UNSUPERVISED SEGMENTATION OF COLOR TEXTURED IMAGES USING A MULTI-LAYER MRF MODEL

Zoltan Kato[1], *Ting-Chuen Pong*[2], *Song Guo Qiang*[3]

[1] *University of Szeged, Dept. of Informatics, P.O.Box 652, H-6701 Szeged, Hungary, Fax:+36 62 420 292*
[2] *Hong Kong University of Science & Technology, Computer Science Dept., Clear Water Bay, Hong Kong, China, Fax:+852 2358 1477*
[3] *National University of Singapore, School of Computing, 3 Science Drive 2, Singapore 117543, Fax: +65 779 4580,*
E-mail: kato@inf.u-szeged.hu, tcpong@cs.ust.hk, songguoq@comp.nus.edu.sg

ABSTRACT

Herein, we propose a novel multi-layer Markov random field (MRF) image segmentation model which aims at combining color and texture features: Each feature is associated to a so called *feature layer*, where an MRF model is defined using only the corresponding feature. A special layer is assigned to the combined MRF model. This layer interacts with each feature layer and provides the segmentation based on the combination of different features. The model is quite generic and isn't restricted to a particular texture feature. Herein we will test the algorithm using Gabor and MRSAR texture features. Furthermore, the algorithm automatically estimates the number of classes at each layer (there can be different classes at different layers) and the associated model parameters.

1. INTRODUCTION

In this paper, we develop a Markovian approach to perform color textured image segmentation. The algorithm should identify all the regions and then assign each pixel to the most likely region type (i.e. we want to perform a pixel classification). This new MRF model should make use of multiple features in an efficient way during the segmentation process. Color feature usually preserves boundaries but it is more sensitive to local color variations. Hence color-based segmentation provides sharp edges but often inhomogeneous regions. On the other hand, texture features give us fuzzy boundaries but more homogeneous regions. By combining these two features, we hope to get sharp boundaries *and* homogeneous regions.

Basically, there are two approaches in the literature to color texture segmentation: One approach deals directly with *color textures* [1, 2]. In [1], an unsupervised segmentation algorithm is proposed which uses Gaussian MRF models for *color textures*. These models are defined in each color plane with interactions between different color planes. The segmentation algorithm is based on agglomerative hierarchical clustering. A different approach is presented in [2] which uses a multiband smoothing algorithm to generate a multiscale representation of an image. The smoothing is based on human psychophysical measurements of color appearance. First the coarsest level is clustered to isolate core clusters. Other pixels are then reassigned to these core clusters using a probabilistic assignment. Another frequently used approach tries to combine traditional gray level texture features together with pure color features [3]. Our approach falls into this category.

The novelty of our model can be summarized as follows: First, we use *different* features at different layers. This allows us to work with different models or to have varying number of regions at different layers, choosing the one which describes the best our feature data at a given layer. In addition, we have a special layer, called *combined layer*, which does not correspond to any feature but provides a way to combine different features. Our algorithm will automatically estimate the number of pixel classes and the associated model parameters at each layer. Only hyper-parameters are fixed a priori. Second, the layers are fully connected: each pixel interacts with the corresponding pixel at other layers. Multiscale pyramids have also been successfully applied for image segmentation [4]. In these models, each layer usually contains the *same* image data at different resolutions. However, we use *different* data at different layers and we do not perform subsampling, therefore our model is not a pyramid. In this respect, our model is similar to [1, 2].

2. MULTI-LAYER SEGMENTATION MODEL

We use perceptually uniform CIE-L*u*v* color values and texture features derived from the gray-level image. Segmentation requires simultaneous measurements in both spatial and frequency domain. Fortunately, the spatial resolution is considerably increased when we are combining texture features with color. Our model consists of 3 layers. At each layer, we use a first order neighborhood system and higher order inter-layer cliques (Fig. 1). The image features are represented by multi-variate Gaussian distributions. Let us denote the color layer by $\mathcal{S}^c$, the texture layer by $\mathcal{S}^t$ and the combined layer by $\mathcal{S}^x$. All layers are of the same size. Our MRF model is defined over the lattice $\mathcal{S} = \mathcal{S}^c \cup \mathcal{S}^x \cup \mathcal{S}^t$. For each site s, the region-type (or pixel class) that the site belongs to is specified by a class label, ω_s, which is modeled as a discrete random variable taking values in $\Lambda = \{1, 2, \dots, L\}$. The set of these labels $\omega = \{\omega_s, s \in \mathcal{S}\}$ is a random field, called the *label process*. Furthermore, the observed image features (color and texture) are supposed to be a realization $\mathcal{F} = \{\tilde{\mathbf{f}}_s | s \in \mathcal{S}^c \cup \mathcal{S}^t\}$ from another random field, which is a function of the label process ω. Basically, the *image process* $\mathcal{F}$ represents the deviation from the underlying label process. Thus, the overall segmentation model is composed of the hidden label process ω and the observable image process $\mathcal{F}$. Our goal is to find an optimal labeling $\hat{\omega}$ which maximizes the a posteriori probability $P(\omega \mid \mathcal{F})$, that is the *maximum a posteriori* (MAP) estimate [5]: $\arg\max_{\omega\in\Omega} P(\omega \mid \mathcal{F}) = \arg\max_{\omega\in\Omega} \prod_{s\in\mathcal{S}} P(\tilde{\mathbf{f}}_s \mid \omega_s) P(\omega)$, where Ω denotes the set of all possible labelings. We use the MMD algorithm [6] to obtain a good but theoretically suboptimal MAP estimate. Simulated Annealing could also be used at the price of higher CPU times. According to the *Hammersley-Clifford theorem* [5], $P(\omega \mid \mathcal{F})$ follows a Gibbs distribution:

$$P(\omega \mid \mathcal{F}) = \frac{\exp(-U(\omega))}{Z} = \frac{\prod_{C\in\mathcal{C}} \exp(-V_C(\omega_C))}{Z} \quad (1)$$

where $U(\omega)$ is called an *energy function*, Z is the normalizing constant and V_C denotes the *clique potential* of clique $C \in \mathcal{C}$ having the label configuration ω_C. Note that the energies of *singletons* (ie. cliques of single sites $s \in \mathcal{S}$) directly reflect the probabilistic modeling of labels without context, while higher order clique potentials express relationship between neighboring pixel labels. In the remaining part of this section, we will define these clique potentials for each layer.

2.1. Feature Layers

On the color layer, the observed image $\mathcal{F}^c = \{\tilde{\mathbf{f}}^c_s | s \in \mathcal{S}^c\}$ consists of three spectral component values (L*u*v*) at each pixel s denoted by the vector $\tilde{\mathbf{f}}^c_s$. We assume that $P(\tilde{\mathbf{f}}^c_s \mid \omega_s)$ follows a Gaussian distribution, the classes $\lambda \in \Lambda^c = \{1, 2, \dots, L^c\}$ are represented by the mean vectors $\tilde{\mu^{\mathbf{c}}}_\lambda$ and the covariance matrices $\mathbf{\Sigma^c}_\lambda$. The class label assigned to a site s on the color layer is denoted by ψ_s. The energy function $U(\psi, \mathcal{F}^c)$ of the so defined MRF layer has the following form:

$$\sum_{s\in\mathcal{S}^c} \mathcal{G}^c(\tilde{\mathbf{f}}^c_s, \psi_s) + \beta \sum_{\{s,r\}\in\mathcal{C}} \delta(\psi_s, \psi_r) + \gamma^c \sum_{s\in\mathcal{S}^c} V^c(\psi_s, \eta^c_s)$$

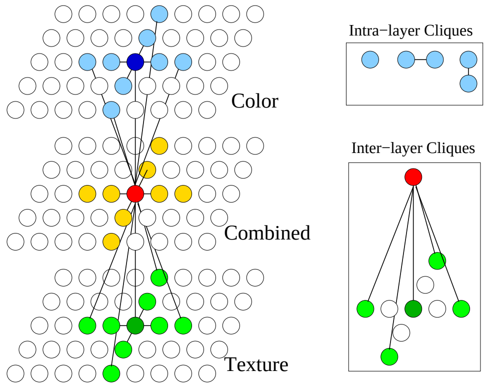

Fig. 1. Multi-layer MRF model.

Since we assume that $P(\tilde{\mathbf{f}}^c_s \mid \omega_s)$ is Gaussian, it follows from Eq. (1) that the corresponding energy potentials $\mathcal{G}^c(\tilde{\mathbf{f}}^c_s, \psi_s)$ should be log Gaussians:

$$\ln\left(\sqrt{(2\pi)^3 \mid \mathbf{\Sigma^c}_{\psi_s} \mid}\right) + \frac{1}{2}(\tilde{\mathbf{f}}^c_s - \tilde{\mu^{\mathbf{c}}}_{\psi_s})\mathbf{\Sigma^c}^{-1}_{\psi_s}(\tilde{\mathbf{f}}^c_s - \tilde{\mu^{\mathbf{c}}}_{\psi_s})^T \quad (2)$$

$\delta(\psi_s, \psi_r) = 1$ if ψ_s and ψ_r are different and -1 otherwise. $\beta > 0$ is a parameter controlling the homogeneity of the regions. As β increases, the resulting regions become more homogeneous. The last term ($V^c(\psi_s, \eta^c_s)$) is the inter-layer clique potential which will be defined later and γ^c is a parameter controlling the influence of the combined layer. As γ^c increases, the influence is higher.

On the texture layer, the observation consists of a set of Gabor [7] and MRSAR [8] image features. We adopt these two type of features because Gabor filters are good at discriminating strongly ordered textures while MRSAR features can better describe weakly ordered or random textures. Out of these features, we only need an optimal subset. The selection of the best features can also be automated [9] but herein we do not consider this issue. We will manually select the best 6-9 feature images.

The MRF model itself is the same as the color one with some obvious changes in notation: The observation consists of $6-9$ dimensional texture feature vectors $\mathcal{F}^t = \{\tilde{\mathbf{f}}^t_s | s \in \mathcal{S}^t\}$. The energy of higher order cliques is $\xi \sum_{\{s,r\}\in\mathcal{C}} \delta(\phi_s, \phi_r) + \gamma^t \sum_{s\in\mathcal{S}^t} V^t(\phi_s, \eta^t_s)$, where ξ (resp. γ^t) has the same role as β (resp. γ^c) in the color layer. Furthermore, ϕ_s denotes the label assigned to a site s.

2.2. Combined Layer

The combined layer only uses the texture and color features indirectly, through inter-layer cliques. A label consists of a pair of color and texture labels such that $\eta = \langle \eta^c, \eta^t \rangle$, where $\eta^c \in \Lambda^c$ and $\eta^t \in \Lambda^t$. The set of labels is denoted by $\Lambda^x = \Lambda^c \times \Lambda^t$ and the number of classes $L^x = L^c L^t$. Obviously,

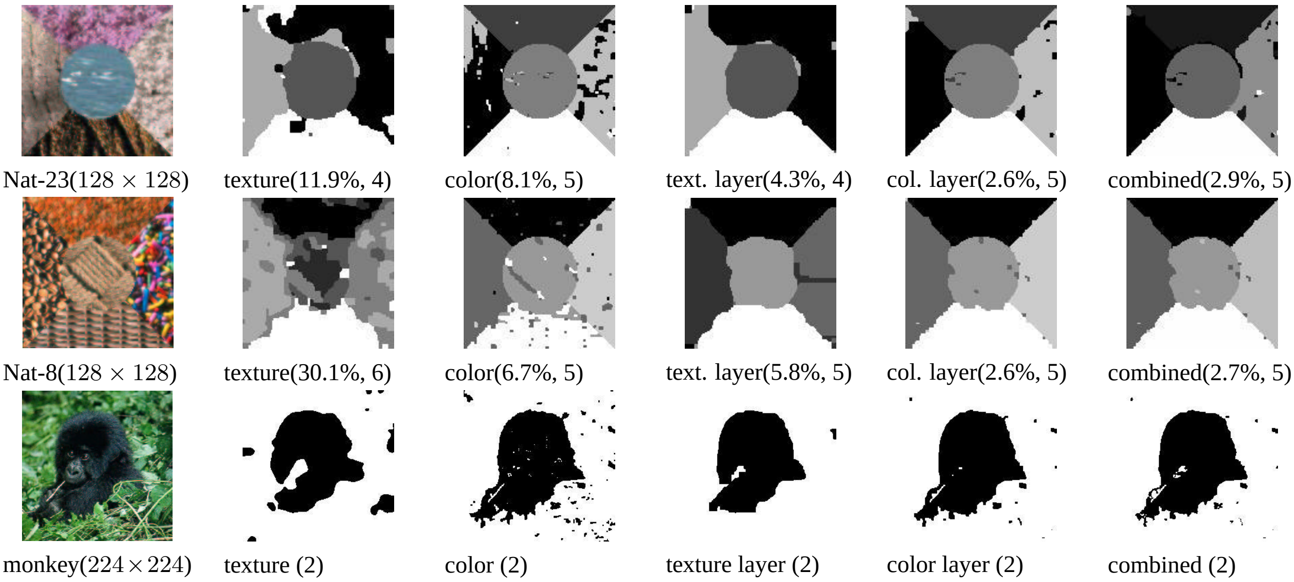

Fig. 2. Misclassification rate and estimated number of classes of color only, texture only, and combined models

not all of these labels are valid for a given image. Therefore the combined layer model also estimates the number of classes and chose those pairs of texture and color labels which are actually present in a given image. The energy function $U(\eta) = \sum_{s\in\mathcal{S}^x} \left(V_s(\eta_s) + \rho^c V^c(\psi_s, \eta_s^c) + \rho^t V^t(\phi_s, \eta_s^t)\right) + \alpha \sum_{\{s,r\}\in\mathcal{C}} \delta(\eta_s, \eta_r)$ where $V_s(\eta_s)$ denotes singleton energies, $V^c(\psi_s, \eta_s^c)$ (resp. $V^t(\phi_s, \eta_s^t)$ denotes inter-layer clique potentials. The last term corresponds to second order intra-layer cliques which ensures homogeneity of the combined layer. α has the same role as β in the color layer model and $\delta(\eta_s, \eta_r) = -1$ if $\eta_s = \eta_r$, 0 if $\eta_s \neq \eta_r$ and 1 if $\eta_s^c = \eta_r^c$ *and* $\eta_s^t \neq \eta_r^t$ or $\eta_s^c \neq \eta_r^c$ *and* $\eta_s^t = \eta_r^t$. The idea is that region boundaries present at both color and texture layers are preferred over edges that are found only at one of the feature layers. Inter-layer interactions are as follows:

$$V^c(\psi_s, \eta_s^c) = \sum_{\{s,r\}\in\mathcal{C}_6} W_r D^c(\psi_r, \eta_s^c)$$

where $D^c(\psi_r, \eta_s^c) = \mid \mathcal{G}^c(\tilde{\mathbf{f}}_r^c, \psi_r) - \mathcal{G}^c(\tilde{\mathbf{f}}_s^c, \eta_s^c) \mid$ (see Eq. (2)). $V^t(\phi_s, \eta_s^t)$ and $D^t(\phi_r, \eta_s^t)$ are defined in a similar way using texture features. At any site s, we have a clique between two layers containing 6 sites (the set of these inter-layer cliques is denoted by $\mathcal{C}_6$), which implements 5 inter-layer interactions: Site s interacts with the corresponding site on the other layer as well as with the 4 neighboring sites two steps away (see Fig. 1). W_r is the weight of the clique $\{s, r\} \in \mathcal{C}_6$. We assign higher weight (0.6) to the corresponding site whereas smaller weights $(0.1$ each) to the other 4 neighboring sites. The latter 4 sites help to ensure homogeneity on the combined layer. Note that D^c and D^t corresponds to the difference of the first order potentials at the corresponding feature layer. Clearly, the difference is 0 if and only if both the feature layer and the combined layer has the same label. If the labels are different then it is proportional to the energy difference between the two labels. ρ^c (resp. ρ^t) controls the influence of the inter-layer cliques on the combined layer. A higher value will increase the importance of the information coming from the corresponding feature layer. Note that we have a similar weight (γ^c, γ^t) at the feature layers. The difference of these weights balances the influence of the feature layers to the combined layer vs. combined layer to the feature layers. Therefore, depending on the value of ρ^t (resp. ρ^c), we can increase ($\gamma > \rho$) or decrease ($\gamma < \rho$) the influence of a feature layer to the combined layer without changing the influence of the combined layer to a feature layer. We found this an important issue in the case of the texture layer.

2.3. Parameter Estimation

To control the number of classes, we add the following term to the energy function of each layer: $R((10N_{\omega_s})^{-3} + \mathcal{P}(L))$. $(10N_{\omega_s})^{-3}$ penalizes small classes (N_{ω_s} is the percentage of the sites assigned to class ω_s), while $\mathcal{P}(L)$ includes some prior knowledge about the number of classes. Currently this is expressed by a log Gaussian term (similar to the one in Eq. (2)) with mean value $\hat{L}$ (basically an initial guess) and variance σ (confidence in the initial guess). R is simply a weight of this term, we set it to 0.5 in our tests. We also have to estimate the Gaussian parameters of each class. This is done using an adaptive segmentation scheme similar to [10]: every 10 iteration, we simply recompute the mean values and covariance matrices based on the current labeling. Initial estimates at feature layers are obtained via mean shift clustering [11].

3. EXPERIMENTS

The proposed algorithm has been tested on a variety of synthetic and real images. The computing time (including both parameter estimation and segmentation) was 3-10 minutes (depending on size and number of classes) on a Pentium III 933. We also compare the results to unsupervised texture only and color only segmentation (basically a monogrid model similar to the one defined for the feature layers but without inter-layer cliques). The number of classes, mean vectors and covariance matrices have been estimated during the segmentation process. Hyper-parameters have been trained on a small subset of images: $\alpha = 1.0$, $\beta = \xi = 10.0$, $\gamma^c = \gamma^t = 1.5$, $\rho^c = 0.5$, and $\rho^t = -0.3$. These values have been found to provide satisfactory results on *all* test images. The values of β and ξ are not crucial, basically any value between 2 and 15 provides good segmentations. γ and ρ values need slightly higher accuracy. Note that by setting $\rho^t < 0$ and $\rho^c > 0$, we decrease the influence of the texture layer and increase the influence of the color layer on the combined layer. This is necessary because texture features (due to filtering and blurring) have weaker spatial localization. Hence, we give a higher weight to the color layer so that edges will be localized correctly while region homogeneity (where color layer is slightly weaker, especially in textured regions) is still maintained. Fig. 2 shows some segmentation results together with the measured misclassification rate. Clearly, the multi-layer model provides significantly better results and a more accurate estimate of the number of classes compared to color only and texture only segmentations. *Nat-23* shows an image with 4 different textures and 5 different colors. We can see, that our method provides accurate segmentations and it is also able to detect the right number of classes on all layers. Note that the combined layer produces slightly higher misclassification rates ($\approx 0.3\%$) than the color layer. This is due to sharper boundaries on the color layer (texture has weaker spatial resolution and the combined layer is directly influenced by the texture layer). We have also compared our results to those reported in [2] and found them equally good. One example is the *monkey* image but more results are available on our website (`www.cs.ust.hk/~kato/research/icip2003/`).

4. CONCLUSION

We have proposed a new unsupervised multi-layer MRF segmentation model which successfully combines color and texture features and estimates associated model parameters. However, the model is not restricted to these features, it can be applied to multi-cue segmentation in general.

5. REFERENCES

[1] D. K. Panjwani and G. Healey, "Markov random field models for unsupervised segmentation of textured color images," *IEEE Trans. on PAMI*, vol. 17, no. 10, pp. 939–954, Oct. 1995.

[2] M. Mirmehdi and M. Petrou, "Segmentation of color textures," *IEEE Trans. on PAMI*, vol. 22, no. 2, pp. 142–159, Feb. 2000.

[3] W. Y. Ma and B. S. Manjunath, "Edge flow: A framework of boundary detection and image segmentation," in *Proc. CVPR*, San Juan, Puerto Rico, June 1997, pp. 744–749.

[4] F. Heitz, P. Perez, and P. Bouthemy, "Multiscale Minimization of Global Energy Functions in Some Visual Recovery Problems," *CVGIP:IU*, vol. 59, no. 1, pp. 125–134, 1994.

[5] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. on PAMI*, vol. 6, pp. 721–741, 1984.

[6] Zoltan Kato, Josiane Zerubia, and Mark Berthod, "Satellite image classification using a modified Metropolis dynamics," in *Proc. ICASSP*, San-Francisco, California, USA, Mar. 1992.

[7] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1186, 1991.

[8] J. Mao and A. K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Pattern Recognition*, vol. 25, no. 2, pp. 173–188, 1992.

[9] T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," *IEEE Trans. on PAMI*, vol. 21, no. 4, pp. 291–310, Apr. 1999.

[10] C. S. Won and H. Derin, "Unsupervised segmnetation of noisy and textured images using Markov random fields," *CVGIP: GMIP Image Processing*, vol. 54, no. 4, pp. 208–328, July 1992.

[11] D. Comaniciu and P. Meer, "Mean shift: A robust approach towards feature space analysis," *IEEE Trans. on PAMI*, 2001.