Face Detection in Color Image using Rectangle Feature

Jun-Dal Lee and Maw-Kae Hor
National Cheng-Chi University, Taiwan
g9136@cs.nccu.edu.tw, hor@cs.nccu.edu.tw

Abstract

We propose a rectangle feature set utilizing multiple color space information for human face detection. Images of RGB color space can be converted into normalized RGB and HSV color spaces and thus the interference of lighting conditions can be reduced. Base on this mechanism, we define 8 rectangle features inside 4 selected color spaces and use a boosting algorithm to select the important features. Experimental results show that the detectors constructed with our approach are able to process nearly one million sub-windows within 2.4 seconds. The selected color features are mostly in Hue domain. These results show the classifying ability of multiple color spaces analysis as well as the possibility to develop a faster system based on the color features.

1: Introduction

Human activity is a major concern in various applications such as video surveillance, automatic camera control, and human computer interface. To detect and recognize human faces is one of the remaining challenges in the human activity related application nowadays. Different approaches have been developed to solve this problem under different environments and conditions. In our interest, we want to solve this problem based on the color information.

Our concept is from the detecting frame proposed in [1] .Before Haar-like features was proposed, Oren et al. were looking for an image representation "which captures the relationship between average intensities of neighboring regions". The word "intensity" can be interpreted into the pixel value of the original image. When speaking of color images, this value and its difference of neighboring regions is not a constant value and varies with the applied color space (also called the *color model* or the *color system*).

Feature in different color space is sensitive to different visual pattern. A good choice of color space is considered to be crucial for color-based feature detection. A wide variety of the color spaces have been applied to human skin-color related applications in different researches, but few papers have provided a strict justification of their color space choice. However, previous researches and surveys [3] have shown that the performance of skin-color modeling is lowly dependent to the color space choice, especially in non-parametric method. Since our approach is one of the non-parametric method, a choice better than average is good enough for our system..

Meanwhile, Stern and Efros obtained an experimental result of adaptively switching the color space when tracking faces in a multi-color lighting environment in 2002 [4]. Judging from the data, the authors have provided evidences showing that, among five chromaticity planes (including RG, normalize RGB, HS, YQ, and CrCb), the normalized RGB and HS planes performed almost equally and much better than the others. This result implies that an over-complete feature set from normalized RGB plane and HS plane might be sufficient for our system.

We propose a set of new color features to utilize the color information in human face detection. Then apply the AdaBoost Algorithm developed in [2]. With this framework, we show the classifying advantages by applying color features to face detection techniques.

2: Rectangle Feature

Our face detection procedures classify images based on the value of simple features. These simple features are called *Rectangle Features*. Each rectangle feature is a threshold function constructed from thresholds and rectangle filters.

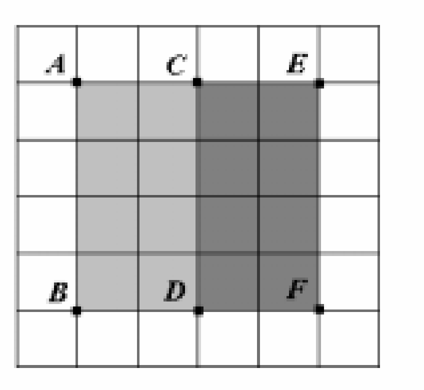

Figure 1: A rectangle feature.

For example, consider a 4×4 rectangle filter as Figure 1 shows. In an image bitmap, we applied this 4×4 filter on the rectangle ***ABFE*** and obtain the feature value *F* by a simple formula as following:

$$F\,(\boldsymbol{ABFE}) = S\,(\boldsymbol{ABDC}) - S\,(\boldsymbol{CDFE})$$

where *S* is means the sum of the pixel values in the specified rectangle. Compare this feature value with one or two thresholds (decided by its bounding mechanism), we can determine the presence of this feature.

2.1 Haar-like Feature

The concept of Haar-like feature is proposed by Oren et al. [1] in 1990's. This concept has being extended with more dimensions and orientations in other systems. The Haar wavelets form a natural set of basis functions which encode differences in average intensity between adjacent regions [11]. Haar-like features works in a similar way to Haar wavelets: they express the relationship between adjacent regions with the difference of the sum of pixel values. In Figure 1, the rectangle feature here is a two-region feature consisting region ***ABDC*** and region ***CDFE***. The feature value of a two-region feature is computed as the sum of the brighter region subtracts the sum of the darker region, as indicated in previous section. The simplicity of this formula implies that we can obtain a huge amount of feature values in short time.

2.2 DC Color Feature

We evaluate the plane color feature according to the DC component which is defined in Discrete Fourier Transform (DFT):

$$F(0,0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

F is the average pixel value of $f(x, y)$ if it is an image. Combined with the haar-like features, 8 basic rectangle features used in our system are illustrated in Figure 2. ***a*** to ***g*** are Haar-like features and ***h*** is the smallest DC color feature. These basic features can be modified with different scales and alignments in both vertical and horizontal directions. With the modifications, 8 features can generate over 1 million graphical features in a 24×24 pixel window.

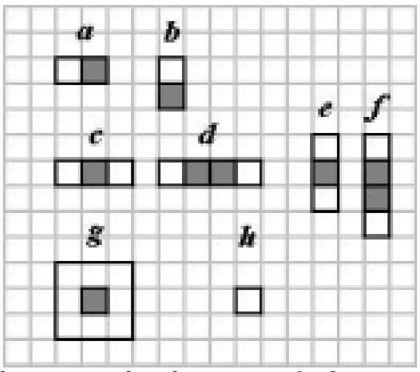

Figure 2: 8 basic rectangle features.

3: Color Space

Color space is a specification of a coordinate system where each color is represented by a single point. By the definition of rectangle feature, the physical feature value changes with the applied color space. A wide variety of color spaces have been applied to the skin-color related application [7] [9]. In [8], normalized RGB color space and HSV color space has shown themselves to be the better color spaces for skin-color modeling among 5 color spaces: RG, normalized RGB, HS, YQ, and CrCb. With this literal evidence, we applied the defined rectangle features to the normalized RGB color space and HSV color space.

3.1: Normalized RGB Color Space

The normalized RGB color space (also noted as *rgb* color space) is an improved version of RGB color space. To reduce the brightness effect on each component, we normalize the components as following:

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}$$

This representation reduces the dimensionality of RGB color space. As the sum of the three normalized component is known, the third component *b* can be represented with *r* and *g*, which means it holds no significant information and can be omitted. So a normalized RGB is in fact a two-dimensional color space and can be expressed in a plane, in our case, an *r-g* plane.

The normalized RGB color space has two notable properties: 1. the dependency of *r* and *g* on the brightness is greatly diminished by the normalization, and 2. with conditions satisfied, normalized RGB color space is invariant to changes of surface orientation related to the light source [10]. With regarding to these properties, normalized RGB color space is chosen to be one of our concerned color spaces.

3.2: HSV Color Space

HSV is another well-known color space. Each component of HSV can be evaluated from RGB with a conversion function as following:

$$H = \begin{cases} \theta & if\ B \le G \\ 360 - \theta & if\ B > G \end{cases},$$

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R, G, B)},$$

and $V = \max(R, G, B)$,

, where $\theta = \cos^{-1}\left\{ \frac{\frac{1}{2}[(R-G)+(R+B)]}{[(R-G)^2 + (R-B)(G-B)]^{\frac{1}{2}}} \right\}$

From its nature, HSV color space has the property of explicit discrimination between luminance and chrominance. This property made the HSV color space become widely popular in the researches on color related applications, such as skin-color modeling [9]. In a more constrained environment, the Hue component can be invariant to changing lighting conditions because of its low dependence to brightness [10]. Additionally, [6] [13] show that HSV has some advantages on skin-color modeling.

Considering of both advantages and disadvantages, we propose to apply rectangle features to HS domain in our system. We modify the Hue value range by -180º then remap the Hue function to the range of [0º, 360º], for the reason that red is located at value 180º after modified and thus make our classifier to avoid a threshold near 0º.

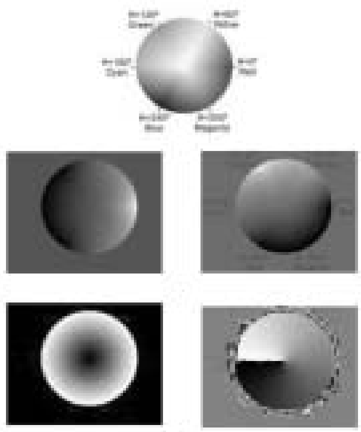

Figure 3: Color conversion.

Figure 3 shows the results of color conversion. In the order of top-left to bottom-right, they are: normalized R, normalized G, Saturation, and Hue.

4: Learning Algorithm

Base on the rectangle features, we can train our classifiers with one of any known learning algorithms. Here we introduce the detecting frame work developed by Viola and Jones for its processing speed [12].

4.1: Weak Classifier

Recall that each rectangle feature is a threshold function, our very basic classifier called "*weak classifier*" is constructed accordingly. We define the meaning of notations as: classification function h_j, basic feature modifications of position *P*, scale *S*, and color space *C*. Haar-like feature f_j has threshold θ_j and parity p_j to form its classification function, and DC color feature c_j has 2 bounding thresholds ω_j and ω_j' in its classification function. Therefore these threshold functions can be expressed as:

$$h_{haar}(x \mid P,S,C) = \begin{cases} 1 & if \; p_j f_j(x \mid P,S,C) < p_j \theta_j \\ 0 & otherwise \end{cases}$$

$$h_{DC}(x \mid P,S,C) = \begin{cases} 1 & if \; \omega_j < c_j(x \mid P,S,C) < \omega_j' \\ 0 & otherwise \end{cases}$$

We can adjust the thresholds to obtain almost any desired true positive (TP) and false positive (FP) rates. -our system, we set the threshold and the parity equal to the value which classifies the most training examples into proper categories.

Most weak classifiers obtained in this way have a detection rate just slightly better than random guess. We are running a boosting algorithm to improve them.

4.2: AdaBoost Algorithm

The whole weak classifier set is too large for detection and not all of them are necessary. Here we apply the AdaBoost Algorithm to select the fittest features and improve them.

AdaBoost Algorithm is used to select the most significant features and to discard the lesser important features. Every selected feature (and its related weak classifier) is given a classifying weight according to its error rate. Several weighted classifiers are combined and form a *strong classifier* and can be expressed as the following:

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$

where *T* is the number of weak classifiers and α_t is its corresponding weight.

4.3: Cascade Classifier

When new training examples are included, we can repeat the AdaBoost to obtain a new strong classifier. By adding false positive examples into the training set, the latter constructed classifier can help eliminating false positives from previous classifier. Thus, we can align the strong classifiers into a cascade structure, as Figure 4 shows.

In the cascade structure, we arrange simple features in the earlier stages to reject the majority of negative sub-images. Simple features consist of lesser classifiers and thus can be processed and passed easily. In this way we can greatly reduce the computational time because the evaluation for the following classifiers does not operate on the majority of input sub-images. Only positive instances which pass the early stage will trigger the more complicated classifiers and then be evaluated. Generally, those complicated classifiers involve more features and have lower false positive rates.

Details of implementation are not described here. More comprehensive information about this detecting frame work can be found in [12].

5: System Structure

Figure 5 shows the constructing flow of our system. We train weak classifiers with our personal collected examples, then run AdaBoost algorithm to boost these weak classifiers. Finally, we construct the detecting frame work for face detection.

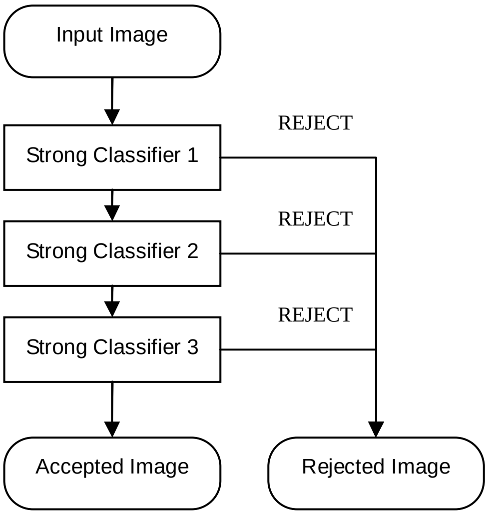

Figure 4: Cascade Classifier

6: Experiment

The total number of weak classifiers we train is list in Table 1. Our detector is constructed from these classifiers. The first detector we construct is a 4-layers detector, consist 46 modified rectangle features. Color spaces selected by AdaBoost are list in Table 2.

We could notice that the features in Hue domain are much more likely being picked by the AdaBoost, which means that the Hue component has more classifying power on human face detection. Face detection results are shown in Figure 6, with computational time varying between 1.6 second to 2.4 seconds (due to image sizes). Figure 7 shows a special result about detecting human faces under low lighting conditions. In Figure 7, we could notice that the face regions are enhanced in different color spaces.

Table 1: The total number of rectangle features.

Index	Size	Scale	Feature number
a, b	2×1, 1×2	12×24, 24×12	345,600
c, e	3×1, 1×3	8×24, 24×8	200,800
d, f	4×1, 1×4	6×24, 24×6	158,400
g	3×3	8×8	33,856
h	1×1	24×24	360,000
			Total 1,098,656 features

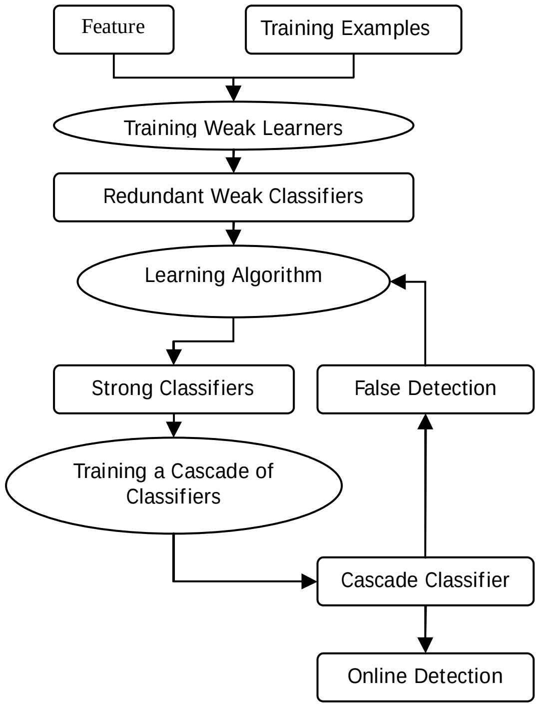

Figure 5: Construction flow of our system.

Table 2: Color space selected.

Layer	Red	Green	Hue	Saturation	Total
#1	0	0	1	0	1
#2	4	2	3	1	10
#3	2	0	7	1	10
#4	3	1	16	5	25
Total	9	3	27	7	46

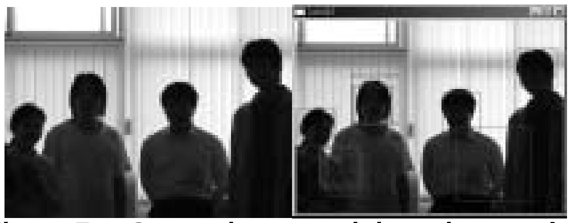

Figure 7-a: Source image and detection result.

Figure 7-b: Color conversions. From top-left to bottom right are: normalized Red, normalized

Green, Hue, and Saturation.

7: Conclusion

We have shown a possibility of constructing a face detection system for color images using a set of rectangle features in different color spaces. Based on these features, we developed a face detection system using non-parametric method that can achieve better accuracy with more training examples. Compare with previous gray-image detecting systems, our system utilizes different color information and, hence, could enhance the detecting ability of the previous systems.

We can construct a general frame work for tracking objects in color images. Combined with AdaBoost Algorithm, this non-parametric framework can perform various detecting task efficiently. The classifying power of color information is higher than gray-scale images and worth of practical implementation. Normalized RGB has good lighting invariant properties, so as HSV. Additionally, skin-color has little difference in Hue color space. Combining these advantages, a face tracking application based on color information techniques is more attractive than other applications. The new technique even has the capability to work under dimmed lighting conditions.

The experiment results show some robustness of our system. Faces can be found even under bad lighting conditions. However, since additional information is processed, our system is not great in the processing speed. It takes about 1.6 seconds to complete the detection of a 320x240 image. Unlike conventional gray-scale face detections, our system takes one more step to convert the color information through double-precision computations. Thus requires more time than the integer computations used in conventional gray-scale techniques. One might implement color conversions into approximate integer calculation to reduce the computation time. However, the cost of lower accuracy has to be considered. Another possible improvement is to train weak classifiers into two fashions: one for fast rejection and the other for precise detection. The cascade model could be constructed accordingly. Though the detection rate and speed is still not satisfying for now, these problems can be solved with further refinements. Better results could be expected in the future.

Reference

[1] Michael Oren , Constantine Papageorgiou, Pawan Sinha, Edgar Osuna, and Tomosa Poggio, “Pedestrian Detection Using Wavelet Templates”, *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[2] Paul Viola, Michael Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features”, *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001

[3] Vladimir Vezhnevets Vassili Sazonov Alla Andreeva, “A Survey on Pixel-Based Skin Color Detection Techniques”, *Proceedings of Graphicon-2003*, pp. 85-92 , 2003.

[4] Helman Stern, Boris Efros, "Adaptive Color Space Switching for Face Tracking in Multi-Colored Lighting Environments", *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, p. 0249, 2002.

[5] Sinha, Edgar Osuna, and Tomosa Poggio, “Pedestrian Detection Using Wavelet Templates”, *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

[6] K. Sobottka and I. Pitas, “Extraction of Facial Regions and Features Using Color and Shape Information”, in *Proc. of Int. Conf. on Pattern Recognition*, 1996

[7] M.H. Yang and N. Ahuja, “Detecting Human Faces in Color Images”, *In Proc. of ICIP'98, volume 1*, pages 127--130, 1998.

[8] Helman Stern and Boris Efros, "Adaptive Color Space Switching for Face Tracking in Multi-Colored Lighting Environments", *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, p. 249, 2002.

[9] K. Sobottka, I. Pitas. “Looking for Faces and Facial Features in Color Images”, *PRIA: Advances in Mathematical Theory and Applications*, Vol. 7, No. 1, 1997.

[11] Constantine P. Papageorgiou, Michael Oren, Tomaso Poggio, “A General Framework for Object Detection”, *Proceedings of International Conference on Computer Vision*, 1998.

[12] Paul Viola and Michael J. Jones, “Robust Real-Time Face Detection”, *International Journal of Computer Vision 57(2)*, p137-154, 2004.

[13] C. H. Lee, J. S. Kim, and K. H. Park, “Automatic Human Face Location in a Complex Background”, *Pattern Recog. 29, 1996, 1877–1889.*