Part I: Experiments on synthetic images

We have performed extensive experiments in order to evaluate our methodology using the datasets discussed in Section 5. Analysis of the saliency histograms is provided so that the behavior of segments belonging to figure and background can be better understood. Comparisons between the naive approach, referred as single-scale, fixed threshold (SSF-T), and the iterative, multi-scale threshold (IMS-T) approach are shown for all datasets. In addition, we have included a direct comparison between our method and WT's method using the original dataset.

6.1.1. Influence of the signal-to-noise ratio

Saliency histograms were plotted for the different SNR values used in [10] (see Fig. 14). For each histogram, we used 81 images (9 figures and 9 backgrounds). It can be observed that, as SNR decreases, figure (red) and background (blue) histograms start overlapping more and more until they become indistinguishable. The larger the overlap between figure and background histograms, the harder is to visually separate the figures from the background. This observation agrees with the visual perception of the objects in the image, as can be seen in Fig. 15. At some point, for instance, when SNR is below 10%, the structures of the background are visually more distinguishable than the figure itself. This effect is mainly due to the use of textures (i.e., leaves, bricks, etc) as background instead of random noise.

Fig. 16a shows the ROC curves obtained using SSF-T. The scale was chosen based on knowledge of the benchmark images (i.e., σ was set equal to 20, yielding a voting field that covers the entire image). When SNR is below 10%, the perception of the figure becomes more difficult. The worst performance is for SNR=10% and SNR=5%. Fig. 16b shows the ROC curves obtained using IMS-T. The scale parameter σ varies from 2 to 20 (covering from 5% to 100% of the image), ΔT_s was equal to 5%, and T_a was equal to 50% (i.e., the saliency curve must be above T_s in at least half of the processed scales). This allows structures to pop out in any region of the scale range. Significant improvements can be noted by comparing Fig. 16b to a. In addition, the curve corresponding

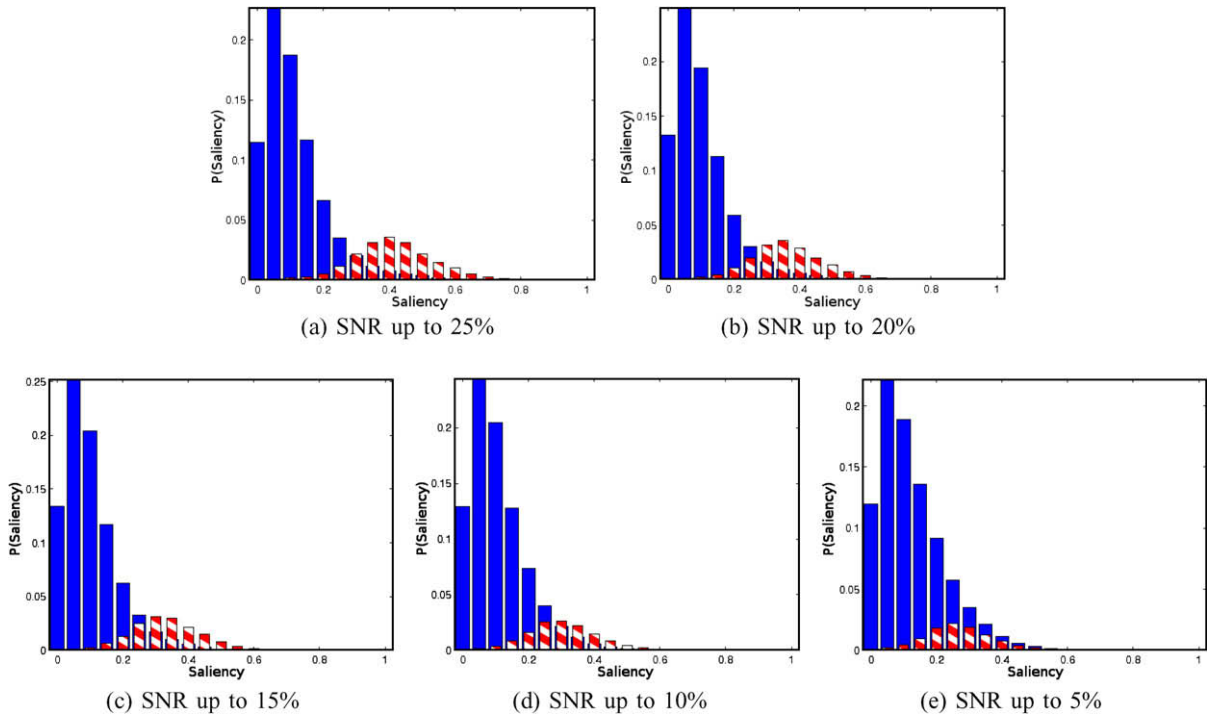


Fig. 14. Saliency histograms assuming various SNR values (striped for figure), σ was set to 20 (i.e., voting field covers the entire image). As SNR decreases, background and figure histograms overlap more and more until they become indistinguishable.

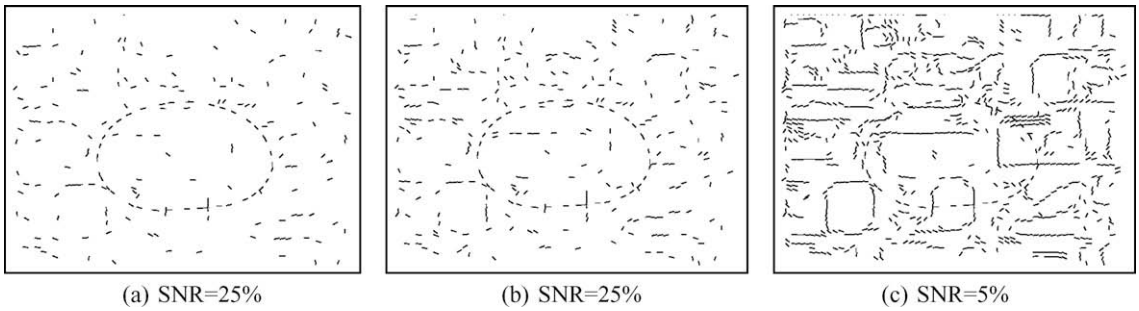


Fig. 15. Examples of dataset images assuming increasing SNR. Visual perception of the objects in these images agrees with the saliency histograms for figure and background produced by tensor voting (Fig. 14). The larger the overlap between figure and background histograms, the harder is to visually segment the objects from the background.

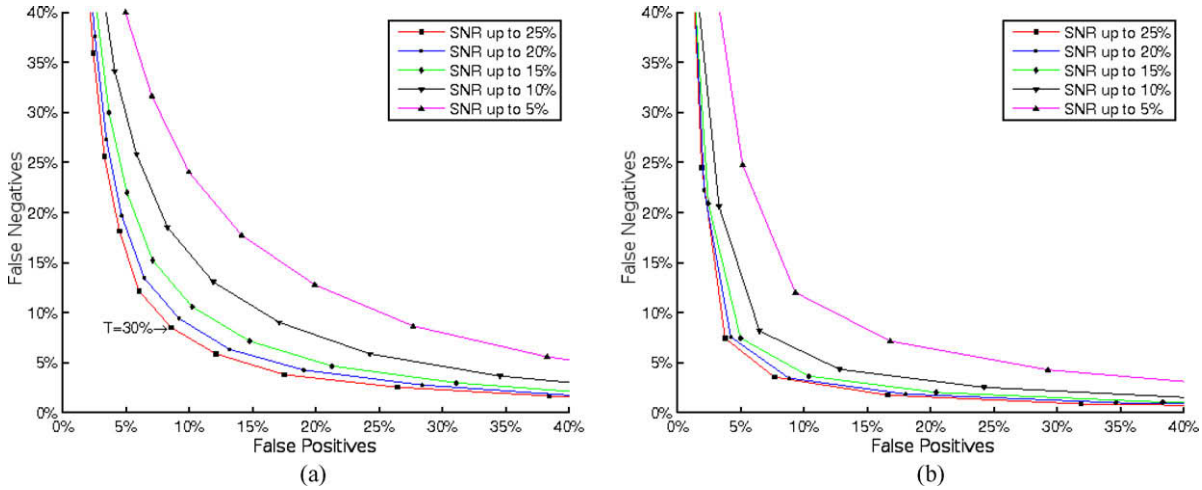


Fig. 16. (a) ROC curves corresponding to different SNR values using SSF-T. When SNR is up to 10%, the perception of the figures becomes more difficult. This is reflected by the overlapping saliency histograms shown in Fig. 14. (b) ROC curves corresponding to different SNR values using IMS-T with $\Delta T_S = 5\%$. We can observe improvements in all ROC curves compared to those obtained using SSF-T (a). In addition, the ROC curve for SNR up to 10% is closer to those corresponding to higher SNR values indicating that IMS-T can deal better with cluttered images.

to SNR up to 10% is closer to the ones corresponding to higher SNR values (i.e., up to 25%, up to 20% and up to 15%). This indicates that IMS-T deals with cluttered scenes much better. Fig. 17 shows some representative results using IMS-T.

The ROC curves of each approach can be compared side-by-side for quantitative evaluation purposes in Fig. 18. For the iterative approach, different step sizes ΔT_S were used (i.e., 5%, 10% and 15%), showing no remarkable differences between each other, while showing a considerable improvement over SSF-T for all SNR values.

To compare our results with those in [10], we have created plots of SNR vs FP, shown in Fig. 19a. Specifically, it compares the results obtained using SSF-T at $T = 30\%$—Fig. 16a), the best result obtained by IMS-T (i.e., 3 iterations using $AT_S = 5\%$—Fig. 16b), and the results reported in [10]. Since the results in [10] were not provided explicitly, we used a ruler over a hard copy of their plots to infer the values shown for their method in Fig. 19a.

Fig. 19b is a plot of SNR vs FN. In this case, a direct comparison with [10] is not possible since they do not report FN rates. As it can be seen from the plots, IMS-T shows improvements of more than 14% over [10] when SNR is up to 25%, and improvements of almost 90% whenSNR is up to 5%, while keeping a low FN rate. Compared to SSF-T, IMS-T improves figure vs noise discrimination by 5% on the average for all SNR values considered. The graphs also show a significantly smaller performance deterioration as SNR decreases.

6.1.2. Incomplete contour figures

Objects with incomplete boundaries were included in our benchmark to evaluate the performance of our method in the case of open contours. Gaps varying from 1/5 to 1/3 of the silhouette's length were introduced in each figure by eliminating adjacent segments (see Fig. 12a). Fig. 20 shows the saliency histograms of the same figure when its contour is closed or open. Specifically, Fig. 20a shows the saliency histogram of the complete contour in clean background while Fig. 20b shows the saliency histogram assuming cluttered background. Fig. 20c shows the saliency histogram of the same figure, with part of its contour deleted, in clean

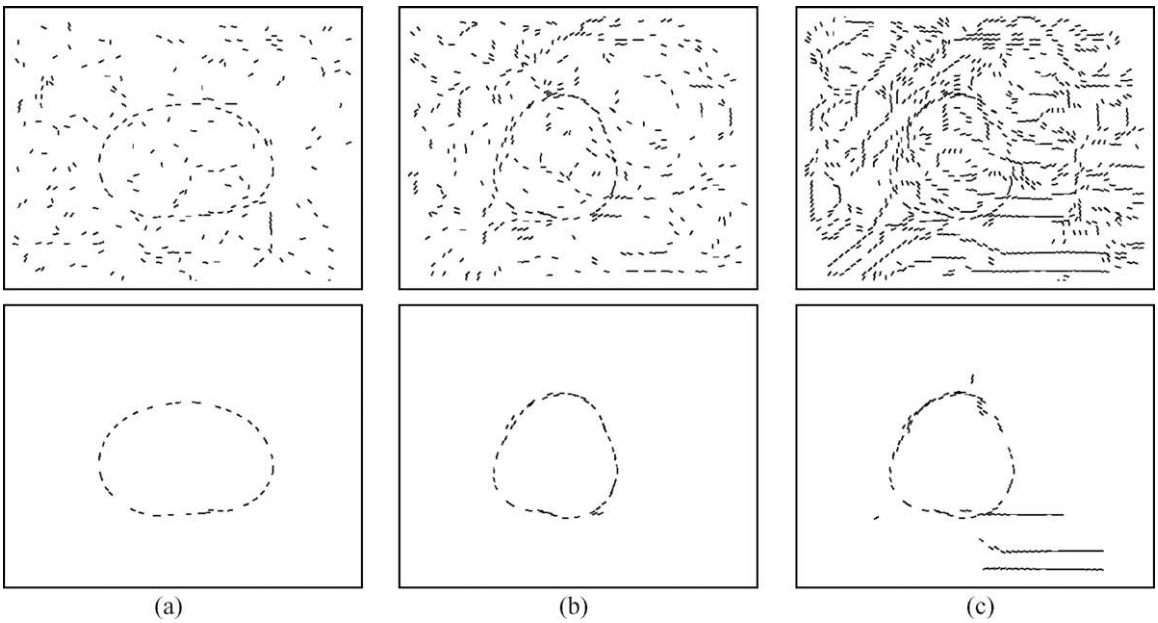


Fig. 17. Representative results using IMS-T: (a) avocado on bark with SNR up to 20%, (b) pear on wood background with SNR up to 15%, (c) pear on wood with SNR up to 5%.

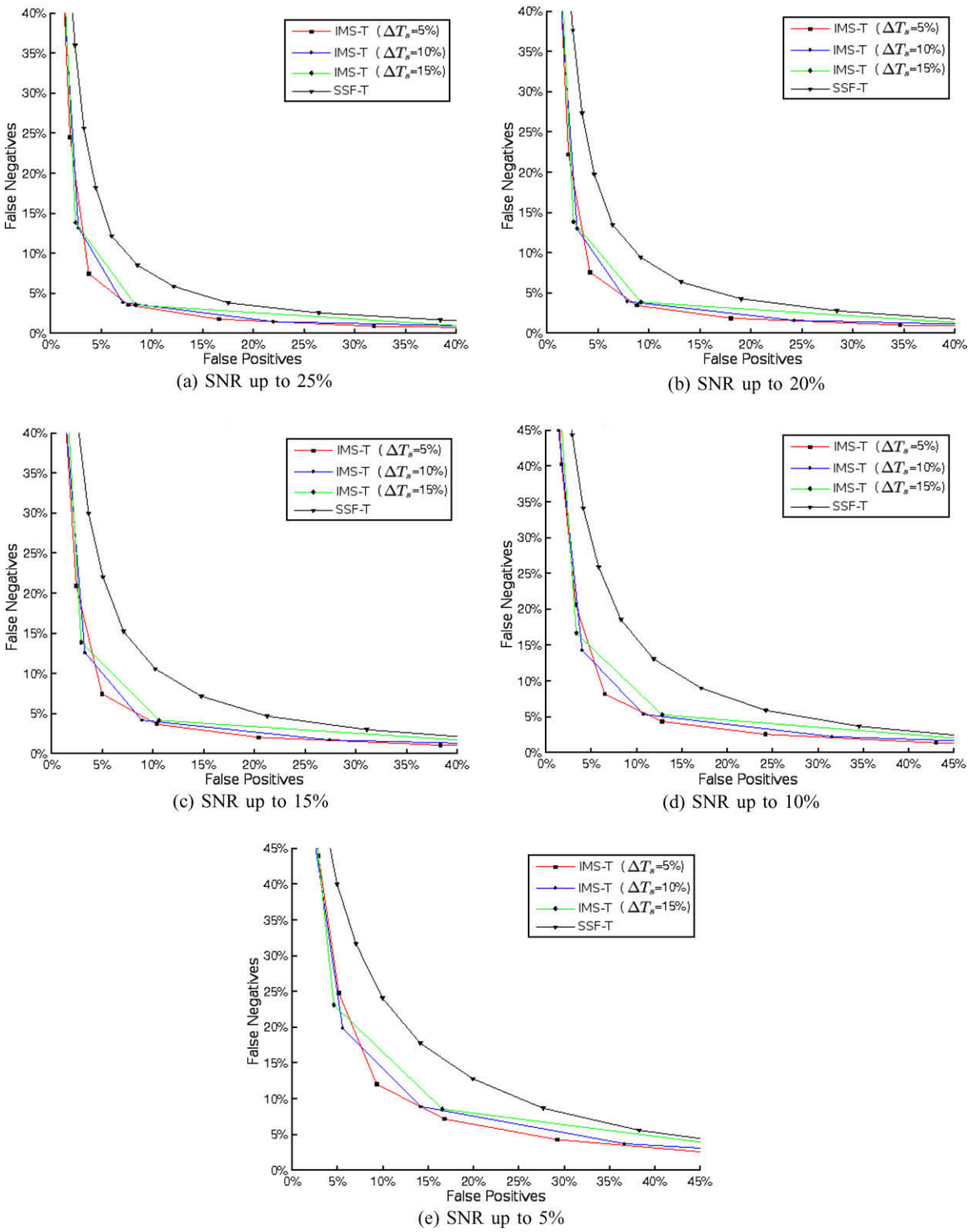


Fig. 18. ROC curves for SSF-T and IMS-T.

background, while Fig. 20d shows the saliency histogram of the same incomplete contour in cluttered background. The histograms corresponding to incomplete contours peak at the same position as those corresponding to the complete contours, however, they are rather wider. This is because the end segments are slightly less salient, due to the fact that they receive votes from one side of the contour only.

Fig. 21a shows the ROC curves obtained using SSF-T. The scale was chosen based on knowledge of the benchmark images (i.e., σ was set equal to 20, yielding a voting field that covers the entire image). Fig. 21b shows the ROC curves obtained using IMS-T. The scale parameter σ varies from 2 to 20 (covering from 5% to 100% of the image), ΔT_s was equal to 5%, and T_σ was equal to 50% (i.e., the saliency curve must be above T_s in at least half of the processed scales). This allows structures to pop out in any region of the scale range. Significant improvements can be noted again by comparing Fig. 21b to Fig. 21a. In addition, the ROC curve corresponding to SNR up to 10% is closer to the ones corresponding to higher SNR

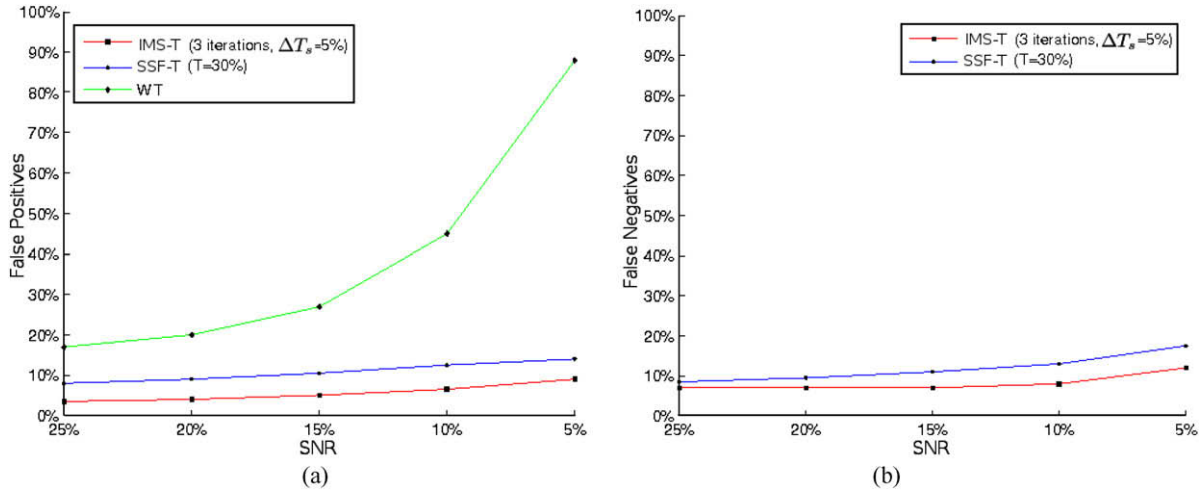


Fig. 19. Plots of (a) SNR vs FP and (b) SNR vs FN. IMS-T outperforms Williams and Thornber's method [10] as well as SSF-T. Also, it has a low FN rate and performs consistently as SNR decreases.

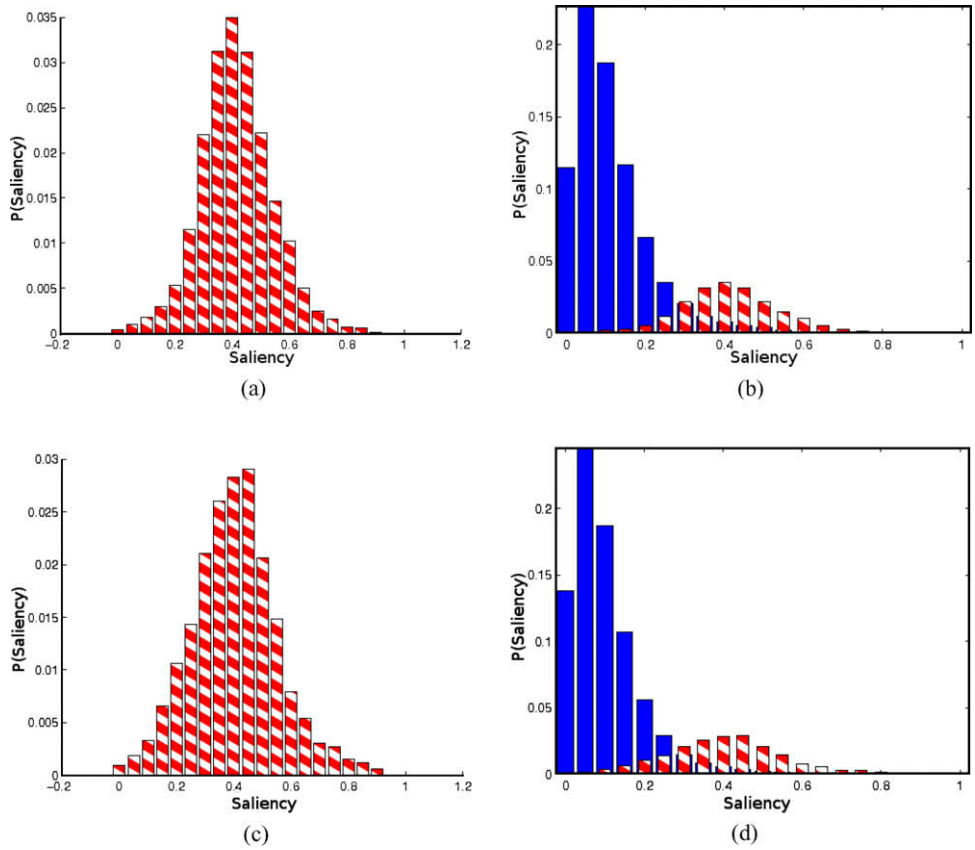


Fig. 20. Saliency histograms (dashed for figure), σ was set to 20 so that the voting field covers the entire image, (a) saliency histogram of closed contour in clean background, (b) saliency histogram in cluttered background, (c) incomplete contour in clean background, and (d) incomplete contour in cluttered background.

values (i.e., up to 25%, up to 20% and up to 15%). This indicates that IMS-T can deal with cluttered scenes much better even when the objects have incomplete contours. Fig. 22 shows some representative segmentation results using IMS-T. The ROC curves of each approach can be compared side-by-side for quantitative evaluation purposes in Fig. 23. As it can be observed, IMS-T improves segmentation results for all SNR values.

6.1.3. Multiple figures

In this set of experiments, we inserted multiple figures of the same absolute size over the background textures (e.g., see Fig. 12b). Fig. 24 shows several representative saliency histograms obtained in this case. As it can be observed, saliency histograms corresponding to different objects tend to overlap with each other. This tends to make the differentiation between each figure more

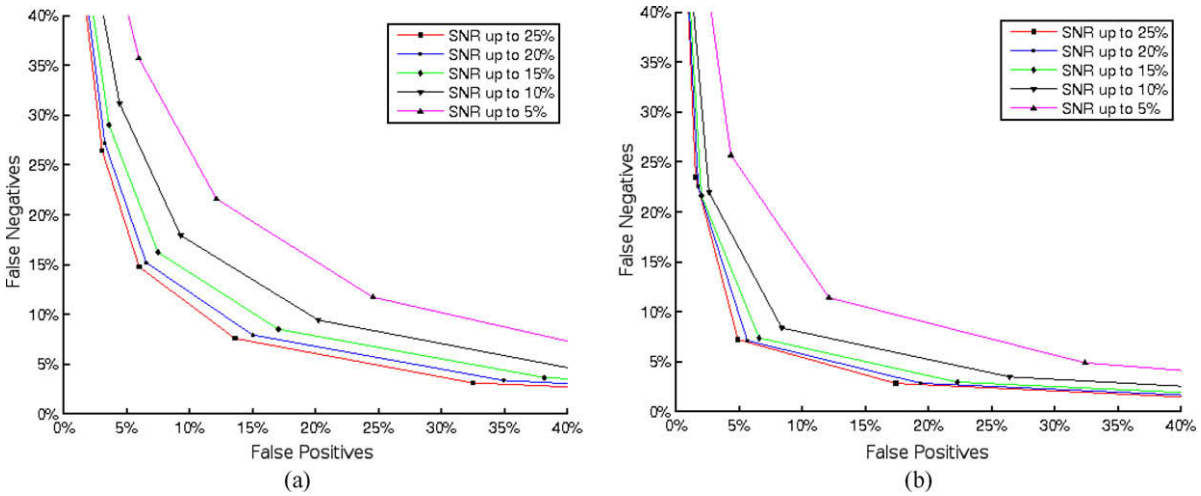


Fig. 21. (a) ROC curves using SSF-T in the case of incomplete contours, (b) ROC curves using IMS-T for the same dataset. We can observe improvements in all ROC curves compared to those obtained using SSF-T shown in part (a). In addition, the ROC curve for SNR up to 10% is closer to those corresponding to higher SNR values (i.e., SNR up to 25%, SNR up to 20% and SNR up to 15%), indicating that IMS-T can deal better with cluttered images.

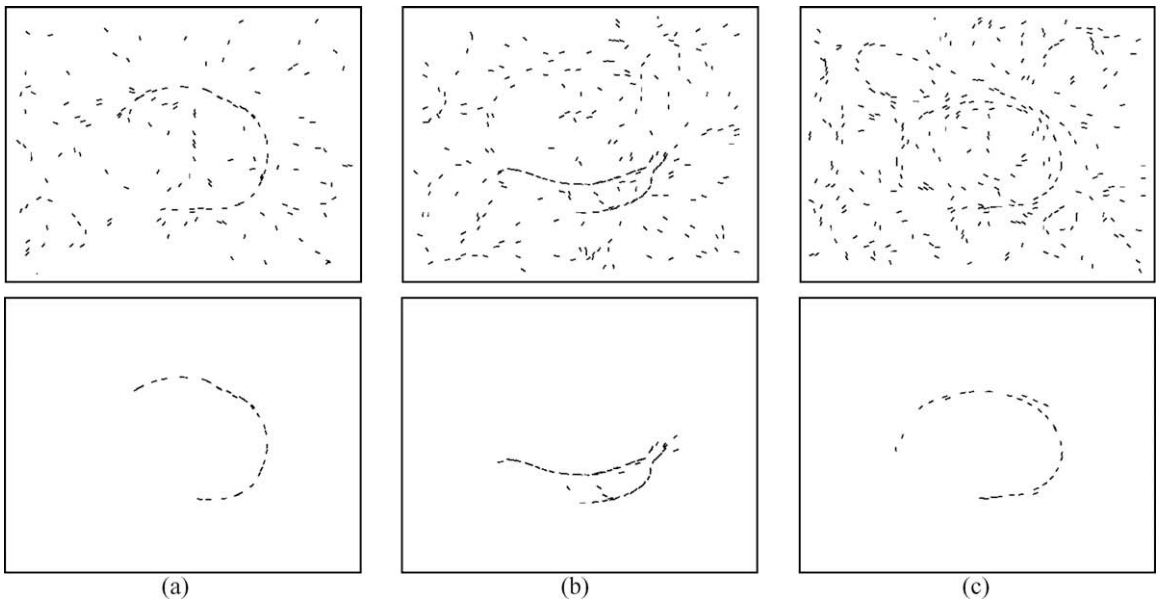


Fig. 22. Representative results using IMS-T in the case of incomplete contours: (a) peach on leaves with SNR up to 25%, (b) banana on bark with SNR up to 25%, (c) avocado on leaves with SNR up to 10%.

difficult, but also strengthens the figure saliency compared to the background. Fig. 25 shows the ROC curves corresponding to SSF-T and IMS-T. Again, wecan observe remarkable improvements using IMS-T. Fig. 26 shows representative results using IMS-T in three images belonging to the multiple figure dataset.

6.1.4. Figure size variation

To bring up the scale analysis issue (i.e., Fig. 12c), we have also experimented with multiple figures having different size. Specifically, we used three different absolute sizes in our experiments: 20, 32 and 40 squared pixels. Fig. 27 shows representative saliency histogram corresponding to one, two, and three objects of different size. A shift in the histograms of the second figure (green) can be noticed due to its variation in size. This reflects the fact that the scale chosen was more adequate for one object than the other. In real cases, these differences are even bigger, making objects to pop out in different scales, that is, objects present stronger saliency in certain scales than others. Fig. 28 shows the ROC curves for SSF-T and IMS-T. Again, we can observe remarkable improvements using IMS-T. Fig. 29 shows representative results using IMS-T.

6.2. Part II: Experiments on natural images

Among the five boundary detection methods evaluated on the Berkeley dataset and post-processed by our method, four of them (i.e., GM, MGM, TG, and BG) perform boundary detection using a single cue while one of them (i.e., BTG) combines information from two different cues using the method of Martin et al. [33]. Each method produces a BPP map which is used as input to IMS-T. The input then consists in boundary pixels encoded as tensors whose magnitudes and directions are given, respectively, by their BPP values and gradient directions computed from the original image. Note, though, that since accuracy is more important than time consumption in this part of the experiments, the sampling suggested in Section 3.2 was not performed. IMS-T outputs a new BPP map by incorporating perceptual organization cues.

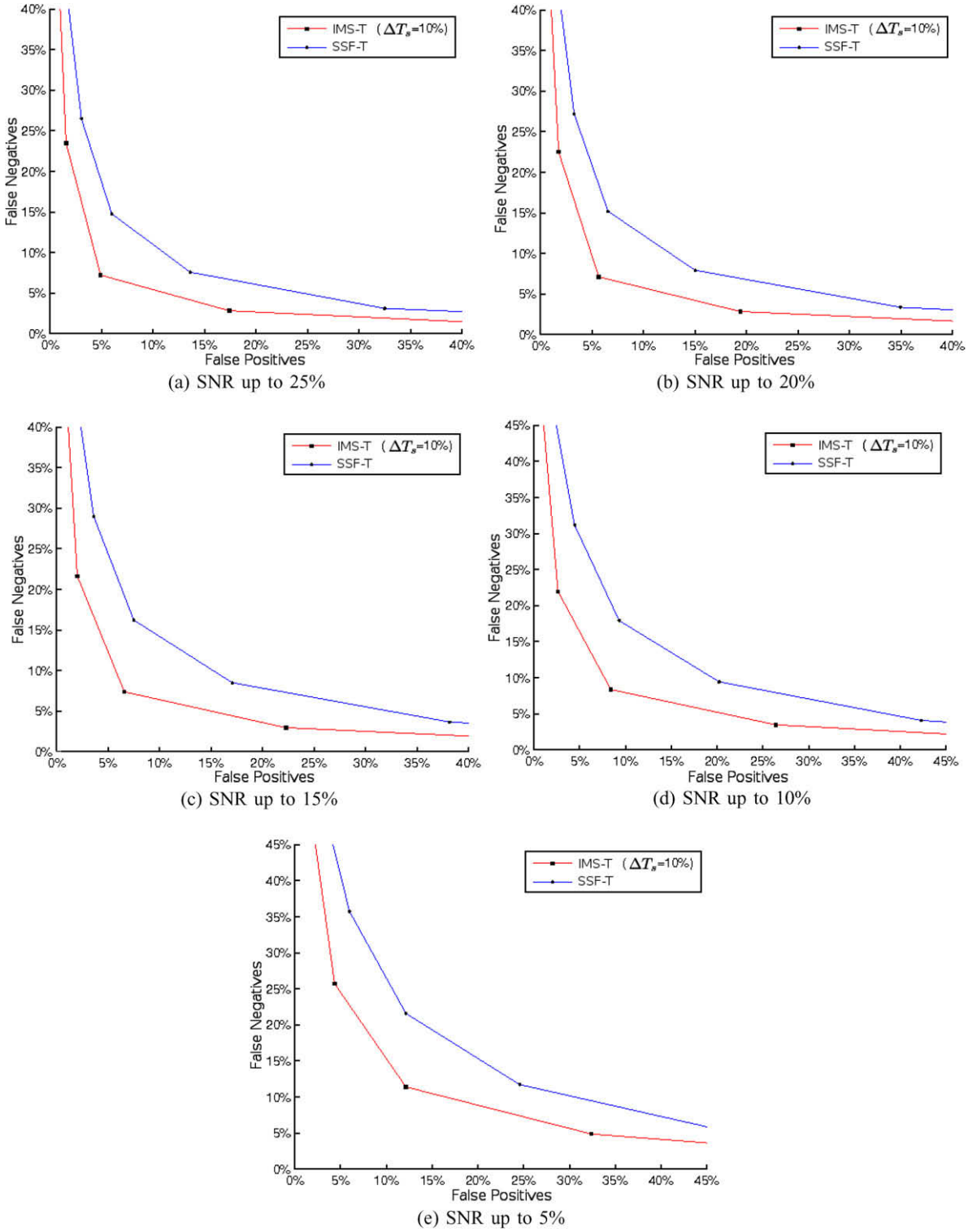


Fig. 23. Side-by-side comparison of SSF-T and IMS-T for the dataset composed of open or incomplete figures.

A common characteristic to all five methods is their reliance on image photometric information to build a BPP map. The GM method computes image gradient magnitudes at each pixel to produce the BPP map. The gradients are estimated using a pair of Gaussian derivative filters at a unique, learned, optimal scale. Learning was performed using 200 training images from the Berkeley segmentation dataset. The MGM method computes image gradient magnitudes at two different scales to produce the BPP map. The gradients are estimated at each pixel using pairs of Gaussian derivative filters at two, also learned, optimal scales. The BG method uses local brightness gradients to obtain the BPP map. The gradients are estimated using a χ^2 difference in the distribution of pixel luminance values of two half discs centered at a given pixel and divided in half at the assumed boundary orientation.

The TG method uses local texture gradients to produce the BPP map. The gradients are estimated using a χ^2 difference in the dis-

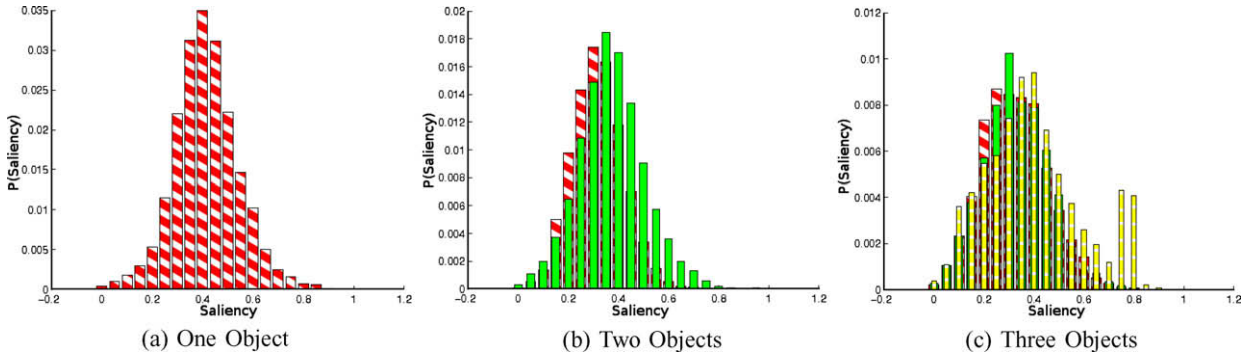


Fig. 24. Saliency histograms using multiple objects of the same absolute size. The parameter σ was set to 20 so that the voting field covers the entire image.

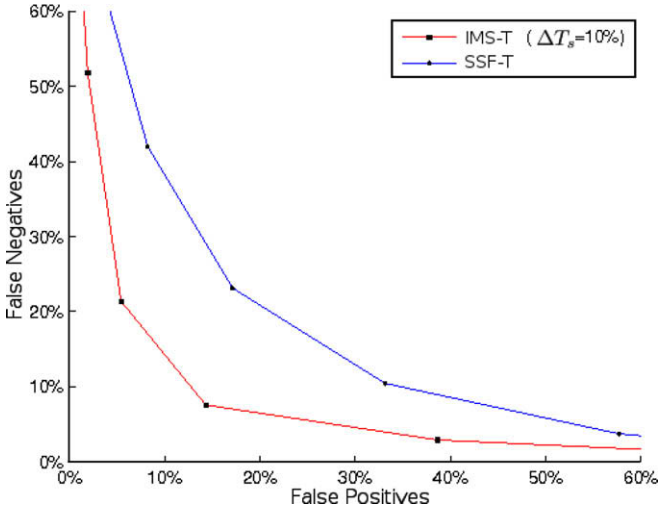


Fig. 25. ROC curves corresponding to SSF-T and IMS-T using images composed of multiple figures of the same absolute size. Remarkable improvements can be observed in the case of IMS-T.

tribution of *textons* of two half discs centered at a given pixel and divided in half at the assumed boundary orientation. Textons are computed by clustering the responses of a bank of filters using K-means. The bank of filters was composed of standard even- and odd-symmetric quadrature pair elongated linear filters. The BTG method combines local brightness and texture gradients to obtain the BPP. BTG has demonstrated one of the best performances to date on the Berkeley segmentation benchmark. Additional information about each of these methods can be found in [33].

To get a better insight, we have analyzed below certain local configurations in natural images. This analysis can reveal upfront situations where IMS-T would be most beneficial, and others where it would be expected to make no improvements or even degrade the results. Let us consider Fig. 30, for example. The regions within the red square in each of the images shown in Fig. 30a and b have been magnified for clarity and shown in Fig. 30a.l and b.1. The respective BG and BTG PB maps are shown in Fig. 30a.2 and b.l, where lighter intensities correspond to a lower probability. One can notice that parts of the contour around the main objects in each image are diminished due to the low contrast between them and the background. However, let us suppose now that we encode

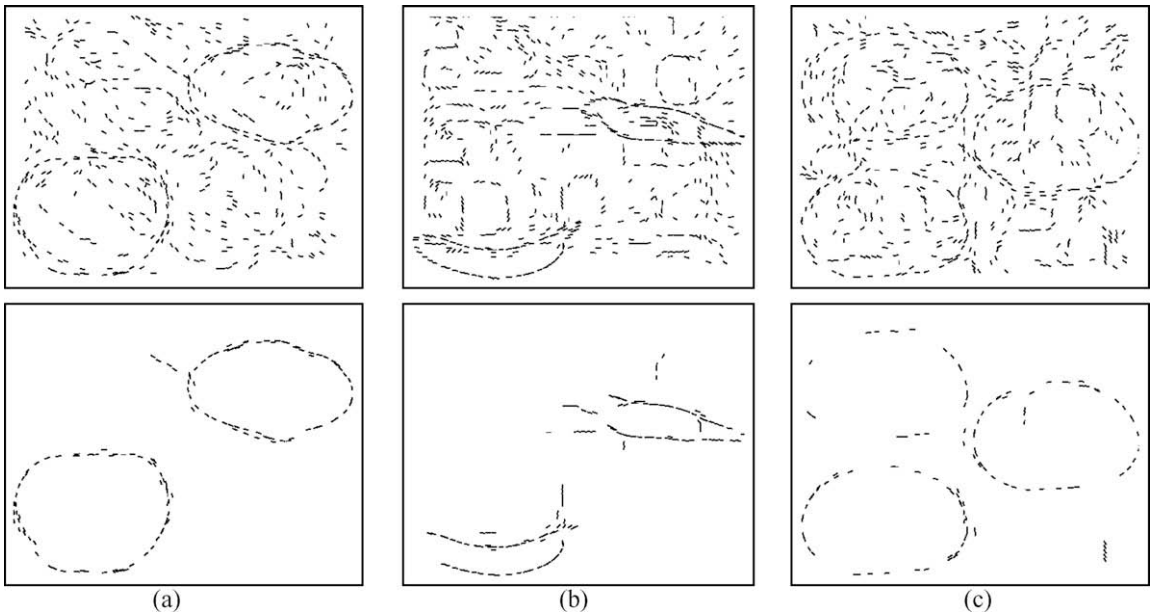


Fig. 26. Representative results using IMS-T in the case of multiple figures of the same absolute size: (a) apple and red onion on fabric ground, (b) banana and sweet potato on bark ground, (c) three avocados on bark ground.

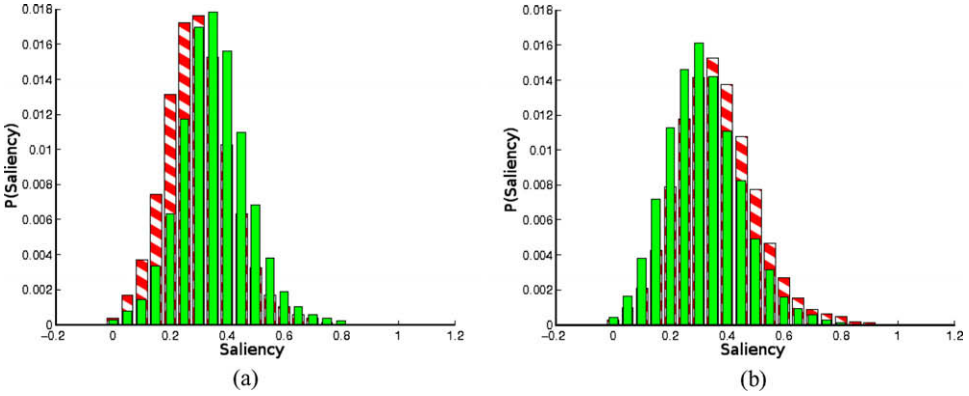


Fig. 27. Saliency histograms corresponding to multiple objects having different size (striped—first, unchanged figure), σ was set to 20 so that the voting field covers the entire image. A shift in the histograms of the second figure can be noticed due to its variation in size.

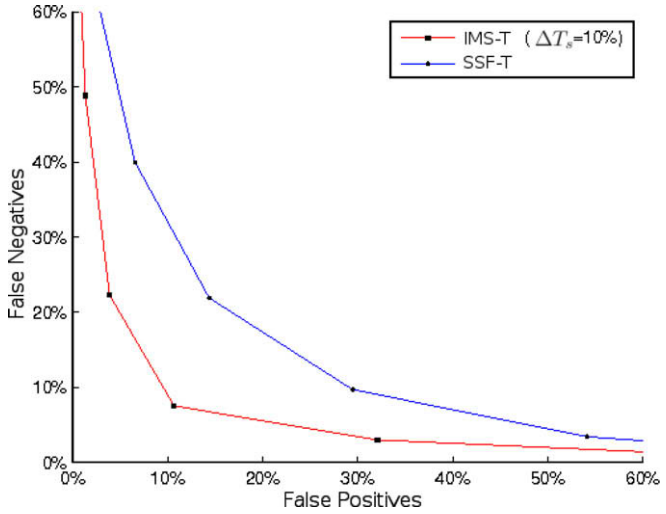


Fig. 28. ROC curves corresponding to SSF-T and IMS-T for the case of multiple figures having different size.

these values as tensors, as shown in Fig. 30a.3 and b.3, where tensor size is given by the PB map value and tensor direction by the normal to the edge direction. If we apply IMS-T, these same contours can be intensified as shown in Fig. 30a.4 and b.4. This is because the communication between neighboring segments reveals the locally organized structure underlying those contours. In other words, a plausible continuation between the penguin's neck and chest, as well as between the sail's parts, can be found, improving the results produced by BG and BTG.

On the other hand, let us consider Fig. 31. The regions shown by the red squares in each of the images in Fig. 31a and b have been magnified for clarity and shown in Fig. 31a.1 and b.1. Fig. 31a.2 and b.2 show the respective GM and BG PB maps. It should be noted in these cases that GM and BG produced strong responses due to the high contrast between the object and the background. If we encode these values as tensors, as shown in Fig. 31a.3 and b.3, and apply IMS-T, then these contours will be deteriorated as shown in Fig. 31a.4 and b.4. This is because the communication between neighboring segments from the jagged edges is weak, since

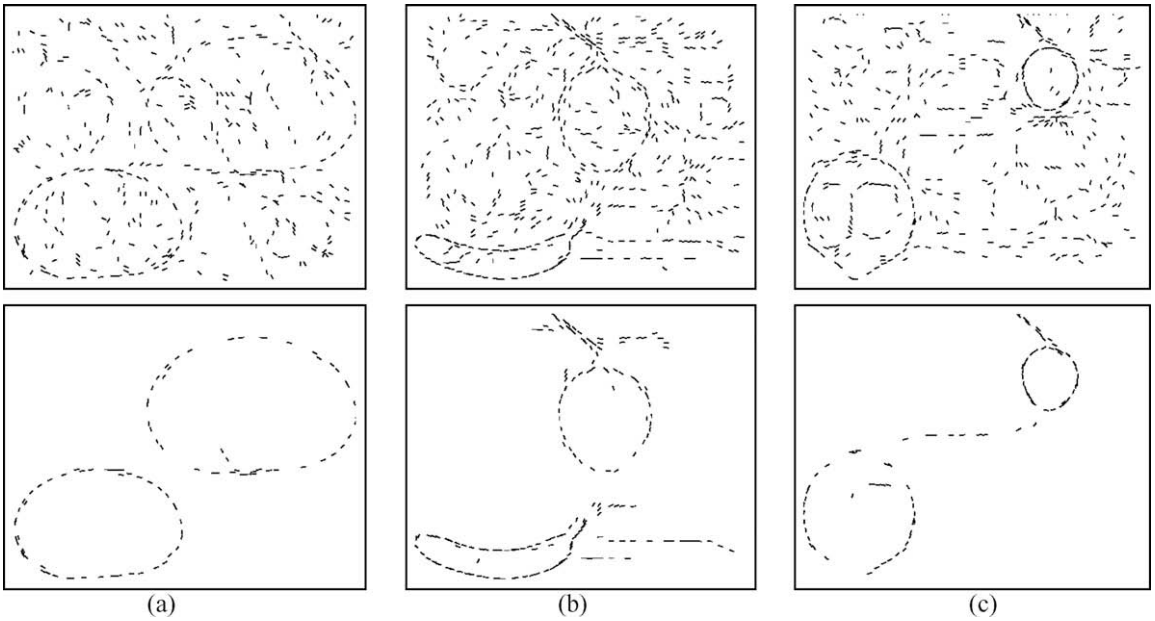


Fig. 29. Representative results using IMS-T in the case of multiple figures having different size: (a) two avocados on sand ground, (b) banana and tamarillo (larger) on wood ground, (c) lemon and tamarillo (smaller) on brick ground.

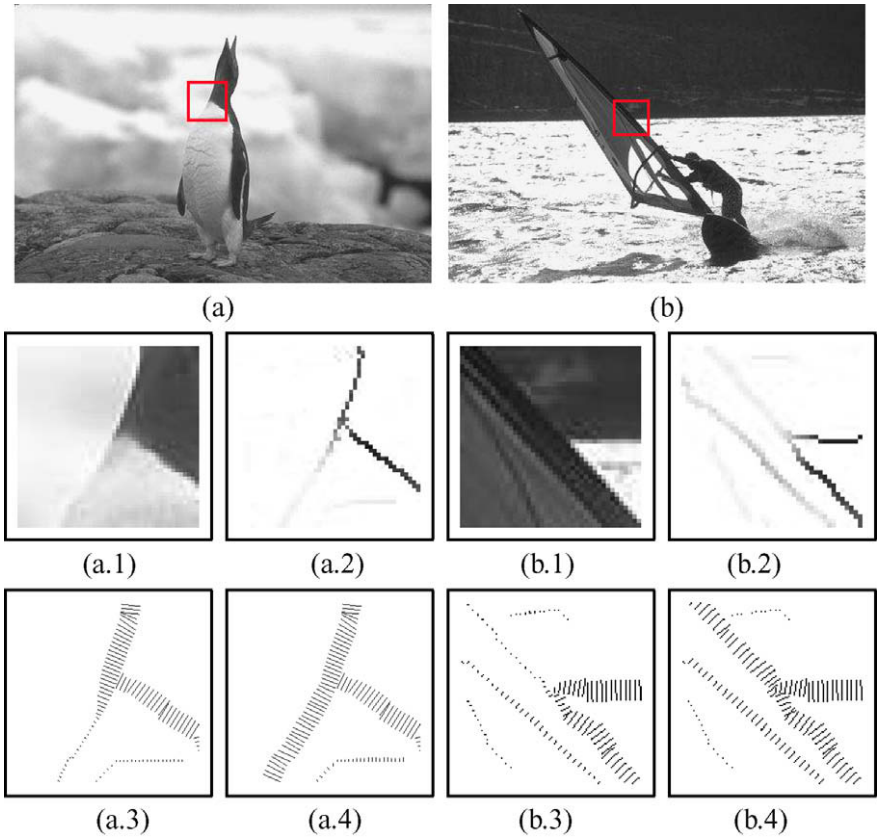


Fig. 30. Examples illustrating cases where IMS-T improves boundary detection (see text for details): (a and b) original images from Berkeley dataset, (a.1 and b.1) region within the red squares magnified, (a.2 and b.2) BG and BTG PB maps, (a.3 and b.3) gradients encoded as tensors in IMS-T, (a.4 and b.4) tensors after iterative voting using IMS-T. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

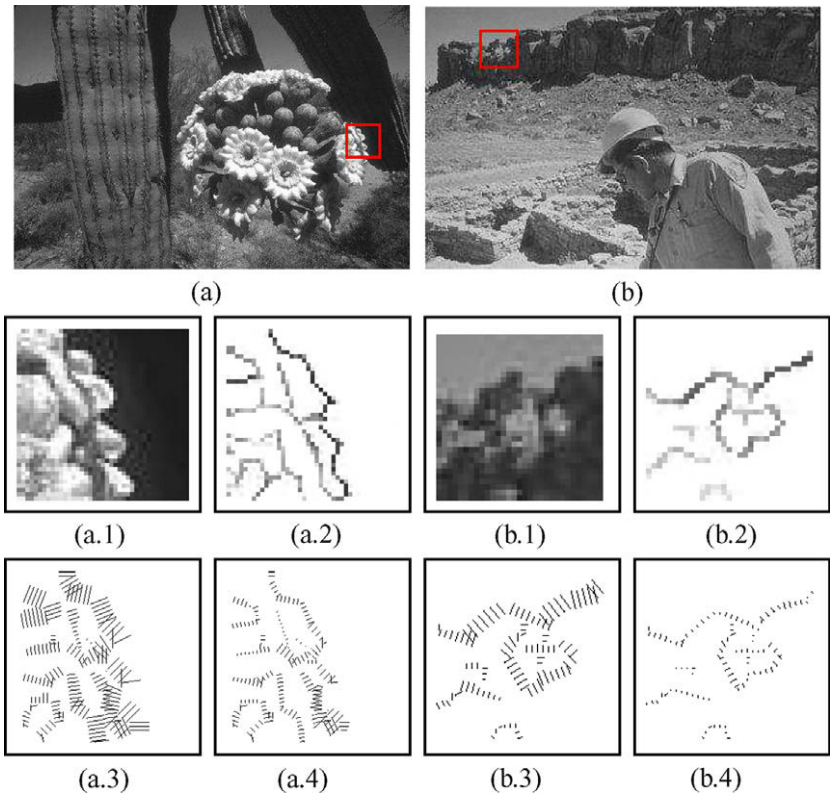


Fig. 31. Examples where perceptual grouping degrades boundary detection (see text for details): (a) original images from the Berkeley dataset, (a.1 and b.1) regions within the red squares magnified, (a.2 and b.2) GM and BG PB maps, (a.3 and b.3) magnitudes encoded as tensors in IMS-T, (a.4 and b.4) tensors after iterative voting using IMS-T. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

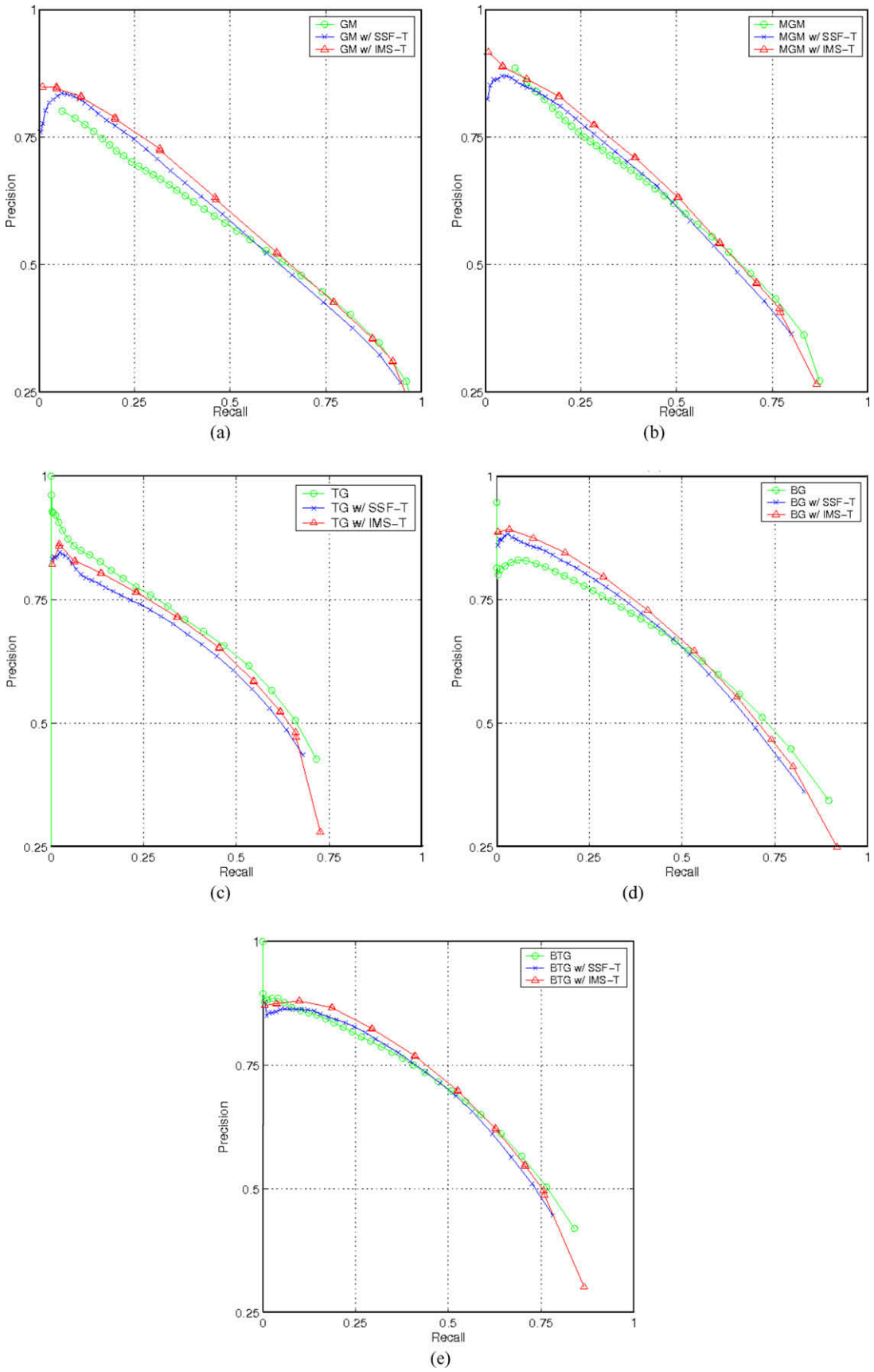


Fig. 32. Average PRCs comparing each method with and without post-processing: (a) GM, (b) MGM, (c) TG, (d) BG, (e) BTG. The resulting *F*-measure and AAC are shown in Table 4.

Table 4
Resulting *F*-measure (*F*) at equal regime and AAC for the five methods tested with and without post-processing

Method	Original		w/ SSF-T		w/ IMS-T	
	F	AAC	F	AAC	F	AAC
GM	.56	.43	.56	.41	.57	.38
MGM	.58	.31	.57	.32	.58	.28
TG	.58	.21	.56	.26	.57	.24
BG	.60	.34	.59	.33	.60	.31
BTG	.63	.28	.61	.29	.62	.26

Table 5
Results based on the *F*-measure at equal regime obtained by post-processing the 100 test images from the Berkeley dataset using IMS-T

Method	NII	AIR (%)	NID	ADR (%)
GM	40	9.0	60	5.3
MGM	35	5.5	65	3.7
TG	24	3.1	76	4.1
BG	40	4.9	60	4.9
BTG	36	4.4	64	3.4

Table 6
Results based on AAC obtained by post-processing the 100 test images from the Berkeley dataset using IMS-T

Method	NII	AIR (%)	NID	ADR (%)
GM	71	8.4	29	5.2
MGM	72	5.2	28	3.8
TG	62	4.1	38	3.5
BG	82	6.9	18	3.6
BTG	84	7.2	16	3.6

they do not satisfy the rules of good continuation and smoothness. This degrades the results produced by GM and BG.

The selection of the most adequate features for each case seems to be a promising issue here. However, a wise consideration may not be straightforward, thus left out of the scope of this work.

Fig. 32 shows the PRCs for each of the five boundary detection methods tested. Each graph also shows the corresponding PRC using SSF-T and IMS-T for post-processing. Each curve is the average over 100 PRCs corresponding to the 100 test images in the Berkeley segmentation dataset. SSF-T curves represent the best result obtained by testing different scales. Table 4 shows the *F*-measure and AAC values for each PRC. As it can be noted, at equal regime, SSF-T is not able to improve any method, while IMS-T partially improved one method (i.e., GM), slightly degraded another method (i.e., TG), and partially improved or degraded the rest (i.e., MGM, BG, and BTG). Considering the AAC measure, however, SSF-T improved two methods (GM and BG), degrading the others, while IMS-T improved all methods except TG. The reason why TG was not improved by IMS-T is because most boundaries found using texture gradient violate the perceptual organization rules used by IMS-T. For the methods shown improvement,it is interesting to note that post-processing improved the results at certain thresholds, that is, more improvements can be noticed at a high precision regime.

Looking at the PRCs alone does not provide sufficient information to appreciate the benefits of integrating perceptual organization cues with segmentation. Tables 5 and 6 provide more information to further analyze the results obtained by IMS-T. Specifically, each table shows the actual Number of Images Improved (NII) after post-processing, the Average Improvement Rate (AIR), the Number of Images Degraded (NID) after post-processing, and the Average Degradation Rate (ADR) for each method. Table 5 shows the same statistics using the *F*-measure while Table 6 shows the same statistics using the AAC value. The results based on the *F*-measure indicate that although the number of images improved is lower than the number of images degraded, the average rate of improvement is usually higher than the average rate of degradation. In other words, the rate of improvement is higher for the images improved than the rate of degradation for the images damaged. Considering the same statistics in the case of AAC, it is more clear that IMS-T is really beneficial as a post-processing step. It has not only improved more images, the rate of improvement is also higher on the average. At the same time, it has degraded less images with a lower rate on the average.

A detailed analysis of these results can reveal even more information about the kind of images that are more likely to be improved by IMS-T. Table 7 shows the number of images improved by IMS-T, considering the *F*-measure at equal regime, relative to the *F*-measure obtained by the original methods. The results show that 53.3–84.6% of the images resulting in *F*-measures originally below .5 were improved. As the resulting *F*-measure increases, the rate of improved images decreases. These results indicate that perceptual organization cues are especially beneficial to images having low *F*-measures. Although we would have to experiment more to further verify this observation, it appears that such images

Table 7
Improvement based on the *F*-measure at equal regime relative to the original *F*-measure

Method	[.0,.5]	(.5,.6]	(6,.7]	(.7,.8]	(.8,.1]
GM	14/17 (82.4%)	16/30 (53.3%)	8/29 (27.6%)	2/20 (10.0%)	0/4 (0.0%)
MGM	12/16 (75.0%)	14/30 (46.7%)	8/32 (25.0%)	1/19 (5.3%)	0/3 (0.0%)
TG	8/15 (53.3%)	8/35 (22.8%)	6/40 (15.0%)	2/7 (28.6%)	0/3 (0.0%)
BG	11/13 (84.6%)	16/25 (64.0%)	11/29 (37.9%)	2/28 (7.1%)	0/5 (0.0%)
BTG	7/10 (70.0%)	12/17 (70.6%)	13/33 (39.4%)	4/34 (11.8%)	0/6 (0.0%)

Table 8
Improvement based on the AAC relative to the original *F*-measure

Method	[.0,.5]	(.5,.6]	(.6,.7]	(.7,.8]	(.8,.1]
GM	15/17 (88.2%)	28/30 (93.3%)	16/29 (55.2%)	10/20 (50.0%)	2/4(50.0%)
MGM	14/16 (87.5%)	25/30 (83.3%)	19/32 (54.4%)	13/19 (68.4%)	1/3(33.3%)
TG	13/15 (86.7%)	21/35 (60.0%)	23/40 (57.5%)	4/7 (57.1%)	1/3(33.3%)
BG	12/13 (92.3%)	24/25 (96.0%)	19/29 (65.5%)	22/28 (78.6%)	5/5(100.0%)
BTG	7/10 (70.0%)	17/17 (100.0%)	28/33 (84.9%)	28/34 (82.4%)	4/6(66.7%)

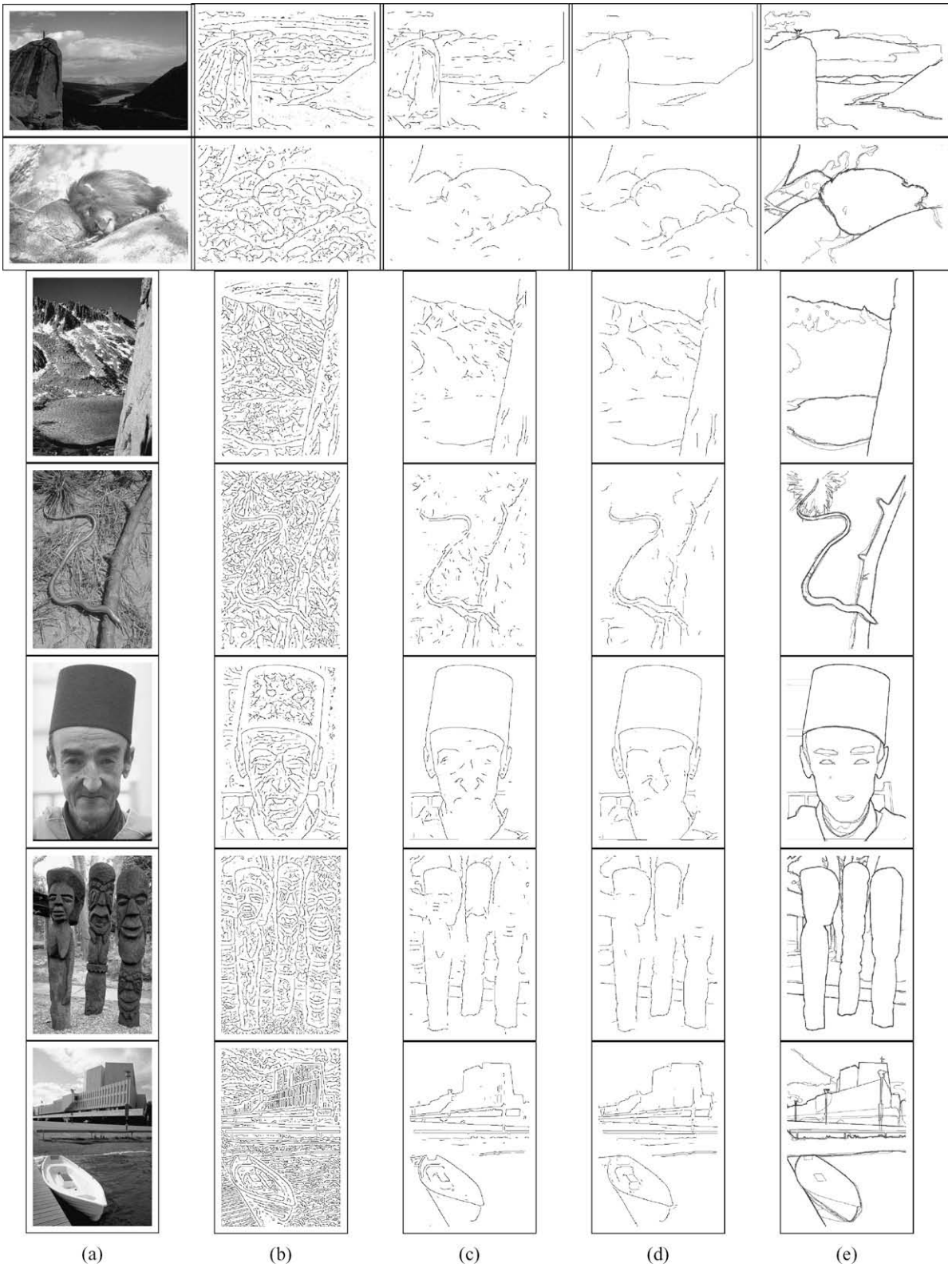


Fig. 33. Visual comparison of results: (a) original gray-scale images, (b) initial boundaries detected, (c) resulted boundaries by thresholding at the optimal *F*-measure (d) resulted boundaries using post-processing, thresholded at the optimal *F*-measure, (e) ground truth.

are not well explained by the features extracted. On the other hand, when the features extracted can explain an image well, then post-processing seems to have less effect.

Table 8 shows the number of images improved by IMS-T, considering the AAC value relative to the *F*-measure obtained by the original methods. Besides the fact that 70.0–92.3% of the images resulting in *F*-measures originally below .5 were improved, it is interesting to note that high rates in general were achieved throughout the whole *F*-measure range. These results suggest that independently of the performance achieved by a given method, it

might be always possible to improve its overall performance using perceptual organization cues for post-processing.

Fig. 33 shows some boundary detection results for each method with and without IMS-T. As it can be observed, IMS-T eliminates noisy segments more effectively, preserving boundary segments that satisfy the perceptual organization principles underlying IMS-T.

7. Conclusions and future work

We have presented a new approach for perceptual grouping of oriented segments in highly cluttered images using an iterative, multi-scale tensor voting approach. Our approach removes noisy segments conservatively using multi-scale analysis and re-votes on the retained segments. We have tested our approach on various datasets composed by synthetic and real images. Our experimental results with synthetic images indicate that our method can segment successfully objects in images with up to twenty times more noisy segments than object ones. Moreover, it can handle objects with incomplete boundaries as well as multiple objects having different size. Overall, IMS-T has shown to work well when applied on highly cluttered images, and it does not depend on any assumptions regarding the size, number, or boundary completeness of the objects in the image. Our experimental results using real images show that IMS-T improved up to 40% of the test images, when considering the *F*-measure at equal regime as a performance measure. These improvements were especially noticed among images having low *F*-measures originally, although, in general, a higher performance is more obvious at high precision regime. When considering the AAC measure, IMS-T improved up to 84% of the test images and across the entire range of original *F*-measure. In general, the improvements happen on the high precision-extreme of the PRCs across the entire database, as revealed by the average curves in Fig. 32. Consequently, the improvements are higher in precision than in recall. This is the expected behavior of the conservative elimination of segments, which aims at preserving figure edges while eliminating clutter. In an ideal result, the recall rate should be kept constant while the precision rate increases. Of course, in the process of eliminating clutter, some figure edges are also eliminated and the recall rate decreases.

The results obtained in this study look particularly interesting and encouraging to us. The benefits of iterative, multi-scale segmentation are quite clear. For future work, we plan to improve and extend IMS-T in several ways. First, we plan to investigate the issue of choosing the parameters of our method (i.e., T_s, ΔT_s, T_σ, I) automatically. We have reported preliminary using on this issue a case-based thresholding scheme in [44]. The idea is classifying saliency histograms in several cases by considering the relative position of the modes of the figure/ground distributions and applying specific actions in each case. Another idea would be employing learning using the 200 training images in the Berkeley dataset. Second, we plan to improve segmentation results by better preserving junctions and corners. Small scales result in higher saliency for points very close to a corner, however, as scale increases votes from the other edge of the corner blur the orientation estimate and reduce the saliency of such points. As a result, certain corners and junctions might be removed during the iterative process. One idea is to use polarity information in order to preserve such points [19]. Third, we plan to consider ways to speed-up our method. Although our analysis in Section 4 shows that our method has asymptotically the same complexity as voting at a single fixed scale, it might not be appropriate for real-time applications. Finally, we plan to apply IMS-T in the context of different segmentation problems such as region segmentation or finding text regions in images for automatic map annotation.

References

[1] L. Williams, Fruit and texture images. Available from: <http://www.cs.unm.edu/~williams/saliency.html>, 2008 (last viewed in July 2008).
[2] C.F.P. Arbelaez, D. Martin, The berkeley segmentation dataset and benchmark. Available from: <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>, 2008 (last viewed in July 2008).
[3] J. Dolan, R. Weiss, Perceptual grouping of curved lines, IUW (1989) 1135–1145.
[4] F. Ulupinar, Perception of 3-d surfaces from 2-d contours, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1993) 592–597.
[5] F. Stein, G. Medioni, Recognizing 3d objects from 2d groupings, IUW (1992) 667–674.
[6] W. Singer, C. Gray, Visual feature integration and the temporal correlation hypothesis, Annual Review of Neuroscience 18 (1995) 555–586.
[7] B. Moore, Information extraction and perceptual grouping in the auditory system, in: V. Cantoni, V. Di Ges, A. Setti, D. Tegolo (Eds.), Human and Machine Perception: Information Fusion, Plenum Press, New York, 1997.
[8] M. Wertheimer, Untersuchungen zur lehre von der gestalt (english translation), II. Psychologische Forrschung 4 (1929) 301–350.
[9] N. Ahuja, M. Tuceryan, Extraction of early perceptual structure in dot patterns: integrating region boundary, and component, gestalt, CVGIP (1989) 304–356.
[10] L. Williams, K. Thornber, A comparison measures for detecting natural shapes in cluttered background, International Journal of Computer Vision 34 (2/3) (2000) 81–96.
[11] S. Mahamud, L. Williams, K. Thornber, K. Xu, Segmentation of multiple salient closed contours from real images, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (4) (2003).
[12] D. Lowe, Three-dimensional object recognition from single two-dimensional images, Artificial Intelligence 31 (1987) 355–395.
[14] L. Hérault, R. Horaud, Figure-ground discrimination: a combinatorial optimization approach, IEEE Transactions on Pattern Recognition and Machine Intelligence 15 (1993) 899–914.
[15] S. Ullman, A. Sha'ashua, Structural saliency: the detection of globally salient structures using a locally connect network, in: Second International Conference on Computer Vision—ICCV'88, 1988.
[16] S. Sarkar, K. Boyer, Quantitative measures of change based on feature organization: eigenvalues and eigenvectors, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition—CVPR'96, 1996, pp. 478–483.
[17] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.
[18] G. Guy, G. Medioni, Inference of surfaces 3-d curves and junctions from sparse noisy 3-d data, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (11) (1997) 1265–1277.
[19] G. Medioni, M.-S. Lee, C.-K. Tang, A Computational Framework for Segmentation and Grouping, Elsevier Science, 2000.
[20] P. Mordohai, G. Medioni, Junction inference and classification for figure completion using tensor voting, in: Workshop on Perceptual Organization in Computer Vision, 2004.
[21] M. Hund, B. Mertsching, A computational approach to illusory contour perception based on the tensor voting technique, LNCS 3773, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 71–80.
[22] P. Mordohai, G. Medioni, Stereo using monocular cues within the tensor voting framework, European Conference on Computer Vision, Lecture Notes in Computer Science. 3024 (2004) 588–601.
[23] P. Mordohai, G. Medioni, Dense multiple view stereo with general camera placement using tensor voting, in: International Symposium on 3-D Data Processing, Visualization and Transmission, 2004, pp. 725–732.
[24] P. Kornprobst, G. Medioni, Tracking segmented objects using tensor voting, CVPR (2000) 118–125.
[25] W.-S. Tong, C.-K. Tang, G. Medioni, Simultaneous two-view epipolar geometry estimation and motion segmentation by 4d tensor voting, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1167–1184.
[26] M. Nicolescu, G. Medioni, A voting-based computational framework for visual motion analysis and interpretation, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (5) (2005) 739–752.
[27] C.-K. Tang, G. Medioni, M.-S. Lee, N-dimensional tensor voting and application to epipolar geometry estimation, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (8) (2001) 829–844.
[28] J. Jia, C.-K. Tang, Image repairing: robust image synthesis by adaptive Nd tensor voting, in: Proceedings of IEEE Computer Vision and Pattern Recognition, vol. 1, 2003, pp. 643–650.
[29] W.-S. Tong, C.-K. Tang, P. Mordohai, G. Medioni, First order augmentation to tensor voting for boudary inference and multiscale analysis in 3d, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (5) (2004) 594–611.
[30] Y.-W. Tai, W.-S. Tong, C.-K. Tang, Perceptually-inspired and edge-directed color image super-resolution, in: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006.
[31] S. Fischer, P. Bayerl, H. Neumann, G. Cristobal, R. Redondo, Are iterations and curvature useful for tensor voting? in: Proceedings of European Conference on Computer Vision (ECCV04), vol. 3, 2004, pp. 158–169.
[32] S. Fischer, P. Bayerl, H. Neumann, G. Cristobal, R. Redondo, Iterated tensor voting and curvature improvement, Signal Processing 87 (2007) 2503–2515.

[33] D.R. Martin, C.C. Fowlkes, J. Malik, Learning to detect natural image boundaries using local brightness color and texture cues, IEEE Transactions on Pattern Recognition and Machine Intelligence 26 (5) (2004) 530–549.
[34] L. Loss, G. Bebis, M. Nicolescu, A. Skurikhin, Perceptual grouping based on an iterative multi-scale tensor voting, in: Second International Symposium on Visual Computing (ISVC06), LNCS 4292, 2006, pp. 1786–1797.
[35] R. Mohan, R. Nevatia, Segmentation and description based on perceptual organization, CVPR (1989) 333–341.
[36] R. Mohan, R. Nevatia, Using perceptual organization to extract 3-d structures, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (11) (1989) 1121–1139.
[37] S. Ullman, Filling-in the gaps: the shape of subjective contours and a model for their generation, Biological Cybernetics 25 (1976) 1976.
[38] G. Guy, G. Medioni, Inferring global perceptual contours from local features, International Journal of Computer Vision and Image Understanding. 20 (1996) 113–133.
[39] P. Parent, S.W. Zucker, Trace inference, curvature consistency, and curve detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (8) (1989) 823–839.
[40] G. Medioni, S.B. Kang, A Computational Framework for Segmentation and Grouping. The Tensor Voting Framework, Prentice Hall, 2005 (Chapter 5).
[41] A.P. Witkin, Scale-space filtering, in: International Joint Conference in Artificial Intelligence, 1983, pp. 1019–1022.
[42] C.V. Rijsbergen, Information Retrieval, second ed., Dept. of Computer Science, University of Glasgow, 1979.
[43] A. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (7) (1997) 1145–1159.
[44] L. Loss, G. Bebis, M. Nicolescu, A. Skurikhin, An automatic framework for figure-ground segmentation in cluttered backgrounds, in: Proceedings of the British Machine Vision Conference 2007 (BMVC07), vol. 1, 2007, pp. 202–211.