

Research on Centrosymmetric Graph based on DAM6416P

Chun-shan Yang, Gui-jun Liu
 Department of Computer Science and Technology,
 ChengDong College of Northeast Agricultural University
 Harbin, 150025, China
 ycszy1999@126.com

Abstract—In order to explore the feasibility of application of virtual technical platform, At the moment, mechanical & electrical equipment was used to rolling indicator. It is studied the control of graphic display on DSP graphics processing technology, combined with the sensor signal, virtual level indicating shall be putted into effect. Building the development platform as the core of DAM6416P, Research & analysis of line generating algorithm and match corresponding colors for graphics. On this basis, apply bresenham algorithm to digital signal that tilt sensor output, design centrosymmetric graph, output in real time to the dual plasma display, at the same time, In order to more visually intuitive, match different colors for output.

Keyword: DAM6416P; Centrosymmetric Graph; Bresenham Algorithm; Color Space

I. INTRODUCTION

Ship-borne aircraft flying on the meteorological vagaries of the ocean, taking off and landing on the small size and unstable flight deck, consequently, there are many factors of the crisis in flight safety [4]. According to the United States Security Center statistics, flight accident that ship-borne helicopter pilot is 10 times of the bomber pilot, is 54 times of the civil aviation pilot. Flight accidents not only endanger the safety of the helicopter, but also endanger the survival of the ship, because of containing a large number of combustible on the ship. So, safe and effective ship-borne aircraft guidance systems will be an important means to ensure the pilot life safety and the helicopter land safely. Currently, ship-borne aircraft guidance systems mainly use mechanical & electrical equipment, and use the mechanical arm built-in incandescent lamp as the display of the rolling indicator. With the continuous development of electronic devices, the performance of the electronic components is increasing, application of high-precision, high stability of electronic devices to replace mechanical equipment has become a trend.

II. RELEVANT WORK

In order to explore the feasibility of application of virtual technical platform to rolling indicator, built the test platform for rolling indicator as the core of DAM6416P, tilt sensor and plasma display. At the same time, according to the DSP's super capacity in the graphic image processing [2], designed a level indicator graphics to indicate the rolling condition by collecting digital signal that tilt sensor output.

A. Hardware Design

Test platform was composed of three parts. The part of

signal acquisition is used to collect the rolling condition by tilt sensor. The part of digital signal processing was used to process digital signal that tilt sensor output. The part of video output was used to convert digital signal into analog signal and output it. The test platform is shown in figure1.

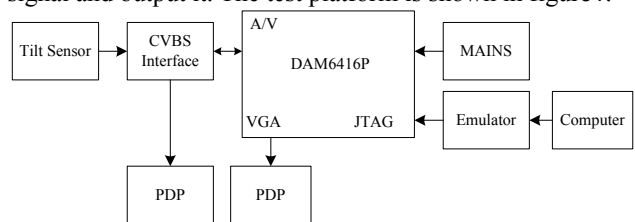


Figure I. TEST PLATFORM

Therein, ZC-TD-30S, as a single axis tilt sensor, can output RS-232 serial digital signal. Its measurement range from -30° to $+30^{\circ}$, resolution 0.1° , accuracy of less than 0.5° , operating temperature from -40° to $+40^{\circ}$.

AS image-processing platform, DAM6416P was a development board based on TMS320C6416. DAM6416P provided RS-232, RS-422 and other communication interface. At the sametime, it supported for composite video and S terminal input, output, and individual output VGA format image.

DAM6416P took a video and audio interface box, which provided audio, composite video, S-Video, MIC, and RS-232 interface.

Simulation used the TDS510 emulator.

Plasma display owned wide perspective, the pure plane without distortion, no electromagnetic interference, no electromagnetic radiation, durable and some unique advantages [2]. The system adopted the Xoceco PS-42D8, which had multiple input interfaces, such as CVBS, VGA, and other interfaces.

B. Software Flow

Because some of the pins in decoder chip BT864 have been reused, DAM6416P could not simultaneously output two channels video signal, so time-sharing multiplexing technology was used in programming. Programs and software flow were given with modular programming idea, the software flow was shown in figure II .

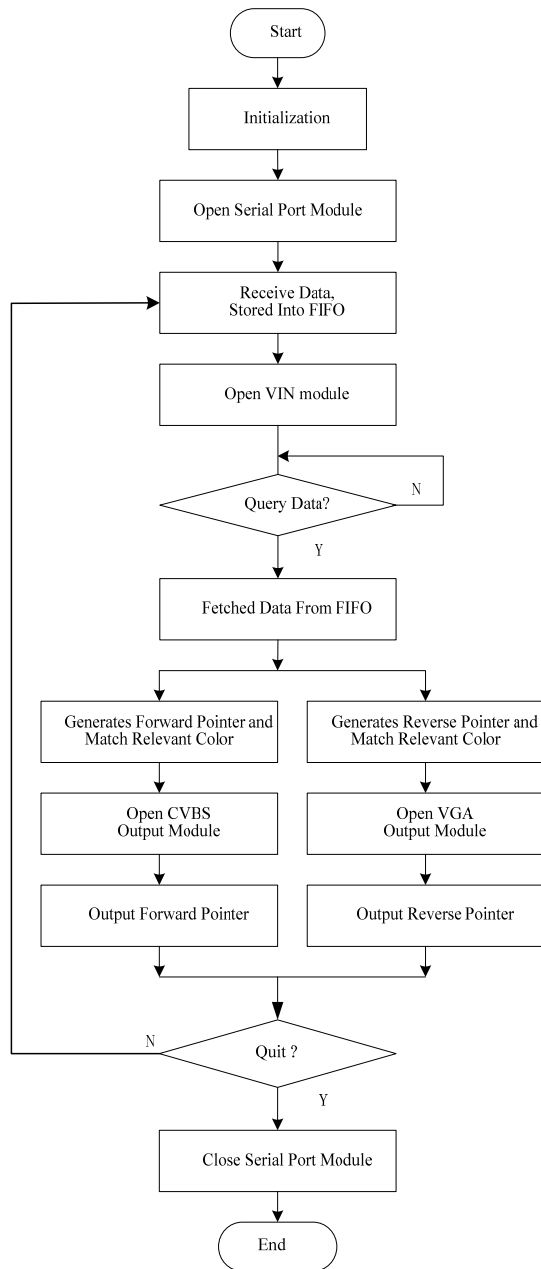


Figure II. SOFTWARE FLOW

III. EXPERIMENTAL ANALYSIS AND RESULTS

A. Analysis and Comparison of Drawing Line Algorithm

Graphical output to plasma display was mainly linear. There were three commonly used line drawing algorithm, namely the numerical differential line drawing algorithm, the dotted line drawing algorithm, and the bresenham line drawing algorithm. So we should choose the right line drawing algorithm, and select algorithm according to the drawing line speed. three separately programs were written to test the speed of these line drawing algorithm, the number

of clock cycles consumed by the line drawing process could be seen under the code composer studio environment, thereby calculated out the execution time. The data as shown in table I .

TABLE I. EXECUTION TIME

Algorithm	Numerical Differential	Dotted	Bresenham
clock cycles	76892	52444	13901
execution time (ms)	0.128	0.088	0.023

It could be seen that Bresenham algorithm has the fastest performance from the above table, So Bresenham algorithm was adopted to write the graphics display program.

B. Bresenham Algorithm

The Bresenham algorithm was proposed by Bresenham in 1965. With bresenham line drawing algorithm writing program, the coordinate system was divided into eight symmetrical regions first of all[6]. as in figureIII.

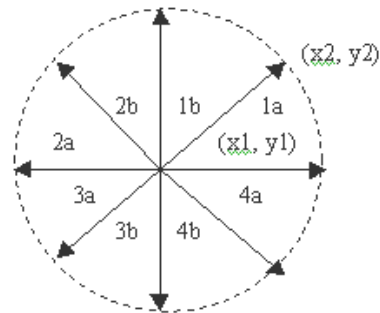


Figure III. DIAGRAM OF BRESENHAM ALGORITHM

Bresenham algorithm in the first quadrant was given as follows:

- (1) draw (x, y) , $dx = x_2 - x_1$, $dy = y_2 - y_1$, calculated initial error $d_1 = 2dy - dx$, $i = 1$;
- (2) Calculated the next point $x_{i+1} = x_i + 1$, if $d_i > 0$, $y_{i+1} = y_i + 1$; else $y_{i+1} = y_i$;
- (3) draw (x_{i+1}, y_{i+1}) ;
- (4) Calculated next error d_{i+1} , if $d_i > 0$, $d_{i+1} = d_i + 2(dy - dx)$; else $d_{i+1} = d_i + 2dy$;
- (5) $i = i + 1$, if $i < dx + 1$, Jump to (2); else finish.

According to the symmetry, straight lines could be drawn in all 8 directions by the above algorithm.

C. The Analysis and Experiment on Color Space

For more intuitive representation of the ship's swing state, the output graphics must be displayed in different colors; I respectively selected yellow, green and red.

Because of influence of the brightness on color, different display devices would be slightly different[2], so the values of colors matched the colors of R03, G04, Y05 in film color standard sample card GSB (2000 edition), need to be obtained by experiment.

DAM6416P output two types of video data, namely format of RGB565 and format of YUV422. The RGB565 format stored in the memory as shown in table II .

TABLE II. THE RGB565 FORMAT STORED IN THE MEMORY

Bits	D15—11	D10—5	D4—0
Component	Red	Green	Blue

Wrote a graphical display program, and set color value for the pixel changed from 0 to 65535, I obtained the color numerical value of RGB565 format on plasma display, As shown in table III.

TABLE III. THE COLOR VALUE OF RGB565

Color	National Standard	Values of the Color
raise the red	R03	43208
lemon yellow	Y05	1312
medium green	G04	32768

In the YUV422 format, the color of the even pixel points decided jointly by the luminance signal Y, the blue color difference signals U itself, and the red color difference signal V of next pixel point. The color of the odd pixel points decided jointly by the luminance signal Y, the red color difference signals V itself, and the blue color difference signal U of previous pixel point; According to this characteristic, wrote a graphical display program, kept adjacent pixel in equal to the luminance value, and U values changed from 0 to 255, and turn one circle, V values will add to one count between 0 and 255, consequently observed color variation. The main program as follows:

```

for (U= 0;U<=255;U++) {
for (V=0;V<=255;V++) {
for (x=0;x<=351;x++) {
for (y=0;y<100;y++) {
m=x;
n=m/2;
if (n>floor(n)) {
*(pVideoOutFrame+x+h*y)=pow(2,8)+U; }
else {
*(pVideoOutFrame+x+h*y)=pow(2,8)+V; }
}
}
VOUT_putFrame (hVout, pVideoOutFrame,
IEKC64_VIDEO_NO_WAIT) ;
}
outVideoIndex=(outVideoIndex+1)%2;
pVideoOutFrame=&OutputBuffer [outVideoIndex][0];
temp_address2=pVideoOutFrame;
for (i=0;i<FRAME_SIZE_IN_PIXELS;i++)
*(temp_address2++) = *(temp_address1++);
}

```

Experimental results show that pixel color was the same as adjacent pixel when the luminance signal Y was the same.

When U values ranged from 0 to 255, the graphics color changed from green to blue, the cut-off point was closed to 128. and when V values ranged from 0 to 255, the graphics color changed from green to red, the cut-off point was closed to 128. I obtained the color numerical value of YUV422 format on plasma display by experiment, As shown in table IV.

TABLE IV. THE COLOR VALUE OF YUV422

Color	National Standard	Y	U	V
raise the red	R03	16	0	255
lemon yellow	Y05	255	0	157
medium green	G04	20	0	0

D. Forward and Reverse Pointer

Defined the size of image for 352 ×240 pixels in each plasma display, and the straight lines had a width of 240 pixels and height of 52 pixels. Arrow had a width of 60 pixels. Rectangle area had a width of 50 pixels. According to the coordinate and the endpoint of each segment graphics, assigned to pixels one by one when generated forward pointers.

Reverse pointers was centrosymmetric about forward pointers. So called centrosymmetric graph is coincident with rotation of 180° in the plane. In program of reverse pointer, simply traverse the coordinates according to the characteristics of centrosymmetric.

E. Experimental Results

Before used DAM6416P development board, TI code composer studio with version 2.2 or above must been installed. And must used DAM6416.gel as initializations GEL file when configure code composer studio. Board-level development libraries, IEKLIB were provided by DAM6416P for DSP program development, which provided nine functional modules for users, and must be initialized before used these modules [5].

1) UART Module Initialization

UART module initialization was completed by function UART_open (&uartcfg). As a data structure, uartcfg was used to configure UART port, and used to specific configure baud rate, parity, buffer size and address. The main initialization program as follows:

```

uartcfg.BaudRate = UART_9600BDS;
uartcfg.Parity = UART_NO_PARITY;
uartcfg.RxBufferSize = BUFFER_SIZE;

```

After successful initialization, the serial port could be opened, and red sensor data and stored them in the FIFO.

2) Video Module Initialization

Video module initialization was divided into two parts and covers VIN (video input) and VOUT (video output) module initialization. VIN module initialization was completed by function VIN_open (&videoIn, &hVin). As a data structure, videoIn was used to configure video input, and used to specific configure resolution, data format, nature of the signal, frame time decimation ratio, working mode, the buffer. The main initialization program as follows:

```

videoIn.Standard=DEMO_STANDARD;

```

```
videoIn.Resolution=DEMO_RES;
videoIn.FrameFormat=YUV422PIXEL;
VideoInSelect=COMPOSITE;
videoIn.isOneShot=FALSE;
videoIn.nFramesInBuffer=Frames_Count;
videoIn.nFramesToKeep=Frames_ToKeep;
videoIn.pCaptureBuffer= (Uint32*) CaptureBuffer;
```

If initialization was successful, it created VIN objects, and returned the handle to hVin. At the same time, returned a status indicating the success of operation.

VOUT module initialization was completed by function VOUT_open (&videoOut, &hVout). As a data structure, videoOut was used to specific configure video output procedure parameters. Took VGA output as an illustration, the initialization procedures as follows:

VOUT module initialization was completed by function VOUT_open (&videoOut, &hVout), As a data structure, videoOut was used to specific configure video output.

```
videoOut.Standard=DEMO_STANDARD;
videoOut.Resolution=DEMO_RES_VGA;
videoOut.FrameFormat=RGB565PIXEL;
videoOut.VideoOutSelect=VGA;
```

If initialization was successful, it created VOUT objects, and returned the handle to hVout. At the same time, returned a status indicating the success of operation.

3) Data Processing

The slope value of roll indicator pointer was associated with the roll angle of the ship, and roll indicator pointer was controlled by custom variable z before linked to sensor. The main program as follows:

```
for (z= 30; z<=210; z++) {
    .....
    h=352;
    y1=120;
    p=120-4*(120-z)/5;
    dx=240;
    dy=p-y1;
    k=dy/dx ;
    ..... }
```

To control the rolling pointer by tilt sensor, here should first proportionally converted its output value into variable z. Variable z was from 30 to 210, and output angle value of the tilt sensor was from - 150 to + 150, which indicated the actual angle was from - 15° to + 15°. Thus, I figured out conversion formula between the angle and variable z, as shown in following equation:

$$angle = 5 \cdot (z - 120) / 3 \quad (1)$$

4) Output signal to the dual plasma display

Before output video image, the VOUT module, has been initialized, must be started by calling the function VOUT_start(&hVout). Then, it was allowed to output VGA format image by calling the function VOUT_putFrame (hVout,OutputBuffer[outVGAIndex]). And output YUV format image by calling the function VOUT_putFrame (hVout, pVideoOutFrame). Finally, output them by VOUT module through analog front-end circuit.

Output interfaces provided by DAM6416P had composite video interface and VGA interface. Composite video interface output forward pointer, and VGA interface output reverse pointer. Video signal acquisition is completed by video decoder called BT835, but data format output by BT835 was YUV4:2:2, so it was need to convert YUV4:2:2 format to VGA format by the following equation [5]:

$$coeff[5]=\{0x2543,0x3313,-0x0C8A,-0x1A04, 0x408D\} \quad (2)$$

5) Colors In Graphics

There was a yellow band parallel to the ground in the plane formed by the dual plasma display, and included angle between the arrows, on the edge of band, and horizontal midline on the plasma display indicated the current swing angle of ship. In order to convenient observation, the angle was represented by rectangular area. The regional color was yellow when the angle was between 0 to 5 degrees. There were two kinds of color when the angle was between 5 to 8 degrees, it was still yellow between 0 to 5 degrees, but green all the rest. There were three kinds of color when the angle was greater than 8 degrees; it was still yellow between 0 to 5 degrees, green between 5 to 8 degrees, but red all the rest.

IV. CONCLUSION

In laboratory, the test platform is fitted on three degree of freedom turntable based on parallel mechanism, and as signal source of rolling indicator, the turntable is used in attitude simulation for ship. In the process of testing and running, the centrosymmetric graphics display program has been stable and reliable, and guarantee the system's flexibility and real-time, the double plasma display is better than ordinary LED display module, moreover also improved the effect of the graphics display. It is proved that the test platform has good practicability and value to popularize.

V. REFERENCES

- [1] Xiao-lin Zhu, Yong Cai and Jian-sheng Zhang, "line drawing algorithm based on sorting pixel chains," Application Research of Computers, Vol.29 No.3 Mar.2012.
- [2] Qiong Cheng, Sheng-chao Huang and Hui-shan Bai, "The Amendment of Dot-Matrix LCD Painting Algorithm Implementation Based on DSP," Journal of Hubei University of Technology, Vol.24 No.1 Feb.2009.
- [3] Yun-lin Liu and Shu-dong Wang, "Conversion of Color Space Between YUV and RGB on SSE2 Technique," Journal of Image and Graphics, Vol.15 No.1 Jan.2010.
- [4] Hai-dong Wang, Kui Huang, Yu-quan Bi and Bing-heng Yang, "Kinematics analysis and simulation of the test tilter for the optical landing aid system," Ship Science and Technology, Vol.34 No.2 Feb.2012.
- [5] Wintech Digital Systems Technology Corp, "software development guide based on DAM6416P image processing platform," 2003
- [6] S. Fawad, "Adapting Bresenham Algorithm," Journal of Theoretical and Applied Information Technology, Vol.2 No.2 Aug. 2006, pp.27-30.