Statistical Models for Skin Detection *

Bruno Jedynak
Laboratoire de
Mathématiques Appliquées, USTL
Bât M2, Cité scientifique
59655 Villeneuve d'Ascq, France
Bruno.Jedynak@univ-lille1.fr

Huicheng Zheng and Mohamed Daoudi
MIIRE Group
GET / INT / ENIC-Telecom Lille 1
Rue G. Marconi, Cité scientifique
59655 Villeneuve d'Ascq, France
(Zheng,Daoudi)@enic.fr

Abstract

We consider a sequence of three models for skin detection built from a large collection of labelled images. Each model is a maximum entropy model with respect to constraints concerning marginal distributions. Our models are nested. The first model is well known from practitioners. Pixels are considered as independent. The second model is a Hidden Markov Model. It includes constraints that force smoothness of the solution. The third model is a first order model. The full color gradient is included. Parameter estimation as well as optimization cannot be tackled without approximations. We use thoroughly Bethe tree approximation of the pixel lattice. Within it , parameter estimation is eradicated and the belief propagation algorithm permits to obtain exact and fast solution for skin probability at pixel locations. We then assess the performance on the Compaq database.

1. Introduction

1.1. Skin Detection

Skin detection consists in detecting human skin pixels from an image. The system output is a binary image defined on the same pixel grid as the input image.

Skin detection plays an important role in various applications such as face detection [13], searching and filtering image content on the web [14][5]. Research has been performed on the detection of human skin pixels in color images by use of various statistical color models. Some researchers have used skin color models such as Gaussian , Gaussian mixture or histograms [12][10]. 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 in an arbitrary image, so 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) we can still model the distribution of skin and non-skin colors in the color space. Recently Jones and Rehg [9] proposed techniques for skin color detection by estimating the distribution of skin and non-skin color in the color space using labeled training data. The comparison of histogram models and Gaussian mixture density models estimated with EM algorithm was analyzed for the standard 24-bit RGB color space. The histogram models were found to be slightly superior to Gaussian mixture models in terms of skin pixel classification performance for this color space.

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 output 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 have in our hands the publicly available Compaq Database [9]. It is a catalog of almost twenty thousand images. Each of them is manually segmented such that the skin pixels are labelled. 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 [7] for the underlying philosophy. It works as follows: 1) choose relevant

* This work was partially supported by European project Internet Action Plan Contract Number POESIA-2117/27572 www.poesia-filter.org

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 classifi cation. 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 defi ned 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 [21] 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 for one pixel and two adjacent pixels given "skinness". 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 Beth tree approximation, see [16]. 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. This is a particularity of our features. In 5) we pursue the approximation in 4): we use the Belief Propagation algorithm, see [18], 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 [6].

The main practical contribution of this paper is to propose a reasonably quick skin detection algorithm that outperform the standart method.

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 Bethe tree approximation 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 fi x 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 a moment 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 are estimated by using classical empirical estimators and are denoted with the letter q.

In what follows, we build models for the probability distribution of the skinness image given the color image using maximum entropy modeling.

2.2. Baseline Model

First, we build a model that respects the one pixel marginal observed in the Compaq Database. That is, for each image x, consider the set of probability distributions over binary images defi ned on the same grid as x that verify:

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

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

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

This model is the most commonly used model in the literature [12][10]. We will use it as a baseline for evaluating subsequent models.

2.3. Hidden Markov Model (HMM)

The baseline model is certainly too loose and one might hope to get better detection results by constraining it to a model that takes into account the fact that skin zones are not purely random but are made of large regions with regular shapes. Hence, we fi x the marginals of y for all the neighboring pixels couples. We use 4-neighbors system for simplicity in all that follows. For two neighboring pixels s and t, the proportion of times that we observe $(y_s = a, y_t = b)$ should be $q(a, b)$ for $a = 0, 1$ and $b = 0, 1$, the corresponding quantities measured on the training set. Hence let us defi ne the following constraints:

$$\mathcal{D} : \forall s \in S, \forall t \in \mathcal{V}(s), \forall a \in \{0, 1\}, \forall b \in \{0, 1\}$$

$$p(y_s = a, y_t = b) = q(a, b)$$

where $\mathcal{V}(s)$ are the neighbors of s. For each image x, the MaxEnt model under

$$\mathcal{C}_1(x) = \mathcal{C}_0(x) \cap \mathcal{D}$$

is then the following Gibbs distribution [15].

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

where the sum indexed by $< s, t >$ ranges over all pairs of neighbors pixels and $\lambda(y_s, y_t) > 0$ defines parameters that should be set up such that the constraints $\mathcal{D}$ are satisfied. The sign $\approx$ indicates here and after equality up to a function that depends possibly on x but not on y. This function is called the partition function in statistical mechanics.

2.4. First Order Model (FOM)

The baseline model was built in order to mimic the one pixel marginal of the posterior, that is $q(y_s|x_s)$ as observed on the database. Then, in building the HMM model we added constraints on the prior $p(y)$ in order to smooth the model. Now, we constrain once more the MaxEnt model by imposing the two-pixel marginal of the posterior, that is $p(y_s, y_t|x_s, x_t)$, for 4-neighbor s and t, to match those observed in the training data. Hence we define for each image x, the following constraints:

$$\mathcal{C}_2(x) : \forall s \in S, \forall t \in \mathcal{V}(s), \forall x_s \in C, \forall x_t \in C,$$

$$\forall y_s \in \{0, 1\}, \forall y_t \in \{0, 1\},$$

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

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

Clearly, for each x, $\mathcal{C}_2(x) \subset \mathcal{C}_1(x) \subset \mathcal{C}_0(x)$. Using once more Lagrange multipliers, the solution to the MaxEnt problem under $\mathcal{C}_2(x)$ is then the following Gibbs distribution:

$$p(y|x) \approx \prod_{<s,t>} \lambda(x_s, x_t, y_s, y_t) \tag{3}$$

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 $256^3 \times 256^3 \times 2 \times 2$.

2.5. 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], [4] and mean field methods [20], [3] are among them. Another possibility is to use stochastic gradient as in [19]. 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 $p(ys = 1|x)$. These quantities for the HMM model as well as for the FOM 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.

Bethe Tree approximation is a model approximation that deals both with parameter estimation and fast computing of the one pixel marginal of the posterior as we shall see now.

3. Bethe Tree approximation of Maximum Entropy Models

3.1. Maximum Entropy Models in Tree Graphs

The FOM defined in (3) is a Markov Random Field on the non-oriented pixel graph with 4-neighbor connectivity. Let us assume for now that this graph was a tree: that is a connected graph without loops. Then, the Maxent solution for fixed x under $\mathcal{C}_2(x)$ would be

$$p(y|x) \approx \prod_{<s,t>} \frac{q(x_s, x_t|y_s, y_t) q(y_s, y_t)}{q(x_s|y_s) q(x_t|y_t) q(y_s) q(y_t)} \prod_{s \in S} q(x_s|y_s) q(y_s) \tag{4}$$

The proof is as follows: we know from [11] that any pairwise MRF on a tree graph can be written

$$p(z) \approx \prod_{<s,t>} \frac{q(z_s, z_t)}{q(z_s) q(z_t)} \prod_{s \in S} q(z_s) \tag{5}$$

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

Applying this result to $z = (x, y)$ permits to obtain the model in equation (4). By construction it is in $C_2(x)$. Moreover it has the same form as the one in equation (3) which concludes the proof.

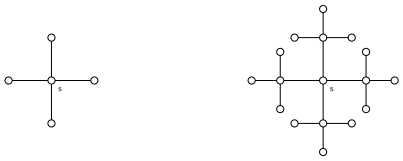

Figure 1: **Left:** a Bethe tree of depth 1 rooted at s. **Right:** a Bethe tree of depth 2 rooted at s

The main point to be made here is that the model in equation (4) is parameter free. This model will be referenced later as TFOM for Tree First Order Model.

Still assuming that the graph is a tree, the MaxEnt model under $C_1(x)$ can be derived from the one in equation (4) by assuming conditional independence, that is, for each couple $< s, t >$ of neighboring pixels, $q(x_s, x_t|y_s, y_t) = q(x_s|y_s)q(x_t|y_t)$, leading to

$$p(y|x) \approx \prod_{<s,t>} \frac{q(y_s, y_t)}{q(y_s)q(y_t)} \prod_{s \in S} q(x_s|y_s)q(y_s) \qquad (6)$$

As the model in (4), this one is parameter free. It will be referenced later as THMM for Tree Hidden Markov Model. Going further, and assuming that the prior is made of independent and equally distributed components, one can derive from equation (6) that the MaxEnt solution under $C_0(x)$ for tree graph coincides with the Baseline model for the image lattice given in equation (1).

3.2. Bethe Tree Approximation

Bethe tree are named after the physicist H.A. Bethe who used trees in statistical mechanics problems. They have been introduced in computer vision as a way of approximating estimators in Markov Random Field models in [16]. We shall revisit this work in connection with maximum entropy models.

The key idea is to provide a tree that approximates locally the pixel lattice. More precisely, for each pixel s, we consider a sequence of trees $\mathcal{T}_1^s, \mathcal{T}_2^s, \ldots$ of increasing depth. The construction is as follow: the root node of the tree is associated with s. For each neighbor t of s in the pixel-graph, a child node indexed by t is add to the root node. This defines $\mathcal{T}_1$. Subsequently, for each u, neighbor of a neighbor of s, (excluding s itself), a grandchild node indexed by u is added to the appropriate child node. This defines $\mathcal{T}_2$, and so on, see [16] for a detailed account. An important remark is that a single pixel might lead to several different nodes in the tree! For example $\mathcal{T}_2^s$ is build with s, the neighbors of s and the neighbors of theses. Using 4-neighbors , and assuming that s is not in the border of the image, this makes up 13 pixels, but the associated tree has 17 nodes, 4 pixels being replicated twice each, see figure 1.

3.3. Belief Propagation Algorithm (BP)

Our aim is to compute for each pixel s, the quantity $p(y_s|x_s, s \in \mathcal{T}_k)$, for p one the model 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 [18] and the references therein for a detailed account.

For the generic pairwise model

$$p(y|x) \approx \prod_{<s,t>} \psi(x_s, x_t, y_s, y_t) \prod_{s \in S} \phi(x_s, y_s) \qquad (7)$$

The BP algorithms consists in computing k times :

$$m_{ts}(y_s) \leftarrow \sum_{y_s} \phi(x_t, y_t)\psi(x_s, x_t, y_s, y_t) \prod_{u \in \mathcal{V}(t), u \neq s} m_{ut}(y_t) \qquad (8)$$

where $\mathcal{V}(t)$ are the neighbors of t. The quantities m_{st} are interpreted as a message coming from t to s and are initialized with the value one. We then obtain:

$$p(y_s = 1|x_s, s \in \mathcal{T}_k^s) \approx \phi(x_s, y_s) \prod_{t \in \mathcal{V}(s)} m_{ts}(y_s) \qquad (9)$$

4. Experiments

All experiments are made using the following protocol. The Compaq database contains about 18,696 photographs. It is split into two almost equal parts randomly. The first part, containing nearly 2 billion pixels is used as training data while the other one, the test set, is left aside for ROC curve computation.

4.1. Baseline Model

The key ingredients of the Baseline model in (1) are the two 3-dimension histograms, $q(x_s|y_s = 1)$ and $q(x_s|y_s = 0)$ describing the distribution of the color of a pixel for skin regions and non-skin regions respectively. We used for each histogram 32^3 bins. Each bin made of 32 color values.

Figure 2, Top Left is one of the test images. It is a large color image of 635×918 pixels. Top right is a grey level image. The grey-level is proportional to the quantity $p(y_s = 1|x)$ evaluated with the Baseline model. Many non-skin pixels are detected. Figure 3 shows ROC curves computed from 100 images (more than 10 millions pixels), randomly extracted from the test set. The Baseline model (with crosses) permit to detect 69% of the skin pixels with 5% of false positive rate. All the experiments are performed on a PC with a Pentium 4 processor at 1.7 Ghz and 256 MB memory. The execution time is given for a 1000x1000 pixel image. It is 0.9 seconds for the baseline model.

4.2. THMM Model

Result for the THMM model in (6) and one iteration of the BP algorithm is presented for a single image in figure 2. The grey level is, as before, proportional to the belief for skin at pixel locations. We remark that the beliefs are almost binary. Bulk results are in figure 3 showing uniform improvement over the baseline model. We report an increase of 3%, from 69% (baseline model) to 72% (THMM model) of the detection rate for the same false positive rate. This is measured over a test set of more than 10 millions pixels not intersecting the training set. The standard Normal test for proportions indicates that under the baseline model, the 99% confidence interval for the detection rate is $[68.94\%, 69.06\%]$. The value of 72% found for the THMM model is statistically significant. In [8], we estimated the parameters of the HMM model using a simulation technique borrowed from [17]. We showed that the resulting model was indeed very close to the THMM model. Moreover, when using a Gibbs sampler to estimate the probability for skin at pixel locations, we obtained comparable results with the ones reported here, but with a running time increased by a factor of more than ten. The execution time for the THMM is 14.2 seconds to be compared with 0.9 seconds for the baseline model.

4.3. TFOM Model

The TFOM model in (4) 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. In effect the four histogram involved have a support of dimension six, three dimensions for each pixel. Hence, some kind of dimension reduction is needed. We have experimented the following:

$$q(x_s, x_t|y_s, y_t) \sim q(x_s|y_s)q(x_t - x_s|y_s, y_t) \qquad (10)$$

That is, we assume that the color gradient at s, measured by the quantity $x_t - x_s$, is, given the labels at s and t, independent of the actual color x_s. Evaluation of the right side of the sign $\sim$ requires to compute 6 histograms with a support of dimension 3. We use 32^3 bins of 32 colors each.

Result for this model with one iteration of the BP algorithm is presented for a single image in figure 2. Bulk results are in figure 3 showing slight but uniform improvement over the THMM model. For example, the TFOM model (with squares) permit to detect 72% of the skin pixels with 5% of false positive rate. Computational time is 14.4 seconds for a 1000x1000 image which is about the same as for the THMM model. Increasing the number of iterations didn't increase significantly the overall performances. Moreover, using the Gibbs sampler or Mean Field approximation didn't improve the results. Detailed experiments will be presented elsewhere.

Figure 2: **Top left:** original color image. **Top right:** result for the Baseline model. **Bottom left:** result for the THMM model. **Bottom right:** result for the TFOM model

An alternative to the approximation in (10) is the following:

$$\frac{q(x_s, x_t|y_s, y_t)}{q(x_s|y_s)q(x_t|y_t)} \sim \frac{q(z_s, z_t|y_s, y_t)}{q(z_s|y_s)q(z_t|y_t)} \qquad (11)$$

where z_s and z_t are the grey level at s and t. The joint distribution of colors is replaced by the more tractable joint distribution of grey levels. However, the resulting ROC curve is not as good as the one using the approximation in (10) showing that the color gradient for skin might indeed contain useful features.

5. Summary and Conclusions

From the practical point of view, we have derived several algorithm for skin detection that perform uniformly better than the wide spread color model referred here as the Baseline model. The computational time is increased by about a factor ten.

Moreover, we have shown that the nowadays popular Maximum Entropy Modeling method can lead to an efficient algorithm for a supervised image segmentation problem. We have used extensively the Bethe Tree method that

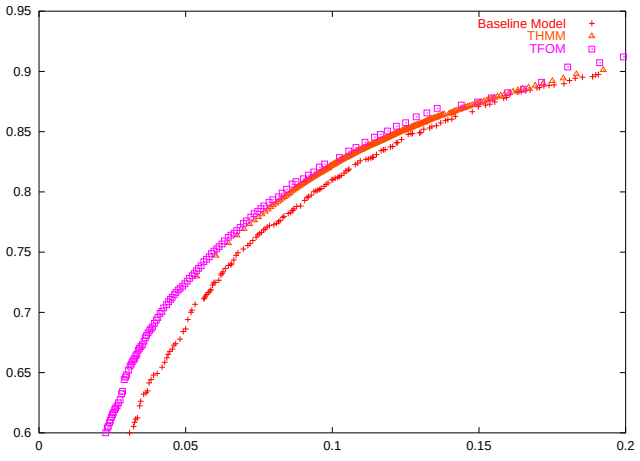

Figure 3: Receiver Operating Characteristics (ROC) curve for each model. x-axis is the false positive rate, y-axis is the detection rate which is the complement to one of the false negative rate. Baseline model is shown with crosses, THMM with triangles, while TFOM is shown with squares.

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. Experimental results have shown that it is much faster than a stochastic sampling scheme as the Gibbs sampler. It compares also favorably in term of performance with the Mean Field Approximation Method.

If asymptotic properties of the Beth tree approximations, i.e. letting the depth of the tree tend to infi nity are related to a well known approximation in statistical physics, [18], we have witnessed in practice that large tree models do not seem effi cient.

References

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

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

[3] G. Celeux, F. Forbes, and N. Peyrard. Em procedures using mean fi eld-like approximations for markov model-based image segmentation. *to appear in Pattern Recognition*, 2002.

[4] F. Divino and A. Frigessi. Penalized pseudolikelihood inference in spatial interaction models with covariates. *to appear on the Scandinavian Journal of Statistics*, 2000.

[5] M. Fleck, D. 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.

[6] 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.

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

[8] B. Jedynak, H. Zheng, M. Daoudi, and D. Barret. Maximum entropy models for skin detection. In *Proceedings Third Indian Conference on Computer Vision, Graphics and Image Processing*, pages 276–281. Allied Publishers Private Limited, December 16-18 2002.

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

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

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

[12] J.-C. Terrillon, M. David, and S. Akamatsu. Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments. In *IEEE Third International Conference on Automatic Face and gesture Recognition*, pages 112–117, 1998.

[13] J.-C. 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.

[14] J. Z. Wang, J. Li, G. Wiederhold, and O. Firschein. Classifying objectionable websites based on image content. *Notes in Computer Science, Special issue on iteractive distributed multimedia systems and telecommunication services*, 21/15:113–124, 1998.

[15] G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods*. Springer-Verlag, 1995.

[16] C. Wu and P. C. Doerschuk. Tree approximations to markov random fi elds. *IEEE Transactions on PAMI*, 17(4):391–402, April 1995.

[17] Y. Wu, S. Zhu, and X. Liu. Equivalence of julesz ensemble and frame models. *International Journal of Computer Vision*, 38(3):247–265, July 2000.

[18] 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.

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

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

[21] S. Zhu, Y. Wu, and D. Mumford. Filters, random fi elds and maximum entropy (frame): towards a unifi ed theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.