Image Processing – A1, part 2

# Improving Image Contrast

The topic I chose for part two of this assignment, was contrast enhancement. The three algorithms I chose to use where: morph\_CE, morph\_toggleCE, and histhyper. I also created a simple algorithm through some experimentation which I called drew\_CE.

I performed a little bit of a twist on this part of the assignment, as instead of performing the contrast enhancements on grey scale images, I decided to apply these algorithms to colour images. To apply these algorithms to colour images required converting them from RGB to HSV, and then applying the algorithm to the H component of the image. Next the enhanced image are converted back to RGB so it can be saved.

# Discussing the Algorithms

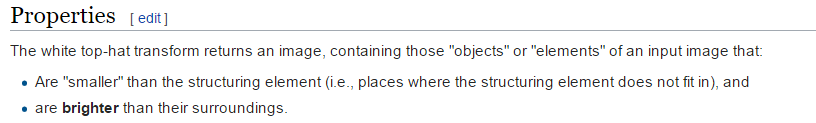
Each of the algorithms I used will now be gone through.

## morph\_CE

The morph algorithm creates two additional versions of the original image, and then merges these new images directly with the original image using simple addition and subtraction.

The first of the two images is created by using an opening top hat algorithm applied to the original image, and the second of the two new images is created by applying a closing top hap function to the image.

What do the top hat functions do? A white top hat transformation takes the surrounding pixels, and makes the current pixel brighter than those surrounding pixels.



Closing Top hat does the opposite by creating an image with “elements” that are darker than their surroundings. So the openth and closeth functions are creating a darker and lighter version of the image, using a structuring element to determine pixels taken into consideration to determine what happens to the current pixel being modified.

|  |  |  |
| --- | --- | --- |
| Opening Top Hat White top-hat has the effect of making light colours surrounded by light colours even lighter without big impact on darks | [Openth Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUxpPrxN-oJvq-YRQ) | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\lowcontrast1.jpg  [Original Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUNuid1jzYaSWC2bg) |
| Closing Top Hat black top-hat has the effect of making dark colours surrounded by dark colours even darker without big impact on lights. | [Closeth Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUyVSR92kLl43VKdg) | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\lowcontrast1.jpg  [Original Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUNuid1jzYaSWC2bg) |
| Morph\_CE By combing the original with the darkened, and lightened images, the darks become darker, and the lights become lighter improving the images contrast. | [Morph\_CE](https://1drv.ms/i/s!As-TTArLWDqlhoUzgeg-D9Fgko1R_g) | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\lowcontrast1.jpg  [Original Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUNuid1jzYaSWC2bg) |

Darkening the dark parts of the image, lightening the light parts of the image, and then combining the results seems like a good idea, however the result on my demo makes the eyes look eerie. In fact for the above demo image, I think that the closing top hat alone applied looks better than the full morph\_CE variant.

## morph\_toggleCE

There morph toggle mixes two parts: an eroded version of the image, and a dilated version of the image.

The gist of the morph toggle algorithm is to create a disk structuring element that determines what pixels are considered when determining the effect on the current pixel. Using the disk structuring element, two versions of the original image are created. First: an eroded image, and second a dilated image. Then for each pixel, the difference between the original pixel, and the eroded/dilated pixel is calculated. The pixel with the smaller difference from the original pixel is chosen.

|  |  |  |
| --- | --- | --- |
| Erode Erode, as its name intuitively suggests, erodes the image. This darkens the dark areas, even the dark areas that border on white areas interestingly. It also makes the image appear “fuzzy”. | [Eroded Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUWMhVH3R9pL6pBZg) | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\lowcontrast1.jpg  [Original Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUNuid1jzYaSWC2bg) |
| Dilate Dilate does opposite of erode, meaning dilate lightens the image. Its seems to fairly aggressively eat away at dark areas surrounded by very light areas. | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\contrast1 - dilate.jpg  [Dilated Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUVKvjCIsWf-innEQ) | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\contrast1.jpg  [Original Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUNuid1jzYaSWC2bg) |

From the above examples, it is easy to see that neither erode nor dilate gets us closer to improving the original images contrast in a good-looking manner. So somehow, morph toggle blends these two images to get much better results. The way morph toggle blenders the two images is by comparing the pixels from the two images against the original, and choosing the pixels that are closest or “most like” the original image pixel.

By taking the pixel closest to the original images pixel, some of the darkening benefits of erode can be gained, while having that darkening rereigned in by the dilated version in the lighter areas of the image. A battle and balance between dark and light if you will.

|  |  |  |
| --- | --- | --- |
| C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\lowcontrast1 - less like original.jpg  [Pixel most different from original](https://1drv.ms/i/s!As-TTArLWDqlhoUYLXnmDuSz-2QDUw) | [Original Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUNuid1jzYaSWC2bg) | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\lowcontrast1 - morph_toggleCE.JPG  [Pixel least different from original](https://1drv.ms/i/s!As-TTArLWDqlhoUX4vtiRKNh-GEZBg)  (AKA. toggle\_morphCE) |

## Histhyper

To be honest, I get the general idea of this algorithm, but I do not understand the math that goes into calculating the transformation factor.

|  |  |
| --- | --- |
| Image result for i'm afraid we'll have to use math | *“I’m afraid we’ll have to use…*  *MATH”* |

|  |  |
| --- | --- |
| 1. The image is split into “bins” by the histogram function. Each bin condenses the y range of the image to a single histogram value, for each given subsection of the image. 2. Each histogram value / bin value is gone through and used to calculate a transformation value for each shade of grey. 3. Each pixel is gone through, and has the adjustment / transformation value applied to it. |  |

|  |  |  |
| --- | --- | --- |
| HistHyper When testing HistHyper did have very good contrast enhancement results, but it seemed to cause a decrease in quality, in some images, this trade off is worth it.  This image is an extreme example, being mostly white. In this extreme example, the contrast has been massively improved, but the quality of the image is much worse. | [HistHyper Link](http://i.imgur.com/rrs7vhT.jpg) | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\contrast1.jpg  [Original Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUNuid1jzYaSWC2bg) |

## drew\_CE

This algorithm is very simple. Go through each pixel, and if the value is greater than 255/2 or 127.5, then increase it, and if the pixel value is less than 127.5 decrease it. The idea being to make darks darker, and lights lighter.

How much does this algorithm adjust the pixel value by? Half of the difference between the current pixel value, and 127.5.

This algorithm is a result of me spending several hours trying to come up with a contrast improvement algorithm without having background knowledge, or any idea what I was doing. This was one of the first methods I tried, and it was pretty much the only method that worked.

The advantages of this algorithm are that: it is very simple, and it does slightly improve contrast of grey images, while basically not affecting quality of the image at all.

The disadvantages of this algorithm is that it is only applicable to images that are mostly grey. The results of applying this image algorithm to the lady’s face was like histhyper however, which makes me feel a little better about my algorithm being terrible.

|  |  |  |
| --- | --- | --- |
| Drew\_CE Does have useful results on images that are very grey. For images that are almost all darker than 127.5, or lighter than 127.5, they won’t have their contrast enhanced, in fact, these images will just get worse.    I tried variations where it would take the mean into account, but I could not get it to work well. | [Drew\_CE link](http://i.imgur.com/pWpTYqG.jpg) | C:\Users\andre\AppData\Local\Microsoft\Windows\INetCacheContent.Word\contrast1.jpg  [Original Image Link](https://1drv.ms/i/s!As-TTArLWDqlhoUNuid1jzYaSWC2bg) |

# Experiments Performed

The experiments I performed were very straight forward.

<EXPLAIN HERE>

# Quantitative Analysis of Algorithms

# Notable Computational Differences Between Algorithms

# Difficulties Processing Images within the Context of the Topic Choosen