ROBT 310 Image Processing

Homework Project 3

By Zarema Balgabekova

1. Introduction

The aim of this Homework Project is to apply the knowledge obtained during lectures on the color image processing in practice. This is done by implementing the Selective Coloring effect using MATLAB. In addition, GUI is created with the help of App Designer, the input is obtained from a webcam using support package for USB webcams, and the "bonus feature button" is added to the GUI. The details on how this work has been performed are provided below.

2. Main Part

In the main task, we perform the Selective Coloring effect, which is used in object tracking, cinematography, and photo editors. Among two provided methods, "colors of interest enclosed by a sphere" method has been chosen as it seems to provide a slightly better result.

MATLAB code is not provided in the report because it is too large. However, it can be seen in the corresponding MATLAB file, where comments are provided for clarification.

2.1. GUI

The created GUI looks as follows:

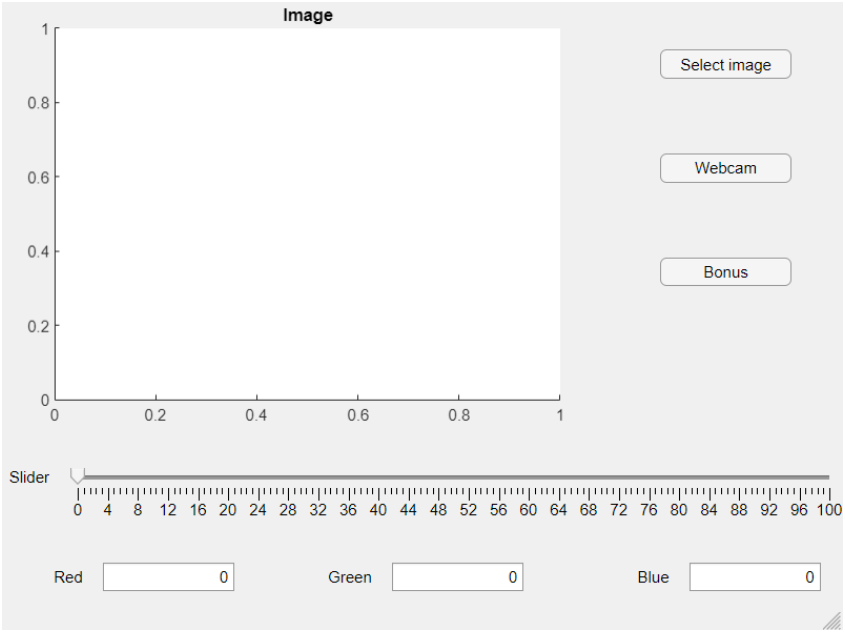


Figure 1: The created GUI

From Figure 1, it is seen that the GUI consists of an axis, where the image is displayed; three buttons: to select the input image, to obtain the input image from a webcam, and the "bonus feature button"; three edit field to display the RGB values of a chosen pixel, and a slider to control the size of the radius of the sphere around the selected color.

All buttons and the slider have their callback functions to perform needed actions.

2.2. Selective coloring for prepared images

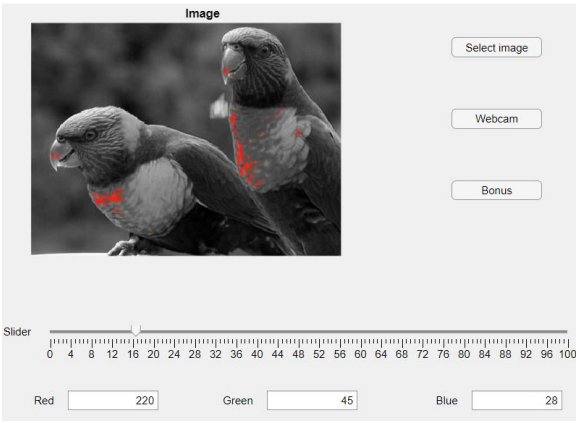
“Select image” button is used to select the input image from prepared images. Within its callback function, we decrease the size of the image by 2 if its size is too big to speed up the selective coloring process. In other case, it will take a longer time. The function ginput( ) is used to pick the pixel. As it cannot be used within the axis, an additional figure, which is closed after choosing the pixel, is shown. Furthermore, we display the RGB values of the pixel in the corresponding edit fields.

After that the slider is used to perform the selective coloring effect. Within its callback function, we use the algorithm similar to one studied during lectures (Shindler’s List effect). The difference is that the color center is equal to the color of the chosen pixel, and the color radius equals the value of the slider.

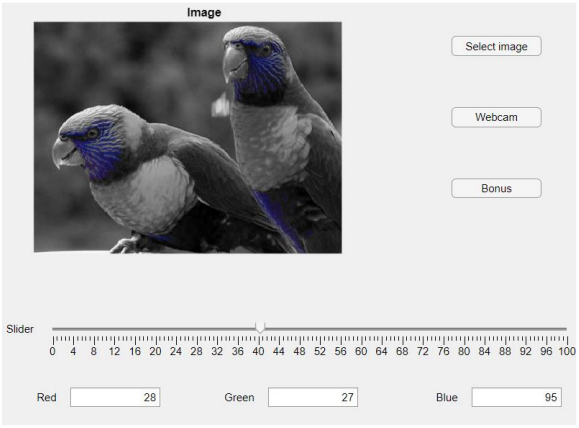
The obtained results are presented below.

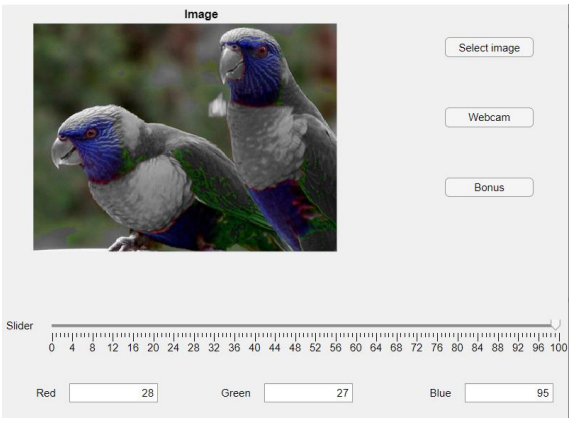
Figure 2: Original image 1

Figure 3: Original image 2

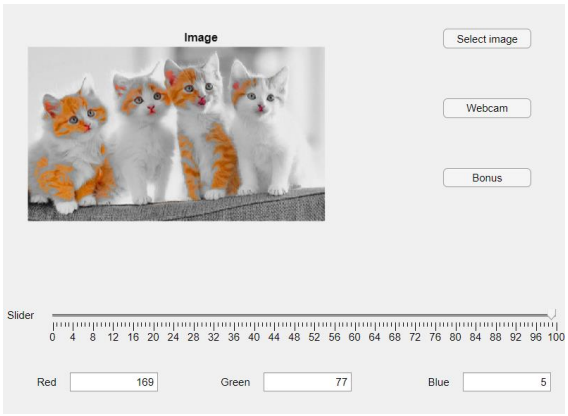


Figures 4-5: Selective coloring for the shade of red for different values of the slider





Figures 6-7: Selective coloring for the shade of blue for different values of the slider



Figures 8-9: Selective coloring for the shade of orange for different values of the slider

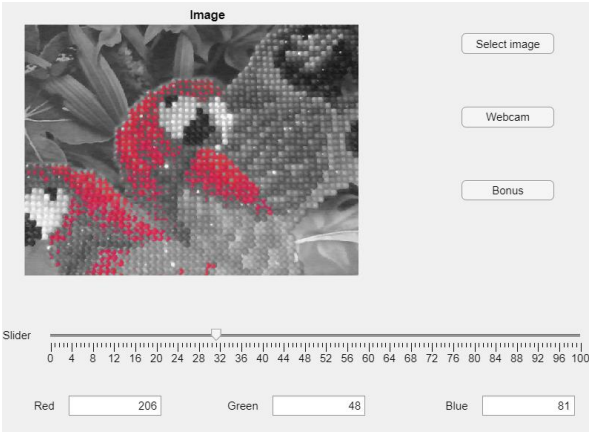
From Figures 2-9, it is seen that with the increasing value of the slider, more pixels of the selected color are displayed on the grayscale image as the radius of the sphere enclosing colors of interest increases. The values of the slider vary from 0 to 100 and can be set to other values. For example, in Figure7, not only shades of blue but also shades of green are displayed. Overall, however, it seems that the default range of the slider provides the appropriate results.

2.3. Selective coloring for images from a webcam

“Webcam” button is used to obtain the input from a webcam. Support package for USB webcams must be used to perform this task. However, I could not install it to my computer due to technical issues that is why online version of MATLAB has been used. Due to the slowness of the online version, I decided to take one snapshot instead of performing the operation continuously in real time. The following steps are the same as for the prepared image.

The following results are obtained.

Figure 10: Image from webcam



Figures 11-12: Selective coloring for the shade of red for different values of the slider for the image from webcam

From Figures 10-12, the results for the image from webcam are similar to the results for prepared images.

2.4. “Bonus Feature Button”

“Bonus” button is used to obtain the result given by the “Bonus Feature Button.” In this task, I rotate the red and blue bands of the input image with the help of the function imrotate( ), and then overlay these bands and the original image using imfuse( ). As a result, the obtained effect is similar to ghost and glitch effects. Initially, I rotated the input image itself and then overlayed the results. Consequently, the ghost effect has been generated. However, with the addition of red and blue bands, a more attractive result is obtained.

The results are as follows:

Figure 13: Bonus for prepared image

Figure 14: Bonus for the image from webcam

Figure 15: Original image 3

Figure 16: Bonus for original image 3

From Figures 13 and 16, it is seen that the pursued result (ghost and glitch effect) has been obtained. The result for the image from webcam is not good enough because the image is not appropriate for the given effects, and it is shown for demonstration purposes only.

3. Conclusion

To conclude, in the course of this Homework Project, we implemented Selective Coloring effect using MATLAB. Furthermore, the GUI has been created and the input from a webcam has been obtained. Due to the difficulties with the support package for USB webcams, it was needed to simplify the problem by taking a snapshot. For the “Bonus Feature Button” task, different ideas were considered, and the most attractive was implemented. Overall, all obtained results can be further improved by using another range of values for the slider or enabling a continuous stream of frames for the USB camera.