Fall 2015	Computational Art	Zoë Wood

Lab 8 - Andy Warhol silkscreen type image filter

Goals

The goals for this lab are:

1. Practice creating an Andy Warhol silkscreen image filter
2. Practice using images in Processing
3. Practice manipulating pixels of an image
4. Practice using arrays
5. Practice writing for loops
6. Practice indexing a 1D ray with 2D coordinates
7. Practice using boolean logic to control which parts of an image are modified (including implicit circles)

Modality

Pair or Individual (per instructors specifications)

Overview

This lab will provide a chance similate the style of the pop artist Andy Warhol, who used various print making and silk screening techniques on images of famous people to create iconic pop art images.

From wikipedia:"Warhol's work both as a commercial artist and later a fine artist displays a casual approach to image making, in which chance plays a role and mistakes and unintentional marks are tolerated. The resulting imagery in both Warhol's commercial art and later in his fine art endeavors is often replete with imperfectionsmudges and smears can often be found. In

his book POPism Warhol writes, "When you do something exactly wrong, you always turn up something.""

Details

Tasks: For this lab, you will create your own computational filter to simulate his silkscreened images. In particular, we will practice some very simple changes to the values of pixels in an image to create a new altered image with strong colors and contrast.

Figure 1: Andy Warhol, Marilyn 1967 Silk Screen

To complete this assignment, please follow these steps

- Take a picture of yourself with a mostly blank background and transfer the pictures to the lab computers - make sure to resize the images to be small enough to then fit multiple copies on the image next to one another in your final sketch.
- Write a processing sketch to read in the image and load the images pixels
- Allocate as many other image(s) (at least 4), to store the altered image(s)

- Write code to loop through the whole image and modify the pixel values to acheive an Andy Warhol silk screen look. For example, try:
    1. start by making the background solid
    2. consider finding colors that stand out in the image and threshold them (ie make redish parts of the image all red, etc. for blue and green) - you can use red(), blue() to pull out the exact value of each olor component for each pixel
    3. for thresholding, you may need to confine your pixel filters to only certain regions of the image, i.e. only change the red pixels near the lips to bright red, etc. Use booleans to test the pixels x and y location to identify where in the image the pixel lays.
    4. consider thresholding the brightness to draw pure white, grey and black for any other pixels
- Produce a final image that includes several versions of your original image with various bright colorings similar to the silkscreen images produced by Andy Warhol.

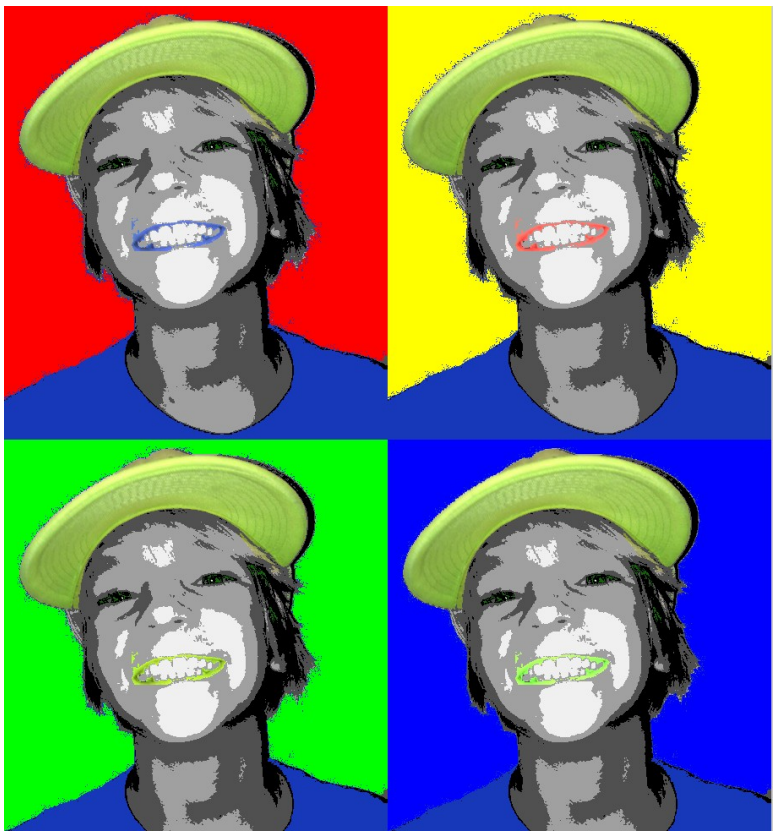

Figure 2: Example output from a Processing sketch to simulate an Andy Warhol silkscreen

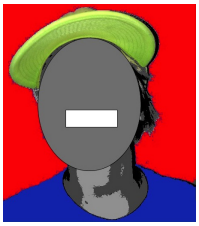

Figure 3: While developing your sketch, you may need to only apply a pixel filter to small parts of the image, for example, the above example only modifies the lips by only modifying pixels in a rectangular region shown in the above 'debugging' image - likewise since the hair and hat in the original image had such similar tones, an implicit ellipse was used to distinguish between the two, also shown above.

Demo:

In order to receive credit for this lab, you must demo your sketch to your instructor or TA. For every lab, your score will be broken down 75% for meeting the technical requirements and 25% for aesthetics.

Submitting your sketch

You must post an image of your sketch to your pinterest Computational Art board. Also pin the frame from Tant de Forets that you used as your reference images to your pinterest board as well.

Resources:

some useful commands:

```
PImage img1;
loadImage
createImage
loadPixels
updatePixels
brightness
image();
red()
blue()
green()
brightness()
```

Here is a sample program that loads in an image and manipulates every other column of pixels to give an example of using 2D counting converted to 1D array indexing:

```
//Example of some image manipulation operations 2D -> 1D indexing
//draws white stripes every other pixel 'column'
//ZJ Wood
PImage img1, img2;

void setup() {
  int loc;

  noLoop();
  /*you will need to have the image in the same directory - you can rename */
  /* just set your 'size' to match the image size */
  img1 = loadImage("beach_diggers.jpg");
  img2 = createImage(img1.width, img1.height, RGB);
  size(600, 320);

  img1.loadPixels();
  img2.loadPixels();

  /* 2D loop to show how to loop in 2D into a 1D array */
  for( int y=0; y < img1.height; y++) {
    for (int x=0; x < img1.width; x++) {
      /*map from 2D to 1D */
      loc = img2.width*y + x;
      /* for every other x, copy in the pixel to make stripes */
      if (x%2 ==0) {
        img2.pixels[loc] = img1.pixels[loc];
      }else {
        img2.pixels[loc] = color(255);
      }
    }
  }
}

void draw() {
  image(img2, 0, 0);
}
```