Image Colorization

Ramin Assadi

November 19, 2014

Abstract

This project describes how it is possible to colorize an image. It will be shown how two input images can be used to colorize a grayscale image. The main focus in this project is to show how the theory behind it works. There will also be an example that shows a simple implementation of the theory in action.

1 Introduction

In this project there will be four different abbreviations used to indicate what image is focused.

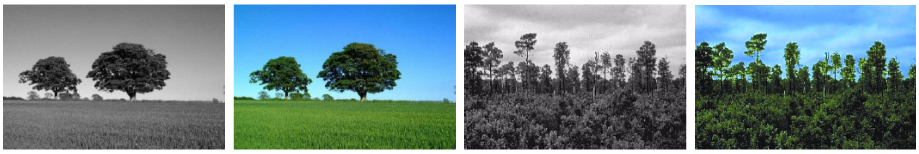

Figure 1: From left to right - A A' B B'

The second image A' will represent the colored input image that will be picked out by the user. The first image is the grayscale image of A'. The third image B is the image that is going to be colorized. The last image B' is the result image and what B will end up looking. The user will be able to choose what input images A and A' that will be used. The input target image B will aslo be picked by the user. The goal is to determine what color from A' will be the best match for each pixel in B. By determining what color that corresponds with each pixel in B will result in B'.

2 Experimental Procedure

The first step is to find the image B. This is the image that is going to be colorized. The next step is to find the input image A'. It is advised to use a image that is similar to the image B. To get the grayscale image A the luminance values for A' will be retrieved and inserted into a matrix. This matrix is our grayscale image A. The next step is to calculate the new image B' that is the colored version of B. The first step is to make a neighborhood for each pixel in A and B. A neighborhood is a block with pixels with the selected pixel in the middle. Now with the help of the neighborhoods, each neighborhood in B needs to find the best matching neighborhood in A. One way of finding the best matching neighborhood is by comparing each neighborhood in B with all of the neighborhoods in A. Another way of finding the best matching neighborhood is by doing the same thing as before, but this time it should be able to use the neighborhoods that is already synthesised in B'. The methods are called "best approximate match" and "best coherence match" respectively. Although the best approximate matches are numerically better matches, the coherent matches may look better to a human viewer because it helps with the consistency of the image. Also the best approximate matches needs to check each neighborhood with all other neighborhoods and this will take an unreasonable amount of time. When the neighborhood in B finds the best neighborhood in A, then the next step is to get the position of the pixel in that neighborhood and map that position in the image A'. The pixel in that position will have the information about the RGB values. The last step is to take the RGB value and adding it to the new pixel in B'. Doing this for all the neighborhoods in B will end up with the new colored image B'.

3 Results

The objective was to try if it was possible to color a grayscale image of myself by using the input image of an female.

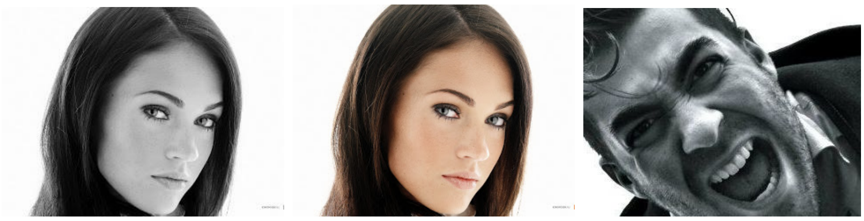

Figure 2: From left to right - A A' B

In this project the best approximate match was used. After calculating the neighborhoods and the best approximate match the RGB values were saved in the new image B'.

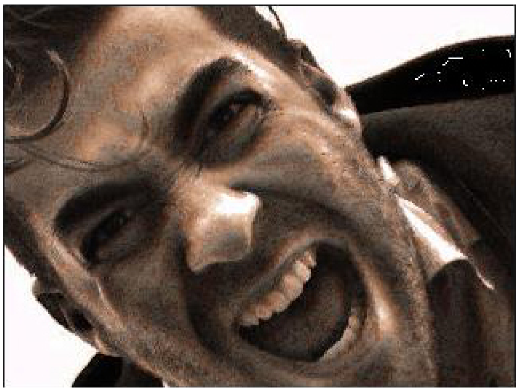

Figure 3: Image B'

We can see that B' is colored with few colors, that is because the colors that is in use is from image A'. There was also a test between the two different matching methods.

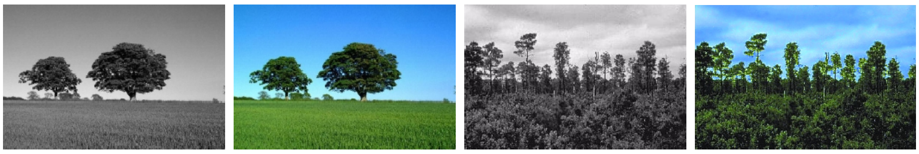

Figure 4: With the "best coherence match"

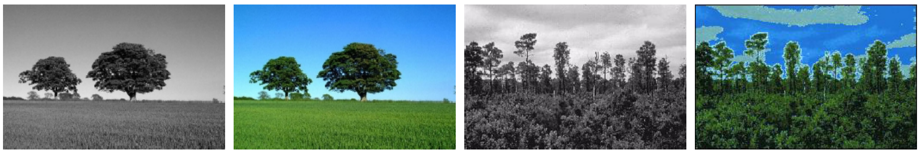

Figure 5: With the "best approximate match"

We can see that the best coherence match gives a better visual result. Although the best approximate match is a more precise result. The results may look good when using good images but the next results will be shown how it can give bad results while doing this method.

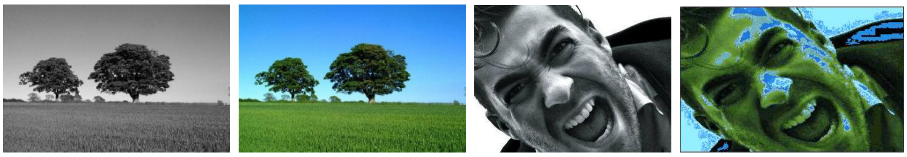

Figure 6: Test with using a bad B image

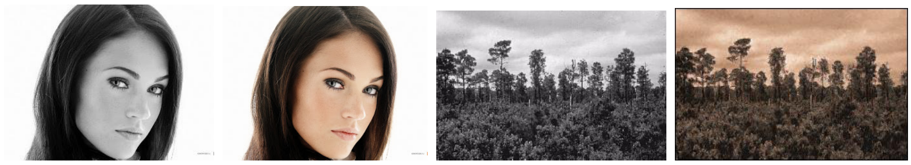

Figure 7: Another test with using a bad B image

For the figure 6 and 7, the B image has been swaped out with eachother. We can see that the method gives us a result but not a good one.

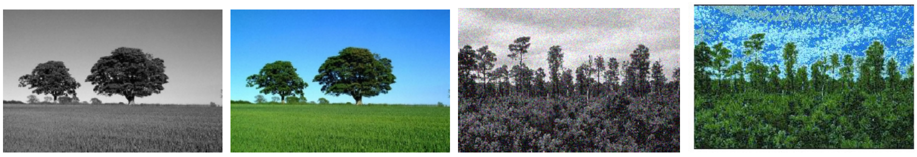

Figure 8: Image Colorization with noise in image B

We can see in figure 8 that the image B has noise in it. The resulting image B' is also really bad compared to figure 4 and 5.

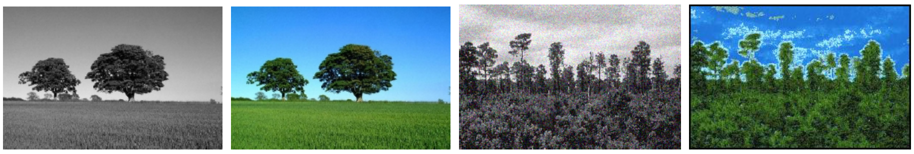

Figure 9: Image Colorization with noise in image B - 5x5 neighborhood block

This is the result when using 5x5 neighborhood blocks. We can see that the result becomes better.

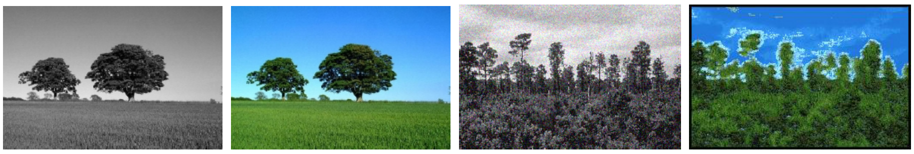

Figure 10: Image Colorization with noise in image B - 7x7 neighborhood block

This is the result when using 7x7 neighborhood blocks. We can see that the result becomes a little better then the 5x5 blocks.

4 Discussion

There is a couple aspects that needs to be considered when doing image colorization with this method. The first thing and the most important is the decision of images. If we look at the result at Figure 3 we can see that the result isn't that good. This is because of the input image A'. To get a better result it is advised to use images that are similar. The selection of what to color the image B' with is from the image A'. A more suitable image would be a man with the similar "face shadows" and clothes. Another thing to consider is what that is needed for the colorization. Depending on what matching method that is used, will not only save or lose time but it will also effect the result. For example the difference between best coherence match and best approximate match is that the result from coherence match is better for when someone wants the image to look good. But somebody else maybe wants a precise and accurate image. But that will cost more time to calculate. Depending on what result and performance is desired it is important to first find out what method is the best. The last thing to consider is when it comes to comparing the images. In this project the grayscale images A and B consists of the luminance values. When it comes to comparing luminance values it can occur that different RGB values have the same luminance value. This is because luminance is how bright the pixel will appear on the surface. So two different colors can have the same luminance and therefore the program can choose the wrong color. This is why it is considered to choose an input image A' that does not consist of many colors. At figure 6 and 7 it can be seen how the wrong matching of images can give a bad colorization. We can see that the method works but it uses the wrong colors because the only colors to colorize with comes from the A' image. Now there is some other problem with this method as well. Figure 8 shows how noise can effect the result. The image B' in figure 8 has been colorized but we can see that the upper part of the image has alot of wrong values in it. This is caused by the noise from image B. Why this happens is because of how the neighborhoods are setup. In this example a 3x3 block is used for each pixels neighborhood. In each neighborhood there is 9 pixels and if we get noise some neighborhoods will have spiked pixel values and that will give the neighborhood the wrong median value. To fix this problem we can make the neighborhood blocks bigger. How this will help is because the bigger the block is the less importance will the noise have. We compare the neighborhoods with each others, and the more values to compare with, the more will the noise be "ignored". If we look at Figure 9 we can see how the same image will look with 5x5 neighborhood blocks. We can see that the resulting image becomes much better. In figure 10 the result becomes even better with 7x7 blocks. But this does not mean that bigger blocks always give a better result. By doing the blocks bigger it will make the system slower because now there is a lot more pixels to calculate. But if the blocks become to big the result will be worse. This is because the blocks will contain information of more artifacts in the image then it should. When increasing the block size the neighborhood median value will be more equal to each other. This will make the matching of neighborhoods almost totally random.

5 Conclusion

By examining the images the result indicates that our methods and calculations are correct. Although there is a lot of aspects to take in hand. By using the right methods and images the result of coloring an image can be achieved, depending on what is desired.

References

[1] *Aaron hertzmann, Charles E. Jacobs, Nuria Oliver, Brian Curless, David H. Salesin*, Proceedings of the 28th annual conference on Computer graphics and interactive techniques

[2] *Bryce Aebi*, CS129 Final Project: Image Colorization with Image Analogies