An Automatic Framework for Figure-Ground Segmentation in Cluttered Backgrounds

Leandro A. Loss, George Bebis, Mircea Nicolescu
Computer Vision Laboratory
University of Nevada, Reno
{loss,bebis,mircea}@cse.unr.edu

Alexei Skurikhin
Space and Remote Sensing Sciences
Los Alamos National Laboratory
alexei@lanl.gov

Abstract

This paper proposes an automatic framework for figure-ground segmentation of edged images in the presence of cluttered background. Our work employs perceptual grouping concepts to characterize image segments by means of their saliency, which is computed via tensor voting. The main innovation of our work is a case-based thresholding scheme which iteratively eliminates edge segments with low-saliency in multiple scales, preserving those that are more likely to belong to foreground. The key 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. We have performed extensive experiments in order to evaluate our framework both quantitatively and qualitatively, including real images from the Berkeley dataset.

1 Introduction

Perceptual grouping can be defined as the ability to detect organized structures or patterns in the presence of missing and noisy information. Using local and low-level relations, such as proximity, smoothness, continuity etc. (i.e., Gestalt principles of perceptual organization), it can reveal important information about the global organization of structures in an image. A measure of structural organization, usually referred as *saliency*, can then be used as a discriminative feature to classify image elements as figure or background, therefore, supporting figure-ground segmentation.

Several important approaches have been reported in the past including [3], [7], and [8],. Recently, Williams and Thornber [9] have proposed a probabilistic approach based on closed random walks, where saliency is defined relatively to the number of times an edge is visited by a particle in a random walk. The use of a voting process to infer salient structures from noisy and sparse data was introduced by Guy and Medioni [2] and then formalized into a unified tensor voting framework [6]. Tensor voting represents input data as tensors and interrelates them through voting fields built from a saliency function that

BMVC 2007 doi:10.5244/C.21.32

incorporates the Gestalt principles of proximity and good continuation. In the past, we have employed tensor voting in order to characterize the salience of edge segments [4]. In particular, we performed figure-ground segmentation by analyzing saliency information at multiple scales and removing low-saliency segments in an iterative fashion. This approach has shown to produce good results, however, its success relies on several parameters that were chosen manually.

In this paper, we propose a new approach for figure-ground segmentation which relies on perceptual grouping of features embedded in the tensor voting framework. Following Loss et al. [4], we perform figure-ground segmentation by analyzing the saliency of edge segments at multiple scales and removing low-salience segments using an iterative scheme. The main contribution of our work is automating the above framework by introducing a case-based thresholding scheme to eliminate low-saliency segments more effectively and criteria for stopping the iterative voting without user intervention.

Our method works by first decomposing the image edges into directional tensors and applying tensor voting in multiple scales. Then, histograms of segment saliencies are built for each scale and their modes are detected using Mean Shift (MS) [1], an adaptive gradient ascent method. Next, a case-based thresholding methodology is applied which categorizes saliency histograms into six general cases. This is performed using information about the relative position of the modes of the figure/ground distributions in the saliency histogram. Finally, low-salience image edges are eliminated according to the case detected. The whole process keeps repeating until certain stopping criteria are met.

The proposed method has several advantages including that: (1) there is no need to assume a global model for the structures present in an image; (2) it imposes no *a-priori* assumptions on the scale of the objects present in an image or their number; (3) it does not assume any distribution model for the saliency histograms; (4) thresholding low-saliency edge segments is done in an automatic fashion; (5) it yields results that are competitive with state-of-the-art methods; and (6) it can provide higher quality input to attentional methods, object detectors and recognizers.

2 Segmentation Using Tensor Voting

In the framework proposed by Guy and Medioni [2], the input data is encoded as elementary tensors. Supporting information, including proximity, smoothness and 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 (i.e., given by the tensor orientation), and their saliency (i.e., given by the tensor shape and size). Important features can be then extracted by examining the tensors resulting after voting. The method is robust to considerable amounts of outlier noise and does not depend on critical thresholds. The only free parameter is the scale factor σ, which defines the voting fields. Specific details can be found in [2] and [4]

Although the scale factor σ of the tensor voting framework has been shown to be fairly stable [6], it is subjected to the same trade-off of every scale-dependent method: small scales capture local structures while large ones capture global configurations. In real scenarios, however, structures emerge from different scales and the prediction of the optimal scale is fairly complicated. Moreover, it is impossible in general to choose a fixed threshold that would provide a good figure segmentation due to the complexity and amount of background in images. We have addressed both issues in [4] by introducing a multi-scale scheme that removes background segments conservatively in an iterative fashion, leading to an improved figure-ground segmentation.

3 Automatic Framework

Like in [4], we eliminate image edges iteratively by analyzing their behavior at multiple scales, however, the main advantage of our method is that it works in an automatic fashion. To halt the iterative process automatically but also to determine at each iteration what action is more effective for a certain type of histogram, we have devised a methodology that categorizes saliency histograms in different cases. This is done by detecting the modes of the figure/ground distributions in the saliency histogram using MS and analyzing the relative position of their peaks. If the figure/ground distributions are multimodal, then we only consider the nearest pair of figure-ground modes for classification purposes.

Considering the relative position between figure/ground modes in the saliency histogram provides strong information about the amount of overlap between figure and ground distributions. When the modes are positioned close together, then the figure distribution overlaps more with the ground distribution making segmentation hard. On the other hand, when the distance is large, then the amount of overlap is smaller making segmentation much easier. Our objective is to distinguish among these cases and apply specific actions in each case in order to automate the iterative tensor voting scheme but also make the segmentation process more effective (e.g., avoid using fixed thresholds, minimize iterations, and possibly improve segmentation results).

The histogram categories were determined during a training phase by clustering a large number of saliency histograms using MS (i.e., MS is used twice in this work; first, to detect the modes of a saliency histogram and second, to cluster saliency histograms). The saliency histograms used for training were computed from a well known dataset which contains real objects (i.e., fruits and vegetables) in textured backgrounds [9]. Using this dataset rather than a collection of various images for training allowed us to consider a much larger space of possible configurations between figure and ground distributions. We have experimentally verified that saliency histograms obtained from quite different images do resemble saliency histograms obtained from the dataset used for training (see Section 4.3).

Applying MS on the dataset of [9] yielded six clusters as shown in Fig. 1(a). Based on these results, we classify saliency histograms in six cases as shown in Fig. 1(a) (i.e., cases have been labeled for easy reference). The only parameter involved in determining the histogram categories is MS's bandwidth which was set to 0.20 both for the detection of the histogram modes as well for clustering of the histograms into cases. The cases were noted to be immune to small changes of this parameter. Depending on the case that the saliency histogram of a new image falls to, we apply one of three different actions. These actions were formulated carefully in order to preserve the conservativeness criterion proposed in [4], but also to take advantage of well separated distributions and avoid unnecessary iterations. At a given iteration, a specific case is applied if and only if this case appears in at least 50% of the image's saliency histograms (i.e., corresponding to different scales), otherwise, Case 6 is chosen for its respective more conservative action.

Case 1 is the simplest and most well-behaved case. It represents a very low salience peak (i.e., background) next to one or more highly salient peaks (i.e., figure). This case usually appears in high SNR, non-salient background, salient figure images. In this case, the action taken is a direct clustering of the elements. Clustering is performed using MS's gradient information to determine what peak each element belongs to. To be eliminated, an element must appear in the same cluster in all Case-1 saliency histograms of the image. Cases 2 and 4 represent configurations where a very low salient peak is detected, but a second peak is either detected close to the first or not detected at all. These cases are present in low SNR, non-salient background, low salient figure images. In this case, thresholding is applied at the first mode position, and an element eliminated if it is below the threshold in all Case-2 or Case-4 saliency histograms of the image. Finally, Cases 3, 5 and 6 represent configurations where no low salient peak is detected. In this case, thresholding is performed at a very low value in order to generate some change in the image for the next iteration, preserving figure integrity. Again, an element is eliminated if it is below the threshold in all Case-3, Case-5 or Case-6 saliency histograms of the image. This last action represents the implementation of [4]'s methodology.

Thresholding actions have the objective of reducing image complexity and, hopefully, bring the histogram configuration to Case 1. There are three stopping criteria based on the changes that occur in various cases. Specifically, the algorithm converges when a Case-1 saliency histogram changes to any other case. This criterion takes into consideration that once the peaks are enough far apart, any attempt to switch the configuration to a more complex one might imply that the background was successfully eliminated. The most common transitions from Case 1 are to Case 3 (i.e., no background left, figures of different saliency intensity create multiple peaks) or Case 5 (i.e., no background left, figure populates the whole saliency spectrum).

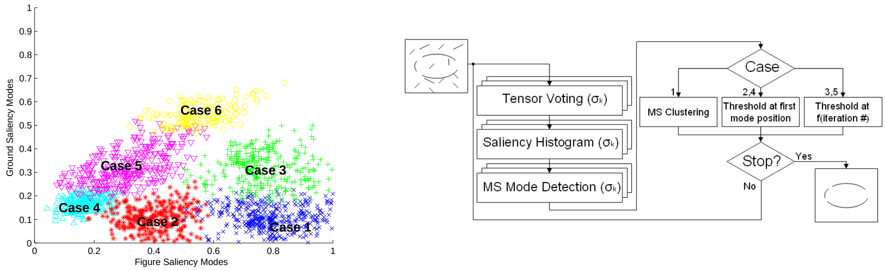
The number of elements eliminated is another criterion used to stop the algorithm. As mentioned before, MS clustering is performed if Case 1 is detected, and elements are eliminated if they belong to the least salient cluster in all scales categorized as this case. Therefore, differently from the others, this is the only case where there might be no elimination of elements. If this happens, the algorithm is assumed to have converged and stops. This captures the cases where small peaks are formed close to the low saliency region, but there is no element weak enough to be there for more than a few scales.

The third stopping criterion covers configurations that never converge to Case 1. In this case, the algorithm will keep eliminating parts of the image without stopping. Instead of choosing a maximum number of iterations, we adapt the threshold of Cases 3, 5 and 6 proportionally to the iteration index (i.e., similarly to [4]'s methodology), and assume that the algorithm has converged if this value exceeds the boundary between Cases 3 and 5. Specifically, we adapt the threshold value according to the expression $T_i = 0.06 + i \times 0.02$, where i stands for the iteration index. Fig. 1(b) shows the block diagram of our automatic framework.

4 Experimental Results

4.1 Experiment 1: Tests on Fruits and Textures Dataset

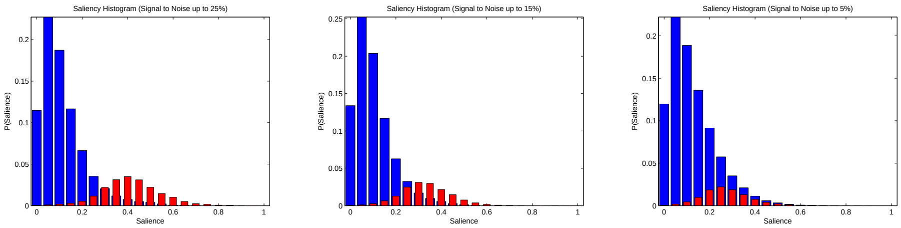
First, we experimented with the set of fruit and texture sampled silhouettes, used in [9] and [4]. This dataset contains real objects (i.e., fruits and vegetables) in textured backgrounds



(a) (b)

Figure 1: (a) The six cases obtained during training by applying MS clustering on pairs of figure-ground modes detected from a large number of saliency histograms. (b) Block diagram of our automatic framework.

(i.e., leaves, rocks, etc). As in [9] and [4], each benchmark image was built from a pair of sampled silhouettes belonging to a fruit or a vegetable and textured 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. In addition, 5 different signal to noise ratios reduced the number of ground segments proportionally to the number of figure segments. This dataset has a total of 405 images (9 figures and 9 backgrounds at 5 different SNR). Fig. 3(a)-(c) show some examples of the benchmark images at different SNR. Fig. 2 shows plots of saliency histograms. The red bars correspond to figure segments while the blue ones to background segments. Note that traditional approaches would fail in determining a threshold value that could generate a good figure-ground segmentation, especially for SNR lower than 20% (see [4]'s Fig. 3).



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

Figure 2: Saliency histograms according to SNR (red bars - figure, blue bars - ground). As SNR decreases, the overlap between the curves belonging to ground and figure increases.

Fig. 5(a) shows the results of applying our framework on the fruit and texture dataset. Each curve reveals the behavior of our approach as a function of SNR. Each curve represents average results over all the images in the dataset and each dot corresponds to the result of one iteration. Curves with fewer dots converged faster to the final result. Fig. 3 shows some visual results.

In summary, our framework was able to eliminate more than 90% of the background, preserving on average more than 85% of the figure. For images with up to 5 times more

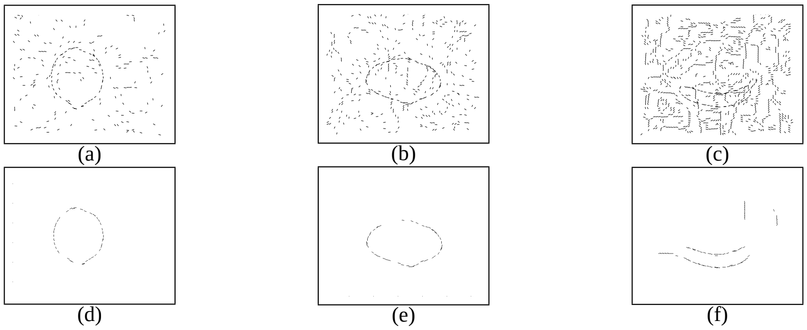
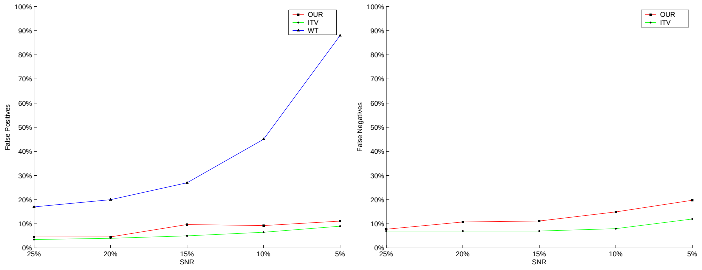


Figure 3: Visual results on fruit and texture dataset. (a) Lemon on brick with SNR at 25%, (b) red onion on leaves with SNR at 15%, (c) banana on bark with SNR at 5%, (d) lemon on brick upon 3 iterations, (e) red onion on leaves upon 6 iterations, (f) banana on bark upon 8 iterations.

background elements than figure segments, our framework eliminated more than 95% of the background, preserving more than 90% of the foreground. Highly noisy images tend to converge slower (i.e., 8 iterations) but the results obtained were very good considering the nature and conditions of them. Traditional approaches have much worse performance on this kind of images due to the large overlap between background and figure distributions (see Fig. 2c).

We have also performed comparisons between the proposed framework (OUR) and the methods proposed by Williams and Thornber (WT) [9] and Loss et al. (ITV) [4] considering the same fruit and texture dataset . The methods were compared based on the False Positive (FP) and False Negative (FN) rates according to the Signal-to-Noise Ratio (SNR). Fig. 4 shows both FP versus SNR and FN versus SNR curves. No curve is shown for WT in the case of FN versus SNR because no false negative rates were provided in [9]. ITV's curve shows results from the iteration that produced the best outcome, as if an optimal stopping criterion existed for that method.



(a) FP vs SNR

(b) FN vs SNR

Figure 4: Comparison with [9] (WT) and [4] (ITV) methods.

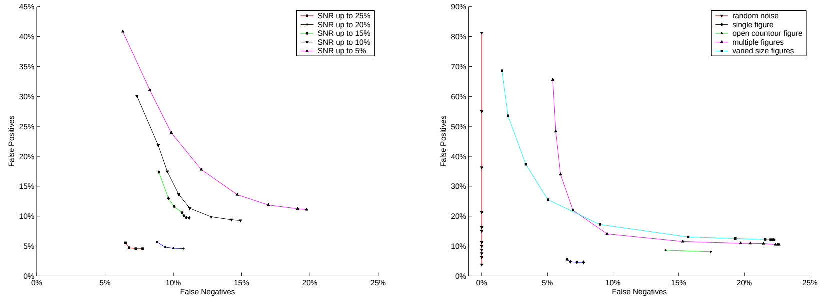
This result shows that our framework is very competitive with ITV (on average 2% worse in terms of FP and 4% in terms of FN), and in the simplest case, 12% better than

WT. Differently though from ITV, our method is fully automatic by taking decisions based on a set of predetermined cases. Our method and ITV are shown to deal better than WT with highly noisy images. This can be seen by the quasi-linear behavior of the curves corresponding to these methods. This is mainly due to the characteristic of these frameworks that eliminate elements iteratively, reducing the complexity of the image step by step.

4.2 Experiment 2: Tests on Extended Fruit and Textures Dataset

The fruit and texture dataset offers a good means of experimentation and comparison. It is composed by real images, that approximate real applications in computer vision. It is also important to notice that the background is well organized. However, since the figures are always from closed objects and displayed in the same position and scale, the benchmark lacks challenges of this sort. We have extended this benchmark by using the same fruits and textures, but incorporating new characteristics that make it more challenging and complete. In particular, we have created more test images by varying the number of fruits and their sizes, and by removing sequences of fruit segments, opening their silhouettes. We also included a set of images composed by a circle and random noise for the sake of comparison.

In summary, four new datasets were created: (1) open contours - 405 images; (2) multiple objects - 810 images; (3) varied-size figures - 810 images (4) random noise background - 200 images. Fig. 6(a,b,c) shows some examples from these new datasets. The objective of these extra datasets is to show that our method does not rely on the closure of a contour and does not make any assumptions on the size of the objects present in an image. Fig. 5(b) shows the results of our framework for each dataset. The SNR of all the images in this experiment was set to 25%. Fig. 6 shows some visual results.



(a)

(b)

Figure 5: (a) Results on the fruit and texture dataset. (b) Comparison of our framework using different datasets.

These results illustrate that our framework has no bias toward contour closure, size or nature of foreground. On average, our framework eliminated more than 90% of the background, preserving more than 85% of the foreground. The higher number of iterations needed for multiple and varied-size figures is due to the higher number of background elements in those images. In the case of backgrounds made of random noise, our system eliminates more than 97% of the background elements, preserving 100% of foreground. Still, most of the background preserved lies close and aligned to the figure segments.

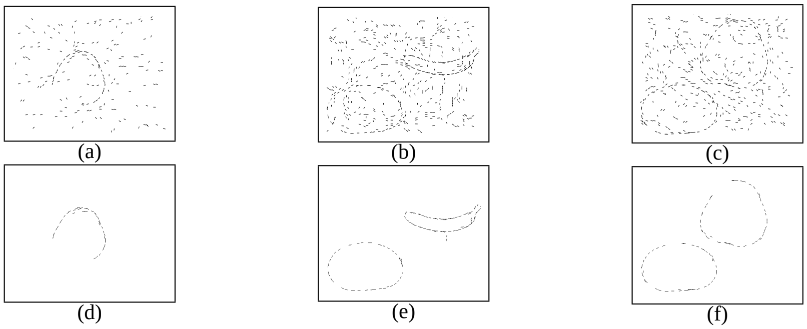


Figure 6: Visual results on extended datasets with SNR at 25%. (a) Incomplete pear on water, (b) avocado and banana on bark, (c) avocado and larger pear on fabric, (d) incomplete pear on water upon 2 iterations, (e) avocado and banana on bark upon 7 iterations, (f) avocado and larger pear on fabric upon 7 iterations.

4.3 Experiment 3: Tests on Real Images

Finally, we tested our framework on real images taken from the Berkeley Segmentation Dataset [5]. This dataset contains ground truth information obtained by asking several human subjects to identify the most important segments in each image. For consistency, we considered only those segments found by two or more human subjects. Fig. 7(top) shows three examples. Since our framework works with segments, we have pre-processed each image using the Canny edge detector (Fig. 7 (second from the top)). Edges smaller than 10 pixels were removed.

To assess the results of our method, we chose the parameters of the Canny edge detector such that it preserved as many segments present in the ground truth as possible. This is because our current implementation of tensor voting can not account for missing edges by filling the gaps between edges. Therefore, if a ground truth edge is not among the edges found by the Canny edge detector, it cannot be found by our method. Obviously, this assumption can be relaxed by employing a more powerful version of tensor voting called ”dense tensor voting” [6] which can fill in gaps between edges.

Fig. 7 (third from the top) shows the ground truth segments superimposed on the edge image. For visualization purposes, blue represents segments present both in the ground truth and edge image, red represents segments present in the ground truth but not in the edge image, and gray represents edges present in the edge image but not in the ground truth. For evaluation purposes, we are interested in maximizing the number of blue segments that our method can keep while minimizing the number of gray pixels. Red pixels cannot be detected by our method since they are not present at all in the input to our algorithm.

Figure 7(bottom two) shows the resulting salient segments found by ITV and OUR methods. In this case, all blue segments correspond to True Positives (TP), all red segments correspond to False Negatives (FN), while all green segments correspond to False Positives (FP). TPs were computed by comparing the salient segments found by our method to those present in the ground truth segments which were also present in the edge image. FNs were the ground truth segments present in the edge image that were eliminated by our method. Finally, FPs were salient edges kept by our method that were not present in the ground truth. Table 1 shows specific results for each of the images shown in Fig. 7.

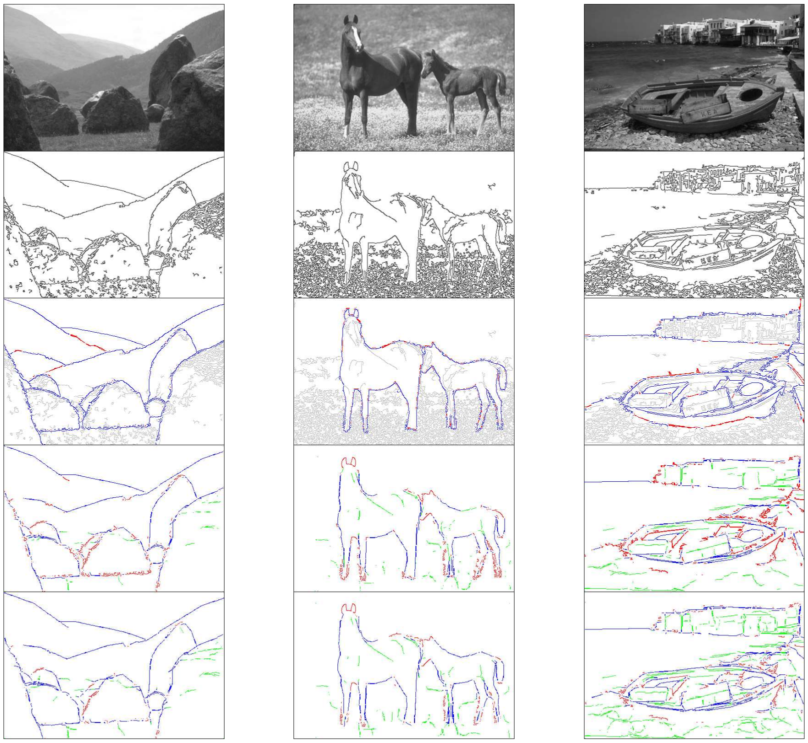

(a) Test Image #36
(b) Training Image #146
(c) Training Image #180

Figure 7: Images from the Berkeley segmentation dataset. From the Top: original image, Canny edge image, Ground truth superimposed over the edge image, result of ITV, result of OUR method. Refer to text for color scheme.

These results reveal the high accuracy of our method in finding meaningful segments in an image. It should be noted that FPs should not be regarded as errors since most of them correspond to edges belonging to some meaningful salient curves. The results using the Berkeley dataset are very encouraging since although our framework was designed mainly using synthetic data, it generalizes very well on real images.

5 Conclusions and Future Work

We have proposed an automatic framework for figure-ground segmentation based on an iterative, multi-scale tensor voting scheme. Our experimental results and comparisons indicate that our framework is very competitive with state-of-the-art approaches, with the advantage of being automatic. The key to automating the segmentation framework was the case-based thresholding methodology proposed here. For future work, we plan to formalize our case-based thresholding into a reasoning framework by employing fuzzy

Table 1: Segmentation results using images from Berkeley dataset [5].

	ITV			OUR		
Berkeley Image	TP	FN	FP	TP	FN	FP
Test #36	55%	29%	5%	88%	12%	18%
Training #146	57%	43%	6%	76%	24%	6%
Training #180	38%	46%	13%	75%	25%	29%

rules. Moreover, we plan to improve the time requirements of iterative voting scheme by updating the votes at each iteration instead of recomputing them from scratch.

References

[1] K. Fukunage and L.D. Hostetler. The estimation of the gradient of a density function, with application in pattern recognition. *IEEE Trans. Information Theory*, 21:32–40, 1975.

[2] G. Guy and 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):1265–1277, 1997.

[3] L. Hérault and R. Horaud. Figure-ground discrimination: A combinatorial optimization approach. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 15:899–914, 1993.

[4] L.A. Loss, G. Bebis, M. Nicolescu, and A. Skourikhine. Perceptual grouping based on iterative multi-scale tensor voting. *Lecture Notes in Computer Science 4292 - ISVC'06*, 2:1786–1797, 2006.

[5] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[6] G. Medioni, M.-S. Lee, and C.-K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier Science, 2000.

[7] S. Sarkar and K. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Proc. IEEE Conference on Computer Vision and Pattern Recognition - CVPR'96*, pages 478–483, 1996.

[8] S. Ullman and A. Sha'ashua. Structural saliency: The detection of globally salient structures using a locally connect network. *2nd. International Conference on Computer Vision - ICCV'88*, 1988.

[9] L. Williams and K.K. Thornber. A comparison of measures for detecting natural shapes in cluttered background. *International Journal of Computer Vision*, 34(2/3):81–96, 2000.