Assignment3_D1228803 Adeline Chen 陳宣妤 Report Programming

Assignment 3: Image Reduction and Merge with Frame

How I develop my assignment solution

First of all, I declared the structure for the header to show the information for the result and print it out. Then I defined a function write_image_file, which is responsible for writing the image data and header information to a new BMP file.

In the main function, I first declared some variables for future use, and input users input. After that, I input the image, first, I checked the number of command line, then I opened and read the image file using binary mode ("rb") fopen. If the file exists, you proceed to read various components of the BMP header using fread. The header included information such as file type, size, dimensions, color depth, etc. Then I allocate memory for the color palette (io_palette) and image pixel data (io_imageData) using malloc based on the information read from the header. Then I read the color palette data and image pixel data from the image file using fread and store them in the allocated memory space and print the header information of the input image file.

In the reduced part, I performed several operations related to image processing and file manipulation. I first defined header structures and assigned header values. Then I allocated the memory for storing pixel data of the reduced images.

Then I calculated the row sizes of the original, reduced, and merged images based on their respective dimensions. Lastly, I printed the header information of the reduced image file out.

In the merged part, I first call the `write_image_file` function to write the reduced image to a file. Then by using nested loops, I iterate through each pixel of the reduced image to merge it with a frame. For each pixel of the reduced image, I calculate the corresponding pixel positions in the merged image for all four quadrants. Then I iterated through each pixel of the merged image and add a frame around it. The frame is added based on conditions within a certain distance from the image border by frameSize, and colored with the specified frame color frameColorB, frameColorG, frameColorR. This process creates a border around the merged image. Then, I called `write_image_file` to write the

merged image with the frame to a file. Finally, I free the dynamically allocated memory for the palette of the input image, the pixel data of the input image , the merged image, and the reduced image.

Overall, the process of merging a frame around a reduced image and then writing the resulting merged image to a file.