ARI 2129 – Principles of Computer Vision for AI

Daniel Grech [daniel.grech.19@um.edu.mt](mailto:daniel.grech.19@um.edu.mt) Owen Agius [owen.agius.19@um.edu.mt](mailto:owen.agius.19@um.edu.mt)

## Part 1 – Computer Vision Functions

The task at hand is separated into two different sections where the first part tackles image blending and the second part tackles a chroma key implementation. Image blending is achieved in four unique steps. Consider Scene 1 [*Figure 1*] to contain only one object in the image and Scene 2 [*Figure 2*] to contain two or more images in the images.

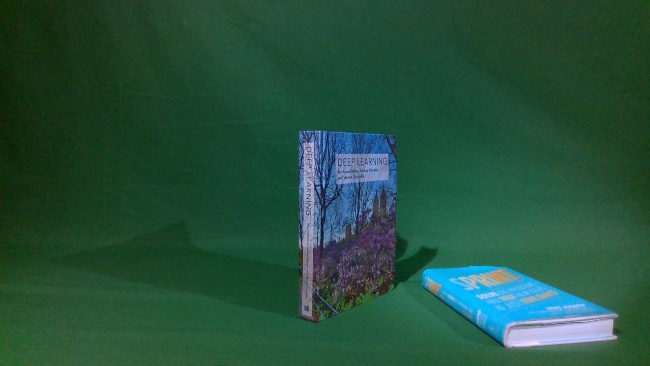


Figure – Scene 2

Figure – Scene 1

1. Extracting an object from Scene 2
2. Applying a filter to the object
3. Blending the object extracted to Scene 1
4. Comparing the blended image to Scene 2

Step 1:

The object is extracted from scene 2 with a simple implementation of a bitwise and operation on Scene 2 and a corresponding mask [*Figure 4*]. The illustration of the output of this stage can be seen in *Figure 3*.

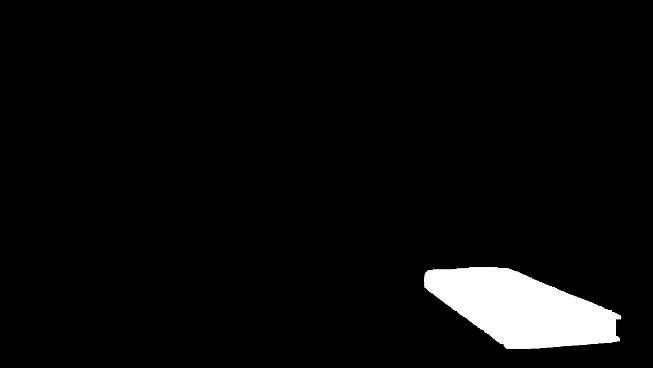


Figure – Output of Step 1

Figure – Mask

Step 2:

The three filters implemented were chosen to be the Bilinear, Gaussian and Sobel X filters. The Sobel X and the Gaussian filters are achieved via an implementation of convolution. Meanwhile, the Bilinear filter uses a hard coded array to achieve its results. The respective results can be found in the below images.



Figure – Image with Bilinear Filter

Figure – Image with Gaussian Filter



Figure – Image with Sobel X Filter

Step 3:

Object blending is the focal point of this task where the aim is to add the extracted images found in *Figures 5-7* to Scene 1 [*Figure 1*]. The addition of the object to Scene 1 would result in an output image very similar to that of Scene 2 [*Figure 2*]. The object blending procedure is processed as illustrated in the flowchart below.

*Figure 8* represents the final and resultant image of all the processes defined above combined, producing a satisfying result of object blending. As it can be seen, the image in *Figure 8* is very similar to that of Scene 2 [*Figure 2*]. The only difference that can be spotted is the occlusion of the shadow of the extracted object. Although a short coming of the object detection algorithm, this was expected since the extraction procedure only captures the image rather than both the image and its shadow.

Figure – Blended Image

Step 4:

Finally, we can measure the error metric between the blended image [*Figure 8*] and Scene 2 [*Figure 2*]. Although to the naked eye one can spot the differences between the images, a more formal and mathematically proven approach has to be implemented. For the error metric the SSD and MSE error metrics were applied.

Sum of Square Differences is one of the measures of a match that is based on pixel-by-pixel intensity between the two images. [1][2] The summation of the square of the product of pixel subtraction is calculated between two images or rather defined by the equation below.

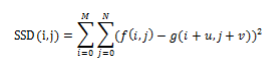
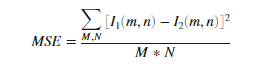


Figure 10 – SSD Equation

Figure – MSE Equation

The result of SSD between *Figure 7* and *Figure 2* returns a value of 201. This value suggests a rather close matching between both images but as predicted earlier, the value is larger due to the occlusion of the shadow.

Comparative to the SSD, MSE defined as the Mean Square Error represents the cumulative squared error between the compressed and original image defined as the *Figure 8*. [3]

The value for MSE on the same images defined in the SSD implementation results to be 15.5145. The value is much more satisfying since the algorithm for MSE computes over the whole image at once rather than iterating through all the pixels as the SSD does.

Part B of Task 1:

Contrastingly to the above functions, the below takes a scene and converts the green background into a background of our choice. This is achieved via two functions who respectively remove the green colour from the image and replace the blank space with a predefined background.

Step 1: Remove Green:

This function sets two sets of 3 values representing BGR values. The two sets act as a threshold for a lower value and a higher value of green pixels. Any colour within the range of these two sets will be defined as an intense green colour and hence occluded from the image. As seen in both *Figures 11 -12*.

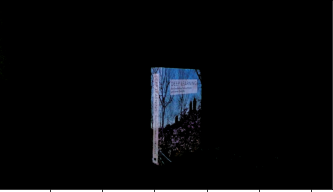
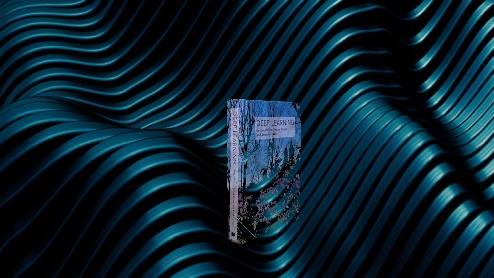


Figure 11 – Chroma Keyed Image

Figure 12- Chroma Keyed Image

Step 2: New Background:

The new background is added via a simple addition blending technique as seen in the images below.



As can be seen on the images to the left, when attempting to add the new background to the extracted book, some features are left out from the book. This was expected since the remove green function removed the green colouring on the cover of the book. Although, this reciprocates when testing with a shoe. The shoe has no green elements, ensuring a loss-less green screen effect.

## Part 2 – Implementing a CV Research Paper

## References

[1] M. B. Hisham, S. N. Yaakob, R. A. A. Raof, A. B. A. Nazren and N. M. Wafi, "Template Matching using Sum of Squared Difference and Normalized Cross Correlation," 2015 IEEE Student Conference on Research and Development (SCOReD), 2015, pp. 100-104, doi: 10.1109/SCORED.2015.7449303.

[2] S. Ourselin, X. Pennec, R. Stefanescu, X. Pennec, and R. Stefanescu, “Robust Registration of Multi-Modal Medical Images : Towards Real-Time Clinical Applications, 2001.

[3] <https://www.mathworks.com/help/vision/ref/psnr.html#:~:text=The%20mean-square%20error%20(MSE,MSE%2C%20the%20lower%20the%20error>. [Accessed: 23/06/21]