A Spatially Constrained Generative Model and an EM Algorithm for Image Segmentation

Aristeidis Diplaros, *Student Member, IEEE,* Nikos Vlassis, and Theo Gevers, *Member, IEEE*

***Abstract*— We present a novel spatially constrained generative model and an EM algorithm for model-based image segmentation. The generative model assumes that the unobserved class labels of neighboring pixels in the image are generated by prior distributions with similar parameters, where similarity is defined by entropic quantities relating the neighboring priors. In order to estimate model parameters from observations, we derive a spatially constrained EM algorithm that iteratively maximizes a lower bound on the data log-likelihood, where the penalty term is data-dependent. Our algorithm is very easy to implement, and is similar to the standard EM algorithm for Gaussian mixtures with the main difference that the labels posteriors are 'smoothed' over pixels between each E- and M-step by a standard image filter. Experiments on synthetic and real images show that our algorithm achieves competitive segmentation results compared to other Markov-based methods, and is in general faster.**

***Index Terms*— Image segmentation, Hidden Markov random fields, EM algorithm, Bound optimization, Spatial clustering.**

I. Introduction

In the seminal papers of [1], [2], Markov random field (MRF) models were introduced for image analysis. Subsequently, they have been used by many researchers for the solution of a number of important problems in image analysis such as image restoration, segmentation, edge-preserving filtering to name a few (see, e.g., [3]–[5] and references therein). MRF models provide a powerful and formal way to account for spatial dependencies between image pixels. A drawback of the aforementioned models is that it is typically very expensive to properly account for the pixels spatial dependencies during inference/learning. Various approximations have been introduced in order to make the problem tractable (e.g., multiresolution MRF [6]), but the high cost of MRF-based methods, as compared to other methods, still remains.

In order to overcome this computational cost, several alternatives to MRF models have been proposed. These include modeling approaches that aim at directly defining hierarchical (Markovian) models on trees as in [7], [8]. Also, Markov chains [9], [10] have been used, where the 2-D image is transformed into one-dimensional chain using some predefined sweep. These approaches, while being in general more computationally efficient compared to MRF, are less powerful in capturing spatial dependencies. In particular, as stated in [7], hierarchical models have a tendency to produce block-like artifacts in the final estimates. In [9] it is reported that Markov chains, while being more robust, they tend to produce more irregular borders.

A particular problem that has been addressed by MRF models is image segmentation, the task of identifying homogeneous image regions or determining their boundaries. Formally, the task of image segmentation is to partition an image into a set of nonoverlapping regions $\{R_1, \ldots, R_K\}$, so that the variation of some property (such as intensity, color, texture, etc.) within each region R_k is either constant, or follows a simple model Φ_k (e.g. Gaussian) [11]. What makes this problem especially difficult is that the parameters for each model Φ_k, as well as the corresponding regions R_k have to be simultaneously estimated from the input image. To solve it, prior MRF models are commonly used in conjunction with iterative estimation procedures like the Expectation Maximization (EM) or other iterative algorithms [5].

In this paper we introduce a novel generative model and an EM algorithm for Markov-based image segmentation. The proposed generative model assumes that the hidden class labels of the pixels are generated by prior distributions that share similar parameters for neighboring pixels. In order to define a notion of similarity between neighboring pixels priors, we introduce a pseudo-likelihood quantity that couples neighboring priors by means of entropic quantities like the Kullback-Leibler divergence. To estimate the unknown parameters of the pixels prior distributions, as well as the parameters of the observation model, we derive an EM algorithm that iteratively maximizes an appropriately constructed lower bound on the data log-likelihood. The proposed algorithm is very similar to the standard EM algorithm for mixture models, with the main difference that the mixing weights (posterior distributions) of neighboring pixels are coupled in each EM iteration by an averaging operation. This results in a simple and efficient scheme for incorporating spatial constraints in an EM framework for image segmentation. Experimental results demonstrate the potential of the method on synthetic and real images.

The rest of the paper is organized as follows. In Section II we briefly review the problem of image segmentation by discussing three classes of generative models that are commonly used in the literature. In Section III we describe our proposed algorithm in detail and draw parallels with other existing approaches. In Section IV we show experimental results, and we conclude with a discussion in Section V.

II. A Review of MRF-based models for image segmentation

In this section we discuss three commonly used probabilistic graphical models for image segmentation. The first one is a

The authors are with the Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands. E-mail: diplaros@science.uva.nl, vlassis@science.uva.nl, gevers@science.uva.nl

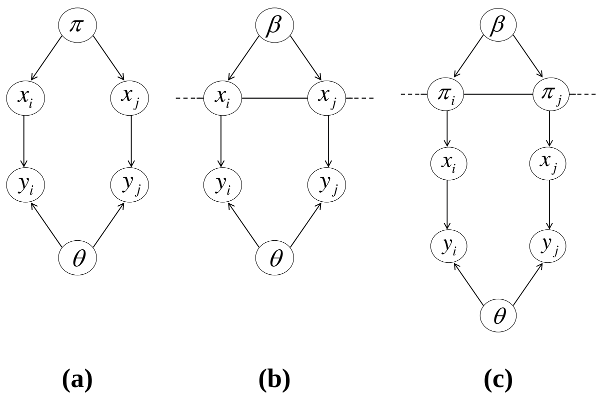

Fig. 1. Three commonly used probabilistic graphical models for image segmentation. Pixel j is assumed to be in the neighborhood of pixel i. (a) The standard mixture model. (b) Markov random field on pixels labels. (c) Markov random field on pixel label priors.

standard mixture model in which spatial dependencies between pixels are not explicitly incorporated into the generative model. The second one assumes that the hidden pixel labels form a Markov field. The third one, which is the one adopted in our method, assumes that the prior distributions that generate the pixel labels form a Markov field.

We first introduce the notation used throughout the paper. We are dealing with images consisting of n pixels. For a pixel i we denote by y_i its observed value; for gray scale images this is a scalar with values from 0 to 255, for color images this can be, e.g., a three-component vector with R,G,B values. Moreover we assume that each pixel i belongs to a single class (image segment or region) which is indexed by the hidden random variable x_i. The latter takes values from a discrete set of labels $1, \ldots, K$. In all models we consider, we assume an observation model in the form $p(y_i|x_i)$ that describes the likelihood of observing y_i given pixel label x_i. This model is a Gaussian [1] density conditional on the class label k, i.e.:

$$p(y_i|x_i = k, \theta) = \mathcal{N}(m_k, C_k) \tag{1}$$

that is parameterized by its mean m_k and (co)variance C_k, collectively denoted for all components by θ. In all models we consider in this paper the observation model is shared among all pixels, that is, the parameters $\theta = \{m_k, C_k\}_{k=1}^{K}$ are independent of the pixel index i.

A. Standard mixture model

This is the standard (Gaussian) mixture model [13] in which the spatial dependencies between pixels can be implicitly introduced by using the pixels coordinates as an extra feature [14]. This model is also employed in our previous work [15]. The corresponding generative model is shown in Fig. 1(a), where we show two neighboring pixels i and j. The model assumes a common prior distribution π that independently generates all pixel labels x_i. This prior π is assumed to be a discrete distribution with K states, whose parameters π_k, $k = 1, \ldots, K$ are unknown, and holds:

$$p(x_i = k|\pi) = \pi_k \tag{2}$$

where we see that no spatial dependence between pixels is a priori assumed (the prior π_k has no dependence on pixel index i). Each pixel label x_i generates a pixel observation y_i from a shared Gaussian distribution $p(y_i|x_i, \theta)$ with parameters θ as described above. The log-likelihood of all observations $y = \{y_1, \ldots, y_n\}$ for the n pixels is given by

$$\mathcal{L}_1(\theta, \pi) = \sum_{i=1}^{n} \log \sum_{x_i=1}^{K} p(y_i|x_i, \theta) p(x_i|\pi) \tag{3}$$

$$= \sum_{i=1}^{n} \log \sum_{k=1}^{K} p(y_i|x_i = k, \theta) \pi_k. \tag{4}$$

The EM algorithm [16], [17] learns the parameters π and θ by iteratively maximizing a lower bound of the log-likelihood $\mathcal{L}_1$. This bound is a function of the model parameters and a set of auxiliary distributions q_i:

$$\mathcal{F}_1(\theta, \pi, \{q_i\}) = \mathcal{L}_1(\theta, \pi) - \sum_i D(q_i \parallel p_i) \tag{5}$$

where D denotes the Kullback-Leibler divergence between two discrete distributions which is defined as $D(A \parallel B) = E_A[\log A - \log B]$, and which is always nonnegative and becomes zero when $A = B$. The distribution $p_i \equiv p(x_i|y_i)$ is the Bayes posterior of label x_i given y_i and parameters θ, π:

$$p(x_i = k|y_i) = \frac{p(y_i|x_i = k)\pi_k}{p(y_i)}. \tag{6}$$

In the EM algorithm we repeatedly maximize $\mathcal{F}_1$ over its parameters, in a coordinate ascent fashion. In the E-step we fix θ, π and optimize over q_i, and in the M-step we fix q_i and optimize over θ, π. This gives:

$$\pi_k = \frac{1}{n} \sum_{i=1}^{n} p(x_i = k|y_i), \tag{7}$$

$$m_k = \frac{1}{n\pi_k} \sum_{i=1}^{n} p(x_i = k|y_i) y_i, \tag{8}$$

$$C_k = \frac{1}{n\pi_k} \sum_{i=1}^{n} p(x_i = k|y_i) y_i y_i^{\top} - m_k m_k^{\top}. \tag{9}$$

Similar equations we obtain in our algorithm which we will explain in detail in Section III.

B. Markov random field on pixel labels

This model has been used for instance in [1], [18]–[23], and is graphically shown in Fig. 1(b). Here the vector of pixel labels $x = \{x_1, \ldots, x_n\}$ is assumed to be a (hidden) Markov random field (MRF) with Gibbs joint probability distribution

$$p(x|\beta) = \frac{1}{Z(\beta)} \exp(-H(x|\beta)) \tag{10}$$

where H is an energy function

$$H(x|\beta) = \sum_{C} V_c(x_c|\beta) \tag{11}$$

[1] This model cannot handle highly textured regions but there are alternatives (e.g., FRAME [12]) that can.

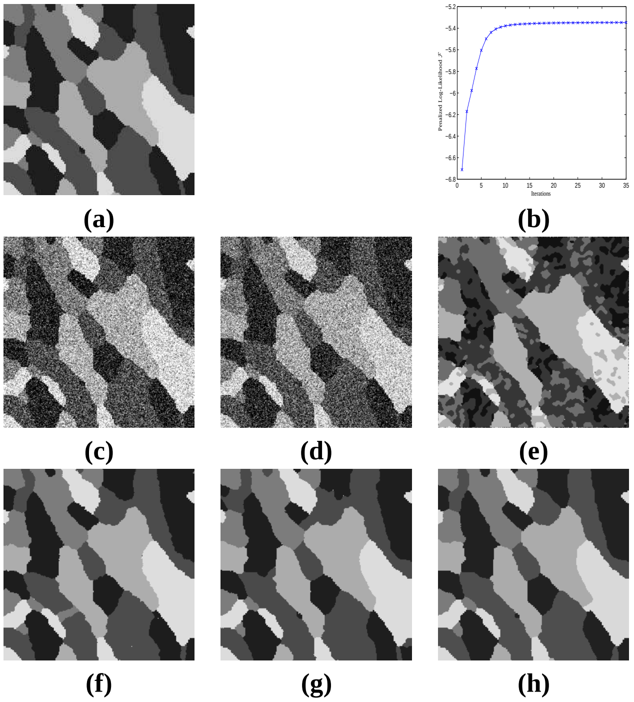

Fig. 2. Experiment with synthetic image and white Gaussian noise. (a) The five-class synthetic image. (b) The maximization progress of the penalized log-likelihood $\mathcal{F}$ of our algorithm for this experiment. (c) The noise corrupted five-class image with additive white Gaussian noise ($\sigma = 52$). (d) The segmentation result of the standard EM algorithm (MCR 53.7%). (e) The segmentation result of ICM with running time of 39 sec (MCR 31.7%). (f) The segmentation result of SimF with running time of 90 sec (MCR 2.88%). (g) The segmentation result of MeanF with running time of 100 sec (MCR 3.89%). (h) The segmentation result of our approach with running time of 92 sec (MCR 1.78%).

parameterized by a set of clique potentials V_c and some nonnegative scalar β. To deal with the inherent intractability of MRF (due to the normalizer in (10)), a standard approximation suggested by Besag [2], [24] and used, e.g., by [18], [23] involves factorizing the joint distribution as

$$p(x|\beta) \approx \prod_i p(x_i|x_{N_i}, \beta) \tag{12}$$

where N_i denotes the set of neighboring pixels of pixel i. Using this approximation, the likelihood of the complete data (hidden pixel labels and pixel observations) reads

$$p(y, x|\theta, \beta) = \prod_i p(y_i|x_i, \theta) p(x_i|x_{N_i}, \beta). \tag{13}$$

In particular, by clamping x_{N_i} for each pixel i to $\widetilde{x}_{N_i}$ the observed data log-likelihood becomes

$$\mathcal{L}_2(\theta, \beta) = \sum_i \log \sum_{x_i=1}^{K} p(y_i|x_i, \theta) p(x_i|\widetilde{x}_{N_i}, \beta). \tag{14}$$

Note a similarity between $\mathcal{L}_2$ in (14) and $\mathcal{L}_1$ in (4); they are both mixture likelihoods with a parameter vector θ in the observation model that is shared by all pixels. Their difference is that the prior π in $\mathcal{L}_1$ is shared by all pixels, whereas the prior $p(x_i|\widetilde{x}_{N_i}, \beta)$ in $\mathcal{L}_2$ is different for each pixel i and depends on the neighbors N_i of the pixel and the parameter β. We kindly refer the reader to [18] for more details.

Maximizing $\mathcal{L}_2$ w.r.t. θ and β can be carried out by the EM algorithm. In [23], for instance, each EM iteration involves a mean-field like procedure in which the label $\widetilde{x}_i$ of a pixel i is sequentially estimated from the values of its neighboring pixels N_i as, e.g.,

$$\widetilde{x}_i = \arg\max_{x_i} p(x_i|y_i, \widetilde{x}_{N_i}, \theta, \beta), \tag{15}$$

where $p(x_i|y_i, \widetilde{x}_{N_i}, \theta, \beta)$ is the Bayes posterior given the parameters θ and β of the previous iteration, and $\widetilde{x}_{N_i}$ includes a mix of previous and current estimated values (with respect to the current sweep over pixels). For each EM iteration the above procedure effectively requires computing a complete image restoration. We kindly refer the reader to [23] for more details.

C. Markov random field on pixel label priors

This is the model that we adopt in this work, and which has also been used in [9], [11], [25], [26]. It is graphically shown in Fig. 1(c). Here the pixel label priors $\pi = \{\pi_1, \ldots, \pi_n\}$ are treated as random variables that form a Markov random field, whereas the pixel labels x_i are assumed conditionally independent given the priors. In [25] the random field of the priors is defined as

$$p(\pi|\beta) = \frac{1}{Z(\beta)} \exp(-U(\pi|\beta)) \tag{16}$$

where U is an energy function in the form

$$U(\pi|\beta) = \beta \sum_i \sum_{j \in N_i} \sum_{k=1}^{K} (\pi_{ik} - \pi_{jk})^2 \tag{17}$$

parameterized by a scalar β. In the above notation, π_{ik} refers to the component k of the prior distribution of pixel i. In this model the priors $\{\pi_i\}$ are estimated together with θ by the EM algorithm. Translating the conditional independencies induced by the above graphical model, the penalized log-likelihood of the observed data reads (ignoring constants)

$$\mathcal{L}_3(\theta, \pi) = \sum_i \log \sum_{k=1}^{K} p(y_i|x_i = k, \theta)\pi_{ik} \\ - \beta \sum_i \sum_{j \in N_i} \sum_{k=1}^{K} (\pi_{ik} - \pi_{jk})^2. \tag{18}$$

We note the similarity of $\mathcal{L}_3$ with the $\mathcal{L}_1$ of the standard mixture model. Here, however, there are n different π_i distributions, one for each pixel i, and additionally there is a penalty term (the energy U) that penalizes neighboring pixels with different labels. Note that this model enforces spatial dependencies between pixels in a different way than the MRF model of the previous section. Namely, here the assumption is that neighboring pixels have similar prior distributions that generate their pixel labels, whereas in the classical MRF model we postulate a Markov random field directly on the pixel labels. An attractive property of this model, as we explain below, is that the E-step of the EM algorithm is easier to carry out since we do not need to estimate a restoration of the image. On the other hand, the M-step is more complicated as

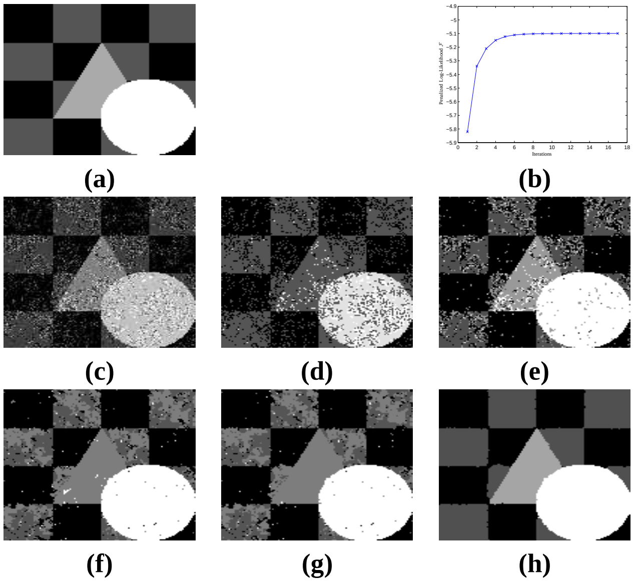

Fig. 3. Experiment with synthetic image and spatially correlated Gaussian noise. (a) The original four-class image. (b) The maximization progress of the penalized log-likelihood $\mathcal{F}$ of our algorithm for this experiment. (c) The noise corrupted image with spatially correlated Gaussian noise. (d) The segmentation result of the standard EM algorithm (MCR 28.72%). (e) The segmentation result of ICM with running time of 5 sec (MCR 16.8%). (f) The segmentation result of SimF with running time of 22 sec (MCR 17.6%). (g) The segmentation result of MeanF with running time of 23 sec (MCR 17.4%). (h) The segmentation result of our approach with running time of 6 sec (MCR 0.68%).

it also involves the penalty term $-U(\pi)$ of (16). Indeed, the computational effort in [11], [25], [26] goes in the estimation of the $\{\pi_i\}$ priors in the M-step which requires solving a constrained optimization problem (since π_i is a discrete distribution with $\sum_k \pi_{ik} = 1$ for each i).

The main motivation for using this model as opposed to the traditional MRF model on pixel labels (of Section II-B), is its flexibility with respect to the initial conditions. This flexibility is manifested in the shape of the objective function; In the traditional MRF model of Section II-B, the penalized log-likelihood function will be sharper and will contain several local maxima, and hence it will be more sensitive to the initial solution. In the current model, the field constraints are directly enforced over the parameters of the label priors, resulting in a smoother objective function. This can be intuitively seen by noting that distinct parameter values for some priors may induce exactly the same pixel labels, and therefore searching in the space of priors (in the M-step of our algorithm as we show next) will be easier than searching in the space of labels. This search will be also facilitated by the fact that the space of prior parameters is continuous, as opposed to the discrete nature of the space of pixel labels. Interestingly, although the current model contains more parameters (Kn, a vector of K parameters for each π_i, as opposed to n of the traditional MRF model), the smoothness of the objective function, as argued above, allows the result to be less dependent on the initialization. The above arguments will be experimentally verified in Section IV.

III. The Proposed Method

In our method we also use the graphical model of Fig. 1(c), but we use a different modeling strategy for the spatial dependencies between the priors and a different algorithm for learning the unknown parameters. As in [23] we employ the Besag approximation for modeling the joint density over pixel priors:

$$p(\pi|\beta) \approx \prod_i p(\pi_i|\pi_{N_i}, \beta), \tag{19}$$

where we define π_{N_i} as a mixture distribution over the priors of neighboring pixels of pixel i, i.e.,

$$\pi_{N_i} = \sum_{\substack{j \in N_i \\ j \neq i}} \lambda_{ij} \pi_j, \tag{20}$$

where λ_{ij} are fixed positive weights and for each i holds $\sum_j \lambda_{ij} = 1$. The mixing weight λ_{ij} depends on the relative offset between the pixels i and j, while the mixture does not include the prior of the same pixel i. Note that the evaluation of this mixture corresponds to a convolution operation $\pi_{\cdot k} * \lambda$, for each component k, where λ is a symmetric linear image filter with zero coefficient in its center and nonnegative coefficients elsewhere that sum to one. See Section IV-B for more details about filter related issues.

Further, for the conditional density $p(\pi_i|\pi_{N_i}, \beta)$ we assume an approximate log-model in the form (ignoring constants)

$$\log p(\pi_i|\pi_{N_i}, \beta) = -\beta\big[D(\pi_i \parallel \pi_{N_i}) + H(\pi_i)\big]. \tag{21}$$

where $D(\pi_i \parallel \pi_{N_i}) = \sum_{k=1}^K \pi_{ik} \log \pi_{ik} - \sum_{k=1}^K \pi_{ik} \log \pi_{N_i k}$ is the Kullback-Leibler divergence between π_i and π_{N_i} which is always nonnegative and becomes zero when $\pi_i = \pi_{N_i}$, and $H(\pi_i) = -\sum_{k=1}^K \pi_{ik} \log \pi_{ik}$ is the entropy of the distribution π_i which is always nonnegative and reaches its maximum when π_i is uniform. The KL term $D(\pi_i \parallel \pi_{N_i})$ intuitively expresses the degree of similarity between the prior of pixel i and the priors of its neighbors, and it provides a way of constraining neighboring pixels to have similar class labels. Similarly, the entropy term $H(\pi_i)$ constrains the label priors to be as informative as possible: in homogeneous regions it is reasonable to expect that neighboring pixels have similar priors, and that these priors are far from uniform. It is important to emphasize that the entropy term $H(\pi_i)$ does not enforce the priors π_i to be of a particular shape, but it merely constrains them to be as informative as possible.

MAP (maximum a posteriori) estimation of the parameters θ and π_i of our model involves maximizing the data log-likelihood (the first term of $\mathcal{L}_3$ in (18)) penalized by the approximate log-prior term (21). Direct optimization of this penalized log-likelihood is, however, difficult because of the coupling of neighboring pixels priors π_i (which would require constraint optimization techniques as those in [26]). To facilitate optimization, we introduce an approximation that makes use of an auxiliary set of distributions s_i as follows:

$$\log p(\pi_i|\pi_{N_i}, \beta, s_i) \approx -\beta\big[D(s_i \parallel \pi_i) + D(s_i \parallel \pi_{N_i}) + H(s_i)\big] \tag{22}$$

which decouples the pixels priors and allows for an efficient coordinate ascent EM-like optimization as we show next. Note

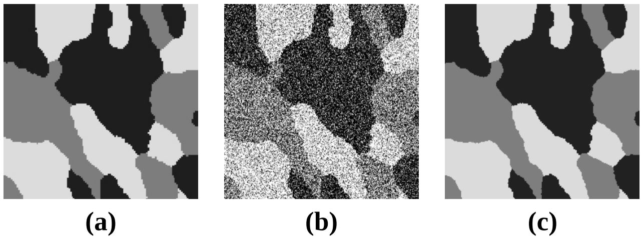

Fig. 4. The effects of using different filters on the segmentation results. (a) The three-class synthetic image. (b) The noise corrupted image with additive white Gaussian noise of $\sigma = 95$. (c) The segmentation result using Gaussian kernel with $\sigma = 1$ (MCR 1.08%).

that in the above approximation, when $s_i = \pi_i$ then (22) becomes identical to (21).

The above approximate log-prior involves only entropic quantities and therefore is a nonnegative quantity that lower bounds the data log-likelihood. Note that this penalty term (Bayesian prior) does not depend on the observed data (image pixels). Recently, other approaches have appeared in the machine learning literature that incorporate constraints into a learning problem by lower bounding the data log-likelihood using data-dependent penalty terms [27], [28]. Typically those bounds involve a KL distance between posterior distributions, thus explicitly incorporating the observed data into the bounding terms, and utilizing useful domain knowledge. In the same spirit we introduce an additional penalty term involving posterior distributions in the form

$$-\frac{1}{2}\big[D(q_i \parallel p_i) + D(q_i \parallel p_{N_i}) + H(q_i)\big] \tag{23}$$

where q_i is an arbitrary class distribution for pixel i, and p_i is the posterior class distribution of a pixel i computed for model parameters θ and prior π_i by the Bayes rule:

$$p_{ik} \equiv p(x_i = k|y_i, \theta, \pi_{ik}) = \frac{p(y_i|x_i = k, \theta)\pi_{ik}}{\sum_{l=1}^{K} p(y_i|x_i = l, \theta)\pi_{il}}. \tag{24}$$

The coefficient $1/2$ in the penalty term above was chosen because it allows a tractable M-step as we show next. Putting all terms together, the penalized log-likelihood of the observed data as a function of the model parameters and the introduced auxiliary distributions $s = \{s_i\}$ and $q = \{q_i\}$ reads (ignoring constants):

$$\begin{aligned}\mathcal{F}(\theta, \pi, s, q) = \sum_i \Big[&\log \sum_k p(y_i|x_i = k, \theta)\pi_{ik} \\ &- \frac{1}{2}\big[D(q_i \parallel p_i) + D(q_i \parallel p_{N_i}) + H(q_i)\big] \\ &- \beta\big[D(s_i \parallel \pi_i) + D(s_i \parallel \pi_{N_i}) + H(s_i)\big]\Big].\end{aligned} \tag{25}$$

Our EM algorithm maximizes the energy $\mathcal{F}$ in (25) by coordinate ascent. In the E-step we fix θ and π and maximize $\mathcal{F}$ over s and q. In the M-step we fix s and q and maximize $\mathcal{F}$ over θ and π. Next we show how these two steps can be performed.

E-step: In the E-step we optimize over s_i for a pixel i, assuming θ and π fixed. A similar derivation holds for q_i. The terms of $\mathcal{F}$ involving s_i are:

$$\begin{aligned}&-\sum_k s_{ik}\log s_{ik} + \sum_k s_{ik}\log \pi_{ik} - \sum_k s_{ik}\log s_{ik} \\ &\quad + \sum_k s_{ik}\log \pi_{N_i k} + \sum_k s_{ik}\log s_{ik} = \\ &\quad\quad -\sum_k s_{ik}\log s_{ik} + \sum_k s_{ik}\log(\pi_{ik}\pi_{N_i k}).\end{aligned} \tag{26}$$

The latter is an (un-normalized) negative KL-divergence which becomes zero when

$$s_i \propto \pi_i \pi_{N_i}. \tag{27}$$

Similarly we obtain

$$q_i \propto p_i p_{N_i}. \tag{28}$$

M-step: In the M-step we fix s and q and maximize $\mathcal{F}$ over θ and π. The terms of $\mathcal{F}$ involving the priors π_i and the posteriors p_i (and therefore θ and π_i) are:

$$\begin{aligned}&\log \sum_k p(y_i|x_i = k, \theta)\pi_{ik} \\ &\quad - \frac{1}{2}\big[D(q_i \parallel p_i) + \sum_{\substack{j\in N_i \\ j\neq i}} D(q_j \parallel p_{N_j})\big] \\ &\quad\quad - \beta\big[D(s_i \parallel \pi_i) + \sum_{\substack{j\in N_i \\ j\neq i}} D(s_j \parallel \pi_{N_j})\big].\end{aligned} \tag{29}$$

We show now the derivation involving the posteriors. The terms of (29) involving only p_i are:

$$-\frac{1}{2}\big[D(q_i \parallel p_i) + \sum_{\substack{j\in N_i \\ j\neq i}} D(q_j \parallel p_{N_j})\big], \tag{30}$$

which, ignoring terms independent of p_i, reads

$$-\frac{1}{2}\big[-\sum_k q_{ik}\log p_{ik} - \sum_{\substack{j\in N_i \\ j\neq i}} \sum_k q_{jk}\log p_{N_j k}\big], \tag{31}$$

where

$$p_{N_j} = \sum_{\substack{m\in N_j \\ m\neq j}} \lambda_{jm} p_m = \lambda_{ji} p_i + \sum_{\substack{m\in N_j \\ m\neq i,j}} \lambda_{jm} p_m. \tag{32}$$

The mixture p_{N_j} appears in the last term of (31) for all pixels j that are neighbors of pixel i. To make the M-step tractable, we bound these terms using Jensen's inequality:

$$\begin{aligned}\log p_{N_j k} = \log \sum_{\substack{m\in N_j \\ m\neq j}} \lambda_{jm} p_{mk} \quad &\geq \\ \lambda_{ji}\log p_{ik} + \sum_{\substack{m\in N_j \\ m\neq i,j}} \lambda_{jm}\log p_{mk}\ ,\ &\forall k.\end{aligned} \tag{33}$$

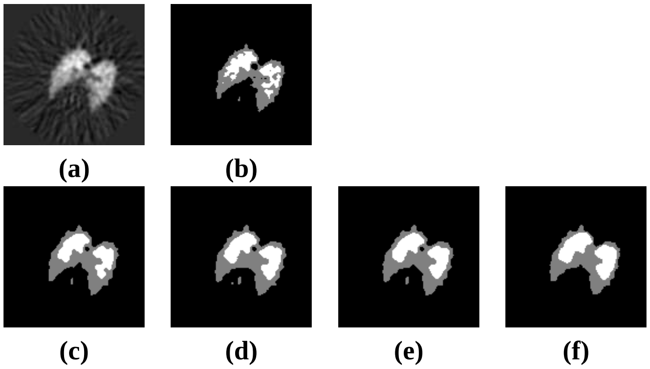

Fig. 5. Segmentation example of the PET image of a dog lung with $K = 3$. (a) The original image. (b) Simple thresholding based initialization. (c) The segmentation result of ICM. (d) The segmentation result of SimF. (e) The segmentation result of MeanF. (f) The segmentation result of our approach.

Using (31) and (33) and noting that $\lambda_{ji} = \lambda_{ij}$, we finally get (ignoring again terms independent of p_i):

$$\frac{1}{2}\big[\sum_k q_{ik}\log p_{ik} + \sum_k \sum_{\substack{j\in N_i\\ j\neq i}} \lambda_{ji} q_{jk}\log p_{ik}\big] \quad \Rightarrow$$

$$\frac{1}{2}\sum_k \left(q_{ik} + q_{N_i k}\right)\log p_{ik} \tag{34}$$

where the distribution q_{N_i} is

$$q_{N_i} = \sum_{\substack{j\in N_i\\ j\neq i}} \lambda_{ij} q_j. \tag{35}$$

An identical derivation holds for the priors producing a term:

$$\beta \sum_k \left(s_{ik} + s_{N_i k}\right)\log \pi_{ik}. \tag{36}$$

In total, the terms of $\mathcal{F}$ (actually a lower bound of it since we employed (33)) involving the priors π_i and the posteriors p_i are:

$$\log \sum_k p(y_i|x_i = k, \theta)\pi_{ik} + \frac{1}{2}\sum_k \left(q_{ik} + q_{N_i k}\right)\log p_{ik} + \beta \sum_k \left(s_{ik} + s_{N_i k}\right)\log \pi_{ik}. \tag{37}$$

Expanding the posterior p_i in the above terms and noting that $\frac{1}{2}\sum_k \left(q_{ik} + q_{N_i k}\right) = 1$, we immediately see that the log-likelihood term cancels. Then (37) reads:

$$\frac{1}{2}\sum_k \left(q_{ik} + q_{N_i k}\right)\log p(y_i|x_i = k, \theta)\pi_{ik} + \beta \sum_k \left(s_{ik} + s_{N_i k}\right)\log \pi_{ik} \quad \Rightarrow$$

$$\frac{1}{2}\sum_k \left(q_{ik} + q_{N_i k}\right)\log p(y_i|x_i = k, \theta) + \sum_k \left[\frac{1}{2}\left(q_{ik} + q_{N_i k}\right) + \beta\left(s_{ik} + s_{N_i k}\right)\right]\log \pi_{ik} \tag{38}$$

Collecting all terms of (38) involving π_i and differentiating w.r.t. π_i (using a Lagrange multiplier to ensure $\sum_k \pi_{ik} = 1$), we can easily show that we get:

$$\pi_i = \frac{1}{(1+2\beta)}\left[\frac{1}{2}(q_i + q_{N_i}) + \beta(s_i + s_{N_i})\right]. \tag{39}$$

Similarly, differentiating (38) over θ, we get the following update equations for the means and covariances of the K Gaussian components:

$$m_k = \frac{\sum_i (q_{ik} + q_{N_i k})\, y_i}{\sum_i (q_{ik} + q_{N_i k})}, \tag{40}$$

$$C_k = \frac{\sum_i (q_{ik} + q_{N_i k})\, y_i y_i^\top}{\sum_i (q_{ik} + q_{N_i k})} - m_k m_k^\top. \tag{41}$$

Note that the update equations for θ are analogous to those in the standard mixture model, with the difference that here the weights correspond to 'smoothed' pixel posteriors. The use of such spatially smoothed weights in the M-step of the EM algorithm is a key element in our approach that distinguishes it from other works. The complete algorithm is shown in Alg. 1.

Concerning the initialization of the parameter vector θ, we employ the k-means algorithm, but we note that other clustering algorithms [29] can be used also. The initialization of the priors $\{\pi_i\}$ in this work is uniform. Concerning time complexity, each EM step has cost that is linear in the number of pixels in the image and linear in the number of class labels, as we can directly see in the EM update equations, for instance, (27) and (40). Additionally, our update equations involve a convolution operation for computing the 'smoothed' distributions π_{N_i}, p_{N_i}, s_{N_i}, and q_{N_i}, which, for each pixel i, has constant runtime complexity (since the size of the filter is fixed). Concerning the convergence rate of our algorithm, we have experimentally observed that our method can quickly reach a good solution indicated by high values of $\mathcal{F}$ (see Fig. 2(b) and Fig. 3(b)). This is in accordance with similar findings for the batch EM algorithm in the literature [17], but for which theoretical evidence is, to our knowledge, still lacking.

Finally, in this work we set $\beta = 0.5$. It is a matter of future work to investigate ways to incorporate β in the optimization process (as in [22], [23]).

IV. Experiments

In this section we demonstrate the performance of our algorithm on synthetic and real images. Specifically, in Section IV-A we evaluate the performance of our segmentation algorithm in the presence of noise. Section IV-B includes additional experiments to test the algorithm's behavior w.r.t the choice of parameters and initialization. Section IV-C presents segmentation results on gray level images. Finally, in Section IV-D we show segmentation results on color images.

A. Noise-Corrupted Synthetic Images

We first illustrate our algorithm on synthetic images and consider its robustness against noise. We use the same synthetic images as in [20], [26]. These are simulated three-class and five-class images (see Fig. 4(a) and Fig. 2(a)) sampled from an MRF model using the Gibbs sampler [1]. In Fig. 2(c) we show the five-class image after adding white Gaussian

```
 1: Initialize the parameter vector $\theta$, e.g., using k-means.
 2: Initialize the priors $\{\pi_i\}$, e.g., uniform $\pi_{ik} = \frac{1}{K}, \forall i, k$.
 3: E-step: Compute posterior probabilities $p_i$ using (24) and
    the current estimates of $\theta$ and $\{\pi_i\}$.
 4: Compute $\{s_i\}$ according to (27) and (20) and normalize
    each $s_i$ so that $\sum_k s_{ik} = 1$.
 5: Compute $\{q_i\}$ according to (28) and normalize each $q_i$ so
    that $\sum_k q_{ik} = 1$.
 6: M-step: Update the parameter vector $\theta$ using (40) and
    (41).
 7: Update $\{\pi_i\}$ according to (39).
 8: Evaluate $\mathcal{F}$ from (25).
 9: If convergence of $\mathcal{F}$, e.g., $|\mathcal{F}_{t+1}/\mathcal{F}_t - 1| < \epsilon$
10: then stop.
11: else go to step 3.
12: end if
```

Algorithm 1: The proposed EM algorithm for image segmentation

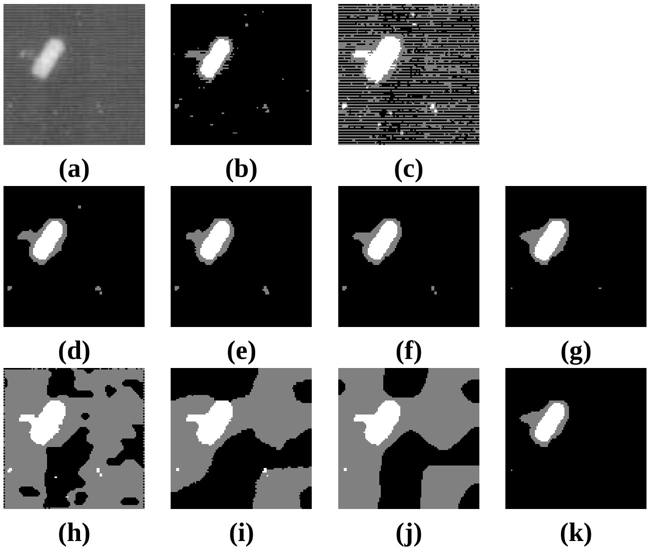

Fig. 6. Segmentation example of the buoy image with $K = 3$. (a) The original image. (b) K-means initialization. (c) Initialization based on EM for independent mixtures. (d) The segmentation result of ICM with k-means initialization. (e) The segmentation result of SimF with k-means initialization. (f) The segmentation result of MeanF with k-means initialization. (g) The segmentation result of our approach with k-means initialization. (h) The segmentation result of ICM with EM initialization. (i) The segmentation result of SimF with EM initialization. (j) The segmentation result of MeanF with EM initialization. (k) The segmentation result of our approach with EM initialization.

noise with $\sigma = 52$. In Fig. 2(d) we show the segmentation results of the standard EM algorithm using the generative model discussed in Section II-A. It is clear from these examples that in presence of noise, an algorithm that does not use spatial constraints cannot produce meaningful segmentation results. On the contrary, a method like ours that does take into account the spatial relation of pixels can successfully segment these noisy images, as we demonstrate in Fig. 2(h). The total running time of our method (from initialization till convergence) was 15 seconds for the three-class image and 92 seconds for the five-class one (as shown in Fig. 2(h)). All times (for our method) refer to a Matlab implementation running on a 3.0GHz PC-based workstation. In these synthetic images the ground truth is known (the true assignment of pixels to the K classes), which allows us to evaluate the performance of the various methods in terms of the misclassification ratio (MCR). This is simply the number of misclassified pixels divided by the total number of pixels [20].

TABLE I

MISCLASSIFICATION RESULTS FOR THE THREE-CLASS IMAGE.

	Noise σ					
	18	25	28	47	52	95
EM	0.61%	3.64%	5.82%	20.5%	23.9%	42.3%
HMRF-EM	-	-	0.12%	1.04%	-	8.73%
ICM	0.03%	0.15%	0.21%	0.81%	0.96%	4.01%
SimF	0.02%	0.08%	0.13%	0.43%	0.62%	1.52%
MeanF	0.02%	0.08%	0.12%	0.39%	0.54%	1.32%
SVFMM	0.2%	1.5%	-	-	21%	-
SVFMM-QD	0.1%	1%	-	-	13%	-
Our Approach	0.03%	0.1%	0.1%	0.46%	0.50%	1.18%
run time of our approach	5.8sec	7.7sec	8.8sec	14sec	15sec	29sec

We compare our method with related methods based on hidden Markov random fields: the ICM algorithm [2] (termed ICM); the Spatially Variant Finite Mixture Model method [25] (termed SVFMM); the extension to SVFMM [26] (termed SVFMM-QD); the Hidden Markov Random Field Model based on EM framework proposed in [20] (termed HMRF-EM); the Mean field and the Simulated field methods of [21], [23] (termed MeanF and SimF respectively). Finally, we have also included the comparison with the standard EM algorithm.

Tables I and II contain the misclassification ratio results of the previously mentioned methods for the same synthetic images and for various amounts of noise. For the methods SVFMM, SVFMM-QD and HMRF-EM we replicate the misclassification ratio results reported in the corresponding papers. For SimF and MeanF methods we used a software implementation developed by the authors of [21], [23] (in C) and is publicly available at http://mistis.inrialpes.fr/software/SEMMS.html. This software also includes an implementation of the ICM algorithm. We initialize ICM, SimF and MeanF methods in the same way that we initialize our algorithm. As in [21], [23] the number of iteration for SimF and MeanF was set to 100 and the ICM was run until convergence. For our method the restorations shown result from the maximization of the estimated prior distribution π_i (which the algorithm learns). The SVFMM, SVFMM-QD and HMRF-EM methods use a first order neighborhood system, while, ICM, SimF and MeanF methods use a second order neighborhood system.

In Fig. 3 we present the segmentation results for a synthetic four-class image in which we have added spatially correlated noise. To generate this noise we sampled a configuration of binary indicators from a standard Potts-model MRF [1] using Gibbs sampling. If the drawn indicator of a pixel was one, then we added Gaussian noise with standard deviation $\sigma = 52$ to the corresponding pixel, otherwise we didn't add noise. Clearly this kind of noise invalidates the Gaussian observation

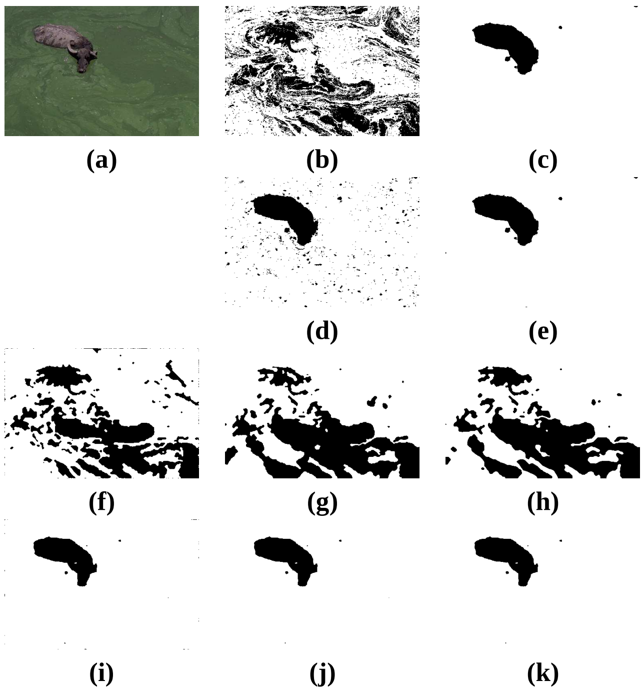

Fig. 7. Segmentation example of the water buffalo image with $K = 2$. (a) The original image. (b) K-means initialization. (c) The segmentation result of our approach with k-means initialization. (d) Initialization based on EM for independent mixtures. (e) The segmentation result of our approach with EM initialization. (f) The segmentation result of ICM with k-means initialization. (g) The segmentation result of SimF with k-means initialization. (h) The segmentation result of MeanF with k-means initialization. (i) The segmentation result of ICM with EM initialization. (j) The segmentation result of SimF with EM initialization. (k) The segmentation result of MeanF with EM initialization.

model assumption used by all methods discussed in the paper (since the true observation model is now a Gaussian mixture), and renders the observations spatially correlated. Thus, it presents a case where the true generative model of the image is different than the postulated one, which makes it an interesting experiment for real-world data. In Fig. 3(c) we show the noise corrupted image. In Fig. 3(d)-(g) we present the segmentation results of the EM, ICM, SimF and MeanF. Fig. 3(h) shows the segmentation obtained by our method. Clearly we see an advantage of our method over the other methods on this problem. We note that we initialized all algorithms in the same way.

In Fig. 2(b) and Fig. 3(b) we present the maximization progress of the penalized log-likelihood $\mathcal{F}$ for the simulated five-class image of Fig. 2 and the simulated five-class image of Fig. 3 respectively.

The experiments point out that in all methods the misclassification ratio increases as the amount of noise and the number of labels K increases. Our method performs much better than all other methods when moderate or high amount of noise is present, and it is competitive to other methods for low amounts of noise. Our method, implemented in Matlab, was faster than SimF and MeanF methods, which were implemented in C and optimized. In Fig. 2 we present the running time and the segmentation results of ICM, SimF and MeanF methods for the five-class image.

TABLE II
MISCLASSIFICATION RESULTS FOR THE FIVE-CLASS IMAGE.

	Noise σ					
	18	23	25	33	47	52
EM	15.1%	27.1%	28.2%	43.1%	53.8%	53.7%
HMRF-EM	-	0.2%	-	1.36%	7.68%	-
ICM	0.47%	0.94%	1.24%	4.44%	24.6%	31.7%
SimF	0.26%	0.46%	0.47%	0.92%	2.19%	2.88%
MeanF	0.23%	0.38%	0.49%	0.73%	2.41%	3.89%
SVFMM	6%	-	30%	-	-	42%
SVFMM-QD	4%	-	10%	-	-	28%
Our Approach	0.24%	0.39%	0.49%	0.73%	1.49%	1.78%
run time of our approach	19sec	23sec	27sec	40sec	76sec	92sec

B. *Choice of Parameters*

Our algorithm has in total three parameters, the number of components K, the priors parameter β, and the image filter used in (20). In this work we assume that the number of components K as well as the observation model family are given to us for a particular image. Also, in experiments not shown here we determined that 0.5 is a good choice for the parameter β. As already mentioned, it is a matter of future work to investigate ways to incorporate this parameter in the optimization process. In this section we will consider the only other free parameter in our algorithm which is the filter used for evaluating the mixtures π_{N_i}, p_{N_i}, q_{N_i} and s_{N_i} (see Eq. (20) and Eq. (35)).

Specifically, as mentioned in Section III the evaluation of such mixtures can be achieved by convolving with an image filter. In order for these mixtures to be valid distributions (without the need of extra normalization) the applied filter must have coefficients that sum to one. Our algorithm also requires that the center coefficient of the filter be zero. Also, we choose the filter to be symmetric. In the case where domain knowledge would imply some (simple) spatial relation between pixels with the same class label, this knowledge could be easily incorporated into our framework by employing a non symmetric filter that encourages these spatial relations.

Clearly the choice of filter can affect the performance of the algorithm and the quality of the segmentation. In addition the mixture calculation can be performed by a variety of images filters. In the section IV-A all reported results of our algorithm were obtained using a 'modified' (i.e. the center coefficient set to zero) pillbox filter with diameter δ equal to 5. We show the coefficients of this filter in Table III.

In Table IV we report the misclassification ratio results using different filters for the three class image (see Fig. 4(a)), after adding white Gaussian noise with $\sigma = 95$ (see Fig. 4(b)). We used 'modified' versions (i.e. the center coefficient is set to zero) of a low pass Gaussian filter and a mean filter. In Fig. 4(c) we show the corresponding segmentation results for Gaussian filters with $\sigma = 1$. The results illustrate that a larger filter size can compensate for the presence of high noise levels in the image. A closer look at Fig. 4(a),(c) reveals that large size filters, while providing increased robustness to noise, tend to oversmooth the edges. When the noise levels in an

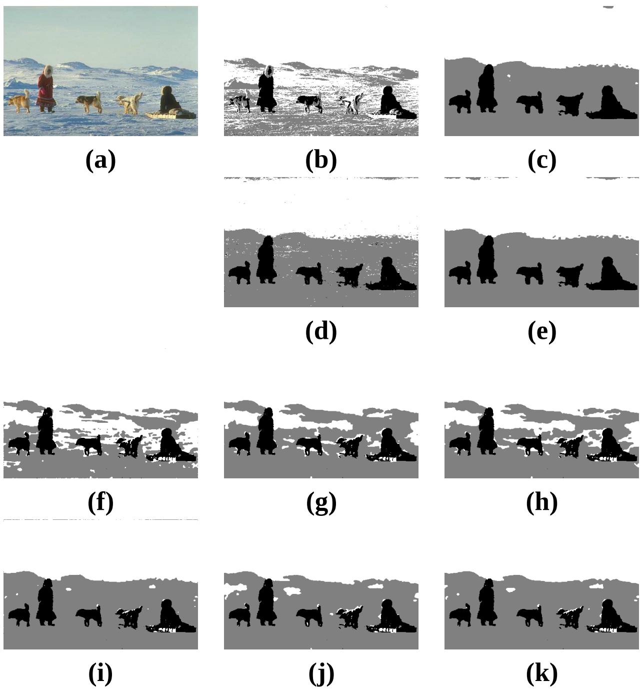

Fig. 8. Segmentation example of the dog sled image with $K = 3$. (a) The original image. (b) K-means initialization. (c) The segmentation result of our approach with k-means initialization. (d) Initialization based on EM for independent mixtures. (e) The segmentation result of our approach with EM initialization. (f) The segmentation result of ICM with k-means initialization. (g) The segmentation result of SimF with k-means initialization. (h) The segmentation result of MeanF with k-means initialization. (i) The segmentation result of ICM with EM initialization. (j) The segmentation result of SimF with EM initialization. (k) The segmentation result of MeanF with EM initialization.

TABLE III
COEFFICIENTS OF THE FILTER USED IN THE EXPERIMENTS.

0	0.0185	0.0414	0.0185	0
0.0185	0.0852	0.0865	0.0852	0.0185
0.0414	0.0865	0	0.0865	0.0414
0.0185	0.0852	0.0865	0.0852	0.0185
0	0.0185	0.0414	0.0185	0

TABLE IV
MISCLASSIFICATION RESULTS FOR FIG. 4(b) WITH DIFFERENT FILTERS.

Gaussian ($\sigma = 0.5$)	Gaussian ($\sigma = 1$)	mean 3×3	mean 5×5
1.19%	1.08%	1.23 %	1.16%

image are low there is no benefit from using large size filters, and the resulting oversmoothing of the edges just decreases the segmentation accuracy. Clearly, if the noise level of images in a particular domain is known a priori, then the filter type and its parameters can be fine-tuned for optimal performance.

C. Gray-Level Images

In natural images a number of difficult aspects of image segmentation come together, like noise and varying imaging conditions. Additionally, the true value of image labels K is not known. In the following experiments we use the same images as in [21], [30], [31] where K was estimated using (approximations of) the Bayesian Information Criterion (BIC). Fig. 5(a) shows a 128×128 Positron Emission Tomography (PET) image of a dog lung and Fig. 6(a) shows an aerial 100×100 image of a buoy against a background of dark water (see [30] for more details on their nature and origin).

For the PET image of a dog lung the image's labels K was estimated to be 3. In Fig. 5(c)-(f) we demonstrate the segmentation results of ICM, MeanF, SimF and our approach. For our method we used a 'modified' (i.e. the center coefficient set to zero) pillbox filter with diameter δ equal to 7. All other settings for all methods were the same as in Section IV-A. All algorithms were initialized by a simple thresholding of the image as shown in Fig. 5(b). The result shows that MeanF, SimF and our approach perform approximately the same and are producing more homogeneous regions for the lungs than the ICM algorithm.

For the buoy image the image's labels K was estimated to be 3. In Fig. 6(d)-(g) we demonstrate the segmentation results of ICM, MeanF, SimF and our approach. We used the same settings for all methods as in Section IV-A. All algorithms were initialized using the k-means based initialization which is shown in Fig. 6(b). It is clear that all algorithms given this particular initialization perform approximately the same and they are all capable of correctly assigning the pixels belonging to the buoy to one cluster. In Fig. 6(h)-(k) we show the segmentation results of the four methods using the same setting as before but employing an alternative initialization based on EM for independent mixtures. This initialization is shown in Fig. 6(c), where the horizontal scan lines from the imaging process of Fig. 6(a) can be clearly observed. Only our approach was able to correctly assign the pixels belonging to the buoy to one cluster given this EM based initialization.

The buoy experiment demonstrates a significant aspect of our approach, namely, that it is relatively insensitive to initialization compared to other methods. We feel that this is an important aspect, since in natural images, not only the true value of image labels K is hard to estimate, but also an appropriate initialization cannot be known a priori.

D. Color Images

In Fig. 7–8 we show the segmentation results of two different color images from the Berkeley Segmentation Dataset [32]. In these figures, we show the original image, the initializations obtained by K-means and EM, the results of our method with these two initializations, and the corresponding results of ICM, SimF and MeanF. In these examples our algorithm is able to consistently segment the original image independently of the initialization, whereas all three MRF-based methods are rather sensitive to the initial conditions.

In Fig. 9 we show the segmentation results of an image in rgb color space when the Gaussian observation model assumption is violated. Normalized rgb color (chromaticity) has been widely used by many researchers in the field of image segmentation, e.g. [33], [34], because of its important invariant properties. Specifically, it has been shown in [35] that, under the assumption of the dichromatic reflection model,

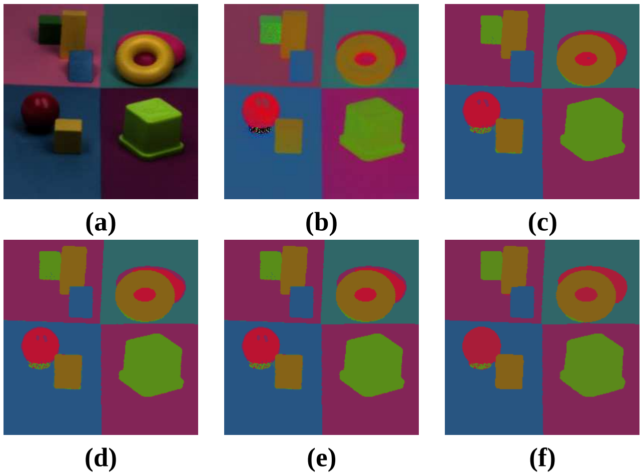

Fig. 9. Segmentation example of a color image with $K = 6$. (a) The original image. (b) The normalized *rgb* color image. (c) The segmentation result of ICM in *rgb* color space. (d) The segmentation result of SimF in *rgb* color space. (e) The segmentation result of MeanF in *rgb* color space. (f) The segmentation result of our approach in *rgb* color space.

normalized color is to a large extent invariant to a change in camera viewpoint, object pose, and the direction and intensity of the incident light. In addition, the color transformation from RGB to normalized color rgb is simple and easy to compute while does not necessitate extra color-reduction steps as in [36]. Namely, based on the measured RGB-values, normalized color rgb is computed as follows:

$$r = R/(R+G+B) \quad (42)$$

$$g = G/(R+G+B) \quad (43)$$

$$b = B/(R+G+B) \quad (44)$$

However, this (non-linear) color transformation has a serious drawback, as it becomes unstable for some RGB sensor values. Particularly, rgb is undefined at the black point ($R = G = B = 0$) and is unstable near this singular point. As a consequence, a small perturbation for dark (low-intensity) sensor values (e.g. due to sensor noise), will cause a significant jump in the transformed values.

In Fig. 9(b), which is the normalized rgb color transformation of Fig. 9(a), the unstable invariant values are clearly visible at the bottom of the red ball. The presence of these unstable invariant values violate the assumed Gaussian observation model. Fig. 9(c)-(e) show the segmentation results for $K = 6$ for the normalized color image of Fig. 9(b) with ICM, SimF and MeanF, respectively. In Fig. 9(f) we show the corresponding results with our approach. All figures show a reconstruction of the normalized rgb image, with pixel values in the final result corresponding to the estimated mean of their assigned Gaussian component. The results show that our approach produces more homogeneous regions than other algorithms and consequently a better segmentation.

V. Conclusions

We proposed a graphical model and a novel EM algorithm for Markov-based image segmentation. The proposed model postulates that the unobserved pixel labels are generated by prior distributions that have similar parameters for neighboring pixels. The proposed EM algorithm performs iterative bound optimization of a penalized log-likelihood of this model. The derived EM equations are similar to the standard (unconstrained) EM algorithm, with the only difference that a ‘smoothing’ step is interleaved between the E- and the M-step, that couples the posteriors of neighboring pixels in each iteration. Compared to other MRF-based algorithms for segmentation, we note that our algorithm enjoys a simple implementation and demonstrates competitive performance in terms of speed and solution quality.

Acknowledgments

The authors are thankful to K. Blekas, A. T. Cemgil, N. Galatsanos, I. Patras and J. J. Verbeek for their valuable comments. The authors are grateful to N. Peyrard for making publicly available the implementation of MeanF, SimF and ICM methods. This work is an IOP project sponsored by the Dutch Ministry of Economic Affairs, project number IBV99001.

References

[1] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[2] J. Besag, “On the statistical analysis of dirty pictures,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.

[3] R. Chellappa and A. Jain, *Markov Random Fields: Theory and Application.* Boston: Academic Press, 1993.

[4] S. Z. Li, *Markov Random Field Modeling in Computer Vision.* London, UK: Springer-Verlag, 2001.

[5] M. V. Ibanez and A. Simo, “Parameter estimation in Markov random field image modeling with imperfect observations: a comparative study,” *Pattern Recogn. Lett.*, vol. 24, no. 14, pp. 2377–2389, 2003.

[6] D. Geman and B. Gidas, *Image analysis and computer vision.* National Academy Press, 1991, ch. 2, pp. 9–36.

[7] J.-M. Laferte, P. Perez, and F. Heitz, “Discrete Markov image modeling and inference on the quadtree,” *IEEE Trans. on Image Processing*, vol. 9, no. 3, pp. 390 – 404, May 2000.

[8] J.-N. Provost, C. Collet, P. Rostaing, P. Perez, and P. Bouthemy, “Hierarchical Markovian segmentation of multispectral images for the reconstruction of water depth maps,” *Comput. Vis. Image Underst.*, vol. 93, no. 2, pp. 155–174, 2004.

[9] R. Fjortoft, Y. Delignon, W. Pieczynski, M. Sigelle, and F. Tupin, “Unsupervised classification of radar images using hidden Markov chains and hidden Markov random fields,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 3, pp. 675– 686, Mar. 2003.

[10] M. Robinson, M. Azimi-Sadjadi, and J. Salazar, “A temporally adaptive classifier for multispectral imagery,” *IEEE Trans. on Neural Netw.*, vol. 15, no. 1, pp. 159–165, 2004.

[11] J. L. Marroquin, E. A. Santana, and S. Botello, “Hidden Markov measure field models for image segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1380–1387, 2003.

[12] S. C. Zhu, Y. Wu, and D. Mumford, “Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling,” *Intern. Journal of Computer Vision*, vol. 27, no. 2, pp. 107–126, 1998.

[13] G. McLachlan and D. Peel, *Finite Mixture Models.* John Wiley & Sons Inc, November 2000.

[14] C. Carson, S. Belongie, H. Greenspan, and J. Malik, “Blobworld: Image segmentation using Expectation-Maximization and its application to image querying,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1026–1038, 2002.

[15] A. Diplaros, T. Gevers, and N. Vlassis, “Skin detection using the EM algorithm with spatial constraints,” in *IEEE International Conference on Systems, Man and Cybernetics*, The Hague, Netherlands, 2004.

[16] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[17] R. Neal and G. Hinton, “A view of the EM algorithm that justifies incremental, sparse, and other variants,” in *Learning in Graphical Models*, M. Jordan, Ed. Kluwer Academic Publishers, 1998, pp. 355–368.

[18] W. Qian and D. M. Titterington, “Estimation of parameters in hidden Markov models,” *Phil. Trans. R. Soc. A*, vol. 337, pp. 407–428, 1991.

[19] J. Zhang, “The mean field theory in EM procedures for Markov random fields,” *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2570–2583, 1992.

[20] 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. on Medical Imaging*, vol. 20, no. 1, pp. 45–57, January 2001.

[21] F. Forbes and N. Peyrard, “Hidden Markov random field model selection criteria based on mean field-like approximations,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1089–1101, 2003.

[22] J. Sun and D. Gu, “Bayesian image segmentation based on an inhomogeneous hidden markov random field,” in *17th International Conference on Pattern Recognition (ICPR 2004)*, vol. 1, 2004, pp. 596–599.

[23] G. Celeux, F. Forbes, and N. Peyrard, “EM procedures using mean field-like approximations for Markov model-based image segmentation,” *Pattern Recognition*, vol. 36, no. 1, pp. 131–144, 2003.

[24] J. Besag, “Statistical analysis of non-lattice data,” *The Statistician*, vol. 24, no. 3, pp. 179–195, 1975.

[25] S. Sanjay-Gopal and T. J. Hebert, “Bayesian pixel classification using spatially variant finite mixtures and the generalized EM algorithm,” *IEEE Trans. on Image Processing*, vol. 7, no. 7, pp. 1014–1028, 1998.

[26] K. Blekas, A. Likas, N. Galatsanos, and I. E. Lagaris, “A spatially-constrained mixture model for image segmentation,” *IEEE Trans. on Neural Netw.*, vol. 16, no. 2, pp. 494–498, 2005.

[27] S. T. Roweis, L. K. Saul, and G. E. Hinton, “Global coordination of local linear models,” in *Advances in Neural Information Processing Systems 14 (NIPS’01)*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 889–896.

[28] J. J. Verbeek, N. Vlassis, and B. Kröse, “Self-organizing mixture models,” *Neurocomputing*, vol. 63, pp. 99–123, 2005.

[29] R. Xu and D. Wunsch II, “Survey of clustering algorithms,” *IEEE Trans. on Neural Netw.*, vol. 16, no. 3, pp. 645 – 678, 2005.

[30] D. C. Stanford, “Fast automatic unsupervised image segmentation and curve detection in spatial point patterns,” Ph.D. dissertation, University of Washington Statistics Department, 1999.

[31] D. C. Stanford and A. E. Raftery, “Approximate Bayes Factors for image segmentation: The Pseudolikelihood Information Criterion (PLIC),” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 11, pp. 1517–1520, 2002.

[32] 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 *Eighth IEEE International Conference on Computer Vision*, vol. 2, July 2001, pp. 416–423.

[33] G. Healey, “Segmenting images using normalized color,” *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 22, no. 1, pp. 64–73, 1992.

[34] T. Gevers, “Adaptive image segmentation by combining photometric invariant region and edge information,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 848–852, 2002.

[35] T. Gevers and A. W. M. Smeulders, “Color-based object recognition,” *Pattern Recognition*, vol. 32, no. 3, pp. 453–464, 1999.

[36] G. Dong and M. Xie, “Color clustering and learning for image segmentation based on neural networks,” *IEEE Trans. on Neural Netw.*, vol. 16, no. 4, pp. 925–936, 2005.