## 行政院國家科學委員會專題研究計畫 成果報告

## 多媒體影音高階處理、傳輸及設計--子計畫一:球賽視訊的 場景偵測,分類與總覽(3/3) 研究成果報告(完整版)

計	畫	類	別	:	整合型
計	畫	編	號	:	NSC 96-2221-E-002-089-
執	行	期	間	:	96年08月01日至97年07月31日
執	行	單	位	:	國立臺灣大學電機工程學系暨研究所

計畫主持人:貝蘇章

計畫參與人員: 教授-主持人(含共同主持人):貝蘇章

處理方式:本計畫可公開查詢

### 中華民國 97年05月30日

## 球賽視訊的場景偵測,分類與總覽—子計畫(3/3) Semantic Scenes Detection, Classification, and Summarization in Sports Video(III) 計畫編號: NSC 96-2221-E-002-089 執行期限: 96 年 8 月 1 日至 97 年 7 月 31 日 主持人:貝蘇章 台灣大學電機系教授

#### 摘要

專憑經驗的模式分解為一新穎非線性訊號分析方 法,它能分解一度空間訊號及二度空間影像為許 多基本性質模式函數,本論文提出一方向性影像 的經驗模式分解,較傳統的二度空間經驗模式分 解方法,運算簡單而且有效率可以應用在紋理及 自然影像之分析。

#### ABSTRACT

Empirical mode decomposition (EMD) is first proposed by Huang et al. in 1998. The EMD is local, adaptive and useful for analyzing non-linear and non-stationary signals. Bidimensional empirical mode decomposition (BEMD) is the extended version of EMD and can decompose an image into several multiscale components called intrinsic mode functions (IMFs). In this paper, a new algorithm for BEMD is proposed. This BEMD is realized using one dimensional EMD in two directions or more directions. Compared with the existing bidimensional sifting process interpolating the surface by radial basis function or thin-plate smoothing spline, the proposed method is much faster and more suitable for extracting fine directional textures. The experimental results show that the proposed approach can effectively decompose both textures and natural images.

#### **1. INTRODUCTION**

Empirical mode decomposition (EMD) is a powerful method for generating adaptive multiscale structure of non-stationary signals and analyzing them [1][2][3]. There are several variations of EMD algorithm introduced in [4]. The EMD is recently extended to analyze two dimensional signals [5][6][7][8][9][10] and the two dimensional extension is called bidimensional empirical mode decomposition (BEMD). There are

mainly two major methods to extend EMD to BEMD [5][6][7][8][9][10]. One carries out BEMD by real 2Dsifting process including extrema detection by neighboring window or morphological operators and surface interpolation by radial basis function [7][8] or thin-plate smoothing spline [5][6]. The other carries out BEMD for 2D-data by dividing a 2D-data into 1D data [9][10].

In this paper, we propose a new BEMD where one dimensional EMD is used to decompose two dimensional signals by dividing a 2D-data into 1D-data in two or more directions separately. Then, we combine the results of EMD in each direction by averaging them.

This paper is organized as follows. Section 2 and 3 describe the algorithm of EMD and BEMD. Section 4 shows the experimental results. Finally, conclusions are made in Section 5.

# 2. ONE-DIMENSIONAL EMPIRICAL MODE DECOMPOSITION

The EMD is first introduced by Huang et al. [1] for signal processing. This novel tool decomposes nonlinear or non-stationary signals into several frequency components called IMFs. The flow chart of EMD algorithm is plotted in Fig.1. IMFs must satisfy the following two conditions:

• the numbers of extrema and the numbers of zerocrossings must either equal or differ at most by one in the whole data set;

• at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima is zero.

The process of extracting the IMFs from a signal is called the sifting process introduced in [1].For a realvalued signal f(t), we can extract IMFs from it by the following steps:

1) Initialize  $h_0(t) = r_0(t) = f(t)$ , i=j=1;

2) Identify all local maxima and minima of  $h_{i-1}(t)$ ;

- 3) We generate the upper envelope by interpolating the local maxima, denoted  $h_{upper}(t)$ , and the lower envelope by interpolating the local minima, denoted  $h_{lower}(t)$ ;
- 4) Compute the envelope mean,  $m_{j-1}(t) = (h_{upper}(t) + h_{lower}(t))/2;$

5) Compute  $h_i(t) = h_{i-1}(t) - m_{i-1}(t)$ ;

- 6) Repeat steps 2-5 and set j=j+1 until  $h_i(t)$  is an IMF;
- 7)  $imf_i(t) = h_j(t)$  and compute the residue  $r_i(t) = r_{i-1}(t) - imf_i(t);$

8) Repeat steps 2-7 and set i=i+1 until  $r_i(t)$  is monotonic. When  $r_i(t)$  is monotonic, we have accomplished the EMD, and set  $r_i(t) = r_i(t)$ .

The original signal f(t) can be expressed as the sum of IMFs and the last residue as follow:

$$f(t) = \sum_{i=1}^{L} imf_i(t) + r_L(t)$$

The stopping criterion, *SD*, used to know whether  $h_j$  is an IMF, is defined as,

$$SD = \sum_{t=0}^{T} |m_{j-1}(t)|/T$$

In other words, the sifting process stops when SD is smaller than some predefined constant C.



Fig. 1. The flow chart of EMD for one dimensional signal.

#### 3. BIDIMENSIONAL EMPIRICAL MODE DECOMPOSITION

Texture analysis and multiscale structure are important techniques in image processing. BEMD is a new technique proposed to extract two-dimensional IMFs from images [5][6][7][8][9][10]. BEMD is better than Fourier or wavelet, and the other multi-scale structure when extracting features of textures or filtering images, because it is local, adaptive and suitable for non-linear, non-stationary data analysis. This section introduces two methods of BEMD. The first method is proposed in [5] called BEMD-1 and it is real 2D method. The second method called BEMD-2 is the proposed method using 1D EMD in two or four directions to approach the real 2D method.

#### 3.1. BEMD-1[5]

The steps of 2D-sifting process are described as follows: 1) Identify the local maxima and minima of the image I by neighboring window.

2) Generate the 2D 'envelope' by interpolating maxima points (respectively, minima points) by thin-plate smoothing spline.

3) Determine the local envelope mean m1 by averaging the two envelopes.

4) Since IMF should have zero local mean, subtract out the mean from the image: I-m1=h1.

5) Repeat steps 2-4 until h1 is an IMF

Repeat steps 1) to 5) after subtracting precedent IMF from input image until the residue has no more IMF, just like the algorithm of one-dimensional EMD.

#### 3.2. BEMD-2

This method is proposed to decompose images by 1D EMD and form 2D IMFs. Before doing the following steps, we find two orthogonal directions respectively:

 $q_1$  and  $q_2$ . Then we apply 1D EMD to image I(x,y) in

two direction,  $q_1$  or  $q_2$  ,respectively. In other words, we do steps as follows:

1) Initialize  $q = q_1$ ,

 $r_{i-1}^{q}(x, y) = h_{i-1}(x, y) = I(x, y).$ 

- 2) Apply one-dimensional EMD to each line in the first direction of the image.
- 3) Get residue  $r^{q_1}(x, y)$  and several IMFs.

- 4) Set  $q = q_2$ ,  $r_{i-1}^q(x, y) = r^{q_1}(x, y)$ , and repeat steps 2-3.
- 5) Change  $q_1$  to  $q_2$  and  $q_2$  to  $q_1$  in above steps, and do steps 1-4.

Finally, we can get the following equation:

$$I(x, y) = 0.5 \times \left(\sum_{i=1}^{N_1} \inf_{i} q_1(x, y) + r_{N_1}^{q_1}(x, y)\right) + 0.5 \times \left(\sum_{i=1}^{N_2} \inf_{i} q_2(x, y) + r_{N_2}^{q_2}(x, y)\right)$$

where N1 is the number of IMFs by steps 1-4, and N2 is the number of IMFs by step5.

We set N1=N2=N, and get the following equation, N

$$I(x, y) = \sum_{i=1}^{N} imf_i(x, y) + r_N(x, y)$$

where  $imf_i(x, y) = 0.5[imf_i^{q_1}(x, y) + imf_i^{q_2}(x, y)],$  $r_N(x, y) = 0.5 \times (r_{N_1}^{q_1}(x, y) + r_{N_2}^{q_2}(x, y)).$ 

The IMFs from EMD in two orthogonal directions can form 2D IMFs. Besides, this method can be extended to four directions by averaging these IMFs using two directions and other IMFs using another two directions. Similarly, we can extend this method to more and more directions to approximate 2D IMFs using real 2D-sifting process. The flow diagram of this BEMD algorithm for the direction of row and column case is plotted in Fig. 2. For textures, we use stage1 and stage2 and add the similar scales to form 2D IMFs as Fig. 3. Different from textures, we only apply stage1 of our algorithm to natural images.



**Fig. 2.** The flow diagram of BEMD-2 algorithm for two-dimensional signals. All the steps above red line is called stage1 and all the steps below red line is called stage2.

Different from the directional EMD (DEMD) in [9], our algorithm is to decompose 2D image by 1D EMD in two or more directions separately at the same time, then take the average of *i*-th IMF in all directions to form the 2D IMF. However, the DEMD in [9] generates the middle mean envelope by extrema detection and interpolation along each row, then generates the 2D mean envelope of the middle mean envelope by extrema detection and interpolation along each column. The other steps of DEMD is the same as BEMD-1.

#### 4. EXPERIMENTAL RESULTS

In order to compare the 2-D IMFs of BEMD-1 using real 2D-sifting process with the proposed BEMD-2 using 1D EMD to approach, both textures and natural images are used.

#### 4.1. For textures

Fig. 3 shows the results of synthetic image with the sum of three horizontal sinusoidal components and three vertical sinusoidal components. BEMD can decompose the image and generate different scales form fine to coarse. We add the same scale IMFs to approximate the BEMD-1's results due to this texture is symmetric. Fig. 4 shows the results of portions of Brodatz D101 [11]. From the experimental results, the proposed BEMD decomposes textures into more scales and faster than BEMD-1. In addition, we can add some IMFs to approximate the BEMD-1's IMFs.



Fig. 3. First row: synthetic image, IMF1, IMF2, and residues using BEMD-1. Second row: sum of IMF1 and IMF3, sum of IMF2 and IMF4,

residue. Third row: IMF1, IMF2, IMF3, IMF4, using BEMD-2.



Fig. 4. First row: Brodatz D101 [11], IMF1, IMF2, and residue using BEMD-1. Second row: sum of all IMFs, sum of IMF1 and IMF3, sum of IMF2 and IMF4, residue.



Fig. 5. Original images: Lena, Baboon, and Barbara.

#### 4.2. For natural images

We use three natural images, Lena, Baboon, and Barbara (Fig. 5.) for bidimensional empirical mode decomposition. First, we apply BEMD-1 to Lena image and get 2D IMFs of them (Fig. 6.). Secondly, we apply EMD to each row and each column of Lena image to form 2D IMFs (Fig. 7.), and compare with the results of BEMD-2 in four directions (0°, 45°, 90°, and 135°) shown in (Fig. 8.). From the experimental results of natural images, we can find that 2-D IMFs using 1D EMD in four directions or more directions will approximate 2D IMFs using BEMD-1 further. Therefore, Fig. 9 shows the results of Baboon, and Barbara decomposed by BEMD-2 in four directions and we compare them with BEMD-1's first IMF. The first IMF is like the image extracted the uneven illumination. We only show several IMFs due to other IMFs have little information.

In addition to approximating the first IMF of BEMD-1, BEMD-2 is faster and more suitable for extracting fine directional textures. Fig. 10 shows the parts of Barbara and Baboon, we can see the small directional textures are extracted from the images.



Fig 6. Lena image decomposition using BEMD-1. First row: IMF1, IMF2. Second row: IMF3 and residue.

#### **5. CONCLUSION**

In this paper, a new algorithm for BEMD is proposed using 1D EMD. The experimental results show that the proposed approach can effectively decompose both textures and natural images into 2D IMFs. These 2D IMFs are multi-scale images having similar frequencies locally and these features can help image recognition or texture analysis. The proposed BEMD not only extracts clear fine directional textures but also maintains local contrast when adding some IMFs. However, the IMFs generated by the proposed BEMD have small vertical lines or horizontal lines because using 1D interpolation method makes the inconsistent amplitude between row by row or column by column.

Compared with BEMD-1, the results of BEMD-1 are smooth in all directions, but BEMD-1 costs too much time to interpolate the 2D envelopes and decomposes the image into more fine scale with difficulty. Also, our BEMD only needs low computational cost because of line-based EMD and can approximate the results of BEMD-1 with high computational cost. The cost time of BEMD-1 is probably 2 hours for the image of size 256X256 pixels and SD=10. However, our method only costs several minutes with the same SD. In addition, BEMD-1 needs much more memory than ours.

There are still some boundary problems with our algorithm, and they will be left to the future work.



Fig 7. Lena image decomposition using BEMD-2 in two directions (0°and 90°).First row: IMF1, IMF2. Second row: IMF3, IMF4.



Fig.8. Lena image decomposition using BEMD-2 in four directions (0°, 45°, 90°, and 135°). First row: IMF1, IMF2. Second row: IMF3, IMF4.



Fig. 9.Left column: IMF1 using BEMD-1. Right column: sum of IMF1,IMF2,IMF3 using BEMD-2 in four directions.



Fig.10. First row: part of Barbara, IMF1 using BEMD-2. Second row: part of Baboon, IMF1 using BEMD-2.

#### REFERENCES

[1] N.E. Huang, Z. Shen, S.R. Long, M.L. Wu,H.H. Shih, Q. Zheng, N.C. Yen, C.C. Tung and H.H. Liu, "The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis." Proc. Roy. Soc.London A, Vol. 454, pp. 903–995, 1998.

[2] G. Rilling, P. Flandrin and P. Goncalves, Empirical Mode Decomposition As a Filter Bank, IEEE Signal Processing Letters, vol. 11, no. 2, pp. 112-114, 2004.

[3] G. Rilling, P. Flandrin and P. Goncalves, Detrending and Denoising with Empirical Mode Decomposition EUSIPCO-04,Wien.

[4] Rilling, G., Flandrin, P., and Goncalves, P., "On empirical mode decomposition and its algorithms." Pro. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing, 2003.

[5] A. Linderhed, Compression by Image Empirical Mode Decomposition, Proceedings of International Conference on Image Processing, ICIP-05 Genoa (I), September 11-14 2005, pp: 553-556.

#### [6]http://www.icg.isy.liu.se/~anna/emdcoding.html.

[7] J. C. Nunes, Y. Bouaoune, E. Delechelle, O. Niang and Ph. Bunel, "Image analysis by bidimensional empirical mode decomposition" Image and Vision Computing, Vol.21,Issue 12, pp.1019-1026, Nov 2003.

[8] J. C. Nunes, S. Guyot, E. Delechelle, "Texture analysis based on local analysis of the Bidimensional empirical mode decomposition" Machine Vision and Applications 16, pp.177-188, 2005.

[9] Z. X. Lin, S. L. Peng, "Texture segmentation using directional Empirical Mode Decomposition." IEEE International Conference on Image Processing (ICIP) October 24–27, 2004, Singapore.

[10] Z. X. Lin, S. L. Peng, "Boundary processing of bidimensional EMD using texture synthesis" IEEE Signal Processing Letters, Vol. 12. No 1. Jan 2005.

[11] <u>http://www.ux.uis.no/~tranden/brodatz.html</u>

## 多媒體影音高階處理、傳輸及設計--總計畫 (3/3) Multimedia Audio/Video High Level Processing, Transmission and Design (III) 計畫編號: NSC 96-2221-E-002-088 執行期限: 96 年 8 月 1 日至 97 年 7 月 31 日 主持人:貝蘇章 台灣大學電機系教授

#### 摘要

在智慧型視訊及影像分析,人臉偵測及辨識將是重要課題,膚色將是一個非常重要而且有效的臉部特徵可以應用,本論文探討人類膚色的彩色空間分佈圖,同時進一步分析白種,黃種及黑種人類膚色的彩色區域分布,並進行不同人種的偵測及辨識。

#### ABSTRACT

With the growing technique of communication between human and robot, the problem of human face recognition has attached more importance and become the current research in the popular domain of computer vision and recognition model. Thus, human's skin color is always an important mechanism and principle basis of human face detection. Human's skin color has the relative stability with the difference of the majority background object appearance. The skin color does not rely on the face detail characteristic and do not change with the face expression and rotation. Therefore, utilizing skin color to examine human face in color image is an important context of human face recognition.

In this paper, we provide a fast algorithm to identify human race with face skin color. The basic construction is roughly dividing human race into three parts: white, yellow and black race, then using Gaussian Mixture Model to train the feature parameter of each human race with large number of training images. Afterward, utilize Bayesian Decision Rule to determine the human race of test images.

#### **1. INTRODUCTION**

Face detection techniques based on the use of color information have been proposed recently. Take into account that the major dissimilarities between different races of people who have different skin color lie mostly in their chrominance so that skin color can be considered as good face segmentation and racial recognition feature. The main advantage in segmentation and racial recognition through the color characteristics is that facial detection can be performed independently on the size, position and expression of the face within the image. Also using color characteristics to classify human race is more convenient than using facial feature.

In our proposed method, we classify human races into three typical categories: white, yellow and black people (or Caucasian, Mongoloid and Negroid). First, several images are trained to get parameters of each race with GMM analysis. According to the analysis of GMM, there are tiny differences among these parameters of three different races and we use these parameters to classify the racial category of each testing image. We gathered the statistical result and displayed the experiment consequence in section 5.

#### 2. HUMAN SKIN COLOR DISTRIBUTION

#### 2.1 YCbCr color space

The formula between RGB and YCbCr is list below:

$\begin{bmatrix} Y \end{bmatrix}$		16		0.257	0.504	0.098	$\left\lceil R \right\rceil$
Cb	=	128	+	-0.148	-0.291	0.439	G
Cr		128		0.439	-0.368	-0.071	B

According to [1], most skin color pixels distribution and boundary box in the Cb-Cr plane show in Fig 1. The ranges of boundary box are:  $76 \le Cb \le 124$ and  $130 \le Cr \le 200$ .



Fig 1.Skin color distribution and boundary box in CbCr color space in Ref [1].

Considering only color information, a pixel will be classified as skin if both of its color components are within each of these ranges. So, the technique used for the racial recognition and face segmentation consists of defining maximum and minimum thresholds for each of the two chromatic components.

#### 2.2 YCgCr color space

There is another useful color coordinate: YCgCr, a novel color space based on YCbCr, was mainly proposed in [1, 2] for face segmentation. It is based on YCbCr, but it differs on the use of the Cg color component instead of Cb. The color spaces used in television systems (YUV, YCbCr) are transmission oriented, so in order to minimize the encoding decoding errors, they use the biggest color differences: (R-Y) and (B-Y). The YCgCr color uses the smallest color difference (G-Y) instead of (B-Y).

The YCgCr components can be obtained in a similar way than the YCbCr equations described in the ITU Rec. BT. 601 and expressed in terms of Y', G'-Y', R'-Y' components defined in the [0,1] range using the following matrix expression:

$$Y = 16 + 219 \times Y' ,$$
  

$$Y' = 0.299 \times R' + 0.413 \times G' - 0.144 \times B' ... Eq(1)$$
  

$$Cg = 128 + 112 \times [\frac{1}{1 - 0.587} (G' - Y')] ... Eq(2)$$
  

$$Cr = 128 + 112 \times [\frac{1}{1 - 0.299} (R' - Y')] ... Eq(3)$$

Luminance and chrominance are coded in 8 bits. Y has an range of 219 and an offset of 16. The chromatic components are defined in the rage [16,240], with range of  $\pm$  112 and an offset of 128. Each component is coded in 8 bits. Expressed in matrix form, R'G'B' components can be easily transformed to YCgCr components:

$$\begin{bmatrix} Y \\ Cg \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.316 & 0.439 & -0.121 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \dots Eq(4)$$

According to [2], most skin pixel color distribution can be detected in the chrominance bounding box in CgCr domain shown in Fig 11. A pixel will be considered as skin if both of its color components are within each of the ranges defined by the maximum and minimum thresholds of the chrominance plane coordinates Cg and Cr. Where (Cg min, Cg max)=[76, 125] and (Cr min, Cr max)=[136, 202].



Fig 2.Skin color distribution and boundary box in CgCr color space in Ref[2].

Consider the skin color distribution depicted in Fig 2, it is possible to achieve a better performance of the face segmentation in the YCgCr color space by defining a boundary box in the direction of the line that connects the Red and Cyan colors, where most of the skin values are concentrated.

New decision thresholds constituting a nonrectangular region (the black region inside the boundary box shown in Fig 2) can be defined, so that the Cr thresholds are the same vertical limits of the boundary box, while the Cg thresholds, Cg min and Cg max lines, are parallel to the Red-Cyan line. Hence the Cr min and Cr max thresholds and the Cg min and Cg max lines, according to the following equations:

 $Cr \min \le Cr \le Cr \max \qquad ....Eq(5)$ (Cr min, Cr max)= [136, 202].  $\frac{(305 - Cr)}{1.38} \le Cg \le \frac{Cr \min + 1.38 \times Cr \max - Cr}{1.38} Eq(6)$ 

Instead of using two binary masks for face segmentation based on the Cg and Cr thresholds of the boundary box, in the case of a nonrectangular decision region, for every Cr value, a binary mask is used for each Cg min and Cg max pair of Cg thresholds. The final mask for face segmentation is obtained as the intersection of all the binary masks.

#### 2.3 The reasons of choosing YCgCr

The black cluster of Fig 1 and 2 represent the majority of human skin chrominance in different domains. Fig 1 displays human skin chrominance of CbCr and Fig2 represents that of CgCr. Compare Fig 1 to Fig 2, the shape of black cluster of Fig 2 in CgCr domain is more regular than that of Fig 1. The human

skin distribution in CgCr is less widespread than that in CbCr and more centralized as ellipse shape.

Because the chromatic distribution of CgCr has better shape than that of CbCr, it will achieve better performance in GMM detection. The slanted thin ellipse shape in CgCr domain means the negative correlation between Cg and Cr. The large value of Cg results in small value of Cr so that Cg and Cr have strong correlation.

Contrariwise, the relation between Cb and Cr displays a random-like distribution in Fig 1 and the shape is more widespread so that it is more difficult to analyze the distribution in CbCr domain.

#### 2.4 Distribution in YCgCr domain of different races

We use YCgCr domain to compute our experiment with GMM analysis of different races' pictures which only include the region of nose, because this region is purity of skin color without the effects on other colors of lip or eyes. There are three typical human races (white, yellow and black (or Caucasian, Mongoloid and Negroid)) and their distribution of CgCr domain shown in Fig 3.



Fig 3.Different racial color distribution in CgCr domain

We choose three typical pictures to represent each racial distribution of CgCr shown in Fig 3. The red cluster displays Caucasian's distribution in CgCr domain, the blue region displays Negroid's distribution and the yellow cluster displays Mongoloid's. It can be seen that the majority of black people's distribution are at the location between white people and yellow people. Also the Mongoloid's distribution has larger value in CgCr domain than others while he Caucasian's distribution has smallest value in CgCr domain.

#### 2.5 Brightness and other chromatic distribution

0.5

0.52

There are several other kinds of color coordinates to represent the distribution with different races. The researches on Lab and Yuv represented in [3,4] show





Fig 4. Saturation and brightness chromatic distribution

of flesh tone in Ref [3][4] (a) uv; (b) La chromaticity

In Fig 4(b), Caucasian has greater brightness value than other races and in (a), saturation value which is represented to the magnitude of u and v is distributed greatly to Negroid, but the hue value which is represented to the angle of u and v is almost same for races. Thus, the flesh tones for races are discriminated for the combination value of both saturation and brightness.





Fig 5.Skin color distribution in YCg and Y Cr domain.

In Fig 5, there are two figures which show the experiment result of distributions in Y-Cg and Y-Cr domain. They show that Negroid really has less brightness than Caucasian and Mongoloid. Furthermore, Mongoloid has the widest distribution of luminance; we will consider the effect of illumination in GMM analysis.

#### **3. GAUSSIAN MIXTURE MODEL**

GMM (Gaussian mixture model) which is the extension of single model Gaussian probability function is a conventional method to analyze non-uniform distributed data and is among the most statistically mature methods for clustering (though they are also used intensively for density estimation. In this tutorial, we introduce the concept of clustering, and see how one form of clustering in which we assume that individual data points are generated by first choosing one of a set of multivariate Gaussians and then sampling from them can be a well-defined computational operation.

Mixture models are a semi-parametric alternative to non-parametric histograms (which can also be used as densities) and provide greater flexibility and precision in modeling the underlying statistics of sample data. They are able to smooth over gaps resulting from sparse sample data and provide tighter constraints in assigning object membership to colorspace regions. Such precision is necessary to obtain the best results of optimum means, covariance and weighting possible from color-based pixel classification for qualitative segmentation requirements.

We then see how to learn such a thing from data, and we discover that an optimization approach not used in any of the previous Andrew Tutorials can help considerably here. This optimization method is called Expectation Maximization (EM). We'll spend some time giving a few high level explanations and demonstrations of EM, which turns out to be valuable for many other algorithms beyond Gaussian Mixture Models.

1-D data for example in Fig 6, we can see that 2-3 clusters occupy separate sub-space and the probability of mixture clusters reveals Gaussian like distribution. We can use EM algorithm to find out optimum mean, co variance and weighting value of each cluster. 2-D data for example in Fig 7 also can be estimated by EM algorithm.



Fig 6. The compare of original 1-D data and GMM fitted. (a)The fitting 2 components of GMM. (b) The fitting 3 components of GMM.



Fig 7. 2D data with 3-components of GMM fitted

We applied 3-components Gaussian mixture models to the cluster of each race, Fig 8 for example. Each component of Gaussian model has its optimum mean, covariance and weighting which are obtained by EM algorithm.



Fig 8. The compare of original 2-D data and GMM fitted with different races. (a)Negroid (b) Mongoloid (c) Caucasian.

#### 3.1 Mixture Gaussian model with EM algorithm

If the shape of these data points' distribution is not like ellipse in d- dimension, then we can not use single Gaussian model to describe the probability density function of these data points. So we use several Gaussian distributions with weighted average to describe it, Take 3-components GMM for example, the probability density function can be expressed as:  $p(X) = a_1g(X, \mathbf{m}_1, \Sigma_1) + a_2g(X, \mathbf{m}_2, \Sigma_2) + a_3g(X, \mathbf{m}_3, \Sigma_3)$ 

$$=a_{1}g_{1}+a_{2}g_{2}+a_{3}g_{3}$$

There are several parameters of this density function  $(a_1m_1, \Sigma_1, a_2m_2, \Sigma_2, a_3m_3, \Sigma_3)$ , and there is a condition of  $a_1, a_2, a_3$ :

 $a_1, +a_2+a_3=1$ 

The kind of expression is called GMM. In order to simply the calculation, we let the covariance matrix of each Gaussian density function become:  $\Sigma_j = s_j^2 I_{dxd}$ . So

$$g(X, \mathbf{m}_{j}, \mathbf{s}_{j}) = \frac{1}{\sqrt{(2p)^{d} |\mathbf{s}_{j}|^{2d}}} \exp(-\frac{1}{2} \frac{(X - \mathbf{m}_{j})^{T} (X - j \mathbf{m})}{2\mathbf{s}_{j}|^{2}})$$
  
Let  $(\mathbf{a}_{1} \mathbf{m}_{1}, \mathbf{\Sigma}_{1}, \mathbf{a}_{2} \mathbf{m}_{2}, \mathbf{\Sigma}_{2}, \mathbf{a}_{3} \mathbf{m}_{3}, \mathbf{\Sigma}_{3}) = \Theta$ 

we use EM algorithm to find optimum ⊖. • Estimation Step:

1. Define the initial value of 
$$\Theta =$$

$$(a_1m_1, S_1, a_2m_2, S_2, a_3m_3, S_3)$$
. We set

 $a_1 = a_2 = a_3 = \frac{1}{3}$  and use k-means to calculate the

three centers of the cluster as  $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ . Initial

$$\mathbf{s}_{1}^{2} = \mathbf{s}_{2}^{2} = \mathbf{s}_{3}^{2} = \mathbf{s}_{j}^{2} = \frac{1}{n-1} (Xi - \mathbf{m}_{j})^{T} (Xi - \mathbf{m}_{j}).$$

2. Use these initial parameters to calculate  $b_1(Xi)$ ,  $b_2(Xi)$ ,  $b_3(Xi)$ , i=1~n.

Maximum Step:

1. Calculate new parameters with iteration loop.

$$\tilde{\mathbf{m}}_{j} = \frac{\sum_{i=1}^{n} b_{j}(Xi)Xi}{\sum_{i=1}^{n} b_{j}(Xi)},$$
,  
$$\tilde{\mathbf{s}}_{j}^{2} = \frac{1}{d} \frac{\sum_{i=1}^{n} b_{j}(Xi)(Xi - \mathbf{m}_{j})^{T}(Xi - \mathbf{m}_{j})}{\sum_{i=1}^{n} b_{j}(Xi)},$$
,  
$$\tilde{\mathbf{a}}_{j} = \frac{1}{n} \sum_{i=1}^{n} b_{j}(Xi), j = 1, 2, 3, i = 1...,n$$
$$\Rightarrow p(\tilde{X}) = \tilde{\mathbf{a}}_{1} g(X, \tilde{\mathbf{m}}_{1}, \tilde{\mathbf{s}}_{1}) + \tilde{\mathbf{a}}_{2} g(X, \tilde{\mathbf{m}}_{2}, \tilde{\mathbf{s}}_{2}) + \tilde{\mathbf{a}}_{3} g(X, \tilde{\mathbf{m}}_{3}, \tilde{\mathbf{s}}_{3})$$
If  $\left| p(\tilde{X}) - p(X) \right|$  is smaller than a specific value,

then break the iteration loop or back to Maximum Step. This iteration loop will raise the value of  $J(\Theta)$  step by step, and finally gain the local maximum of  $J(\Theta)$ .

#### 4. CLASSIFICATION RULE

In this section the Bayesian Theory is used to derive a decision machine (classifier) used in a verification system. The machine is then implemented using GMM approach. The k-means, EM algorithm and maximum a posteriori (MAP) adaptation algorithm, used for finding GMM parameters, are described. The section is concluded by a discussion on implementation issues.

#### 4.1 Bayesian Decision Theory

A verification system, on the fundamental level is a two- class decision machine: based on given observation vectors, the client is either an impostor or the true claimant. In this section we shall use Bayesian Decision Theory [7] to implement the decision machine.

Let us denote client specific true claimant and impostor classes as C1 and C2 respectively, and let  $\vec{x} = [x1...xn]^T$  be the observation vector. Moreover, let P(Cj) be the a priori probability of class Cj, and  $\vec{x}$  | Cj ) be the conditional probability density function (pdf) of  $\vec{x}$ , given class Cj. We seek to find the class that  $\vec{x}$  belongs to. Using the Bayes formula [9], we obtain:

$$P(Cj \mid \vec{x}) = \frac{P(x \mid Cj)P(Cj)}{P(\vec{x})}$$
  
where  $P(\vec{x}) = \sum_{i=1}^{2} P(\vec{x} \mid Ci)P(Ci)$  for two-class decision.

Thus using the Bayes formula we obtain the a posteriori probability of Cj, j=1, 2. It follows that the Bayes decision rule is then:

Choose C1 if  $P(C1|\vec{x}) > P(C2|\vec{x})$ 

Or more generally, index of chosen class =  $\arg \max P(C_i | \stackrel{\rightarrow}{x})$  which is known as the maximum a

posteriori decision rule. It must be noted that p(x) is not required for making the decision. Thus the decision rule becomes:

 $\arg\max_{i} P(\vec{x} | Cj) P(Cj) \cdot$ 

Intuitively, the decision machine will make fewer mistakes when using more observations vectors. Thus in practice, multiple observation vectors are used: X= xi, i=1...n. Assuming that the observation vectors are independent and identically distributed (iid), then the joint likelihood is:  $P(X | Cj) = \prod_{i=1}^{n} P(\overrightarrow{x_i} | Cj) P(Cj)$ . In practice, the true form of the pdf P(X | Cj) is

unknown, hence a parametric representation, P(X | Ci), estimated from training data, is used instead. Since  $P(X | C_i)$  is only an approximation, a correction function,  $\tilde{P}(X | C_i)$ , is required.

$$P(X | Cj) = \prod_{i=1}^{n} \tilde{P}(\vec{x}_i | Cj) \tilde{P}(\vec{x}_i | Cj) P(Cj)$$

$$| | |$$
[Approximation] Estimation]

Taking into account the multiple observation vectors and rewriting  $\arg \max P(Cj \mid x) P(Cj)$  into ratio test yields:

Choose class =  $C1 \quad if \quad \frac{\tilde{P}(X \mid C1)}{\tilde{P}(X \mid C2)} > \frac{\tilde{P}(X \mid C2)P(C2)}{\tilde{P}(X \mid C1)P(C1)}$  $C2 \quad if \quad \frac{\tilde{P}(X \mid C1)}{\tilde{P}(X \mid C2)} < \frac{\tilde{\tilde{P}}(X \mid C2)P(C2)}{\tilde{\tilde{P}}(X \mid C1)P(C1)}$ 

Due to precision issues in computational implementation, it is more convenient to use a summation rather than series of multiplications. Since log(.) is a monotonically increasing function, the decision rule can be modified to:

Choose class = C1 if  $\log(\frac{\tilde{P}(X \mid C1)}{\tilde{P}(X \mid C2)}) > \log(\frac{\tilde{P}(X \mid C2)P(C2)}{\tilde{P}(X \mid C1)P(C1)})$ C2 if which translates to:

Choose class =

C1 if 
$$\log(\tilde{P}(X \mid C1)) - \log(\tilde{P}(X \mid C2)) > \log(\frac{\tilde{P}(X \mid C2)P(C2)}{\tilde{P}(X \mid C1)P(C1)})$$
  
C2 if others

Where, for clarity,  $\log \tilde{P}(X | Cj) = \sum_{i=1}^{n} \log \tilde{P}(x_i | Cj)$ 

Due to practical considerations described later, the number of observation vectors needs to be taken account

Choose class =

$$C1 \quad if \quad \frac{1}{n} [\log(\tilde{P}(X \mid C1)) - \log(\tilde{P}(X \mid C2)] > \frac{1}{n} \log(\frac{P(X \mid C2)P(C2)}{\tilde{P}(X \mid C1)P(C1)})$$

$$C2 \quad if \qquad others$$

Let us define

$$L(X \mid Cj) = \frac{1}{n} \log(\tilde{P}(X \mid Cj)) = \frac{1}{n} \sum_{i=n}^{n} \log(\tilde{P}(x \mid Cj))$$

This can be interpreted as the average log likelihood of X. Thus we can choose class =

C1 if 
$$L(X | C1) - L(X | C2) > \frac{1}{n} \log(\frac{\tilde{P}(X | C2)P(C2)}{\tilde{P}(X | C1)P(C1)})$$

others

C2 if

Let us define  $\Lambda(x) = L(X | C1) - L(X | C2)$ 

Since the true form of the pdf p(x|Cj) is unknown, the correction function,  $P(X | C_j)$ , is also unknown; moreover, in real life situations the a priori probabilities P(C1) and P(C2) are often unknown. Thus in practice,  $\frac{1}{n} \log(\frac{\tilde{P}(X \mid C2)P(C2)}{\tilde{P}(X \mid C1)P(C1)})$  is replaced with an

experimentally found threshold, t.

Choose class = C1 if  $\Lambda(x) > t$ , t=0, usually. C2 if others

Strictly speaking, the normalization factor 1/n is not necessary to make a decision. However, in practical situations variable length observations are often encountered. Since  $\Lambda(x)$  is observation length independent, it allows the approximation of the distributions of  $\Lambda(x)$  for true clients and known impostors, which in turn simplifies the selection of the threshold.

#### 4.2 Block diagram of decision procedure

Testing Images



PDF created with pdfFactory trial version www.pdffactory.com

other colors of lip or eyes. So we cut each full-face picture into region with noise.

We use MATLAB 6.5 to implement the experiment and count accuracy, error rate and uncertainty of these two sets of experiment and we will discuss result in the following sections.

Training	Races	Amount
Images		of samples
Set 1	Caucasian (white)	16
	Mongoloid (yellow)	20
	Negroid (black)	18
Set 2	Caucasian (white)	16
	Mongoloid (yellow)	18
	Negroid (black)	18

Table 1 : Amount of Training images in Set 1 and Set 2

We first calculate 3-components GMM parameters of each race in every two of three color space and average-normalize those means, covariance and average weightings of each training image, then put testing images into Bayesian Classification Rule to find max P(X|Cj),j=1,2,3 in two of three color space and make decision with majority of voting result.

If the majority of voting result is correct, then we said that it is an accurate detection. If the majority of voting result is not correct, then we call it is an error detection.

Result of Uncertainty will also be discussed. There are several conditions to result in uncertainty.

1. When the voting results of two of three color spaces are equal, for example, if decision is Caucasian in YCg, Mongoloid in YCr and Negroid in CgCr, then it will be an uncertainty.

2.If | P(X|Cj) - P(X|Ci)| < t,  $i \neq j$ , i, j=1,2,3 in any two of three color spaces, it is also an uncertainty.

Usually the tiny distinction of P(X|Cj) among three race categories are unable to make decision by the voting result.

#### 5. EXPERIMENT RESULT

We collect two sets of three races pictures, each race of every set includes about more than 15 training images and more than 30 testing images, these pictures in two sets are all distinct. These pictures are collected from several face image databases on internet and web albums of daily life. So the training images and testing images include many different kinds of picturing situations such as different luminance condition, night model with flashlight, daytime model with much exposure.

We use two of three YCgCr domains to compute our experiment with GMM analysis of different races' pictures which only include the region of nose, because this region is purity of skin color without the effects on

Testing	Races	Amount	Accuracy	Erro	Uncertainty
Images		OI		r	5
		samples		rate	
Set 1	Caucasian	30	83.3	16.7	0%
			%	%	
	Mongoloid	30	83.3	13.3	3.3%
			%	%	
	Negroid	31	80.6	13%	6.4%
			%		

Table 2 : Classified Accuracy of each race in Set 1

Testing	Races	Amount	Accuracy	Error	Uncertainty
Images		of	,	rate	,
		samples			
Set 2	Caucasian	30	80.0	20.0	0%
			%	%	
	Mongoloid	30	86.6	10.0	3.3%
			%	%	
	Negroid	30	80.0	20.0	0%
			%	%	

Table 3: Classified Accuracy of each race in Set 2

#### 5.1Discussion on false detection

According to the result, we can see that the accuracy of Negroid is the lowest; this is because the distribution of Negroid in CgCr domain shown in Fig 3 is between Caucasian and Mongoloid. This kind of distribution may result in the false judgment easily and lead to low accuracy. For example, there are several false detections of Negroid in Set 1, one of these errors is show in Fig 20 which is classified as Caucasian due to the higher illumination. Because the original picture of Negroid was taken during the daytime, the violent sunlight made the skin brighter than the average parameters of Negroid and Fig 9's result of GMM analysis of Y domain is a deviation of Caucasian's side.



Fig 9. The false detection of Negroid due to sunlight.

Here is another error detection shown in Fig 10 in Set 1. Fig 10 was classified as Mongoloid; the primary picture of Negroid was taken at night with the flashlights. The strong intensity of flashlights projected on the dark-brown skin color and made the skin color look like yellow- brown. The distribution of GMM analysis in YCgCr domain is a deviation of Mongoloid's side. Consequently, the voting machine classified Fig 10 as Mongoloid.



Fig 10. The false detection of Negroid due to flashlights.

The false classifications usually occur when the violent illumination changes such as sunlight and flashlights or strong surrounding lighting condition. However, the illumination Y domain still has a great help for the classification with the testing images under normal lighting condition.

#### **5.1Conclusion and Future Work**

We proposed new automatic skin color detection of human races with Gaussian Mixture Model analysis on a novel color space, YCgCr. It has been applied on two training sets of many images of different races and lighting conditions for obtaining the Gaussian Mixture Model's parameters of YCgCr. Its performances of different races have been tested with two sets of images which only include the region of nose with several different lighting conditions.

The false detections only occur when the violent illumination changes such as sunlight and flashlights or strong surrounding lighting condition. However, the illumination Y domain still has a great help for the classification with the testing images under normal lighting condition. The performance of Negroid is the poorest among three races detections because the distribution of Negroid in CgCr domain is between Caucasian's and Mongoloid's distributions.

This method of skin color detection on YCgCr domain with GMM analysis can be applied easily on

the interface of robot machine and security monitor in the future. This method also can be applied on preprocess of face recognition to help raise the accuracy of recognition.

#### **6. REFERENCE**

- J.J. de Dios, N. Garcia, "Face Detection Based on an New color space YCgCr", Proc. Of the IEEE int. Conf. on Image Processing ICIO'03, vol.3, pp. 909-912, Barcelona, Sept. 2003.
- [2] J.J. de Dios, N. Garcia, "Fast Face Segmentation in Component Color Space", Volume 1, 24-27 Oct. 2004 Page(s):191 - 194 Vol. 1 Digital Object Identifier IEEE CNF.
- [3] Do-Hun Kim; Hyun-Chul Do; Sung-Il Chien;," Preferred skin color reproduction based on adaptive affine transform, Volume 51, Issue 1, Feb. 2005 Page(s):191 – 197. Digital Object Identifier IEEE CNF.
- [4] Eung-Joo Lee; Yeong-Ho Ha, "Automatic flesh tone reappearance for color enhancement in TV". Volume 43, Issue 4, Nov. 1997 Page(s):1153 -1159 .Digital Object Identifier IEEE CNF
- [5] Sudanthi N R. Wijewickrema, Andrew P.Paplinski, "Principal component analysis for the approximation of an image as an ellipse"
- [6] Demas Sanger, Takuya Asada," Facial pattern detection and its preferred color reproduction", IS&T and SID's 2nd color imaging conference: color science systems and applications, pp 149-153, 1994.
- [7] Jason Palmer, Ken Kreutz-Delgado, Scott Makeig."Super Gaussian Mixture Model". Journal of Machine Learning Research
- [8] Huanfeng Ma and David Doermann. "Adaptive Word style classification using a Gaussian Mixture Model" Volume 2, 23-26 Aug. 2004 Page(s):606 -609 Vol.2 Digital Object Identifier 10.1109/ICPR.2004.1334321
- [9] Conrad Sanderson, B Eng. "Automatic person verification using face and speech information Ch2 Gaussian Mixture Model Based Classification". A Dissertation presented to The School of Microelectronic Engineering Faculty and Information Technology Griff University. Aug. 2002.