TELECOM PARIS

IMA PROJECT
IMA 201

Automatic Color Equalization

Group :
Mayssa HADDAR
Oussama ZAIBI

Supervisor :
Christophe KERVAZO

Contents

1 Introduction

For various reasons, images captured by cameras could show a distorted version of reality. Poor lighting conditions and illuminant variations are usually the reasons behind a distorted color representation in an image. These kinds of variations are usually surmounted by the human visual system. Human vision is able to achieve color constancy by adjusting perception according to the different conditions.

In order to achieve this kind of color constancy for digital images, ACE also known as "Automatic Color Equalization" has been conceived. Inspired by various low-level mechanisms of the human vision, ACE automatically enhances colors in images without needing access to the non distorted version of the scene, making it an unsupervised algorithm.

ACE takes into account the color spatial distribution in the image, which makes it efficient for both local and global color correction. The algorithm is based on two global equalization mechanisms:

- Gray world : This condition supposes that the image could be represented with a gray mean value.
- White patch : This condition supposes the presence of a reference white area in the image.

Taking into account the color spatial distribution in the image, and using the Gray World and White Patch global equalization mechanisms, ACE affects a chromatic and spatial adjustment to the image followed by a dynamic tone reproduction scaling in order to get information from the image regardless of lighting conditions.

Because of its high computation cost, two approximations of ACE are presented, the first is based on polynomial approximations and convolutions, the second is based on interpolating intensity levels.

2 Presenting the algorithm

2.1 Verifying White patch gray world conditions

Gray World is an assumption telling that our image, on average, is a neutral gray. In other words, the mean of pixels has a value around a neutral gray one which is equal to 127 for pixels ranging from 0 to 255.

For RGB image the mean is computed separately for each channel. This hypothesis holds when we have a good color-distribution in the image's scene.

White Patch is used of the Color Constancy adaptation. It searches for the lightest pixel to consider it as a white reference, an hypothesis inspired by the human eye behavior.

Thus, each channel for our RGB color space should reach its max, i.e the max value taken by its pixel is 255.

These two hypothesises were verified before using the images we chose for testing our algorithm.

2.2 Implementing different r(.) functions

r(.) takes part in computing the influence of each pixel in the input image over each output pixel. In order to respect our two core hypothesis (White Patch and Gray World) our r(.) function is submitted to certain constraints:

- r(.) is odd : Assert Gray World Behavior
- Non-linearity of r(.): Assert White Patch Behavior

We need to mention that the non-linearity of r(.) is assured by its saturation. The different r(.) function used are the following:

- Linear : $r(x) = x$
- signum : $r(x) = sign(x)$
- tanh: $r(x) = tanh(x)$
- saturation: $r(x) = min(1.0, max(x/a, -1.0))$

To select the most convenient r function we will test every one of them on a single image with a fixed distance function and discuss the results.

2.3 Implementing different distances

In order to implement ACE, we need to take into account the color spatial distribution in the image. To do so, we need to compute the distance between each pixel and other pixels in the subset. For that, we define the different distances mentioned in the paper:

- Euclidean : $\sqrt{(dx^2 + dy^2)}$
- Inverse Exponential : $\frac{1}{exp(-\alpha Ed)}$; where Ed is the Euclidean distance
- Manhattan : $dx + dy$
- Maximum : $Max(dx, dy)$

To choose which distance works best, we test the algorithm with different distances and a fixed r(.) function. This will be discussed in the result analysis section.

2.4 Chromatic / spatial Adjustment

In this step, we recompute each pixel of separate channels according to the values of pixels of the original image. We obtain an intermediate image R. This operation is done according to the following equation :

$$R_c(p) = \frac{\sum_{j \in \text{Subset}, j \neq p} \frac{r(I_c(p) - I_c(j))}{d(p,j)}}{\sum_{j \in \text{Subset}, j \neq p} \frac{r_{\max}}{d(p,j)}}$$

Where:

- c is the channel
- p is the pixel we're working on
- d is the chosen distance
- The subset is either the whole image (first phase) or a subset of pixels centered around the pixel p (second phase)

2.5 Dynamic tone reproduction scaling

This step is useful for mapping pixels from the intermediate image R into the output image O. It could either be done with a simple dynamic maximization or by taking into account gray world and white patch conditions and mapping relative lightness appearance values of each channel [2].

2.5.1 Linear scaling

This method linearly scales values of each pixel of R in each channel independently of other pixels according to the following equation:

$$Oc(p) = round(sc(Rc(p) - mc))$$

where:

- sc is the slope of the segment [(mc,0),(Mc,255)]
- mc is the minimum pixel value in the channel c
- Mc is the maximum pixel value in the channel c

2.5.2 White Patch Gray World scaling

This method considers Mc as a white reference and the zero in Rc as the medium gray reference, following the equation below:

$$Oc(p) = round[127.5 + scRc(p)]$$

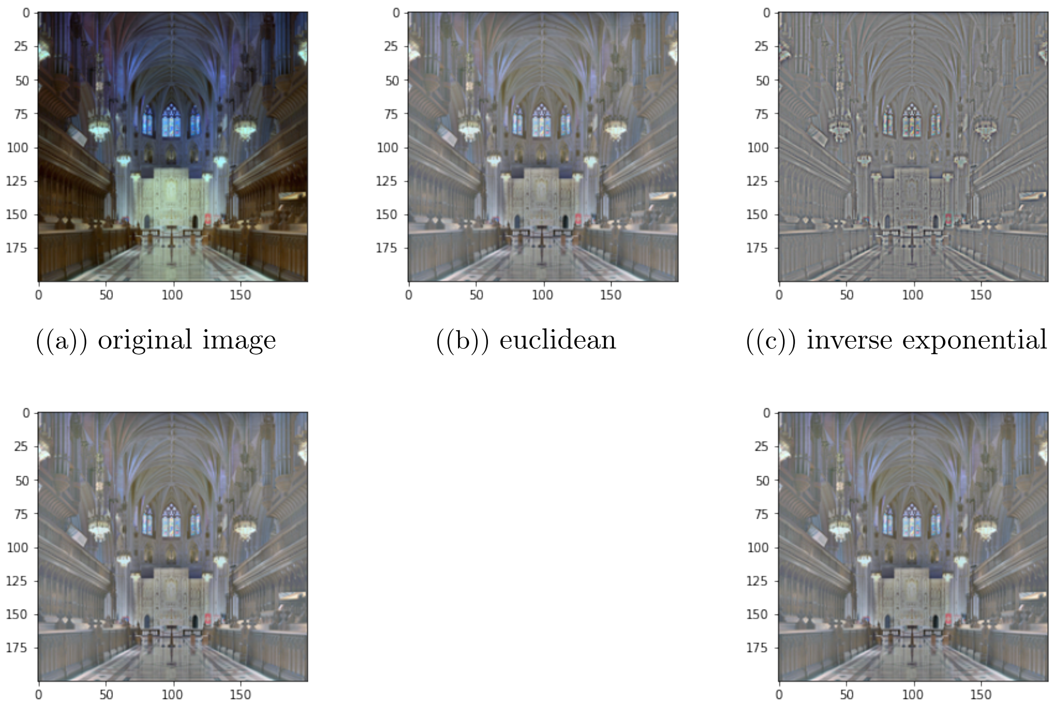
2.6 Dealing with high computation time

ACE's running time was a major difficulty during this project. That is why we had to limit the testing images to small one of a max width equals to 100. Another approach used to reduce computational time is using a subset centered around the concerned pixel while computing Rc(pixel).
After choosing the size of the subset we are going to work with, we notice that the running time decreased by more than 50%.
It remains to verify the possibility of doing such simplification will not damage the output image (see section 3.4).

3 Result analysis

3.1 ACE with different distances

In order to choose the best distance to work with, we tried to run the algorithm with a fixed r(.) function while varying the used distance each time. The results could be seen in Figure 1.



((a)) original image ((b)) euclidean ((c)) inverse exponential

((d)) Manhattan ((e)) maximum

Figure 1: Variation of distance with fixed r(.)

Euclidean distance seems to be doing a slightly better job than the Manhattan and maximum distances. The inverse exponential distance distorted the input image, leading to most information on the scene being lost.
After observing the consistency the euclidean distance gives from different images, we decided to opt for the euclidean distance for our algorithm.

3.2 ACE with different r(.) functions

While we tested different r(.) to make the compromise of color-correction/contrast. The variation of the slope of r(.) acts as a contrast tuner.
In fact, the greater the slope is, the higher the contrast. Yet, since no big visual difference to be mentioned, we have chosen tanh as a reference function in the following tests.

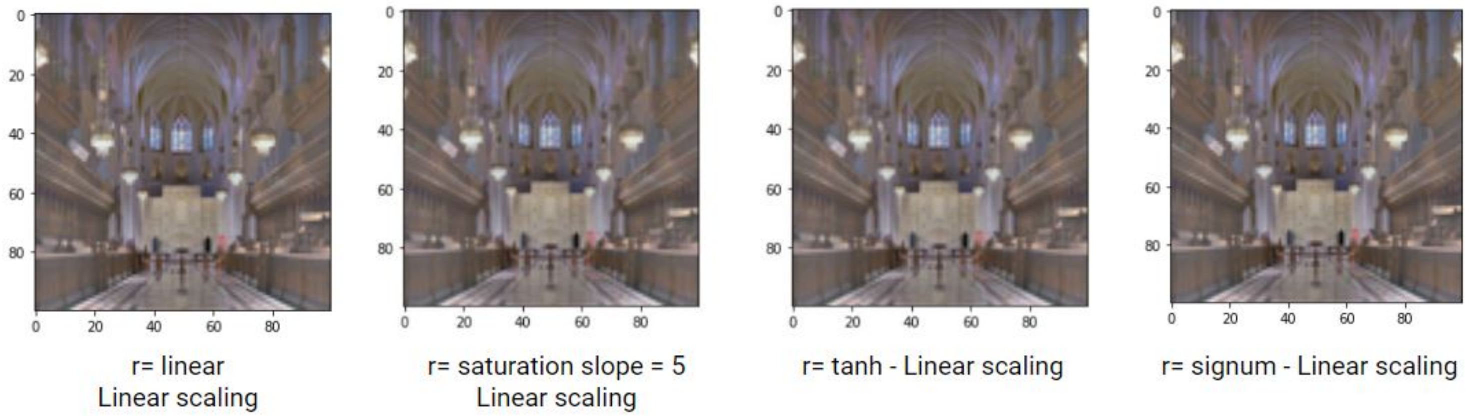


Figure 2: Different r functions - Linearly Scaled

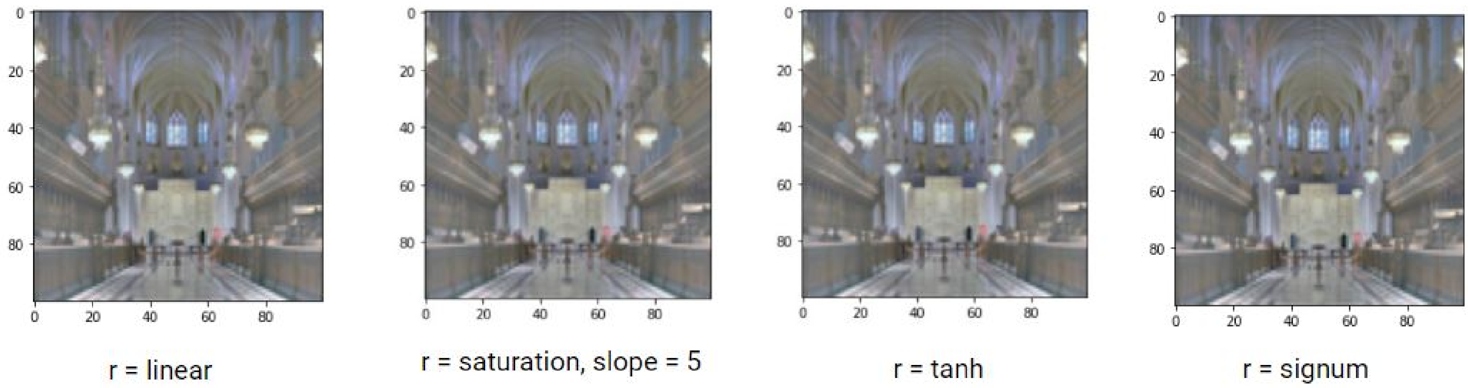


Figure 3: Different r functions - White Patch Gray World Scaled

3.3 ACE results

3.3.1 Output Images and histograms discussion

After exploring the figures below, we notice that ACE moves lightness mean towards the medium gray value.
Moreover, this technique increases the ranged of the used dynamic and the flatness of the histogram.
As a result, we get a more flattened histogram covering almost the whole range of possible values that a pixel can take.

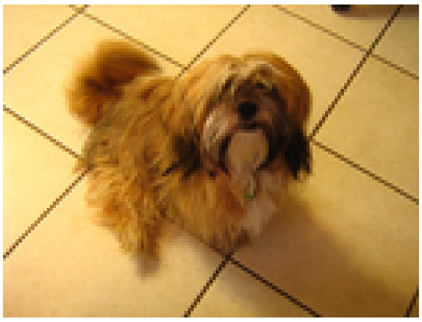


Figure 4: Input

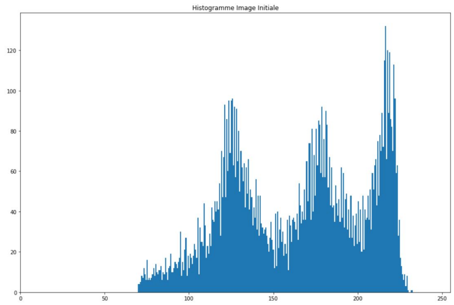


Figure 5: Input Dog Image Histogram

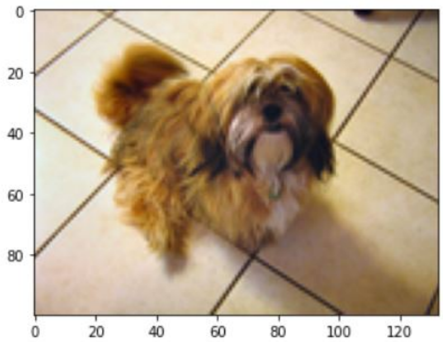


Figure 6: Linearly Scaled Output

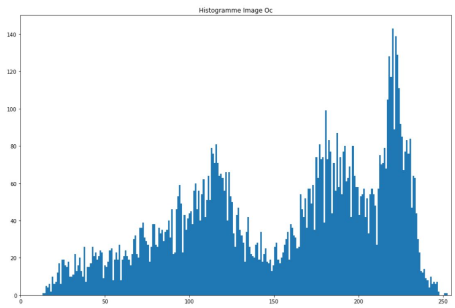


Figure 7: Linearly Scaled Output - Histogram

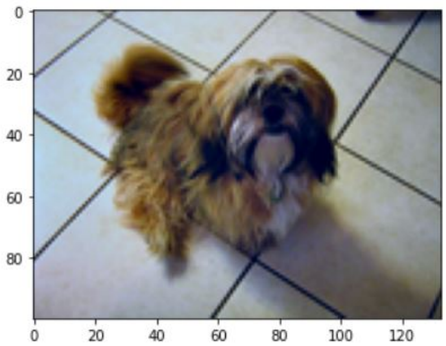


Figure 8: White Patch Gray World Scaled Output

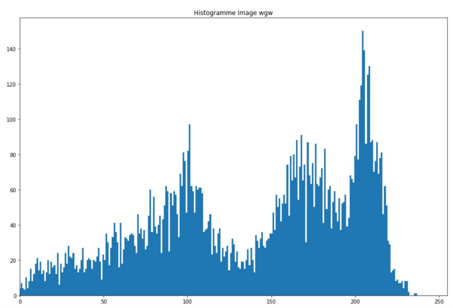


Figure 9: White Patch Gray World Scaled Output - Histogram

3.3.2 Scaling : linear scaling vs white patch gray world scaling

By checking the figures below, we notice that the result of the linear scaling method gives a pale yellowish output. However, the result of the white patch gray world scaling method gives better vivid images that corresponds more to reality.

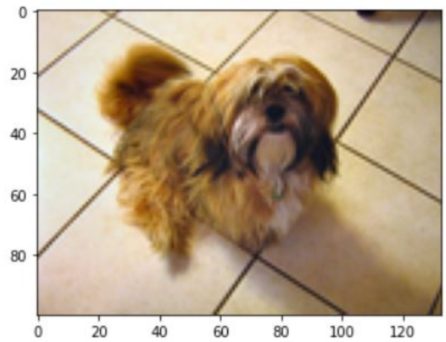


Figure 10: Linearly Scaled Output

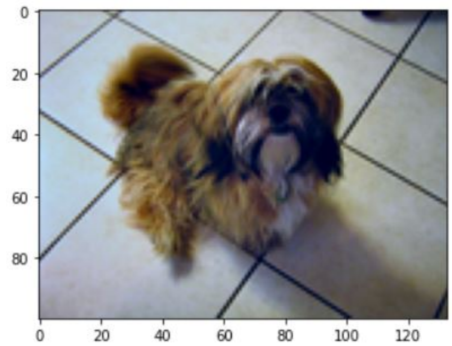


Figure 11: White Patch Gray World Scaled Output

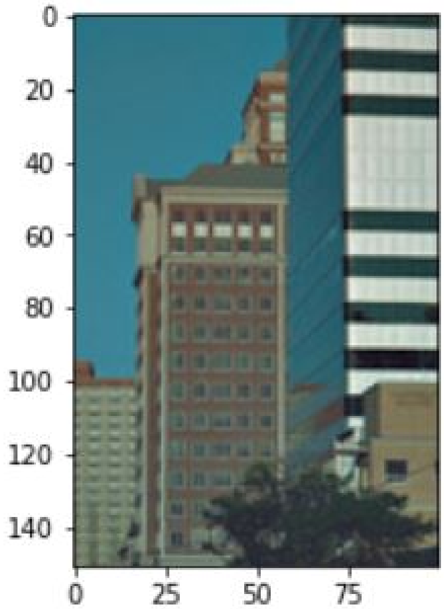


Figure 12: Linearly Scaled Output

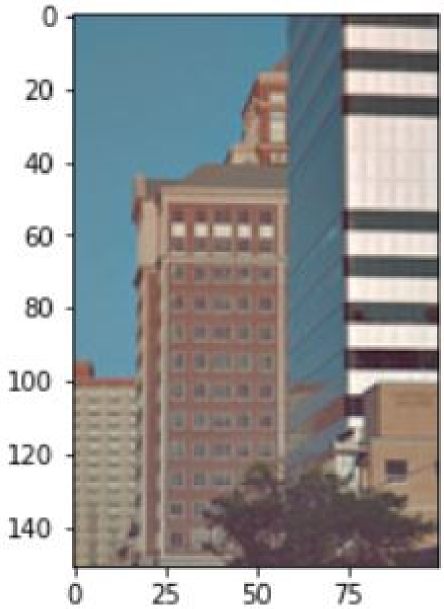


Figure 13: White Patch Gray World Scaled Output

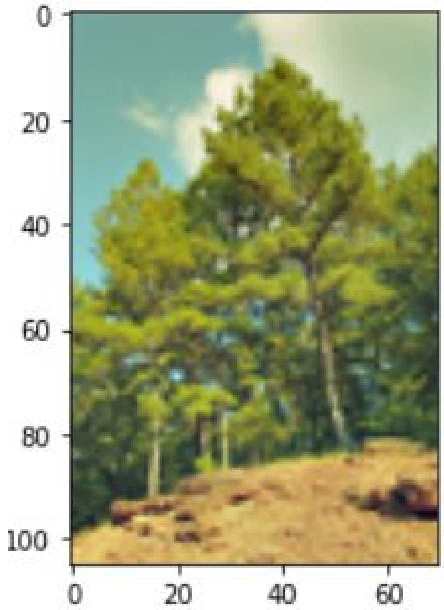


Figure 14: Linearly Scaled Output

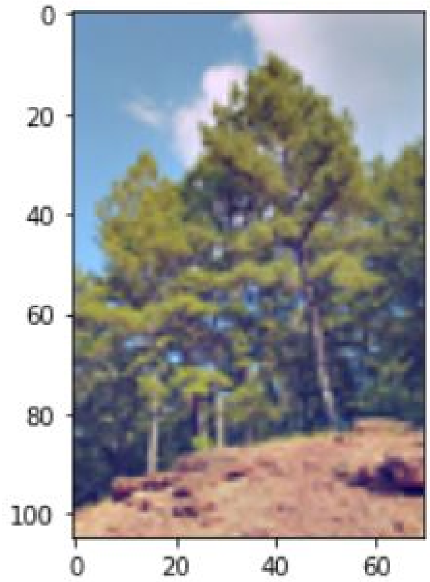


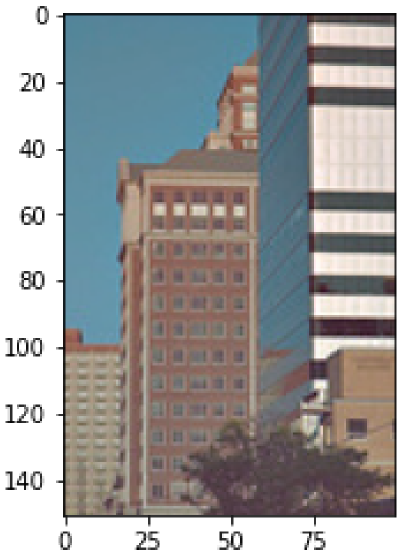
Figure 15: White Patch Gray World Scaled Output

3.4 ACE on subsets

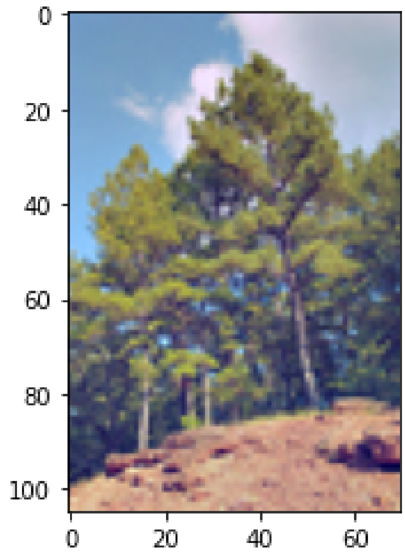
As we have already mentioned before, ACE is a great algorithm for color enhancement, however, it could be very costly in terms of execution time.
To reduce the execution time, we decided to work on smaller subset of the image, centered around the concerned pixels while computing the intermediate image R.
To prove the utility and the possibility of doing so without doing much damage to the original algorithm, we tried to compute the ideal average size n of the subset to use.
In the images represented in Figure 16, we use the whole image as a subset. In Figure 17, we use a subset of size 40 pixels.

Comparing the obtained images in each case, it is visually apparent that the quality of color enhancement using the subset has dropped significantly in some images but remained unchanged for other images.

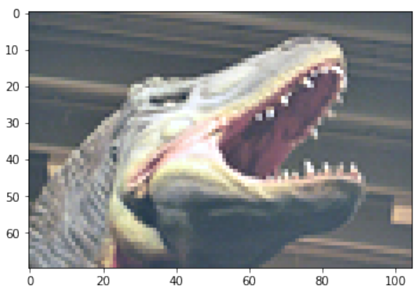
We notice the presence of gray areas at the level of the sky in both the image of the building and the image of the tree. We could also clearly see that the colors became more faded.
The images of the dinosaur and the church remained practically the same.
We conclude that the choice of the size of the subset could be very crucial in terms of determining the quality of the color correction of the resulting image.



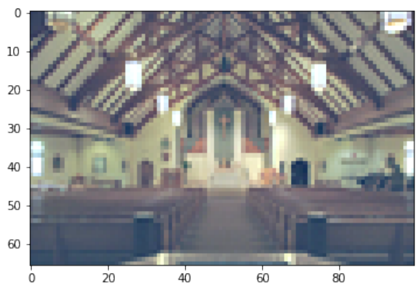
((a))
1



((b))
2

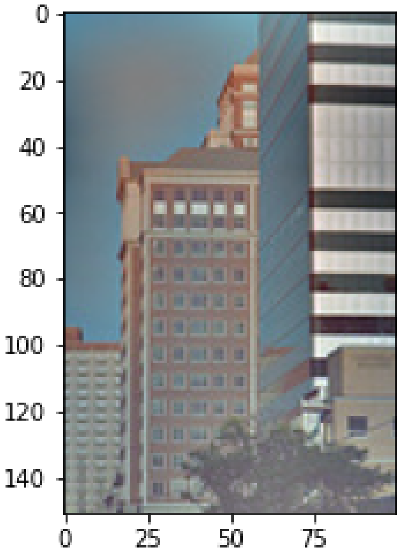


((c))
3

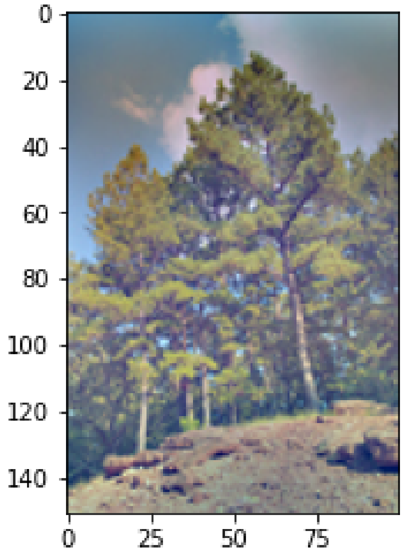


((d))
4

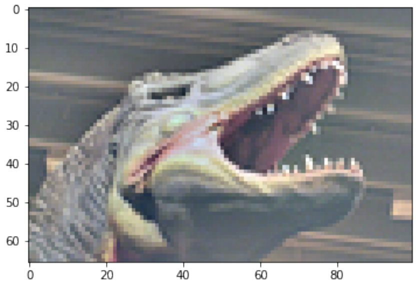
Figure 16: Whole image as a subset



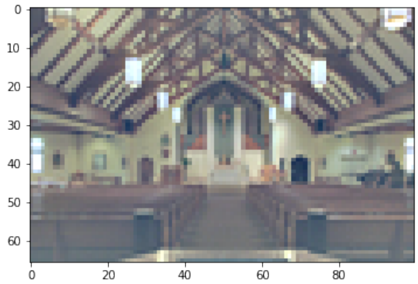
((a))
1



((b))
2



((c))
3



((d))
4

Figure 17: Subset of size 40

4 Improvements

As we have already concluded in the previous section, using ACE is really costly and using the subset method could prove tricky on certain examples. Another way of improving the complexity and computation time of ACE while also conserving the quality of the color enhancement offered by it is using one of two approximation methods: [1]

- Polynomial approximation : This method consists of doing a polynomial approximation on the slope function.
- Interpolation approximation : This approximation decomposes the computation of the intermediate image R into convolutions by using interpolations.

5 Conclusion

ACE is a very useful algorithm for color correction. However, its complexity and computation time are really important.

While we tried to overcome this using subsets, our method proved to have some constraints on the quality of the result.

Using the methods mentioned in the last section of the report could prove to be more useful.

References

[1] Pascal Getreuer. "Automatic color enhancement (ACE) and its fast implementation". In: *Image Processing On Line* 2 (2012), pp. 266–277.

[2] Alessandro Rizzi, Carlo Gatta, and Daniele Marini. "A new algorithm for unsupervised global and local color correction". In: *Pattern Recognition Letters* 24.11 (2003), pp. 1663–1677.