# A Real Time Human Face Detection Scheme on Digital Signal Processor Based on Trimmed Hierarchical Tree

Chen-Chung Liu, I-Li Chiu
*Department of Electronic Engineering,*
*National Chin-Yi Institute of Technology, Taichung, Taiwan, R.O.C.*
E-mail: ccl@chinyi.ncut.edu.tw, easycig@hotmail.com

***Abstract-- In this paper, we implement a real time human face detection system on a digital signal processor (DSP) by using Texas Instruments' TMS320C6416 DSP device as the development platform and using DSKcam-CMOS modulus to offer the input images. Since the computation speed of DSP is slower than the computation speed of computer (or note- book computer). This paper uses the Feature-based method to detect human face to compensate the operation speed of the proposed detection system. The input RGB image is firstly transformed into the YCbCr domain. In YCbCr domain, the image is segmented into several no overlapping regions by the color feature of skin, and these skin color regions that have large area are selected as the regions of interesting (ROI). Each ROI is represented with a trimmed hierarchical tree. If the ROI has the feature of the five sense organs of mankind (ears, eyes, lips, nose and tongue), then the ROI is detected as a region of human face by our system. The experiment results show that our system should detect the human face correctly and quickly while the Pan angle and the tilt angle of the camera toward the front face of the human are restricted in -135° ~ +135° and -65° ~ +65° respectively, even though there are other skin color objects ( hands, cloths in skin color, and so on). On the other hand, this paper use an eyes features block (EFB) mask to distinguish whether the detected human face is real human face or is only a human face picture.***

***Keywords*：*real time, DSP, face detection, RGB***

## 1. INTRODUCTION

Recently, DSP can work well for the jobs that need abundant calculations due to the effective promotion of DSP's computation efficiency. Today's DSP can be applied on low- level controls such as digital motor control, sensor, and power source control and so on. DSP is also used in MP3, VoIP, and GPS acceptor and so on.

Human face detection has received significant attention due to its wide applications [1-4], such as personal identification and access control, human face recognition, model-based video coding, intelligent human–computer interaction, and low-bandwidth communication for videophone. Human face detection is always an interesting and difficult subject in pattern recognition, and there are numerous published papers about the schemes of human face detection nowadays. But, there are only a very small part of them are suitable to be used in a real- time embedded system.

M.H. Yang [5] proposed that there are four classes in human face detection system: 1) Knowledge-based; this scheme uses the rules of the sense organs on human face to detect human faces from an image, 2) Feature-based; this scheme detects human face by finding the structure features of human face, 3) Template matching; this scheme compares the selected objects in the input image and the previous defined human face temples database to detect human face from the input image, 4) Image-based; this scheme bases on the machine learning and the statistic analysis to detect human faces from an image.

J. S. Liao et al.[6] propose a human faces scheme based on ARM-LINUX and Feature-based. They transform an input RGB image into the HSV (Hue, Saturation, and Value) color space and find the ranges of H, S, and V of human skin respectively. They then find out ROI from the input image according to the ranges of H, S, and V of human skin. They also determine the features of eyes and lips by using the Feature-based scheme. They use the features of eyes and lips to detect human faces from ROI. Their scheme not only detects human faces effectively and quickly but also detects multiple human faces in the same time. This paper proposes a real time human face detection system that uses DSP as the development platform and modifies the J. S. Liao et al. scheme.

C. M. JUO et al. propose a human faces detection research that has the interaction between a robot and people [7]. His hardware platform uses a FPGA to match TMS320C6416 image development platform. They adopt the Feature-based method in their algorithm of human faces detection; they decrease the resolution of input images to 160x120 firstly, they then transform the lower resolution RGB images into

YCbCr color space, they then combine the human skin detection and the projection method to find the ROI, and they finally use a Sub-window to filter out the ROI that does not have five sense organs features and detect the human faces from the input image. They use abundant data to disciple the neural network system to obtain features of human faces. The neural network system has well performance in human faces detection, but the computation is a time and memory consuming. This algorithm is not suitable to apply in the real time embedding system for human faces detection.

After our analysis on human faces detection system, the drawbacks of Template matching or Image-based detection is that they need a numerous computation. Any embedding system always can not support such numerous computations. Knowledge-based human faces detection method and Feature-based human faces detection method are suitable for an embedding system; they only need to combine the location of human skin color feature and the human five sense organs features then the system can detect human faces from input image effectively and quickly.

Hierarchical clustering can divide a data set into groups of items such that items within a group are similar to one another and different from those in other groups [8-11]. These groups are represented with a hierarchical tree-shaped structure. The hierarchical clustering is applied in many areas such as multi-resolution signal analysis, sub-band coding for speech and image segmentation. We believe that the hierarchical clustering will have much effect in the detection of human faces. The remainder of this paper is organized as follows: Section 2 is the simple discussion of the development environment structure of our embedding system. Section 3 presents the human faces detection algorithm. Empirical tests are presented in Section 4. Finally, we conclude this paper in Section 5.

## 2. THE STRUCTURE OF DEVELOPMENT ENVIRONMENT

The hardware design of TI-C6000 chip use the VelociTI structure, it can carry out eight instructions of 32-bit in a period time clock. The communication between control of peripheral and DSP is using the memory mapping. Texas_Instrument want to configure the hardware structure and the software of the chip in more constructive, they insert a functions library called Chip Support Library (CSL) between the DSP chip layer and the application layer (software). The chip support library offers all the API programs of peripheral controls of the DSP chip, such that engineers only need to call the normalized API programs when they are using the chip to develop a new work. The relation among the DSP chip layer, the application layer, and the chip library layer is shown in Fig. 1.

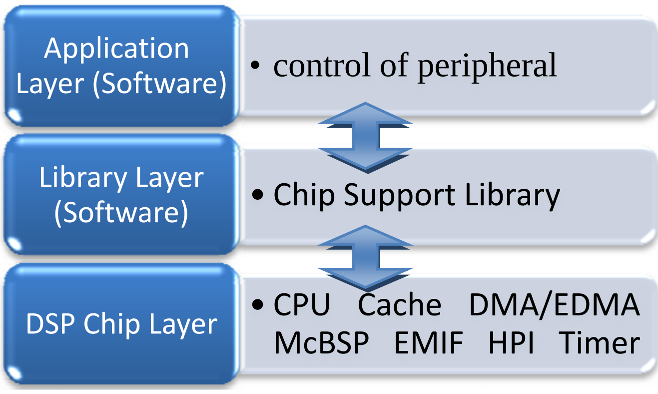



**Fig.1 Architecture Library Based on Chip**

### 2.1. Hardware Structure

This paper takes TI- TMS320C6416 fabricated by Texas Instrument as the hardware of the human faces detection system. Fig. 2 shows the whole hardware structure and the CMOS Image Module is shown in Fig. 3.

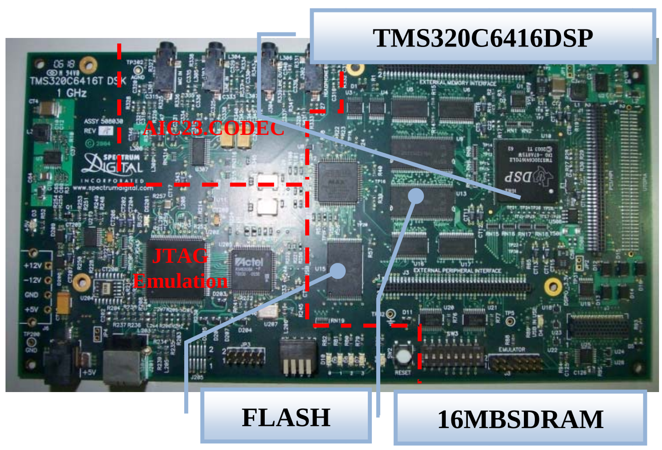



**Fig.2 TMS320C6416**

### 2.1.1. TMS320C6416

Texas Instrument classifies the TMS320 DSP into three grades; they are C2000 class, C5000 class, and C6000 class: C2000 class belongs to the optimal control grade, C5000 class is in the low power grade, and C6000 class is the high performance grade. There are C62x, C64x, and C67x in C6000 class, where C67x is a floating point operator.

C6416 is a high level fixed point operator in TI-C6000class. The operation frequencies of C6416 are 600MHZ and 1GHZ; it is high speed performance and is suitable to be used in the image processing. Where the processing speeds of 1GHZ C6416 should be 8000MIPS while it uses a USB as the communication interface between the computer and it. This paper combines C6416, 16MB-SDRAM, and 512K-word FLASH to construct the hardware of the human face detection system.

### 2.1.2. Image Module

This paper uses DSKcam V3.0 fabricated by BiTEC company as the CMOS image module. The CMOS sensor the CMOS image module is Omnivision OV7620 CMOS sensor. OV7620 CMOS sensor has 124 registers to support the 30fps VGA/QVGA whose pixel recognition ratio on resolution is 664*492 and the available image modes are YUV422 and rawRGB.

DSKcam V3.0 use First In First Out (FIFO) mode to output 16-bit image data to the 6Mbit image module buffer to save the time that transfers the data to the main memory. The block diagram of CMOS Image Module System is shown in Fig.4.

DSKcam V3.0 CMOS image module also assort W3100A TCP/IP Ethernet ether network module, which uses RJ-45 connector to offer 10/100M ether network.

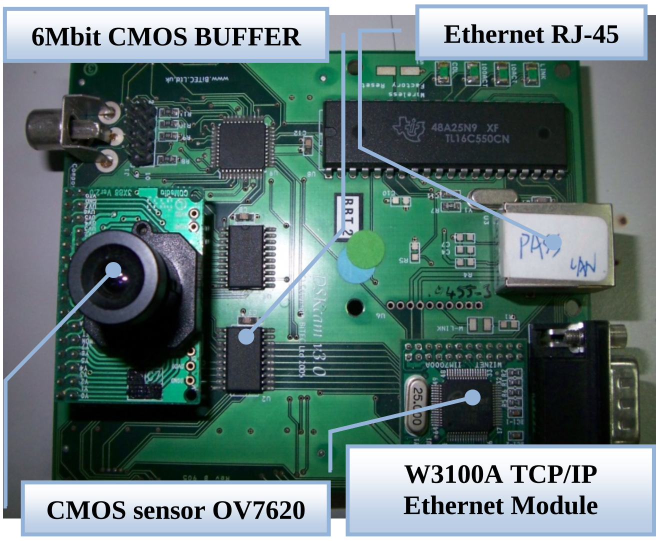



**Fig.3 CMOS Image Module**

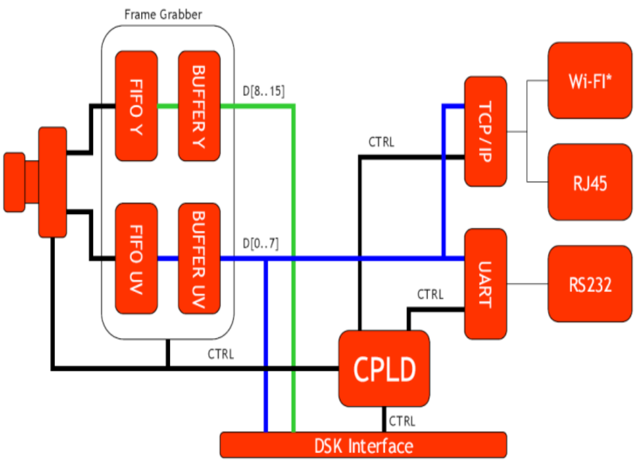



**Fig.4 The block diagram of CMOS Image Module System**

## 2.2. Memory Structure

The computation speed of DSP is always lower than the computation speed of computers. We use L2 flash memory to handle the program operation to save operation time, because one uses flow bus to catch the data in the outer SDRAM should take more time.

The flash memory in TMS320C6416 has two parts; each part is divided into program code memory and data region memory. TMS320C6416 possesses a 32Kbyte-L1 flash memory 600MHZ operation time clock. Both program code memory and data region memory are 16Kbytes. The 1024Kbyte-L2 flash memory's operation frequency is

300MHZ. The C6416 memory architecture is shown in Fig.5. The 1024Kbyte-L2 flash memory can declare to work by itself and has 1Mbyte memory. We take the 1024Kbyte-L2 flash memory as the memory of the operation image data.

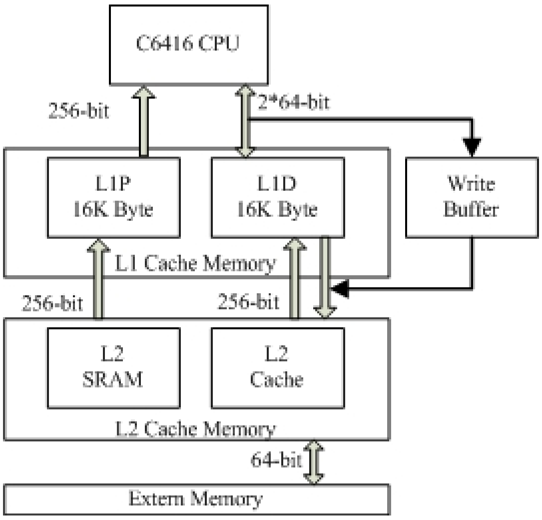



**Fig.5 C6416 Memory Architecture**

## 2.3. Turn on Mode

There is only one type of memory mapping in C6416. The starting address of interior memory is 0h such that the definition of turning on is very simple. C6416 offers three turning on modes; ROM mode, HPI mode, and None mode. All the three turning modes should be defined by BOOTMODE [1:0].

The ROM turning on mode uses EDMA to move the turning on program code from the CE1 position of the outer memory to the address 0h of memory. CPU then begin to carry out program from 0h; EDMA in C6416 moves 1Kbyte turning on program code each time, the 1Kbyte turning on program code should be used by Boot loader. Boot loader should load the main program code from the FLASH to the interior memory of CPU and should not load the main program code from the USB. This procedure coincide the definition of embedding system.

# 3. Human Face Detection Algorithm

This paper uses Feature-based method to do the human face detection, the detail steps of the Feature-based method using in human face detection is shown in Fig. 6. The detail descriptions are presented in the following subsections respectively; they are human skin color detections, noise detection,
the hierarchical tree of human faces, the hierarchical tree of human faces, and the detection of human five sense organs.

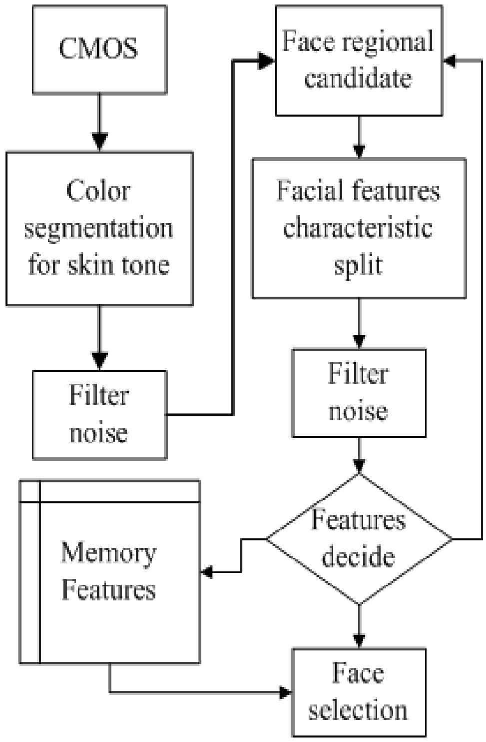



**Fig.6 The flow chart of human face detection**

## 3.1. Skin Color Detection

Firstly, the system detects the human skin color regions from the input image. The human skin color regions detection is not an easy task because the input images are frequently not only manifold but also involutes. The input image is loaded into buffer by CMOS with mode YUV4:2:2.So, our system only need to process the YUV image transformed from input RGB image, where Y represents the illuminates, U and V represent the chrominance and concentration respectively. Fig. 7 shows the U- V color plane.

Visual Inf. Process. Res. Group [12] proposed that the Cb value interval of human skin is [75, 135] and the Cr value interval of human skin is [130, 180]. Many other objects around the human faces like as the clothes and human hands also have the same values of Cb and Cr of human faces. If one person only uses the values of Cb and Cr of human faces to detect human faces from an YCbCr image, he should detect some regions that are not the human faces regions and increases many computations in the human faces detecting procedure.

The chromaticity and the concentration of color are the most important roles in the judgment methods of human skin color. There is not any color information in luminance and we want to modulate the detection method of human faces that only uses the values Cb and Cr [12]. This paper uses formula (1) and formula (2) that combine Y, Cb, and Cr to detect the human skin regions from a YCbCr image. Fig. 8 shows the experiment results of human skin detection; Fig8.

(a) is the detected result of pure using the Cb value interval of human skin and the Cr value interval of human skin, Fig8. (b) is the detected result of modulated detection using the Cb and Cr value intervals of human skin and formulas (1) and (2). An extrude wall is detected in Fig. 8 (a) and the wall does not be detected in Fig. 8 (b).

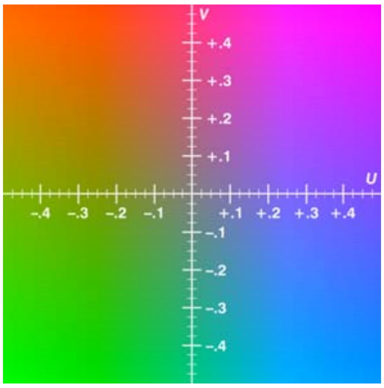
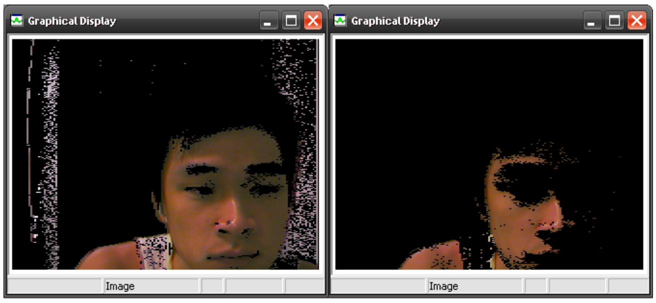



**Fig.7 U-V color plane**

$$\begin{cases} if\ (Y \le 128) \\ \theta_1 = -2 + \dfrac{256-Y}{16}; \theta_2 = 20 - \dfrac{256-Y}{16}; \theta_3 = 6; \theta_4 = -8 \\ if\ (Y > 128) \theta_1 = 6; \theta_2 = 12; \theta_3 = 2 + \dfrac{Y}{32}; \theta_4 = -16 + \dfrac{Y}{16} \end{cases} \quad (1)$$

$$\begin{cases} Cr \ge -2(Cb+24); Cr \ge -(Cb+17); Cr \ge -4(Cb+32); \\ Cr \ge 25(Cb+\theta_1); Cr \ge \theta_3; Cr \ge -0.5(Cb-\theta_4); \\ Cr \le -\dfrac{(Cb-220)}{6}; Cr \le -1.34(Cb-\theta_2); \end{cases} \quad (2)$$




**(a)** **(b)**

**Fig.8 Skin color detection result (a) using Cb&Cr only (b) using Cb, Cr, and formulas (1) and (2).**

### 3.2. Noise Deleting

For increasing the detecting efficiency and decreasing the influence of noise in the image, we have to do something for the noise. The median filter has the suitable filter property that we need, so, we took it as a processor in our human face detection system to reduce the noise and increase the detection rate. We use the median filter to filter out noises in the skin color detected image; The median filter is a nonlinear spatial filter, is a powerful tool at removing outlier type noise. It does suit for the job at preserving edges of an image. The filter mask simply defines what pixels must be included in the median calculation. The computation of the median filter starts at ordering those n pixels defined by the filter mask, in the order from minimum to maximum value of the pixels as given in Equ. (3).

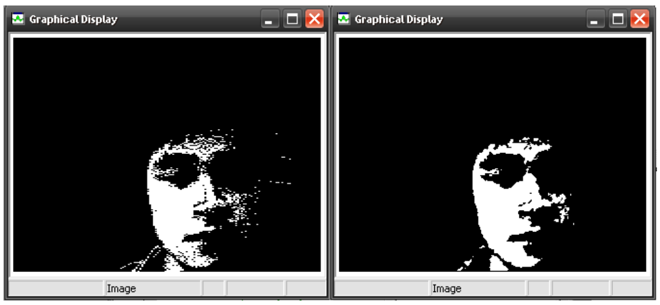
$$F_0 \le F_1 \le F_2 \cdots \le F_{n-2} \le F_{n-1}, \quad (3).$$

where $F_0$ denotes the minimum and $F_{n-1}$ is the maximum of all the pixels in the filter calculation. The output of the median filter is the median of these values and is given by

$$F_{med} = \begin{cases} \dfrac{F_{n/2} + F_{n/2-1}}{2} & for \quad n \quad even \\ F_{n/2} & for \quad n \quad odd \end{cases} \quad (4).$$

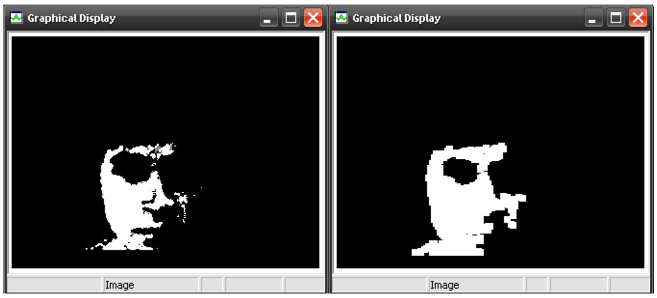
Typically, an odd number of filter elements are chosen, to avoid the additional step in averaging the middle two pixels of the order set when the number of elements is even.

There still have some survived small area noise in the image that has been filtered by the median filter. We use morphological operation to delete those residual small area noises; the opening operation morphology is conducted to delete the no correlative details in the median filtered image. The corresponding filtered results are shown in Fig. 9 and Fig. 10.




**(a)** **(b)**

**Fig.9 Median Filtering result: (a) regions of skin color (b) regions of skin color after Median Filtering**




**(a) (b)**

**Fig.10 Morphological operation result: (a) after median filtering, (b) after morphology opening.**

## 3.3. The Hierarchical Tree of Human Faces

In the finding candidate regions of human faces step, we adopt a tree structure method to decrease the computation loading and to delete those no correlative regions of human skin color. In the binary image that human skin regions detected and noise deleted, human skin regions are colored in white (pixel value equals 255) and background is black colored in black (pixel value is zero). We frame the human face candidates from the black-white image by finding the seed of human face candidate and construct a human face candidate by a tree with the seed as the root of the human face candidate:

1) We find the first white pixel to take as a seed by starting at the left- upper pixel of the black-white image, and shifting from left-to-right and top-to-bottom in the black- white image.

2) A 2X3 window shown in Fig. 11 is taken as the mask to filter other white pixels as the leaves of the human face candidate hierarchical tree. Each human face candidate hierarchical tree represented an object of the black-white image. Fig.12 shows an example of the construction of a human face candidate hierarchical tree. We also use the threshold value of region size, 300 pixels, to filter out those candidate regions that are small size.

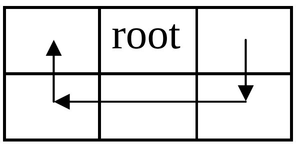
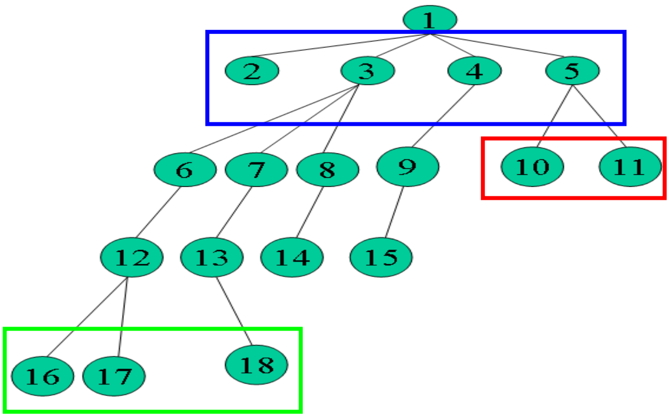



**Fig.11 The 2X3 window used to filter in other white pixels as the leaves of the human skin hierarchical tree.**

| | | | | |
|---|---|---|---|---|
| 0 | 0 | 1 (1) | 1 (2) | 0 |
| 1(11) | 1 (5) | 1 (4) | 1 (3) | 0 |
| 1(10) | 1 (9) | 1 (8) | 1 (7) | 1 (6) |
| 1(15) | 1(14) | 1(13) | 1(12) | 0 |
| 0 | 1(18) | 1(17) | 1(16) | 0 |

**(a)**



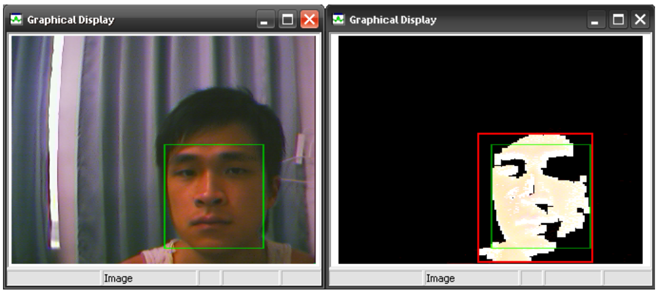

**(b)**

**Fig. 12 The construction of a human skin hierarchical tree; (a) black- white image, (b) the hierarchical tree of (a).**

## 3.4 The Trimmed Hierarchical Tree

We want to frame all the five sense organs on human face in each frame, but we also want to exclude all redundancy pixels in the frame at the same time. So, we refer the characteristics of the hierarchical tree to trim out three kinds of nodes from the human faces object image; the nodes of middle layer leaves (nodes in the red rectangle in Fig. 12), the father nodes of redundancy leaves, and the terminal leaves nodes of bottom layer (nodes in the green rectangle in Fig. 12). Most nodes of the middle layer leaves are the pixels of human face skin, and the terminal leaves nodes of bottom layer should be the shoulder or neck region of a human. The three kinds' nodes are less useful in human face detection, so they are deleted from the candidate region of human face by using formula (5). Forehead is the nodes that around the root of a human face candidate hierarchical tree (nodes in the blue rectangle in Fig. 12), forehead is also useless in human face detection and should be deleted in our scheme to improve the performance of human face detection. Finally, we adjust the image size of the trimmed human face hierarchical tree to find the candidate regions of human faces. Fig. 13 shows an example of the candidate regions of human faces, where the red rectangle is the original outer contour of the candidate regions of human faces and the green rectangle is the original outer contour of the candidate regions of human faces respectively.

$$\begin{cases} t_{high} = t_{high} - k \\ t_{low}(x) - t_{medina}(x) > 0; t_{low} = 0 \\ t_{low}(x) - t_{medina}(x) \le 0; t_{low} = t_{low} \end{cases} \quad (5)$$




**(a)** **(b)**

**Fig.13 After trimming and size adjustment frame in (a) original RGB image, (b) in white- black image.**

### 3.5 Detection of Five Sense Organs

On the detection of human's five sense organs (ears, eyes, lips, nose and tongue), we take the Sobel operation [14, 15] on candidate regions of human faces to find out the features of human faces edges. Sobel operator is an operator used mainly in image processing for edge detection. Sobel operator is a discrete difference operator in technology and it is used to find the magnitude of each image pixel's gradient on the gray value of image pixel. The Sobel operation on a gray image A is defined in the following formulas:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * A \quad (6)$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (7)$$

$$G = \sqrt{G_x^{\ 2} + G_y^{\ 2}} \quad (8)$$

Where $G_x$, $G_y$, and G are the x − component matrix, y − component matrix, and the magnitude of the grdient of image matrix A respectively. In the human face objects segmentation stage, we use a Sobel gradient operator to get the human face edge information of the human face object, and a binary closing operation was used to fill the object that is the interior region of the edge contour. After mapping the location of the object back into original gray-scale image, the human face object location was found and the human face object image was extracted exactly. Fig. 14 is an example of the candidate region extraction of human faces. On the other hand, we boost the quality of edges created by Sobel operation with contrast enhancement. An example of contrast enhancement result is shown in Fig. 15.

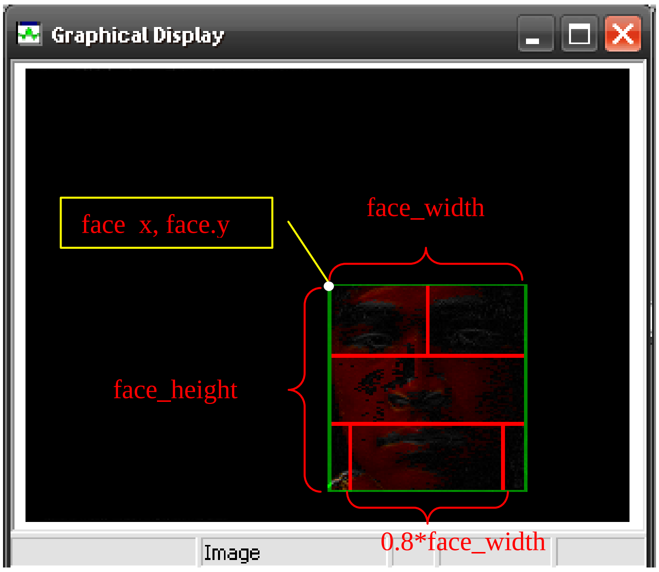



**Fig.14 Extracted candidate of human faces in UV skin color and the corresponding segmentation.**

For each candidate rectangle region of human face, we define the upper left corner with coordinate ( $face\_x, face\_y$ ) as the searching starting point, determine the height of the rectangle region $face\_height$ , and find the width of the rectangle region $face\_width$. The rectangle region is divided into three no overlapped equal layers in the vertical direction; they are named as eyes region, nose region, and lips region. The eyes region is divided into left eye region and right eye region in advance. Fig. 14 shows the segmentation of a candidate rectangle region of human face. Sobel operation is then conducted on left eye region and right eye region to obtain the edges, those edges are judged to be the edges of eyes or not.

Referencing to Fig. 15, our system certificates that

1) the left ( or right) eye region has a human eye while the pixel number of edges in the region is not less than 20 and the length of edges is longer then the one fourth of the $face\_width$ (described by formula (9)).

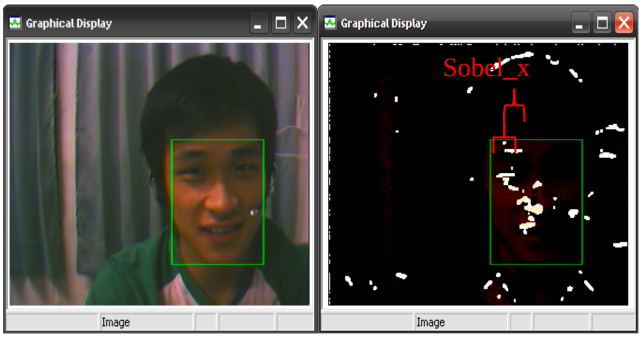
$$\begin{cases} Sobel\_x > \frac{1}{4} * face\_width \quad ; \\ \quad\quad Sobel\_w \geq 20 \end{cases} \qquad (9)$$

Where $Sobel\_x$ indicates the length of edge in pixel and $Sobel\_w$ is the pixel number of edge in the checked region.

2) The nose region has a human nose while the pixel number of edges in the region is not less than 20.

3) The lips region has human lips while the pixel number of edges in the region is not less than 20.

When the number of positive answers in above items is not less than two, our system should certificate that the candidate rectangle region of human face is really is a human face image.




**(a)** **(b)**

**Fig.15 Example of contrast enhancement result; (a) original candidate region of human faces, (b) Sobel Boost edges in candidate region**

### 3.6 Eyes features blocks masking

Human's eyes cannot stay at one pose for a long time; they must close and open frequently. So, we add a block-comparison step in the human face detection flow chart; we store the last three frames image of eyes features blocks (EFB) in a register as masks to mask the current input image. The three EFB masks are all normalized into a standard size. The current input EFB is normalized into the standard size first. Then the same features of the normalized input EFB is masked out by the three EFB masks; we take the excluded OR operation (XOR) for the normalized input EFB and the three EFB masks. We also calculate the density of difference $D_i$ between the $i$-th input EFB and the EFB masks by the following formula:

$$Di = \sum_{j=1}^{m} \sum_{k=1}^{n} I(j,k) \oplus M(j,k)/(m*n) \qquad (10)$$

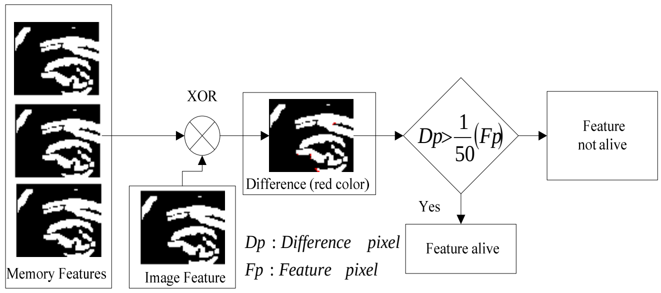



**Fig.16 The flow chart of EFB comparison**

Where m*n is the size of the mask, I(j,k) is the pixel value of the input EFB at the position (j,k), and M(j,k) is the pixel value of the mask EFB at the position (j,k). The input EFB is a human eyes block while $D_i$ is not less than 0.02, otherwise the input EFB is not a human eyes block. The detail flow chart of EFB comparison is shown in figure 16 and the test results are shown in figure 17.

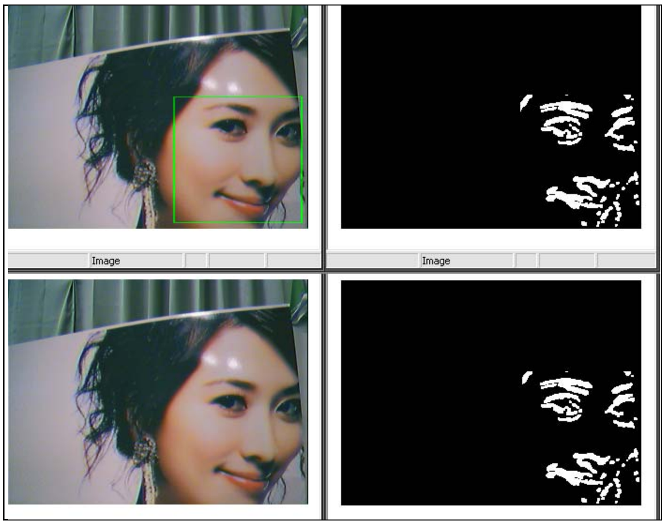
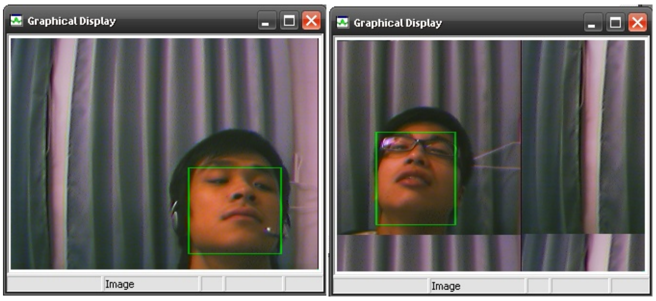


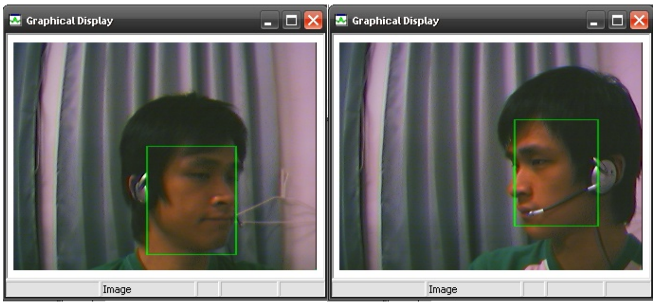

**Fig.17 The EFB test results; the picture of human face is detected originally (upper- left figure), the picture of human face is selected originally as the candidate of human face (upper- right figure), the picture of human face is not detected finally (lower- left figure), and the picture of human face is still selected finally as the candidate of human face (lower- right figure).**

## 4. Experiment Results

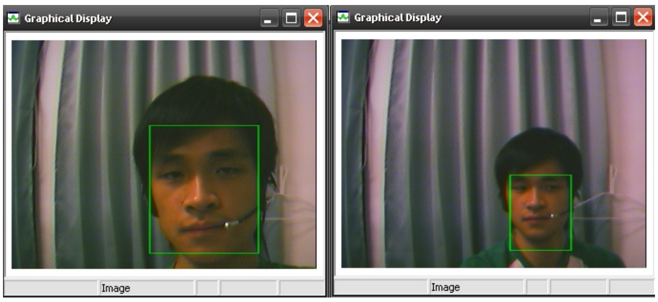
In order to demonstrate the performance of the proposed scheme, a series video sequence images of human faces captured by a DSKcam of CMOS module (320 by 240), in which there are many lighting effects, human face poses, complex backgrounds and human faces captured from any direction at varying distance $d$ were used in simulation. The experiment results show that our system should detect the human face correctly and quickly while the pan angle $\varphi$ and the tilt angle $\theta$ of the camera toward the front face of the human are restricted in $-90^{o} \sim +90^{o}$ and $-65^{o} \sim +65^{o}$ respectively, even though there are other skin color objects ( hands, cloths in skin color, and so on). Several experiment results under different pan angles, tilt angles, and complex background are shown in Fig. 18.



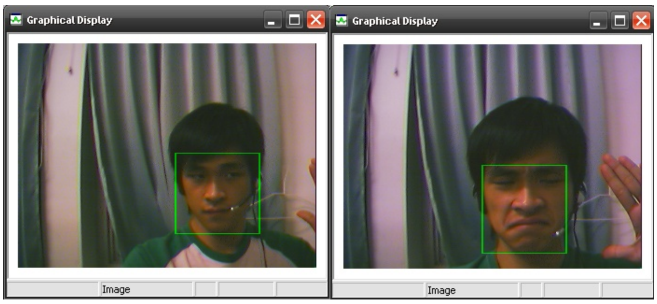

**(a)** **(b)**



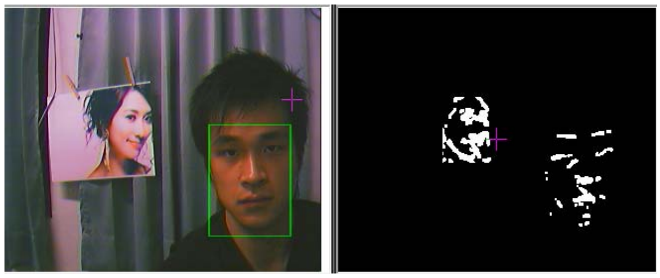

**(c)** **(d)**



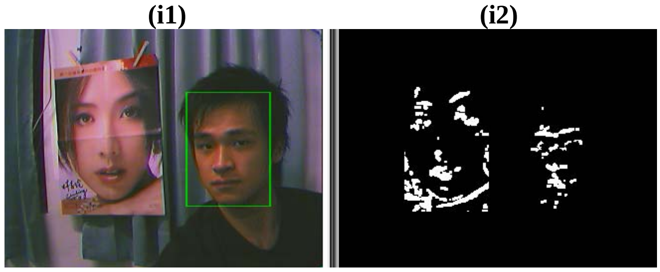

**(e)** **(f)**




**(g)** **(h)**



**(i1)** **(i2)**



**(j1)** **(j2)**

**Fig.18 Example of experiment results: (a) $d$= 2 m, $\varphi$= 10°, $\theta$=15°. (b) $d$= 2.5 m ,$\varphi$= 10°, $\theta$=25°. (c) $d$= 1.5 m , $\varphi$= −75°, $\theta$=0°. (d) $d$= 1 m ,$\varphi$= 65°, $\theta$= 0°. (e) $d$= 1 m ,$\varphi$= 0°, $\theta$= 0°. (f) $d$= 2.5 m ,$\varphi$= 0°, $\theta$=0°,other skin object. (g) $d$= 2.5 m ,$\varphi$= −15°, $\theta$= −5°, oblique human face. (h) $d$= 1.2 m ,$\varphi$= 30° , $\theta$=0°, other skin object. (i1) the real human face is finally detected and the picture of human face is not selected finally. (i2) the real human face and the picture of human face are all detected originally as the candidates of human face. (j1) the real human face is finally detected and the picture of human face is not selected finally. (j2) the real human face and the picture of human face are all detected originally as the candidates of human face.**

It takes 28~35ms (well fit the real time standard) to detect the human face from an input image by using L2, the detection time depends on each image's structure. It should take other more 3~5ms to detect the human face in the same case by using an outer memory. The time consumption on the eyes features block masking step is 33~61ms. The time consumption depends mainly on the size of the detected human face.

## 5. Conclusion

This paper proposed a real time embedded system of human faces detection implemented on DSP. The system can locate the positions of human faces and detect the human faces from input images efficiency, effectively, and quickly under varying visual angles, even though in complex background. On the other hand, the system can also distinguish the real human face and the picture of human face. There is not any LCD that can immediately display the outputs of this DSP chip now. We shall find some suitable ARMs and LCDs to be connected in this DSP chip to improve the performance of the real time human faces detection system in the future; the ARMs shall share some computations of DSP and the LCD is used to display the results of human faces detections.

## References


[1] H. Bae, S. Kim, "Real-time face detection and recognition using hybrid-information extracted from face space and facial features", *Image and Vision Computing*, Vol. 23, 2005, pp. 1181–1191.

[2] S. Paschalakis, M. Bober, "Real-time face detection and tracking form mobile videoconferencing", *Real-Time Imaging*, Vol. 10, 2004, pp. 81–94.

[3] M. Castrillon, O. Deniz, C. Guerra, M. Hernandez, "ENCARA2: Real-time detection of multiple faces at different resolutions in video streams", *J. Vis. Commun. Image R.*, Vol. 18, 2007, pp. 130–140.

[4] Z. Zhu, Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations", *Computer Vision and Image Understanding*, Vol. 98, 2005, pp. 124–154.

[5] M.H. Yang, D.J. Kriegman and N. Ahuja, Detecting faces in images: a survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.24, no.1, 2002.

[6] J. S. Liao, C. H. Tsai, W. Y. Chen, "Real-time face tracking scheme based on Embedded system " *Intelligent Living Technology 2008* ,2008.

[7] C. M. JUO, "Real-time face tracking under illuminant change" *National Chiao Tung University Department of Electrical and Control Master thesis,* July ,2005

[8] R. M. Castro, M. J. Coates, and R. D. Nowak," Likelihood Based Hierarchical Clustering", *IEEE Transactions on Signal Processing*, Vol. 52, No. 8, 2004, pp. 2308- 2321

[9] S. Rajasekaran, "Efficient Parallel Hierarchical Clustering Algorithms", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 16, No. 6, 2005, pp. 497- 502.

[10] J. Basak and R. Krishnapuram, "Interpretable Hierarchical Clustering by Constructing an Unsupervised Decision Tree ", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 1, 2005, pp. 121- 132.

[11] O. Maqbool, H. A. Babri , "Hierarchical Clustering for Software Architecture Recovery", *IEEE Transactions on Software Engineering*, Vol. 33, No. 11, 2007, pp. 759- 779.

[12] Visual Inf. Process. Res. Group, Edith Cowan Univ., Churchlands, WA, Australia, "A novel skin color model in YCbCr color space and its application to human face detection," *Image 2002 International*.

[ 13] C. Lin, "Face detection in complicated backgrounds and different illumination conditions by using YCbCr color space and neural network", *Pattern recognition letter*, Vol. 28, 2007, pp. 2190–2200.

[14] Y. D. Qu, C. S. Cui, S. B. Chen, J. Q. Li, "A fast subpixel edge detection method using Sobel–Zernike moments operator ", *Image and Vision Computing*, Vol. 23, 2005, pp. 11–17.

[15] S. Jin, W. Kim, J. Jeong, "Fine Directional De-interlacing Algorithm Using Modified Sobel Operation ", *IEEE Transactions on Consumer Electronics*, Vol. 54, No. 2, pp. 857–862.