Comparative Study on the Influence of Mahalanobis Distance and Skin Color Range for Face Detection using AdaBoost

Ong Guang Yi [1,2]
[1]Dept. of Intelligent Interaction Technologies,
University of Tsukuba, Tsukuba Science City,
Tennodai, Ibaraki Pref., 305-8573 JAPAN.
onggyi@gmail.com

Zulhadi Zakaria[2], Shahrel Azmin Suandi[2]
[2]School of Electrical and Electronic Engineering,
Engineering Campus, Universiti Sains Malaysia,
14300 Nibong Tebal, Pulau Pinang, MALAYSIA.
shahrel@eng.usm.my

***Abstract*—This paper reports a comparative study on using Mahalanobis distance and skin color range to segment the skin region. It also studies the effect of using skin region segmentation for face detection. Range of 1σ from the skin color mean in CbCr channel is used to segment the skin region, instead of using Mahalanobis distance. Ranges more than 1σ should be avoided, as they will give precision and recall worse than that of 1σ. Using skin color region to segment the face candidate does not necessarily reduce the detection time, and it is not always true that the detection rate of the detector will be improved. In most of the tests, the precision does not fall much when we increase the range. This means that the performance of the detector is 'buffered' by the AdaBoost-based detector.**

I. Introduction

Face detection is becoming more and more important. In 2002, Viola and Jones [1] implemented a robust real-time detection, inspired by Freund and Schapire [2]. They introduced haar-like features for face detection and used a method called integral image for fast face evaluation. Then a cascaded classifier is used, so that only potential regions of face will be focused in the detection. Viola and Jones [1] used the AdaBoost algorithm to select and train the classifier. They have achieved face detection at 15 frames per second.

Ever since, much of the research in face detection has been focused on optimizing the performance of the AdaBoost-based detector. Ma and Ding [3] introduced cost-sensitive AdaBoost to achieve robust and high detection rate with modest false alarm. Young and Ferryman [4] introduced optimized AdaBoost, by altering the implementation of AdaBoost, so that the computational load can be reduced.

He et al. [5] used YIQ model to locate the face, then using the eyes detected in the candidate region, the other facial features are located and the geometrical information of all facial features was used to verify the human face. Peng et al. [6] implemented color model based face detection using AdaBoost, by using a look-up table method. Wu and Ai [7] detected faces in color images using AdaBoost algorithm, based on skin color information. They used YCbCr color space to extract the skin color region, before passing the face candidate region to the AdaBoost-based detector.

II. Proposed Method

We are using the method proposed by Wu and Ai [7], that is using the YCbCr model to filter the image, then pass the filtered image to the AdaBoost classifier proposed by Viola and Jones [1].

A. Skin Color Region Segmentation

To segment certain color from an image, we need to perform a comparison of the color with a dataset. A useful distance measure used for color segmentation is known as Mahalanobis distance [8]. It is given by

$$D(\mathbf{z},\mathbf{a}) = [(\mathbf{z}-\mathbf{a})^T \mathbf{C}^{-1} (\mathbf{z}-\mathbf{a})]^{\frac{1}{2}} \tag{1}$$

where, $\mathbf{C}$ represents the covariance of the skin color region, $\mathbf{a}$ is the mean of the skin color region and $\mathbf{z}$ is the image vector in YCbCr model. Wu and Ai [7] defined values in Eq. (1) as follows:

$$\mathbf{z} = \begin{pmatrix} Cb \\ Cr \end{pmatrix} \tag{2}$$

$$\mathbf{a} = \begin{pmatrix} \mu_{Cb} \\ \mu_{Cr} \end{pmatrix} = \begin{pmatrix} 112.1987 \\ 151.3993 \end{pmatrix} \tag{3}$$

$$\mathbf{C} = \begin{pmatrix} C_{CbCb} & C_{CbCr} \\ C_{CrCb} & C_{CrCr} \end{pmatrix} = \begin{pmatrix} 89.3255 & 32.2867 \\ 32.2867 & 252.9236 \end{pmatrix} \tag{4}$$

B. Binary Image Conversion

Using the Mahalanobis distance, we can adjust the threshold of our binary image to mark the face region. For Mahalanobis distance in a pixel in the image, which goes beyond 30, the threshold value, we set the pixel as white, whereas black for those below it.

978-1-4244-7406-6/10/$26.00 ©2010 IEEE

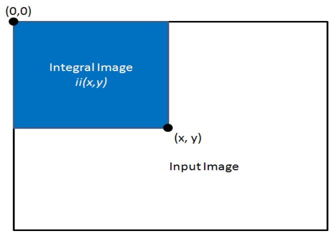

Fig. 1. Integral image at location (x, y)

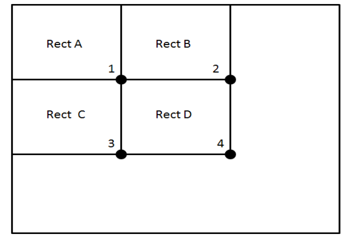

Fig. 2. The sum of pixels in Rect D can be calculated by four array references

C. Noise Filtering

Wu and Ai [7] used dilation to fill the holes in the binary image generated, and erosion to restore the shape of the face. They used a 3-by-3 structuring elements several times, followed by the same number of erosion operations, using the same structure. In our design, we used three dilations and one erosion.

D. AdaBoost-Based Detector

Viola and Jones [1] used a variant of AdaBoost, operating on integral images by using rectangle features and have obtained rather good results, that is faces in a 384 by 288 pixel image can be detected at 15 frames per second on a conventional 700MHz Intel Pentium III.

1) Rectangle Features: There are three main types of rectangles proposed by Viola and Jones [1]. They are two-rectangle features (edge features), three-rectangle features (line features) and four-rectangle features (diagonal features). Lienhart and Maydt [9] further expanded the features with $45°$ rotated rectangle features and the center-surround features. The features can be located at any location and any size and of any scale in the sub-window to be scanned across the evaluation image. In our case, we used 24×24 as the base scale of our sub-window.

2) Integral Image: Integral images are used to calculate the sum of pixels in a designated area rapidly. According to [1], the integral image at location (x, y) can be computed using the following pair of occurences:

$$s(x, y) = s(x, y-1) + i(x, y) \tag{5}$$

$$ii(x, y) = ii(x-1, y) + s(x, y) \tag{6}$$

where, $s(x, y)$ is the cumulative row sum, $s(x, -1) = 0$ and $ii(-1, y) = 0$.

Using Eq. (6), we can calculate the sum of pixels of any rectangle of any size at any location in the image rapidly, using the same amount of time. This function is useful when we would like to calculate the sum of pixels of a rectangle features in the evaluation image.

3) AdaBoost Algorithm: Table I describes the AdaBoost algorithm proposed by Viola and Jones [1]. This algorithm is used to train and select the rectangle features to classify the faces in the evaluation image. In the end of the training process, classifiers will be created from the features selected.

TABLE I
THE ADABOOST ALGORITHM

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0$ for negative examples and $y_i = 1$ for positive examples.
- Initialize weights $\omega_{1,i} = \frac{1}{2m}$ for $y_i = 0$ and $\omega_{1,i} = \frac{1}{2l}$ for $y_i = 1$, where m and l are the number of negative and positive samples respectively.
- For $t = 1, \ldots, T$, where T is the Tth hypothesis constructed using a single feature.
  1) Normalize the weight
  $$\omega_{t,i} \leftarrow \frac{\omega_{t,i}}{\sum_{j=1}^{n} \omega_{t,j}} \tag{7}$$
  so that ω_t is a probability distribution.
  2) For each feature j, train a classifier h_j, which is restricted to using a single feature. The error, ϵ_j is evaluated with respect to ω_t, where $\epsilon_j = \sum_i \omega_i |h_j(x_i) - y_i|$.
  3) Choose the classifier, h_t, with the lowest error, ϵ_t.
  4) Update the weights
  $$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i} \tag{8}$$
  where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is
$$h(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_t h_t \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0, & otherwise \end{cases} \tag{9}$$

In the process of detection, the classifiers trained and selected by the AdaBoost algorithm will classify the sub-windows into face and non-face candidates. Those which are non-face candidates will be eliminated directly after the cascade stage. In each cascade stage, the classifier will focus on harder sub-windows than the previous ones. Therefore, less detection time is needed to process the whole image, which therefore allows the classifer to allocate more time onto the more probable regions of faces. Figure 3 explains the detection cascade.

E. Skin Color Range-Based Segmentation

Using the means of Cb and Cr channel, added and/or subtracted with the standard deviation, all given by [7], we can obtain the range of the skin color in Cb and Cr channels. The standard deviations, σ_{Cb} and σ_{Cr}, can be obtained from Eq. (4), where $C_{CbCb}$is the variance of skin color from the mean in Cb channel and C_{CrCr} is the variance in Cr channel. To acquire the standard deviations, we squareroot the variances.

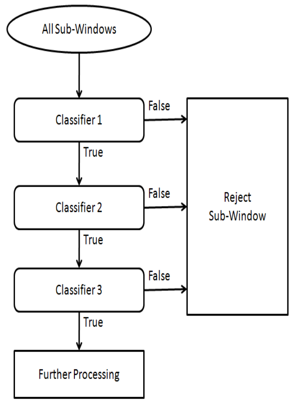

Fig. 3. The schematic representation of the detection cascade

$$\sigma_{Cb} = \sqrt{C_{CbCb}} = \sqrt{89.3255} \tag{10}$$

$$\sigma_{Cr} = \sqrt{C_{CrCr}} = \sqrt{252.9236} \tag{11}$$

To compute the range of the skin color in Cb and Cr channels, we can use the formulae below:

$$R_{Cb} = \mu_{Cb} \pm \sigma_{Cb} \tag{12}$$

$$R_{Cr} = \mu_{Cr} \pm \sigma_{Cr} \tag{13}$$

where, R_{Cb} and R_{Cr} stand for the range of the skin color region in Cb and Cr channels, respectively, μ_{Cb} and μ_{Cr} are the mean of the skin color region in Cb and Cr channels, σ_{Cb} and σ_{Cr} are the standard deviations of the skin color region in Cb and Cr channels. For the pixels which their values fall in the range, we will set the pixel as white, while black for those which falls out of the range. Then the skin region will be segmented based on the binary image formed, and the segmented region will be passed to the AdaBoost-based detector.

III. Results

To evaluate the performance of the detectors, we used a number of color images containing multiple faces, collected from the internet. Most of the images are images of celebrities, particularly from drama series, obtained from the image search of the search engine. This is due to the reason that, to date, we do not know any database containing color images with multiple faces.

We have allocated the images into several categories, which are two faces, three to five faces, six to ten faces and more than ten faces. For single face images, we use the frontal images of the FERET database.

Table II lists the number of images contained in each of the categories mentioned above.

The evaluation of the system has been done on an Intel Celeron 1.5GHz, 512MB RAM computer. In Table III, 'Color' denotes the Mahalanobis distance skin color segmentation face detector and 'Pure' denotes the pure AdaBoost-based detector, whereas 0.25σ, 0.5σ, 1σ, 2σ and 3σ each represents the range being used to segment the skin color region, respectively.

TABLE II
Categories of Images used for Evaluation

Category	Number of Images	Source
Single Face	119	FERET Database
Two Faces	70	Internet
Three to Five Faces	70	Internet
Six to Ten Faces	50	Internet
More than Ten Faces	9	Internet

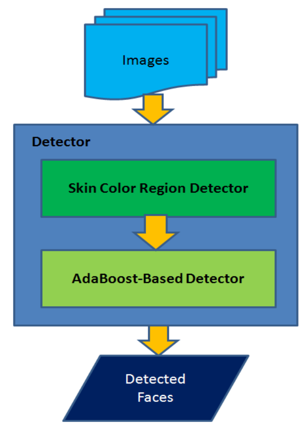

Fig. 4. The overall view of the proposed system

For the Color and 0.25σ, 0.5σ, 1σ, 2σ and 3σ ranges, we use the same method and flow of execution, except the parameters used for the skin color region segmentation are different. For the Color, we use Mahalanobis distance with threshold 30 to segment the skin color region; whereas for the ranges, we are adding and subtracting the range from the mean in Cb and Cr channels, using the standard deviation values given by [7] and subsection II-E. Figure 4 illustrates the overall view of the proposed system.

The number of detected faces decreases when the range (σ) is increased. The number of correct detection decreases significantly when we increase the range more than 1σ. There is no significant difference between the Color, Pure, 0.25σ and 0.5σ techniques. Using range of 1σ yields the best result (by comparing the precision). However, considering recall results, it does not perform as good as others.

Ranges of 0.25σ and 0.5σ approximate the results of skin color region segmentation using Mahalanobis distance, but they do not reduce the detection time. However, using range of 1σ, the result is better than the Mahalanobis distance while at the same time gives a shorter detection time.

In general, we can say that the false positive decreases when we increase the range, while at the same time, false negative increases drastically.

IV. Conclusion

In this paper, we performed comparative study on the influence of Mahalanobis distance and skin color region for face detection using AdaBoost. From the experiments, we have

TABLE III
OVERALL RESULTS

	Color	**Pure**	0.25σ	0.5σ	1σ	2σ	3σ
Detected	1319	1352	1334	1319	1038	742	567
Ground Truth	1038	1038	1038	1038	1038	1038	1038
Correct Detection	954	964	961	947	796	523	395
False Positive	365	388	373	372	242	219	172
False Negative	84	84	77	91	242	489	669
Precision	72.33%	71.30%	72.04%	71.80%	76.69%	70.49%	69.66%
Recall	91.91%	92.87%	92.58%	91.23%	76.69%	50.39%	38.05%
Total Detection Time (ms)	675362	717847	1063067	1044186	582292	700512	428340
Average Detection Time (ms)	2123.78	2257.38	3342.98	3282.60	1831.11	2202.87	1346.98

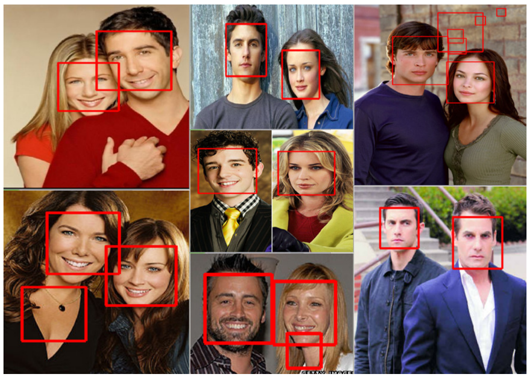

Fig. 5. Results of skin color region segmentation using Mahalanobis distance and cascaded to AdaBoost-based detector

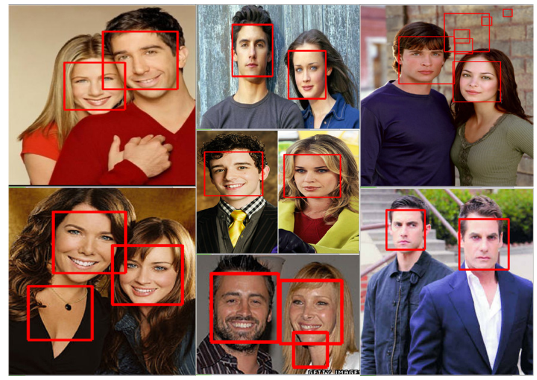

Fig. 6. Results of skin color region segmentation using range of 1σ and cascaded to AdaBoost-based detector

observed that in most cases, 0.25σ yields almost the same result as of 0.5σ. From this observation, we can conclude that 0.25σ is the optimum range, and any range smaller than that would give results comparable to that of 0.25σ or worse than it. As a solution to this, a range of 1σ from the mean of skin color in CbCr channel can be used to segment the skin region, instead of using Mahalanobis distance. On top of this, we should avoid using ranges more than 1σ as they will give precision and recall worse than that of 1σ.

Using skin color region to segment the face candidate does not necessarily reduce the detection time, and it is not always true that we can improve the detection rate of the detector. In most of the tests, the precision does not fall much when we increase the range. This means that the performance of the detector is 'buffered' by the AdaBoost-based detector. The recall in general decreases significantly, while we move from a smaller range to a larger range. This is due to the reason that less face candidates are detected, and thus less detected faces will comply to the ground truths. The false positive decreases when we increase the range, while at the same time, false negative increases drastically. This could be due to more face candidates will be passed into the AdaBoost-based detector, causing a high percentage of the detected faces to comply to the ground truths.

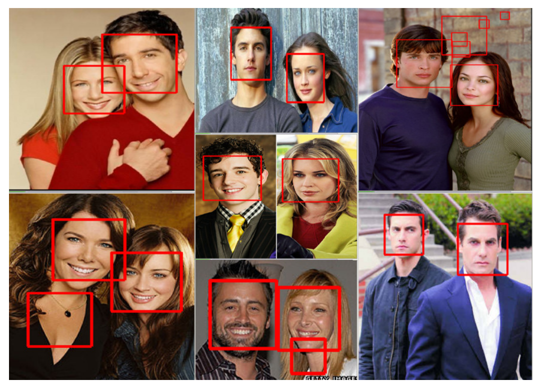

Fig. 7. Results of skin color region segmentation using range of 0.25σ and cascaded to AdaBoost-based detector

REFERENCES

[1] P. Viola and M. Jones, "Robust real-time object detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2001.

[2] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Computational Learning Theory: Eurocolt '95*, pp. 23–37, 1995.

[3] Y. Ma and X. Ding, "Robust real-time face detection based on cost-sensitive adaboost method," *Proceedings of International Conference on Multimedia and Expo 2003*, vol. 2, pp. 465–468, 2003.

[4] D. P. Young and J. M. Ferryman, "Faster learning via optimized adaboost," *IEEE Conference on Advanced Video and Signal Based Surveillance 2005*, pp. 400–405, 2005.

[5] X. He, Z.-M. Liu, and J.-L. Zhou, "Real-time human face detection in color image," *International Conference on Machine Learning and Cybernetcis 2003*, vol. 5, pp. 2915–2920, 2003.

[6] Y. Peng, Y. Jin, K. He, F. Sun, H. Liu, and L. Tao, "Color model based real-time face detection with adaboost in color image," *International Conference on Machine Vision 2007*, pp. 40–45, 2007.

[7] Y.-W. Wu and X.-Y. Ai, "Face detection in color images using adaboost algorithm based on skin color information," *2008 Workshop on Knowledge Discovery and Data Mining*, pp. 339– 442, 2008.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Prentice-Hall, 2002.

[9] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," *IEEE ICIP 2002*, vol. 1, pp. 900–903, 2002.