

Lab Exercise - Week 10

This week we use 2D arrays to do some cool image processing. You might have heard of many popular image file formats like JPG, PNG etc. We will use an unpopular but very simple image file format called PGM. It is designed to be extremely easy to learn and write programs for. (It's so simple that most people will simply reverse engineer it because it's easier than reading this specification). Here is an example image file:

```
P2
24 7
15
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 3 3 3 3 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 15 0
0 3 3 3 0 0 0 7 7 7 0 0 0 11 11 11 0 0 0 15 15 15 15 0
0 3 0 0 0 0 0 7 0 0 0 0 0 11 0 0 0 0 0 15 0 0 0 0
0 3 0 0 0 0 0 7 7 7 7 0 0 11 11 11 11 0 0 15 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

A PGM image represents a grayscale graphic image. The name "PGM" is an acronym derived from "Portable Gray Map." This is a simplified description of its format. (if you want the complete description, see https://en.wikipedia.org/wiki/Netpbm_format)

1. The first line of the file contains a magic number. It has to be P2 for PGM files. The file may have lines starting with a "#" symbol which are comments. For today, none of the image files that we work with will have comments and hence we can forget about that.
2. The second line has the width and height of the image
3. The third line contains the maximum value M of the pixel brightness.
4. Then we have a sequence of (width x height) integer values, each in the range 0 to M. For small images, each row will correspond to one horizontal line of the image, but for larger images one line of the image may be split into many rows in the file.

You can download these sample PGM images: cat.pgm and dog.pgm to your working directory (CS1200/Week9). Open it using the command

```
eog cat.pgm &
```

to view the image (eog is an image viewer in Linux).

Then open it using the command

```
gedit cat.pgm &
```

to see the file contents.

Experiment 1.

Program name: negative.c

Study the structure of PGM files explained above and write a C program to read the pixels of an image from a PGM file in to a 2D array where the pixels from one line of the image are stored in one row of the array. (You can assume that the maximum size of the images are 500 x 500). Now invert the image by replacing each pixel value x by $\text{maximum} - x$. Save this new array as another PGM image file and view this PGM file in the image viewer.

The program should ask the user to type the input filename and output filename. You can test your program with cat.pgm and dog.pgm as input files. Save the outputs as negative_cat.pgm and negative_dog.pgm

Experiment 2a.

Program name: fliplr.c

Write a program to read an image from a pgm file and store the left-to-right mirror image of the image into a new file.

The program should ask the user to type the input filename and output filename. You can test your program with cat.pgm and dog.pgm as input files. Save the outputs as fliplr_cat.pgm and fliplr_dog.pgm

Experiment 2b.

Program name: flipud.c

Write a program to read an image from a pgm file flip it upside down and store the upside down image into a new file.

The program should ask the user to type the input filename and output filename. You can test your program with cat.pgm and dog.pgm as input files. Save the outputs as flipud_cat.pgm and flipud_dog.pgm

Experiment 3

Program name: superimpose.c

Write a program to read two images from two pgm files and superimpose the two images. That is, first check whether the two images are of the same size. If no, then exit. If yes then replace each pixel value by the average (rounded to the lower integer) of the corresponding pixel values from the two images. Store the resulting image into a new file.

The program should ask the user to type the two input filenames and one output filename. You can test your program with cat.pgm and fliplr_cat.pgm as input files. Save the outputs as two_cats.pgm

Experiment 4

Program name: decolour.c

PPM is an image format that is very similar to PGM but for colour images. Here is a very small example

```
P3
3 2
255
255 0 0 0 255 0 0 0 255
255 255 0 255 255 255 0 0 0
```

Note that the magic number is now P3 instead of P2. More importantly after the width, height and max pixel value we have 3 times (width x height) integers. Every 3 integers represent the Red, Green and Blue values of one pixel. You can open cat.ppm and dog.ppm files in gedit to see more examples. You can open them in an image viewer (eog) to see the colour pictures.

Write a program to read a PPM colour image and convert it to a PGM grayscale image of the same size. You can obtain the grayscale value by averaging the Red, Green and Blue pixel values.

The program should ask the user to type the input filename and output filename. You can test your program with cat.ppm and dog.ppm as input files. Save the outputs as gray_cat.pgm and gray_dog.pgm

Experiment 5

Program name: twincats.c

Write a program which will produce this image - twincats.ppm as output.

Homework 1

Program name: transpose.c

Write a program to read an image from a pgm file into a 2D array, find the matrix-transpose of the 2D array and store the transposed image into a new file.

The program should ask the user to type the input filename and output filename. You can test your program with cat.pgm and dog.pgm as input files. Save the outputs as transposed_cat.pgm and transposed_dog.pgm

Can you visualize the transpose operation as some kind of mirroring or rotation?

Homework 2a

Program name: rotate90.c

Write a program to read a colour image from a ppm file, rotate the image counter clockwise by 90 degrees and store the rotated image into a new file.

The program should ask the user to type the input filename and output filename. You can test your program with cat.ppm and dog.ppm as input files. Save the outputs as r90_cat.pgm and r90_dog.pgm

Homework 2b

Program name: rotate180.c

Write a program to read a colour image from a ppm file, rotate the image by 180 degrees and store the rotated image into a new file.

The program should ask the user to type the input filename and output filename. You can test your program with cat.ppm and dog.ppm as input files. Save the outputs as r180_cat.pgm and r180_dog.pgm