CMP SCI 6420 **Image Processing and Computer Vision**
Assignment # 1 Due Date: February 15, 2023

Porter-Duff Operators

Purpose

The purpose of this assignment is to implement the Porter-Duff operators for composition of images. The operators will be tested by a driver program.

Task

Porter-Duff operators are used for composition of images with binary masks. Their most common use is in green screen normally used for weather telecasts. The binary masks define the support of pixels in each input image. The pixels that contribute in the composition of final images are known as valid pixels and are indicated by 1 (or 0XFF); the remaining pixels are called invalid pixels and are indicated by 0. Invalid pixels are ignored.

In the following table, the Porter-Duff operators [**?**] are defined for images I_1 and I_2 while their corresponding masks are defined by M_1 and M_2.

Operation	Implementation I_r	Mask M_r
`clear`	0	0
`copy` I_1	I_1	M_1
I_1 `over` I_2	$(I_1 \wedge M_1) \vee (I_2 \wedge M_2 \wedge \neg M_1)$	$M_1 \vee M_2$
I_1 `in` I_2	I_1	$M_1 \wedge M_2$
I_1 `out` I_2	I_1	$M_1 \wedge \neg M_2$
I_1 `atop` I_2	$(I_1 \wedge M_1) \vee (I_2 \wedge \neg M_2)$	M_2
I_1 `xor` I_2	$(I_1 \wedge M_1 \wedge \neg M_2) \vee (I_2 \wedge \neg M_1 \wedge M_2)$	$(M_1 \wedge \neg M_2) \vee (\neg M_1 \wedge M_2)$

Your task is to define functions corresponding to each of the Porter-Duff operators and test the output using a driver program as described in the implementation steps.

Invoking the solution

Your solution will be invoked using the following command:

```
porter-duff [image1] [image2] [mask1] [mask2]
```

where all the parameters are optional. Feel free to use options to apply specific operators.

Suggested implementation steps

1. If there are no parameters supplied, create two images. Both images will be of size 640×480 pixels. The first image contains a circle in the center with radius 150 pixels in blue color. The second image contains a cross, created by two rectangles perpendicular to each other. The first rectangle is 512×96 pixels while the second rectangle is 128×384 pixels. Both rectangles are centered into the image and are colored red.
2. Apply each operator and display the result. Since most of the operators are non-commutative, show the results with *image1 op image2* as well as *image2 op image1*.
3. If there is no mask specified for the operators, create a mask using the non-zero pixels as valid pixels.

Criteria for Success

Each specified function is successfully implemented and demonstrated with at least the two synthetic images (circle and cross). Your code should ideally work with images supplied as command line parameters. A good way to demonstrate such code will be by using the camera and a green screen to capture a photo and demonstrate the operators in conjunction with the captured photo.

Grading

I'll use the following rubric to assess your submission.

1. *Overall submission; 30pts* Program compiles and upon reading, seems to be able to solve the assigned problem.
2. *Command line parsing; 10pts* Program is able to parse the command line appropriately, assigning defaults as needed; issues help if needed.
3. *Operators `clear` and `copy`; 5pts each* Program is able to demonstrate the operators `clear` and `copy`.
4. *Operators `over`, `in`, `out`, `atop`, and `xor`; 10pts each* Program correctly implements and demonstrates the operators `over`, `in`, `out`, `atop`, and `xor`.

Submission

Submit an electronic copy of all the sources, `README`, Makefile(s), and results. Create your programs in a directory called *username*.1 where *username* is your login name on delmar. This directory should be located in your `$HOME`. Once you are done with everything, *remove the executables and object files*, and issue the following commands:

```
% cd
% chmod 755 ~
% ~bhatias/bin/handin cs6420 1
% chmod 700 ~
```

Do not copy-paste these commands from the PDF; type in those commands.

References

[1] Thomas Porter and Tom Duff, "Compositing digital images," *SIGGRAPH Computer Graphics*, **18** (3) 253-259; July 1984.