Assignment name : Image viewing program.

Date : 11.11.15

Submitted by : Redwan Ahmed Rizvee,Roll- 09, Group – G3

Concept:

Here we have worked with pgm image.Pgm Format basically have pixels of different color and their different type orientation shows the same image in different view. To take it logically we can think that the pixels are in a two dimentional matrix and to use this matrix’s elements we can have different view of our image.

Processes:

1. Rotating the image (+90) degree:

if we want to rotate an image (+90) degree , we can do like this

1 2 3 3 6 9

4 5 6 -> 2 5 8

7 8 9 1 4 7

so previous length and breadth will be now new width and length

Pseudo code:

for(i = col - 1; i >= 0; i--)

{

for(j = 0; j < row; j++)

{

fprintf(fp1,"%d ",matrix[j][i])

}

fprintf(fp1,"\n");

}



picture : 1

2 : Rotating the image (-90) degree :

logic :

1 2 3 7 4 1

4 5 6 - > 8 5 2

7 8 9 9 6 3

Code :

for(i = 0; i < col;i++)

{

for(j = row - 1; j >= 0; j--)

{

fprintf(fp1,"%d ",matrix[j][i]);

}

}

Picture :



1. Rotating the image 180 degree

1 2 3 9 8 7

4 5 6 -> 6 5 4

7 8 9 3 2 1

Code :

for(i = row - 1; i>= 0; i--)

{

for(j = col - 1;j>=0;j--)

{

fprintf(fp1,"%d ",matrix[i][j]);

}

}

Picture – 03 :



Process 4: Flipping the image

1 2 3 3 2 1

4 5 6 -> 6 5 4

7 8 9 9 8 7

Code :

for(i = 0; i < row; i++)

{

for(j = col - 1; j >= 0; j--)

{

fprintf(fp1,"%d ",matrix[i][j]);

}

Picture – 04 :



Picture 05 : Negative image :

to do this we will take the negation of our pixels or prime colour of the pixels, means we will substract every pixel from the highest pixel

like

1 2 3

4 5 6

7 8 9

here highest colour is 9

so now

new pixels

9 7 6

5 4 3

2 1 0

Code :

for(i = 0; i < row; i++)

{

for(j = 0; j < col; j++)

{

fprintf(fp1,"%d ",high - matrix[i][j]);

}

}



Process – 06 : Giving a border

to give this border with black color we made out ouline pixels black and main pixels unchainged

like

1 2 3 0 0 0

4 5 6 -> 0 5 0 0 = Black Color pixel

7 8 9 0 8 0

Code :

1. make pixels 0 from (oth row to thickness – 1th row)
2. make pixels 0 from (end – thicknessth + 1th row to the end row)
3. 0 th col to thickness – 1th col
4. from col – thickness + 1th row to end col

Picture : 06



Process 07 : Cropping

to do this first we are given our necessary co – ordinates

top leftx , top lefty

high rightx, high righty

so new length = high rightx – top leftx + 1

so new width = high righty – top lefty + 1

new dimension :

(new length, new width)

like

1 2 3 5 6

4 5 6 -> co – ordinates : (1,2) to (2, 2) 8 9

7 8 9

Code :

fprintf(fp1,"%d %d\n",highx - leftx + 1,highy - lefty + 1);

fprintf(fp1,"%d\n",high);

for(i = lefty; i <= highy; i++)

{

for(j = leftx; j <= highx; j++)

{

fprintf(fp1,"%d ",matrix[i][j]);

}

fprintf(fp1,"\n");

}

}



By this way we can process an image and see them in different view.I have used function 7 here to cropping image differently and also function 8 to crop image with a different view .

We have used here Lena.ascii.pgm to do all the works.It was a square matrix, through this program we can control and handle with any type image and must be careful about our dimension.

Thank you.