CS 576 – Assignment 1
Instructor: Parag Havaldar

Assigned on Wednesday 09/02/09,
Solutions due on Tuesday 09/22/09 by 12:00 noon

This assignment will help you gain a practical understanding of Resampling and Filtering and how it affects visual media types like images and video.

Firstly, you will have to write a program to display images in the RGB format that we will give to you for testing. We have also provided a Microsoft Visual C++ project and java to display two images side by side which are the original and the output of your operation. This source has been provided as a reference for students who may not know how to read and display images. You are free to use this as a start, or write your own.

Input to your program will take four parameters where

- The first parameter is the name of the image, which will be provided in an 8 bit per channel RGB format (Total 24 bits per pixel). You may assume that all images will be of the same size for this assignment, more information on the image format will be placed on the DEN class website
- The second parameter will be a scale value. This will control by how much the image has to be scaled. It will be a floating point number, such as 0.5 or 1.2 and so on. Effectively, this will result in re-sampling your image.
- The third parameter will be angle (given in degrees) that will suggest by how much the image has to be rotated (about its center). The details for the rotation are given below.
- The fourth parameter will be a boolean value (0 or 1) suggesting whether or not you want to deal with aliasing. A 0 signifies do nothing (aliasing will remain in your output) and a 1 signifies that anti-aliasing should be performed.

To invoke your program we will compile it and run it at the command line as

```
YourProgram.exe C:/myDir/myImage.rgb S R A
```

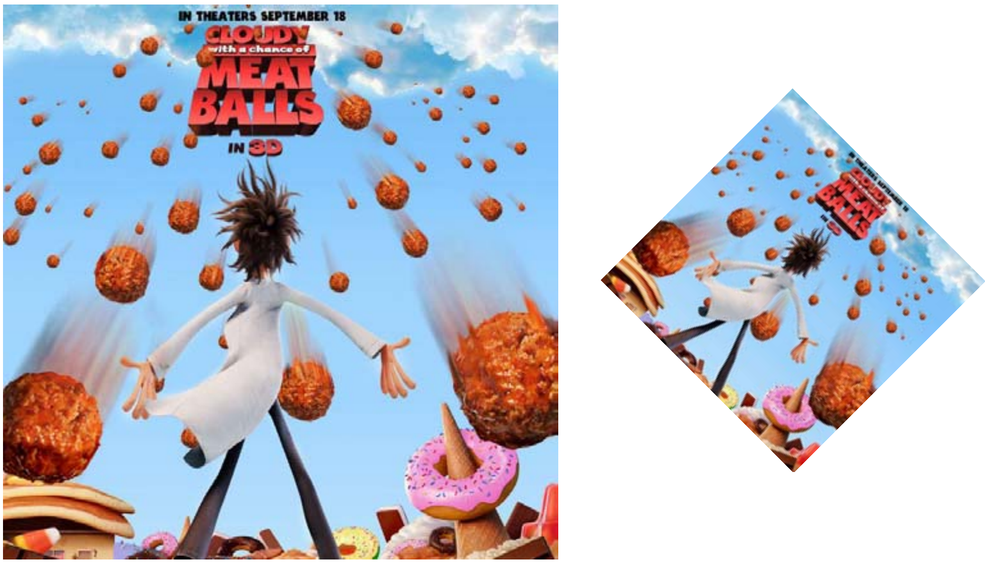
where *S R A* are the parameters as described above. Example inputs are shown below and this should give you a fair idea about what your input parameters do and how your program will be tested.

1. *YourProgram,exe image1.rgb 1 0 0*

Here the image will be scaled by 1.0, rotated by 0 degrees and no antialiasing performed. This effectively means that your output will be the same as your input

2. *YourProgram,exe image1.rgb 0.5 45 1*

Here the image will be scaled by 0.5, rotated by 45 degrees and anti-aliasing appropriately performed, giving the following output.



Now for the details - Every pixel has an RGB value and a x, y location. You will need to compute the new location of the pixel (x_new, y_new) and copy its RGB value (if not filtered) or an average neighborhood value (if filtered) into the new location. The matrices to scale and rotate are given below, where w and h are the width and height of the image, s is the scale factor and theta is the angle of rotation. Remember that your image's coordinates start from the upper left corner, you will need to offset the origin to the center of the image before you rotate and scale, else you will be rotating about the left corner and not the center of the image.

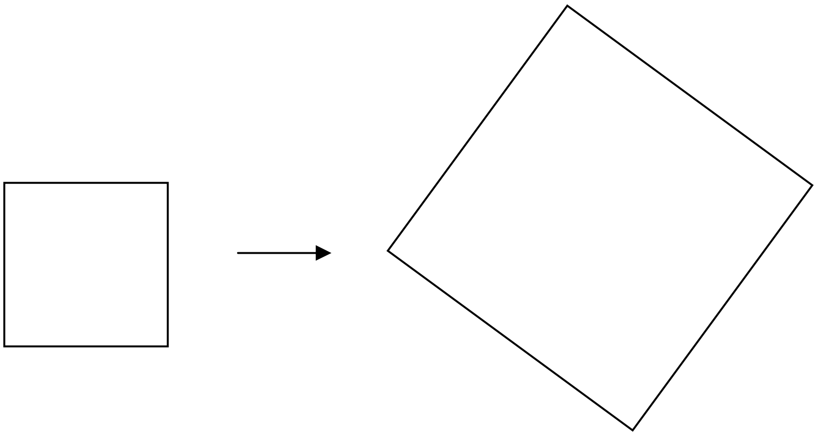
$$\begin{bmatrix} x\_origin \\ y\_origin \end{bmatrix} = \begin{bmatrix} x - w/2 \\ y - h/2 \end{bmatrix}$$

$$\begin{bmatrix} x\_transformed \\ y\_transformed \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x\_origin \\ y\_origin \end{bmatrix}$$

$$\begin{bmatrix} x\_new \\ y\_new \end{bmatrix} = \begin{bmatrix} x\_transformed + w/2 \\ y\_transformed + h/2 \end{bmatrix}$$

For anti-aliasing, you can perform a low pass filter by averaging the values of a 3x3 neighborhood. Remember to anti-alias each channel separately.

Give some thought to your implementation. If your original image samples (or pixels) define the domain of the transform function above and the final image samples define the range of the function, then the number of samples in your domain and range may not be the same (see example below). You will need to make sure that all the pixels in your final image have valid defined values!



What should you submit ?

- Your source code, and your project file or makefile, if any, using the submit program. ***Please do not submit any binaries or images***. We will compile your program and execute our tests accordingly.
- If you need to include a readme.txt file with any special instructions on compilation, that is fine too.