Perceptual Grouping Based on Iterative Multi-scale Tensor Voting

Leandro Loss[1], George Bebis[1], Mircea Nicolescu[1], and Alexei Skourikhine[2]

[1] Computer Vision Laboratory, University of Nevada, Reno
[2] Space and Remote Sensing Sciences Group, Los Alamos National Laboratory
{loss, bebis, mircea}@cse.unr.edu, alexei@lanl.gov

Abstract. We propose a new approach for perceptual grouping of oriented segments in highly cluttered images based on tensor voting. Segments are represented as second-order tensors and communicate with each other through a voting scheme that incorporates the Gestalt principles of visual perception. An iterative scheme has been devised which removes noise segments in a conservative way using multi-scale analysis and re-voting. We have tested our approach on data sets composed of real objects in real backgrounds. Our experimental results indicate that our method can segment successfully objects in images with up to twenty times more noise segments than object ones.

1 Introduction

Grouping processes, which "organize" the given data by eliminating the irrelevant items and sorting the rest into groups, each corresponding to a particular object, are indispensable in computer vision. Determining groups in a given set of points or edgels can be a very difficult task, as the actual measurement of compatibility within a sub-set is not well defined. There has been considerable research work in this area over the last two decades.

In [1], Lowe discusses the importance of the Gestalt principles of collinearity, co-curvilinearity and simplicity for perceptual grouping. Ahuja and Tuceryan [2] have introduced methods for clustering and grouping sets of points having an underlying perceptual pattern. Mohan and Nevatia [3] have assumed *a-priori* knowledge of the contents of the scene (i.e., aerial images). A model of the desired features was then defined, and groupings were performed according to that model. Ullman [4] has suggested that a curve joining two edge fragments is formed by a pair of circular arcs that minimizes the integral of the square of the curvature. He also proposed a network model, but no results were shown. Clearly, elliptical curves cannot be constructed by joining only a pair of circular arcs. Also, this scheme cannot be easily generalized to a set of three or more edge fragments, and does not allow for outliers. Parent and Zucker [5] have described a relaxation labeling scheme where local kernels were used to estimate tangent and curvature. These kernels used support functions based on co-circularity. Somewhat similar kernels are used in our methodology, but applied in a different way. Ullman and Sha'ashua [6] have proposed the use of a saliency measure to

G. Bebis et al. (Eds.): ISCV 2006, LNCS 4292, pp. 1786–1797, 2006.

guide the grouping process, and eliminate erroneous features in the image. Their scheme prefers long curves with low total curvature by using an incremental optimization scheme.

More recently, Williams and Thornber [7] have proposed a probabilistic approach based on *Closed Random Walks*. In their approach, saliency is defined relatively to the number of times an edge is visited by a particle in a random walk. The main restriction assumed in that work was that the movement has to start and finish on the same edge (i.e., closed random walk). This reduces the number of paths to consider along with the complexity of the problem, however, it imposes a restriction that is not practical. Their technique was compared with five other methods in the literature and found to outperform them considering a benchmark of real objects.

The use of voting for salient feature inference from sparse and noisy data was introduced by Guy and Medioni [8] and then formalized into a unified tensor voting framework [9]. Tensor voting represents input data as tensors and interrelates them through voting fields which are built from a saliency function that incorporates the Gestalt laws of proximity and continuation. The methodology has been used in 2D for curve and junction detection and for figure completion. It has also been applied in 3D for dense reconstruction from stereo or multiple views and for tracking.

In this paper, we propose a new approach for perceptual grouping of oriented segments in highly cluttered images based on tensor voting. Specifically, we have devised an iterative scheme that removes noise segments using multi-scale analysis and re-voting. In contrast to traditional tensor voting approaches, that use hard thresholding and single-scale analysis, our method removes noise segments conservatively according to their behavior across a range of scales, and applies re-voting on the remaining segments to estimate saliency information more reliably. Our experimental results illustrate that this iterative, multi-scale thresholding scheme, coupled with re-voting, improves segmentation results remarkably.

The rest of this paper is organized as follows: Section 2 reviews the tensor voting framework, discusses the main challenges in employing tensor voting for grouping, and presents the new approach. Section 3 describes the datasets used in our experiments and the evaluation methodology. Section 4 presents our experimental results and comparisons. Finally, conclusion and directions for future work are presented in Section 5.

2 Perceptual Grouping Using Tensor Voting

2.1 The Tensor Voting Framework

In the framework proposed by Guy and Medioni [8], input data is encoded as elementary tensors. Support information (including proximity and smoothness of continuity) is propagated from tensor to tensor by vote casting. Tensors that lie on salient features (such as 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 be then extracted by examining the tensors resulted after voting.

In 2D, a generic tensor can be visualized as an ellipse. It is described by a 2×2 eigen-system, whose eigenvectors e_1, e_2 give the ellipsoid orientation, while eigenvalues λ_1, λ_2 (with $\lambda_1 \geq \lambda_2$) give its shape and size. The tensor is represented as a matrix S:

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

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 (along the curve normal), while the length of the stick represents the curve saliency (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 2-D case. Specific details about the voting generation process can be found in [8].

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:

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

Each type of feature can be characterized as: (a) *Curve* (saliency is $(\lambda_1 - \lambda_2)$, normal orientation is e_1), and (b) *Point* (saliency is λ_2, no preferred orientation). After voting, curve elements can be identified as they have a large curve saliency $\lambda_1 - \lambda_2$ (and appear as elongated tensors), junction points have a local large point saliency λ_2 and no preferred orientation (appear as large ball tensors), while noise points have low point saliency. Therefore, the voting process infers curves and junctions simultaneously, while also identifying outliers (tokens that receive 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.

2.2 Applying Tensor Voting for Grouping

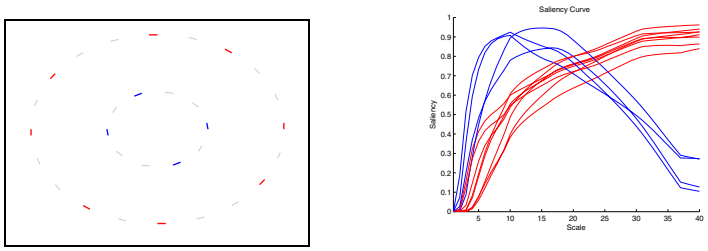
Although the tensor voting framework has only one free parameter, the scale σ, several other issues must be considered when using it for perceptual grouping

and segmentation. The voting dimensionality, the features to be used as tokens, and the encoding of the input tokens are examples of issues that need our attention. The voting dimensionality is determined by the number of features to be used to represent the information at hand. Ideally, a small number of features with maximal representation capability is desired. This raises the issue of what features to use as input tokens. Token encoding has considerable impact on the performance of the framework. In the case of edges, one can choose several different tensor representations. For example, one way would be assigning a ball tensor at each pixel of the contour. Alternatively, one can assign a stick tensor at each pixel location with position and orientation information computed according to its adjacent neighbor; since tensors have no direction, the choice of the neighbor does not matter. Finally, one could also select representatives of the contour and initialize them as stick tensors. In this study, we follow this last approach. In our case, we re-sample the contour using a fixed sampling step, and initialize the framework using the sampled pixels which are encoded as stick tensors. Their position is determined by the position of the sample and their orientation by using the direction proportional to the gradient.

Another issue that needs careful consideration is the selection of the scale parameter σ. In [10], it was found that the framework 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 the scale parameter 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. Moreover, analyzing information at a single scale can compromise or make hard the detection of structures having different sizes.

This situation can be illustrated using an image with two similar figures, one smaller than the other, shown in Fig. 1. To help the visualization, we have plotted "Scale versus Saliency" curves, thereafter called *saliency curves*. Specifically, a saliency curve is created 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. This is done in order to prevent a monotonically increasing curve. Such a situation could result from the fact that, as the voting neighborhood increases, the segment saliency 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 covers all its segments, (i.e., when scale σ is around 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 the voting neighborhood. After the larger circle reaches its saliency maximum, at scale σ around 35, its saliency curves stabilizes since there are no more segments to consider beyond this scale.



(a) (b)

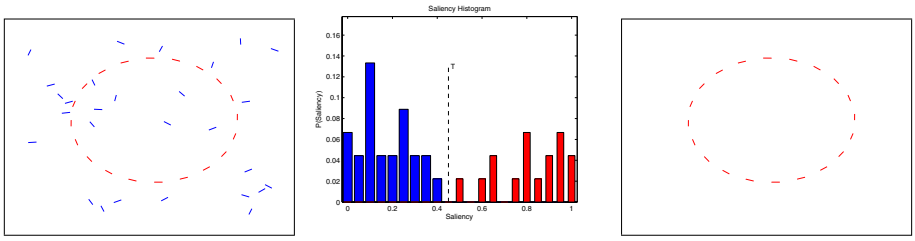
Fig. 1. Two circles having different size and saliency. (a) Segments having their saliency behavior analyzed. (b) Normalized saliency curves corresponding to the segments chosen in (a). The saliency of the smaller circle grows until the voting neighborhood covers all of its segments. After this point, the saliency of the larger circle surpasses the saliency of the smaller circle and keeps growing until it reaches its own maximum.

Another important issue when segmenting a figure from background is the choice of a threshold to filter out non-figure segments. It is reasonable to expect that if the saliencies of the figure are quite different from those of the background, then it would be easy to find a threshold value to separate them completely. Fig. 2 shows a simple example. In this example, we consider a well-formed circle surrounded only by random noise at a signal-to-noise ratio (SNR) equal to 70%. If we apply tensor voting to the image segments and plot the saliency histogram, it is easy to see that eliminating segments with saliencies below a threshold value T equal to 45% filters out noise completely.

However, this is hardly the case in practice. Let us consider, for example, the image shown in Fig. 3(a). If we apply tensor voting to its segments and plot the corresponding saliency curves or saliency histogram at a large scale, we can easily verify that there is no way to get a perfect segmentation of the figure from the background. This would be also true at different scales. Moreover, even if we choose an optimal threshold value using the saliency histogram, the number of misclassified segments would be unavoidably large (i.e., see Fig. 3(d), 3(e), 3(f)).

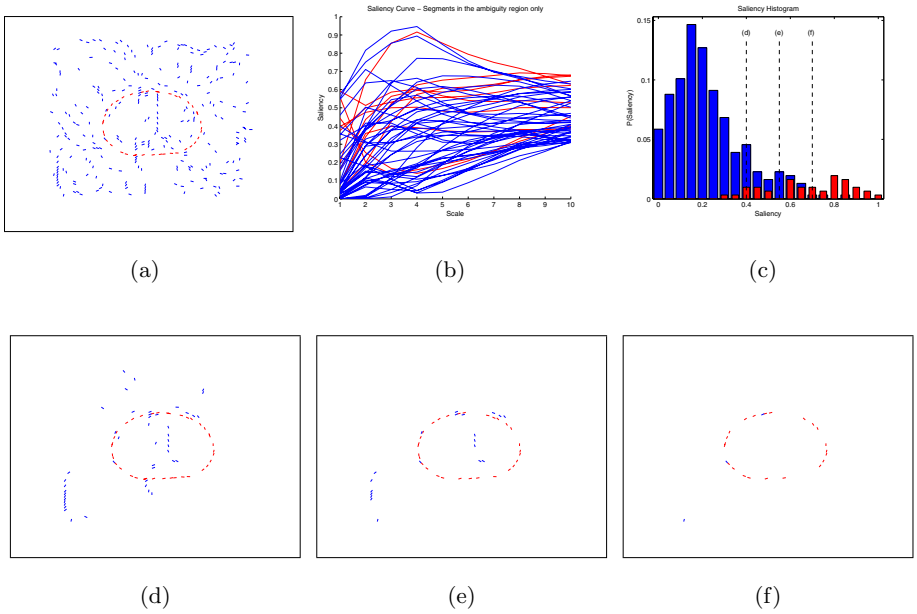
2.3 Iterative Multi-scale Tensor Voting Scheme

Aiming at eliminating the largest number of background segments while preserving most figure ones, we have developed an iterative scheme based on multi-scale analysis and re-voting. The main idea is removing segments from the image conservatively in an iterative fashion, and applying re-voting on the remaining segments, to estimate saliency information more reliably. Improvements in figure segmentation come from the fact that, after each iteration, low salient segments are filtered out and, after the subsequent re-voting sessions, the support to background segments is reduced. After a conservative elimination of segments, the difference in saliency between figure and ground segments becomes much more pronounced.



(a) (b) (c)

Fig. 2. A simple example where figure and background can be separated by using a single threshold. (a) Original image. (b) Saliency histogram and choise of threshold T. (c) Resulted segmentation.



(a) (b) (c)

(d) (e) (f)

Fig. 3. An image with SNR equal to 15% processed by different threshold values. A fixed threshold value (T), cannot result in a good segmentation at any scale. (a) Original image. (b) Saliency curves corresponding to segments of the figure and background in the ambiguity region. (c) Saliency histogram with various threshold choices. (d) Thresholding at 40%. (e) Thresholding at 55%. (f) Thresholding at 70%.

In our methodology, the conservative elimination is done by applying a low threshold value T_s, which removes almost always background segments. A new application of tensor voting follows so that a new saliency map is obtained, without considering the eliminated segments. After re-voting, the threshold value is increased to adapt to the strengthening of figure saliency due to elimination

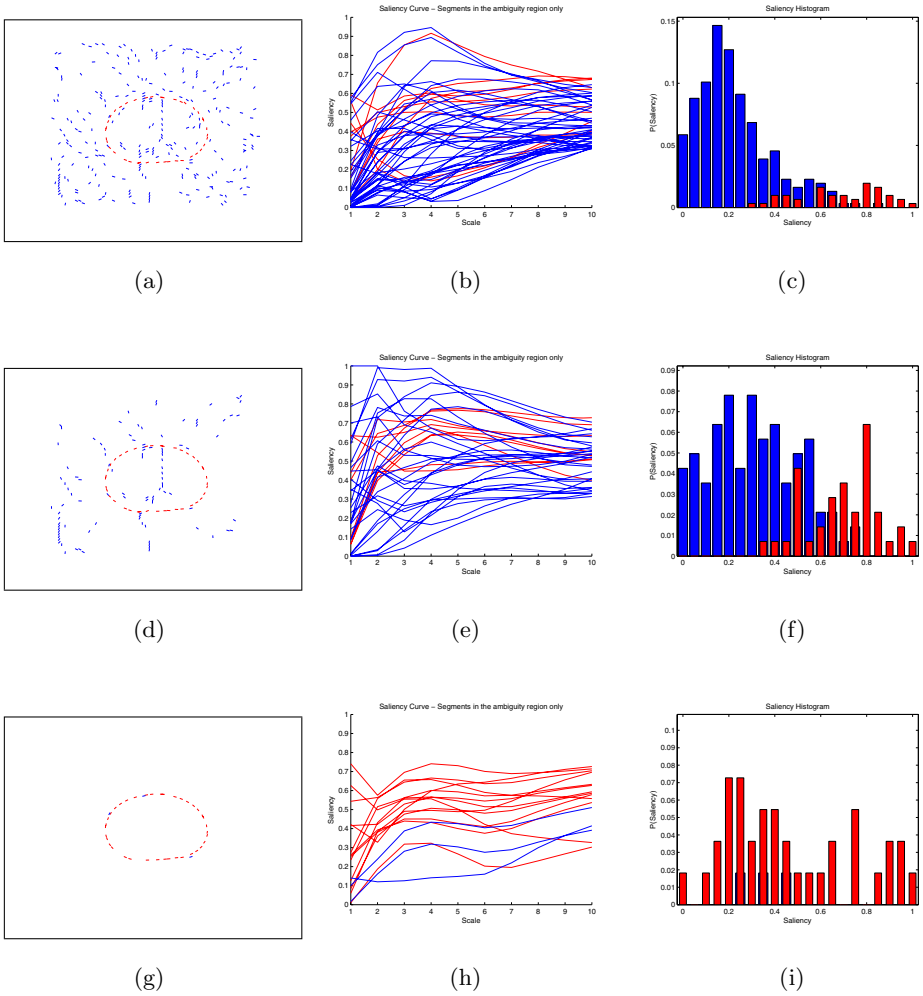


Fig. 4. Image with SNR equal to 15% processed by our iterative, multi-scale thresholding scheme. By conservatively eliminating segments, the saliency difference between figure and background starts becoming more pronounced. (a) Original image. (b) Saliency curves of background segments in the ambiguity region. (c) Saliency histogram at the largest scale. (d) Resulting image using $T_s = 20\%$. (e) Saliency curves of the segments in the ambiguity region for image (d). (f) Saliency histogram at the largest scale for image (d). (g) Resulting image using $T_s = 40\%$. (h) Saliency curves of the segments in the ambiguity region for image (g). (i) Saliency histogram at the largest scale for image (g).

of noise. In practice, we increase T_s after each re-voting session by a fixed step $Step_s$.

Multi-scale analysis is incorporated to this scheme by voting in a number of scales and thresholding according to the behavior of saliency along them.

Specifically, the saliency curve is computed by voting in different scales and computing the saliency of each segment in each scale. Segments are then eliminated if they do not present any significant peaks of saliency across a range of scales. This preserves salient segments of any size. Algorithmically, this is implemented by counting how many times the saliency curve of a segment is above the threshold T_s. If the number of times does not exceed another threshold T_σ, we consider that the segment does not have a significant saliency peak and it is eliminated. As mentioned in the previous section, we normalize the saliency curves according to the average saliency of all segments in the image.

Fig. 4 shows the behavior of figure (red) and background (blue) saliencies as the multi-scale, adaptive thresholding is applied. The image has a SNR equal to 15% (i.e., about 7 times more background segments than figure ones). The threshold values T_s goes from 10% up to 40% with $Step_s$ equal to 10%. The voting was performed with a σ ranging from 1 (5% of image size) to 20 (100% of image size). The improvements over the using a fixed threshold and single scale (i.e., naïve approach) are remarkable.

3 Datasets and Evaluation Methodology

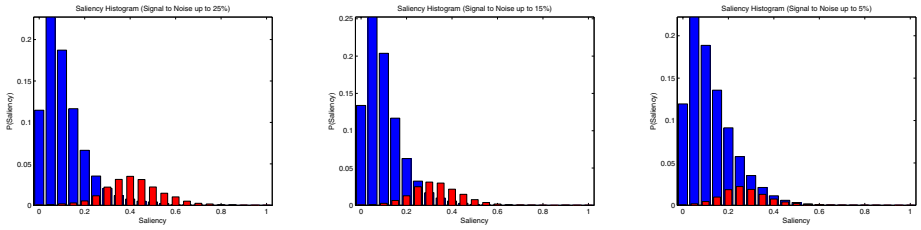
Experiments were performed based on the set of fruit and texture sampled silhouettes provided in [7]. As in [7], each benchmark image was created from a pair of sampled silhouettes belonging to a fruit or a vegetable (thereafter called figure) and textured background (thereafter called background). Nine figure silhouettes were re-scaled to an absolute size of 32x32 and placed in the middle of nine 64x64 re-scaled ground windows. We experimented with five different SNRs 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 [7]. Fig. 7 shows some examples of benchmark images at different SNRs. The total number of images used in our experiments was 405.

Quantitative evaluations and comparisons of different methods were performed using Receiver Operational Characteristic (ROC) curves, (i.e. False Positives (FP) versus False Negatives (FN)). A FN is a figure segment detected as background; conversely a FP is a background segment detected as figure. The ROC curves presented are average curves over all images in the dataset. In order to allow a direct comparison with Williams and Thornber's method (WT) [7], SNR vs FP and SNR vs FN graphs are also shown.

4 Experimental Results and Comparisons

Saliency histograms were plotted for the different SNRs used in [7] (see Fig. 5). For each histogram, we used 81 images (9 figures and 9 backgrounds).

It can be noted that, as SNR decreases, figure (red) and background (blue) start overlapping up to a point where figure becomes indistinguishable from background. The larger the overlap between figure and background, the harder is to visually separate the object in the image. At some point (for instance,

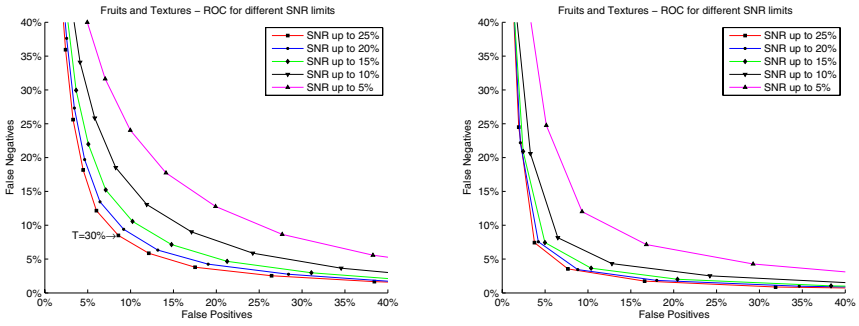


(a) SNR at 25% (b) SNR at 15% (c) SNR at 5%

Fig. 5. Saliency behaviors assuming various SNRs (red corresponds to figure and blue corresponds to background). σ was set to 20 (i.e., voting field covers the entire image). As SNR decreases, background and figure start to overlapping up to the point where figure becomes indistinguishable from background.

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 sampled textures (leaves, bricks, etc) as background instead of random noise.

Figure 6(a) shows the ROC curves assuming a single-scale and a fixed threshold. 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).



(a) (b)

Fig. 6. (a) ROC curves for different SNR limit acceptance. At SNR below 10%, the perception of the figure becomes more difficult. This is reflected by the overlapped curves in their histograms of Fig. 5, and the worst performance for the ROC curves for SNR up to 10% and 5%. (b) ROC curves for different SNR limits according to the multi-scale, adaptive threshold with Step_s equal to 5%. It is possible to note improvements in all ROC curves comparing to the curves for the naïve approach (single-scale, fixed threshold) shown in (a). In addition, the curve for SNR up to 10% is closer to the higher ones (25%, 20% and 15%), showing that the iterative, multi-scale adaptive thresholding deals better with cluttered images.

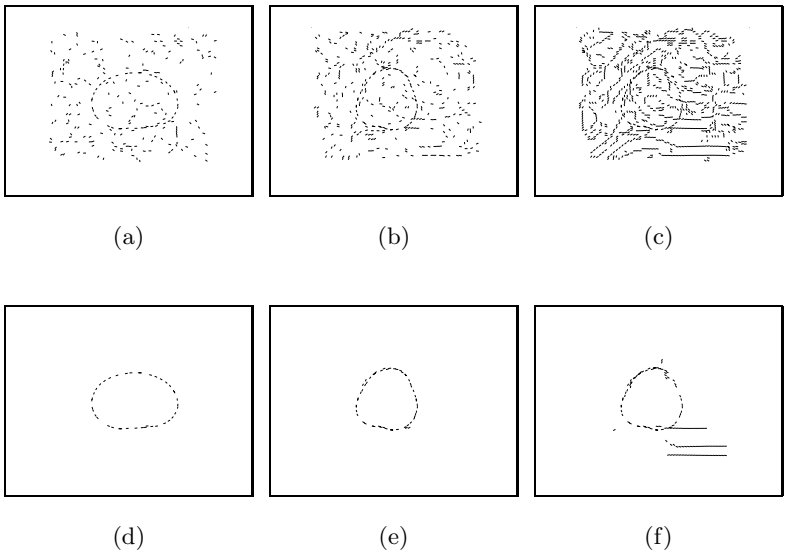
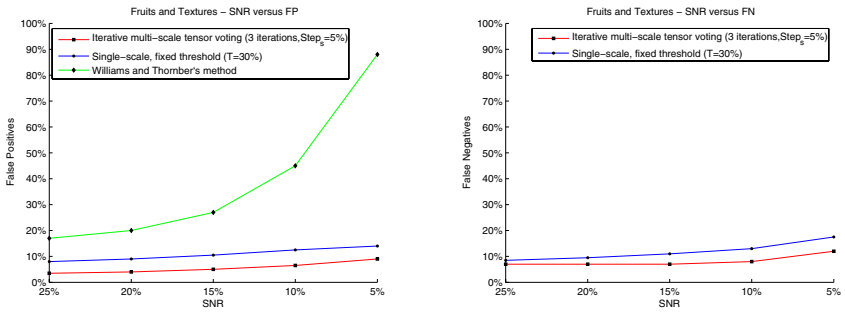


Fig. 7. Representative results based on our methodology: (a,d) avocado on bark at SNR equal to 20%, (b,e) pear on wood background at SNR equal to 15%, (c,f) pear on wood at SNR equal to 5%

Figure 6(b) shows the ROC curves using the proposed iterative, multi-scale adaptive thresholding scheme. The scale parameter σ varies from 2 to 20 (covering from 5% to 100% of the image), Step_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 by comparing Figure 6(b) to Figure 6(a). In addition, the curve corresponding to SNR equal to 10% is closer to the ones corresponding to higher SNRs (i.e., 25%, 20% and 15%). This indicates that the iterative multi-scale adaptive thresholding approach deals with cluttered scenes much better. Fig. 7 shows three representative results using the iterative multi-scale tensor voting approach.

To compare our results with those given in [7], we have created plots of SNR vs FP, shown in Fig. 8(a). Specifically, Fig. 8(a) compares results obtained using the naïve approach (single-scale, fixed threshold at T=30% - Fig. 6(a)), the best result obtained by our iterative, multi-scale tensor voting scheme (i.e., 3 iterations using Step_s=5% - Fig. 6(b)), and the results reported in [7]. Since the results in [7] 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. 8(a).

Fig. 8(b) is a plot of SNR vs FN. In this case, a direct comparison with [7] is not possible since they do not report FN rates. As it can be seen from the graphs, our iterative, multi-scale tensor voting approach shows improvements of more than 14% over [7] when SNR is equal to 25%, and improvements of almost 90% when SNR is equal to 5%, while keeping a low FN rate (i.e., eliminates noise



(a) (b)

Fig. 8. Plots of (a) SNR vs FP and (b) SNR vs FN. The iterative, multi-scale tensor voting approach outperforms the method in [7] (WT) as well as the naïve approach. Also, it has a low FN rate and performs consistently as SNR decreases).

without compromising figure information). Compared to the naïve approach, the iterative multi-scale approach improves figure vs noise discrimination by 5% on the average for all SNRs considered. The graphs also show a significantly smaller depreciation of performance as SNR decreases.

5 Conclusions and Future Work

We have presented a new approach for perceptual grouping of oriented segments in highly cluttered images based on tensor voting. Our approach uses an iterative scheme that removes noise conservatively using multi-scale analysis and re-voting. We tested our approach on data sets composed of real objects in real backgrounds. Our experimental results indicate that our method can segment successfully objects in images with up to twenty times more noise segments than object ones.

For future work, we plan to test our method on more challenging data sets including objects with open contours as well as multiple objects of the same and of different sizes. Moreover, we plan to devise a procedure for choosing Step_s and T_σ automatically at each iteration. Although the choice of Step_s did not seem to be very critical in our experiments, we feel that choosing this parameter in a more optimal way would probably help in certain situations.

References

1. Lowe, D.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence **31** (1987) 355–395
2. Ahuja, N., Tuceryan, M.: Extraction of early perceptual structure in dot patterns: integrating region, boundary, and component gestalt. CVGIP (1989) 304–356

3. Mohan, R., Nevatia, R.: Segmentation and description based on perceptual organization. CVPR (1989) 333–341
4. Ullman, S.: Filling-in the gaps: The shape of subjective contours and a model for their generation. Biological Cybernetics **25** (1976) 1976
5. Parent, P., Zucker, S.W.: Trace inference, curvature consistency, and curve detection. IEEE Trans. PAMI **11** (1989) 823–839
6. Ullman, S., Sha'ashua, A.: Structural saliency: The detection of globally salient structures using a locally connect network. 2nd. International Conference on Computer Vision - ICCV'88 (1988)
7. Williams, L., Thornber, K.: A comparison measures for detecting natural shapes in cluttered background. International Journal of Computer Vision **34** (2000) 81–96
8. Guy, G., Medioni, G.: Inference of surfaces, 3-d curves, and junctions from sparse, noisy 3-d data. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 1265–1277
9. Medioni, G., Lee, M.S., Tang, C.K.: A Computational Framework for Segmentation and Grouping. Elsevier Science (2000)
10. Medioni, G., Kang, S.B.: A Computational Framework for Segmentation and Grouping - Chapter 5. The Tensor Voting Framework. Prentice Hall (2005)