

Factorial Markov Random Fields

Junhwan Kim and Ramin Zabih

Computer Science Department
Cornell University
Ithaca, NY 14853

Abstract. In this paper we propose an extension to the standard Markov Random Field (MRF) model in order to handle layers. Our extension, which we call a Factorial MRF (FMRF), is analogous to the extension from Hidden Markov Models (HMM's) to Factorial HMM's. We present an efficient EM-based algorithm for inference on Factorial MRF's. Our algorithm makes use of the fact that layers are a priori independent, and that layers only interact through the observable image. The algorithm iterates between wide inference, i.e., inference within each layer for the entire set of pixels, and deep inference, i.e., inference through the layers for each single pixel. The efficiency of our method is partly due to the use of graph cuts for binary segmentation, which is part of the wide inference step. We show experimental results for both real and synthetic images.

Keywords: Grouping and segmentation, Layer representation, Graphical model, Bayesian inference, Markov Random Field, Factorial Hidden Markov Model

1 Introduction

Markov Random Fields (MRF's) have been extensively used in low level vision because they can naturally incorporate the spatial coherence of measures of interest (intensity, disparity, etc) [12]. However, MRF's cannot effectively combine information over disconnected spatial regions. Layer representations are a popular way of addressing this limitation [16,14,1].

The main contribution of this paper is to propose a new graphical model that can represent image layers, and to develop an efficient algorithm for inference on this graphical model. We extend the standard MRF model to several layers of MRF's, which is analogous to the extension from Hidden Markov Models (HMM's) to Factorial HMM's (see Figure (1) and Figure (2)) [6]. A Factorial HMM has the structure shown in Figure (1), and consists of i) a set of hidden variables, which are a priori independent, and ii) a set of observable variables, whose state depends on the hidden variables as shown in the figure. The inference algorithm for Factorial HMM's iterates between (exact) inference within each hidden variable (via the forward-backward algorithm), and (approximate) inference using indirect dependencies between hidden variables through the observables. Similarly, our algorithm alternates between wide inference and deep

inference. Wide inference, i.e., inference within each layer for the entire set of image pixels, utilizes what we call *pseudo-observables*, which we define in section 4.2. These play the same role as observables in standard MRF's. Parallel to the forward-backward algorithm in HMM's, we use graph cuts for binary segmentation [7], where the binary value signifies whether the object is present or not at the given pixel and layer. Deep inference, on the other hand, is inference through the layers for each single pixel. We develop an efficient EM-based algorithm to evaluate pseudo-observables by deep inference. The algorithm uses dependencies between observables and hidden variables (but not within the hidden variables), as well as the result of the binary segmentation from the wide inference step.

The rest of the paper is organized as follows. Section 2 summarizes related work. In section 3, we define a Factorial MRF and show how this model can be used for layers. Our EM-based algorithm for inference on Factorial MRF's is presented in section 4. We demonstrate the effectiveness of our algorithm by experimental results for both real and synthetic images in section 5. Some technical details, especially regarding transparent layers, are deferred to the appendix.

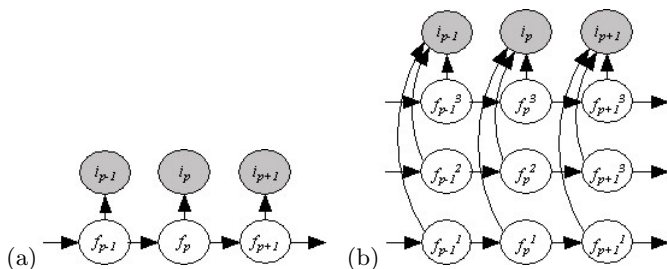


Fig. 1. Extension from HMM to factorial HMM: (a) A directed graph specifying conditional dependence relations for an HMM. (b) A directed graph for factorial HMM with three underlying Markov chains (borrowed from [6]). From now on, a gray node represents an observable variable, whereas a white node represents a hidden variable.

2 Related Work

Inference problems on probabilistic models are frequently encountered in computer vision and image processing. In the structured variational approximation [10], exact algorithms for probability computation on tractable substructures are combined with variational methods to handle the interaction between the substructures which make the system as a whole intractable. Factorial HMM's [6] are a natural extension of HMM's, in which the hidden state is given by the joint configuration of a set of independent variables. In this case, the natural tractable structure consists of the HMM's for each hidden variable, for which the forward-backward algorithm can be used. On the other hand, in the case of

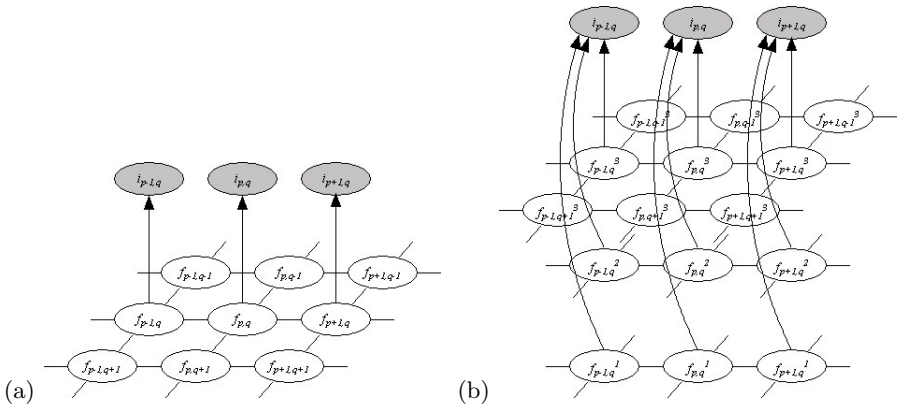


Fig. 2. Extension from MRF to factorial MRF: (a) A graph specifying conditional dependence relations for an MRF. (b) A graph for a factorial MRF with three underlying MRF's. Some of the nodes and links are omitted from the drawing for legibility

the Hidden Markov decision trees, there are two natural tractable substructures, the “forest of chains approximation”, and the “forest of trees approximation” [10]. Transformed HMM's [9] can be considered as Factorial HMM's with two hidden variables, i.e., transformation variable and class variable. They use the model to cluster unlabeled video segments and form a video summary in an unsupervised fashion. Besides the Factorial HMM, researchers have proposed other extensions such as Coupled HMM's for Chinese martial art action recognition [3], or Parallel HMM's for American sign language recognition [15].

Layer representations are known to be able to precisely segment and estimate motion for multiple objects, and to provide compact and comprehensive representations. Wang and Adelson [16] approached the problem by iteratively clustering motion models computed using optical flow. Each layer, ordered in depth, contains an intensity map and alpha map, and they occlude each other with image compositing rules. Ayer and Sawhney [1] proposed an EM algorithm for robust maximum-likelihood estimation of the multiple models and their layers of support. They also applied the minimum descriptive length principle to estimate the number of models. Weiss [17] presented an EM algorithm that can segment image sequences by fitting multiple smooth flow fields to the spatiotemporal data. He showed how to estimate a single smooth flow field, which eventually leads to the multiple model estimation. The number of layers is estimated automatically using similar methods to the parametric approach. Torr et al. [14] concentrated on 3D layers, consists of approximately planar layers that have arbitrary 3D positions and orientations. 3D layer representations can naturally handle parallax effects on the layer as opposed to 2D approaches. Frey [5] recently proposed a Bayesian network for appearance-based layered vision which describes the occlusion process, and developed iterative probability propagation to recover the identity and position of the objects in the scene.

3 Problem Formulation

The observable image, $i = \{i_p \mid p \in \mathcal{P}, \mathcal{P} \text{ is set of pixels}\}$, results from several layers of objects. Let f_p^l be a binary random variable which is 1 if an object exists at pixel p in layer l , and 0 otherwise. We assume that the layers are a priori independent of each other and we model each layer as Markov Random Field whose clique potentials involve pairs of neighboring pixels. This is the *Ising Model* [12],

$$P(f) = \prod_l P(f^l) \tag{1}$$

$$P(f^l) \propto \exp\left(-\sum_p \sum_{q \in N_p} 2\theta_{\{p,q\}}^l \delta(f_p^l \neq f_q^l)\right). \tag{2}$$

Here f^l indicates $f^l = \{f_p^l \mid p \in \mathcal{P}\}$, and $N = \{N_p \mid p \in \mathcal{P}\}$ denotes a neighborhood system, where $N_p \subset \mathcal{P}$. $\theta_{\{p,q\}}^l$ is a coefficient roughly signifying the degree of connection between two pixels p and q in layer l (we will describe this in more detail later). Given the configuration of layers, the likelihood of the observable image at each single pixel is independent from other pixels:

$$P(i \mid f) = \prod_p P(i_p \mid f_p), \tag{3}$$

where f_p indicates $f_p = \{f_p^l \mid l \in \mathcal{L}, \mathcal{L} \text{ is set of layers}\}$. By convention we will number the layers so that the largest numbered layer is closest to the viewer. By equations (1) and (3), we have:

$$P(i, f) = \prod_p P(i_p \mid f_p) \prod_l P(f^l) \tag{4}$$

The likelihood of the observable, $P(i_p \mid f_p)$, hinges on the characteristic of layers.

We will consider two kinds of layers, either transparent or opaque. For opaque layers, the observable image at each single pixel comes from the image containing an object that is closest to the viewer, as usual with layers. Let $\mathcal{N}(i; \mu, C^{-1})$ denote a gaussian distribution with mean μ and covariance C ; we have

$$\begin{aligned} P(i_p \mid f_p) &= \mathcal{N}(i_p; F(f_p), C(f_p)^{-1}) \\ F(f_p) &= \sum_l W^l M_p^l \\ C(f_p)^{-1} &= \sum_l C^{l-1} M_p^l, \end{aligned}$$

where W^l denotes the attributes of the object at layer l , which can be intensity, displacement, etc, and C^l is their covariance. M_p^l is a binary random variable which is 1 if at pixel p there is an object present (i.e., $f_p^l = 1$) and this object

is not occluded (i.e., for all $l' > l, f_p^{l'} = 0$) (see Figure(2)). For transparent layers, on the other hand, the observable image at each pixel results from a linear combination of the objects that present at each layer, contaminated by gaussian noise:

$$P(i_p | f_p) = \mathcal{N}(i_p; F(f_p), C^{-1})$$

$$F(f_p) = \sum_l W^l f_p^l$$

From now on however, for the sake of readability, we will proceed with opaque layers. In the appendix we derive the formulas for the transparent layers.

4 EM Algorithm

Given the probabilistic model, the inference problem is to compute the posterior probabilities of the hidden variables (objects in each layers, f), given the observations (image, i). In some other cases, we may want to infer the single most probable state of the hidden variables. The parameters of a factorial MRF can be estimated via the Expectation Maximization algorithm [4], which iterates between assuming the current parameters to compute posterior probabilities over the hidden states (E-step), and using these probabilities to maximize the expected log likelihood of the parameters (M-step). The EM algorithm starts by defining the expected log likelihood of complete data:

$$\mathcal{Q}(\phi^{new} | \phi) = E\{\log P(i, f | \phi^{new}) | \phi, i\} \tag{5}$$

For the factorial MRF model, the parameters are $\phi = \{W^l, C^l, \theta_{\{p,q\}}^l\}$. For the sake of simplicity however, we will proceed with setting $\theta_{\{p,q\}}^l = const.$

4.1 M-Step

The M-Step for each parameter is obtained by setting the derivatives of \mathcal{Q} with respect to those each parameters to zero, and solving. For the opaque layers, we have:

$$W^{l,new} = \frac{\sum_p i_p \overline{M_p^l}}{\sum_p \overline{M_p^l}}$$

$$C^{l,new} = \sum_p i_p i_p' \overline{M_p^l} - \sum_{p,l} W^l \overline{M_p^l} i_p'$$

Note that, for $L = 1$, these equations reduce to the standard single-layered MRF with similar observable. (See the appendix for the derivation.)

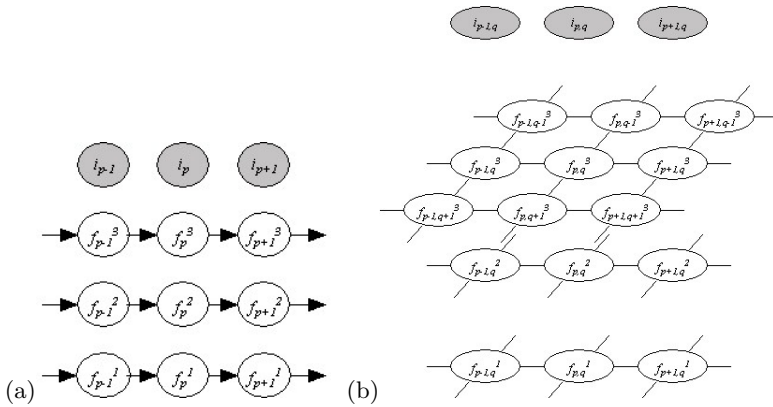


Fig. 3. Structural approximation: (a)Hidden variables retain their Markov structure within each chain, but are independent across chains [6]. (b)Hidden variables retain their Markov structure within each layer, but are independent across layers.

4.2 E Step

As in the factorial HMM [6], the factorial MRF is approximated by L uncoupled MRF's (see Figure(3)). We approximate the posterior distribution P by a tractable distribution Q . We write the structured approximation as:

$$Q(i) = \prod_p \left[\prod_l h_p^l \right] \prod_l P(f^l) \tag{6}$$

Comparing to Eq(4), we note that h_p^l has taken the place of $P(i_p | f_p^l)$, which h_p^l is expected to approximate. From now on, we will call h_p^l a *pseudo-observable*, because it plays a similar role to observables in the standard MRF. We can safely regard h_p^l as the likelihood of f_p^l , i.e., $h_p^l = 0.2$ implies that the likelihood of $f_p^l = 1$ (present) is 0.2, and that of $f_p^l = 0$ (absent) is 0.8. We describe how to evaluate h_p^l in the next subsection, which leads to calculation of each variable's maximum *a posteriori* (MAP) estimates (f^*), which in turn leads to evaluation of each variable's expectation (\bar{f}).

Pseudo-observable: To evaluate the pseudo-observable (h_p^l) for the entire set of layers, we calculate h_p^l for each layer under the assumption that the probability for f_p^l for other layers are given. In that case, we can evaluate the effect on the observable by setting f_p^l of the desired layer. Thus we have:

$$\begin{aligned} h_p^l &= P(i_p | f_p^l) = \sum_{l'} P(i_p | M_p^{l'}, f_p^l) P(M_p^{l'} | f_p^l) \\ &= \sum_{l'} P(i_p | M_p^{l'}) P(M_p^{l'} | f_p^l), \end{aligned}$$

where

$$P(M_p^{l'} | f_p^l) = \begin{cases} \prod_{l''=l'}^L (1 - \overline{f_p^{l''}}) & \text{if } l' \geq l \\ 0 & \text{otherwise} \end{cases}$$

MAP Estimate: The above fixed point equations for h_p^l requires the evaluation of $\overline{f_p^l}$ in terms of h_p^l . Unlike the case of factorial HMM’s, where there is a tractable forward-backward algorithm, we don’t have a tractable algorithm to evaluate $\overline{f_p^l}$ from h_p^l . This is because the inference in even a single MRF is intractable unless the joint distribution is also Gaussian, in which case, an analytic solution is available. Thus we use an algorithm using the MAP estimates, which can be rapidly obtained via graph cuts [7]. Parallel to the case of factorial HMM’s, we find the MAP estimates considering the h_p^l as the observable. In our experiments, we used the graph cut algorithm of [2], which is specifically designed for the kind of graphs that arise in computer vision problems.

Expectation: Given the MAP estimates f^* obtained above, we can approximate \overline{f} as:

$$\overline{f_p^l} = \sum_{f_p^l} f_p^l P(f_p^l | i) \simeq \sum_{f_p^l} f_p^l \delta(f_p^l = f_p^{l*}) = f_p^{l*},$$

where we assume that $P(f_p^l | i) \simeq \delta(f_p^l = f_p^{l*})$. Unfortunately, this is too crude an approximation (it is basically binary thresholding). We can make a better approximation using the MRF priors:

$$\begin{aligned} \overline{f_p^l} &= \sum_{f_p^l} f_p^l P(f_p^l | i) = \sum_{f_p^l} f_p^l \sum_{f_{N_p}^l} P(f_p^l | f_{N_p}^l) P(f_{N_p}^l | i) \\ &\simeq \sum_{f_p^l} f_p^l \sum_{f_{N_p}^l} P(f_p^l | f_{N_p}^l) \delta(f_{N_p}^l = f_{N_p}^{l*}) = \sum_{f_p^l} f_p^l P(f_p^l | f_{N_p}^{l*}), \end{aligned}$$

where we assume that $P(f_{N_p}^l | i) \simeq \delta(f_{N_p}^l = f_{N_p}^{l*})$. It is possible relax the crudeness of the assumption by setting $P(f_{N_{N_p}}^l | i) \simeq \delta(f_{N_{N_p}}^l = f_{N_{N_p}}^{l*})$, or $P(f_{N_{N_{N_p}}}^l | i) \simeq \delta(f_{N_{N_{N_p}}}^l = f_{N_{N_{N_p}}}^{l*})$, but there is a price to pay in terms of computational load. However, we find that the above 1st-order neighbor approximation suffices.

To summarize, we need the pseudo-observable (h) in order to evaluate the expectation (f), whereas we need the expectation (\overline{f}) in order to evaluate the pseudo-observable (h). However the evaluation is done indirectly through the MAP estimates (f^*). Thus the E step eventually consists of iteration through above three steps until convergence. Algorithm 1 shows the overall algorithm.

Algorithm 1 Overall algorithm

-
- 1: Initialize parameters and hidden variables randomly
 - 2: **repeat**
 - 3: Update W^l, C^l
 - 4: **repeat**
 - 5: Calculate pseudo-observable (h) from \bar{f}
 - 6: Calculate MAP estimates (f^*) from h by graph cuts for binary segmentation
 - 7: Calculate expectation (\bar{f}) from f^* by 1st order neighbor approximation and MRF prior
 - 8: **until** convergence
 - 9: **until** convergence
-

Convergence: One of the underlying assumptions of the theory of the EM algorithm is the use of an exact E step [4]. The exact EM algorithm maximizes the log likelihood with respect to the posterior probability distribution over the hidden variables given the parameters. The structural approximation algorithm, on the other hand, does the same job with the additional constraint that the posterior probability distribution over the hidden variable is of a particular tractable form [6], such as Q in Eq(5) for instance. The convergence argument for our approach is slightly more complicated than the standard structural approximation algorithm. In our case, the E step is not exact even in a single layer MRF, as opposed to the exact E step in a single HMM chain used in factorial HMM's. We can use Monte Carlo sampling methods using Markov chain for each MRF layer [8], which offers the theoretical assurance that the sampling procedure will converge to the correct posterior distribution *ultimately*. It is not a particularly attractive approach though, since inference on a single MRF layer is not the overall goal of our algorithm, but a subroutine. Although we do not have a formal justification of convergence per se (which remains for future work), the experimental results offer strong evidence for convergence property of our approach.

Table 1. Two different decomposition: Depending on the initial guess on the intensity in M-Step, the algorithm can end up with two different decompositions.































































input		
	result 1	result 2
layer 1		
layer 2		

Table 2. E-steps: The bottom rows show the final results after convergence. (a)box-box synthetic image (b)box-box-grid syntehtic image (c)face-box-box synthetic image with appearance model for the face

input		(a) 			(b) 			(c) 		
iter		layer 1	layer 2	layer 3	layer 1	layer 2	layer 3	layer 1	layer 2	layer 3
0	\bar{f}									
1	h									
1	f^*									
1	\bar{f}									
2	h									
2	f^*									

5 Experiments

5.1 Synthetic Image

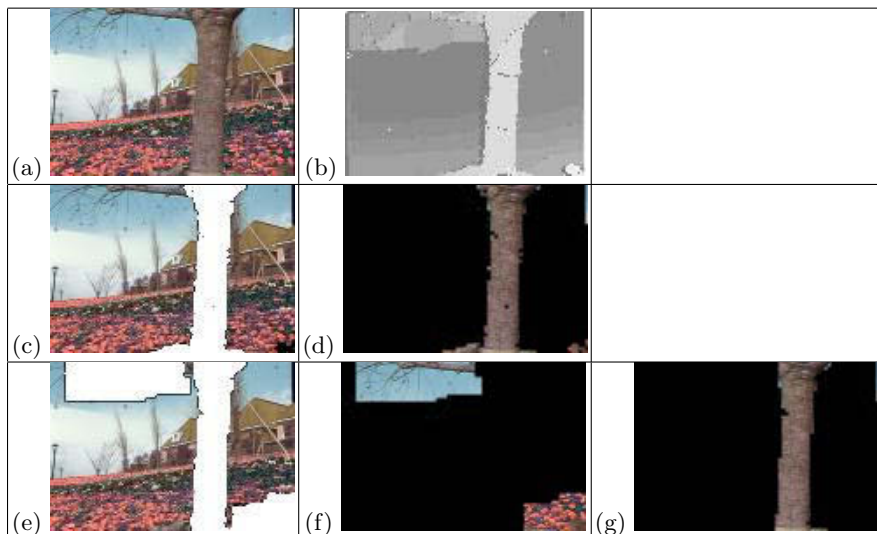
We did experiments for a couple of synthetic images, shown in (a) and (b) of Table (2). Without any prior knowledge, depending on the initial guess on the intensity in M-Step, the algorithm can end up with two different decompositions (see Table (1)). In this case, result1 is more “natural”, but without prior knowledge, result2 is not terrible either. The question of how to incorporate some bias toward more natural decomposition (i.e. the closer, the brighter) remains for future work. Also, it is relatively straightforward to incorporate the appearance model to our probabilistic formulation, as shown in (c) of Table (2).

5.2 Real Image

We used the disparity map of the Garden-flower images and Tsukuba stereo images from a recent algorithm for computing motion and stereo with occlusions [11] as shown in Table(3) and Table(4) respectively. Notice that the result depends on the number of layers (which user must specify) as shown in Table(3). Although both of them give reasonable layer decomposition, the result for three layers give

a better result, i.e., the final composition is closer to the input. Our algorithm deals with only layer occupancy, but not layer texture as shown in Table(3) (c) and (e), and Table(4) (d) and (f). We can either (1) incorporate texture in our formulation, or (2) use an image inpainting method such as [13].

Table 3. Layer analysis result on the flower garden sequence. (a) Left image. (b) Disparity map adopted from [11]. We run our algorithm on this disparity map, not on the original image. (c),(d) Extracted layers for two layer decomposition. (e),(f),(g) Extracted layers for three layer decomposition. Depending on the number of layers that user specifies, the algorithm can end up with two different decompositions.

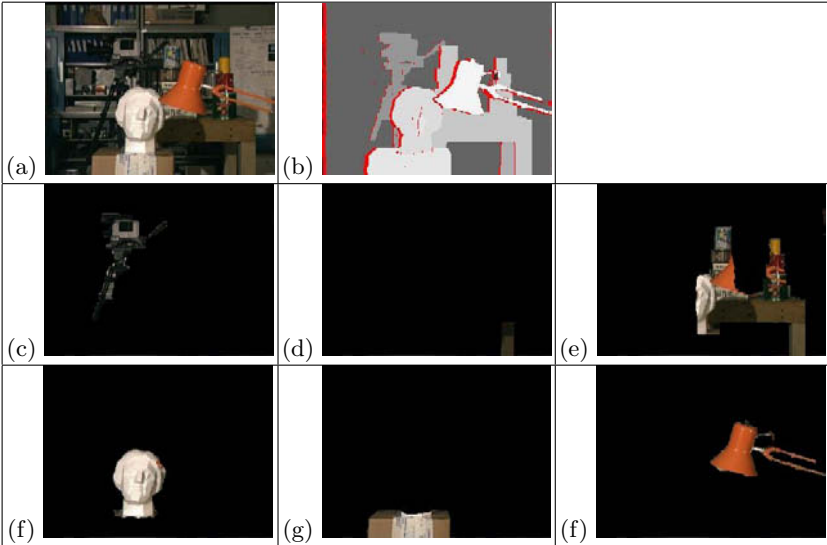


6 Conclusion

We have proposed an extension to the standard Markov Random Field (MRF) model suitable for layer representation. This new graphical model is a natural model of the image composition process of several objects. We also presented an efficient EM-based algorithm for inference. Our inference algorithm makes use of the fact that each MRF layer is natural substructure of the whole factorial MRF. Although each MRF layer is intractable in a strict sense, we developed an efficient inference based on the graph-cuts algorithm for binary segmentation. As the Table(3), Table(4) shows, our algorithm can decompose an image composed of several layers in a reasonable way, as long as user specifies appropriate number of layers, which makes automatic determination of number of layers a natural extension of this work.

Acknowledgments. We wish to thank Yuri Boykov and Vladimir Kolmogorov for the use of their software, and Rick Szeliski for providing us with imagery. This research was supported by NSF grants IIS-9900115 and CCR-0113371, and by a grant from Microsoft Research.

Table 4. Layer extraction result on Tsukuba dataset : (a) Left image, (b) Disparity map adopted from [11]. (c) First layer. (d) Second layer. (e) Third layer. Left part demonstrates an example where our algorithm gives random arbitrary result for completely occluded region without any priori knowledge. (f) Fourth layer. Notice a small area with unrecovered texture because of layer 1 occluding it. (g) Fifth layer. (h) Sixth layer. We don't show here the background layer, which covers all over the image



7 Appendix

7.1 M Step

Opaque layer: We start by expanding \mathcal{Q} :

$$\begin{aligned} \mathcal{Q} = & -1/2 \sum_p [i_p' \overline{C(f_p)^{-1}} i_p - 2i_p' \overline{C(f_p)^{-1}} F(f_p) \\ & + \overline{F(f_p)' C(f_p)^{-1} F(f_p)}] + \sum_l \sum_{p,q \in \varepsilon_N} -2\theta_{p,q \in \varepsilon_N} \overline{\delta(f_p^l \neq f_{ql})} - \log Z \quad (7) \end{aligned}$$

From the model of opaque layers we have:

$$\begin{aligned}
 P(i_p | f_p) &= \mathcal{N}(i_p; F(f_p), C(f_p)^{-1}) \\
 F(f_p) &= \sum_l W^l M_p^l \\
 C(f_p)^{-1} &= \sum_l C^{l-1} M_p^l \\
 M_p^l &= f_p^l \prod_{l'=l+1}^L (1 - f_p^{l'})
 \end{aligned}$$

The average values are given by:

$$\begin{aligned}
 \overline{C(f_p)^{-1}} &= \sum_l C^{l-1} \overline{M_p^l} \\
 \overline{C(f_p)^{-1} F(f_p)} &= \sum_l C^{l-1} W^l \overline{M_p^l} \\
 \overline{F(f_p)' C(f_p)^{-1} F(f_p)} &= \sum_l \text{tr}\{W^{l'} C^{l-1} W^l \text{diag}\{\overline{M_p^l}\}\}
 \end{aligned}$$

By taking derivative of \mathcal{Q} with respect to the parameters, we get the parameter estimation equations:

$$\begin{aligned}
 \frac{\partial \mathcal{Q}}{\partial W^l} &= \sum_p C^{l-1} i_p \overline{M_p^l} - \sum_p C^{l-1} W^l \overline{M_p^l} = 0 \rightarrow W^{l \text{new}} = \frac{\sum_p i_p \overline{M_p^l}}{\sum_p \overline{M_p^l}} \\
 \frac{\partial \mathcal{Q}}{\partial C^{l-1}} &= -1/2 \sum_p i_p i_p' \overline{M_p^l} + \sum_p W^l \overline{M_p^l} i_p' - 1/2 \sum_p W^l W^{l'} \overline{M_p^l} = 0 \\
 \rightarrow C_l^{\text{new}} &= \sum_p i_p i_p' \overline{M_p^l} - \sum_{p,l} W^l \overline{M_p^l} i_p'
 \end{aligned}$$

Transparent layer: Similar to the case of opaque layers, we expand \mathcal{Q} :

$$\begin{aligned}
 \mathcal{Q} &= -1/2 \sum_p [i_p' C^{-1} i_p - 2i_p' C^{-1} \overline{F(f_p)} \\
 &\quad + \overline{F(f_p)' C^{-1} F(f_p)}] + \sum_l \sum_{p,q \in \varepsilon_N} -2\theta_{p,q \in \varepsilon_N} \overline{\delta(f_p^l \neq f_{ql})} - \log Z \quad (8)
 \end{aligned}$$

From the model of transparent layers we have $P(i_p | f_p) = \mathcal{N}(i_p; F(f_p), C^{-1})$, where $F(f_p) = \sum_l W^l f_p^l$.

The average values are given by:

$$\begin{aligned}
 \overline{F(f_p)} &= \sum_l W^l \overline{f_p^l} \\
 \overline{F(f_p)' C^{-1} F(f_p)} &= \overline{\left(\sum_l W^l f_p^l\right)' C^{-1} \left(\sum_l W^l f_p^l\right)} = \sum_{l'} \sum_l \text{tr}\{W^{l'} C^{-1} W^l \overline{f_p^l f_p^{l'}}\}
 \end{aligned}$$

By taking derivative of Q with respect to the parameters, we get the parameter estimation equations:

$$\begin{aligned} \frac{\partial Q}{\partial W^l} &= \sum_p \left[\sum_{l'} W^{l'} \overline{f_p^{l'} f_p^{l'}} - i_p \overline{f_p^{l'}} \right] = 0 \rightarrow W^{new} = \left(\sum_p i_p \overline{f_p} \right) \left(\sum_p \overline{f_p f_p'} \right)^+ \\ \frac{\partial Q}{\partial C^{-1}} &= \frac{C}{2} + \sum_p \left[\sum_{l'} i_p \overline{f_p^{l'}} W^{l'} - 1/2 i_p i_{p'} - 1/2 \sum_{l'} \sum_l W^l \overline{f_p^l f_p^{l'}} W^{l'} \right] = 0 \\ \rightarrow C^{new} &= \sum_p i_p i_{p'} - \sum_{p,l} W^l \overline{f_p^l i_{p'}} \end{aligned}$$

References

1. S. Ayer and H. Sawhney. Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding. In *Proceedings of the International Conference on Computer Vision*, pages 777–784, 1995.
2. Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In *Proceedings of the International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 2134 of *LNCS*, pages 359–374, 2001.
3. M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. Technical Report 407, Vision and Modeling Group, MIT Media lab, November 1996.
4. N. M. Dempster, A.P. Laird and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:185–197, 1977.
5. B.J. Frey. Filling in scenes by propagating probabilities through layers and into appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages I:185–192, 2000.
6. Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 472–478. The MIT Press, 1996.
7. D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. In *J. R. Statist. Soc. B*, 51(2):271–279, 1989.
8. W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
9. N. Jovic, N. Petrovic, B. Frey, and T. Huang. Transformed hidden markov models: Estimating mixture models of images and inferring spatial transformations in video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2000.
10. Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. An introduction to variational methods for graphical models. In M.I. Jordan (Ed.), *Learning in Graphical Models*, MIT Press, 1999.
11. Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the International Conference on Computer Vision*, pages 508–515, 2001.
12. S. Z. Li. *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, Tokyo, 1995.

13. V. Caselles M. Bertalmio, G. Sapiro and C. Ballester. Image inpainting. In *SIG-GRAPH*, pages 417–424, 2000.
14. P.H.S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. *PAMI*, 23(3):297–303, March 2001.
15. C. Vogler and D. Metaxas. Parallel Hidden Markov Models for American Sign Language recognition. In *Proceedings of the International Conference on Computer Vision*, pages 116–122, 1999.
16. J. Y. A. Wang and E. H. Adelson. Representing Moving Images with Layers. *IEEE Transactions on Image Processing*, 3(5):625–638, September 1994.
17. Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn.*, pages 520–526, 1997.