UNSUPERVISED IMAGE SEGMENTATION UTILIZING PENALIZED INVERSE EXPECTATION MAXIMIZATION ALGORITHM

Jesmin F. Khan, Reza R. Adhami, Sharif M. A. Bhuiyan

University of Alabama in Huntsville
Department of ECE
Huntsville, AL 35899, USA

ABSTRACT

This work is on accurate segmentation of images using local image characteristics. An appropriate Gabor filter with customized size, orientation, frequency and phase for each pixel is selected to measure the image features. A new image property called *phase divergence* is introduced to select the filter size. Brightness, color, texture and position features are extracted for each pixel and the joint distribution of these pixel features is modeled by a mixture of Gaussians. A new version of the expectation maximization (EM) algorithm called Penalized Inverse EM (PIEM) is formulated for estimating the parameters of the mixture of Gaussians model. Furthermore, we determine the number of models that best suits the image based on Schwarz criterion. The performance on the Berkeley segmentation benchmark proves the efficacy and accuracy of the proposed method.

Index Terms - Clustering, EM, segmentation, Schwarz criterion.

1. INTRODUCTION

There are a wide variety of methods in the literature for color image segmentation, which can be categorized as: edge detection, histogram thresholding, region based methods, characteristic feature clustering, neural network, and statistical approach [1]. Histogram thresholding is the most widely used technique for image segmentation [2, 3]. Clustering of characteristic features is the multidimensional extension of the concept of thresholding [4, 5]. In this paper we use multiple features clustering to perform image segmentation. The outline of the segmentation algorithm involves three stages. 1) Selecting an appropriate Gabor filter of adaptive size, orientation, frequency and phase for each pixel, to extract brightness, color, texture and position features. We propose a novel method of filter size selection, which is based on a newly introduced local image property called *phase divergence*. 2) Clustering the pixels into regions by modeling the joint distribution of pixel features with a mixture of Gaussians. To estimate the parameters of this model a novel EM algorithm - PIEM is employed and the resulting pixel-cluster memberships provide the segmentation of that image. This proposed PIEM algorithm is a combination of existing PEM [6] and IEM [7]. 3) Finally, determining the optimum number of clusters that best suits a given image, using Schwarz criterion [8]. This criterion provides an attractive way to perform data clustering without knowing the exact number of clusters.

2. FEATURE EXTRACTION

To formulate a Gabor filter with customized size, w; phase, ϕ; orientation, θ; and frequency, f, we consider 8 different sizes of the filter as w$= 2k-1$, for $k = 1, 2, \ldots, 8$. Given an image $I(x, y)$, its analytic image is defined as $I_A(x, y) = I(x, y) - jI_H(x, y)$, where $I_H(x, y)$ is the Hilbert transform of $I(x, y)$. After having the Hilbert transform, we can compute the local phase, which is referred to as the argument of the analytic image. For a window size w$= 2k-1$, we represent the phase at pixel (x, y) as, $\phi_k(x, y)$.

We have proposed a new image property called *phase divergence* that is used for selecting the controlling size of the filter window, which is also called the integration scale. Phase divergence is a measure of the extent to which the phase angles in a certain window around a pixel are in the same direction with respect to the phase angle of that pixel. The phase divergence at pixel (x, y) for a given window size w$= 2k-1$, as we define it, is

$$\bigtriangledown\phi_k(x, y) = \frac{|\phi_{k+}(x, y) - \phi_{k-}(x, y)|}{\phi_{k+}(x, y) + \phi_{k-}(x, y)} \tag{1}$$

where, $\phi_{k+}(x, y)$ and $\phi_{k-}(x, y)$ are the number of phase angles in the window w, that are on the positive side and the negative side, respectively, of the phase angle at pixel (x, y). By computing the phase divergence for w$= 2k-1, k = 1, 2, \ldots, 8$; we produce a stack of phase divergence images across scale, k. Then, the phase divergence image at scale k is smoothed by convolving with a Gaussian filter of size, w. Finally the scale is selected based on the derivative of the phase divergence with respect to scale. For a given pixel (x, y) the scale $\hat{k}(x, y)$ is selected as the first value of $k(x, y)$ for which the difference between values of phase divergence at successive scales $(\bigtriangledown\phi_k(x, y) - \bigtriangledown\phi_{k-1}(x, y))$ is less than 1%. In the uniform regions of an image, the selected scale should be 1, because uniform region appears not to change across scale. A region can be declared to be uniform if the mean contrast (standard deviation of intensity) of that region across scale is less than 0.1 [9].

The method of orientation determination that is used in this paper is based on the idea of utilizing the information present in the horizontal and vertical gradients, $\bigtriangledown_x$ and $\bigtriangledown_y$ of the image, respectively. The orientation angle at pixel (x, y) is determined by following the approach proposed in [10],

$$\theta_k(x, y) = 90^\circ + \frac{1}{2}\tan^{-1}\left(\frac{2G_{xy}}{G_{xx} - G_{yy}}\right), \tag{2}$$

where, $G_{xy} = \sum_{\mathrm{w}} \bigtriangledown_x \bigtriangledown_y$, $G_{xx} = \sum_{\mathrm{w}} \bigtriangledown_x^2$, $G_{yy} = \sum_{\mathrm{w}} \bigtriangledown_y^2$.

The spatial derivative of the local phase gives the local frequency of an image. For the scale k we denote the frequency at pixel (x, y) as $f_k(x, y)$, which is estimated by computing the standard deviation of the phase values in the window of size w. After selecting the appropriate scale $\hat{k}(x, y)$ for a given pixel, we compute filter size $\hat{\mathrm{w}}_k(x, y)$, phase $\hat{\phi}_k(x, y)$, orientation $\hat{\theta}_k(x, y)$ and frequency $\hat{f}_k(x, y)$ at that scale. Thus, finally we have an appropriate Gabor

filter with customized parameters for each pixel that is denoted as $G_{k,\theta,\phi,f}(x,y)$.

2.1. Image Features

We consider the following features for image segmentation such as, two brightness features: brightness gradient and local energy content of the L* component; three color features: color gradient, local energy content of the a* and b* components; three texture features: phase divergence, homogeneity and homogeneous intensity; and two position features: (x,y) coordinates of the pixels. These features are discussed below.

Local energy is defined as the square root of the sum of the squared response of two matched filters - one has the even-symmetric and the other has the odd-symmetric line-spread function [11].

$$LE^{L^*} = \left(L^* * *G^o_{k,\theta,\phi,f}\right)^2 + \left(L^* * *G^e_{k,\theta,\phi,f}\right)^2 \quad (3)$$

where, $G^o_{k,\theta,\phi,f}(x,y)$ and $G^e_{k,\theta,\phi,f}(x,y)$ are the pair of odd and even symmetric filters, which are the real and imaginary part of the complex Gabor filter $G_{k,\theta,\phi,f}(x,y)$, respectively.

Gradient based features detect local changes. To measure the gradient feature we place a window of size $\hat{\mathrm{w}}_k(x,y) \times \hat{\mathrm{w}}_k(x,y)$ at pixel (x,y) and divide that window along the line at an orientation $\hat{\theta}_k(x,y)$. We then compare the histogram contents of the resulting two regions by using the χ^2 histogram difference operator [12], given by $\chi^2(s,t) = \frac{1}{2}\sum_{m=0}^{255} \frac{(s_m - t_m)^2}{s_m + t_m}$, where s and t are the two histograms of the two halves of the square window at (x,y) and m represents the bins of the histograms. For the brightness gradient we compute histograms of L* image. We determine the marginal color gradients for a* and b* and consider the full color gradient to be the sum of these two marginal gradients: $CG = CG^a + CG^b$ [12]. This is justified by the fact that the a* and b* channels are orthogonal with each other.

The homogeneity feature is calculated by taking into account both local and global information [3]. Homogeneity is defined as a composition of standard deviation and discontinuity of the intensities. At pixel (x,y), the homogeneity is expressed as

$$H(x,y) = 1 - E(x,y) \times V(x,y), \quad (4)$$

where $E(x,y)$ is the normalized gradient magnitude calculated by employing a Sobel filter of 3×3 window, and $V(x,y)$ is the normalized standard deviation of the intensities within the window $\hat{\mathrm{w}}_k \times \hat{\mathrm{w}}_k$. The homogeneity value at each pixel has a range from 0 to 1. The more uniform the local region surrounding a pixel is, the larger the homogeneity value the pixel has.

Homogeneity value is used for replacing the intensity values of the L* image by the corresponding homogeneous intensity. For a given intensity value, L*$(x,y) = I$ chosen from 0 to 255, the steps for computing the homogeneous intensity is as follows: 1) First, the homogeneity values corresponding to I, that are greater than a threshold value are added to give $H_t(I)$. 2) Then $H_t(I)$ is divided by the number of pixels which have the intensity value equal to I to produce $H_{nt}(I)$. 3) Finally, the homogeneous intensity corresponding to the intensity I is the multiplication of $H_{nt}(I)$ and I. The homogeneous intensity reflects how uniform a region is and plays an important role in image segmentation, since the outcome of image segmentation would be several homogeneous regions.

The last texture descriptor, phase divergence, can be calculated using Eq. 1 at the selected scale for each pixel (x,y). Including the (x,y) coordinates as position features, we have 10 features, which form a 10-dimensional feature vector for each pixel. The orientation, frequency and scale do not appear in the feature vector formulated by the brightness, color, texture and position descriptor. As a result, clustering can occur across variations in orientation, frequency and scale.

3. CLUSTERING AND EM ALGORITHM

After extracting all the features for an image of size $M \times N$ we have a set of $L = M \times N$ feature vectors, which can be regarded as points in a ten-dimensional feature space. In order to cluster those points, we make use a new variant of EM algorithm, PIEM to determine the parameters of a mixture of C Gaussians in the feature space. The basic idea in the EM algorithm is to iteratively find the maximum likelihood estimate of the unknown parameters (associated with a sample of observations), which maximize the probability density function of the sample, called the likelihood function. The Gaussian mixture model that we consider is given by the data consists of L independent random samples $X = x_1, x_2, \ldots, x_L$ from a C-component Gaussian mixture, $P_{x_i|\Theta} = \sum_{j=1}^{C} p_j N(x_i|\mu_j, \Sigma_j)$, where x_i is a feature vector; p_j are the mixing weights, $0 < p_j < 1$ satisfying $\sum_{j=1}^{C} p_j = 1$; and $N(x_i|\mu_j, \Sigma_j)$ is the multivariate normal density of j-class parameterized by μ_j and Σ_j. The unknown parameters of the mixture model and the log-likelihood can be denoted as

$$\Theta = (p_1, p_2, \ldots, p_C, \mu_1, \mu_2, \ldots, \mu_C, \Sigma_1, \Sigma_2, \ldots, \Sigma_C).$$
$$\mathcal{L}(\Theta) = \log\left[\prod_{i=1}^{L} P_{x_i|\Theta}\right] = \sum_{i=1}^{L} \log \sum_{j=1}^{C} p_j N(x_i|\mu_j, \Sigma_j)$$

Maximum likelihood parameter estimate $\hat{\Theta}$ may efficiently be computed with the iterative application of the following two steps of EM algorithm [13]: 1) In the E-step, based on the current parameter estimates at k-th iteration, the posterior probability that class ω_j is responsible for the generation of sample x_i is estimated as $P_{\omega_j|x_i,\Theta^{(k)}} = \frac{p_j^{(k)} N(x_i|\mu_j^{(k)}, \Sigma_j^{(k)})}{\sum_{j=1}^{C} p_j^{(k)} N(x_i|\mu_j^{(k)}, \Sigma_j^{(k)})}$.

2) In the M-step, we obtain new parameter estimates

$$\begin{aligned} p_j^{(k+1)} &= \frac{1}{L}\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}} \\ \mu_j^{(k+1)} &= \frac{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}} x_i}{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}}} \\ \Sigma_j^{(k+1)} &= \frac{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}} (x_i - \mu_j^{(k+1)})(x_i - \mu_j^{(k+1)})^T}{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}}} \end{aligned} \quad (5)$$

3.1. Penalized EM algorithm

The EM learning algorithm can lead to overfitting during the maximization of the likelihood function due to singularities and local maxima in the log-likelihood function. As a solution to this problem, in PEM, a penalty term is added to the log-likelihood function as a regularizer [6]. In particular the negative logarithm of a conjugate prior is chosen as the penalty function so that EM update rules can be derived to obtain the optimal parameter estimates.

A conjugate prior of a single multivariate normal density is a product of a normal density $N(\mu_j|\nu_j, \eta_j^{-1}\Sigma_j)$ and a Wishart density $W_j(\Sigma_j^{-1}|\alpha_j, \beta_j)$ [14]. A proper conjugate prior for the mixture weights $\vec{p} = (p_1, p_2, \ldots, p_C)$ is a Dirichlet density $\mathcal{D}(\vec{p}|\vec{\gamma})$. $\alpha_j, \beta_j, \gamma_j, \eta_j$ and ν_j are hyperparameters which are defined using three "equivalent sample sizes" τ_p, τ_μ and τ_Σ [6].

Consider an additional data set Y^* of size L' such that Y_j^* of size l_j denotes the subset of Y^* generated by Gaussian j, then the hyperparameters can be expressed as $\alpha_j = \frac{\tau_\Sigma + d}{2}$, $\gamma_j = \tau_p + 1$,

$\eta_j = \tau_\mu$, $\beta_j = \frac{\tau_\Sigma}{2}\tilde{S}_j$, $\nu_j = \bar{Y}_j^*$; for $j = 1, 2, \ldots, C$. The concrete values for the statistics, $\bar{Y}_j^*$ and $\tilde{S}_j$ are chosen as 0 and $I^{d\times d}$, respectively. Next the degree of regularization is determined by varying the equivalent sample sizes and those values are set to $\tau_p = \tau_\mu = 0$ and $\tau_\Sigma = 0.1$ as suggested in [6].

For the PEM algorithm, the prior of the Gaussian mixture is $p_j = \mathcal{D}(\vec{p}|\vec{\gamma})\prod_{j=1}^{C} N(\mu_j|\nu_j, \eta_j^{-1}\Sigma_j)W_j(\Sigma_j^{-1}|\alpha_j, \beta_j)$. In the case of PEM [6], 1) the MAP parameter estimate maximizes the following log-posterior function,

$$\begin{aligned}\mathcal{L}_{\mathrm{p}}(\Theta) = &\textstyle\sum_{i=1}^{L}\log\sum_{j=1}^{C} p_j N(x_i|\mu_j, \Sigma_j) + \log\mathcal{D}(\vec{p}|\vec{\gamma}) \\ &\textstyle+\sum_{j=1}^{C}\left[\log N(\mu_j|\nu_j, \eta_j^{-1}\Sigma_j) + \log W_j(\Sigma_j^{-1}|\alpha_j, \beta_j)\right]\end{aligned} \quad (6)$$

2) the E-step remains identical to the EM algorithm and 3) the M-step becomes

$$\begin{aligned} p_j^{(k+1)} &= \frac{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}} + \gamma_j - 1}{L + \sum_{j=1}^{C}\gamma_j - C} \\ \mu_j^{(k+1)} &= \frac{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}} x_i + \eta_j\nu_j}{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}} + \eta_j} \end{aligned} \quad (7)$$

$$\begin{aligned}\Sigma_j^{(k+1)} &= \frac{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}}(x_i - \mu_j^{(k+1)})(x_i - \mu_j^{(k+1)})^T}{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}} + 2\alpha_j - d} \\ &+ \frac{\eta_j(\mu_j^{(k+1)} - \nu_j)(\mu_j^{(k+1)} - \nu_j)^T + 2\beta_j}{\sum_{i=1}^{L} P_{\omega_j|x_i,\Theta^{(k)}} + 2\alpha_j - d}\end{aligned}$$

The described PEM procedure leads to faster convergence and Σ_j does not approach the null matrix, as often happened in case of standard EM.

3.2. Inverse EM algorithm

In the IEM algorithm [7], the basic modification consists of virtually updating the observed covariance matrices in the first stage and then, in the second stage, the reversed updating of the estimated covariances. This algorithm is devoted to the estimation of the parameters of multivariate Gaussian mixture where the covariance matrices are constrained to have a linear structure such as Toeplitz, Hankel, or circular constraints. The E-step for IEM is the same as EM and the M-step updates are as follows. 1) At first, a basis $\{\mathcal{Q}\}_{l=1}^{M}$ (independent matrices, not necessarily orthogonal) of the constrained linear space $\mathcal{S}$ is provided, where M is the dimension of the constrained linear space. 2) The parameters $p_j^{(k+1)}$, $\mu_j^{(k+1)}$ and $A_j^{(k+1)}$ are updated following the EM algorithm at the $(k+1)$-th iteration, where A_j is the observed weighted covariance matrix. Next, the empirical covariance matrix Γ_j is calculated as $\Gamma_j^{(k+1)} = \frac{A_j^{(k+1)} + \frac{\upsilon_j}{|\rho_j|}E_j^{-1}}{1 + \frac{\upsilon_j + d + 1}{|\rho_j|}}$, where E_j is a positive definite matrix belong to the class j, and υ_j is the degree of the freedom of the distribution for Γ_j, which is inverse Wishart density.

3) Finally, the unknown covariance matrices are $\Sigma_j^{(k+1)} = \Sigma_j^{(k)} + a_k D_j$, where $D_j = \sum \hat{x}_l Q_l - \Sigma_j^{(k)}$ is the amount of increment needed in $\Sigma_j^{(k)}$ and a_k is the step size. Now, the vector $\hat{x}$ needs to be computed which can be expressed as $\hat{x} = B^{-1}b$, where $B_{il} = \text{trace}\left[{\Sigma_j^{(k)}}^{-1} Q_l {\Sigma_j^{(k)}}^{-1} Q_i\right]$, $b_i = \text{trace}\left[{\Sigma_j^{(k)}}^{-1}\Gamma_j^{(k)}{\Sigma_j^{(k)}}^{-1} Q_i\right]$; for $i, l = 1, \ldots, M$.

The step size a_k is calculated as $a_k = \frac{\text{trace}[E]}{2\text{trace}[F] - \text{trace}[E]}$, where $E = {\Sigma_j^{(k)}}^{-1} D_j {\Sigma_j^{(k)}}^{-1} D_j$ and $F = {\Sigma_j^{(k)}}^{-1} D_j {\Sigma_j^{(k)}}^{-1} D_j {\Sigma_j^{(k)}}^{-1}\Gamma_j^{(k)}$.

4) If the updated $\Sigma_j^{(k+1)}$ is non positive, then a_k is changed by $\frac{a_k}{2}$ and the algorithm returns to step 3, otherwise the iteration number k is incremented by one and the algorithm iterates back from step 2.

3.3. Penalized Inverse EM algorithm

In this paper, we have incorporated the favorable features of both the PEM and the IEM to formulate PIEM. Similar to the IEM algorithm, the PIEM algorithm also starts with the generation of a basis $\{\mathcal{Q}\}_{l=1}^{M}$ of the constrained linear space $\mathcal{S}$. Then at the second step the parameters $p_j^{(k+1)}$, $\mu_j^{(k+1)}$ and $\Gamma_j^{(k+1)}$ are updated following the PEM algorithm [6]. Finally the covariance matrices of the Gaussian mixture models at $(k+1)$-th iteration $\Sigma_j^{(k+1)}$ are calculated by following steps 3 and 4 of the Inverse EM algorithm. This algorithm is found to converge much faster than the EM algorithms discussed previously. Moreover, the clustering performance is proved to be the best for the proposed PIEM algorithm.

3.4. Clustering Using PIEM

The above discussed PIEM algorithm is applied for the clustering of a set of L feature vectors. Given the number of clusters C and the image to be segmented, we randomly divide the image in C windows. Then, the means of the Gaussians are initialized by the average feature vectors in those windows and the mixing weights are initialized by the number of feature vectors present in those windows normalized by the total number of feature vectors. The initial covariance matrices are set to be the identity matrix.

The approach for estimating the number of clusters C, is based on choosing $\hat{C}$ which will maximize $p(C|X)$. By Bayes theorem, $p(C|X) = \frac{p(X|C)p(C)}{p(X)}$ and $p(X|C) = \int p(X|\Theta)\pi_C(\Theta)d\Theta$, where $\pi_C(\Theta)$ is the prior probability, which is rarely known and $p(C) = \frac{1}{C}$. The Schwarz approximation [8] for $p(X|C)$ is, $\hat{m}_C = \hat{p}(X|C) = L^{-\frac{n}{2}}p(X|\hat{\Theta}_C)$, where $n = C - 1 + Cd + C\frac{d(d+1)}{2}$ is the number of unknown parameters. Then $p(C|X)$ can be estimated as $\hat{p}(C|X) = \frac{\hat{m}_C}{\sum_i \hat{m}_i}$ and the optimum number of clusters $\hat{C}$ will be chosen in such way, that $\log\hat{p}(C|X)$ or equivalently $\log\hat{m}_C$ is maximized where $\log\hat{m}_C = -\frac{n}{2}\log L + \mathcal{L}(\Theta)$.

In our experiment we first estimate an empirical value of the number of clusters $\tilde{C}$ from the number of valleys present in the histogram of the intensity image. Then the optimum value for C is determined by employing the Schwarz criterion for C ranging from $(\tilde{C} - 1)$ to $(\tilde{C} + 1)$, satisfying $C \geq 2$.

For clustering, we produce an image of C-levels by encoding the pixel-cluster relationships where each pixel value is replaced by the cluster label for which it attains the highest likelihood value, $P_{x_i|\Theta}$. At this stage, the clustered image is spatially smoothened by the repeated application of 3×3 mode filter (the value that occurs most often) until the number of pixels that are different between two successive images is less than 1%. In the smoothed image, there may be some regions of a few pixels located inside another large region. Those smaller regions are required to be classified as the member of that large region to produce homogeneous and united regions in the segmented image, which is accomplished by employing a connected-component algorithm [15].

4. EXPERIMENTAL RESULTS

We have applied the proposed scheme for the segmentation of several images. In this section, we present the experimental results ob-

tained at different stages of the clustering algorithm. At first, the histogram of an image is analyzed to predict the approximate number of clusters. From the histogram of the image shown in Fig 1 (a), the estimated number of clusters is $\tilde{C} = 3$. The clustered images obtained from PIEM for $C = 2, 3, 4$ is illustrated in Fig 1 (b), (c), and (d), respectively. According to the Schwarz criterion, $C = 2$ gives the best segmentation performance. For $C = 2, 3, 4$ the final segmented images obtained after the employment of spatial smoothening and connected-component algorithm are presented in Fig 1 (e),(f) and (g), respectively.

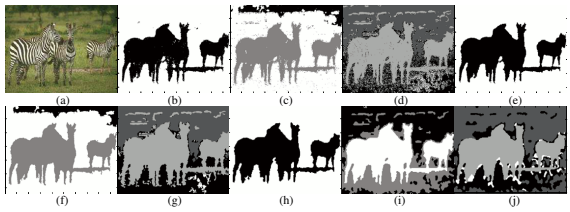

Fig. 1. a) Original image. Clustering using PIEM for (b) $C = 2$, (c) $C = 3$, (d) $C = 4$. Final segmentation result after the postprocessing for (e) $C = 2$, (f) $C = 3$, (g) $C = 4$. Results from the sampled data for (h) $C = 2$, (i) $C = 3$, (j) $C = 4$

The segmentation results from the PIEM algorithm are generally good. However, the computational burden is significant, because the number of feature samples is very large for the images of size 321×481. Hence, to reduce the computational cost without compromising the quality of the segmentation, we formulate a good representative fragment of the whole data by retaining the alternate rows and columns of the image. After the clustering of all the pixels in this sampled data set using PIEM, the pixel-cluster membership computed for a pixel is assigned to the 3 neighbors of that pixel in a 2×2 window. In Fig 1 (h), (i) and (j) we present the results of clustering the sampled data for $C = 2, 3$ and 4 using PIEM. The segmentation quality of the sampled data set is as good as the original data. Fig 2 shows the segmentation results on the subsampled version of some randomly chosen typical images taken from the Berkley segmentation benchmark. From this result, it can be claimed that the presented method can be used as an effective and reliable technique for color image segmentation.

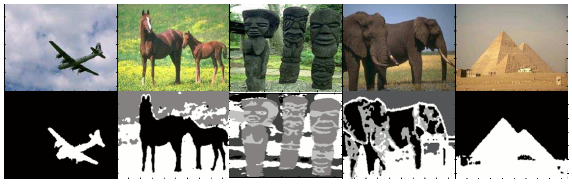

Fig. 2. Segementation results of some randomly chosen images.

5. CONCLUSION

In this paper we have presented an unsupervised color image segmentation algorithm, where we group the pixels into clusters based on color, texture, brightness and position features extracted by using a customized Gabor filter. A novel scale selection technique based on the proposed image property *phase divergence* is used to select the size of the filter. The proposed PIEM algorithm for clustering is the integration of PEM and IEM. The basic modification for PIEM consists of updating the observed covariance matrices following the PEM algorithm and then, the reverse updating of the estimated covariances according to the IEM algorithm. In addition, Schwarz criterion is employed to determine the appropriate number of clusters suitable for the image. We have also analyzed the performance of the clustering algorithm on a subsample version of the original image, to reduce the computational cost without degrading the segmentation quality. The proposed method performs favorably on the Berkley image segmentation Benchmark. The generality of this automatic system makes it applicable in a wide range of computer vision tasks.

6. REFERENCES

[1] Y. Sun H. Cheng, X. Jiang and J. Wang, "Color image segmentation: Advances and prospects," *Pattern Recognition*, vol. 34, pp. 2259–2281, 2001.

[2] Haralick R and Shapiro L, "Image segmentation techniques," *Computer Vision, Graphics and Image Processing*, vol. 29, pp. 100–132, 1985.

[3] H. Cheng and Y. Sun, "A hierarchical approach to color image segmentation using homogeneity," *IEEE Trans. on Image Processing*, vol. 9, no. 12, pp. 2071–2082, 2000.

[4] W. Chien H. Chen and S. Wang, "Contrast-based color image segmentation," *IEEE Signal Processing Letters*, vol. 11, no. 7, pp. 641–644, 2004.

[5] Q. Luo and T. Khoshgoftaar, "Unsupervised multiscale color image segmentation based on mdl principle," *IEEE Trans. on Image Processing*, vol. 15, no. 9, pp. 100–104, 2006.

[6] D. Ormoneit and V. Tresp, "Averaging, maximum penalized likelihood and bayesian estimation for improving Gaussian mixture probability density estimates," *IEEE Trans. on Neural Network*, vol. 9, no. 4, pp. 639–650, 1998.

[7] H. Snoussi and A. Mohammad-Djafari, "Estimation of structured Gaussian mixtures: The inverse EM algorithm," *IEEE Trans. on Signal Processing*, vol. 55, no. 7, pp. 3185–3191, 2007.

[8] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, 1978.

[9] H. Greenspan C. Carson, S. Belongie and J. Malik, "Blobworld: Image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. on PAMI*, vol. 24, no. 8, pp. 1027–1038, 2002.

[10] A. Rao, *A Taxonomy for Texture Description and Identification*, Springer-Verlag, New York, 1990.

[11] M. Morrone and D. Burr, "Feature detection in human vision: A phase-dependent energy model," *Proc. of the Royal Society of London*, vol. 235, no. 1280, pp. 221–245, 1988.

[12] C. Fowlkes D. Martin and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. on PAMI*, vol. 26, no. 5, pp. 1–19, 2004.

[13] N. Laird A. Dempster and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[14] A. Bruce and H. Y. Gao, *Bayesian Theory*, Wiley, New York, 1994.

[15] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison Wesley, Massachusetts, 1993.