From Maximum Entropy to Belief Propagation: An application to Skin Detection *

Huicheng Zheng, Mohamed Daoudi
MIIRE Group LIFL (CNRS UMR 8022)/ INT
ENIC-Telecom Lille 1
59655 Villeneuve d'Ascq, France
(Zheng,Daoudi)@enic.fr
Bruno Jedynak
Center for Imaging Science,
The Johns Hopkins University
and
Laboratoire de Mathématiques,
USTL, France
Bruno.Jedynak@jhu.fr

Abstract

We build a maximum entropy model for skin detection. This model imposes constraints on various marginal distributions. Parameter estimation as well as optimization cannot be tackled without approximations. We propose to use a tree approximation of the pixel lattice. Parameter estimation is then reduced to the estimations of color histograms for neighbor pixels. Moreover, the belief propagation algorithm permits to obtain fast solution for skin probability at pixel locations. We assess the performance on the Compaq database.

1 Introduction

1.1 Skin Detection

Skin detection consists in detecting human skin pixels from an image. It plays an important role in various applications such as face detection [14], searching and filtering image content on the web [6].

Research has been performed on the detection of human skin pixels in color images by use of various skin color models. One method is to define explicitly the skin color boundaries in some color space [6]. The simplicity of this method leads to a fast classifier. However, the challenge is to find a good color space and adequate decision rules empirically. Some researchers have used skin color models such as Gaussian , Gaussian mixture [14]. There are also nonparametric skin modeling method such as histograms based[11] and self-organizing map(SOM) based methods[3]. A recent evaluation appeared in [15].

*This work is supported by CNRS under MathStic project

In most experiments, skin pixels are acquired from a limited number of people under a limited range of lighting conditions. Unfortunately, the illumination conditions are often unknown and the variation in skin colors is much less constrained in practice. This is particularly true for web images captured under a wide variety of conditions. However, given a large collection of labeled training pixels including all human skin (Caucasians, Africans, Asians), one can still model the distribution of skin and non-skin colors. Jones and Rehg [12] compared an histogram-based model with a Gaussian mixture density model. The histogram models were found to be slightly superior.

A skin detection system is never perfect and different users use different criteria for evaluation. General appearance of the skin-zones detected, or other global criteria might be important for further processing. For quantitative evaluation, we will use false positives and detection rates. False positive rate is the proportion of non-skin pixels classified as skin and detection rate is the proportion of skin pixels classified as skin. The user might wish to combine these two indicators his own way depending on the kind of error he is more willing to afford. Hence we propose a system where the decision of each pixel is not binary but a floating number between zero and one, the larger the value, the larger the belief for a skin pixel. The user can then apply a threshold to obtain a binary image. Error rates for all possible thresholding are summarized in the Receiver Operating Characteristic (ROC) curve.

We use the Compaq Database [12]. It is a catalog of almost twenty thousand images. Each of them is manually segmented such that the skin pixels are labeled. Our goal is to infer a model from this set of data in order to perform skin detection on new images.

1.2 Methodology

Maximum Entropy Modeling (MaxEnt) is a method for inferring models from a data set. See [9] for the underlying philosophy. It works as follows: 1) choose relevant features 2) compute their histograms on the training set 3) write down the maximum entropy model within the ones that have the feature histograms as observed on the training set 4) estimate the parameters of the model 5) use the model for classification. This plan has been successfully completed for several tasks related to speech recognition and language processing. See for example [1] and the references therein. In these application the underlying graph on which the model is defined is a line graph or even a tree but in all cases it has no loops. When working with images, the graph is the pixel lattice. It has indeed many loops. A break through appeared with the work in [19] on texture simulation where 1) 2) 3) 4) was performed for images and 5) replaced by simulation.

We adapt to skin detection as follows: in 1) we specialize in colors and "skinness" for one pixel and two adjacent pixels. In 2) we compute the histogram of these features in the Compaq manually segmented database. Models for 3) are then easily obtained. In 4) we use the tree approximations. It consists in approximating locally the pixel lattice by a tree. The parameters of the MaxEnt models are then expressed analytically as functions of the histograms of the features. In 5) we pursue the approximation in 4): we use the Belief Propagation algorithm, see [16], which is exact in tree graph but only approximative in loopy graphs.

Indeed, one of us had already witnessed in a different context that tree approximation to loopy graph might lead to effective algorithms, see [8].

The rest of the paper is organized as follows: in section 2, we detail the features

used and compute the associated MaxEnt models. In section 3 we present the various tree approximations and the related Belief Propagation algorithm. section 4 is devoted to experiments and comparisons with alternative methods. Finally, the conclusion is in section 5.

2 Maximum Entropy Models

2.1 Notations

Let's fix the notations. The set of pixels of an image is S. The color of a pixel $s \in S$ is x_s. It is a 3 dimensional vector, each component being coded on one octet. We notate $C = \{0,\ldots,255\}^3$. The "skinness" of a pixel s, is y_s with $y_s = 1$ if s is a skin pixel and $y_s = 0$ if not. The color image, which is the vector of color pixels, is notated x and the binary image made up of the y_s's is notated y.

Let's assume for now that we knew the joint probability distribution $p(x,y)$ of the vector (x,y), then Bayesian analysis tells us that, whatever cost function the user might think of, all that is needed is the posterior distribution $p(y|x)$.

From the user's point of view, the useful information is contained in the one pixel marginal of the posterior, that is, for each pixel, the quantity $p(y_s = 1|x)$, quantifying the belief for skinness at pixel s. In practice the model $p(x,y)$ is unknown. Instead, we have the segmented Compaq Database. It is a collection of samples

$$\{(x^{(1)},y^{(1)}),\ldots,(x^{(n)},y^{(n)})\}$$

where for each $1 \leq i \leq n$, $x^{(i)}$ is a color image and $y^{(i)}$ is the associated binary skinness image. We assume that the samples are independent of each other with distribution $p(x,y)$. The collection of samples is referred later as the training data. Probabilities estimated using the classical empirical estimators are denoted with the letter q.

In what follows, we build models for the joint distribution of the skinness image and the color image $p(x,y)$ using maximum entropy modeling.

2.2 Baseline and Hiddend Markov (HMM) Models

First, we build a model that respects the one pixel marginal observed in the Compaq Database. That is, consider probability distributions over for (x,y)that verify:

$$\mathscr{C}_0 : \forall s \in S, \forall x_s \in C, \forall y_s \in \{0,1\}, p(x_s,y_s) = q(x_s,y_s)$$

In this expression, the quantity on the right side is the proportion of pixels with color x_s and label y_s among all the pixels in the training data. The MaxEnt solution under $\mathscr{C}_0$, using Lagrange multipliers is the following independent model:

$$p(x,y) = \prod_{s\in S} q(x_s,y_s) \tag{1}$$

This model is the most commonly used model in the literature [11]. We will use it as a baseline for evaluating subsequent models.

Including constraints on the labels of neighbor pixels leads naturally to a HMM model. This model has been studied thoroughly in [10] and shown to statistically improve the performances of the baseline model.

2.3 First Order Model (FOM)

The baseline model was built in order to mimic the one pixel marginal, that is $q(x_s, y_s)$ as observed on the database. Now, we constrain once more the MaxEnt model by imposing the two-pixel marginal, that is $p(x_s, x_t, y_s, y_t)$, for neighboring s and t, to match those observed in the training data. Hence we define the following constraints:

$$\mathscr{C}_1 : \forall s \in S, \forall t \in \mathscr{V}(s), \forall x_s \in C, \forall x_t \in C,$$
$$\forall y_s \in \{0,1\}, \forall y_t \in \{0,1\},$$
$$p(x_s, x_t, y_s, y_t) = q(x_s, x_t, y_s, y_t)$$

The quantity $q(x_s, x_t, y_s, y_t)$ is the proportion of times we observe the values (x_s, x_t, y_s, y_t) among all the couples of neighboring pixels, regardless of the orientation of the pixels s and t in the training set.

Clearly, $\mathscr{C}_1 \subset \mathscr{C}_0$. Using once more Lagrange multipliers, the solution to the MaxEnt problem under $\mathscr{C}_1$ is then the following Gibbs distribution:

$$p(x,y) \approx \prod_{<s,t>} \lambda(x_s, x_t, y_s, y_t) \tag{2}$$

where $\lambda(x_s, x_t, y_s, y_t) > 0$ are parameters that should be set up to satisfy the constraints. Assuming that one color can take 256^3 values, the total number of parameters is as huge as $256^3 \times 256^3 \times 2 \times 2$.

2.4 Parameter Estimation

Parameter estimation in the context of MaxEnt is still an active research subject, especially in situations where the likelihood function cannot be computed for a given value of the parameters. This is the case here, since the partition function cannot be evaluated even for very small size images. One line of research consists in approximating the model in order to obtain a formula where the partition function no longer appears: Pseudo-likelihood [2], [5] and mean field methods [18], [7] are among them. Another possibility is to use stochastic gradient as in [17]. However, due to the large number of parameters in the FOM model, this is a real challenge.

Moreover, recall that the quantities of interest for the users are the one pixel marginal of the posterior, that is for each s the quantity $p(y_s = 1|x)$. These quantities are not easily available due once more to the impossibility of evaluating the partition function. One has then to use stochastic algorithm as the Gibbs sampler which is time consuming or to rely on an approximate model.

By approximating the local pixel lattice with a tree, the parameter estimation is eradicated. We can further take advantage of the Belief Propagation algorithm for the fast and exact computing of the one pixel marginal $p(y_s = 1|x)$ as we shall see now.

3 Tree Approximations of Pixel Lattice

3.1 Maximum Entropy Models in Tree Graphs

The FOM defined in (2) is a Markov Random Field on the non-oriented pixel graph. Let us assume for now that this graph was a tree: that is a connected graph without loops.

Then, the Maxent solution under $\mathscr{C}_1$ would be

$$p(x,y) \approx \prod_{<s,t>} \frac{q(x_s,x_t,y_s,y_t)}{q(x_s,y_s)q(x_t,y_t)} \prod_{s\in S} q(x_s,y_s) \tag{3}$$

The proof is as follows: we know from [13] that any pairwise MRF on a tree graph can be written

$$p(z) \approx \prod_{<s,t>} \frac{p(z_s,z_t)}{p(z_s)q(z_t)} \prod_{s\in S} p(z_s) \tag{4}$$

where $p(z_s)$ is the one-site marginal of p and $p(z_s,z_t)$ is it's two-site marginal.

Applying this result to $z=(x,y)$ and replacing p with q on the right side permits to obtain the model in equation (3). By construction it is in C_1. Moreover it has the same form as the one in equation (2) which concludes the proof.

3.2 Tree-based First Order Model (TFOM)

The model in (3) cannot be used as it is. Indeed, the quantities $q(x_s,x_t,y_s,y_t)$ cannot be directly extracted from the database without drastic over-fitting due to high dimensionality.

We propose to estimate $q(x_s,x_t,y_s,y_t)$ using the classic MaxEnt procedure. The constraints include the marginal histograms for one pixel as well as the histogram of the difference of two adjacent pixels. This can be done off-line and will not introduce any additional online load to skin detection. To be more precise, denoting by $p(x_s,x_t,y_s,y_t)$ the two-site joint distribution, we define the following constraints:

$$\mathscr{C}^* : \forall x_s \in C, \forall x_t \in C, \forall y_s \in \{0,1\}, \forall y_t \in \{0,1\},$$

$$p(x_s,y_s) = q(x_s,y_s)$$

$$p(x_t,y_t) = q(x_t,y_t)$$

$$p(x_t - x_s, y_s, y_t) = q(x_t - x_s, y_s, y_t)$$

The entropy of the joint distribution is defined as:

$$H(p) = -\sum_{x_s,x_t,y_s,y_t} p(x_s,x_t,y_s,y_t)\log p(x_s,x_t,y_s,y_t)$$

Our goal is to find the MaxEnt solution

$$p^*(x_s,x_t,y_s,y_t) = \text{argmax}_p H(p)$$

subject to the constraints $\mathscr{C}^*$. Using Lagrange multipliers, then the MaxEnt solution is

$$p^*(x_s,x_t,y_s,y_t) = \mathscr{P}_\lambda \cap \mathscr{C}^*$$

in which

$$\mathscr{P}_\lambda = \{\frac{\exp[\lambda(x_s,y_s)+\lambda(x_t,y_t)+\lambda(x_t-x_s,y_s,y_t)]}{Z_\lambda}\} \tag{5}$$

where $Z_\lambda = \sum_{x_s,x_t,y_s,y_t} \exp[\lambda(x_s,y_s)+\lambda(x_t,y_t)+\lambda(x_t-x_s,y_s,y_t)]$ is a normalizing constant.

The MaxEnt model $p^*(x_s, x_t, y_s, y_t)$ is equivalent to a model $p^*_\lambda(x_s, x_t, y_s, y_t) \in \mathscr{P}_\lambda$ that maximizes the likelihood of the training data [1]. However, the parameters λ^* that maximize the likelihood cannot be found analytically. Instead, we have to resort to numerical methods. From the perspective of numerical optimization, the likelihood function is well behaved since it is smooth and convex-$\cap$ in λ. An optimization method specifically tailored to the maximum entropy problem is the generalized iterative scaling algorithm of Darroch and Ratcliff[4]. In the algorithm, we first initialize the parameters λ's. Then the iterative scaling process follows. For each iteration that updates the parameters, the likelihood function will climb a little toward the maximum value. We stop the iterative scaling when such updating will introduce very little gain in the likelihood.

The generalized iterative scaling algorithm for our solution of λ^* has been proposed by [1]. It is accelerated as follows:

1. Initialize λ

$$\forall x_s \in C, \forall x_t \in C, \forall y_s \in \{0,1\}, \forall y_t \in \{0,1\},$$
$$\lambda_0(x_s, y_s) = \lambda_0(x_t, y_t) = \lambda_0(x_t - x_s, y_s, y_t) = 0.0$$

2. Update λ

(1) Calculate marginals and Z_λ

(a) Initialize marginals and Z_λ to 0.

(b) For each value (x_s, x_t, y_s, y_t), we calculate

$$g(x_s, x_t, y_s, y_t) = \exp[\lambda(x_s, y_s) + \lambda(x_t, y_t) + \lambda(x_t - x_s, y_s, y_t)]$$

once and update the normalizing constant and marginal arrays,

$$Z_\lambda \leftarrow Z_\lambda + g(x_s, x_t, y_s, y_t)$$
$$p(x_s, y_s) \leftarrow p(x_s, y_s) + g(x_s, x_t, y_s, y_t)$$
$$p(x_t, y_t) \leftarrow p(x_s, y_s) + g(x_s, x_t, y_s, y_t)$$
$$p(x_t - x_s, y_s, y_t) \leftarrow p(x_t - x_s, y_s, y_t) + g(x_s, x_t, y_s, y_t)$$

(c) For all the marginals

$$p(x_s, y_s) \leftarrow p(x_s, y_s)/Z_\lambda$$
$$p(x_t, y_t) \leftarrow p(x_t, y_t)/Z_\lambda$$
$$p(x_t - x_s, y_s, y_t) \leftarrow p(x_t - x_s, y_s, y_t)/Z_\lambda$$

(2) Calculate all $\Delta\lambda$

$$\begin{aligned}
\Delta\lambda(x_s, y_s) &= \ln \frac{q(x_s, y_s)}{p(x_s, y_s)} \\
\Delta\lambda(x_t, y_t) &= \ln \frac{q(x_t, y_t)}{p(x_t, y_t)} \\
\Delta\lambda(x_t - x_s, y_s, y_t) &= \ln \frac{q(x_t - x_s, y_s, y_t)}{p(x_t - x_s, y_s, y_t)}
\end{aligned}$$

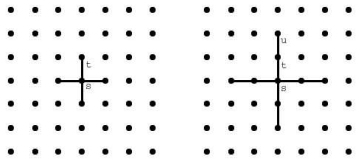

Figure 1: **Left:** 4-star tree of depth 1. **right:** 4-star tree of depth 2.

(3) Update λ according to: $\lambda \leftarrow \lambda + \Delta\lambda$

3. Goto 2 until convergence.

The above algorithm is more than 3 times faster than the "basic" version. This is probably due to the avoidance of the subscript searching process in the calculation of $\Delta\lambda(x_t - x_s, y_s, y_t)$.

3.3 4-Star Tree Approximation

The 4-star tree of depth 1 rooted at s, $\mathscr{T}_1^s$, is built by adding to s its 4 neighbors in the pixel lattice. For the 4-star tree with one more depth, we add one node u for each leaf t of $\mathscr{T}_1^s$, following the direction $s \to t$. We repeat this process until the tree reaches the depth needed. Figure 1 shows the construction of 4-star trees.

3.4 One-Site Marginal—Belief Propagation Algorithm (BP)

Our aim is to compute for each pixel s, the quantity $p(y_s|x_t, t \in \mathscr{T}_k^s)$, for p in one of the models above, and for k ranging from 1 to say 5. This computation can be done exactly. Moreover, it can be done efficiently using the Belief Propagation Algorithm (BP). This algorithm has been discovered in different scientific communities. It is called BP in A.I., Viterbi algorithm in the special case of line graphs and dynamic programming in combinatorial optimization. See [16] and the references therein for a detailed account.

For the generic pairwise model

$$p(x,y) \approx \prod_{<s,t>} \psi(x_s, x_t, y_s, y_t) \prod_{s \in S} \phi(x_s, y_s) \tag{6}$$

The BP algorithms consists in updating messages along the edges of the tree. $\forall y_u \in \{0,1\}$, we compute:

$$m_{vu}(y_u) \leftarrow \sum_{y_v} \phi(x_v, y_v) \psi(x_u, x_v, y_u, y_v) \prod_{w \in \mathscr{V}(v), w \neq u} m_{wv}(y_v) \tag{7}$$

where $\mathscr{V}(v)$ are the neighbors of v. The quantities m_{vu} are interpreted as a message coming from v to u. The message updating process proceeds from the leaves to the root. We compute a message m_{vu} only when $\forall w \in \mathscr{V}(v), w \neq u$, $\forall y_v \in \{0,1\}$, $m_{wv}(y_v)$ is available. When we have got all the messages to the root site s, we can calculate the following marginals for $y_s \in \{0,1\}$:

$$p(y_s, x_t, t \in \mathscr{T}_k^s) \approx \phi(x_s, y_s) \prod_{t \in \mathscr{V}(s)} m_{ts}(y_s)$$

Figure 2: **Left:** Original image. **Center :** Skin detection. Baseline model. **Right:** Skin detection. Tree First Order Model.

Then

$$p(y_s = 1|x_t, t \in \mathscr{T}_k^s) = \frac{p(y_s = 1, x_t, t \in \mathscr{T}_k^s)}{p(y_s = 0, x_t, t \in \mathscr{T}_k^s) + p(y_s = 1, x_t, t \in \mathscr{T}_k^s)}$$

4 Experiments

All experiments are made using the following protocol. The labeled Compaq database contains about $13,562$ photographs, in which there are $4,649$ photographs with skin and $8,913$ photographs without skin. Each photograph with skin is accompanied with a binary mask image indicating skin and non-skin regions. These masks were obtained by manual labeling[11]. This database is split into two almost equal parts randomly. The first part, containing nearly 674 million pixels is used as training data while the other one, the test set, is left aside for ROC curve computation. We use 32 bins per RGB channel in the following experiments.

Fig. 3 shows ROC curves of 4-star TFOM and Baseline model computed from the test set. The Baseline model permit to detect 80.6% of the skin pixels with 10% of false positive rate.

4.1 TFOM Model

Result for the TFOM model with one iteration of the BP algorithm is presented in Fig. 2. More iterations do not improve the performance of TFOM. Bulk results in Fig. 3 shows that the 4-star TFOM has a uniform improvement over the Baseline model as well as over the HMM model. For example, the 4-star TFOM model permits to detect 82.9% of the skin pixels with 10% of false positive rate. Computational time is about 0.70 second per image for 4-star TFOM. All the experiments are performed on a PC with a Pentium 4 processor at 1.7 Ghz and 256 MB memory.

5 Summary and Conclusions

We have shown in this paper that the nowadays popular Maximum Entropy Modeling method can lead to an efficient algorithm for a supervised image segmentation problem. We have used extensively a tree approximation that consists in approximating locally the loopy pixel lattice by a tree graph. The natural algorithm for assessing probability for skin at pixel locations in this context is the Belief Propagation algorithm. The resulting algorithm performs uniformly better than the wide spread Baseline model. It also improves over a Hidden Markov Model that was considered in an early version of this work.

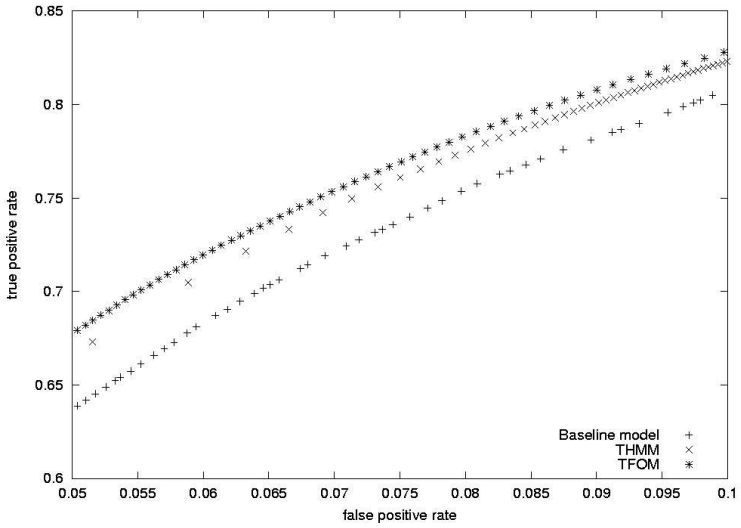

Figure 3: Receiver Operating Characteristics (ROC) curves. x-axis is the false positive rate, y-axis is the detection rate which is the complement to one of the false negative rate. **Lower curve:** Baseline model. **Center curve:** HMM model. **upper Curve:** Tree First Order model

References

[1] A. Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, B*, 48(3):259–302, 1986.

[3] I. Craw D. Brown and J. Lewthwaite. A som based approach to skin detection with application in real time systems. In *Proc. of the British Machine Vision Conference*, 2001.

[4] J.N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, (43):1470–1480.

[5] Arnoldo Frigessi Fabio Divino and Peter J. Green. Penalized pseudolikelihood inference in spatial interaction models with covariates. *The Scandinavian Journal of Statistics*, 27(3), 2003.

[6] M.M. Fleck, D.A. Forsyth, and C. Bregler. Finding naked people. In *Proc. European Conf. on Computer Vision*, pages 593–602. B. Buxton, R. Cipolla, Springer-Verlag, Berlin, Germany, 1996.

[7] F. Forbes G. Celeux and N. Peyrard. Em procedures using mean field-like approximations for markov model-based image segmentation. *Pattern Recognition*, 36(1):131–144, 2003.

[8] D. Geman and B. Jedynak. An active testing model for tracking roads in satellite images. *IEEE Trans. on PAMI*, 18(1):1–14, January 1996.

[9] E. Jaynes. Probablity theory: The logic of science. http://omega.albany.edu:8008/JaynesBook.

[10] B. Jedynak, H. Zheng, and M. Daoudi. Statistical models for skin detection. In *IEEE Workshop on Statistical Analysis in Computer Vision*, June 2003.

[11] Michael J. Jones and James M. Rehg. Statistical color models with application to skin detection. Technical Report CRL 98/11, Compaq, 1998.

[12] M.J. Jones and J. M. Rehg. Statistical color models with application to skin detection. In *Computer Vision and Pattern Recognition*, pages 274–280, 1999.

[13] J. Pearl. *Probabilistic Reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[14] Jean-Christophe Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Fourth International Conference On Automatic Face and gesture Recognition*, pages 54–61, 2000.

[15] V. Sazonov V. Vezhnevets and A. Andreeva. A survey on pixel-based skin color detection techniques. In *Graphicon2003, 13th International Conference on the Computer Graphics and Vision*, Moscow, Russia, September 2003.

[16] J. S. Yedida, W. T. Freeman, and Y. Weiss. Understanding belief propagation and it's generalisations. Technical Report TR-2001-22, Mitsubishi Research Laboratories, January 2002.

[17] Laurent Younes. Estimation and annealing for gibbsian fields. *Annales de l'Institut Henry Poincaré, Section B, Calcul des Probabilités et Statistique*, 24:269–294, 1998.

[18] J. Zhang. The mean field theory in em procedure for markov random fields. *IEEE Transactions on Signal Processing*, 40(10):2570–2583, October 1992.

[19] S.C. Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (frame): towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.