

```

và cho  $\{i = 0 \text{ đến } 7\}$ 
{
    tính :
        line =  $m[i]*x + c[i] - y$ 
    nếu (line < 0)
        thì  $l[i] = 1$ 
        nếu không thì  $l[i] = 0$ 
    }

```

Từ hình 11.10 ta có thể tính màu sắc của một điểm dựa trên giá trị bên trên của  $l[i]$ ;  $\{i = 0 \text{ đến } 6\}$  theo các bước:

```

if (circ < 0) color = white

else if ( $l[0] = 1$  and  $l[6] = 1$ ) color = green

else if ( $l[0] = 0$  and  $l[1] = 1$ ) color = yellow

else if ( $l[1] = 0$  and  $l[2] = 1$ ) color = skin tone

else if ( $l[2] = 0$  and  $l[3] = 1$ ) color = red

else if ( $l[3] = 0$  and  $l[4] = 1$ ) color = magenta

else if ( $l[4] = 0$  and  $l[5] = 1$ ) color = blue

else if ( $l[5] = 1$  and  $l[6] = 0$ ) color = cyan

```

Kết quả trên mô tả một lớp màu mà có thể dùng lựa chọn sửa lại sắc của một màu đặc biệt mà không làm ảnh hưởng đến các màu còn lại trên ảnh như sau:

1. Chọn một sắc màu hoặc là các sắc màu mà bạn muốn sửa trên ảnh.
2. Quét ảnh và dùng lớp màu để xác định liệu một điểm có thuộc về sắc màu đã được lựa chọn.
3. Nếu một điểm không thuộc sắc màu lựa chọn, thì cho phép hiển thị thẳng lên màn hình mà không cần có một sự thay đổi nào. Dù thế nào đi chăng nữa, nếu nó thuộc về sắc màu lựa chọn, thì thay đổi đã được làm đối với điểm này là nó được phản xạ với sắc màu tự nhiên nhiều hơn.

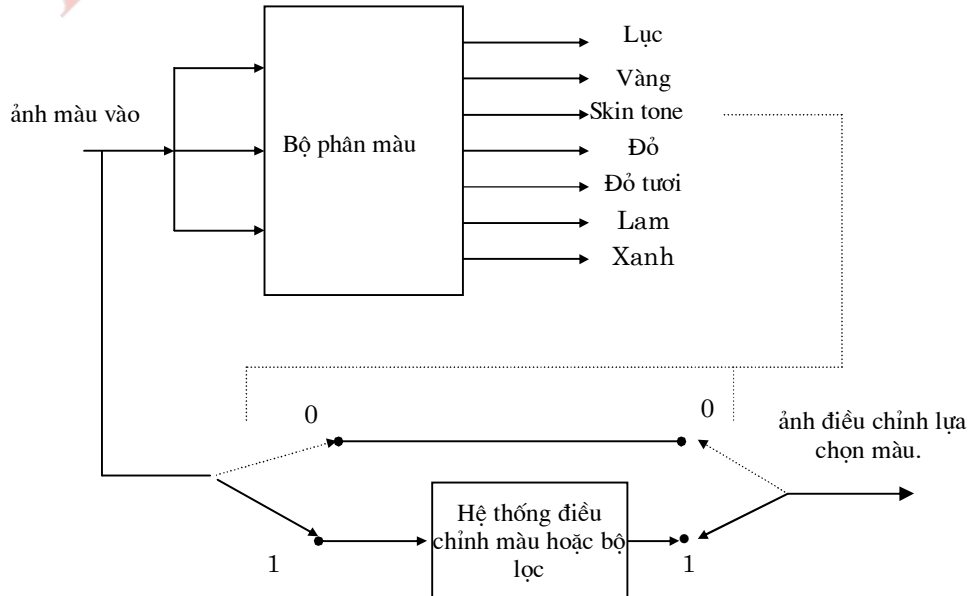
Các bước trên thể hiện trong hình 11.11. Lớp màu giới thiệu hình 11.11 cho giá trị 1 cho chỉ một trong số 7 đầu ra của sắc màu mà điểm

ảnh này thuộc về, các đầu ra còn lại có giá trị 0. Tín hiệu ra của sắc màu lựa chọn là một mạch logic OR và làm chuyển mạch đường dẫn tín hiệu của điểm ảnh. Nó có thể đi qua bộ lọc sửa lại màu hoặc đi thẳng ra màn hình. Giải thuật này cho ở chương trình 11.7.

### **Chương trình 11.7 "ADJUST.C" chương trình sửa lại sắc màu của một số sắc màu đặc biệt có lựa chọn.**

```
/* This program illustrates the procedure
for selective color adjustment for images using
color tones derived from the chromaticity
diagram. */
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <io.h>
#include <ctype.h>
#include <math.h>
#define sqr(x) ((x)*(x))
```



**Hình 11.11** Điều chỉnh sắc màu có lựa chọn.

```
#define white (circ<0)
#define green (l[0]&&[16])
#define yellow ((!l[0])&&l[1])
#define skin_tone ((!l[1])&&l[2])
#define red ((!l[2])&&l[3])
#define magenta ((!l[3])&&l[4])
#define blue ((!l[4])&&(!l[5]))
#define cyan (l[5]&&(!l[6]))
#define condition (skin_tonellred)&&white)

void main()
{
    int i,j,n1,n2,image_width, image_length,dR,dG,dB,yt;
    int R1,G1,B1;
    char file_name[14],ch;
    unsigned int *buff,color,R,G,B;
    float Y,scale;
    float nsq,tmp;
    float X,Z,D,x,y;

    float m[]={3.3600, 1.260274, 0.663317, -1.029762,
               -61.75, 0.384, -0.875};
    float c[]={-0.785880, -0.086671, 0.112116, 0.675911,
               20.89575, 0.205128, 0.624375};
    float r=0.025;
    float lt,circ;
    int l[7];
    FILE *fptri, *fptro;

    clrscr();
    printf("Enter file name for input image ->");
    scanf("%s",file_name);
    if((fptri=fopen(file_name,"rb"))==NULL)
    {
        printf("%s does not exist.", file_name );
        printf("\nPress any key to exit.");
        getch();
        exit(1);
    }
    nsq=filelength(fileno(fptri))/2.0;
```

```

printf("Is this a square image ?");
printf
("\n i.e. Is image_length=image_width (y or n)? -> ");
while(((ch=(char)getche())!='y')&&(ch!='n'));

switch(ch)
{
case 'y':
    image_length=image_width-sqrt(nsq);
    printf("\n Image size = %d x %d" ,
    image_length, image_width);
    break;
case 'n':
    printf("\nEnter image_width-->");
    scanf("%d",&image_width);
    image_length=nsq/image_width;
    printf("Image length is %d", image_length);
}
yt=wherey();
again:
gotoxy(1,yt+1);
printf
("Enter file name for saving color adjusted image.-->");
scanf("%s",file_name);
if(access(file_name,0)==0)
{
printf("File exists. Wish to overwrite? (y or n)-->");
while(((ch=tolower(getch()))!='y')&&(ch!='n'));
putch(ch);
switch(ch)
{
case 'y' :
    break;
case 'n' :
    gotoxy(1,yt+1);
    delline();
    goto again;
}
}
}
fptro=fopen(file_name,"wb");
printf("\n For selected color tone(s) enter:");

```

```

printf("\n Change in Red primary ---->");
scanf("%d",&dR);
printf("Change in Green primary-->");
scanf("%d",&dG);
printf("Change in Blue primary --- >");
scanf("%d",&dB);
buff=(unsigned int *)malloc(image_width*sizeof(int));
gotoxy(70,25);
textattr(WHITE+(GREEN<<4)+BLINK);
cputs("WAIT");

/* Algorithm */
for(n1=0; n1<image_length;n1++)
{
    gotoxy(1,14);
    printf(" Transferred line %-4d. ",n1);

    /* Transfer row n1 of the image. */
    fread(buff,sizeof(int),image_width,fptri);

    for(n2=0; n2<image_width; n2++)
    {
        color=buff[n2];
        B= ( 0x001F & color);
        G= ( 0x03E0 & color) >> 5;
        R= ( 0x7C00 & color) >> 10;
        X=2.7690*R+1.7518*G+1.1300*B;
        Y=R+4.5907*G+0.0601*B;
        Z=0.0565*G+5.5943*B;
        D=X+Y+Z ;
        if(D==(float)0.0)
            continue;
        else
        {
            x=X/D; y=Y/D;
            circ=sqr(x-0.333)+sqr(y-0.333)-sqr(r);
            for(j=0;j<7;j++)
            {
                lt=m[j]*x+c[j]-y;
                if(lt<0) l[j]=1;
                else l[j]=0;
            }

```

```

    }

if(condition)
{
    R1=R; G1=G; B1=B;
    if((((R1+dR)>=0)&&((R1+dR)<=31))&&
        (((G1+dG)>=0)&&((G1+dG)<=31))&&
        (((B1+dB)>=0)&&((B1+dB)<=31))))
    G1+=dG;
    B1+=dB;
    R=R1; G=G1; B=B1;
}
color=B+(G<<5)+(R<<10);
buff[n2]=color;
}
}
fwrite(buff,sizeof(int),image_width,fptro);
}
fclose(fptri); /* close input image file. */
fclose(fptro); /* close output image file. */
gotoxy(70,25);
textattr(WHITE+(BLACK<<4));
cputs("    ");
gotoxy(1,15);
printf("Filtering is now completed.");
}

```

Trong chương trình 11.7 câu lệnh

```
#define condition (skin_tonellred)&&(!white)
```

định nghĩa vùng cần điều chỉnh trên ảnh. Tất nhiên bạn cũng có thể định nghĩa bất kỳ màu khác hoặc những màu sắc bạn muốn. Để minh họa chương trình bạn có thể sử dụng ảnh "AUTHOR.IMG". Bởi vì đây là một ảnh của chính tôi, tôi có thể nói chắc chắn với bạn rằng màu da sẽ sáng khi được hiển thị qua vĩ mạch PIB ATI. Hầu hết người Ai cập có nước da nâu nhạt và hơi đỏ. Nước da sẽ đen hơn nếu bạn đi về hướng nam Ai cập (vào châu Phi sâu hơn). Để chỉnh màu da, bạn có thể chạy chương trình trên với những tham số sau: thay đổi màu đỏ = 3; thay đổi màu xanh da trời = -1; thay đổi màu xanh lá cây = -1. Tất nhiên chương trình sẽ nhắc nhở bạn những số này. Kết quả cuối cùng sẽ được lưu trữ trên đĩa trong

một file mà bạn định nghĩa lúc bắt đầu chương trình. Chú ý rằng chỉ có nước da và khu vực màu đỏ được chỉnh sửa, và những phần khác của ảnh không bị ảnh hưởng. Màu da rất giống với màu da tự nhiên của tôi. Nếu bạn muốn bạn có thể làm chương trình thể hiện sự liên hệ và cho phép thay đổi R,G,B thực hiện qua các phím mũi tên và các phím + và -, chương trình này đã được làm có tên DISPPIB.C.

Chương trình 11.7 đồng nhất màu của mỗi điểm độc lập riêng biệt với các điểm lân cận của nó. Vì vậy một điểm bản thân nó không định nghĩa màu của khu vực, sẽ là tốt hơn khi lấy giá trị trung bình trên các điểm ảnh xung quanh điểm đó. Giá trị trung bình này dùng để đồng nhất lớp màu cho điểm đó. Phương pháp xấp xỉ được dùng để kiểm tra các điểm lân cận của mỗi điểm, và xác định rõ màu cơ bản của nó dựa trên các màu của những điểm lân cận. Ví dụ việc trộn giữa màu đỏ và xanh da trời cho ta màu nâu, đó không phải là màu cơ sở trong biểu đồ màu sắc.

### **Bài tập 11.3**

1. Lập một thuật toán xác định các màu không thấy trên sơ đồ màu từ các hỗn hợp màu tìm thấy trên sơ đồ màu.
2. Chuyển thuật toán thành chương trình C, dùng nó để thay đổi các sắc màu.
3. Kiểm tra chương trình của bạn trên ảnh “AUTHOR.IMG”.

Có thể sẽ là tốt hơn nếu chúng ta có được sự phân lớp màu sắc. Rõ ràng là biểu đồ màu sắc chỉ cung cấp phân lớp thô của màu. Điều này sẽ rõ ràng nếu chúng ta nói về màu da mà dùng R, G, B phân biệt trên một ảnh hiển thị trên vĩ mạch PIB. Nếu sau đó chúng ta ánh xạ kết quả lên biểu đồ màu, chúng ta sẽ thấy màu chiếm giữ ở các khu vực màu vàng, màu xám da, và khu vực màu đỏ (giống như chương trình được cho ở chương 12). Nó không chiếm tất cả các khu vực như đã nói ở trước. Vì vậy, nếu đặt một sự phân lớp để nhận dạng dải màu trong khu vực, thì kết quả phân biệt mức xám sẽ tốt hơn. Như vậy phân biệt màu có thể học màu được biết như *mạng thần kinh nhân tạo*-chủ đề của chương 12.