

# A New Approach of Face Detection-based Classification of Image Databases

**Szabolcs Sergyán**

Budapest Tech

Bécsi út 96/B, H-1034 Budapest, Hungary, sergyan.szabolcs@nik.bmf.hu

---

*Abstract: In the case of using content-based image retrieval algorithms is very important that the image database let be classified in certain sense. One possible way of classification is the face detection based classification.*

*In this paper a face detection algorithm is presented which was used in color images, and some classification methods and these usage for the classification of image database are reviewed.*

*Keywords: face detection, image classification, image retrieval*

---

## 1 Introduction

Color is a low-level cue for object detection that can be implemented in a computationally fast and effective way for locating objects. It also offers robustness against geometrical changes under a stable and uniform illumination field. In some cases, it can clearly discriminate objects from a background. Therefore, its popularity in machine vision applications is most understandable. Unfortunately, color information is very sensitive to changes in illumination. Therefore, several strategies have been developed to cope with such changes. One strategy is to use only the chromaticity coordinates of a color space which properly separate chromaticity from intensity. Although this works quite well with intensity-related light changes, it does not cancel out illumination chromaticity variations. In order to eliminate light changes due to illumination chromaticity, a mass of color constancy algorithms have been suggested, but so far they have not produced reliable results for machine vision purposes, except in very limited cases. Instead of correcting colors or canceling illumination changes, some other work aims at developing approaches that can tolerate or can be adapted to changes. Such methods are yielding promising results, as will be demonstrated in this paper.

An earlier version of this paper was presented in [7].

## 2 Color Cue and Facial Image Analysis

The properties of the face pose a very difficult problem for facial image analysis: a face is a dynamic and nonrigid object that is difficult to handle. Its appearance varies due to changes in pose, expressions, illuminations, and other factors, such as age and makeup. As a consequence, most of the facial analysis tasks generally involve heavy computations such as color or motion, in order to assist and accelerate the analysis. These additional cues also offer an indication of the reliability of the face analysis results: the more the cues appearance-based face detection, an exhaustive scan (at different locations and scales) of the images is conducted when searching the face [11]. However, when the color cue is available, one can reduce the search regions by preprocessing the images and selecting only the skin-like areas. Therefore, it is not surprising that the color of skin has been commonly used to assist face detection.

Let us consider the general block diagram of face analysis shown in Fig. 1. The color cue is involved at different stages. In the first stage, the color images are acquired and pre-processed. The preprocessing may include gamma correction, color space transformation, and so on. It is often preferable to get rid of the dependencies on lighting intensity as much as possible. The perfect case would be to also cancel-out the effect of the illuminant color (by defining a color representation that is only a function of the surface reflectance), but so far, this has not been achieved in machine vision. The human visual system is superior in this sense, because human visual perception in which the color is perceived by the eye depends quite significantly on surface reflectance, although the light reaching the eye is a function of surface reflectance, illuminant color, and lighting intensity.



Figure 1

A general block diagram of face analysis that shows different phases of facial image analysis

Among the different stages shown in Fig. 1, the use of color in face detection is probably the most obvious. It is generally used to select the skin-like color regions. Then, simple refining procedures can be launched to discriminate the faces from other skin-like regions, such as hands, wood, etc. Thus, much faster face detectors are generally obtained when the color cue is considered.

Using the fact that some facial features, such as eyes, are darker than their surrounding regions, holes should then appear in the face area when labeling the skin pixels. Such an observation is commonly exploited when detecting facial features in color images [2], [3], [11].

Does color information contribute to face recognition? The answer to this question is not obvious, although some studies have suggested that color also plays a role in face recognition, and this contribution becomes evident when the shape cues are degraded [12].

### 3 Skin Color Modeling

When starting to build a skin color model, three issues should be taken into account: the first is the selection of the color spaces for modeling; the second is the illumination conditions and camera calibration; the last is the selection of a mathematical model. All these issues affect the effectiveness and optimality of the use of a skin color cue.

#### 3.1 Color Spaces for Skin

Skin color modeling has been applied to several color spaces. So far, none of the color spaces has been shown to be clearly superior to the others, and it might well be that the optimal color space depends on the data. Those color spaces that do not separate intensity properly from chromaticity are not so different from RGBs. Color spaces, that do separate intensity from chromaticity differ much more.

One popular choice for the color space is the NCC. This color space and CIE  $xy$  have been shown to be more efficient for face detection than several other color spaces and to produce more transferable skin color models between different cameras [9].

NCC effectively cancels intensity information from chromaticity information. The uniform intensity level changes in a pixel are effectively eliminated in chromaticity coordinates, because they are calculated by dividing the descriptor value of the channel by the sum of all descriptor values at the pixel:

$$r = \frac{R}{R + G + B} \quad (1)$$

$$g = \frac{G}{R + G + B} \quad (2)$$

$$b = \frac{B}{R + G + B} \quad (3)$$

In NCC, a color is uniquely defined by its intensity information and two chromaticity coordinates, because  $r+g+b=1$ . Therefore, one can use only the two chromaticity coordinates of NCC to scope with illumination intensity changes. The intensity changes are typically common in videos and images, because the skin areas can be located from deep shadows to bright lighting. This strategy is commonly used in skin detection, because it is easy to do. Of course, some models include intensity, but it is questionable if this information is useful, because the amount of data needed to construct a reliable model with all possible intensity levels is huge, and the third component increases the amount of computation.

## 3.2 Skin Color Model and Illumination

The color appearance of skin depends on the illumination in two ways: by the prevailing illumination and by the illumination used in white balancing. This dependency is camera specific: the camera sensors and internal image preprocessing of the camera affect the color production and thus the end results. Therefore, creating a universal model is difficult.

Most of the research published so far concentrates on the situation in which the images have canonical or near canonical color appearance. This is not bad approach if the image data are acquired mostly under these kinds of conditions, because the canonical skin colors overlap quite well. This kind of image data set can include personal photos, which are most likely taken under ideal conditions. However, this approach will only work on certain fixed cases.

If the illumination conditions are not fixed, the previous approaches are most likely to fail. Of course, it would be possible to use suggested by Hsu et al. [3]. The basic idea is to correct the image for segmentation. The color correction is based on finding pixels with high brightness value. They are assumed to belong to a white object on the image, and they are used for calculating the correction coefficient. An often used color normalization process can be found in [6].

Another approach was presented Cho et al. [1], who suggested an adaptive skin color filter. The basic idea is to first roughly select the possible skin color range and the fine-tune to find the face. The initial color range was determined from a set of images, but no so assumptions about the color distribution, such as the assumption that the facial skin color histogram is unimodal. It can also fail if the face is not the dominant skin-colored object on the image.

To build a better model, one should determine the possible illumination variation, white balancing of the camera, and camera settings and use this information to create the model. This is the basic idea behind the skin locus-based approach [8]: to understand the behavior under different conditions. When the behavior of skin is known, it is possible to exclude particular nonskin tones – however, this is not the case with simple constraint. The model can be constructed either by simulations or by actually taking images under different conditions. One should note that the skin loci of different cameras may not overlap very well.

The skin locus-based models generally consist of more color tones than the canonical color models. This means that more nonskin objects will be considered skin candidates by the locus model than by the canonical model. This is the price one pays for the generality of the model. On the other hand, color is rarely enough to determine whether the target is skin or not, and canonical models tolerate illumination changes very poorly.

### 3.3 Mathematical Models for Skin Color

The mathematical skin color modeling can be approached from two different perspectives according to whether one can attach a probability for color tones or not. The first approach assumes that skin is more likely to appear with certain color tones or provide higher discrimination between backgrounds and skin. The second one presumes that the probability of a certain color varies between images or that skin colors have equal appearance probability.

The challenge of the probability-based approach is to be able to reliably find the probability of color tones. This requires one to collect a representative data set of images. So far, a statistical reliable color model for skin and nonskin Internet images was presented in [4]. They calculated the histogram and Gaussian models using over 1 billion labeled pixels. In addition to the statistical model, one has to determine the threshold limit for separating skin from nonskin areas. It is difficult to automatically find the threshold value, because the probability model found may not be valid for all images.

The second approach uses a spatial constraint in the color space to define possible skin areas. The shape of the constraint can be made up of simple thresholds like in [1] or a more complex shaped function like in [3]. No thresholding is needed, because there were no assumptions about the distribution of skin color in images. This can sometimes be overgenerous: all pixels within the constraint are labeled as skin.

## 4 Color Cue in Face Detection

As mentioned above, color is a useful cue for face detection, as it can greatly reduce the search area by selecting only the skin-like regions. However, it is obvious that the use of skin color only is not enough to distinguish between faces and other objects with a skin-like appearance. Therefore, other procedures are needed to verify whether the selected regions are faces or not. Depending on the robustness of the skin model and changes in the illumination conditions, one can notice two cases:

Case 1: The initial skin color detection step produces consistently reliable results. The skin color model is valid for the illumination conditions, the camera, and its settings. The skin color model can be designed either for stable, controlled illumination or for variable illumination. In such cases, it is generally enough to consider each connected resultant component from the skin detection as a face candidate. Then, one can verify the "faceness" of the candidate by simple and fast heuristics.

Case 2: The initial skin color detection step produces unsatisfactory results or even fails. In this case, the skin color model does not correspond to the prevailing illumination, camera used, or settings of the camera. One may hope that the results would indicate the locations of the faces, but their size estimation is too unreliable. Therefore, a different method for face detection or a different skin color model should be used when searching the faces in and around the detected skin regions.

In both cases, the use of color accelerates the detection process. In the following, we review some methods based on color information for detecting faces.

## 4.1 Overview of Color-based Face Detection Methods

Most color-based face detectors start by determining the skin pixels that are then grouped using connected component analysis. Then, for each connected component, the best fit ellipse is computed using geometric moments, for example. The skin components that verify some shape and size constraints are selected as face candidates. Finally, features are searched for inside each face candidate based on the observation that holes inside the face candidate are due to these features being different from skin color.

Therefore, most of the color-based face detection methods mainly differ in the selection of the color space and the design of the skin model. In this context many methods for skin modeling in different color spaces have been proposed.

Among the works using color for face detection is Hsu et al.'s system that consists of two major modules: face localization for finding face candidates and facial feature detection for verifying detected face candidates [3]. For finding the face candidates, the skin tone pixels are labeled using an elliptical skin model in the YCbCr color space, after applying a lighting compensation technique. The detected skin tone pixels are iteratively segmented using local color variance into connected components that are then grouped into face candidates. Then, a facial feature detection module constructs eye, mouth, and face boundary maps to verify the face candidates. Good detection results have been reported on several test images. However, no comparative study has been made.

A case study of the above-mentioned face detection algorithm can be found in [5].

## 5 Pattern Classification

Pattern classification [10] involves taking the features extracted from the image and using them to automatically classify image objects. This is done by developing classification algorithms that use the feature information. The distance or similarity measures are used for comparing different objects and their feature

vectors. In this section we will define pertinent terms, conceptually discuss the most widely used current methods, and look in detail at some of the basic algorithms.

The simplest algorithm for identifying a sample from the test set is called the Nearest Neighbor method. The object of interest is compared to every sample in the training set, using a distance measure, a similarity measure, or a combination of measures. The “unknown” object is then identified as belonging to the same class as the closest sample in training set. This is indicated by the smallest number if using a distance measure, or the largest number if using a similarity measure. This process is computationally intensive and not very robust.

The Nearest Neighbor method can be made more robust by selecting not just the closest sample in the training set, but by consideration of a group of close feature vectors. This is called the  $K$ -Nearest Neighbor method, where, for example,  $K=4$ . The unknown feature vector is assigned to the class that occurs most often in the set of  $K$ -Neighbors. This is still very computationally intensive, since each unknown sample has to be compared to every sample in the training set, and the training set is wanted as large as possible to maximize vectors.

We can reduce this computational burden by using a method called Nearest Centroid. Here, we find the centroids for each class from the samples in the training set, and then we compare the unknown samples to the representative centroids only. The centroids are calculated by finding the average value for each vector component in the training set.

To identify an unknown sample, we need only compare it to these two representative vectors, not the entire training set. The comparison is done using any of the distance or similarity measures. With a distance measure, the distance between the unknown sample vector and each class centroid is calculated and it is classified as the one it is closest to the one with the smallest distance. With a similarity measure, the similarity between the unknown sample vector and each class centroid is calculated and it is classified as the one it is closest to the one with the largest similarity.

The technique of comparing unknown vectors to classified vectors, and finding the closest one, is also called template matching. In addition to applying this technique to feature vectors, template matching can be applied to the raw image data. A template is devised, possibly via a training set, which is then compared to subimages by using a distance or similarity measure. Typically, a threshold is set on this measure to determine when we have found a match, that is, a successful classification. This may be useful for applications where the size and orientation of the objects is known, and the objects shapes are regular.

Bayesian theory provides a statistical approach to the development of a classification algorithm. To apply Bayesian analysis we need a complete statistical model of the features, and usually normalize the features so that their distribution

is standard normal density. The Bayesian approach provides a classifier that has an optimal classification rate. This means that the boundaries that separate the classes provide a minimum average error for the samples in the training set. These boundaries are called discriminant functions.

In practice, the feature space is typically larger than two-dimensional, and since the three-dimensional form of a linear function is a plane, the  $n$ -dimensional form of a linear function is called a hyperplane. In general, discriminant functions can also be quadratic, or take an arbitrary shapes. Generally, the forms are limited to those that can be defined analytically via equations, such as circles, ellipses, parabolas, or hyperbolas. In  $n$ -dimensional vector spaces these decision boundaries are called hyperquadrics.

Neural networks represent another category of techniques used for pattern classification. Neural networks are meant to model the nervous system in biological organisms, and are thus referred to as artificial neural networks. Mathematical models have been developed for these biological systems, based on a simple processing element called a neuron. These neurons function by outputting a weighted sum of the inputs, and these weights are generated during the learning or training phase. The element's output function is called the activation function, and the basic types are: (1) the identify function, the input equals to output, (2) a threshold function where every input greater than the threshold value outputs a 1, and less than the threshold outputs 0, (3) a sigmoid function, an S-shaped curve, which is nonlinear and is used most often. Single layer networks are limited in their capabilities and nonlinear functions are required to take advantage of the power available in multilayer networks.

The neural network consists of the input layer, the output layer, and possibly hidden layers. The main distinguishing characteristics of the neural network are: (1) the architecture, which includes the number of inputs, outputs, and hidden layers, and how they are all connected, (2) the activation function, which is typically used for all processing elements in a layer, but this is not required, and (3) the learning algorithm. Many learning algorithms have been developed, but they all work by inputting the training vectors, calculating an error measure and the adjusting the weights to improve the error. The training continues for a specific number of iterations, specified by the user, or until a specified error criterion is reached. To achieve optimal results with neural networks it is important that the feature vectors are preprocessed with a PCT to decorrelate the input data. Neural networks have been used successfully in many applications, for example in the recognition of hand written characters.

## **Conclusions**

Color is a useful cue in facial image analysis. Its use for skin segmentation and face detection is probably the most obvious, while its contribution to face recognition is not very clear. The first important issues when planning the use of color in facial image analysis are the selection of a color space and the design of a



skin model. Several approaches were proposed for these purposes, but unfortunately, there is no optimal choice. Choice depends on the requirement of the application and also on the environment.

Once a skin model has been defined, the contribution of color to face detection, not surprisingly, plays an important role in preprocessing the images and in selecting the skin-like areas. Then, other refining stages can also be launched in order to find faces among skin-like regions.

In relation to the contribution of color to face recognition, our conclusion is that it makes sense for automatic face recognition, systems not to rely on color for recognition because its contribution is not well established.

### References

- [1] K. M. Cho, J. H. Jang, K. S. Hong, "Adaptive Skin-color Filter," *Pattern Recognition*, Vol. 34, pp. 1067-1073, 2001
- [2] A. Hadid, M. Pietikainen, B. Martinkauppi, "Color-based Face Detection Using Skin Locus Model and Hierarchical Filtering," *16<sup>th</sup> International Conference on Pattern Recognition*, Quebec, Canada, pp. 196-200, 2002
- [3] R. L. Hsu, M. Abdel-Mottaleb, A. K. Jain, "Face Detection in Color Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, pp. 696-706, 2002
- [4] M. Jones, J. Rehg, "Statistical Color Models with Application to Skin Detection," *International Journal on Computer Vision*, Vol. 46, pp. 81-96, 2002
- [5] B. Martinkauppi, A. Hadid, M. Pietikainen, "Color Cue in Facial Image Analysis," *Color Image Processing (ed. R. Lukac, K. N. Plataniotis)*, CRC Press, pp. 285-308, 2007
- [6] D. Paulus, L. Csink, H. Niemann, "Color Cluster Rotation," *International Conference on Image Processing*, pp. 161-165, 1998
- [7] Sz. Sergyán, "Classification of Image Databases Using Face Detection," in *Proceedings of 6<sup>th</sup> International Symposium on Intelligent Systems and Informatics*, Subotica, Serbia, September 26-27, 2008, CD Issue
- [8] M. Soriano, B. Martinkauppi, S. Huovinen, M. Laaksonen, "Adaptive Skin Color Modeling Using the Skin Locus for Selecting Training Pixels," *Pattern Recognition*, Vol. 36, No. 3, pp. 681-690, 2003
- [9] 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," *4<sup>th</sup> IEEE International Conference on Automatic Face and Gesture Recognition*, IEEE Computer Society, pp. 54-61, 2000

- [10] S. E. Umbaugh, "Computer Imaging Digital Image Analysis and Processing", *CRC Press*, pp. 285-291, 2005
- [11] M. H. Yang, D. J. Kriegman, N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, pp. 34-58, 2002
- [12] A. W. Yip, P. Sinha, "Contribution of Color to Face Recognition," *Perception*, Vol. 31, No. 8, pp. 995-1003, 2002