

# Tutorial 1 - RW364

Ieuau, Uys  
18218067@sun.ac.za

August 7, 2017

## 1 Greenscreen

The image was first separated with a vectorized calculation of boolean arrays. The original image is split up into the color channels and then the values are compared to threshold values that were found to best remove the green screen background.



Figure 1: Green-screen.png



Figure 2: Thresh-old.png



Figure 3: NewImage.png

## 2 Unsharp Mask

The method of the unsharp mask is a simple one.

I began with a simple greyscale image.

See the entire transformation (Figure 4 - 5).

Between steps it is necessary to blur and take the difference of the blur and the initial image.

See Figure 6 & 7.

To compare my sharpened image I used the openCV library to make a second unsharped mask and this is what you see compared below.

### 2.1 Overflow

As you can clearly see, there are many pixels with rgb values outside of the range of [0, 255] which results in bogus image data. This is caused by the



Figure 4: Image before any effects

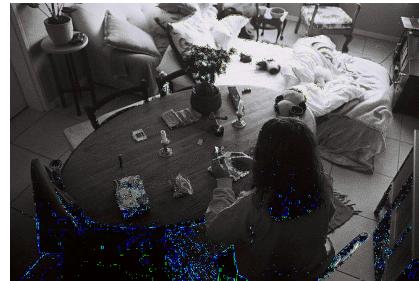


Figure 5: Final image after effects



Figure 6: Apply Gaussian blur



Figure 7: difference of (4) and (6)

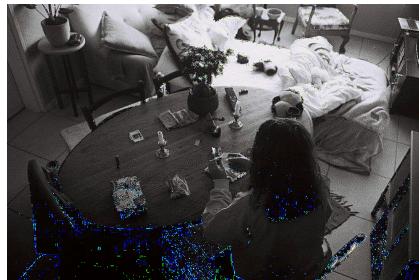


Figure 8: My Unsharp Mask



Figure 9: openCV library

mathematics behind the method of unsharp masking and is a normal result. I have also added a unsharp mask implemented by the openCV library which has made provision for the case. I believe that it would be better to scale all the intensities to set the highest overflow value to 255 and the lowest underflow values to 0 so as to maintain the contrast.

### 3 Image Scaling

Resizing of images is made simple through inverse transformation, here are examples of my Nearest Neighbour and Bilinear implementations.

#### 3.1 Nearest Neighbour

The nearest neighbour makes use of very simple rounding to interpolate the pixels in the new rescaled image. See figure 11 & 12.



Figure 10: The Original Image



Figure 11: NN with scale 0.5

#### 3.2 Bilinear Interpolation

The bilinear interpolation makes use of 4 samples when doing the inverse transformation to find the values of the color tuples in the new resized image.



Figure 12: BI with scale 2



Figure 13: NN with scale 2

#### 3.3 Comparison

When looking at Figure 14 and 15, which are zoomed versions of figure 12 & 13, it is clear to see that the bilinear interpolation achieves much better results than the nearest neighbour interpolation. In Figure 14 the edges are smoother and the eyelashes in the image are visible as opposed to figure 15 where the eyelashes are a large mass of pixels.



Figure 14: BI with scale 2



Figure 15: NN with scale 2

## 4 Feature Detection

I incurred errors in the section and was not able to implement Feature Detection nor was I able to complete Question 5

## 5 Libraries

- matplotlib
- skimage
- numpy
- scipy
- math
- cv2