A Tree Distribution for Skin Detection

Sanaa EL FKIHI, Mohamed DAOUDI and Driss ABOUTAJDINE

Abstract— **Skin detection consists in detecting human skin pixels from an image. Skin detection plays an important role in various applications such as face detection, searching and filtering image content on the web. In this paper we propose a novel skin detection algorithm based on tree distribution. A tree distribution that is more general then a bayesian network, can represent a joint distribution in an intuitive and efficient way. We assess the performance of our method on the Compaq database.**

I. 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.

Skin detection plays an important role in various applications such as face detection [1], searching and filtering image content on the web [2][3][4]. Research has been performed on the detection of human skin pixels in color images by use of various statistical color models [5][6]. Some researchers have used skin color models such as Gaussian, Gaussian mixture or histograms [7][8]. 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.

In this paper we are interested in two aspects. First, we will present a reasonably quick skin detection algorithm that outperform the standard method. Second, we will learn the dependencies or the structure between pixels within a skin patch and use it for classification. To achieve this goal, in this work we will use a tree distribution method developed in [10] that we assume more general than a bayesian network [11] and represents a joint distribution in an intuitive and efficient way.

The paper is organized as follows: in section II, we introduce the notation that will be used throughout the paper and we present the features used. Section III details our tree distribution classifiers model. Section IV is devoted to experiments and comparisons with alternative method. Finally, the conclusion and some perspectives are given in section V.

II. NOTATIONS AND METHODOLOGY

Let's fix the notations. The set of pixels of an image is S. We consider the RGB color space, the color of a pixel $s \in S$ is x_s. It is a 3 dimensional vector, each component being coded on one octet. (i_s, j_s) is the coordinate of s. 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. In order to take into account the neighboring influence between pixels, we define the following neighborhood system :

$$V_s = \{(i,j)/|i - i_s| < a, |j - j_s| < a\}; s \in S, a \in \mathbb{N} \quad (1)$$

Thus, we consider a vector of observations X which stands for an image patch ($k \times k, k = 2a - 1$). We consider a vector $B = (x_1, x_2, \ldots, x_{k^2})$ decomposed until a low-level

This work was supported by CNRS under MathStic project and the The 'comite mixte Inter- Universitaire franco-marocain' AI MA/02/46

Mohamed DAOUDI FOX-MIIRE Research Group (LIFL UMR USTL-CNRS 8022) (GET/INT - ENIC Telecom Lille 1) daoudi@enic.fr

Sanaa EL FKIHI and Driss ABOUTAJDINE GSCM (Group Signaux Communications and Multimedia) Faculty of Sciences Rabat University MohammedV elfkihi@enic.fr and aboutaj@fsr.ac.ma

element; we note the resultant vector $X = (x_1, x_2, \ldots, x_n)$ where $n = 3k^2$.

Let's assume for a moment that we knew the joint probability distribution $r(X, y_s)$ of the vector (X, y_s), then Bayesian analysis tells us that, whatever cost function the user might think of, all that is needed is the a-posterior distribution $r(y_s|X)$.

From the user's point of view, the useful information is contained in the one pixel marginal of the a-posterior probability, that is, for each pixel, the quantity $r(y_s = 1|X)$, quantifying the belief for skinness at pixel s. In practice the model $r(X, y_s)$ 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 each other with the distribution $r(X, y_s)$. The collection of samples is referred later as the training data.

Our objective is to construct a probabilistic classifiers that represent the posterior probabilities $r(y_s = 1|X)$ and $r(y_s = 0|X)$ of skinness at pixel s given the patch, using a single tree distribution. To simplify notation throughout the paper, we use $r(X|y_s = 1) = p(X)$ and $r(X|y_s = 0) = q(X)$.

Let's consider a complete non oriented graph $G(V, E)$ corresponding to X; each element x_u of X is a vertex. V and E are the sets of vertices and edges. We note that x_u and u refer to the same vertex. Two neighbor vertices x_u and x_v are noted $u \sim v$. Now, the task is to find a non oriented acyclicgraph (tree) modelling $r(X, y_s)$.

III. Tree distribution

We consider a probabilistic classifier that represent the a-posteriori probability by using tree models. Let us assume for now that the graph G was a tree: that is a connected graph without loops which we note T.

Proposition 1: [12]

$$r_T(x) = \prod_{(u\sim v)\in T} \frac{r_{uv}(x_u, x_v)}{r_u(x_u) r_v(x_v)} \prod_{u\in V} r_u(x_u) \qquad (2)$$

where $r_u(x_u)$ is one-vertex marginal of r and $r_{uv}(x_u, x_v)$ is it's two-vertices marginal, defined as:

$$r_u(x_u) = \sum_{x_v; v\in V; v\neq u} r_T(x) \qquad (3)$$

$$r_{uv}(x_u, x_v) = \sum_{x_u; u\in V; u\neq u,v} r_T(x) \qquad (4)$$

$$1 = \sum_{x_u; u\in V} r_T(x) \qquad (5)$$

We note n_u the number of the neighbors of vertex x_u; it follows $\prod_{u\sim v} r_u(x_u) r_v(x_v) = \prod_{u\in V} r_u^{n_u}(x_u)$. The Eq. (2) becomes :

$$r_T(x) = \prod_{(u\sim v)\in T} r_{uv}(x_u, x_v) \prod_{u\in V} r_u^{1-n_u}(x_u) \qquad (6)$$

A. Learning of tree distribution

The learning problem is formulated as follows: given a set of observations $X = \{x_1, x_2, ..., x_n\}$ we want to find one tree T in which the distribution probability is efficient for two different classes: skin and non skin.

Thus, to deal with the problem, we propose to maximize the Kullback-Leibler divergence[13] between two probability mass functions $p(x)$ and $q(x)$ corresponding to two different classes. Therefore, we give the following statement.

Statement 1: Probabilities distributions of tree dependence $p_T(\mathbf{x})$ and $q_T(\mathbf{x})$ are respectively the optimum approximations to the true probabilities $p(\mathbf{x})$ and $q(\mathbf{x})$ if and only if :

1) their dependence tree T has the maximum weight.
2) the number of the neighbors of each vertex is the minimum.

Proof. We assume that there exists a tree T in which each vertex is a variable in $X = (x_1, x_2, ..., x_n)$; T models two distributions p and q approximated by p_T and q_T according to (6).
The Kullback-Leibler divergence (KL) between p_T and q_T is :

$$KL(p_T, q_T) = \sum_x p_T(x) \log \frac{p_T(x)}{q_T(x)}$$

$$= \sum_x p_T(x) \log p_T(x) - \sum_x p_T(x) \log q_T(x) \qquad (7)$$

On the one hand

$$\sum_x p_T(x) \log p_T(x) = \sum_x p_T(x) *$$

$$\log \left\{ \prod_{(u\sim v)\in T} p_{uv}(x_u, x_v) \prod_{u\in V} p_u^{1-n_u}(x_u) \right\}$$

$$= \sum_x p_T(x) \sum_{(u\sim v)\in T} \log p_{uv}(x_u, x_v) +$$

$$\sum_x p_T(x) \sum_{u\in V} (1 - n_u) \log p_u(x_u) \qquad (8)$$

On the other hand

$$\sum_x p_T(x) \log q_T(x) = \sum_x p_T(x) \sum_{(u\sim v)\in T} \log q_{uv}(x_u, x_v) +$$

$$\sum_x p_T(x) \sum_{u\in V} (1 - n_u) \log q_T(x_u) \qquad (9)$$

From (8) - (9), Eq. (7) becomes

$$KL(p_T, q_T) = \sum_x p_T(x) \sum_{(u\sim v)\in T} \log \frac{p_{uv}(x_u, x_v)}{q_{uv}(x_u, x_v)} +$$

$$\sum_x p_T(x) \sum_{u\in V} (1 - n_u) \log \frac{p_T(x_u)}{q_T(x_u)} \qquad (10)$$

Moreover,

$$\sum_x p_T(x) \sum_{(u\sim v)\in T} \log \frac{p_{uv}(x_u,x_v)}{q_{uv}(x_u,x_v)} =$$

$$\sum_{(u\sim v)\in T} \sum_{u,v\in V} p_{uv}(x_u,x_v) \log \frac{p_{uv}(x_u,x_v)}{q_{uv}(x_u,x_v)}$$

$$= \sum_{(u\sim v)\in T} KL(p_{uv},q_{uv}) \tag{11}$$

and

$$\sum_x p_T(x) \sum_{u\in V} (1-n_u) \log \frac{p_u(x_u)}{q_u(x_u)} =$$

$$\sum_{u\in V} \sum_{u\in V} (1-n_u) p_u(x_u) \log \frac{p_u(x_u)}{q_u(x_u)}$$

$$= \sum_{u\in V} (1-n_u) KL(p_u,q_u) \tag{12}$$

Thus, we obtain :

$$KL(p_T,q_T) = \sum_{(u\sim v)\in T} KL(p_{uv},q_{uv}) \quad +$$

$$\sum_{u\in V} (1-n_u) KL(p_u,q_u) \tag{13}$$

Since, for all $x_u \in V$, the $KL(p_u,q_u)$ are independent of the dependence tree and KL divergence is non-negative, maximizing the closeness measure $KL(p_T,q_T)$ is equivalent to maximizing the total branch weight $\sum_{(u\sim v)\in T} KL(p_{uv},q_{uv})$ and minimizing the number of the neighbors of each vertex n_u. Proof concluded.

The optimal solution of the problem (maximizing the total branch weight of tree dependence and minimizing the number of the neighbors of each vertex) as formulated is not trivial [14]. In this paper we focused on maximizing the total branch weight of tree dependence :

$$T^* = argmax_T \sum_{(u\sim v)\in T} KL(p_{uv},q_{uv}) \tag{14}$$

In order to give a more detailed description of our model, we present the following procedure (1):

Procedure 1:

- Input : Dataset $\left\{\left(x^{(1)},y^{(1)}\right),\cdots,\left(x^{(n)},y^{(n)}\right)\right\}$

1) Fix $a \in \mathbb{N}$ to define the neighborhood system of a pixel s as :
$$V_s = \{(i,j)/|i-i_s|<a, |j-j_s|<a\}; s \in S$$
2) Consider a bloc B $(k\times k)$ where $k = 2a-1$. Build the vector of observables X such as each element of X is a sample.
3) Build a complete non oriented graph $G(V,E)$ corresponding to X; each element of X is a vertex.
4) Let x_u and x_v be two different vertices. Use the maximum-likelihood estimator to compute the two vertices-marginal $p_{uv}(x_u,x_v)$ and $q_{uv}(x_u,x_v)$ of p_T and q_T as
$$p_{uv}(x_u=i,x_v=j) = f^1_{ij}(x_u,x_v)$$
$$q_{uv}(x_u=i,x_v=j) = f^0_{ij}(x_u,x_v)$$
Where, for $(m=1,0)$, $f^m_{ij}(x_u,x_v)$ is the sample joint frequency of $x_u=i$ and $x_v=j$ such as theirs labels are 1 or 0.
5) Compute the simple KL divergence $KL(p_{uv},q_{uv})$, for all x_u and x_v, as
$$KL(p_{uv},q_{uv}) = \sum_{u,v\in V} p_{uv}(x_u,x_v) \log \frac{p_{uv}(x_u,x_v)}{q_{uv}(x_u,x_v)}$$
6) Build a maximum weighted spanning tree (MWST)[15].

- Output : The tree T in which the distribution probability is efficient for two different classes.

B. Inference

We would like to compute the state of the pixel y_s, according to the states of the pixels of the patch X. By applying the Bayes' rule, we obtain:

$$r(y_s=j|X) = \frac{r(y_s=j)r(X|y_s=j)}{r(X)} \quad , \quad j=0,1. \tag{15}$$

Moreover,

$$r(X) = \sum_{y_s=0}^{1} r(X,y_s) = \sum_{i=0}^{1} r(X|y_s=i)r(y_s=i)$$

In which

$$r(X|y_s=0) \approx \frac{\prod_{(u\sim v)\in T} q_{uv}(x_u,x_v)}{\prod_{u\in V} q_u(x_u)^{(n_u-1)}} = q_T(X)$$

$$r(X|y_s=1) \approx \frac{\prod_{(u\sim v)\in T} p_{uv}(x_u,x_v)}{\prod_{u\in V} p_u(x_u)^{(n_u-1)}} = p_T(X)$$

$$r(y_s=0) \approx q_T(y_s=0) \qquad r(y_s=1) \approx p_T(y_s=1)$$

Therefore

$$r(y_s=0|X) \approx$$

$$\frac{q_T(y_s=0)q_T(X)}{p_T(y_s=1)p_T(X)+q_T(y_s=0)q_T(X)} \tag{16}$$

And

$$r(y_s=1|X) \approx$$

$$\frac{p_T(y_s=1)p_T(X)}{p_T(y_s=1)p_T(X)+q_T(y_s=0)q_T(X)} \tag{17}$$

All the elements of Eq. (16) and Eq. (17) are previously computed in Step (4) of our algorithm.

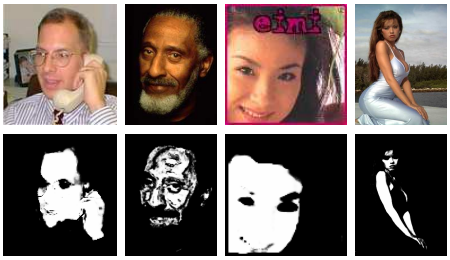

Fig. 1. **Top :** original color images. **Bottom :** results of our algorithm model.

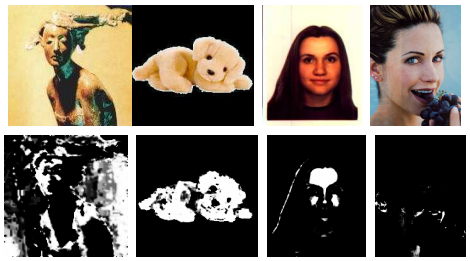

Fig. 2. **Top :** original images. **Bottom :** the corresponding tree distribution model. The first two images show false positive pixels, while the other two columns show false negative pixels

IV. Skin Detection 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.

In our skin detection application we define the neighborhood system of a pixel in which $a = 2$. Thus the image patch considered is 3×3. However, we consider RGB space, therefore the size of the vector observables X is 27. 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 bins. The experiments are presented in figures 1 and 3.

Bulk results in the ROC curve of Figure 3 show an improvement of performance around $1,33\%$. At $4,74\%$ of false positive rate, the baseline which is an independent model[5][8] permits to detect around $74,84\%$ of skin pixels and the tree distribution model around $76,17\%$. Figure 2 shows some cases where our detector failed. It is due to over-exposure or to skin-like color.

Another way to compare classification algorithms over multiple thresholding values is to compute the area under the roc curve (AUC). Using $[0; .075]$ for integration interval, the normalized AUC, that is, the AUC is 0.030 for the baseline model, 0.048 for our approach confirming the results obtained above for a single false positive rate.

V. Conclusions and Future Works

In this paper, we presented a new algorithm based on tree distribution model for skin detection. By making some assumptions, we propose a tree model which maximizes a

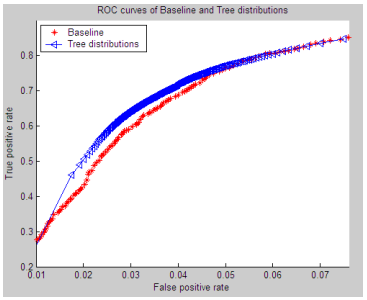

Fig. 3. Receiver Operating Characteristics (ROC) curve for each model. x-axis is the false positive rate, y-axis is the detection rate which is the complement to one of the false negative rate. The baseline model is shown with read crosses and the tree distribution with blue triangles.

Kullback-Leibler divergence between a skin distribution and non skin distribution. Performance measured by the ROC curve on the Compaq database shows an increase in detection rate from 1% to $5,5\%$ for the same false positive rate of the tree distribution comparing to the baseline model which suppose the independence of pixels.

This approach could be used to classify other textures. In future work we apply our skin detection tree distribution scheme to block adult images from Internet.

References

[1] 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*, 2000, pp. 54–61.

[2] 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*, vol. 21/15, pp. 113–124, 1998.

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

[4] H. Zheng, M.Daoudi, and B. Jedynak, "Blocking adult images based on statistical skin detection," *Electronic Letters on Computer Vision and Image Analysis*, vol. 4, no. 2, pp. 1–14, December 2004.

[5] B. Jedynak, H. Zheng, and M. Daoudi, "Statistical models for skin detection," 2003. [Online]. Available: citeseer.ist.psu.edu/jedynak03statistical.html

[6] B. Jedynak, H. Zheng, and M.Daoudi, "Skin detection using pairwise models," *Image and Vision Computing*, vol. 23, no. 13, pp. 1122–1130, November 2005.

[7] 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*, 1998, pp. 112–117.

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

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

[10] M. Meila and M. I. Jordan, "Learning with mixtures of trees." *Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2000.

[11] N. Sebe, I. Cohen, T. S. Huang, and T. Gevers, "Skin detection : A bayesien network approach," August 2004.

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

[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, August 1991.

[14] J. Kleinberg and E. Tardos, *Algorithm Design*, A. Wesley, Ed., 2005.

[15] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, no. IT-14 (3), pp. 462–467, 1968.