

IMAGE RESIZING

(Image Processing in C)

A Project Report

Submitted by

I.V.SRICHANDRA- CB.EN.U4AIE22165.

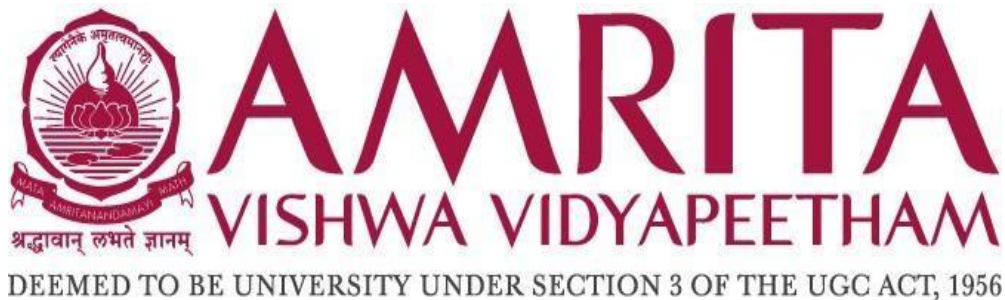
KISHOR .S- CB.EN.U4AIE22128.

Y.SUDHEER KUMAR CHOWDARY- CB.EN.U4AIE22151.

VILLA SRI SAI ABHIRAM- CB.EN.U4AIE22168.

As a part of the subject

Problem Solving and C Programming



Centre for Computational Engineering and Networking

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE – 641112 (INDIA)

February - 2023

Declaration

We declare that the Submitted Report is our original work and no values and context of it have been copy-pasted from anywhere else. We take full responsibility, that if in the future, the report is found invalid or copied, the last decision will be of the Faculty concerned. Any form of plagiarism will lead to disqualification of the report

Place: Ettimadai

Date: 8-02-2023

Contents

Chapters	Page No
Acknowledgment	A
Abstract	5
1. Introduction	
1.1 Overview	6
1.2 Image resizing	7
1.3 Image format	8
2. Methodology	
2.1 Logic	8
2.2 Algorithm	9
3. Source Code	
3.1 Source Code	10
3.2 Explanation of libraries	11
3.3 Explanation of functions	12
4. Results	
4.1 Output	16
5. Conclusion	
5.1 References	20

Acknowledgment

Our Mentor, Ms. Aswathy, was a constant source of inspiration, thus we want to start by expressing our gratitude to her. She challenged us to be creative in our thinking and pushed us to complete this Project without hesitation. Without her assistance and oversight, this endeavour would not have been possible.

We want to express our gratitude to Amrita Vishwa Vidyapeetham Coimbatore for giving me the chance to work on this project. Last but not least, we would want to express our sincere gratitude to everyone who helped make this initiative a success.

Abstract

- 1.** The process of image resizing involves altering the dimensions of an image, either by decreasing its size (downsampling) or increasing it (upsampling). To preserve image quality during resizing, techniques like interpolation are employed. Resizing images is commonly done for fitting the image into a specific space, reducing file size, or preparing for further image processing. Implementing image resizing in C programming involves utilizing algorithms like interpolation to calculate missing pixels. The process starts by reading an image file, adjusting the image data to change its dimensions, and then saving the resized image data to a new file. C provides libraries and functions, such as the image I/O library and graphics library, to aid in reading, manipulating, and writing image data, making it ideal for implementing image resizing

1Introduction:

1.1 Overview of C:

C is a high-level programming language that is widely used for system programming, embedded programming, and other applications. It is known for its efficiency and the ability to produce low-level code, making it an ideal choice for image processing. Image processing involves transforming or analyzing digital images to extract useful information or to produce new images.

In image processing using C, the image data is typically read into the program as an array of pixels and various operations are performed on this array to achieve the desired result. Some common image processing tasks include resizing, thresholding, filtering, edge detection, and segmentation. C provides libraries and functions for reading, manipulating, and writing image data, as well as numerical libraries for mathematical operations.

In addition to its efficiency and low-level capabilities, C also offers a number of advantages for image processing. Firstly, it provides a high level of control over the image data, allowing for fine-tuned manipulation and analysis of images. This is important for tasks that require precise control, such as edge detection and segmentation.

Another advantage of C in image processing is its ability to handle large amounts of data and perform computationally intensive tasks. This makes it suitable for processing large images and working with large datasets. It also provides a number of libraries and functions that are optimized for image processing, allowing for fast and efficient processing of images.

Finally, C is widely used in the field of image processing and has a large community of developers and users. This means that there are many resources available for learning and improving image processing skills, and a wealth of libraries, tools, and code snippets available for use.

Overall, C is a powerful and versatile tool for image processing, making it an ideal choice for many image processing applications and projects

1.2 Image Resizing:

Image resizing is the process of changing the dimensions of a digital image. This can be done either by reducing the size of the image, known as downsampling, or increasing its size, known as upsampling. The goal of image resizing is to adjust the dimensions of an image to fit a specific space or format, to reduce the file size of an image, or to prepare the image for further image processing tasks.

Resizing an image can have an impact on its quality as information is either lost or added when the image dimensions change. This can result in blurry or pixelated images. To mitigate these effects, various algorithms and techniques can be used, such as interpolation and resampling.

Interpolation is a common technique used in image resizing, where missing pixels are estimated based on the surrounding pixels. This allows for a smooth and seamless transition between pixels, helping to preserve the quality of the image. Interpolation algorithms, such as bilinear and bicubic, use different methods to estimate the missing pixels, with bicubic interpolation generally producing higher-quality results.

The process of image resizing typically involves reading an image file, manipulating the image data to change its dimensions, and then writing the resized image data to a new file. This can be done using programming languages such as C or using specialized software or tools.

In conclusion, image resizing is a critical component of many image processing tasks, allowing for the adjustment of image dimensions to meet specific requirements or needs. By using interpolation and resampling techniques, the impact of resizing on image quality can be minimized, producing high-quality images that meet the desired specifications.

1.3 Image Format:

JPEG (Joint Photographic Experts Group) is a commonly used image compression standard that uses lossy compression techniques to reduce the file size of an image without sacrificing too much visual quality.

2.Methodology

2.1Logic:

This C program that implements the logic for resizing an image. It uses the stb_image library to read, resize, and write image data.

The code first reads an image file using the stbi_load function, which returns a pointer to an array of image data. The width, height, and number of channels in the image are also stored in the respective variables.

The user is then prompted to input a scale factor, which determines the new size of the image. The new width and height of the image are calculated by dividing the original width and height by the scale factor. A block of memory is allocated to store the resized image data using the malloc function.

The stbir_resize_uint8 function is then used to resize the image data, and the resized image is written to two files in PNG and JPG format using the stbi_write_png and stbi_write_jpg functions.

Finally, the program frees the memory used for the image and resized image data using the stbi_image_free and free functions, respectively.

2.2 Algorithm:

- Load the image using the `stbi_load` function.
- Check if the image was successfully loaded. If not, print an error message and exit the program.
- Print information about the loaded image, including its width, height, and number of channels.
- Ask the user for the scale factor.
- Calculate the new width and height of the resized image based on the scale factor.
- Allocate memory for the resized image using `malloc`.
- Use the `stbir_resize_uint8` function to resize the original image to the new size.
- Write the resized image as a PNG file and as a JPG file using the `stbi_write_png` and `stbi_write_jpg` functions.
- Print the dimensions of the output image.
- Free the memory used by the original image and the resized image using `stbi_image_free` and `free`.

3.Source Code:

```
#include <stdio.h>
#include <stdlib.h>

#define STB_IMAGE_IMPLEMENTATION
#include "stb_image/stb_image.h"
#define STB_IMAGE_RESIZE_IMPLEMENTATION
#include "stb_image/stb_image_resize.h"
#define STB_IMAGE_WRITE_IMPLEMENTATION
#include "stb_image/stb_image_write.h"

int main(void) {
    int width, height, channels;
    unsigned char *img = stbi_load("abc.jpg", &width, &height,
    &channels, 0);
    if(img == NULL) {
        printf("Error in loading the image\n");
        exit(1);
    }
    printf("Loaded image with a width of %dpx, a height of %dpx and %d
    channels\n", width, height, channels);
    float scalefactor;
    printf("Enter the scale factor: ");
    scanf("%f",&scalefactor);
    int new_width = width / scalefactor;
    int new_height = height /scalefactor;
    unsigned char *resized_img = malloc(new_width * new_height *
    channels);

    stbir_resize_uint8(img, width, height, 0, resized_img, new_width,
    new_height, 0, channels);

    stbi_write_png("abc_resized.png", new_width, new_height, channels,
    resized_img, new_width * channels);
    stbi_write_jpg("abc_resized.jpg", new_width, new_height, channels,
    resized_img, 100);
```

```
printf("Output image with a width of %dpx, a height of  
%dpx",new_width,new_height);  
stbi_image_free(img);  
free(resized_img);  
}
```

3Explanation:

3.1Libraries:

stdio.h - This library provides functions for input and output operations. For example, the "printf" function is used to output messages to the console.

stdlib.h - This library provides functions for memory management and system operations. For example, the "exit" function is used to terminate the program

stb_image.h - This is a header file from the "stb_image" library, which is used to load image files. The "stbi_load" function is used to load the image file "abc.jpg"

stb_image_resize.h - This is a header file from the "stb_image_resize" library, which is used to resize images. The "stbir_resize_uint8" function is used to resize the loaded image.

stb_image_write.h - This is a header file from the "stb_image_write" library, which is used to write image files. The "stbi_write_png" and "stbi_write_jpg" functions are used to save the resized image as PNG and JPG files respectively.

3.2 Functions:

stbi_load - This function loads an image file into memory and returns a pointer to the image data. The first argument is the file name, and the next three arguments are pointers to variables that will store the width, height, and number of channels in the image. The last argument is used to specify the desired number of channels in the image, which is set to 0 to indicate that the original number of channels should be preserved.

printf - This function outputs a message to the console. In this program, it is used to output messages such as "Loaded image with a width of ...px, a height of ...px and ... channels".

scanf - This function reads input from the console. In this program, it is used to read the scale factor entered by the user.

malloc - This function allocates memory dynamically. In this program, it is used to allocate memory for the resized image data.

stbir_resize_uint8 - This function resizes an image. The first argument is a pointer to the original image data, the second and third arguments are the width and height of the original image, the fourth argument is unused, the fifth argument is a pointer to the memory that will store the resized image data, the sixth and seventh arguments are the new width and height of the resized image, the eighth argument is unused, and the last argument is the number of channels in the image.

stbi_write_png - This function writes an image to a PNG file. The first argument is the file name, the second and third arguments are the width and height of the image, the fourth argument is the number of channels in the image, the fifth argument is a pointer to the image data, and the last argument is the number of bytes per scanline in the image.

stbi_write_jpg - This function writes an image to a JPG file. The first argument is the file name, the second and third arguments are the width and height of the image, the fourth argument is the number of channels in the image, the fifth argument is a pointer to the image data, and the last argument is the quality of the JPG image, where a value of 100 indicates maximum quality.

4.Results:

4.1Output:

Input Image:



Size:
5236x 3491

OUTPUT FOR UPSCALING:



Size :7854x5236

OUTPUT OF CODE:

```
Loaded image with a width of 5236px, a height of 3491px and 3 channels  
Enter the scale factor: 1.5  
Output image with a width of 7854px, a height of 5236px
```

Input Image:



Size:1000x1503

Output Image Down Scaling:



Size:800x1202

Output code:

```
P3 C:\Users\I.V.SRICHANDRA\Desktop\Amrita\Amrita 1st semester\problem so
Loaded image with a width of 1000px, a height of 1503px and 3 channels
Enter the scale factor: 0.8
Output image with a width of 800px, a height of 1202px
```

5.Conclusion:

Resizing an image can be useful in image processing tasks such as object detection, image recognition, and image segmentation because it allows the processing to be performed on a smaller image, reducing computational time and memory usage.

Resizing images is frequently required for web development, as images that are too large can slow down website loading times and take up too much storage space.

Resizing an image to a specific size is important for printing because printing an image that is too large or too small can result in a loss of detail or quality.

5.1References:

- 1.) <https://unsplash.com/s/photos/person>
- 2.) <https://books.google.com/books?id=ntmFzQEeLQcC&lpg=PA3&ots=uV7WZOcGh&dq=resampling%20images%20c%20programming&pg=PA3#v=onepage&q=resampling%20images%20c%20programming&f=false>
- 3.) <https://homepages.inf.ed.ac.uk/rbf/BOOKS/PHILLIPS/cips2ed.pdf>