[17] Y. Xia, C. Sun, and W. X. Zheng, "Discrete-time neural network for fast solving large linear estimation problems and its application to image restoration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 812–820, May 2012.
[18] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley, 1987.
[19] K. C. Tan, H. J. Tang, and Z. Yi, "Global exponential stability of discrete-time neural networks for constrained quadratic optimization," *Neurocomputing*, vol. 56, pp. 399–406, Jan. 2004.
[20] A. Greenbaum and G. H. Rodrigue, "Optimal preconditioners of a given sparsity pattern," *BIT Numer. Math.*, vol. 29, no. 4, pp. 610–634, Dec. 1989.

Incorporating Mean Template Into Finite Mixture Model for Image Segmentation

Hui Zhang, *Member, IEEE*, Q. M. Jonathan Wu, *Senior Member, IEEE*, and Thanh Minh Nguyen

***Abstract*— The well-known finite mixture model (FMM) has been regarded as a useful tool for image segmentation application. However, the pixels in FMM are considered independent of each other and the spatial relationship between neighboring pixels is not taken into account. These limitations make the FMM more sensitive to noise. In this brief, we propose a simple and effective method to make the traditional FMM more robust to noise with the help of a mean template. FMM can be considered a linear combination of prior and conditional probability from the expression of its mathematical formula. We calculate these probabilities with two mean templates: a weighted arithmetic mean template and a weighted geometric mean template. Thus, in our model, the prior probability (or conditional probability) of an image pixel is influenced by the probabilities of pixels in its immediate neighborhood to incorporate the local spatial and intensity information for eliminating the noise. Finally, our algorithm is general enough and can be extended to any other FMM-based models to achieve super performance. Experimental results demonstrate the improved robustness and effectiveness of our approach.**

***Index Terms*— Expectation maximization (EM) algorithm, finite mixture model, image segmentation, mean template, spatial constraints.**

I. Introduction

Image segmentation is one of the most important and difficult problems in many applications, such as robot vision, object recognition, and medical image processing. Although different methodologies [1]–[5] have been proposed for image segmentation, it remains a challenge due to overlapping intensities, low contrast of images, and noise perturbation. One of the most widely used clustering models for image segmentation is the well-known finite mixture model (FMM) [1]. The latest research of FMM for classification and image representation is discussed in [6]–[8]. A good survey on the history of the mixture expert can be found in [9]. The Gaussian mixture model (GMM) is most commonly selected as a particular case of the FMM by assuming the conditional probability as a Gaussian distribution [10]–[12]. The GMM is a flexible and powerful statistical modeling tool for multivariate data. The main advantage of the standard GMM is that it is easy to implement and the small number of parameters can be efficiently estimated by adopting the expectation maximization (EM) algorithm. However, as a histogram-based model, the GMM assumes that each pixel in an image is independent of its neighbors and does not take into account spatial dependencies. Moreover, it does not use the prior knowledge that adjacent pixels most likely belong to the same cluster. Thus, the performance of the GMM is sensitive to noise and image contrast levels.

To overcome this limitation, a wide variety of approaches have been proposed to incorporate spatial information into the image. A common approach is the use of a Markov random field (MRF) [13]. Such a method aims to impose spatial smoothness constraints on the image pixel labels. However, most reported methods are limited to using an MRF as a general prior in an FMM-based approach. Recently, a special case of the MRF model—the hidden MRF (HMRF) model—was proposed [14], [15]. The state sequence of HMRF cannot be observed directly, but can be observed indirectly through a field of observations. In the HMRF model, the spatial information in an image is encoded through the contextual constraints of neighboring pixels, which are characterized by conditional MRF distributions. Parameter estimation in HMRF models usually relies on maximum likelihood (ML) or Bayesian methods. Besag [16] introduced the idea of the pseudo likelihood approximation when ML estimation is intractable. Based on this well-known approximation, various HMRF model estimation approaches have been proposed [17], [18]. To deal with different HMRF models, the EM algorithm is most commonly used for parameter learning.

The main drawback of HMRF models is that they are computationally expensive to implement, and require the additional parameter β to control the degree of image smoothness. The chosen parameter has to be both large enough to tolerate the noise, and small enough to preserve image sharpness and details. Thus, the parameter is noise dependent to some degree and selected generally based on experience. In this brief, we propose a simple, fast, and effective algorithm to make the traditional FMM more robust to noise, by incorporating local spatial information and pixel intensity value information with the help of a mean template. The conditional probability (or prior probability) of an image pixel is influenced by the probabilities of pixels in its immediate neighborhood. Moreover, our model is general enough and

Manuscript received May 24, 2012; revised November 2, 2012; accepted November 4, 2012. Date of publication December 20, 2012; date of current version January 11, 2013. This work was supported in part by the Canada Chair Research Program and the Natural Sciences and Engineering Research Council of Canada, and the National Natural Science Foundation of China under Grant 61105007.

H. Zhang is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada, and also with the School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China (e-mail: nrzh@uwindsor.ca).

Q. M. J. Wu and T. M. Nguyen are with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: jwu@uwindsor.ca; nguyen1j@uwindsor.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2012.2228227

the FMM can be considered a special case of our model. For spatial information, we add weighting for distant pixels in order to distinguish among the contributions of different pixels, as the weighted parameters decrease with increasing distance. Different from the HMRF model, our model is fully free of the empirically adjusted parameter β. Furthermore, our approach is based on the FMM—it is simple and fast and can be easily implemented. Parameters are iteratively estimated by the classic EM algorithm. The proposed approach is applied to segment real images and multidimensional images. The performance of our proposed approach, compared with state-of-the-art technologies, demonstrates its improved robustness and effectiveness.

The remainder of this brief is organized as follows. In Section II, we briefly introduce the mathematical background of the FMM and HMRF methods for image segmentation. Moreover, we introduce the problems caused by FMM for noised image application in detail. In Section III, we incorporate the mean template into FMM and use the EM algorithm for parameter learning. In Section IV, the calculation of prior probability is introduced with the help of a mean template. The experimental results of the proposed approach are given in Section V. Finally, some concluding remarks are provided.

II. Mathematical Background

A. FMM

Let us first consider two letters: $Q = \{1, 2, \ldots, K\}$ and $L = \{1, 2, \ldots, D\}$. Let S be a finite index set, $S = \{1, 2, \ldots, N\}$. We shall refer to set S as the set of sites or locations. Let X and Y be two random fields, and their state space and Ψ are indexed by the supposed set of sites S (every site $i \in S$), given by

$$X_i = \{x_i : x_i \in Q\}, \text{ and } Y_i = \{y_i : y_i \in L\}.$$

Their product space $\mathcal{X} = \prod_i x_i$ and $\mathcal{Y} = \prod_i y_i$ shall be denoted as the space of the configurations of the state values of the considered site set, $\boldsymbol{x} = (x_i)$ and $\boldsymbol{y} = (y_i)$.

For image segmentation application, an image consisting of N pixels is segmented into K classes. y_i denotes the observation (intensity value) at the i th pixel of an image and x_i denotes the corresponding class label of the i th pixel.

For every $j \in Q$ and $i \in S$, the probability $p(x_i = j) = \pi_j$ is the prior distribution of the pixel y_i, belonging to the class x_i, which satisfies the constraints

$$0 \le \pi_j \le 1 \text{ and } \sum_{j=1}^{K} \pi_j = 1. \tag{1}$$

$x_i = j$, y_i follows a conditional probability distribution $p(y_i|\theta_j)$, in which θ_j is the set of parameters. Specific to the GMM, the conditional probability $p(y_i|\theta_j)$ is selected as a Gaussian distribution. Under the independent assumption, the FMM can be calculated as

$$p(y_i|\pi, \theta) = \sum_{j=1}^{K} \pi_j p\left(y_i|\theta_j\right). \tag{2}$$

Although FMM is widely used for its simplicity and effectiveness as a model [1], it only describes the data statistically; no spatial information about the data is utilized. In other words, it does not take into account the spatial correlation between the neighboring pixels in the decision process. Images with the same intensity distribution may have entirely different structural properties. In order to overcome the problems with the FMM, and to reduce the sensitivity of the segmentation result with respect to noise, several researchers have suggested modifications to incorporate local spatial information between the neighboring pixels.

B. HMRF Model

MRF accounts for contextual constraints by using spatial information with conditional MRF distributions. In an MRF, the sites in S are related to one another via a neighborhood system, which is defined as $\mathcal{N} = \{\mathcal{N}_i, i \in S\}$, where N_i is the set of sites neighboring $I- \; i \notin \mathcal{N}_i$ and $i \in \mathcal{N}_j$if and only if $j \in \mathcal{N}_i \; \forall i, j \in S$. A random field X is said to be an MRF on S with respect to a neighborhood system N if and only if

$$p(x) > 0 \qquad \forall x \in \mathcal{X} \text{ and } p\left(x_i|x_{S-\{i\}}\right) = p\left(x_i|x_{\mathcal{N}_i}\right).$$

According to the Hammersley–Clifford theorem [19], a given random field is an MRF if and only if its probability distribution is a Gibbs distribution; thus

$$p(x) = Z^{-1} \exp\left(-U(x)\right) \tag{3}$$

where Z is a normalizing constant called the partition function, and $U(\boldsymbol{x})$ is an energy function of the form

$$U(x) = \beta \sum_{c \in C} V_c(x) \tag{4}$$

where C is the all possible cliques and $V_c(\boldsymbol{x})$ stands for the clique potential associated with the clique c. A clique c is defined as a subset of sites in S in which all pairs of distinct sites are neighbors, except for single-site cliques. The parameter β is used to balance robustness to noise and effectiveness of preserving image sharpness and detail.

HMRF is a special case of hidden Markov models, which are defined as stochastic processes with an unobservable state field X and an observable (or emitted) random field Y. The probability distribution $p(\boldsymbol{x})$ satisfies (3). Moreover, for any $x \in \mathcal{X}$, the random variables y_i satisfy the conditional independent assumption

$$p(y|x) = \prod_{i=1}^{N} p(y_i|x_i) \tag{5}$$

which provides a convenient approximation of the posterior field $p(\boldsymbol{x}|\boldsymbol{y})$, still guaranteeing its Markovianity.

The computation of the term Z in (3) involves all possible realizations of $\boldsymbol{x}$ of the HMRF, which is rarely feasible in terms of computational requirements. Besag [16] introduced the idea of pseudo likelihood approximation of the Markov field prior to solve this problem. Under this approximation, the prior of the Markov field is

$$p(x) = \prod_{i=1}^{N} p\left(x_i|x_{\mathcal{N}_i}\right). \tag{6}$$

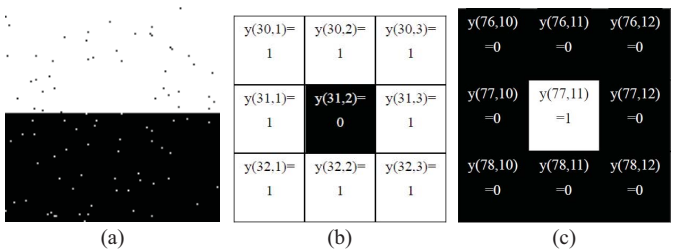

Fig. 1. (a) Original image. (b) Subimage1. (c) Subimage2. The numbers in parentheses are the coordinates of the image; 0 and 1 are the binary image intensity values.

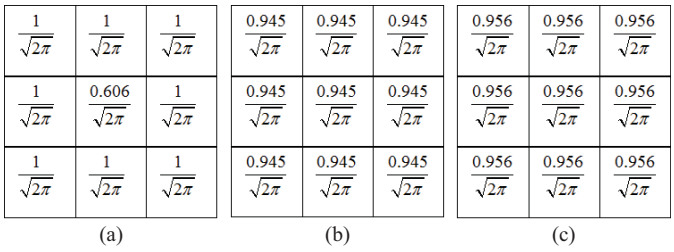

Fig. 2. CPV of subimage1 in Fig. 1(b). (a) CPV in traditional FMM. (b) CPV calculated by geometric template. (c) CPV calculated by arithmetic template.

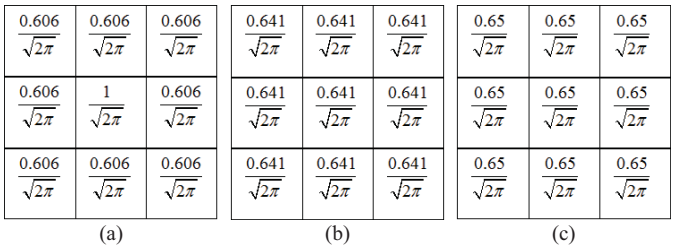

Fig. 3. CPV of subimage2 in Fig. 1(c). (a) CPV in traditional FMM. (b) CPV calculated by geometric template. (c) CPV calculated by arithmetic template.

Relevant studies [1], [2], [15], and [7] show that optimization of the Markov field prior under the pseudo likelihood approximation (6) offers good estimates of the HMRF model parameters.

C. Problem Description

Let us take a binary image as an example. The upper part of the image is white with the intensity value $= 1$ and the lower part is black with the intensity value $= 0$. The image is corrupted by noise, as shown in Fig. 1(a). Two 3×3 windows extracted from the upper and lower parts of the image are shown in Fig. 1(b) and (c), respectively. For the binary image, let pixels with intensity value $= 1$ belong to class A and those with intensity value $= 0$ to class B. It can be easily seen that Fig. 1(b) belongs to class A and Fig. 1(c) belongs to class B; however, the central pixels of the windows in Fig. 1(b) and (c) are corrupted by noise and may cause misclassification. Considering the FMM shown in (2), the possible solution for this problem has two requirements.

1) π_j should be changed to π_{ij} by incorporating the spatial information for pixel i to prior probability π_j. For Fig. 1(b) and (c), the prior probability π_{ij} of the central pixel of the window (noise) should be affected by the prior probability π_{mk} of the other pixels in the window (signal), where $m = i \pm 1$ and $k = j \pm 1$. All pixels in the window should have the same prior probability after correct calculation. As long as the signal strength is greater than the noise strength, the correct prior probability can always be calculated. This is the reason why we use a mean template to calculate "average" prior probability for eliminating the noise effect.
2) For the same component j and intensity value y, traditional FMM satisfies the same conditional probability $p(y_i|\theta_j)$. This is often true, but not always true. The intensity value of the central pixel in Fig. 1(b) is the same as the intensity value of the other pixels around the central pixel in Fig. 1(c); however, these two kinds of pixels should belong to different classes, and the calculation of component conditional probability $p(y_i|\theta_j)$ in traditional FMM cannot distinguish among them. For this purpose, we incorporate the mean template into conditional probability calculation to make the probability of the ith pixel to be influenced by the probability in its immediate neighborhood. In this case, two pixels with the same intensity value y can have different conditional probability values (CPVs) for the same classes.

For deep investigation, in Fig. 2(a), we show the CPV of class A [corresponding to Fig. 1(b)] in FMM. The CPV calculated by geometric mean template and arithmetic mean template are shown in Fig. 2(b) and (c), respectively. Compared to the FMM in Fig. 2(a), our model erases the noise on the central pixels of the windows by applying the mean templates to the CPV calculation. We then show the CPV of class A [corresponding to Fig. 1(c)] for different methods in Fig. 3(a) through (c). Comparing Fig. 2(a) with Fig. 3(a), in FMM, it can be seen that the CPV of the central pixel in Fig. 1(c) is the same as the CPV of the other pixels around the central pixel in Fig. 1(b), equal to $1/\sqrt{2\pi}$. In fact, these two kinds of pixels with the same intensity value 1 should have a different CPV due to the effect of the noise. From Fig. 2(b) and (c) to Fig. 3(b) and (c), we can see our algorithm distinguishes them well with the help of the mean template. In short, the traditional FMM is calculated by the summation of two items: prior probability and component conditional probability. In this brief, we resort to a mean template for calculating these two probabilities to make the standard FMM more robust to noise.

III. Mean Template for Conditional Probability

Let y_i, with dimension d, $i = (1, 2, \ldots, N)$, denote the intensity value at the ith pixel of an image and $j\,(j = 1, 2, \ldots, K)$ denote the corresponding class label of the ith pixel. Considering the spatial information on the prior probability, (2) can be modified as

$$p\left(y_i|\boldsymbol{\pi}, \boldsymbol{\theta}\right) = \sum_{j=1}^{K} \pi_{ij}\, p\left(y_i|\theta_j\right). \tag{7}$$

As a special case of FMM, the conditional probability $p\left(y_m|\theta_j\right)$ with the Gaussian distribution is called the GMM.

The prior probability π_{ij} in (7) represents the prior distribution of pixel y_i belonging to class j, which satisfies the constraint

$$0 \le \pi_{ij} \le 1 \text{ and } \sum_{j=1}^{K} \pi_{ij} = 1. \tag{8}$$

We then apply a mean template for the calculation of condition probability.

A. Weighted Geometric Mean Template

We first apply a weighted geometric mean template for calculating the conditional probability of the ith pixel by its neighborhood probabilities. Thus, (7) can be modified as

$$p(y_i|\pi,\theta) = \sum_{j=1}^{K} \pi_{ij} \prod_{m\in\mathcal{N}_i} p\left(y_m|\theta_j\right)^{\frac{w_m}{R_i}} \tag{9}$$

where N_i is the neighborhood of the ith pixel; including the ith pixel, this is called the conditional probability window (CPW). R_i is the normalized factor, defined as

$$R_i = \sum_{m\in\mathcal{N}_i} w_m. \tag{10}$$

It is noted that our model degrades to the standard Gaussian mixture model when CPW $= 1\times 1$.

We then apply the EM algorithm for parameter learning in our model. First, according to [2], the complete-data log likelihood function is calculated as

$$Q = \sum_i \sum_j z_{ij} \left[\log \pi_{ij} + \sum_{m\in\mathcal{N}_i} \frac{w_m}{R_i} \log p\left(y_m|\theta_j\right)\right]. \tag{11}$$

In E-step, the posterior probability can be calculated as

$$z_{ij}^{(k+1)} = \frac{\pi_{ij}^{(k)} \prod_{m\in\mathcal{N}_i} p\left(y_m|\theta_j^{(k)}\right)^{\frac{w_m}{R_i}}}{\sum_{h=1}^{K} \pi_{ih}^{(k)} \prod_{m\in\mathcal{N}_i} p\left(y_m|\theta_h^{(k)}\right)^{\frac{w_m}{R_i}}}. \tag{12}$$

The M-step evaluates the mean and covariance as follows:

$$\mu_j^{(k+1)} = \frac{\sum_{i=1}^{N} \sum_{m\in\mathcal{N}_i} z_{ij}^{(k)} \frac{w_m}{R_i} y_m}{\sum_{i=1}^{N} z_{ij}^{(k)}}. \tag{13}$$

$$\sum_j^{(k+1)} = \frac{\sum_{i=1}^{N} \sum_{m\in\mathcal{N}_i} z_{ij}^{(k)} \frac{w_m}{R_i} \left(y_m - \mu_j^{(k)}\right)\left(y_m - \mu_j^{(k)}\right)^T}{\sum_{i=1}^{N} z_{ij}^{(k)}}. \tag{14}$$

B. Weighted Arithmetic Mean Template

In this section, we incorporate a weighted arithmetic mean template into the calculation of conditional probability. (7) can be modified as

$$p(y_i|\pi,\theta) = \sum_{j=1}^{K} \pi_{ij} \sum_{m\in\mathcal{N}_i} \frac{w_m}{R_i} p\left(y_m|\theta_j\right). \tag{15}$$

It is noticed that the mark of N_i and R_i is the same as the definition in the previous section.

A simple choice of weighted parameter w_m in (9) and (15) is that $w_m = 1$ for all mth pixels, and R_i equals the number of pixels in the CPW. However, to incorporate the spatial information and pixel intensity value information, the strength of w_m should decrease as the distance between pixel m and i increases. For this reason, we define w_m as the function of d_{mi}, which is the spatial Euclidean distance between pixels m and i

$$w_m = \frac{1}{\left(2\pi\delta^2\right)^{1/2}} \exp\left(-\frac{d_{mi}^2}{2\delta^2}\right) \tag{16}$$

$$\delta = \frac{\text{size of CPW} - 1}{4}. \tag{17}$$

We then apply the EM algorithm for parameter learning in our model. The complete-data log likelihood can be written as

$$\begin{aligned} Q &= \sum_i \sum_j z_{ij} \left[\log \pi_j + \log\left[\sum_{m\in\mathcal{N}_i} \frac{w_m}{R_i} p\left(y_m|\theta_j\right)\right]\right] \\ &= \sum_i \sum_j z_{ij} \left[\log \pi_j + L\right]. \end{aligned} \tag{18}$$

It seems that quantity L cannot be directly calculated. It is noticed that w_m/R_i always satisfies the condition $w_m/R_i \ge 0$ and $\sum_{m\in\mathcal{N}_i} w_m/R_i = 1$. We can now apply the Jensen's inequality [20], which states that, given a set of numbers $\lambda_i \ge 0$ and $\sum_i \lambda_i = 1$, we have $\log\left(\sum_i \lambda_i x_i\right) \ge \sum_i \lambda_i \log(x_i)$. Thus, quantity L in (18) can be modified, and the new log likelihood function is calculated as

$$Q = \sum_i \sum_j z_{ij} \left[\log \pi_{ij} + \sum_{m\in\mathcal{N}_i} \frac{w_m}{R_i} \log p\left(y_m|\theta_j\right)\right]. \tag{19}$$

In E-step, the posterior probability can be calculated as

$$z_{ij}^{(k+1)} = \frac{\pi_{ij}^{(k)} \sum_{m\in\mathcal{N}_i} \frac{w_m}{R_i} p\left(y_m|\theta_j^{(k)}\right)}{\sum_{h=1}^{K} \pi_{ih}^{(k)} \sum_{m\in\mathcal{N}_i} \frac{w_m}{R_i} p\left(y_m|\theta_h^{(k)}\right)}. \tag{20}$$

The mean and covariance calculated in M-step are the same as (13) and (14); thus this step is omitted here. It is noticed that the estimated parameters based on an arithmetic template for conditional probability calculation are the same as those based on a geometric template except for the posterior probability evaluation.

IV. Prior Probability Estimation

For incorporating mean template into prior (posterior) probability, let us first recall the calculation of prior probability in FMM

$$\pi_j = \frac{\sum_{i=1}^{N} z_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}}. \tag{21}$$

Algorithm 1 EM Algorithm for Our Model

Step 1. Initialize the algorithm with the k-means method to obtain initial values and set CPW and PPW size.

Step 2. In E-step, compute the prior probability $\pi_{ij}^{(k)}$ using (22) for GCGP and ACGP, and using (23) for GCAP and ACAP. Then calculate the posterior probability $z_{ij}^{(k)}$ using (12) for GCGP and GCAP, and using (20) for ACGP and ACAP.

Step 3. According to the M-step, compute the quantities $\mu_j^{(k+1)}$ and $\Sigma_j^{(k+1)}$ by (13) and (14) for all methods.

Step 4. Terminate the iterations if the EM algorithm converges; otherwise, increase the iteration ($k = k+1$) and repeat steps 2 through 4.

We then apply weighted geometric and arithmetic mean templates, respectively, for calculating the prior probability, which yields

$$\pi_{ij} = \frac{\prod_{m\in\partial_i} \left(z_{mj}\right)^{\frac{w_m}{R_i}}}{\sum_{k=1}^{K} \prod_{m\in\partial_i} \left(z_{mj}\right)^{\frac{w_m}{R_i}}} \tag{22}$$

$$\pi_{ij} = \frac{\sum_{m\in\partial_i} w_m z_{mj}}{\sum_{k=1}^{K} \sum_{m\in\partial_i} w_m z_{mk}} \tag{23}$$

where $\partial_i = \mathcal{N}_i - \{i\}$ is the neighborhood of the ith pixel, called the prior probability window (PPW). One possible choice of weighted parameter is $w_m = 1/\left(1 + d_{mi}^2\right)$. It is noticed that (22) and (23) are equal to (21) when we set PPW $= 1 \times 1$ (note the difference between π_j in GMM and π_{ij} in our model). Thus, FMM can be considered as a special of our model.

We understand that our algorithm can generate four models by applying geometric and average templates on conditional probability and prior probability, respectively. We call them GCGP, GCAP, ACAP, and ACGP, for short. For example, GCAP represents applying a geometric template to conditional probability calculation and an average template to prior probability calculation. For a deep understanding of our algorithm, we summarize the computation process of our algorithm in Algorithm 1.

V. Experimental Results and Discussion

In this section, we experimentally evaluate our algorithm in a set of real images and a multidimensional noised image. We also evaluate SVFMM [21], FLICM [3], and HMRF-FCM [12] for comparison. The source codes for the FLICM and HMRF-FCM algorithms can be downloaded from the authors' websites [22], [23]. Our experiments have been developed in MATLAB R2009b, and are executed on an Intel Pentium Dual-Core 2.2 GHZ CPU, 2G RAM.

A. Real Images

In this experiment, we evaluate the performance of the proposed GCGP, GCAP, ACAP, and ACGP based on a subset of the *Berkeley* image dataset [24], which is comprised of a set of real-world color images along with segmentation maps provided by different individuals. We employ the normalized probabilistic rand (PR) index [25] to evaluate the performance of the proposed method, with the multiple ground truths available for each image within the dataset. It has been shown that the PR index possesses the desirable property of being robust to segmentation maps that result from splitting or merging segments of the ground truth [26]. The PR index takes values between 0 and 1, with values closer to 0 (indicating an inferior segmentation result) and values closer to 1 (indicating a better result).

Fig. 4 shows the original Berkeley images used for the image segmentation experiment. These images without noise, with Gaussian noise (G), and with salt-and-pepper (SP) noise are segmented by GCGP, GCAP, ACAP, and ACGP, which

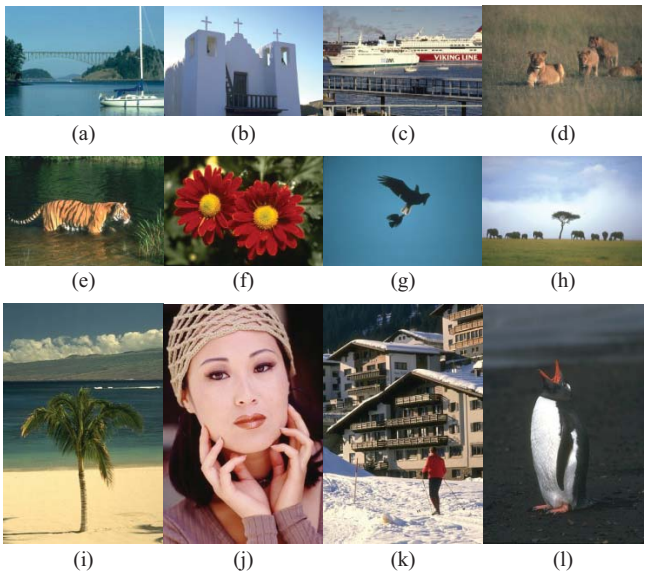

Fig. 4. Original image from the *Berkeley* image segmentation dataset.

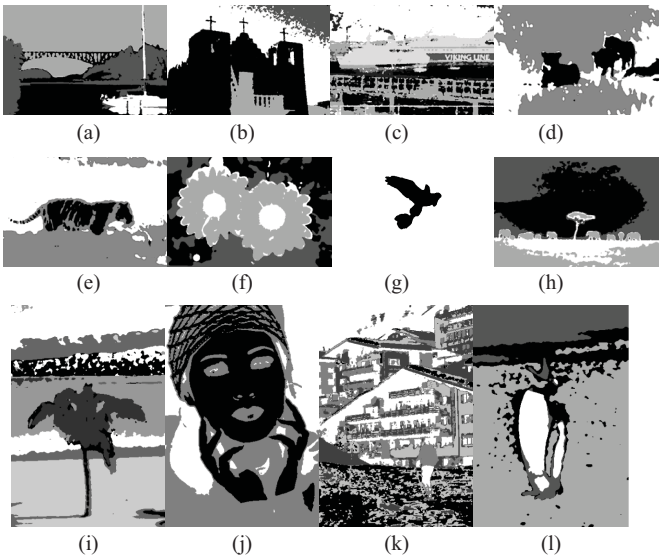

Fig. 5. Image segmentation results by our algorithm. (a)–(c) by ACAP. (d)–(f) by ACGP. (g)–(i) by GCGP. (j)–(l) by GCAP.

TABLE I
COMPARISON OF DIFFERENT METHODS FOR *Berkeley* IMAGE DATASET, PR INDEX

Image #	Class	SVFMM	FLICM	HMRF-FCM	ACAP	ACGP	GCGP	GCAP
108073	4	0.5966	0.5824	0.5855	0.6063	0.6113	0.6234	0.6266
124084	4	0.6067	0.6	0.7203	0.7375	0.7350	0.7307	0.7311
135069	2	0.9835	0.9834	0.9873	0.9855	0.9839	0.9728	0.9744
302003	3	0.7170	0.7172	0.7169	0.7179	0.7107	0.6964	0.7018
105053	3	0.5494	0.51	0.5546	0.6216	0.6385	0.6308	0.6095
22090	4	0.7752	0.7675	0.7777	0.7972	0.8005	0.8028	0.8013
46076+G, 0.01	6	0.7693	0.8151	0.8606	0.8068	0.8623	0.8753	0.8739
61086+G, 0.01	5	0.7143	0.6873	0.7299	0.7304	0.7353	0.7233	0.7329
106025+G, 0.01	4	0.6347	0.8116	0.7988	0.8421	0.8428	0.8463	0.8207
253036+G, 0.02	4	0.6442	0.6838	0.6690	0.7309	0.7119	0.7106	0.7125
78019+G, 0.02	7	0.7790	0.6580	0.8308	0.8500	0.8510	0.8415	0.8571
24063+G, 0.02	4	0.7204	0.8277	0.7672	0.7877	0.7833	0.8110	0.8133
108073+SP, 0.01	4	0.5975	0.6334	0.6311	0.6406	0.6749	0.7054	0.6664
302003+SP, 0.01	3	0.7148	0.7170	0.7174	0.7170	0.7194	0.6943	0.6576
105053+SP, 0.01	3	0.5521	0.5468	0.5643	0.6519	0.6610	0.6536	0.5198
22090+SP, 0.01	4	0.7731	0.7529	0.7981	0.7781	0.8004	0.7332	0.7579
Mean of Nonnoise Images		**0.7047**	**0.6934**	**0.7237**	**0.7443**	**0.7467**	**0.7428**	**0.7408**
Mean of Gaussian Noised Images		**0.7103**	**0.7472**	**0.7761**	**0.7913**	**0.7978**	**0.8013**	**0.8017**
Mean of SP Noised Images		**0.6594**	**0.6625**	**0.6777**	**0.6969**	**0.7139**	**0.6966**	**0.6504**
Mean of Total Images		**0.6955**	**0.7059**	**0.7318**	**0.7501**	**0.7576**	**0.7532**	**0.7411**
Computation Time (seconds)		**32.67**	**60.25**	**91.39**	**15.92**	**18.73**	**19.83**	**17.33**

are illustrated in Fig. 5. The additive Gaussian noise is set mean = 0, covariance = 0.01 or 0.02. The noise density of additive SP noise is set 0.01. For fair comparison, we also evaluate the performance of SVFMM, FLICM, and HMRF-FCM in addition to our methods. The class number K is set by human vision. Table I presents the average PR values for all methods, corresponding to each of the test images in Fig. 4. Compared to other methods, the proposed algorithm yields the best segmentation results with the highest PR values. It is noted that the difference in PR value is 2.3% between the existing method and our method for nonnoise images. However, the difference is 2.56% for Gaussian noised images and 3.62% for SP noised images. It seems that our method obtains much better performance for noised image segmentation. It is noted that the idea of using a template can also be applied to HMRF-FCM, and other FMM-based or retaliated models, in order to improve their performance and obtain better segmentation results.

We also evaluate the computation time for all methods in Table I. It is noted that the computation of our methods is much faster than that of other methods. The computation time of proposed algorithm is nearly 1/2 of SVFMM, 1/3 of FLICM, and 1/4 of HMRF-FCM.

B. Segmentation of Multidimensional Images

In this experiment, we try to segment the multidimensional RGB color image into three classes: the blue sky, the desert nearby, and the ground far away. The original image (481 × 321) shown in Fig. 6(a) is corrupted by heavy Gaussian noise, with mean = 0 and covariance = 0.08.

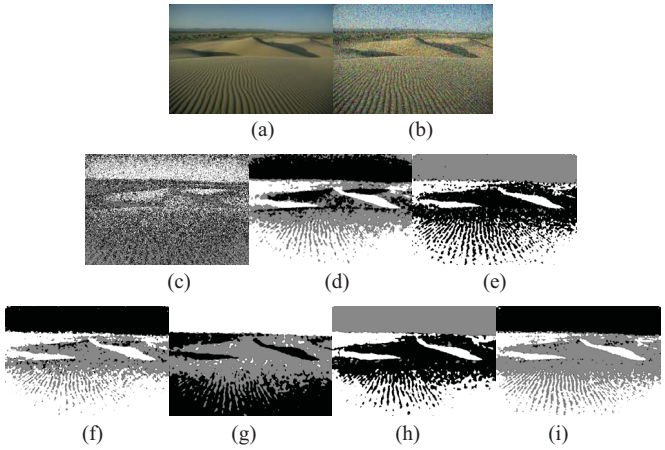

Fig. 6. RGB image segmentation with additive Gaussian noise. (a) Original image. (b) Noised image. (c) SVFMM, PR = 0.5754, t = 520.81s. (d) FLICM, PR = 0.6697, t = 29.78s. (e) HMRF-FCM, PR = 0.7055, t = 180.81s. (f) ACAP, PR = 0.7215, t = 16.01s. (g) ACGP, PR = 0.7156, t = 18.5s. (h) GCGP, PR = 0.7176, t = 19.31s. (i) GCAP, PR = 0.711, t = 16.48s.

The noised image is shown in Fig. 6(b) and the segmentation results of different methods are shown in Fig. 6(c) through (i), respectively. The accuracy of segmentation for SVFMM is quite poor. It can be seen that FLICM and HMRF-FCM misclassify some portions of pixels at the edge region between the sky and the ground, as well as the edge region between the ground and the desert. The accuracy of the segmentation results from our algorithm, as shown in Fig. 6(f) through (i), is better than that of other methods, obtaining the highest PR values.

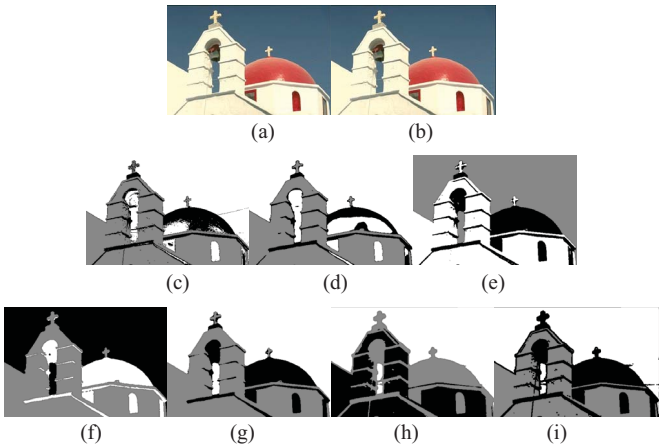

Fig. 7. RGB image segmentation with additive Poisson noise. (a) Original image. (b) Noised image. (c) SVFMM, PR = 0.8276, $t = 22.80$s. (d) FLICM, PR = 0.805, $t = 58.82$s. (e) HMRF-FCM, PR = 0.8614, $t = 68.5$s. (f) ACAP, PR = 0.8676, $t = 15.88$s. (g) ACGP, PR = 0.8677, $t = 17.72$s. (h) GCGP, PR = 0.8331, $t = 19.19$s. (i) GCAP, PR = 0.8337, $t = 19.92$s.

In the last experiment, we segment the multidimensional RGB color image into three classes: the blue sky, the red roof, and the white wall. The original image (481×321) shown in Fig. 7(a) is corrupted by Poisson noise. The noised image is shown in Fig. 7(b) and the segmentation results of SVFMM, FLICM, HMRF-FCM, and proposed algorithms are shown in Fig. 7(c)–(i), respectively. Although SVFMM and FLICM demonstrate good segmentation performances, they still misclassify some portions of pixels at the region of the roof. The accuracy of the segmentation results by HMRF-FCM, ACAP and ACGP, as shown in Fig. 7(e) through (g), is better than other methods, obtaining the highest PR values. It is worth to point out that GCGP and GCAP seem to preserve more image details and contours. This may be caused by the use of weighted geometric mean template.

We also evaluate the computation time for all methods in the previous experiments. The computation time t of the different methods is presented in Figs. 6 and 7. It is noted that the computation of our methods, GCGP, GCAP, ACAP, and ACGP, is much faster than that of other methods. Compared to other methods, our models can be calculated most quickly and achieve the best segmentation results.

VI. Conclusion

In this brief, we proposed a simple and effective algorithm to make the traditional FMM more robust to noise, with consideration of the relationship between the local spatial information and pixel intensity value information by utilizing the mean template. The conditional probability (or prior probability) of an image pixel was replaced by the calculation of the probabilities of pixels in its immediate neighborhood. Moreover, FMM can be considered as a special case of our algorithm when we set the probability window as 1×1. Our method is fully free of the empirically adjusted parameter β, which is used to keep a balance between robustness to noise and image sharpness and details in traditional HMRF method. Finally, our model is simple and easy to implement, and it can be quickly applied to image segmentation.

One limitation of the work, we present in this brief is how to select the proper weighted functions. Another limitation is how to choose the probability window size automatically. It is important to point out that the mean template used in our model can be considered as the mean filter, which is suitable for erasing the Gaussian noise. Other filters can be applied for eliminating the different types of noises. For example, a median filter can be used for eliminating the SP noise, and proper fuzzy filters can be applied for erasing the mixed Gaussian and SP noises.

References

[1] G. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley, 2000.
[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer-Verlag, 2006.
[3] S. Krinidis and V. Chatzis, "A robust fuzzy local information C-means clustering algorithm," *IEEE Trans. Image Process.*, vol. 5, no. 19, pp. 1328–1337, May 2010.
[4] M. Ahmed, S. Yamany, N. Mohamed, A. Farag, and T. Moriarty, "A modified fuzzy C-means algorithm for bias field estimation and segmentation of MRI data," *IEEE Trans. Med. Imag.*, vol. 21, no. 3, pp. 193–199, Mar. 2002.
[5] S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new Kernel-induced distance measure," *IEEE Trans. Syst., Man Cybern.*, vol. 34, no. 4, pp. 1907–1916, Apr. 2004.
[6] N. Bouguila, "Count data modeling and classification using finite mixtures of distributions," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 186–198, Feb. 2011.
[7] F. Wentao, N. Bouguila, and D. Ziou, "Variational learning for finite dirichlet mixture models and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 762–774, May 2012.
[8] W. Hui, W. Xiao-Mei, and L. L. Lei, "Compact image representation model based on both nCRF and reverse control mechanisms," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 1, pp. 150–162, Jan. 2012.
[9] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1177–1193, Aug. 2012.
[10] M. N. Thanh, Q. M. J. Wu, and S. Ahuja, "An extension of the standard mixture model for image segmentation," *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1326–1338, Aug. 2010.
[11] M. N. Thanh and Q. M. J. Wu, "Gaussian-mixture-model-based spatial neighborhood relationships for pixel labeling problem," *IEEE Trans. Syst. Man Cybern. B*, vol. 42, no. 1, pp. 193–202, Jan. 2012.
[12] S. P. Chatzis and T. A. Varvarigou, "A fuzzy clustering approach toward hidden markov random field models for enhanced spatially constrained image segmentation," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1351–1361, May 2008.
[13] P. Clifford, "Markov random fields in statistics," in *Disorder in Physical Systems, a Volume in Honour of J. M. Hammersley on the Occasion of his 70th Birthday*. G. Grimmett and D. Welsh, Eds. Oxford, U.K.: Clarendon Press, 1990.
[14] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[15] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm," *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 45–57, Jan. 2001.
[16] J. Besag, "Statistical analysis of non-lattice data," *Statistician*, vol. 24, no. 3, pp. 179–195, 1975.
[17] J. Zhang, J. W. Modestino, and D. Langan, "Maximum-likelihood parameter estimation for unsupervised stochastic model-based image segmentation," *IEEE Trans. Image Process.*, vol. 3, no. 4, pp. 404–420, Apr. 1994.
[18] W. Qian and D. Titterington, "Estimation of parameters in hidden Markov models," *Philosoph. Trans. Roy. Soc. London A*, vol. 337, no. 1647, pp. 407–428, 1991.

[19] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Stat. Soc. B*, vol. 36, no. 2, pp. 192–326, 1974.
[20] W. Rudin, *Real and Complex Analysis*. New York: McGraw-Hill, 1987, pp. 62–65.
[21] K. Blekas, A. Likas, N. P. Galatsanos, and I. E. Lagaris, "A spatially constrained mixture model for image segmentation," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 494–498, Feb. 2005.
[22] [Online]. Available: http://infoman.teikav.edu.gr/~stkrini/pages/develop/FLICM/FLICM.html
[23] [Online]. Available: http://web.mac.com/soteri0s/Sotirios_Chatzis/Software.html
[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, Jun. 2001, pp. 416–423.
[25] R. Unnikrishnan and M. Hebert, "Measures of similarity," in *Proc. IEEE Workshop Comput. Vis. Appl. Conf.*, May 2005, pp. 394–400.
[26] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "A measure for objective evaluation of image segmentation algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 34–41.

Hyperbolic Hopfield Neural Networks

Masaki Kobayashi

***Abstract*— In recent years, several neural networks using Clifford algebra have been studied. Clifford algebra is also called geometric algebra. Complex-valued Hopfield neural networks (CHNNs) are the most popular neural networks using Clifford algebra. The aim of this brief is to construct hyperbolic HNNs (HHNNs) as an analog of CHNNs. Hyperbolic algebra is a Clifford algebra based on Lorentzian geometry. In this brief, a hyperbolic neuron is defined in a manner analogous to a phasor neuron, which is a typical complex-valued neuron model. HHNNs share common concepts with CHNNs, such as the angle and energy. However, HHNNs and CHNNs are different in several aspects. The states of hyperbolic neurons do not form a circle, and, therefore, the start and end states are not identical. In the quantized version, unlike complex-valued neurons, hyperbolic neurons have an infinite number of states.**

***Index Terms*— Clifford algebra, complex-valued neural networks, Hopfield neural networks (HNNs), hyperbolic algebra.**

I. Introduction

Complex-valued neural networks, which are an extension of ordinary neural networks, have been studied by many researchers. In recent years, further extensions based on Clifford algebra have been proposed. Clifford algebra, also called geometric algebra, encompasses the fields of complex numbers and quaternions, and hyperbolic algebra [1]. Clifford algebras produce inherent geometric properties. For example, the multiplication of a complex number is an extension and rotation operation in the complex plane.

In the field of feed-forward complex-valued neural networks, researchers have mainly concentrated on real–imaginary and amplitude–phase type complex-valued neural networks. The backpropagation learning rule for real–imaginary type complex-valued neural networks was first proposed by Benvenuto and Piazza [2] and Nitta [3], [4]. The backpropagation learning rule for amplitude–phase type complex-valued neural networks was proposed by Hirose [5]. Further extensions to Clifford neural networks have been proposed. Nitta [6] proposed a neural network model using quaternions that realizes partial rotation in a 4-D space [7]. Another proposed neural network model with quaternions realizes full rotation in a 3-D space [8].

In Clifford algebra, hyperbolic algebra can process information in non-Euclidean space. Ritter [9] proposed hyperbolic self-organizing map. Ontrup and Ritter [10], [11] applied it to text categorization and semantic navigation. Buchholtz and Sommer [12] and Nitta and Buchholz [13] studied hyperbolic neural networks.

Hopfield [14] proposed a recurrent neural network model referred to as the Hopfield neural networks (HNNs). HNNs have been extended to complex-valued Hopfield neural networks (CHNNs). Phasor neural networks are the simplest and most popular models of CHNNs [15]–[17]. In particular, quantized phasor neural networks, also referred to as multivalued neural networks, have been applied to multilevel data such as gray-scale images [18]–[20]. Several extensions to Clifford HNNs have been attempted. Isokawa *et al*. [21], [22] studied HNNs using quaternion. Kuroe [23] constructed hyperbolic HNNs (HHNNs) as an analog of real–imaginary type CHNNs. However, it is difficult to construct HHNNs as an analog of phase–amplitude type CHNNs. The HNN model with commutative quaternions proposed by Isokawa *et al*. [24] is incomplete because it requires HHNNs as an analog of phase–amplitude type HHNNs. Thus, HHNNs are necessary to study HNNs with higher dimensional Clifford algebra.

The aim of the present brief is to construct an HHNN model as an analog of a phasor CHNN. The proposed HHNN shares common concepts with a CHNN, such as the angle and energy. However, HHNNs and CHNNs are different in many aspects. The states of hyperbolic neurons do not form a circle, and, therefore, the start and end states are not identical. In the quantized version, unlike complex-valued neurons, hyperbolic neurons can have an infinite number of states.

The rest of this brief is organized as follows. Section II provides an introduction to CHNNs, which is used as an outline for constructing HHNNs. Section III describes hyperbolic algebra, which is necessary for constructing an HHNN. Then, we construct an HHNN in Section IV. Finally, Section V concludes this brief.

II. CHNNs

In this section, we briefly describe the conventional theory behind CHNNs [16]–[18]. The aim of this brief is to construct HHNNs as an analog of CHNNs. This section provides an outline for constructing HHNNs given in Section IV.

Manuscript received February 9, 2012; revised October 11, 2012; accepted November 23, 2012. Date of publication December 20, 2012; date of current version January 11, 2013.

The author is with the Interdisciplinary Graduate School of Medicine and Engineering, University of Yamanashi, Yamanashi 400-8511, Japan (e-mail: k-masaki@yamanashi.ac.jp).

Digital Object Identifier 10.1109/TNNLS.2012.2230450