Skin Detection Using Pairwise Models ⋆

Bruno Jedynak, Huicheng Zheng and Mohamed Daoudi [1,2]

Abstract

We consider a sequence of three models for skin detection built from a large collection of labeled images. Each model is a maximum entropy model with respect to constraints concerning marginal distributions. Our models are nested. The first model, called the baseline model is well known from practitioners. Pixels are considered independent. Performance, measured by the ROC curve on the Compaq Database is impressive for such a simple model. However, single image examination reveals very irregular results. The second model is a Hidden Markov Model which includes constraints that force smoothness of the solution. The ROC curve obtained shows better performance than the baseline model. Finally, color gradient is included. Thanks to Bethe tree approximation, we obtain a simple analytical expression for the coefficients of the associated maximum entropy model. Performance, compared with previous model is once more improved.

Key words: maximum entropy models, skin detection, Markov random field.

1 Introduction

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.

⋆ This work was partially supported by European Community IAP 2117/27572-POESIA www.poesia-filter.org

[1] Bruno Jedynak is within the Laboratoire de Mathématiques Paul Painlevé, USTL, Bât M2, Cité scientifique, 59655 Villeneuve d'Ascq, France. He is currently Visiting Associate Professor at the Center for Imaging Science, The Johns Hopkins Univesity.
Email: bruno.jedynak@jhu.edu

[2] Huicheng Zheng and Mohamed Daoudi are within MIIRE Group, INT/LIFL (CNRS UMR 8022), Rue G. Marconi, Cité scientifique, 59655 Villeneuve d'Ascq, France.
Email: (Zheng,Daoudi)@enic.fr

Skin detection plays an important role in various applications such as face detection [1], searching and filtering image content on the web [2][3]. Research has been performed on the detection of human skin pixels in color images and on the discrimination between skin pixels and "non-skin" pixels by use of various statistical color models. Some researchers have used skin color models such as Gaussian, Gaussian mixture or histograms [4] [5]. 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, in [6], the authors proposed to estimate the distribution of skin and non-skin color 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.

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 positive rate and detection rate. 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 Compaq Database [6]. It is a catalog of almost twenty thousand images. Each of them is manually segmented such that the skin pixels are labeled. Our goal in this paper is to explore different ways in which this set of data can be used to perform skin detection on new images. We will use Markov random field approach [7] [8] combined with Maximum Entropy Modeling [9] [10], referred to as MaxEnt.

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. When working with images, the graph underlying the model is the pixel lattice. It has many nodes and many loops. Task 4) is much more difficult. A break through appeared with the work in [11] on texture simulation where 1) 2) 3) 4) was performed for images and 5) replaced by simulation.

We adapt this methodology 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 Beth tree approximation, see [12]. 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 use the Gibbs sampler algorithm for inferring the probability for skin at each pixel location.

We consider a sequence of three maximum entropy models with respect to various constraints concerning marginal distributions. The first model imposes constraints on one-pixel marginals. The solution is a baseline model in which pixels are considered independent. This model is well known from practitioners[6]. The baseline model is certainly too loose and does not take into account the fact that skin zones are not purely random but are made of large regions with regular shapes. Hence, in the second model, we add constraints on the distribution of neighboring labels in order to smooth the solution. Finally, color gradient is included in building the third model. We hope that the changes in neighboring colors will help discriminate skin pixels from non-skin ones.

The rest of this paper is organized as follows: After setting up the notations in section 2, we present in section 3 the baseline model. In section 4, we present the second model, which is a hidden Markov Random Field model. A novel method for parameter estimation is explored. In section 5, we examine the third model which takes into account the color gradient. Finally, in Section 7 we present concluding remarks.

2 Notations

Let us 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 usually 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 x and the binary image made up of the y_s's is

notated y. The letter "p" will denote "probability of". The actual probability measure will depend upon context.

Let us 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 given the full color image.

In practice the model $p(x, y)$ is unknown. Instead, we have the Compaq Database. It is a collection of samples

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

where for each $1 \leq i \leq n = 18,696$, $x^{(i)}$ 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 joint probability distribution of color and skinness image using maximum entropy modeling.

3 Baseline Model

3.1 Defining the model

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

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

In (1), the quantity on the right side of the equal sign is the proportion of pixels with color x_s and label y_s in the training data. The MaxEnt solution under $\mathcal{C}_0$ is the independent model:

$$p(x, y) = \prod_{s \in S} q(x_s, y_s) \tag{2}$$

The proof is postponed to Appendix A. Using Bayes formula, one then obtains:

$$p(y|x) = \prod_{s \in S} q(y_s|x_s) \tag{3}$$

We call the model in (3) the baseline model. It is the most commonly used model in the literature [4] [5].

4 Hidden Markov Model

4.1 Defining the model

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 fix the marginals of y for all the neighboring pixels couples. We use 4-neighbor system for simplicity in all that follows. For 2 neighboring pixels s and t, the expected 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. We assume that the model is isotropic, aggregating the cases where s and t are in vertical position to the cases where s and t are in horizontal position. Hence let us define the following constraints:

$$\begin{aligned} \mathcal{D} : \forall < s, t > \in S \times S,\ & p(y_s = 0, y_t = 0) = q(0, 0) \text{ and} \\ & p(y_s = 1, y_t = 1) = q(1, 1) \end{aligned} \tag{4}$$

where $< s, t >$ defines a couple of neighbor pixels.

The MaxEnt model under $\mathcal{C}_0 \cap \mathcal{D}$ is then the following Gibbs distribution:

$$p(x, y) \approx \prod_{s \in S} q(x_s|y_s) \exp[\sum_{<s,t>} (a_0(1 - y_s)(1 - y_t) + a_1 y_s y_t)] \tag{5}$$

Here and thereafter, the sign $\approx$ means equality up to a function that might depend on x but not on y. a_0 et a_1 are constant that must be set up such that the constraints are satisfied. The proof is in Appendix A. From (5) one then obtains the following model:

$$p(y|x) \approx \prod_{s \in S} q(x_s|y_s) p(y) \tag{6}$$

with

$$p(y) = \frac{1}{Z(a_0, a_1)} \exp[\sum_{<s,t>} (a_0(1 - y_s)(1 - y_t) + a_1 y_s y_t)] \tag{7}$$

where $Z(a_0, a_1)$ is a normalization function also known in statistical mechanics as the partition function:

$$Z(a_0, a_1) = \sum_y \{\exp[\sum_{<s,t>} (a_0(1-y_s)(1-y_t) + a_1 y_s y_t)]\} \tag{8}$$

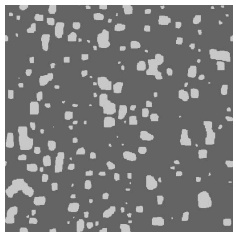
The model in equation (7) is known as a special case of a Potts model, see [7] and [13]. It is a Hidden Markov Model (HMM) if we consider y to be the hidden layer. This model is also simply referred to as a Markov Model elsewhere.

4.2 Parameter estimation

Parameter estimation in the context of MaxEnt is still an active research subject, especially in situations where even 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 [14] [15], mean field methods [16] [17], as well as Bethe Trees models [12] are among them. Another possibility is to use stochastic gradient as in [18]. Here we explore a related method based on the concept of Julesz ensembles defined in [19]. We learn from this work that one can sample an image from the model defined in (7) without knowing the parameters a_0 and a_1. This is true only in the asymptotic case of an infinite image but we will apply the result for a large image, say 512x512 pixels. In a second step, we use this sample image in order to estimate the parameters a_0 and a_1. This is done using the quantity $p(y_s = 1|y_{(s)})$ which is the probability to observe the label 1 at pixel s given all the other values y_t, for $t \in S$ and $t \neq s$. For the model in (7), this quantity can be easily analytically computed as

$$p(y_s = 1|y_{(s)}) = \phi((a_1 + a_0)n_s(1) - 4a_0) \tag{9}$$

where $\phi(x) = (1 + e^{-x})^{-1}$ is the sigmoïd (also known as logistic) function and $n_s(1)$ is the number of neighbors of s that take the label 1. This sum can take only five different values. For each one, the quantity $p(y_s = 1|y_{(s)})$ can be estimated from the sample image, leading to five linearly independent equations from which parameters a_0 and a_1 can be estimated. Now, returning to how to obtain a sample from the model in (7). The key idea which originated in statistical physics [20], is that the MaxEnt model we are looking for is, in an appropriate asymptotic meaning, the uniform distribution over the set of images that respect the constraints $\mathcal{D}$. Now, in the absence of phase transition, sampling from this set can be achieved numerically using simulated annealing, see [21].



	database values	image values
$Pr(Y_s = 0, Y_t = 0)$	0.828	0.827991
$Pr(Y_s = 1, Y_t = 1)$	0.159	0.151646

Fig. 1. **Top:** a sample image from the prior distribution used in the Hidden Markov Model. **Bottom:** probabilities estimated from the training set and from the image on the top.

Figure 1 shows a 512×512 sample of the prior model defined in equation (7). One can qualitatively appreciate how well it models skin regions. Notice that vertical and horizontal borders are preferred. This is a bias of the neighborhood system. Choosing 8 neighbors could improve it at the expense of computational load. The quantities $Pr(Y_s = y_s, Y_t = y_t)$, for neighboring pixels s and t are presented in Figure 1, first, as estimated from the training set, and secondly, as estimated from the image in the same Figure. The constraints are nearly respected. Parameter estimation from the image in Figure 1 leads to the numerical values: $a_0 = 3.76$ and $a_1 = 3.94$.

5 First Order Model

5.1 Defining the model

The baseline model was built in order to mimic the one pixel marginal of the joint distribution of color and skinness as observed on the database. Then, in building the HMM model we added constraints on the prior skinness distribution in order to smooth the model. 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 4-neighbor s and t, to match those observed in the training data. Hence we define the following constraints:

$$\begin{aligned} \mathcal{C}_1 : \forall < s,t > \in S \times S, \forall x_s \in C, \forall x_t \in C, \forall y_s \in \{0,1\}, \forall y_t \in \{0,1\}, \\ p(x_s, x_t, y_s, y_t) = q(x_s, x_t, y_s, y_t) \end{aligned} \tag{10}$$

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

Clearly, $\mathcal{C}_1 \subset (\mathcal{C}_0 \cap \mathcal{D}) \subset \mathcal{C}_0$. The solution to the MaxEnt problem under $\mathcal{C}_1$ is then, see Appendix A, the following Gibbs distribution:

$$p(x, y) \approx \exp[\sum_{<s,t>} \lambda(x_s, x_t, y_s, y_t)] \tag{11}$$

where $\lambda(s, t, x_s, x_t, y_s, y_t)$ are parameters that should be set up to satisfy the constraints. From (11), one gets

$$p(y|x) \approx \exp[\sum_{<s,t>} \lambda(s, t, x_s, x_t, y_s, y_t)] \tag{12}$$

Assuming that one color can take 256^3 values, the total number of parameters is $256^3 \times 256^3 \times 2 \times 2$. The previously mentioned parameter estimation methods clearly do not apply. In [12], the authors present a tree approximation to the pixel grid, called "Bethe tree", after the physicist H.A. Bethe who used trees in statistical mechanics problems. Bethe trees permit us to compute analytically an approximation of the parameters in the model (11) and consequently in (12) as we shall see now.

5.2 Parameter estimation and Bethe Tree Approximation

Bethe tree have been introduced in computer vision as a way of approximating estimators in Markov Random Field models in [12]. 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 follows: 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 added to the root node. This defines $\mathcal{T}_1^{(s)}$. 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 $\mathcal{T}_2^{(s)}$, and so on, see [12] 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 built with s, the neighbors of s and the neighbors of these. 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 2.

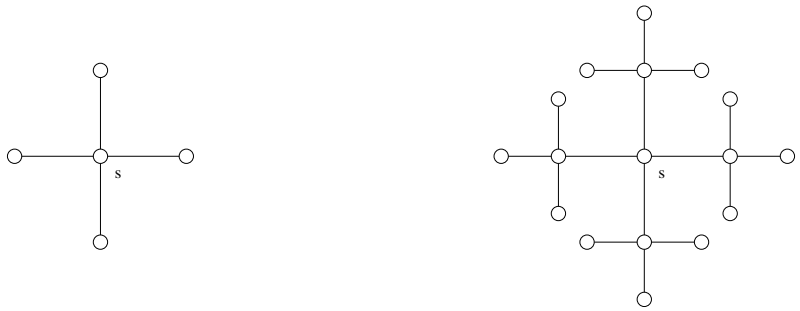


Fig. 2. **Left:** a Bethe tree of depth 1 rooted at s. **Right:** a Bethe tree of depth 2 rooted at s.

Let us consider the following model

$$\begin{aligned} &p(x,y) \approx \exp H(x;y) \text{ with} \\ &H(x;y) = \textstyle\sum_{<s,t>} \log q(x_s, x_t, y_s, y_t) - (n(s)-1) \sum_{s \in \mathring{S}} \log q(x_s, y_s) \end{aligned} \tag{13}$$

where $n(s)$ is the number of neighbors of s and $\mathring{S}$ is the set of interior pixels of S, that is the ones that have exactly four neighbors. First, remark that the model in (13) is a special case of model in (11). Second, under the Beth tree approximation, with arbitrarily finite depth, the model in (13) satisfies the constraints. Indeed, this is a particular case of a more general result, see [22], saying that any pairwise MRF defined on a tree graph can be written as a function of it's marginal distributions as in (13). We can then conclude that under the Bethe Tree approximation, (13) is the MaxEnt solution for $\mathcal{C}_1$.

Now, let us see how in practice one can use the model in (13). As for the HMM model, we fall back on the Markov Chain Monte Carlo algorithm. This requires to compute the conditional distribution of a label y_s given all the other labels and the image of the colors x. For $s \in \mathring{S}$, we obtain

$$\begin{aligned} p(y_s = 1 | y_{(s)}, x) &= \phi(U(x;y)) \text{ with} \\ U(x;y) &= \textstyle\sum_{t \in \mathcal{V}(s)} \log \frac{q(y_s=1, y_t | x_s, x_t)}{q(y_s=0, y_t | x_s, x_t)} - (n(s)-1) \log \frac{q(y_s=1|x_s)}{q(y_s=0|x_s)} \end{aligned} \tag{14}$$

6 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.

Each term of the product on the right side of (3) can be computed using probabilities estimated on the training data as follows using Bayes formula:

$$q(y_s|x_s) = \frac{1}{q(x_s)} q(x_s|y_s)q(y_s) \tag{15}$$

with

$$q(x_s) = \sum_{y_s=0}^{1} q(x_s|y_s)q(y_s)$$

Evaluation of the quantities in (15) is based on two 3-dimension histograms, $q(x_s|y_s = 1)$ and $q(x_s|y_s = 0)$ describing the one pixel color skin regions and non-skin regions respectively. Several authors have tried to get a parametric expression for these histograms as a mixture of Gaussian distribution [6] [1]. Our experience is that the Compaq Database is large enough so that crude histograms made with 512 color value per bin uniformly distributed do not over-fit. Each histogram is then made of 32^3 bins. The ROC curve for this model is presented in Figure 3. Experiments for this model, as well as for the other ones were made using the following protocol. The Compaq database contains about 18,696 photographs. It was split into two almost equal parts randomly. The first part, containing nearly two billion pixels was used as training data while the other one, the test set, was let aside for ROC curve computation. In Figure 4, first column displays test images. The second column displays grey level images. The grey-level is proportional to the quantity $p(y_s = 1|x)$ evaluated with the Baseline model. On the top image, skin pixels are not detected, especially on the neck of the rightmost person. On the bottom image, we notice many false positives. Figure 3 shows ROC curves computed from 100 images (around 10 millions pixels), randomly extracted from the test set. The Baseline model (with crosses) permit to detect more than 80% of the skin pixels with less than 10% of false positive rate.

6.2 Experiments HMM

For a new image x, skin detection requires to compute for each pixel the quantity $p(y_s|x)$. We use Markov Chain Monte Carlo. We generate, using the Gibbs sampler algorithm [7], a sequence of label images

$$y^1, y^2, \ldots, y^{n_0}, \ldots, y^n$$

Algorithm 1. Markov Chain Monte Carlo algorithm

$u \Leftarrow 0$
randomly initialize the binary image y^1
for $j = 1$ to $n - 1$ **do**
  $y \Leftarrow y^j$
  for all $s \in S$ **do**
    if $p(y_s = 1 | y_{(s)}, x) > 0.5$ **then**
      $y_s^{j+1} = 1$
    else
      $y_s^{j+1} = 0$
    end if
  end for
  if $j + 1 > n_0$ **then**
    $u \Leftarrow u + y^{j+1}$
  end if
end for
$u \Leftarrow u/(n - n_0)$

with stationary distribution (6). Then, we estimate the quantity $p(y_s|x)$ by the empirical mean

$$\frac{1}{n - n_0} \sum_{j=n_0+1}^{n} y_s^j$$

The Monte Carlo algorithm used in our experiments is presented in detail in Algorithm 1. Note that u and y are matrices defined on the pixel lattice S and

$$\begin{array}{ll} p(y_s = 1 | y_{(s)}, x) = \frac{p(y|x)}{\sum_{y_s} p(y|x)} = \phi(U(x; y)) \text{ with} \\ U(x; y) \qquad\qquad = \sum_{t \in \mathcal{V}(s)} (a_1 y_t - a_0 (1 - y_t)) + \log \frac{q(x_s | y_s = 1)}{q(x_s | y_s = 0)} \end{array} \tag{16}$$

where ϕ is the logistic function and $\mathcal{V}(s)$ are the neighbors of s. The algorithm is consistent in the sense that as $n \to +\infty, \forall s \in S, u_s \to p(y_s = 1|x)$, see [7].

Our working parameters are $n_0 = 1$ and $n = 100$. Three output images are presented in Figure 4. It compares favorably with the Baseline model. The skin zones detected with the baseline model are generally blended with background false alarms in complex images. The HMM outputs are cleaner with real skin zones emphasized. There is obvious misclassification of non-skin pixels as skin pixels on the dog of the third image for both models. The ROC curve in Figure 3 indicates an increase close to 2% in detection rate for the same false positive rate as the Baseline model. For example, setting 10% of false positive rate, the Baseline model permits to detect 81% of skin pixels in average, while the HMM permits to detect 83% in average. We show now that this is significative. The test set is made of 100 images disjoint from the training set. This amounts to about 10^7 pixels, out of which about 6% are labelled as skin. These 6×10^5 pixels cannot be considered as independent since the color values of the images

are correlated at small distance. Hence, we choose one out of ten of these pixels leading to a sample size of 6×10^4. The standard deviation around the Baseline value is then $\sqrt{0.81(1-0.81)} \times (\sqrt{6 \times 10^4})^{-1} \leq 2 \times 10^{-3}$. The hypothesis that the proportion of 83% was due to random fluctuations is then rejected with a p-value close to 0.

The running time of Algorithm 1 is as follows: there are $n-1$ loops over the image. During each loop, for each pixel, the conditional probability in (16) is evaluated once. The logarithmic operation as well as the logistic function can be tabulated. The labels of the four neighbors as well the color value have to be read from the current image. All these lead to 7 access to look-up tables and 4 additions. Hence the complexity of the algorithm is about $11 \times 100 \times |S|$ operations for an image made of $|S|$ pixels.

6.3 Experiments FOM

Now let us see how each term in (14) can be evaluated. First,

$$\frac{q(y_s = 1|x_s)}{q(y_s = 0|x_s)} = \frac{q(x_s|y_s = 1)}{q(x_s|y_s = 0)} \frac{q(y_s = 1)}{q(y_s = 0)} \tag{17}$$

and the quantities on the right side of (17) are easily obtained from the database as before. Second,

$$\frac{q(y_s = 1, y_t|x_s, x_t)}{q(y_s = 0, y_t|x_s, x_t)} = \frac{q(x_s, x_t|y_s = 1, y_t)}{q(x_s, x_t|y_s = 0, y_t)} \frac{q(y_s = 1, y_t)}{q(y_s = 0, y_t)} \tag{18}$$

Now the quantities on the right side of (18) involving the color values cannot be directly extracted from the database without drastic over-fitting since the histogram involved have a support of dimension six. Hence some kind of dimension reduction is needed.

One natural solution is to assume conditional independence, that is

$$\frac{q(x_s, x_t|y_s = 1, y_t)}{q(x_s, x_t|y_s = 0, y_t)} = \frac{q(x_s|y_s = 1)}{q(x_s|y_s = 0)} \tag{19}$$

The obtained model is then a HMM model, as in equation (6). Hence, Bethe tree method gives another way to estimate parameters a_0 and a_1. Obtained values are $a_0 = 3.94$ and $a_1 = 4$, which are close to the values obtained in section 4. The performances obtained with these values are not distinguishable

to the ones obtained previously, which give some indication of the robustness of the model.

A more promising dimension reduction procedure is the following approximation:

$$q(x_s, x_t|y_s, y_t) \sim q(x_s|y_s)q(x_t - x_s|y_s, y_t) \tag{20}$$

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 only. We use 32^3 bins of 512 colors each. Then we have:

$$\begin{aligned} U(x;y) &= \sum_{t\in\mathcal{V}(s)} \log \tfrac{q(x_s|y_s=1)q(x_t-x_s|y_s=1,y_t)q(y_s=1,y_t)}{q(x_s|y_s=0)q(x_t-x_s|y_s=0,y_t)q(y_s=0,y_t)} \\ &\quad -(n(s)-1)\log \tfrac{q(x_s|y_s=1)q(y_s=1)}{q(x_s|y_s=0)q(y_s=0)} \\ &= \sum_{t\in\mathcal{V}(s)} \log \tfrac{q(x_t-x_s|y_s=1,y_t)q(y_s=1,y_t)}{q(x_t-x_s|y_s=0,y_t)q(y_s=0,y_t)} + \log \tfrac{q(x_s|y_s=1)}{q(x_s|y_s=0)} \\ &\quad -(n(s)-1)\log \tfrac{q(y_s=1)}{q(y_s=0)} \end{aligned} \tag{21}$$

Experiments with this model are presented in Figures 3 and 4. The setup is the same as for the HMM. In Figure 4, one can visually appreciate the improvement in localization of the skin zones compared to the HMM. The detected skin regions are more precise. It is easier to recognize the shapes of the faces and hands than with the HMM results. The mouth of the right hand character in the first image is not detected as skin, as well as the eyes in the second image or the mustache in the third image.

Bulk results in the ROC curve of Figure 3 show an improvement of performance of around 1%. At 10% of false positive rate, the HMM permits to detect around 83% of skin pixels and the First Order Model around 84%. This is evaluated in the same setting as described in Section 6.2. In particular, the number of independent skin pixels is around 6×10^4. The standard deviation around the HMM value is then $\sqrt{0.83(1-0.83)} \times (\sqrt{6 \times 10^4})^{-1} \leq 2 \times 10^{-3}$. The hypothesis that the proportion of 84% was due to random fluctuations is then rejected with a p-value close to 0.

Another to compare classification algorithms over multiple thresholding values is to compute the area under the roc curve (AUC). Using $[.04; .11]$ for integration interval, the normalized AUC, that is, the AUC divided by the length of the interval of integration is .79 for the baseline model, .81 for HMM and .82 for FOM confirming the results obtained above for a single false positive rate.

The running time for the FOM can be evaluated in the same way as HMM. The only difference is the operations involved in the $U(x;y)$ function in (21). As for the HMM, the logarithmic operation as well as the logistic function can

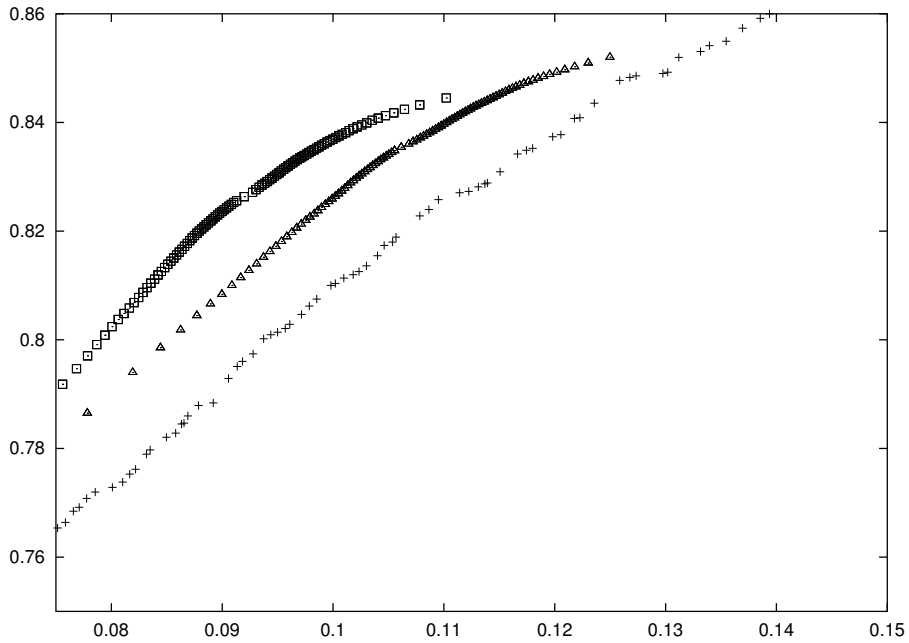


Fig. 3. Receiver Operating Characteristics (ROC) curve for each model. x-axis is the false positive rate, y-axis is the detection rate. Baseline model is shown with crosses, HMM model with triangles, while the First Order Model is shown with squares.

be tabulated. The values of the current pixel and its four neighbors have to be read from the current image. All these lead for an image to about 15 access to look-up tables and 30 additions/subtractions and 1 multiplication. Hence the complexity of the algorithm is about $46 \times 100 \times |S|$ operations which is about 4 times the running time of the HMM. As an example, using a PC with a Pentium 4 processor at 1.7 Ghz and 256 MB memory, the processing time for a 100×100 pixels image is .008 seconds for the baseline model, 1.3 seconds for the HMM and 2.3 seconds for the FOM.

7 Conclusions

We have considered a sequence of three models for skin detection built from a large collection of labeled images. For a given color image, such a model puts weight on binary images defined on the same pixel grid. Each model is a maximum entropy model with respect to constraints. These constraints concern marginal distributions. Our models are nested. The first model, called the baseline model is well known from practitioners. Pixels are considered as independent. Performance, measured by the ROC curve on the Compaq database is impressive for such a simple model. However, single image examination reveals very irregular results. The second model is a Hidden Markov Model. It includes constraints that force smoothness of the solution. The ROC curve

Fig. 4. **First column:** original color images. The image on top is 225×180 pixels . The image on the bottom is 541×361 pixels. **Second column:** Baseline model. **Third column:** hidden Markov model. **Fourth column:** First Order Model. In the computed images, the grey level is proportional to the skin probability evaluated with the specified model.

obtained shows an increase in detection rate from 81% to 83% for the same false positive rate of 10%. Finally, color gradient is included in the set of constraints. Thanks to Bethe tree approximation, we obtain a simple analytical expression for the coefficients of the associated MaxEnt model. The resulting detection rate increases to 84%. The same qualitative behavior is observed when comparing the area under the ROC curve.

For many applications involving skin detection as an intermediate stage, processing time is of major importance. In future work we plan to replace the stochastic sampling algorithm by a deterministic scheme as Mean Field method [16] or Belief Propagation [23] method in order to meet the required time constraints.

Detailed examination of the pictures reveals that the discussed models are still far from reaching human performances. For example, the left arm of the right-most person in the first image of Figure 4 is visible in the baseline model and not in the subsequent ones. Remark that the grey values indicating the probability for skin are very low. A zoom is provided in Figure 5. It is understandable that the regularizing models, HMM as well as the First Order Model, operating at the level of pixels, have produced a posterior probability that put very low likelihood for skin in this region. Indeed, the local evidence for skin is low and the neighboring values are also indicating low evidence. A high level model of limbs might be able to overcome these difficulties.

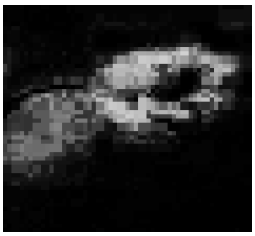


Fig. 5. Zoom of top row, second image in Figure 4. Result of the Baseline model

8 Acknowledgments

We would like to thank the reviewing work. It has helped in improving the overall quality of the paper.

A Appendix

Here we shall derive a MaxEnt solution for the joint distribution $p(x, y)$ under the constraints $\mathcal{C}_0$. See (1).

Remark that the constraints in (1) are expectations with respect to p. Indeed,

$$p(x_s, y_s) = E_p[\delta_{x_s}(X_s)\delta_{y_s}(Y_s)] \tag{A.1}$$

with

$$\delta_a(b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ if } a \neq b \end{cases}$$

Then, following Jaynes' argument [9], the MaxEnt solution under $\mathcal{C}_0$ is unique if it exists, and can be obtained using Lagrange multipliers. One gets:

$$p(x, y) = \exp(\lambda_0 + \sum_{s \in S} \lambda(s, x_s, y_s)) \tag{A.2}$$

Where the parameters λ should be set up such that the constraints are satisfied. Now if

$$\forall x_s \in C, \forall y_s \in \{0, 1\}, q(x_s, y_s) > 0 \tag{A.3}$$

then one can choose

$$\lambda_0 = 0 \text{ and } \lambda(s, x_s, y_s) = \log q(x_s, y_s) \tag{A.4}$$

which leads to the unique solution of the MaxEnt problem:

$$p(x, y) = \prod_{s \in S} q(x_s, y_s) \tag{A.5}$$

Condition in (A.3) is saying that there is no empty bin in the empirical joint histogram $q(x_s, y_s)$. This will be our case. MaxEnt solutions still exist when (A.3) is not verified.

Here we shall obtain a MaxEnt solution for the joint distribution $p(x, y)$ under $\mathcal{C}_0 \cap \mathcal{D}$, see (1) and (4).

As for $\mathcal{C}_0$, the constraints in $\mathcal{D}$ are expectations. Indeed,

$$\forall y_s \in \{0,1\}, \forall y_t \in \{0,1\}, p(y_s, y_t) = E_p[\delta_{y_s}(Y_s)\delta_{y_t}(Y_t)] \tag{A.6}$$

Using once more Lagrange multipliers, one obtains that the MaxEnt solution, if it exists, is

$$\begin{aligned} p(x,y) &= \exp H(x,y,\lambda_0,\lambda_1,\lambda_2,\lambda_3) \text{ with} \\ H(x,y,\lambda_0,\lambda_1,\lambda_2,\lambda_3) &= \lambda_0 + \textstyle\sum_{s\in S} \lambda_1(s,x_s,y_s) + \\ &\quad \textstyle\sum_{<s,t>\in S\times S} \lambda_2(s,t)(1-y_s)(1-y_t) + \\ &\quad \textstyle\sum_{<s,t>\in S\times S} \lambda_3(s,t) y_s y_t] \end{aligned} \tag{A.7}$$

where $< s, t >$ is a couple of 4-neighbors pixels and λ_0, λ_1, λ_2, λ_3 define parameters that should be set up such that the constraints are satisfied. Starting from (A.7),remark that

$$p(x_s, y_s) = \sum_{x_t; t\in S, t\neq s} \sum_{y_t; t\in S, t\neq s} p(x,y) = \exp[\lambda_0 + \lambda_1(s, x_s, y_s)] g(s, y_s) \tag{A.8}$$

with $g(s, y_s)$ a function that doesn't depend on x_s. Now,

$$p(y_s) = \sum_{x_s} p(x_s, y_s) = \exp[\lambda_0] g(s, y_s) \sum_{x_s} \exp[\lambda_1(s, x_s, y_s)] \tag{A.9}$$

hence

$$p(x_s|y_s) = \frac{p(x_s, y_s)}{p(y_s)} = \frac{\exp[\lambda_1(s, x_s, y_s)]}{\sum_{x_s} \exp[\lambda_1(s, x_s, y_s)]} \tag{A.10}$$

Since $p(x, y)$ lies in $\mathcal{C}_0$, it verifies: $p(x_s|y_s) = q(x_s|y_s)$. Assuming positivity (A.3), we can choose

$$\lambda_1(s, x_s, y_s) = \log q(x_s|y_s) \tag{A.11}$$

Now, constraints in $\mathcal{D}$, see (4), do not depend on the location $< s, t >$. Hence, one can reduce to translation invariant models as in (5).

Constraints in $\mathcal{C}_1$, see (10) are also expectations. Indeed,

$$p(x_s, x_t, y_s, y_t) = E_p[\delta_{(x_s)}(X_s)\delta_{(x_t)}(X_t)\delta_{(y_s)}(Y_s)\delta_{(y_t)}(Y_t)] \tag{A.12}$$

Using Lagrange multipliers, one obtains (11).

References

[1] J.-C. Terrillon, M. N. Shirazi, H. Fukamachi, 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, 2000, pp. 54–61.

[2] J. Z. Wang, J. Li, G. Wiederhold, O. Firschein, System for screening objectionable images, Images, Computer Communications Journal 21 (15) (1998) 1355–1360.

[3] J. Z. Wang, J. Li, G. Wiederhold, 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 (1998) 113–124.

[4] J.-C. Terrillon, M. David, 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, 1998, pp. 112–117.

[5] M. J. Jones, J. M. Rehg, Statistical color models with application to skin detection, Tech. Rep. CRL 98/11, Compaq (1998).

[6] M. Jones, J. M. Rehg, Statistical color models with application to skin detection, in: Computer Vision and Pattern Recognition, 1999, pp. 274–280.

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

[8] R. Chellappa, A. Jain (Eds.), Markov Random Fields: Theory and Applications, Academic Press, 1996.

[9] E. Jaynes, Probablity theory: The logic of science., http://omega.albany.edu:8008/JaynesBook, chapter 11.

[10] Cover, Thomas, Elements of Information Theory, Wiley, 1991, chapter 11.

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

[12] C. Wu, P. C. Doerschuk, Tree approximations to markov random fields, IEEE Trans. on PAMI 17 (4) (1995) 391–402.

[13] G. Cross, A. Jain, Markov random field texture models, IEEE Trans. on PAMI 5 (1) (1983) 25–39.

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

[15] F. Divino, A. Frigessi, Penalized pseudolikelihood inference in spatial interaction models with covariates, Scandinavian Journal of Statistics 27 (3) (2000) 445–458.

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

[17] G. Celeux, F. Forbes, N. Peyrard, Em procedures using mean field-like approximations for markov model-based image segmentation, Pattern Recognition 36 (1) (2002) 131–144.

[18] L. Younes, Estimation and annealing for gibbsian fields, Annales de l'Institut Henry Poincaré, Section B, Calcul des Probabilités et Statistique 24 (1998) 269–294.

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

[20] A. Martin-Lof, The equivalence of ensembles and gibbs'phase rule for classical lattice-systems, Journal of Statistical Physics 20 (1979) 557–569.

[21] S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Trans. on PAMI 6 (1984) 721–741.

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

[23] J. S. Yedida, W. T. Freeman, Y. Weiss, Understanding belief propagation and it's generalisations, Tech. Rep. TR-2001-22, Mitsubishi Research Laboratories (January 2002).