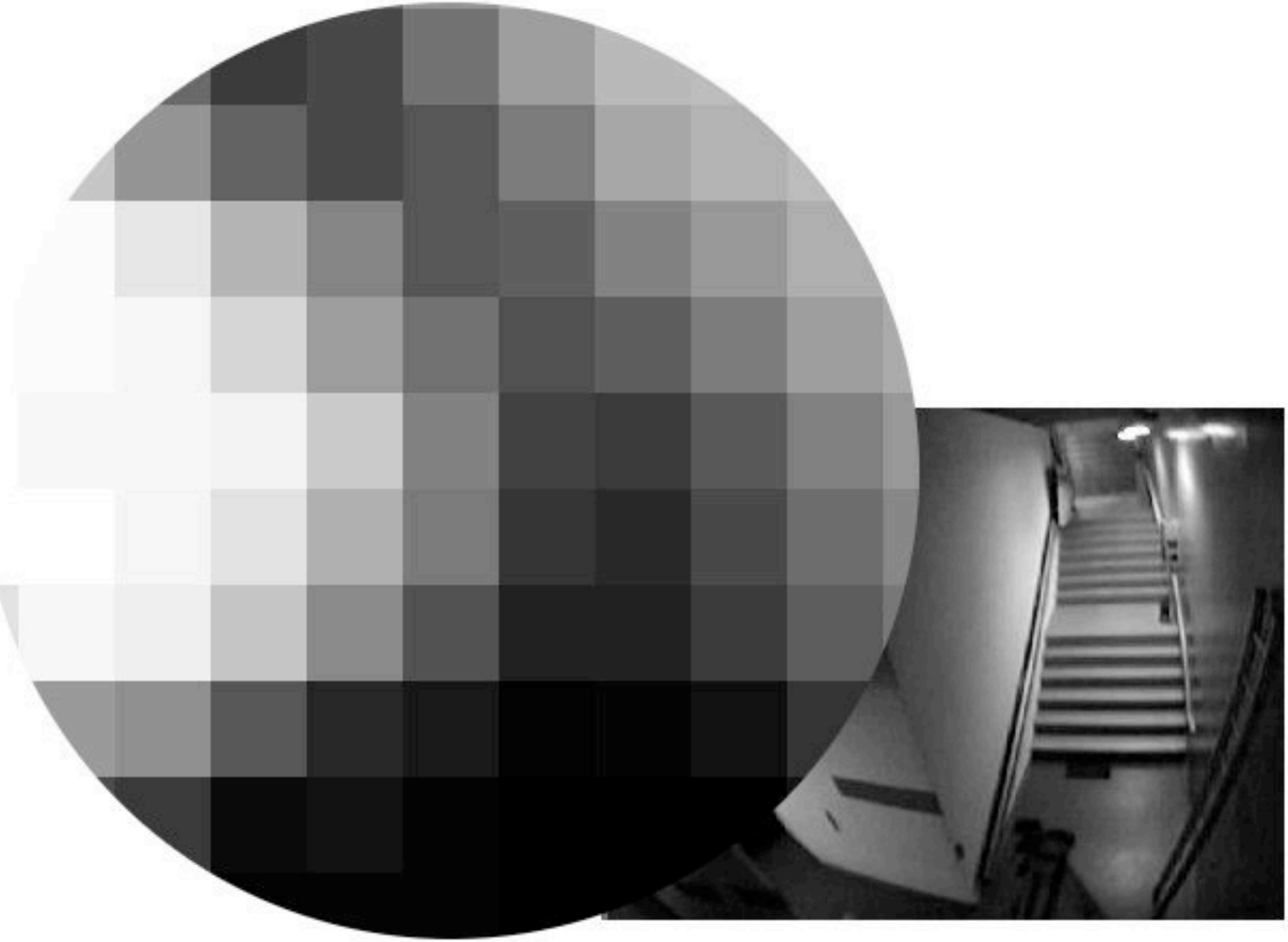
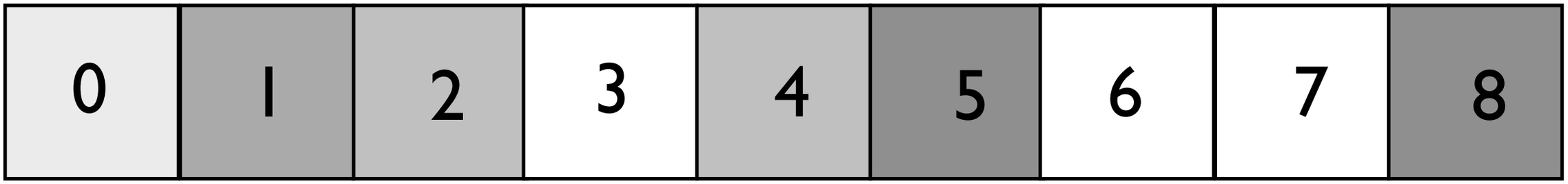
The only people who see the whole picture are the ones who step out of the frame…

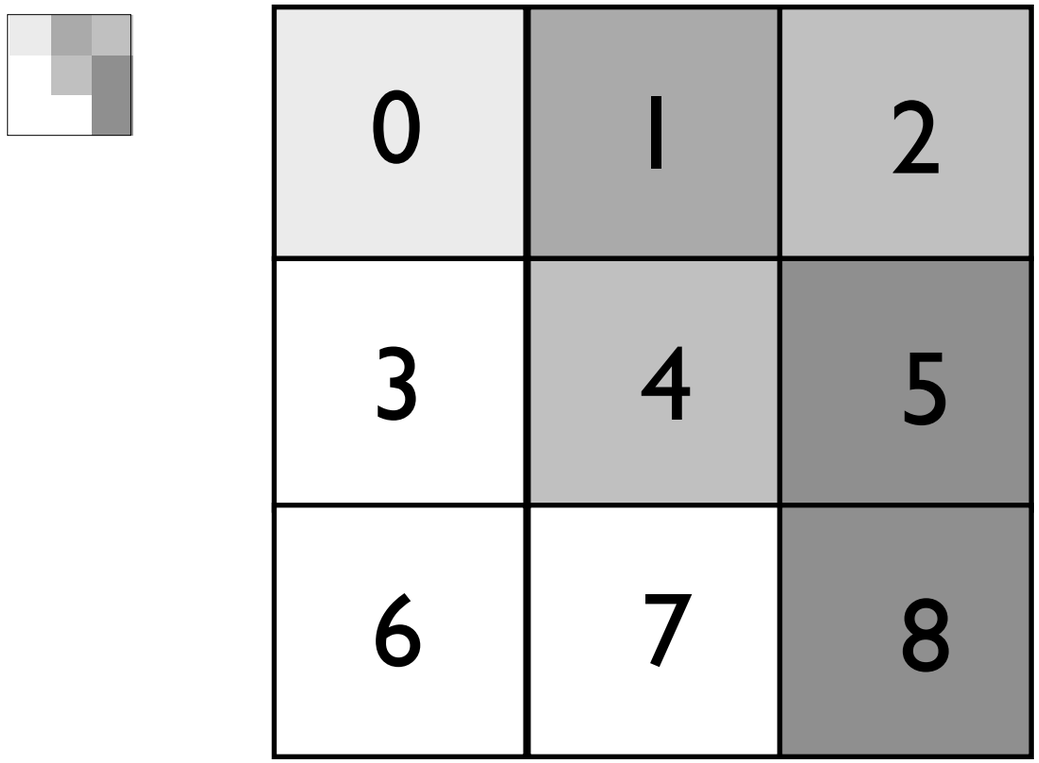
-Atley

OpenCV with Processing



Pixels are stored in 1 dimension although we view them in 2 dimensions..





PImage

Processing class for storing images.

Fields:

`width`	image width
`height`	image height
`pixels[ ]`	Array of the colors of all the pixels in the image

PImage_basic

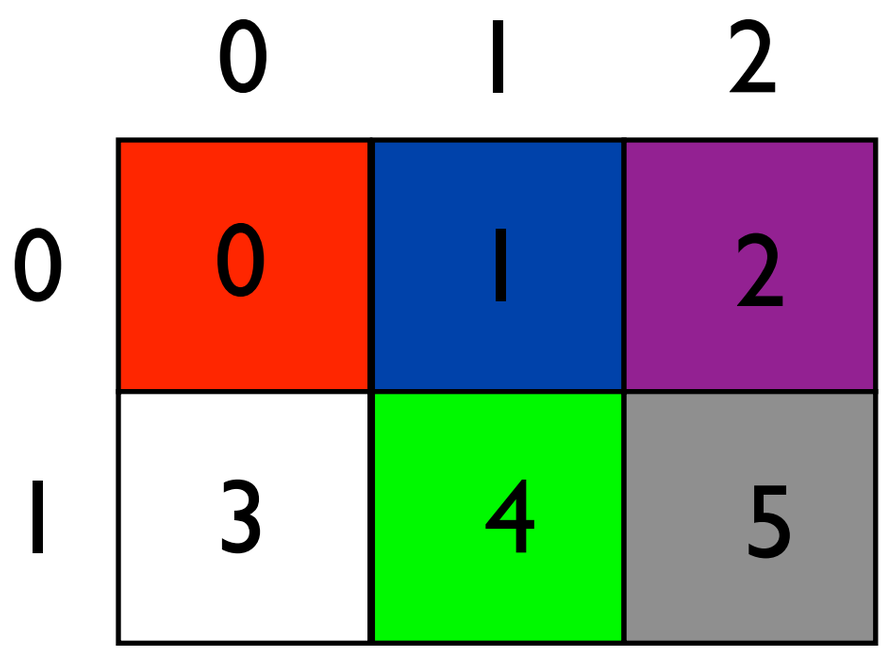
```
PImage myImage;
myImage = loadImage("technical-difficulties.jpg");

size(500,443);

image(myImage,0,0);
```

Images are represented digitally as a series of values corresponding to the color of each pixel.

In Processing, the ***PImage*** class has an array called pixels that stores this pixel information. Each value in the array is of the Processing ***color*** type and has a red, green and blue channel.

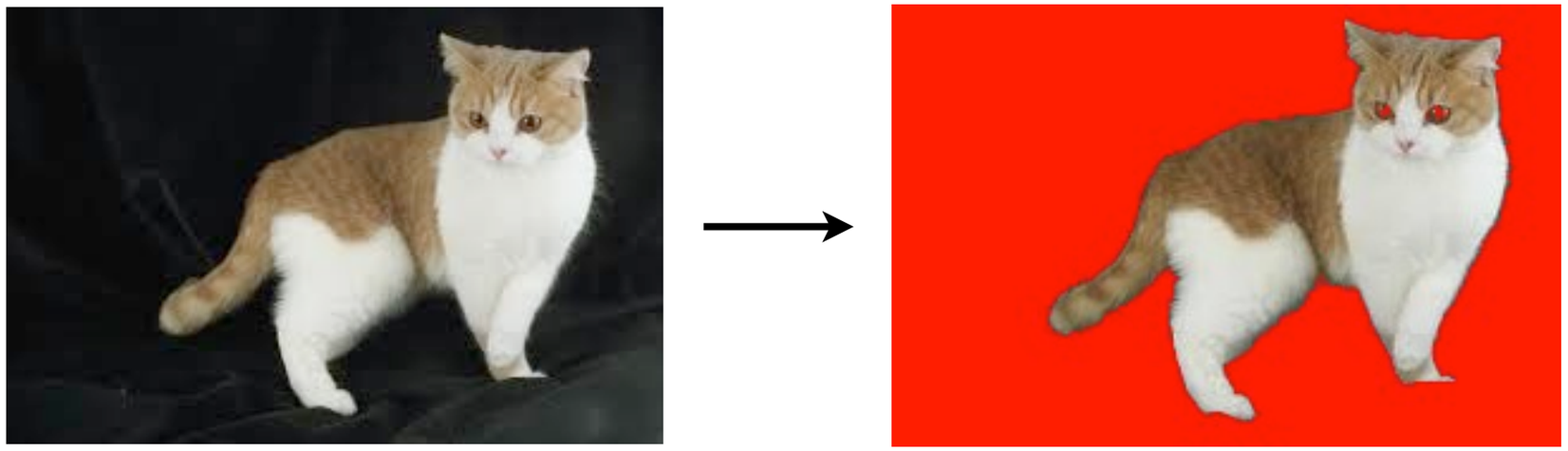



```
img.loadPixels();
img.pixels[0] = color(255,0,0);
img.pixels[1] = color(0,0,255);
img.pixels[3] = color(255,255,255);
img.updatePixels();
```

Iterating through image pixels in 1 dimension

```
int totalPixels = myImage.width*myImage.height;

for( int i = 0; i <  totalPixels; i++){
  color pixelColor = myImage.pixels[i];
}
```



1. Find an image with a black background

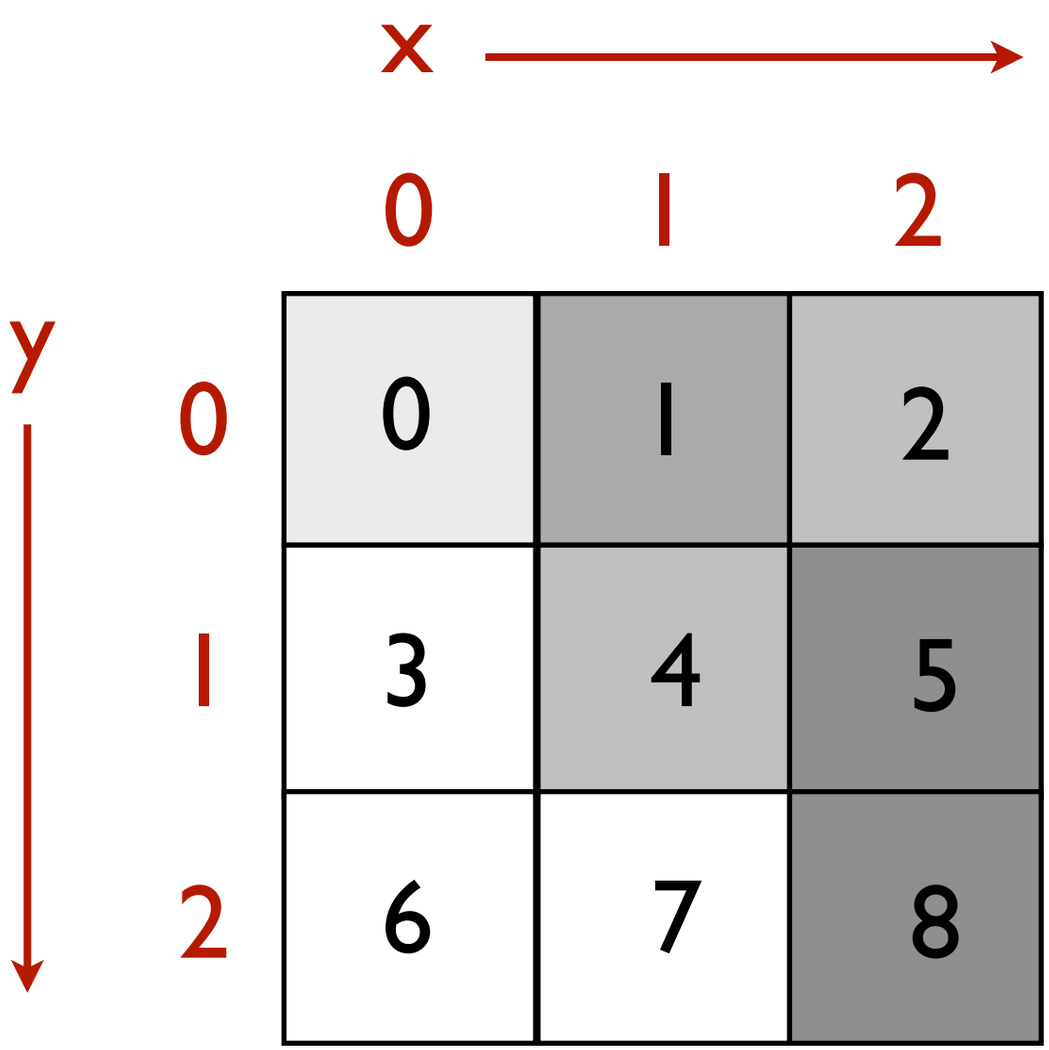
(google “cat on black background” for example)

2. Create 2 PImage objects - image A should load the image you found and image B should be empty but the same size as image A

3. Load image B’s pixels (imageB.loadPixels()). Loop through all pixels of image A and get the pixel color. Using the color, get the brightness of that pixel. Test if the brightness is less than some threshold value. If it is, set the pixel in the image B to red and if not set it to the image A’s color.

4. Update the pixels in image B (imageB.updatePixels()) and draw it

index = y * width + x



The image width here is 3 pixels.

If we want the index of the pixel when is x = 1 and y = 1:

index = (y * width) + x
(1 * 3) + 1 // equal to 4

color myPixelColor = image.pixels[4];

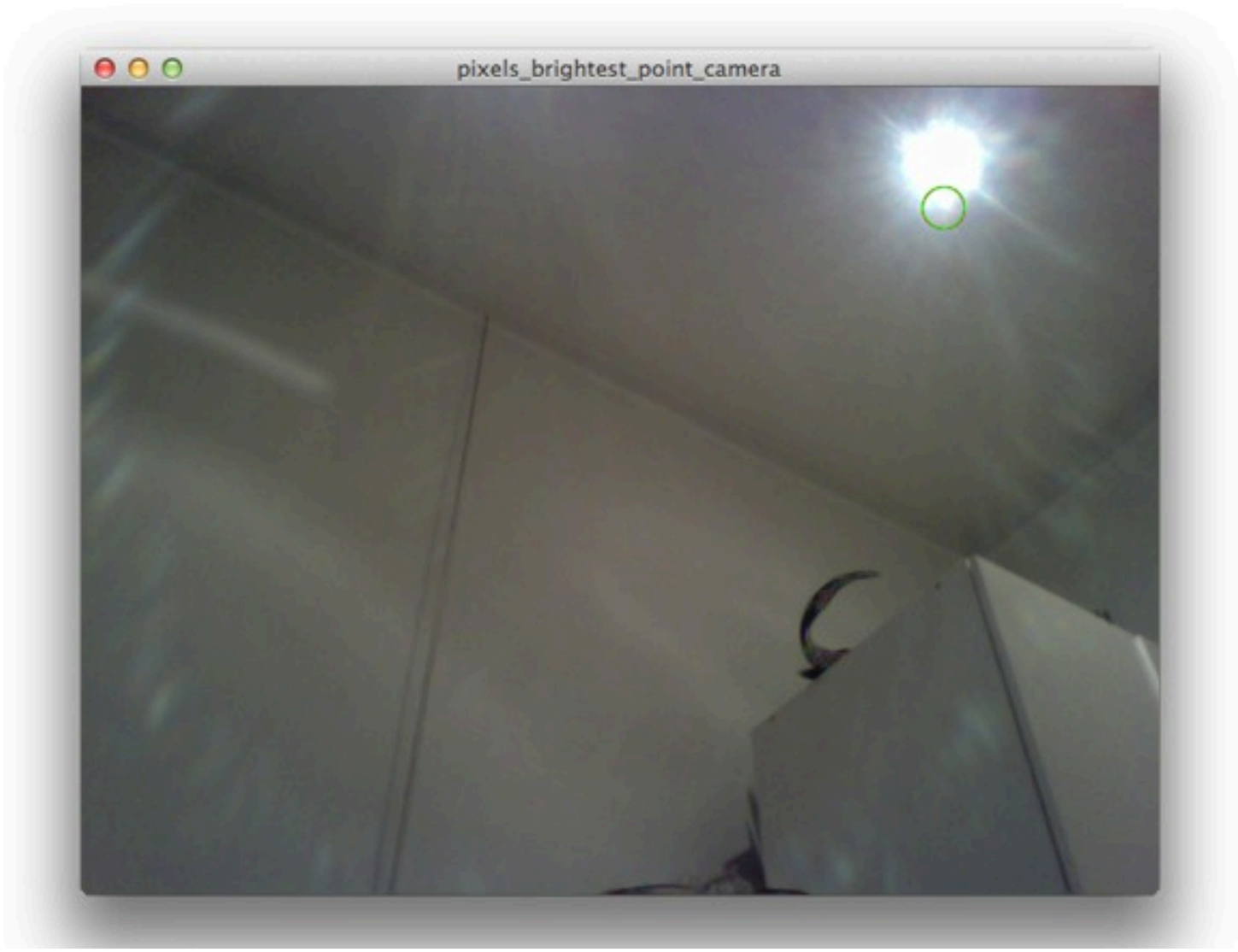
Iterating through image pixels in 2 dimensions

```
for( int y = 0; y <  myImage.height; y++){

  for( int x = 0; x <  myImage.width; x++){
      int pix = y * myImage.width + x;
      color pixelColor = myImage.pixels[pix];
    }

}
```

Video capture and brightness tracking



Open Source Computer Vision Library

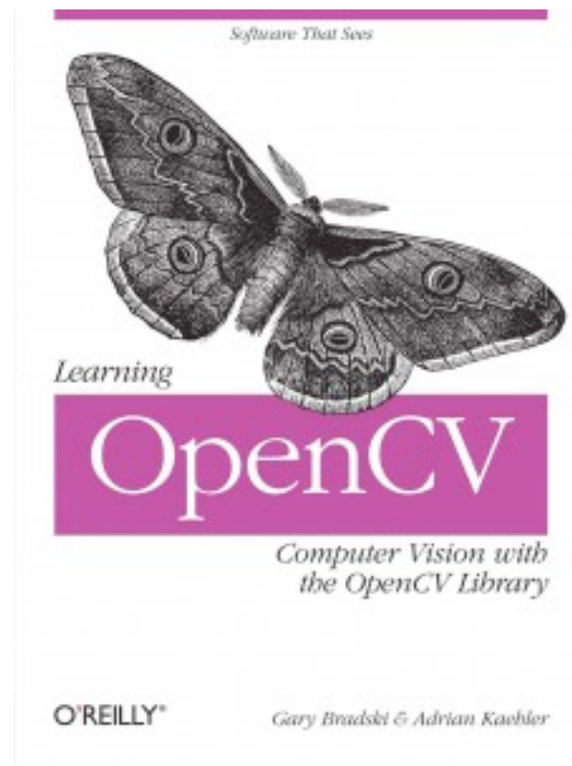


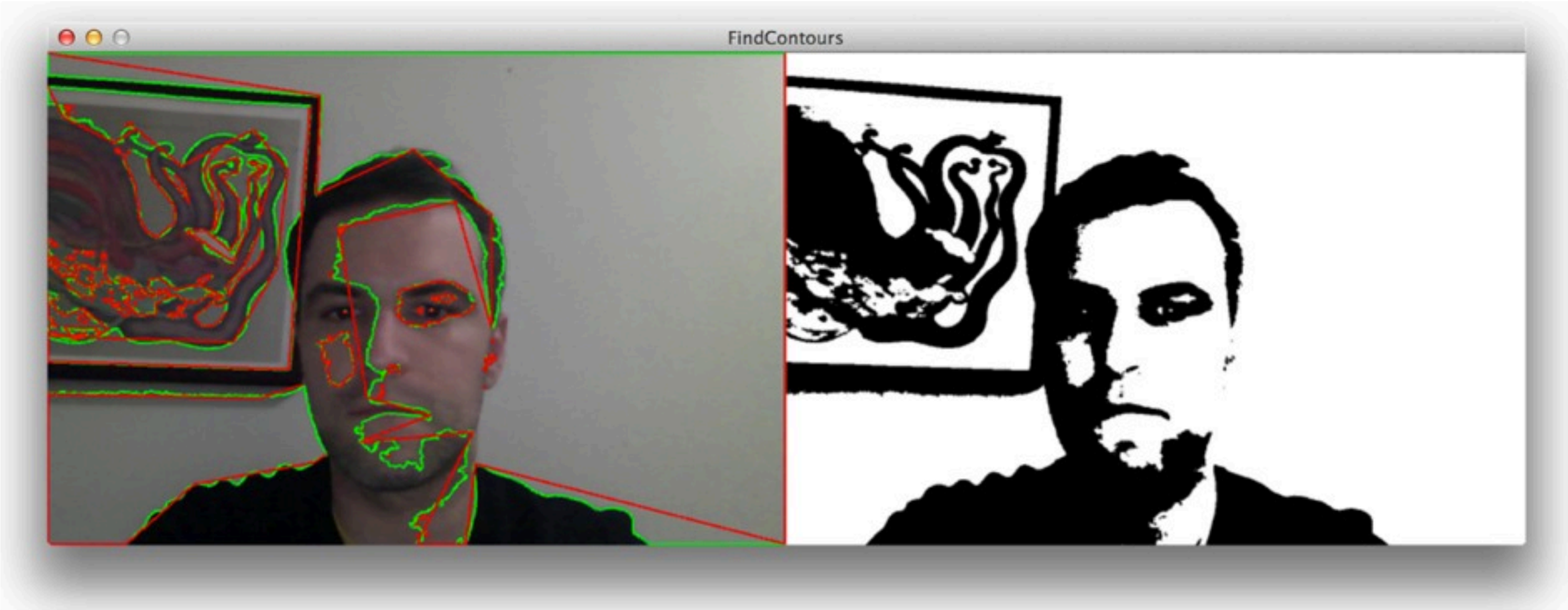
Image Differencing

Get the difference between frames and test if that value is greater than some threshold (50 for example)

abs(10 - 120) = 110 110 > 50 ?

abs(105 - 130) = 25 25 > 50

Contours



Other CV functionality: