Combining Region and Edge Cues for Image Segmentation in a Probabilistic Gaussian Mixture Framework

Omer Rotem
The Engineering Faculty
Tel-Aviv University, Israel
rotem.omer@gmail.com

Hayit Greenspan
The Engineering Faculty
Tel-Aviv University, Israel
hayit@eng.tau.ac.il

Jacob Goldberger
School of Engineering
Bar-Ilan University, Israel
goldbej@eng.biu.ac.il

Abstract

In this paper we propose a new segmentation algorithm which combines patch-based information with edge cues under a probabilistic framework. We use a mixture of multiple Gaussians for building the statistical model with color and spatial features, and we incorporate edge information based on texture, color and brightness differences into the EM algorithm. We evaluate our results qualitatively and quantitatively on a large data-set of natural images and compare our results to other state-of-the-art methods.

1. Introduction

Image segmentation is essentially the partitioning of an image into several constituent components, or objects. The basic task of image segmentation is, then, to assign each pixel in the image to one of the image objects. Segmentation is often not a well-defined task, since the level to which an image is subdivided is determined by the application at hand and it is also a subjective task: people may subdivide a given image differently, using a different number of objects, or to different levels of resolution. In general, autonomous segmentation is one of the most difficult tasks in image processing [6].

Traditionally, work has been focused on either *region-based* approaches or *edge-based* approaches. In edge-based processing, gradients are used to detect local edge fragments which are defined by sharp localized changes in some image feature. The edge fragments can be linked in further processing in order to identify extended contours. The main difficulties with such approaches are the resultant false and broken edges [12, 1]. Edges or edge fragments are generated in locations where no real boundaries exist, due to the local nature of the process and its sensitivity to noise artifacts.

Region-based approaches focus on grouping by similarity in feature space. They include classical approaches such as region-growing, as well as probabilistic clustering based approaches such as K-means clustering and GMM based approaches [4]. Traditional GMM-based methods are intensity based pixel-wise clustering schemes. These schemes have been recently augmented by incorporating spatial features within the feature vector, so as to include spatial context within the model and to provide smooth segmentation maps [2, 7]. Further recent extensions include the utilization of a large set of Gaussians in order to enable the representation of non-convex regions, in cases in which an initial segmentation to an a priori defined set of segments is possible. Such is the case in MRI brain segmentation [8]. Current (region-based) parametric clustering approaches are based on intensity and color features, so far lacking an extension to additional features, such as texture. Gaussians in the model are often generated across boundaries (in cases in which the region color is similar enough) - thus causing a blurring of region boundaries in the output segmentation map, or a complete loss of object segments.

Current work searches for models that combine region and edge cues. Promising results were recently shown on natural image datasets, via pairwise clustering graph-based techniques [15, 13]. Fowlkes et al. [5] used pairwise affinity functions to model the relationship between any two pixels in the image using both region-based features (brightness, color and texture) and gradient-based features derived from these. A large gradient along a straight line connecting a pair of pixels suggests that there is an intervening contour between them and therefore that they do not belong to the same segment. The authors used a large data-set of natural images to optimize all the free parameters in their algorithm. The final segmentation stage is done by the Normalized-Cuts method, which is a global cri-

terion for partitioning a graph, measuring both the total dissimilarity between the different groups as well as the total similarity within the groups [15]. Sharon et al. [14] used segmentation by weighted aggregation (SWA), along with techniques from algebraic multigrid, to find image segments. Here too, a Normalized-Cut type function is optimized to evaluate the saliency of a segment, on a preformed irregular pyramid structure that represents the input image.

In the current work, we propose an extension to Gaussian mixture based modeling, that enables the combination of region and edge cues within the probabilistic GMM framework. The inclusion of edge information within the model generation process, along with a Gaussian-tying step, provides the ability to model non-convex segments in the image as linked sets of Gaussians. Each Gaussian represents a localized coherent neighborhood. The edge information ensures that Gaussians are tied within-object only. Gaussians are formed as needed (based on the image content), thus there is no need to set a priori the number of components. Gaussians are linked (tied) so as to represent an entire image segment. The framework combines the advantage of local modeling with a globalized optimization. The transition from pixels to localized Gaussians is computationally attractive. A validation of the proposed methodology is presented, both qualitatively and quantitatively, on a large set of natural images (COREL) and a comparison is made to additional state-of-the-art methods.

2. The GMM Region-Based Model

The GMM approach for image segmentation [2, 7], groups the data points of an image into k segments, each assigned with a Gaussian distribution. Probabilistic image modeling and segmentation is performed in the color and position (x, y) feature space. The distribution of a d-dimensional random variable is a mixture of k Gaussians, if its density function is:

$$f(x) = \sum_{j=1}^{k} \alpha_j \frac{1}{\sqrt{(2\pi)^d |\Sigma_j|}} \exp\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1}(x-\mu_j)\}$$
(1)

where α_j are probabilities of occurrence of each Gaussian and μ_j, Σ_j, are the mean and the covariance matrix of each Gaussian respectively. An immediate transition is possible between the GMM representation and probabilistic image segmentation. Each pixel of the original image is affiliated with the most probable Gaussian and a localized segmentation map, a map of convex superpixels, is formed. The EM algorithm, which is utilized to estimate the model parameters,

works by first estimating the probability of each pixel to belong to any of the k Gaussians (E-step), and then optimizing the GMM parameters to best fit the current segmentation (M-step). The E-step is composed of computing the posterior probability w_{tj} of each feature vector x_t to belong to the j-th component:

$$w_{tj} = \frac{1}{z}\alpha_j f(x_t|\mu_j, \Sigma_j)$$
(2)

where z is a normalization scalar. The end result of the region-based GMM is a mixture of k components, each consisting of similar points in feature space (e.g. color). However, in natural world images frequently points of similar features may actually belong to different objects in the image plane. Since the color-based model does not take into account edge-type information, the model cannot detect the existence of a boundary between spatially proximal objects.

Figure 1. Combining local features and edges in the EM learning process. Edge map is overlaid in white. Gaussians in the trained model are shown as ellipses. First, the weights from point t to each Gaussian j (w_{tj}) are computed. For those Gaussians with high posterior probability (the three nearest ones in this example), a straight line is drawn and the function E_{tj} determines if it intersects an edge. Such is in fact the case for one of the Gaussians and the corresponding w_{tj} is set to zero. There are two remaining Gaussians for which no edge is intersected.

3. Combining Region and Edge Cues within the GMM Framework

In this study we propose an extension to the GMM framework, which introduces the utilization of edges into the algorithm by setting a zero posterior probability to a cluster which has an edge between its center and a given pixel t. In a preprocessing stage, an edge detector is used to extract edge information from the

image. The edge information is utilized in the Expectation step of the EM process, in training the GMM model, as follows: An edge existence score E_{tj} is computed at each pixel point, t, to each of the Gaussian centroids, j. A straight line is 'drawn' between point t and μ_j in the image plane. If that line intersects an edge in the edge map, then $E_{tj} > 0$. If there are no edges between the two points then $E_{tj} = 0$. For reasons of efficiency, all the weights w_{tj} at a given point (to each Gaussian in the model) are computed first, and then E_{tj} is checked only for those Gaussians with high enough posterior probability. The modified E-step, is therefore as follows:

$$w_{tj} = \begin{cases} \frac{1}{z}\alpha_j f(x_t|\mu_j, \Sigma_j) & \text{if } E_{tj} = 0 \\ 0 & \text{else} \end{cases} \tag{3}$$

where x_t is the feature vector of pixel t and w_{tj} is the posterior probability that pixel t belongs to the j-th component. The Maximization step of the EM algorithm is not altered. The modified EM process is repeated to convergence to extract the final GMM representation of the image. In a follow-up labeling phase, each point in the image is affiliated with the most probable cluster. Here, again, the edge information plays an important role. The modified EM process is illustrated in Figure 1.

3.1. Dynamic Gaussian Allocation

The EM algorithm is initialized with a small number of Gaussian components (via K-means for example, with an arbitrary small k value). New Gaussians are added during the Expectation step in case there are pixels that are not well modeled by any existing Gaussian (i.e. $w_{tj} = 0$ for all j). Such pixels may often be found within small object regions, for which no specific Gaussian is initially allocated. Within-object pixels cannot be linked to any existing Gaussians without crossing an edge (boundary of the object), as is illustrated in Figure 2. As such, they cannot be linked to any of the available clusters.

A dynamic Gaussian allocation is proposed: a new cluster is added at the problematic pixel point. Initial new Gaussian parametrization are the pixel features and its mean spatial position. The covariance of the new Gaussian is set to be the average covariance over all the Gaussians in the image. Gaussian elimination is also enabled: a Gaussian component is eliminated if it is redundant, i.e. it is supported by less points than a minimum threshold. This is done at every iteration independently. Note that adding a new Gaussian is done in the E-step and Gaussian elimination is done in the M-step. The dynamic allocation process ensures a
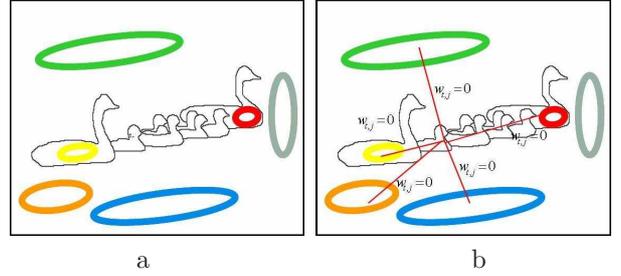

Figure 2. Dealing with closed edges. (a) Initialization with six Gaussians (ellipsoids), (b) A point in the image is shown from which all straight lines to Gaussian centroids intersect an edge. In such a scenario, the sum of all weights is zero and a new Gaussian is formed.
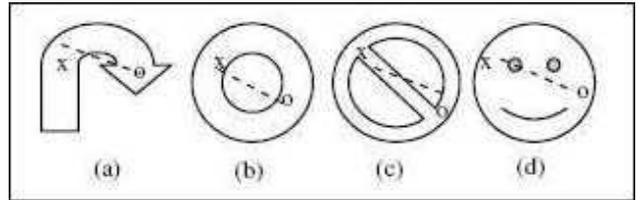

Figure 3. Examples of non-convex scenarios that require multiple blobs per object.

distribution of Gaussians that is adaptive to the given image content. Multiple Gaussians are used in more complex areas of the image, while a small number of Gaussians is used across the more uniform areas. Note that the dynamic Gaussian allocation step, in addition to providing the spatial distribution of the Gaussian components, provides an automated means for defining the appropriate *number* of Gaussians within the model.

3.2. Gaussian Tying

When attempting to segment natural images we regularly deal with non-convex objects. In such cases, it is possible to have an edge separating a point that belongs to the Gaussian representing the object and the Gaussian centroid. An example is shown in Figure 3(a). A point 'o' at the head of the arrow obviously belongs to the Gaussian representing the arrow, with its centroid 'x' located somewhere in the middle of the arrow. However, there are a few edges between them, causing the algorithm to decide that point 'o' is not part of the arrow. Additional examples include objects with holes ((b) and (c)) and objects within the main object (d).

The Gaussian allocation step often results in a multi-Gaussian allocation per object region (in particular in non-convex cases). A Gaussian-tying step attempts to automatically link together such Gaussians. Gaussians with similar features that have no edges between their centroids, are provided a unique label (linked) thus enabling the representation of higher-level object regions.

An illustration of the Gaussian-tying step is shown in Figure 4. Note that the mean-shift segmentation algorithm [3] also effectively merges clusters though in an indirect way.
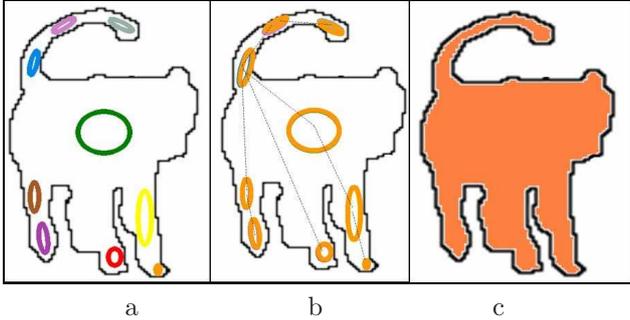

Figure 4. Dealing with non-convex objects: Multiple Gaussian tying. (a) Multiple Gaussians generated within a non-convex object; (b) Gaussian-tying: any two Gaussian centroids with similar features, that have no edge between them, are linked together; (c) Segmentation result.

The Gaussian allocation and linking procedure works well for an ideal scenerio in which the input edge-map is very accurate and provides a reliable representation of the scene. However, when we deal with less than optimal edge maps (as is normally the case), the provided object contours are often not complete (contain 'holes') and many non-related (noise) edges are included. The issue of incomplete boundaries may lead to 'bleeding', i.e. the tying of Gaussians which do not belong to the same object. The issue of extra, noninformative edges, may lead to an oversegmentation of the scene. In order to control the 'bleeding' issue, special measures are taken to determine the degree of similarity between Gaussians prior to their linkage. A fixed threshold is used to decide whether two Gaussians are 'similar enough' to be tied together. This in effect defines the 'within-object' variability. An increased threshold may enable the merging of regions which are more varied in their feature content (e.g. contain different shades of color, shadow or texture). In such cases, the edge map information is highly reliable and may serve as the dominant criteria in the segmentation. An automated scaling of the threshold is suggested based on a measure of the quality and reliability of the edge detector used, as defined in the following section.

4. Probabilistic Edge Detection

We extend the algorithm to enable the utilization of *probabilistic edge maps*. Such maps may provide better information in modeling the relationship between color patches and their boundaries. Given a probabilistic
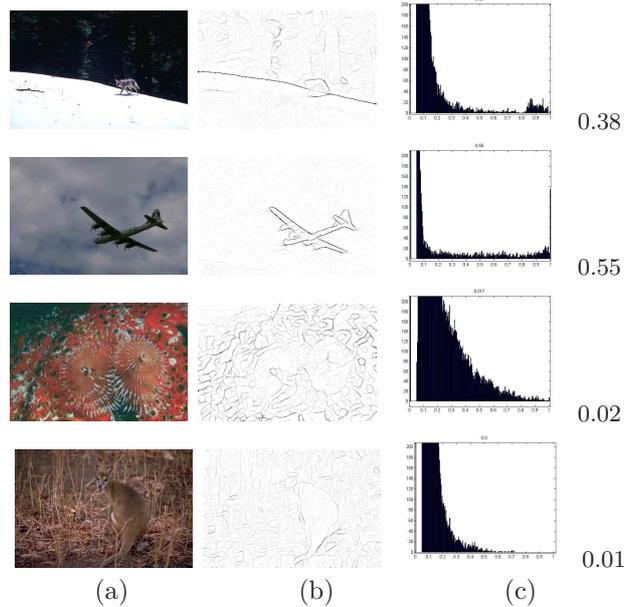

Figure 5. Examples of edge quality scores. (a) Input image, (b) Berkeley edge detector probabilistic map, (c) Corresponding histogram. The number beside each histogram is the quality score Q for that edge map.

edge map, the overall probability of having an edge between any pixel t and Gaussian centroid j is computed as the maximal probability, $p(E_{tj})$, of an edge along the straight line connecting the pixel and the Gaussian centroid. This value is used as a penalty measure for weighing the original E-step weight between point t and μ_j:

$$w_{tj} = \frac{1}{z}\alpha_j f(x_t|\mu_j, \Sigma_j)(1 - p(E_{tj}))^{20} \qquad (4)$$

Equation (4) presents the soft combination (the number 20 was determined empirically) of region-based and edge-based information, as used in this work.

In the current work we utilize edge information via existing edge maps algorithms. We use the Canny edge detector [1], as well as the more recently proposed Berkeley edge detector [9], which is based on much effort in studying human-based edge markings for the natural image archive of interest [10]. The Canny detector works on intensity images and uses brightness cues only, to generate a binary edge map. The Berkeley edge detector [9] combines color, brightness and texture cues to provide a probabilistic edge map, where for each pixel in the image a probability for being an edge, or contour is computed.

The binary version of the proposed algorithm is used for a binary edge input, such as the Canny edge input, while the extended probabilistic version of the algorithm is used for a probabilistic edge map, such as
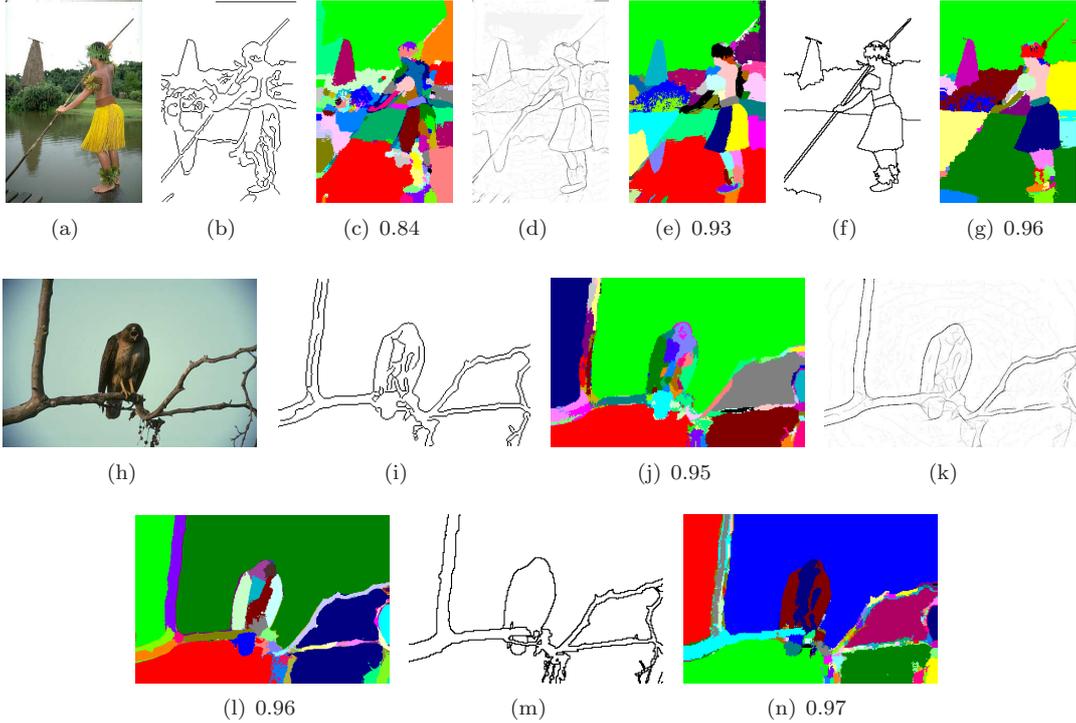
Figure 6. Edge maps and Segmentation results: (a) Input image, (b) Canny edge detector, (c) Canny+our method, (d) Berkeley edge detector, (e) Berkely+our method, (f) Manual segmentation, (g) Manual segmentation + our method. A second example is shown in (h)-(n).

given by the Berkeley edge detector. A given binary edge map undergoes a pre-processing step prior to using it in the model. This includes edge thinning, 4-neighborhood connectivity masks for ensuring closed contours, and the elimination of short edges. For a probabilistic edge input, an additional processing step is required. In several stages of the algorithm, such as adding a new Gaussian and Gaussian-tying, a hard decision is required (does an edge exist). Such binary decisions are done based on thresholding the probabilistic map into a deterministic one. A fixed threshold above which an edge is defined has proven to give poor results. We therefore suggest an adaptive scaling of the threshold, per image, based on an edge-map quality measure as defined next.

Given a probabilistic edge map histogram h, we measure the ratio of the number of pixels with high edge probability [0.8-1.0] to the number of pixels with edge probability in the mid-range (e.g. [0.2-0.8]):

$$Q = \frac{\sum_{i=0.8N}^{N} h_i}{\sum_{i=0.2N}^{0.8N} h_i} \qquad (5)$$

where N is the number of bins in the histogram. A more accurate edge-map is one in which the number of

pixels of high-confidence (e.g. probability above 0.8) is relatively large, and the number of pixels in the mid-probability range is low. A very noisy, low-probability edge map, will result in low values of Q (approaching zero), while a more strongly-defined edge map (often seen as a bimodal histogram) will result in values around 0.5. Figure 5 shows several examples of probabilistic edge maps and their respective histograms. The number beside each histogram is the quality score Q for that edge map. The top two rows in Figure 5 are examples of strong edge maps. The edge map itself can be seen to be 'clean' and the corresponding histogram is of a bi-modal nature: there is a strong peak at the low probability range ($p < 0.2$), a small peak in the high probability range ($p > 0.8$) and very few pixels in the mid-range ($0.2 < p < 0.8$). The two bottom rows in Figure 5 present examples of unreliable edge-maps. In one case, too many pixels are in the mid-range (ambiguity range) which result in a very 'dirty' edge map, containing too many possible edges that might confuse our algorithm. This is often the result of the edge detector failing to deal with the presence of texture in the image. In a second scenario, not enough distinct edge information is given. Although there are only a few

pixels with medium probability ($0.2 < p < 0.8$) there are in fact no pixels with high probability ($p > 0.8$) at all, giving the edge map a zero score. The example shown is of an animal camouflage, which occurs frequently in natural images.

The Q measure is used to scale the threshold that is required in determining a hard decision for an edge. We use the following definition: th = 0.85*(1-Q). This results in a threshold of around 0.38 for a high quality map, and around 0.85 for a low quality map.

The Segmentation Algorithm

- Create a probabilistic edge map (e.g. using the Berkeley edge map algorithm.)

- Initialize the GMM model using the K-means algorithm.

- Perform EM iterations based on both region and edge information:

    - E-step: For each pixel, find the posterior probability for each Gaussian based on both the features and the existence of edge between the pixel and the Gaussian center. If there is no relevant Gaussian, create a new Gaussian centered at the pixel.

    - M-step: Update the Parameters of each Gaussian based on the E-step. If the number of pixels that support a Gaussian is less than a threshold, eliminate the Gaussian.

- Link together Gaussians that are similar in feature space and are not separated by an intervening edge.

Execution speed of the proposed segmentation algorithm is similar to the Berkeley Segmentation Engine (BSE). The added edge information does not add much complexity to the standard GMM-EM algorithm since we check the existence of an edge between a point t and a small number of candidate Gaussians (E-step). Therefore, the complexity does not depend on the size of the image or the number of Gaussians and it is constant in this regard. The tying step is also very efficient since we are only checking those Gaussians that are similar enough in the feature space (in their mean values). However, the complexity does increase simply because we are using more Gaussians in the model. This results in more computations during the M-step.

5. Experimental Results

The proposed algorithm is evaluated using the Berkeley Segmentation Database [10] which is a dataset of natural scene images. This database has expert manual markings which serve as the ground-truth. In the following we provide qualitative as well as quantitative results. A standard non-parametric measure of clustering quality is the Rand index [11]. Several variants of the Rand index were recently introduced for measuring the quality of segmentation algorithms (e.g. [16]). We used the human segmentations as ground truth and used the (normalized version of) the Rand score to evaluate our results. For the sake of completeness, the variant of the Rand matching score that we used is as follows: Let C_1 stand for the true clustering of the n pixels and C_2 stand for the clustering in question of those points. Then:

$$\text{Rand Score} := \frac{1}{2} \cdot \left(\frac{N_{0,0}}{N_0} + \frac{N_{1,1}}{N_1} \right) .$$

where $N_{0,0}$ is the number of pairs of points that do not belong to the same cluster neither in C_1 nor in C_2, N_0 is the number of pairs of points that do not belong to the same cluster in C_1, N_1 is the number of pairs of points that belong to the same cluster in C_1 and $N_{1,1}$ is the number of pairs of points that belong to the same cluster both in C_1 and C_2.

Qualitative and quantitative results are shown in Figure 6. We focus here on a comparison across different edge-maps as input to the segmentation algorithm. Two example cases are shown. In each one, the Canny edge detector, Berkeley edge detector and the human-expert edge markings are presented, along with corresponding segmentation results. The manual edge markings most closely reflect our visual understanding of the scene, the edges are accurate thus enabling a high degree of accuracy in the segmentation results. This result can be regarded as the ground-truth result per case. Of the two automated edge-detectors used, the Berkeley edge-detector in general handles natural images better than the Canny edge-detector, mostly due to its ability to deal with texture. This fact is reflected in the segmentation results, which seem for the most part to be more pleasing visually, and achieve a higher Rand score.

In the next experiment, we evaluated the segmentation results of the proposed algorithm on an extended database. A training set was used to set the probabilistic edge parameters (using Berkeley edge-detector). Once set, a fixed set of parameters was used for the testing of 100 images. Examples of segmentation results are shown in Figure 7. A ground truth segmentation image was randomly selected per image (from the

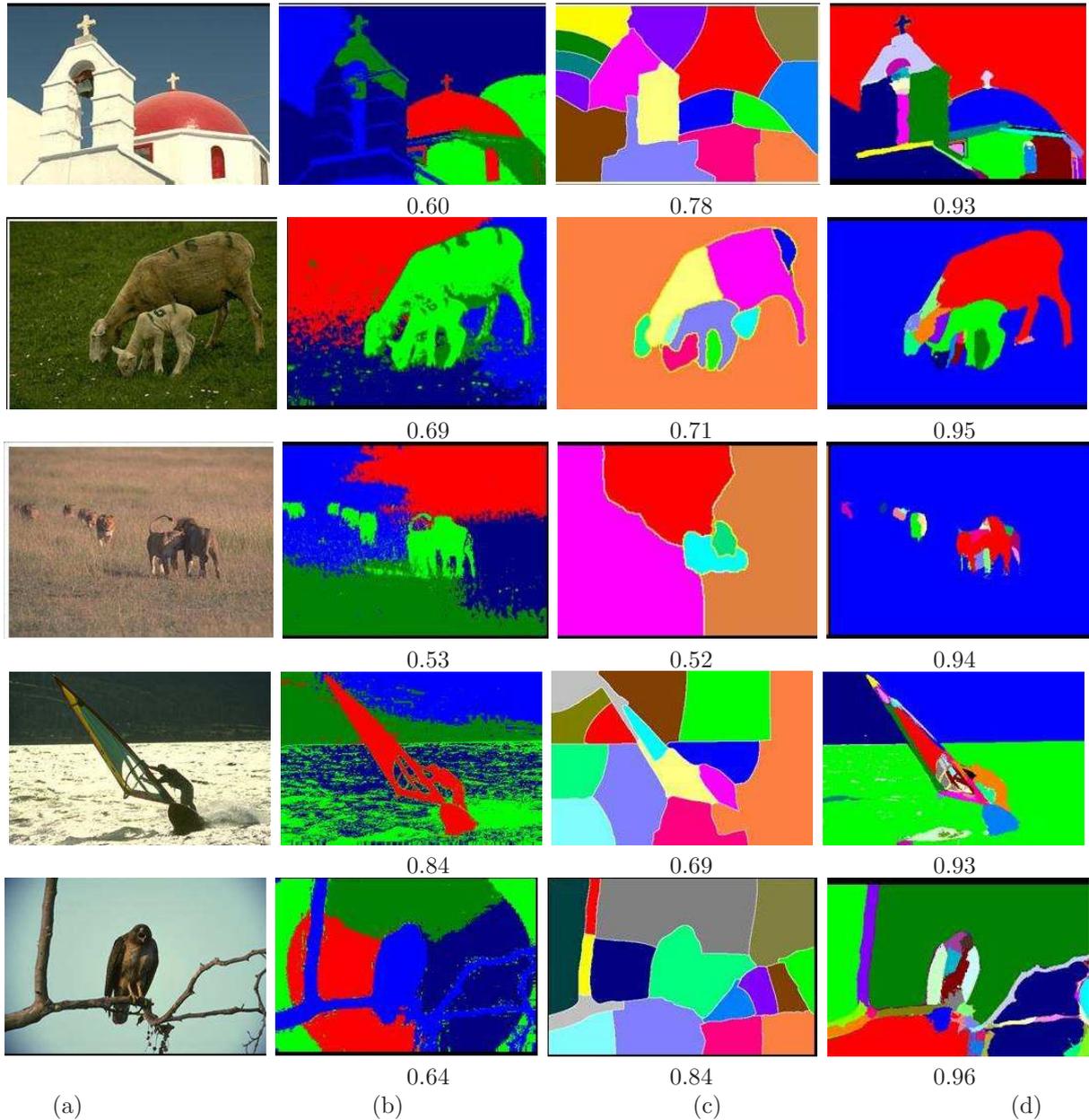|			
	0.60	0.78	0.93
	0.69	0.71	0.95
	0.53	0.52	0.94
	0.84	0.69	0.93
	0.64	0.84	0.96
(a)	(b)	(c)	(d)

Figure 7. Example segmentation results: (a) Input image; (b) GMM segmentation results [7]; (c) BSE segmentation results [5]; (d) Proposed algorithm segmentation results. Rand score per image. First 3 examples are from the training set and the last two are from the testing set.

set of manually marked images) and was used for the Rand evaluation. A comparison is made to the GMM segmentation which is region-based only (color and x,y features) [7], as well as to the Berkeley Segmentation Engine (BSE) as described in [5]. Both qualitative and quantitative results indicate the strength of the proposed algorithm.

A statistical summary of the results (mean Rand score) is presented in figure 8. The proposed scheme compares favorably with other existing state-of-the-art segmentation algorithms.

6. Conclusion

This work presents a methodology for incorporating (probabilistic) edge cues along with region cues within the GMM framework. We believe this is a major step forward in augmenting probabilistic modeling, for image representation and segmentation tasks. Previously, the GMM was shown to work with color and spatial po-
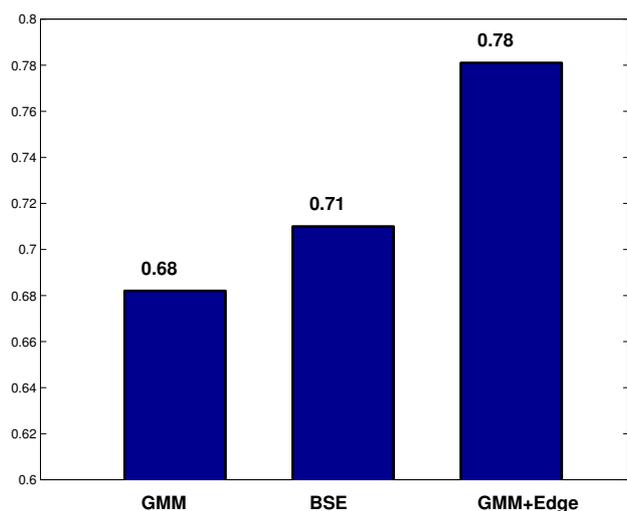
Figure 8. Mean normalized Rand score for three segmentation algorithms on the Berkeley database.

sition features. In the current work, the feature space is augmented to include state-of-the-art edge cues, which are inherently tuned to both color and texture features within the image plane. Thus, color, texture, spatial location and edge information are all accounted for within the developed model. The number of Gaussians that comprise the model is determined automatically and adaptively per image via novel Gaussian allocation and Gaussian-tying schemes. The proposed methodology was shown to outperform earlier GMM versions. Better segmentation results are also achieved as compared to the BSE results, as presented in [5]. A small number of parameters are set once using a small training set, and are shown to work well in a generalized testing scenario. We are currently extending the testing dataset as well as investigating the contribution of the proposed model to specific applications such as medical archives. Additional quantitative measures are investigated for the evaluation.

References

[1] J.F. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. 1, 4

[2] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using Expectation-Maximization and its application to image querying. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002. 1, 2

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis Machine Intell.*, pages 603–619, 2002. 4

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley, New York, 2001. 1

[5] C. Fowlkes, D. Martin, and J. Malik. Learning affinity functions for image segmentation: Combining patch-based and gradient-based approaches. In *Computer Vision and Pattern Recognition (CVPR)*, pages 54–61, 2003. 1, 7, 8

[6] R.C Gonzalez and R.E. Woods. *Digital Picture Processing*. Addison-Wesley Inc., 1993. 1

[7] H. Greenspan, J. Goldberger, and L. Ridel. A continuous probabilistic framework for image matching. *Computer Vision and Image Understanding*, 84:384–406, 2001. 1, 2, 7

[8] H. Greenspan, A. Ruf, and J. Goldberger. Constrained Guassian mixture model framework for automatic segmentation of MR brain images. *IEEE Trans. Med. Imaging*, 25(9):1233–45, 2006. 1

[9] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural boundaries using local brightness, color, and texture cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(5):530–549, 2004. 4

[10] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conf. Computer Vision*, volume 2, pages 416–423, 2001. 4, 6

[11] W. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(366):846–850, 1971. 6

[12] A. Rosenfeld and A. Kak. *Digital Picture Processing*. Academic Press, 2nd edition, New-York, 1982. 1

[13] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 70–77, 2000. 1

[14] E. Sharon, A Brandt, and R. Basri. Segmentation and boundary detection using multiscale intensity measurements. In *Computer Vision and Pattern Recognition (CVPR)*, pages 469–476, 2001. 2

[15] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. 1, 2

[16] R. Unnikrishnan, C. Pantofaro, and M. Hebert. A measure for objective evaluation of image segmentation algorithms. *Computer Vision and Pattern Recognition, Workshop on Empirical Evaluation Methods in Computer Vision*, 2005. 6