SIAM J. IMAGING SCIENCES
Vol. 2, No. 3, pp. 834–857

Expectation-Maximization Algorithm with Local Adaptivity[*]

Shingyu Leung[†], Gang Liang[‡], Knut Solna[§], and Hongkai Zhao[§]

Abstract. We develop an expectation-maximization algorithm with local adaptivity for image segmentation and classification. The key idea of our approach is to combine global statistics extracted from the Gaussian mixture model or other proper statistical models with local statistics and geometrical information, such as local probability distribution, orientation, and anisotropy. The combined information is used to design an adaptive local classification strategy that improves the robustness of the algorithm and also keeps fine features in the image. The proposed methodology is flexible and can be easily generalized to deal with other inferred information/quantities and statistical methods/models.

Key words. expectation-maximization algorithm, Gaussian mixture model, posterior probability, local adaptivity, image segmentation

AMS subject classifications. 62P99, 68U10, 68W01

DOI. 10.1137/080731530

1. Introduction.

Image analysis is of great importance in many application fields, and a large body of methods and algorithms has been proposed for dealing with different tasks from various perspectives. Main image analysis tools include statistical, partial differential equation (PDE), and variational formulation approaches. Each has its own formulation, starting point, advantages, and disadvantages.

The PDE and variational formulation approaches have become important in many applications of image processing and computer vision/graphics. Successful examples include total variation (TV)-based image denoising and deconvolution [29, 37], snakes/active contour models and Mumford–Shah models for image segmentation [10, 27, 11], shape reconstruction from point cloud or range data [25], etc. The main advantage of these methods is their directness and ease in dealing with geometric features, such as edges and boundaries, while their main disadvantage is their underutilization of global statistical information and modeling of randomness/noise.

Another group of image processing methods consists of statistical-based approaches, such as wavelet, Markov random field (MRF), and graphical models, to name a few. Of statistical approaches, the MRF model [3] plays an important role in statistical image processing and has been successfully applied to many image applications, such as image denoising [4], image

[*] Received by the editors July 29, 2008; accepted for publication (in revised form) April 27, 2009; published electronically July 15, 2009.
http://www.siam.org/journals/siims/2-3/73153.html

[†] Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (masyleung@ust.hk).

[‡] Department of Statistics, University of California at Irvine, Irvine, CA 92697 (liang@uci.edu).

[§] Department of Mathematics, University of California at Irvine, Irvine, CA 92697 (ksolna@math.uci.edu, zhao@math.uci.edu).

segmentation [21], and object recognition [40]. The main advantage of statistical modeling is that the randomness information is explicitly taken into consideration, but on the other hand, it is hard to represent geometric information in statistical modeling.

Recently, much attention and research has focused on combining these two different classes of approaches. One common approach is again based on the PDE and/or variational formulation but incorporates statistical information such as prior knowledge for shapes, intensity distributions, and local statistics for textures [24, 35, 14, 16, 19, 17, 12, 33, 2, 18, 20, 30, 38, 8, 36, 39]. A Bayesian inference framework is typically used. Various statistical models, such as parametric models, e.g., Gaussian mixtures, and nonparametric models based on kernel density estimation, are proposed for approximating the probability distribution. One major advantage of the PDE or variational approach is that the regularity (smoothness) of the segmentation boundary can be controlled. In recent work [7, 23, 39], the geodesic active contour model has been turned into a weighted TV formulation which becomes a convex minimization problem for binary images. Hence global minimum can be computed in this setup. For this type of approach a PDE or variational problem has to be solved. For a typical active contour or snake model (see, e.g., [33, 39]), the image is classified into two regions, inside/outside or foreground/background. Extension to vector-valued images is not straightforward.

In this paper, we propose a simple statistical method that can incorporate adaptivity and spatial and geometric information directly. In particular, our method is based on an efficient expectation-maximization (EM) algorithm with local adaptivity that combines the statistical inference and geometric information in image segmentation/classification. The idea of our approach is to couple local statistical and geometric information with global statistics extracted from efficient statistic models, such as the Gaussian mixture model (GMM). This can be done efficiently using an efficient EM algorithm. Using neighborhood and geometric information, such as local orientation and anisotropy, we then determine an optimal adaptive local filtering strategy which can both improve the robustness of the method and also keep fine features in the image. No PDE or variational problem is solved. Furthermore, our method, which is based on a general statistical framework, can be extended to multiple classes, other statistical models, and more general data analysis.

Various methods can be used to improve the GMM and its estimations. Simultaneous feature and model selection [22, 15] is employed to optimize feature saliency, number of components, and parameter estimates of the mixture models. Better estimation of the probability density function for the EM algorithm is proposed in [1, 9]. The main purpose of our work is not to develop new methods for estimating GMMs, but to combine relevant geometrical features with the statistical classification algorithm to generate better classification results. Our method is simple and can be applied on top of other statistical models or methods such as the improved approaches referenced above. With a similar idea [32], region and edge cues (provided by some edge detector) are incorporated into the EM algorithm so that pixels separated by edges do not interfere with each other. However, both the starting point and the approach are different from ours.

Even though we focus our application only on image segmentation in this paper, our approach can be readily generalized to other image or data analysis tasks. This framework can also be easily extended to other inferred information/quantities and statistical methods/models. Moreover, the adaptivity and spatial and geometric information we used in this paper are all locally defined, and hence our algorithm is computationally cheap.

This paper is structured as follows. In section 2, we introduce the GMM and the EM algorithm for extracting global statistics from the whole data set. The posterior probability estimate is computed for each pixel. In section 3, we estimate local statistical and geometrical information such as anisotropy and orientation. A local adaptive neighborhood of each pixel is chosen accordingly, and the classification rule is designed based on the neighborhood information. In section 5, extensive numerical experiments are conducted to demonstrate the performance of our algorithm.

2. Gaussian mixture model. In statistics, the GMM [5, 31, 26] uses a linear mixture of Gaussian distributions to model the phenomenon of interest. Let $\mathbf{y}$ be a measurement vector at a pixel. GMM models its distribution using the form

$$f(\mathbf{y}) = \sum_{k=1}^{K} \alpha_k f_k(\mathbf{y}; \mu_k, \Sigma_k) \quad \text{with} \sum_{k=1}^{K} \alpha_k = 1, \tag{2.1}$$

where K is the total number of mixtures, α_k's are the weights of each Gaussian component, and μ_k, Σ_k are the mean and variance of the kth Gaussian distribution f_k, respectively. The number of components, K, is usually assumed to be known. Otherwise, it is possible to use a model selection procedure to determine a good K along with estimations of other model parameters [31]. This is one of the advantages of the GMM over many PDE-based approaches since the number of classes can be determined from the global data before actual segmentation. Examples in the experiment section will show that our algorithm can still get good results even if K is overestimated.

The GMM is well suited for the task of image segmentation due to its simplicity and power of representation. The Gaussian assumption might seem restrictive at first glance, but it is actually flexible in capturing the image dynamics due to the mixing properties. In practice, Gaussian or near-Gaussian noises are the most prominent noise types. Furthermore, it is possible to use nonlinear transformation to make the transformed signal be more Gaussian mixture–like [6]. In the experiment section, the robustness of our approach is tested on images of different types of noises, such as multiplicative and salt-and-pepper noises.

The EM constitutes an efficient algorithm for obtaining parameter estimates in maximizing the likelihood function. Given the data $\mathbf{y}_1, \ldots, \mathbf{y}_N$, we assume that all data vectors are independent of each other. Suppose the initial guess for the Gaussian mixture model is $\hat{\alpha}_k^{(0)}, \hat{\mu}_k^{(0)}, \hat{\Sigma}_k^{(0)}$; then the EM algorithm iterates between the following two steps until the parameter estimates converge ($t = 1, 2, \ldots$):

- E-step

$$p_k^{(t)}(x) = \frac{\hat{\alpha}_k f_k(\mathbf{y}_x; \hat{\mu}_k^{(t)}, \hat{\Sigma}_k^{(t)})}{\sum_{l=1}^{K} \hat{\alpha}_l f_l(\mathbf{y}_x; \hat{\mu}_l^{(t)}, \hat{\Sigma}_l^{(t)})}, \quad x = 1, \ldots, N,$$

- M-step

$$\hat{\alpha}_k^{(t+1)} = \frac{\sum_{x=1}^{N} p_k^{(t)}(x)}{N}, \quad \hat{\mu}_k^{(t+1)} = \frac{\sum_{x=1}^{N} p_k^{(t)}(x)\mathbf{y}_x}{\sum_{x=1}^{N} p_k^{(t)}(x)},$$
$$\hat{\Sigma}_k^{(t+1)} = \frac{\sum_{x=1}^{N} p_k^{(t)}(x)(\mathbf{y}_x - \hat{\mu}_k^{(t)})'(\mathbf{y}_x - \hat{\mu}_k^{(t)})}{\sum_{x=1}^{N} p_k^{(t)}(x)}.$$

After obtaining estimated parameters given the image data, the usual way of segmenting an image is to classify pixels into the class (or segment) with the largest conditional probability of each pixel:

$$\text{label}(\mathbf{y}_x) = \arg \max_{1\le k\le K} \hat{p}_k(x) = \arg \max_{1\le k\le K} \hat{\alpha}_k f_k(\mathbf{y}_x; \hat{\mu}_k, \hat{\Sigma}_k).$$

The GMM can be viewed as a global method in the sense that all data contribute evenly to the final parameter estimate. The EM algorithm is computationally fast; hence, it is suitable for multiple image or data analysis. However, one shortcoming of this model is that the statistical model is based solely on the distribution of the data, and the classification does not utilize any neighborhood coherence or geometric information, such as orientation or anisotropy. Moreover, this model does not take into account any adaptivity ability due to the local statistical variation. This information might be crucial for image segmentation and other types of data analysis. In the following we will develop an adaptive EM algorithm that can incorporate this information into the model.

3. Local estimation and adaptivity. In practice, the classical EM algorithm usually does not perform well despite its optimality under the GMM assumption. The poor performance is largely due to the lack of geometric information in the GMM. In order to improve the classification performance, we need to utilize some geometric information in the classification phase. Such geometric and local statistical information is vital to the success of the final segmentation.

The idea of our approach is to incorporate information from neighboring pixels. The main underlying assumption is that pixels in an appropriate neighborhood of a given point tend to belong to the same group. Such an assumption holds well for most natural and synthetic images. Nevertheless, because the spatial distribution of image pixels is irregular, the key is to choose an appropriate homogeneous neighborhood to borrow information from. In this paper, we propose using local geometric features such as anisotropy and orientation to determine such a neighborhood. To further improve the method, an adaptive *optimal* classification strategy is designed based on locally estimating both statistics and geometric information. The three main steps of our local adaptation method are as follows:

1. Estimate the local anisotropy and orientation (section 3.1).
2. Determine a local neighborhood according to those estimates in the first step (section 3.2).
3. Compute local statistics in the neighborhood and classify the pixel by minimizing the misclassification rate adapted to these local statistics (section 3.3).

In short, our strategy is to use local information to determine which neighboring pixels' information is used. Ideally, a good neighborhood for classification is a region with only pixels of the same group. For simplicity, we use only elliptical neighborhoods with a certain aspect ratio (related to the strength of the anisotropy) and orientation (related to the orientation of the anisotropy) in our paper.

The above procedure combines global statistics with local information, thus providing adaptivity for improving the robustness and accuracy of the classification. Meanwhile, fine features are well kept, as our experiments show in section 5.

3.1. Local anisotropy and orientation estimation. Local anisotropy and its orientation are important information for determining the appropriate neighborhood of a given pixel. We follow the approach in [28] to estimate the local anisotropy and its orientation at a pixel. At the pixel centered at $\mathbf{x}$, we collect a small neighborhood within a ball of radius r and denote it by $\mathcal{B}(\mathbf{x}, r)$. In general, the local neighborhood for each classification group might be different. But for simplicity, in this paper we will use the same ball for all classes.

The first moment vector (mass center) for each class label k, $k = 1, \dots, K$, at $\mathbf{x}$ is defined by

$$\mathcal{M}_k^0(r,\mathbf{x}) = \frac{\int_{\mathcal{B}(\mathbf{x},r)} \mathcal{W}(r,\mathbf{x},\hat{\mathbf{x}},\mathbf{v},1) p_k(\hat{\mathbf{x}})\hat{\mathbf{x}}\,\mathrm{d}\hat{\mathbf{x}}}{\int_{\mathcal{B}(\mathbf{x},r)} \mathcal{W}(r,\mathbf{x},\hat{\mathbf{x}},\mathbf{v},1) p_k(\hat{\mathbf{x}})\,\mathrm{d}\hat{\mathbf{x}}}, \tag{3.1}$$

where p_k is the posterior of the k-class and $\mathcal{W}$ is a weight function with exponential decay in terms of the distance from the center $\mathbf{x}$ given by

$$\mathcal{W}(r,\mathbf{x},\hat{\mathbf{x}},\mathbf{v},1) = \exp\left(-\frac{|\mathbf{x}-\hat{\mathbf{x}}|^2}{2(r/2)^2}\right). \tag{3.2}$$

Similarly, the second moment at $\mathbf{x}$ is a matrix (2×2 for two-dimensional (2D) images) for each class k defined by

$$\mathcal{M}_k^1(r,\mathbf{x}) = \frac{\int_{\mathcal{B}(\mathbf{x},r)} \mathcal{W}(r,\mathbf{x},\hat{\mathbf{x}},\mathbf{v},1) p_k(\hat{\mathbf{x}})[\hat{\mathbf{x}} - \mathcal{M}_k^0(r,\mathbf{x})] \otimes [\hat{\mathbf{x}} - \mathcal{M}_k^0(r,\mathbf{x})]\,\mathrm{d}\hat{\mathbf{x}}}{\int_{\mathcal{B}(\mathbf{x},r)} \mathcal{W}(r,\mathbf{x},\hat{\mathbf{x}},\mathbf{v},1) p_k(\hat{\mathbf{x}})\,\mathrm{d}\hat{\mathbf{x}}}, \tag{3.3}$$

where $\mathbf{u} \otimes \mathbf{v}$ is the tensor product of the vectors $\mathbf{u}$ and $\mathbf{v}$.

The eigenvalues of each first moment matrix are denoted by $\lambda_k^{(1)}(r,\mathbf{x})$ and $\lambda_k^{(2)}(r,\mathbf{x})$: These eigenvalues along with their corresponding eigenvectors, $\mathbf{v}_k^{(1)}(r,\mathbf{x})$ and $\mathbf{v}_k^{(2)}(r,\mathbf{x})$, essentially measure the anisotropy and its direction of each classification group. To quantify the anisotropy, we define

$$\tau_k(r,\mathbf{x}) = 1 - \frac{\max[\lambda_k^{(1)}(r,\mathbf{x}), \lambda_k^{(2)}(r,\mathbf{x})]}{\lambda_k^{(1)}(r,\mathbf{x}) + \lambda_k^{(2)}(r,\mathbf{x})}. \tag{3.4}$$

If the probability distribution $p_k(\hat{\mathbf{x}})$ for class i is isotropic in the neighborhood centered at $\mathbf{x}$, the two eigenvalues $\lambda_k^{(1)}(r,\mathbf{x})$ and $\lambda_k^{(2)}(r,\mathbf{x})$ will be roughly the same. This implies that the strength of the anisotropy τ_k with respect to the group k will be close to, but less than, 1/2.

On the other hand, if the distribution is very anisotropic in the neighborhood centered at $\mathbf{x}$, there will be a dominant eigenvalue and therefore the value of τ_k will be close to, but greater than, 0.

Since the above moment matrix can be computed for all classes, the anisotropy at a particular pixel is not uniquely defined. In our algorithm, we use the most anisotropic class, its eigenvalues and eigenvectors, to define the local neighborhood region for later adaptive classification (see details in the following sections). This choice is based on the following observations: A pixel that is in the interior of a homogeneous region or an isolated point (which belongs to a different class from its neighbors) does not generate anisotropy. A pixel that is near a smooth boundary between two classes does not generate large anisotropy either. Large anisotropy indicates existence of some spatial structure, such as a long and thin (elongated) object/region. As a consequence our choice can maintain this spatial structure well. At the same time pixels in the interior or near a smooth boundary are hardly affected by the choice of the neighborhood. In particular, isolated pixels are still removed no matter what neighborhood information we use. These are two major desirable properties of our algorithm which the classical EM algorithm does not have.

3.2. Local neighborhood adaptation. Now we incorporate the above anisotropy and orientation estimates into our local adaptive algorithm. We introduce an affine transformation that consists of both rotation and rescaling according to the local anisotropy and orientation. For simplicity, we restrict our discussion to 2D images. Let $v_1^{(k)}, v_2^{(k)}$ be the components of the unit eigenvector $\mathbf{v}^{(k)}(r, \mathbf{x})$ corresponding to the two eigenvalues of the first moment matrix for class k that has the strongest anisotropy. First we translate and rotate the original coordinate system to align with the direction of anisotropy:

$$\begin{aligned} \tilde{x}_1 &= v_1^{(k)}(\hat{x}_1 - x_1) + v_2^{(k)}(\hat{x}_2 - x_2), \\ \tilde{x}_2 &= v_1^{(k)}(\hat{x}_2 - x_2) - v_2^{(k)}(\hat{x}_1 - x_1). \end{aligned} \tag{3.5}$$

Next we scale each direction according to the anisotropy. We define the aspect ratio of the scaling β in terms of the strength of the anisotropy,

$$\beta = (1 - 2\tau)\beta_{\max} + 2\tau, \tag{3.6}$$

with $\beta_{\max} = r$. Denote the scaling in the $\tilde{x}_1$- and the $\tilde{x}_2$-direction by a and b, respectively. An affine distance D is defined as

$$D = \frac{\tilde{x}_1^2}{a^2} + \frac{\tilde{x}_2^2}{b^2}, \tag{3.7}$$

where

$$a^2 = \frac{A\beta}{\pi}, \quad b^2 = \frac{A}{\pi\beta}. \tag{3.8}$$

3.3. Classification by minimizing the misclassification rate. The classical EM algorithm classifies a pixel into one of the k-groups using the posterior at that particular pixel; i.e., one classifies pixel $\mathbf{x}$ into group $\hat{k}$ such that

$$\hat{k} = \arg \max_{1 \le k \le K} \hat{p}_k(\mathbf{x}) \tag{3.9}$$

with the posterior $\hat{p}_k$ computed from the whole image. To take into account the neighborhood information, we define a likelihood function and minimize the misclassification rate in a local patch centered at the pixel $\mathbf{x}$. The local patch is an elliptic neighborhood $\mathcal{D}(\mathbf{x})$ of $\mathbf{x}$ determined by local anisotropy and its direction:

$$\left\{ \frac{\tilde{x}_1^2}{a^2} + \frac{\tilde{x}_2^2}{b^2} - 1 < 0 \right\}. \tag{3.10}$$

Hence local information on anisotropy and orientation is now embedded into such local patches.

Let $(\hat{\mu}_k, \hat{\Sigma}_k)$ be the mean and the variance for class k computed from the EM algorithm in section 2 from the whole image. The likelihood function for the local patch for class k is defined as

$$L(\hat{\mu}_k, \hat{\Sigma}_k; \mathcal{D}(\mathbf{x})) = \prod_{\mathbf{z} \in \mathcal{D}(\mathbf{x})} \frac{1}{\sqrt{2\pi}} |\hat{\Sigma}_k|^{-1/2} \exp\left(-\frac{1}{2} (\mathbf{z} - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (\mathbf{z} - \hat{\mu}_k) \right).$$

Theorem 3.1. *The following rule classifies each pixel into one of k classes by minimizing the misclassification rate based on the local likelihood:*

$$\arg \max_{1 \le k \le K} \hat{\alpha}_k L(\hat{\mu}_k, \hat{\Sigma}_k; \mathcal{D}(\mathbf{x})), \tag{3.11}$$

where $\hat{\alpha}_k$ is the weight for each class in the Gaussian mixture.

A simple proof of the optimality of this decision rule is provided in the appendix. Another interpretation of the above classification rule is by maximum likelihood: We can view $\hat{\alpha}_k$ as the prior (estimated from the whole data set), and $\mathcal{L}(\hat{\mu}_k, \hat{\Sigma}_k; \mathcal{D}(\mathbf{x}))$ is the probability (or likelihood) of the local patch around $\mathbf{x}$ belonging to the group k.

In the classical EM method, only the global statistics of the pixel values are utilized. The classification rules of different pixels are spatially independent of each other; i.e., if the pixels are shuffled around, the classification of a particular measurement will be unchanged. In contrast, our classification rule of a pixel has taken into account statistical, spatial, and geometric information in an appropriately chosen neighborhood. As a consequence, our classification/segmentation can keep fine features, while the number of isolated misclassified points is greatly reduced compared to the classical EM method. We will see these improvements for our method in various numerical examples in section 5.

3.4. Further adaptation.

A further adaptation of the algorithm helps to improve the estimation of anisotropy and hence the choice of local neighborhood. For example, we can incorporate an extra step by modifying the weight function in the moment formulas (3.1) and (3.3) using the affine distance D in (3.7); i.e.,

$$\mathcal{W}(r, \mathbf{x}, \hat{\mathbf{x}}, \mathbf{v}, \tau_k) = \exp\left(-\frac{D}{2} \right), \tag{3.12}$$

where class k has the strongest anisotropy. Once we get the new estimates of local anisotropy and its orientation, we follow the same procedure as described above to pick up the local neighborhood and minimize the misclassification rate. As we will see in section 5 this extra step can improve the segmentation.

4. Algorithm.

To help readers follow the procedures, we now summarize our algorithm.

```
Algorithm:
    Input: An image with $N$ pixels with measurements $\mathbf{y}_1, \dots, \mathbf{y}_N$; the number of groups $k$.
    Output: The label for each pixel label($\mathbf{y}_x$).
    Initialization:
     (a) Mixture parameters $p_k(x)$, $\hat{\alpha}_k(x)$, $\hat{\mu}_k(x)$, and $\hat{\Sigma}_k(x)$ with the GMM using the EM
         algorithm.
     (b) Local neighborhood $\mathcal{D}(\mathbf{x}) = \mathcal{B}(\mathbf{x}, r)$ and a weight function $\mathcal{W}(r, \mathbf{x}, \hat{\mathbf{x}}, \mathbf{v}, 1)$ according
         to (3.2).
    Iteration:
        Do $t = 1, 2, \dots$
           i. Estimate the local anisotropy $\tau_k(r, \mathbf{x})$ using (3.3) and (3.4).
          ii. Determine a local neighborhood for each pixel $\mathcal{D}(\mathbf{x})$ according to (3.10).
         iii. Classify the pixel according to the classification rule (3.11).
          iv. Modify the weight function $\mathcal{W}(r, \mathbf{x}, \hat{\mathbf{x}}, \mathbf{v}, \tau_k)$.
        Enddo
```

5. Experiments.

In this section, we test our algorithm on various intensity scalar images and vector images and also compare these results with the standard EM algorithm and some PDE-based approaches. We show that by incorporating the local statistical, spatial, and geometric information, our method can keep sharp or fine features while maintaining robustness with respect to noise of various types. Based on the efficient EM algorithm, our local adaptive algorithm can also be efficiently implemented.

Unless otherwise specified, we define an intensity image as a function $u : \Omega \rightarrow [0, 255] \subset \mathbb{I}$. Except in the example we have used in Figure 3, where $\Omega = [1, 256]^2$, we are using an image with $\Omega = [1, 128]^2$. The original clean image is binary with the piecewise constant intensity levels of 85 and 170, while the noisy image is obtained by adding a Gaussian noise of standard deviation $\sigma = 50$. To visualize the segmentation results, we plot all pixels within the same group in the same color, either black or white. For instance, to generate the input image in Figure 1(a), we first construct a piecewise constant image with the above prescribed intensity. Then we add to the image a Gaussian noise with known variance. Once we have obtained the segmented results of this noisy input, i.e., pixels classified into groups, we show the corresponding classes with the color black/white in images (b)–(d).

For vector images $u : \Omega \rightarrow \mathbb{R}^d$, we have chosen both colored images in the RGB format ($d = 3$) and images with higher dimensions ($d > 3$). In these cases we segment the images into more than two groups, we color all pixels classified into the same group in black, and we plot each of these groups separately.

We start with some standard model comparisons in the sense that images are indeed satisfying the assumption of Gaussian mixture distribution. Then we will vary some of the assumptions of the GMM to check the robustness of our proposed method. We will then compare some of our solutions with a modified version of the Chan–Vese (CV) model [34].

5.1. Preservation of fine structures.

Preservation of fine features is a difficult task for image segmentation. Figure 1 shows a synthetic example to assess the ability of our algorithm

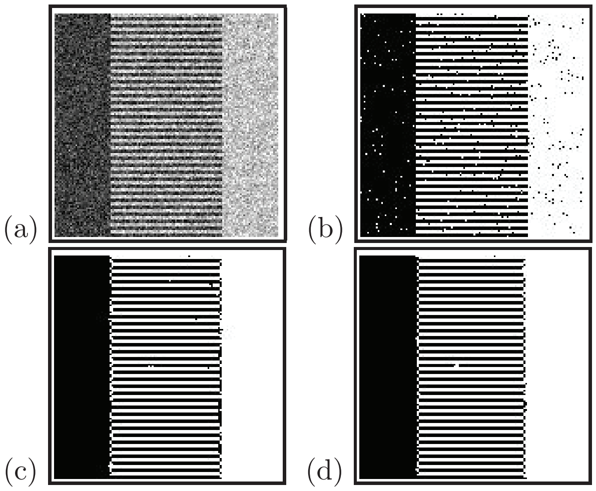

Figure 1. (a) *The original noisy image,* (b) *the classification by EM,* (c) *our classification, and* (d) *with further adaptation.*

to preserve these small structures. The original clean image Figure 1 is binary (dark and light) with three distinct regions. The left and right regions are homogeneous with different intensity values (dark and light, respectively). The middle region consists of horizontal stripes only two pixels wide. Our goal is to segment the image into two groups according to their intensities. The main challenge is the segmentation of the middle stripe region. Anisotropy is therefore an important quantity in incorporating appropriate neighborhood information to improve robustness while preserving fine structures.

Figure 1(a) shows the input image with significant noise added (standard deviation σ of 30). The segment solution from our method performs well, as shown in Figure 1(c). Although the stripes are each only two pixels wide, the initial circular neighborhood, which is the size of a few pixels, can correctly detect the anisotropy and the orientation using the moment estimation described in section 3.1. We also apply in Figure 1(d) the further adaptation step described in section 3.4. The classification result is further improved. Compared to the result from the standard EM algorithm, Figure 1(b), in which pixels are spatially independent of each other, our algorithm has significantly improved the segmentation result in that we have far fewer isolated misclassified pixels.

The next example, Figure 2, shows that our algorithm can keep an extremely elongated fine feature in the image while the noises are nicely removed. This example is similar to the previous example except that there is only one stripe of one pixel width in the middle region. This example shows that our method effectively estimates both the anisotropy and its direction. With this neighborhood information appropriately used, we are able to produce remarkable results.

We further test our algorithm on the example in Figure 3, which consists of stripes of different orientations and a classical square picture. Our method significantly improves the result of the standard EM algorithm. With another extra adaptation, some of the isolated dots are further removed from the final segmentation results as shown in Figure 3(d). Indeed, the boundary of this segmented object (and also results from our other examples) is not as

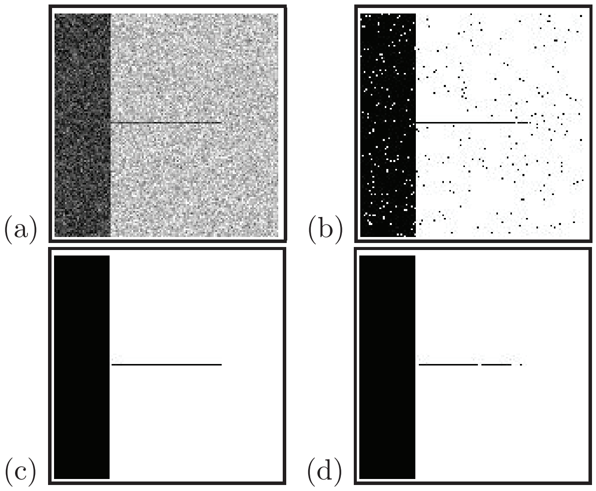

Figure 2. (a) *The original noisy image,* (b) *the classification by EM,* (c) *our classification feedback, and* (d) *with further adaptation.*

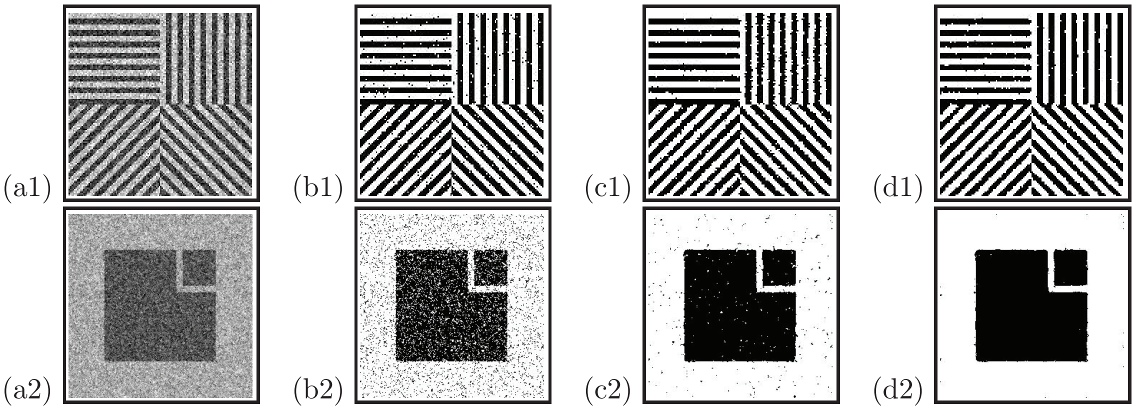

Figure 3. (a) *The original noisy image,* (b) *the classification by EM,* (c) *our classification, and* (d) *with further adaptation.*

smooth as those obtained by typical variational methods where the regularity of the boundary is imposed explicitly in the energy/functional. In the current algorithm, we do not directly penalize on the regularity of the boundary but by adaptively modifying the local neighborhood for computing the local statistics. This type of anisotropic filtering usually gives a less regular boundary than those PDE methods.

5.2. Robustness.

5.2.1. Various noise types. Although we have assumed the GMM in our approach, such an assumption might be violated in practice. Below, we use a few examples of various noise types to test the robustness and the applicability of our algorithm when the Gaussian assumption does not hold.

In this example, we randomly select 5% (the first row in Figure 4) or 10% (the second row in Figure 4) of the pixels of the original clean binary image and switch their intensities from

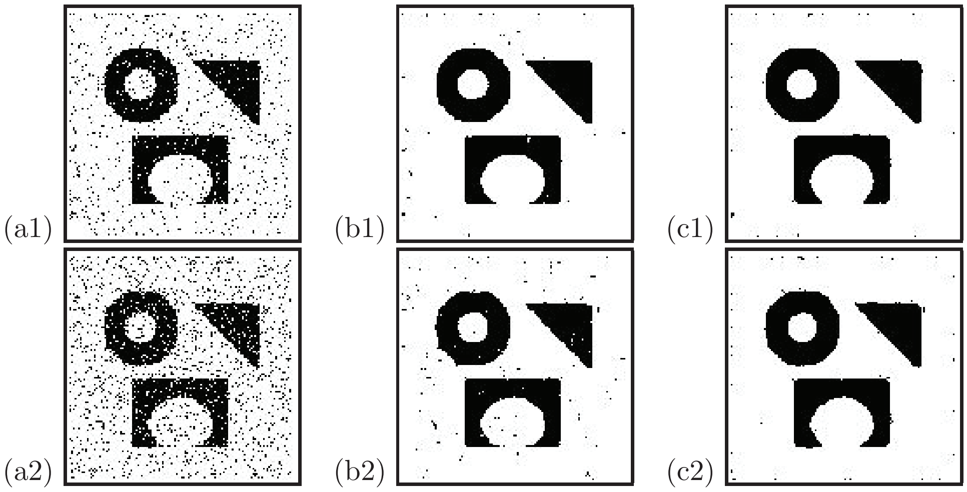

Figure 4. (a) *The original noisy images (the same as their classification results by EM),* (b) *our classification, and* (c) *with further adaptation. First row:* 5% *of pixels are flipped. Second row:* 10% *of pixels are flipped.*

one color to another. Since the EM algorithm does not take into account any local information, the classification by the classical EM is the same as this noisy version. Figure 4(c) shows the classification with further adaptation. Most noise is successfully removed.

Figure 5 shows segmentation results of an image polluted by different types of noise. In the first row, we consider the additive Gaussian noise. The resulting noisy image does satisfy our GMM. In the second row, we study the multiplicative noise. In this case, we multiply the original clean image by a white noise with standard deviation 0.2. The third row in Figure 5 shows segmentation results of an image with 10% salt-and-pepper noise. To better improve the classification results with respect to different types of noises, one might incorporate a more appropriate model to capture the noise dynamics. For instance, a Bernoulli model can be used for the salt-and-pepper noise. Nevertheless, the above examples indicate that the GMM is robust. Even though the Gaussian assumption is violated, our algorithm still nicely classifies all images.

5.2.2. Misspecified number of mixtures.

Although there are various statistical methods for estimating the number of mixtures from the raw data, we would like to demonstrate that our adaptive EM is quite robust with respect to the misspecification of the number of mixtures. We first add to the original binary image (with only the black class and the white class) a Gaussian noise of standard deviation 50. Then we apply our algorithm to classify the image into three, but not two, classes. In the first row in Figure 6, we show the results from EM. Black regions from each subplot correspond to classifications from each class. The percentages of pixels associated to different classes are 23.97%, 68.59%, and 7.44%, respectively. Without any further adaptation, our algorithm already gives a much better classification result, as shown in the second column. One class is almost completely empty. The corresponding percentages of pixels are now 20.54%, 79.46%, and 0.00%, respectively. Comparing these numerical results to the exact ratio (20%, 80%, 0%), our algorithm can automatically remove any unnecessary class in the classification.

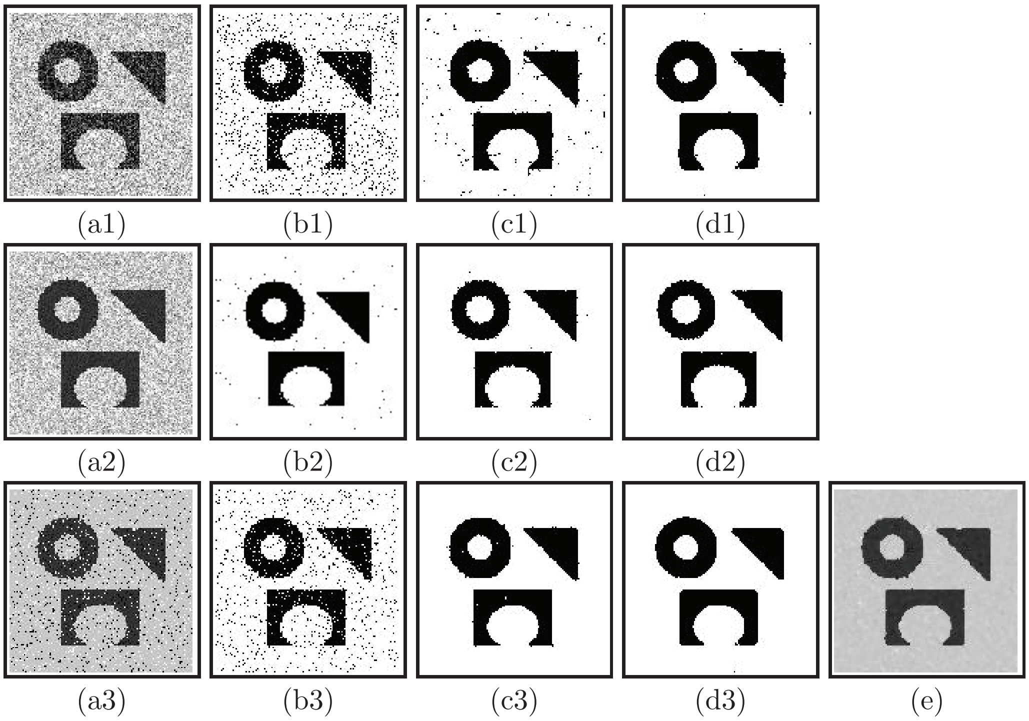

Figure 5. *Application of our algorithm for three different types of noises: additive Gaussian noise (first row), multiplicative noise (second row), and* 10% *salt-and-pepper noise (third row).* (a) *The original noisy image,* (b) *the classification by EM,* (c) *our classification,* (d) *with further adaptation, and* (e) *denoising the salt-and-pepper noise using a median filter.*

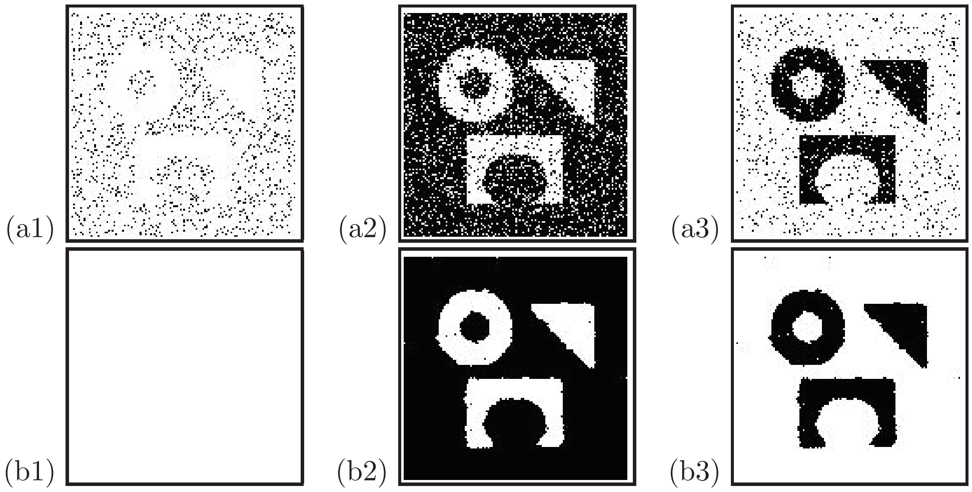

Figure 6. *Classification by* (a) *EM and* (b) *our algorithm.*

5.3. Comparisons with variational methods. The CV model [11] is a popular PDE-based method for segmentation. The method aims to segment images without looking into the gradient of the intensity function. Instead the segmentation from the original version is

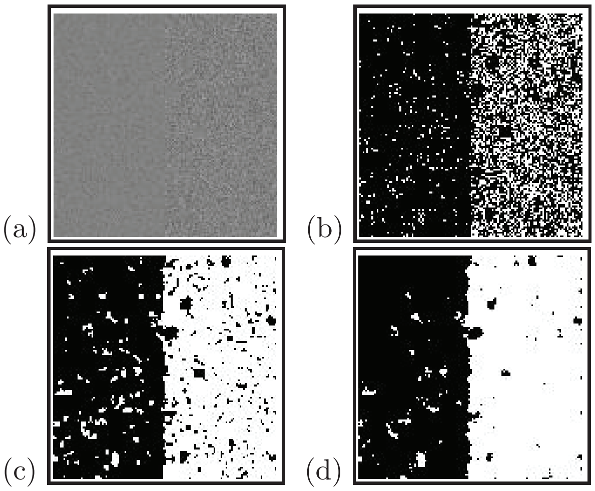

Figure 7. *(Variances* 25 *and* 100.*)* (a) *The original noisy image,* (b) *the classification by EM,* (c) *our classification, and* (d) *with further adaptation.*

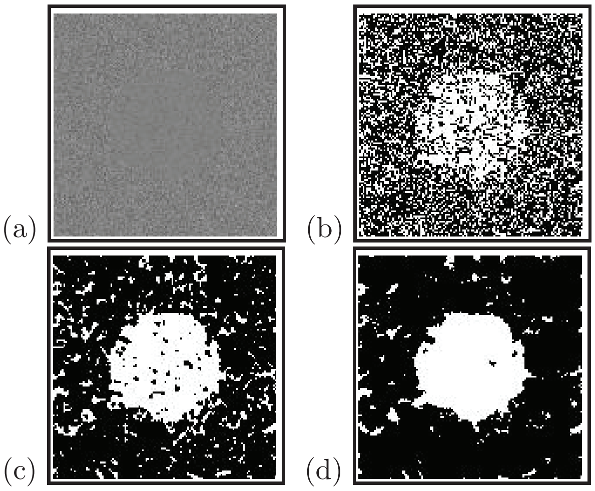

Figure 8. *(Variances* 25 *and* 100.*)* (a) *The original noisy image,* (b) *the classification by EM,* (c) *our classification, and* (d) *with further adaptation.*

based on average (mean) intensity in different regions. However, not only are small features difficult to capture by looking only at the first moment, but higher moments such as the variance are not included. For instance, the original CV model will not work for images consisting of regions of the *same mean* but *different variances.* To take into consideration higher moments, [34] recently generalized the CV energy by using the average of the probability density function.

Figure 7 shows the segmentation results of an image with the same mean (intensity equals 128) in both the left and the right regions but with different variances. The variances are 25 and 100 for the left and the right regions, respectively. Figure 8 is similar to Figure 7, but now the region with lower variance is a disc in the middle of the image. This

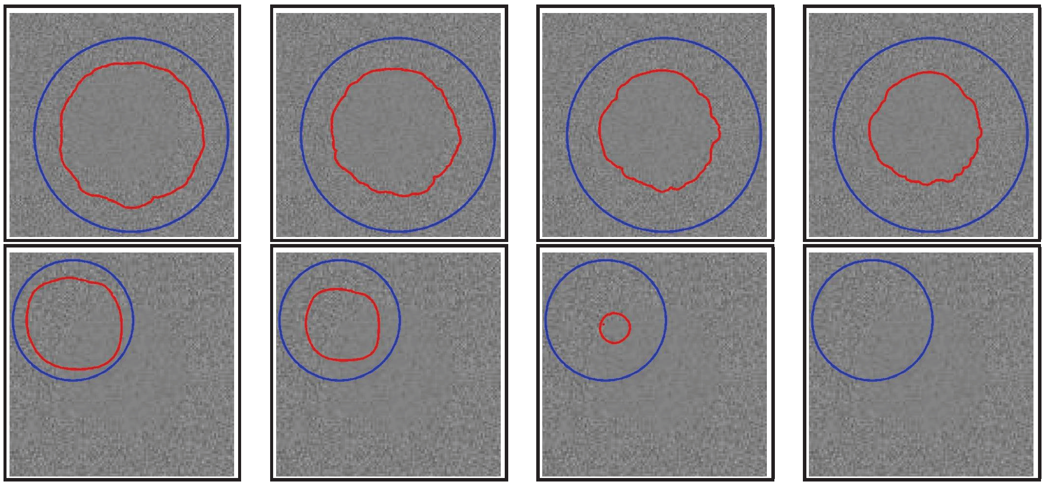

Figure 9. *(Variances* 25 *and* 100*.) Each row shows the evolution of the zero level set in the modified CV energy in* [34] *with a different initial guess. The outer circle is the initial guess, while the inside curve shows the evolutions of the boundary. The model can detect the boundary of the object, but, unfortunately, the result still depends on the choice of the initial guess.*

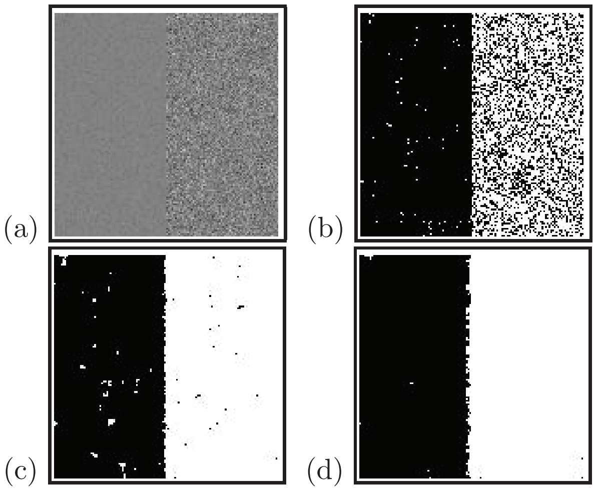

Figure 10. *(Variances* 25 *and* 400*.)* (a) *The original noisy image,* (b) *the classification by EM,* (c) *our classification, and* (d) *with further adaptation.*

task is quite hard for human eyes. But our approach can reasonably segment both images although a little less accurately near the boundary. For this small variance difference, it is hard to distinguish these two classes in a small neighborhood for pixels near the boundary since we have very little consistent information from almost equal mixtures of pixels from these two close classes. When the variances of different regions are further separated, as in Figures 10 and 11, our segmentation results are much better. To compare them with the results from a popular model, we plot in Figures 9 and 12 the results from the modified CV model developed in [34]. The modified CV energy gives reasonable segmentation results. However, such results would still depend on the choice of the initial guess since the energy is in fact nonconvex.

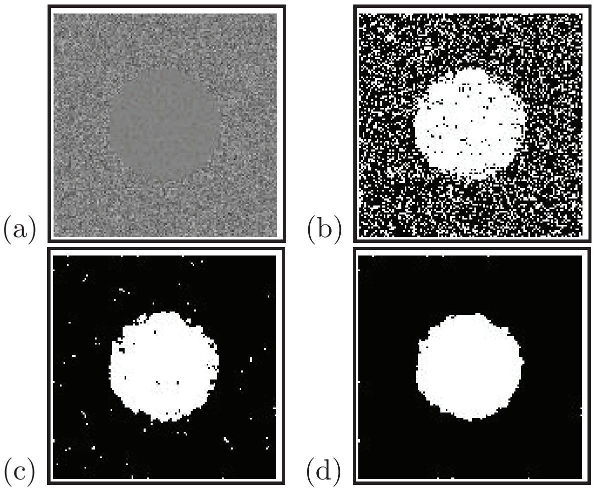

Figure 11. *(Variances* 25 *and* 400.*)* (a) *The original noisy image,* (b) *the classification by EM,* (c) *our classification, and* (d) *with further adaptation.*

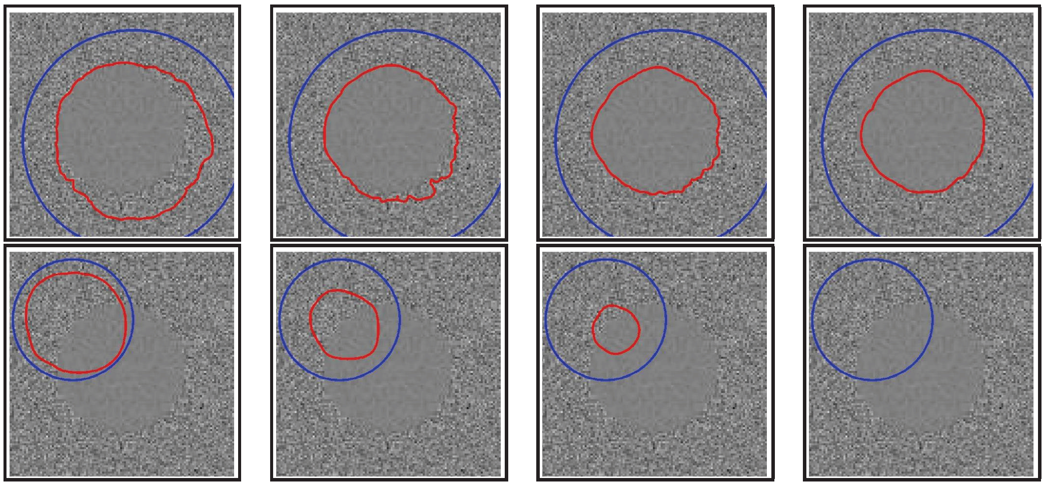

Figure 12. *(Variances* 25 *and* 400.*) Each row shows the evolution of the zero level set in the modified CV energy in* [34] *with a different initial guess. The outer circle is the initial guess, while the inside curve shows the evolutions of the boundary. The model can detect the boundary of the object, but, unfortunately, the result still depends on the initial guess.*

5.4. Vector images.

Our method applies naturally to data of any dimension. In this subsection, we apply our method on vectorial images/data $u : \mathbb{R}^2 \to \mathbb{R}^d$ with $d > 1$. The first example is a color three-channel image; i.e., $d = 3$. Our algorithm can be easily applied to this example, and our solutions are shown in Figure 13. Figure 13(0) shows the noisy image with Gaussian noise added to each channel independently. The segmented solutions by the EM algorithm using three Gaussian classes ($k = 3$) are given in the rest of the first row. We plot all pixels classified into the same group in black. Similarly to the previous examples, we find many isolated dots or misclassified pixels in the segmentation from the EM algorithm. Figure 13(b) shows results from our method without any feedback. The misclassifications

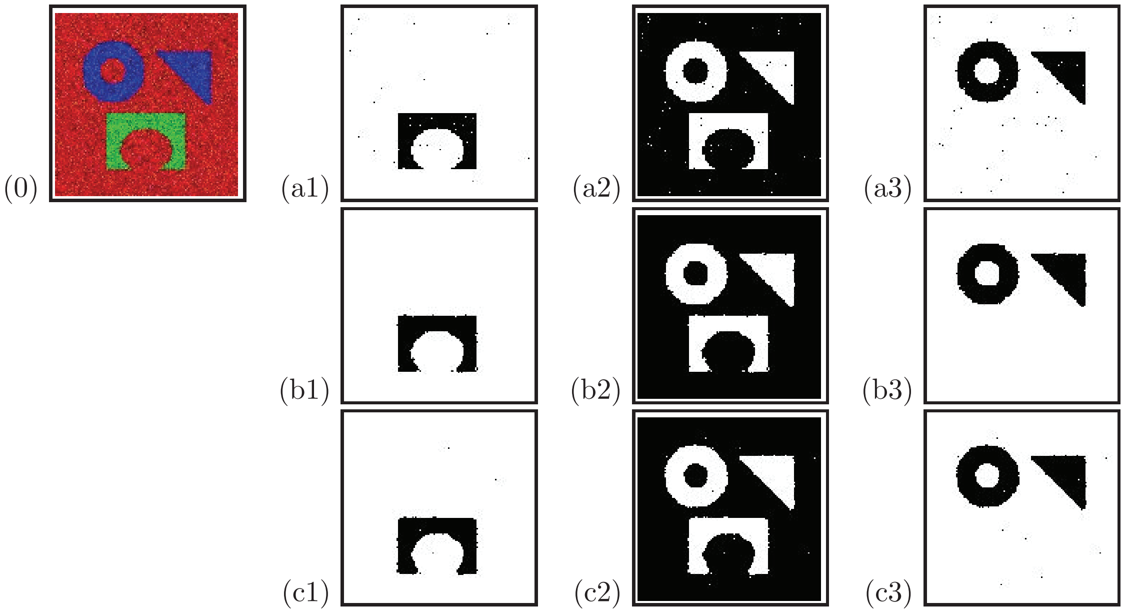

Figure 13. (0) *The original noisy image,* (a) *the classification by EM,* (b) *our classification, and* (c) *with further adaptation. Pixels classified into the same group are plotted in black.*

are nicely removed. We show in Figure 13(c) results with further adaptation described in section 3.4. The hypotenuse of the triangle is captured better. The elliptical part inside the rectangle is also segmented more smoothly.

The following example shows image segmentation based on quantities/information (deterministic or statistical) derived from the original images, such as local homogeneity, texture, and others. Figures 14(a) and (b) show both the clean image and its noisy version with additive Gaussian noise. A similar image has been segmented in a previous section based on the intensity value. However, this example is even more extreme in the sense that each white and black alternating stripe is only one pixel wide now. The texture is extremely fine and homogeneous. Here we segment the image based on pattern information. In other words, we separate the image into regions with different textures. In the first test, we segment the image according to the contrast of a pixel to its neighbors. First we derive a vectorial function in $\mathbb{R}^4$ by transforming the intensity $u_{i,j}$ at each pixel (i, j) into

$$u_{i,j} \to \mathbf{y}_{i,j} = \begin{pmatrix} \Delta_x^+ u_{i,j} \\ \Delta_x^- u_{i,j} \\ \Delta_y^+ u_{i,j} \\ \Delta_y^- u_{i,j} \end{pmatrix} \in \mathbb{R}^4, \tag{5.1}$$

where $\Delta_x^\pm$ and $\Delta_y^\pm$ are the forward and the backward differences in the x- and the y-direction, respectively. For instance,

$$\begin{aligned} \Delta_x^+ u_{i,j} &= u_{i+1,j} - u_{i,j}, \\ \Delta_x^- u_{i,j} &= u_{i,j} - u_{i-1,j}. \end{aligned} \tag{5.2}$$

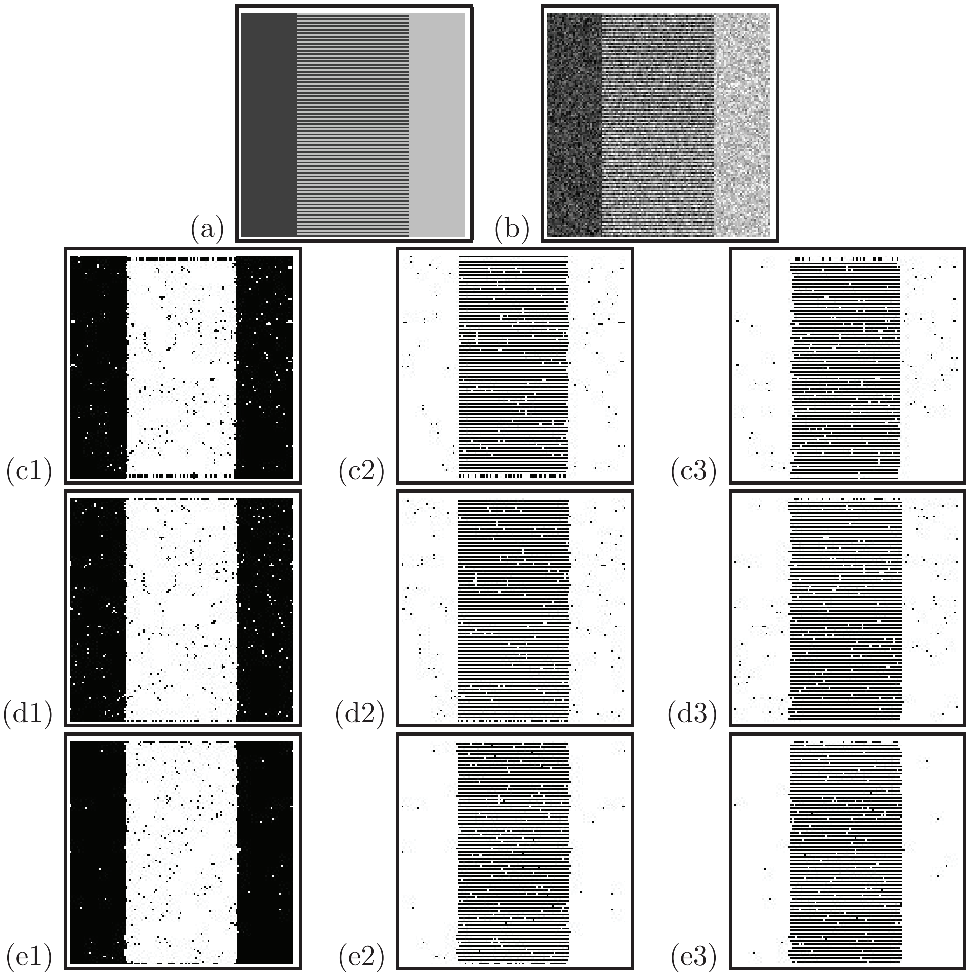

Figure 14. (a) *The original clean image,* (b) *the noisy image,* (c) *the classification by EM,* (d) *our classification, and* (e) *with further adaptation. Pixels classified in the same group are plotted in black.*

The vector function $\mathbf{y}_{i,j}$ gives the contrast of pixel $\mathbf{x}_{i,j}$ to its four neighbors.

As an example, consider the clean image Figure 14(a) and denote the intensity of black by 0 and that of white by 1. The intensities in the left and right regions are homogeneous, which gives $\mathbf{y} = (0, 0, 0, 0)^T$. For the middle region, we have $\mathbf{y} = (0, 0, 1, -1)^T$ and $\mathbf{y} = (0, 0, -1, 1)^T$ for the pixels at the dark and light stripes, respectively.

We test our method on the noisy image with Gaussian noise of standard deviation 30 as shown in Figure 14(b). The second row in Figure 14(c) represents the results from the standard EM algorithm. Our results are shown in Figure 14(d). Segmentation of the stripe region is a very challenging task because the texture is extremely fine and homogeneous. One difficulty is that almost no anisotropy can be detected from a ball of any size around any pixel in the stripe region using the method described in section 3.1. Second, unless we are using an extremely anisotropic neighborhood like a line segment only one pixel wide, neighborhood information might not help much in classifying those pixels in the stripe region. This explains

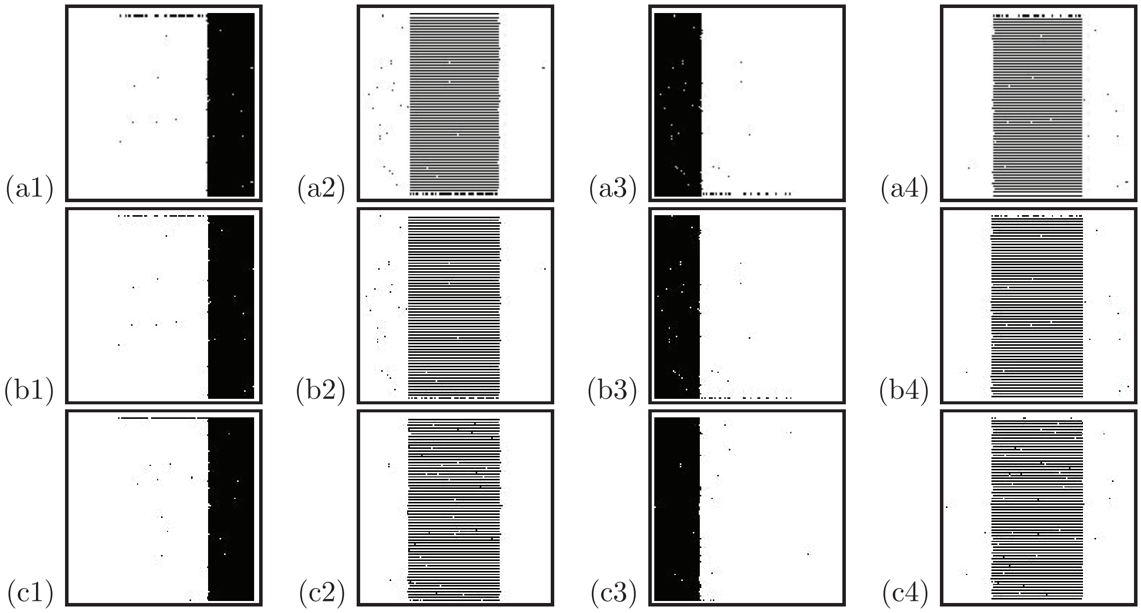

Figure 15. (a) *The classification by EM,* (b) *our classification, and* (c) *with further adaptation. Pixels classified into the same group are plotted in black.*

why our segmentation result in the homogeneous region is better than for the classical EM as in previous examples, while the segmentation of the stripes is comparable.

To further separate the dark and the light homogeneous regions, we embed the image u in $\mathbb{R}^5$ by including the intensity value

$$u_{i,j} \to \mathbf{y}_{i,j} = \begin{pmatrix} \Delta_x^+ u_{i,j} \\ \Delta_x^- u_{i,j} \\ \Delta_y^+ u_{i,j} \\ \Delta_y^- u_{i,j} \\ u_{i,j} \end{pmatrix} \in \mathbb{R}^5. \tag{5.3}$$

Figures 15(b) and (d) show the results from our algorithm.

5.5. Misclassification rate. In this subsection, we study the accuracy of our methods by examining the misclassification rate of our experiment results. Since all examples shown in the experiment section are synthetic, we can always determine the *exact* segmentation results. The misclassification rate is computed by dividing the total number of mismatched pixels by the total number of pixels. In other words, we look at the percentage of mismatch in each classification result.

Table 1 summarizes all the misclassification rate results. It indicates that our segmentation algorithm outperforms the original EM algorithm. The improvement is dramatic, especially for hard segmentation examples. For instance, in Figure 7, the original EM gives more than 30% error in the classification, while our proposed method has a misclassification rate less than 10%. With one more adaptation, the misclassification rate is further driven down to less than 4%.

Table 1

Percentage of misclassified pixels in each example.

Figure	EM	Ours without feedback	Ours with one feedback
1	2.0935	1.3550	2.1606
2	2.6855	0.7874	0.8545
3 (row 1)	2.1545	3.3142	2.2400
3 (row 2)	10.9619	1.2604	0.3876
4 (row 1)	4.8523	0.7751	1.0681
4 (row 2)	9.0820	1.4771	1.3916
5 (row 1)	7.3120	2.1606	1.3306
5 (row 2)	0.3845	2.1606	2.7771
5 (row 3)	4.9622	0.7446	0.8850
6 (redundant)	7.3059	0.0000	0.0000
6 (background)	13.7390	1.4954	1.4587
6 (objects)	6.4331	1.4954	1.4587
7	31.2927	9.1309	3.7842
8	32.5684	11.7676	6.1401
10	16.9373	1.1963	1.3916
11	23.3032	2.1790	2.2949
13 (one object)	0.1953	0.2441	0.4517
13 (two objects)	0.2441	0.3601	0.7141
13 (background)	0.3296	0.6042	1.1414
14 (background)	2.6611	2.6794	1.9775
14 (stripes 1)	1.2268	1.2695	1.2207
14 (stripes 2)	1.4343	1.4099	1.0620
15 (left background)	0.3906	0.3906	0.5493
15 (right background)	0.3723	0.3723	0.2625
15 (stripes 1)	0.3540	0.3540	0.4028
15 (stripes 2)	0.4089	0.4089	0.7629

For some examples, the misclassification rates of our algorithm are comparable to (or slightly worse than) those of the original EM. We analyze them case by case. In general, our method does not show much improvement over the original EM for the vector image examples shown in Figures 13–15. Our analysis suggests that these examples are relatively simple segmentation problems: Vector images are generally relatively easy to segment since pixels in images of higher dimensions are inherently more separated. Our algorithm performs much better in hard segmentation problems. When the original EM already performs well, there lacks room for further improvement. Another observation worth mentioning is that our algorithm effectively removes many isolated misclassified pixels. Even though the improvement of misclassification rates might not be significant for some cases, the final segmentation results are much more appealing to human eyes.

5.6. Real images. In previous sections, we have studied the behavior of our proposed algorithm by applying it to various synthetic images. In this section, we further examine its performance by applying it to several images in the real world. Unlike previous examples, there is no rigorous definition of *segmentation* for these real-world images. Instead, we will classify the pixels into k groups according to the intensity level, with k assumed to be given.

The images in Figures 16 and 17 are widely used in image processing. We apply our

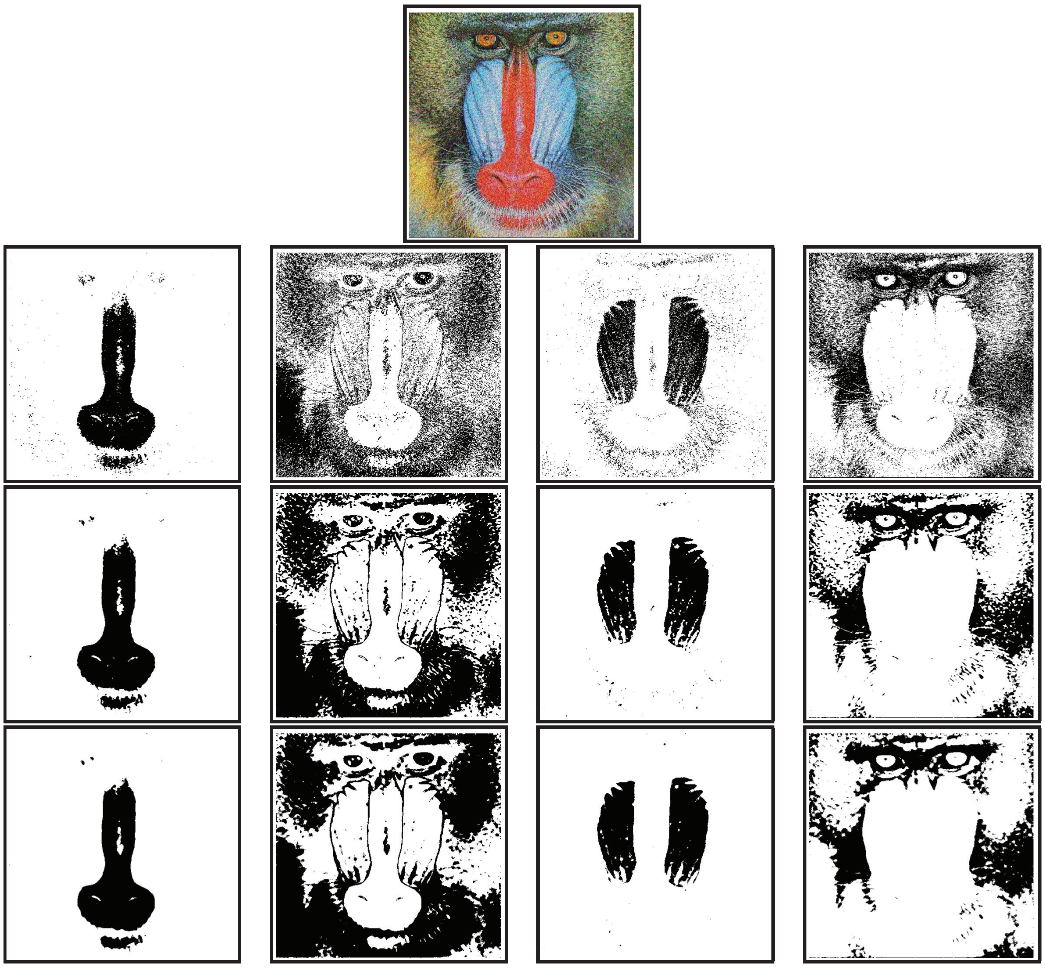

Figure 16. *Classifying a real image into four groups. The first row shows the original color image. The second row shows the classification using the original EM algorithm. We plot our classification without feedback (third row) and with one feedback (fourth row).*

segmentation algorithm using $k = 4$ and $k = 3$, respectively. Since the image intensity is the only input measurement of the segmentation algorithm, our method separates out the red nose (Figure 16) and the green peppers (Figure 17) reasonably well.

The brain image in Figure 18 is taken from [13]. Even though we choose $k = 4$ as an input to the algorithm, our method automatically eliminated a redundant class, leaving only three groups in the output. The white matter in the brain is reasonably well segmented out from both the background and the grey matter.

6. Conclusion. In this paper, we propose an expectation-maximization (EM) algorithm with local adaptivity for image segmentation. A GMM is used due to its simplicity and power to express a variety of images. The idea of our approach is to incorporate statistical and geometric information (anisotropy, orientation) in selecting an appropriate neighborhood to improve the classification rate and preserve fine features of noisy images. Experiments are

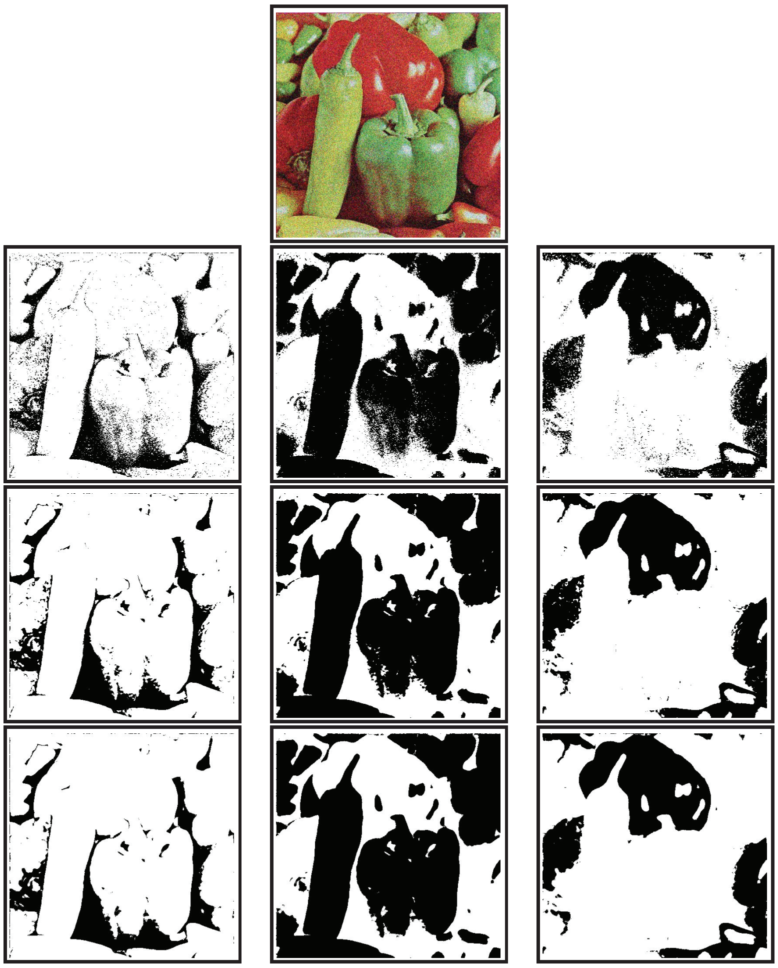

Figure 17. *Classifying a real image into three groups. The first row shows the original color image. The second row shows the classification using the original EM algorithm. We plot our classification without feedback (third row) and with one feedback (fourth row).*

conducted to access the performance of our algorithm for various images. The results show that our algorithm is robust and outperforms many existing image segmentation methods. Furthermore, our proposed methodology is flexible and can be easily generalized to deal with other inferred information/quantities and statistical methods/models.

Appendix: Misclassification rate minimization.

Proof of Theorem 3.1. For the ellipse $D(\mathbf{x})$ centered at a given pixel at $\mathbf{x}$, it is our assumption that all pixels within the ellipse belong to the same group. Hence, the likelihood that the whole ellipse is of class k is

$$L_k = L(\hat{\mu}_k, \hat{\Sigma}_k; \mathcal{D}(\mathbf{x})) = \prod_{\mathbf{z}\in\mathcal{D}(\mathbf{x})} \frac{1}{\sqrt{2\pi}} |\hat{\Sigma}_k|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{z}-\hat{\mu}_k)^T \hat{\Sigma}_k^{-1}(\mathbf{z}-\hat{\mu}_k)\right), \; k = 1, \dots, K,$$

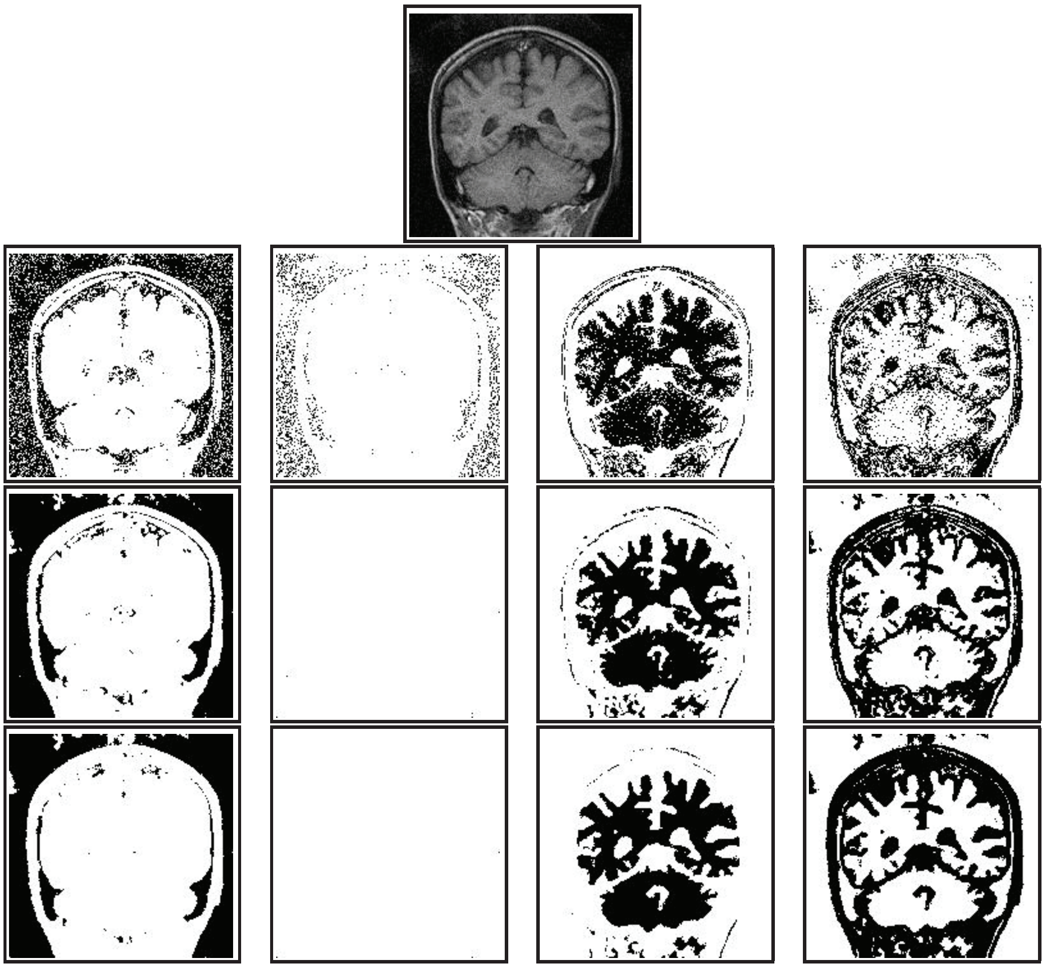

Figure 18. *Classifying a real image into four groups. The first row shows the original image. The second row shows the classification using the original EM algorithm. We plot our classification without feedback (third row) and with one feedback (fourth row).*

where $\hat{\mu}_k$ and $\hat{\Sigma}_k$ are the mean and variance of the kth group. (In practice, we use the estimated version of means and variances.)

A decision rule $\mathcal{A}$ is a partition of the sample space Ω: $A_1, \dots, A_K$; then the expected misclassification rate of a decision rule $\mathcal{A}$ is

$$\begin{aligned} m(\alpha) &= \sum_{k=1}^{K} \int_{A_k} \sum_{j \neq k} \hat{\alpha}_j L_j \, \mathrm{d}\mathcal{D}(x) \\ &= \int_{\Omega} \sum_{j=1}^{K} \hat{\alpha}_j L_j \, \mathrm{d}\mathcal{D}(x) - \sum_{k=1}^{K} \int_{A_k} \hat{\alpha}_k L_k \, \mathrm{d}\mathcal{D}(x). \end{aligned}$$

In the above equation, the first term is a constant. In order to minimize the whole term, we need to pick the classification region A_k such that the second term is maximized. Note that each region will be counted only once in the second term, so in order to minimize the misclassification rate, the optimal decision rule is to classify pixels based on the maximum of local likelihoods,

$$\arg\max_{1\le k\le K} \hat{\alpha}_k L(\hat{\mu}_k, \hat{\Sigma}_k; \mathcal{D}(\mathbf{x})). \qquad \text{(A.1)}$$

■

REFERENCES

[1] W. Abd-Almageed and L.S. Davis, *Density estimation using mixtures of mixtures of Gaussians*, in Computer Vision—ECCV 2006, Part IV, Lecture Notes in Comput. Sci. 3954, Springer-Verlag, Berlin, pp. 410–422.

[2] S. Awate, T. Tasdizen, and R.T. Whitaker, *Unsupervised texture segmentation with nonparametric neighborhood statistics*, in Computer Vision—ECCV 2006, Part II, Lecture Notes in Comput. Sci. 3952, Springer-Verlag, Berlin, pp. 494–507.

[3] J. Besag, *Spatial interaction and the statistical analysis of lattice systems*, J. Roy. Statist. Soc. Ser. B, 36 (1974), pp. 192–236.

[4] J. Besag, *On the statistical analysis of dirty pictures*, J. Roy. Statist. Soc. Ser. B, 48 (1986), pp. 259–302.

[5] C. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[6] G.E.P. Box and D.R. Cox, *An analysis of transformations*, J. Roy. Statist. Soc. Ser. B, 26 (1964), pp. 211–246.

[7] X. Bresson, S. Esedoḡlu, P. Vandergheynst, J.P. Thiran, and S. Osher, *Global Minimizers of the Active Contour/Snake Model*, UCLA CAM Report (05-04), University of California, Los Angeles, 2005.

[8] T. Brox and D. Cremers, *On local region models and a statistical interpretation of the piecewise smooth Mumford-Shah functional*, Int. J. Comput. Vision, 84 (2009), pp. 184–193.

[9] M.A. Carreira-Perpinan, *Gaussian mean shift is an EM algorithm*, IEEE Trans. Pattern Anal. Mach. Intell., 29 (2007), pp. 767–776.

[10] V. Caselles, R. Kimmel, and G. Sapiro, *Geodesic active contours*, Int. J. Comput. Vision, 22 (1997), pp. 61–79.

[11] T. Chan and L. Vese, *Active contours without edges*, IEEE Trans. Image Process., 10 (2001), pp. 266–277.

[12] T. Chan and W. Zhu, *Level set based shape prior segmentation*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 20–25.

[13] K. Chen, D.L. Wang, and X. Liu, *Weight adaptation and oscillatory correlation for image segmentation*, IEEE Trans. Neural Netw., 11 (2000), pp. 1106–1123.

[14] Y. Chen, F. Huang, H.D. Tagare, M. Rao, D. Wilson, and E.A. Geiser, *Using prior shape and intensity profile in medical image segmentation*, in Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003, pp. 1117–1124.

[15] C. Constantinopoulos, M.K. Titsias, and A. Likas, *Bayesian feature and model selection for Gaussian mixture models*, IEEE Trans. Pattern Anal. Mach. Intell., 28 (2006), pp. 1013–1018.

[16] D. Cremers, T. Kohlberger, and C. Schnörr, *Shape statistics in kernel space for variational image segmentation*, Pattern Recognition, 36 (2003), pp. 1929–1943.

[17] D. Cremers, S. Osher, and S. Soatto, *Kernel density estimation and intrinsic alignment for knowledge-driven segmentation: Teaching level sets to walk*, in Pattern Recognition, Lecture Notes in Comput. Sci. 3175, Springer-Verlag, Berlin, 2004, pp. 36–44.

[18] D. Cremers, M. Rousson, and R. Deriche, *A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape*, Int. J. Comput. Vision, 72 (2007), pp. 195–215.

[19] D. Cremers and C. Schnörr, *Statistical shape knowledge in variational motion segmentation*, Image Vision Comput., 21 (2003), pp. 77–86.

[20] G. Gilboa and S. Osher, *Nonlocal Operators with Applications to Image Processing*, UCLA CAM Report 07-23, University of California, Los Angeles, 2007.

[21] K. Held, E.R. Kops, B.J. Krause, W.M. Wells, R. Kikinis, and H.-W. Muller-Gartner, *Markov random field segmentation of brain MR images*, IEEE Trans. Med. Imaging, 16 (1997), pp. 878–886.

[22] M.H.C. Law, M.A.T. Figueiredo, and A.K. Jain, *Simultaneous feature selection and clustering using mixture models*, IEEE Trans. Pattern Anal. Mach. Intell., 26 (2004), pp. 1154–1166.

[23] S. Leung and S. Osher, *Fast global minimization of the active contour model with TV-inpainting and two-phase denoising*, in Proceedings of the 3rd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision, 2005, pp. 149–160.

[24] M. Leventon, W. Grimson, and O. Faugeraus, *Statistical shape influence in geodesic active contours*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00), 2000, pp. 316–323.

[25] D. Lu, H. Zhao, M. Jiang, S. Zhou, and T. Zhou, *A surface reconstruction method for highly noisy point clouds*, in Variational, Geometric, and Level Set Methods in Computer Vision, Lecture Notes in Comput. Sci. 3752, Springer-Verlag, Berlin, 2005, pp. 283–294.

[26] B.G. McLachlan and D. Peel, *Finite Mixture Models*, Wiley-Interscience, New York, 2000.

[27] D. Mumford and J. Shah, *Optimal approximation by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., 42 (1998), pp. 577–685.

[28] O. Nemitz, M. Rumpf, T. Tasdizen, and R. Whitaker, *Anisotropic curvature motion for structure enhancing smoothing of 3D MR angiography data*, J. Math. Imaging Vision, 27 (2007), pp. 217–229.

[29] P. Perona and R. Malik, *Scale space and edge detection using anisotropic diffusion*, IEEE Trans. Pattern Anal. Mach. Intell., 12 (1990), pp. 629–639.

[30] K.M. Pohl, R. Kikinis, and W.M. Wells, *Active mean fields: Solving the mean field approximation in the level set framework*, in Information Processing in Medical Imaging, Lecture Notes in Comput. Sci. 4584, Springer-Verlag, Berlin, 2008, pp. 26–37.

[31] C.E. Priebe, *Adaptive mixtures*, J. Amer. Statist. Assoc., 89 (1994), pp. 796–806.

[32] O. Rotem, H. Greenspan, and J. Goldberger, *Combining region and edge cues for image segmentation in a probabilistic Gaussian mixture framework*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07), 2007, pp. 1–8.

[33] M. Rousson and D. Cremers, *Efficient kernel density estimation of shape and intensity priors for level set segmentation*, in Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Lecture Notes in Comput. Sci. 3750, Springer-Verlag, Berlin, 2005, pp. 757–764.

[34] M. Rousson and R. Deriche, *A variational framework for active and adaptative segmentation of vector valued images*, in Proceedings of the IEEE Workshop on Motion and Video Computing, 2002, pp. 56–61.

[35] M. Rousson and N. Paragios, *Shape priors for level set representations*, in Proceedings of the 7th European Conference on Computer Vision (ECCV), Part II, Lecture Notes in Comput. Sci. 2351, Springer-Verlag, Berlin, 2002, pp. 78–92.

[36] M. Rousson and N. Paragios, *Prior knowledge, level set representations and visual grouping*, Int. J. Comput. Vision, 76 (2008), pp. 231–243.

[37] L. Rudin, S.J. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.

[38] S.R. Thiruvenkadam, T.F. Chan, and B.-W. Hong, *Segmentation under occlusions using selective shape prior*, SIAM J. Imaging Sci., 1 (2008), pp. 115–142.

[39] M. Unger, P. Thomas, T. Werner, D. Cremers, and B. Horst, *TVSeg - Interactive total variation based image segmentation*, in Proceedings of the British Machine Vision Conference, 2008.

[40] M.N.M. Van Lieshout, *Markov Point Processes and Their Applications*, Imperial College Press, London, 2000.