

Skin detection in images using machine learning techniques for color and texture recognition

Stefan Sebastian

April 9, 2018

Contents

1	Theoretical background	2
1.1	Related work	2
1.2	Image segmentation	2
1.3	Detection by color	2
1.3.1	Color spaces	2
1.3.2	Explicit thresholding	4
1.3.3	Skin Probability Map	5
1.3.4	Gaussian classifiers	6
1.4	Detection by texture	7
2	Application development	8
3	Conclusions	9

Chapter 1

Theoretical background

1.1 Related work

1.2 Image segmentation

1.3 Detection by color

Skin pixel detection by color means classifying a pixel while considering only its color features. A first step in applying this approach is selecting a color space.

1.3.1 Color spaces

A color space, also called a gamut, represents a set of colors in a way that is independent of the medium in which they are represented (computer screens, cameras, magazines, etc)[1]. The $L^*a^*b^*$ color space contains all colors that can be seen by the human eye, however most color spaces are smaller due to technical limitations. I will present some of the color spaces which have been used successfully to classify skin pixels.

RGB

To start with, RGB is one of the most popular color spaces for working with image data. It matches the color sensitive receptors of the human eye (red, green, blue) and started as a convenient way to represent the colored

rays used by CRT screens[2]. While this model is simple to use it has the disadvantage of mixing chrominance and luminance features[2].

Normalized RGB

Normalized RGB is a color space with a lighter memory consumption than RGB and its components are calculated as follows[2]:

$$r = \frac{R}{R + G + B}, g = \frac{G}{R + G + B}, b = \frac{B}{R + G + B}. \quad (1.1)$$

The third value can be determined from the other 2 so we can avoid storing it. Other advantages according to [3] include reduced differences caused by illumination and ethnicity, and lower variance of skin color clusters than in the normal RGB space.

HSI, HSV, HSL

HSI, HSV, HSL represent perceptual color spaces and they describe the hue, saturation and intensity (or value, lightness). These color spaces are used because they provide invariance to ambient lighting and surface orientation relative to the source of light[3]. We can convert to HSV from RGB using the following formulas[2]:

$$H = \arccos \frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{((R - G)^2 + (R - B)(G - B))}} \quad (1.2)$$

$$S = 1 - 3 \frac{\min(R, G, B)}{R + G + B} \quad (1.3)$$

$$V = \frac{1}{3}(R + G + B) \quad (1.4)$$

YCbCr

Orthogonal color spaces, which YCbCr is a member of, provide chrominance and luminance separation as they represent colors with statistically independent components. YCbCr is mostly used by European television studios and in image compression[2]. Y represents luma (or luminance) and Cb, Cr are

the blue and red difference chroma components and they can be computed as follows:

$$Y = 0.299R + 0.587G + 0.114Bs \quad (1.5)$$

$$Cb = B - Y \quad (1.6)$$

$$Cr = R - Y \quad (1.7)$$

Having such a simple transformation and a clear separation of the luminance component makes the YCbCr a popular choice for skin detection models.

1.3.2 Explicit thresholding

This is one of the simplest skin-color models that can be built. The method aims to define, through the use of simple rules and thresholds, the boundaries of skin clusters in a specific color space. It has been observed in [4] that the colors of human skin tend to cluster in small regions of the color space and human skin pixels differ more in intensity than in color.

In RGB

An example using the RGB space, from Peer et al.[5] which has been integrated into a face detection system consists of the rules below:

$$\begin{aligned} R > 95 \quad \text{and} \quad G > 40 \quad \text{and} \quad B > 20 \quad \text{and} \\ \max\{R, G, B\} - \min\{R, G, B\} > 15 \quad \text{and} \\ |R - G| > 15 \quad \text{and} \\ R > G \quad \text{and} \quad R > B \end{aligned} \quad (1.8)$$

In YCrCb

A set of thresholds for YCrCb space where proposed by Chai and Ngan[6]. They set the ranges for Cb from 77 to 127 and for Cr from 133 to 173 and worked with the ECU database.

In YIQ

Dai and Nakano[7] created a model for YIQ, an orthogonal color space, which only used the I component (which stands for in-phase). The range they provided was [0, 50], however most of the images in their databases were of people with yellow skin.

In normalized RGB

An interesting approach, proposed by Gomez and Morales[8], is having a learner find these rules. They use RCA, a constructive induction algorithm, to build rules expressed with simple arithmetic operations in the rgb space. Their method achieves better results than the Bayesian SPM on their dataset, however it is computationally slower. RCA stands for Restricted Covering Algorithm which resembles a general covering algorithm with the restriction of trying to build a single rule for each class (in this case, a rule for skin detection). The strategy implemented for RCA was finding attributes which cover either a large number of true positives or a few false positives. The starting attributes were r, g, b and the constant 1/3, which would generate new attributes using the operators : +, *, - and squaring. One of their best and simplest generated models looks like this:

$$\begin{aligned} \frac{r}{g} &> 1.185 \quad \text{and} \\ \frac{r * b}{(r + g + b)^2} &> 0.107 \quad \text{and} \\ \frac{r * g}{(r + g + b)^2} &> 0.112 \end{aligned} \tag{1.9}$$

In comparison with the C4.5 decision tree algorithm, the RCA method obtained slightly worse results but with much simpler rules.

1.3.3 Skin Probability Map

A SPM represents a histogram with multiple bins. Each bin stands for a color or a subset of colors and has a value equal to the probability of holding skin colored pixels. When building a SPM you must choose the color space and the number of bins per color channel.

In order to determine whether a given pixel is a skin colored pixel we apply Bayes' theorem. Here is the form used by Jones and Rehg[9] in one of

the most popular papers on statistical skin detection:

$$P(skin|p) = \frac{P(p|skin)P(skin)}{P(p|skin)P(skin) + P(p|\neg skin)P(\neg skin)} \quad (1.10)$$

In this equation "p" is the notation for the occurrence of the given pixel. Therefore $P(skin|p)$ means the probability of observing skin given our pixel. To determine whether we classify the pixel as skin we compare our probability with the threshold value, β .

$$P(skin|p) > \beta \quad (1.11)$$

The probability of observing skin can be computed as the ratio of the number of skin pixels to the total number of pixels observed in training.

$$P(skin) = \frac{T_S}{T_S + T_N} \quad (1.12)$$

Jones and Rehg[9] made the observation that given even a large training set most pixels are never seen. They explored their dataset of approximately 2 billion pixels and came to the conclusion that around 77% of the RGB space is empty. This suggests that we might get better results by reducing the number of bins per channel. We can observe that a small perturbation in the RGB values of a pixel results in a very similar color. Consequently if we classify p as a skin colored pixel then there is a high probability that its neighbors in the color space are skin colored pixels too. This observation is in support of a smaller number of bins for the SPM histogram.

1.3.4 Gaussian classifiers

Gaussian classifiers are parametric skin distribution models with the advantages of being compact, therefore using less memory than SPMs, and able to generalize better using less training data[3].

Single Gaussian

Considering the observations from the thresholding chapter that skin color pixels tend to cluster in a region of the color space we can model that distribution using an elliptical Gaussian joint probability density function, an example of which was provided by [3]:

$$p(c) = \frac{1}{2\pi^{\frac{1}{2}}|\Sigma|^{\frac{1}{2}}} \cdot e^{-\frac{1}{2}(c-\mu)^T\Sigma^{-1}(c-\mu)} \quad (1.13)$$

In this equation, c is the color vector, μ the mean vector and Σ the covariance matrix. These can be calculated from the training data as follows:

$$\begin{aligned}\mu &= \frac{1}{n} \cdot \sum_{j=1}^n c_j \\ \Sigma &= \frac{1}{n-1} \cdot \sum_{j=1}^n (c_j - \mu)(c_j - \mu)^T\end{aligned}\tag{1.14}$$

Here c_j are all the color samples used in training the model. In order to establish that the given color describes skin we can compare $p(c)$ with a threshold that can be determined experimentally for a given training set.

Gaussian Mixture Models

A Gaussian mixture model are a form of unsupervised learning used to identify subpopulations within an overall population, provided they are normally distributed[10].

1.4 Detection by texture

Chapter 2

Application development

Chapter 3

Conclusions

Bibliography

- [1] A. Frich, “Color management guide.” <https://www.color-management-guide.com/color-spaces.html/>. Accessed: 2018-04-07.
- [2] V. Vezhnevets, V. Sazonov, and A. Andreeva, “A survey on pixel-based skin color detection techniques,” *GRAPHICON03*, pp. 85–92, 2003.
- [3] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, “A survey of skin-color modeling and detection methods,” *Pattern Recognition*, vol. 40, pp. 1106–1122, 2007.
- [4] J. Yang, W. Lu, and A. Waibel, “Skin-color modeling and adaptation,” *ACCV98*.
- [5] J. Kovac, P. Peer, and F. Solina, “Human skin colour clustering for face detection,” *EUROCON*, 2003.
- [6] D. Chai and K. N. Ngan, “Face segmentation using skin-color map in videophone applications,” *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 9, no. 4, pp. 551–564, 1999.
- [7] Y. Dai and Y. Nakano, “Face-texture model based on sgld and its application in face detection in a color scene,” *Pattern Recognition*, vol. 29, no. 6, pp. 1007–1017, 1996.
- [8] G. Gomez and E. F. Morales, “Automatic feature construction and a simple rule induction algorithm for skin detection,” *Proceedings of Workshop on Machine Learning in Computer Vision*, pp. 31–38, 2002.

- [9] M. J. Jones and J. M. Rehg, “Statistical color models with applications to skin detection,” *International Journal of Computer Vision*, vol. 46, pp. 81–96, 2002.
- [10] J. McGonagle, V. Tembo, and A. Chumbley, “Gaussian mixture model.” <https://brilliant.org/wiki/gaussian-mixture-model/>. Accessed: 2018-04-09.