PROGRAMMING LANGUAGES

Individual Project

LENA: Photo Editor

Author:

Filiberto Francisco Vázquez Rodríguez

A01206199

Tutor:

Benjamín Valdés Aguirre

Index

1. Introduction & State of the Art

As defined by Dr. Ganney [b]; digital image processing involves manipulating images using digital computers. Its use has increased exponentially in recent decades. Its applications range from medicine to entertainment, smart agriculture, surveillance systems, etc. Multimedia systems, one of the pillars of the modern information society, rely heavily on the processing of digital images. Today, image processing is one of the fastest growing technologies. It is also a critical area of research in engineering and computer science.

In other words, image processing is a method of performing certain operations on an image to obtain an enhanced image or extract useful information. This is a type of processing in which the input is an image and the output can be another image with different features. In computer science an image can be treated as a matrix by using a numerical representation of the values within the image, where every value of the matrix stores each pixel's RGB value. Filters can be applied to the image by performing specific operations to those values.

Once all the pixels of an image are converted, it can be represented by a matrix that stores integer values. The width and length of this matrix is equivalent to the amount of pixels present in the image. Each pixel saves 2 important data values: position and gray level. The position is set by the two coordinates of the sampling point on the scan line, namely the row and the column. The integer that indicates the brightness of the pixel's position is known as the gray level. Any digital image processing is based on the digital matrix, which are the images displayed by the digital matrix.

The following image represents how the picture's pixels can be represented as a matrix, where each pixel has its gray level with a range from 0 to 255:

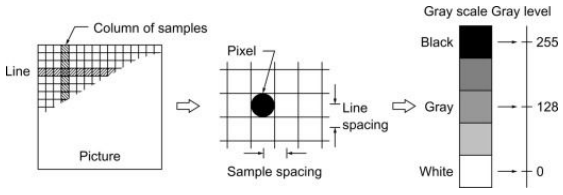


Fig. 1 [Pixel Matrix Representation]

2. Problem Description

One of the many applications of digital image processing is image filtering and editing. We use these features everyday nowadays in mobile and desktop apps to simply edit our photos. The amount of filters that can be implemented is limitless, but the thing that varies from image editor to image editor is the time it takes for the image to be edited.

As users, we expect our photo editors to be as fast as possible, hence the aim of this project is to generate an image editor that takes a shorter amount of time filtering an image than other image editors. What's expected of this photo editor is to be able to apply different filters to a photo by applying efficient image processing techniques. The goal is to achieve a better performance when it comes to the image processing.

3. Solution

Image processing is a heavy number-crunching task, therefore concurrent and parallel programming are viable implementations to solve speed up these photo-editing tasks. Parallel programming provides an efficient way to increase the performance of the image processing by dividing the tasks using 2 cores.

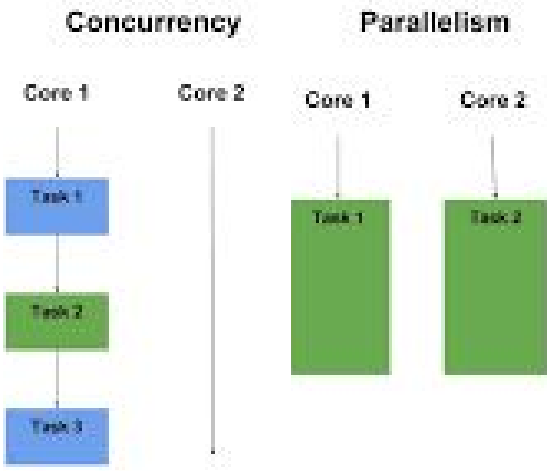


Fig. 2 [Concurrency and Parallelism Thread Representation]

As it can be seen in Fig. 2, in concurrent programming we have many tasks operating at the same time, whereas in parallelism have tasks broken into subtasks that can be processed in parallel by two cores. The Photo Editor is both concurrent and parallel, since it works on multiple tasks at the same time, and also breaks down tasks into subtasks for parallel execution.

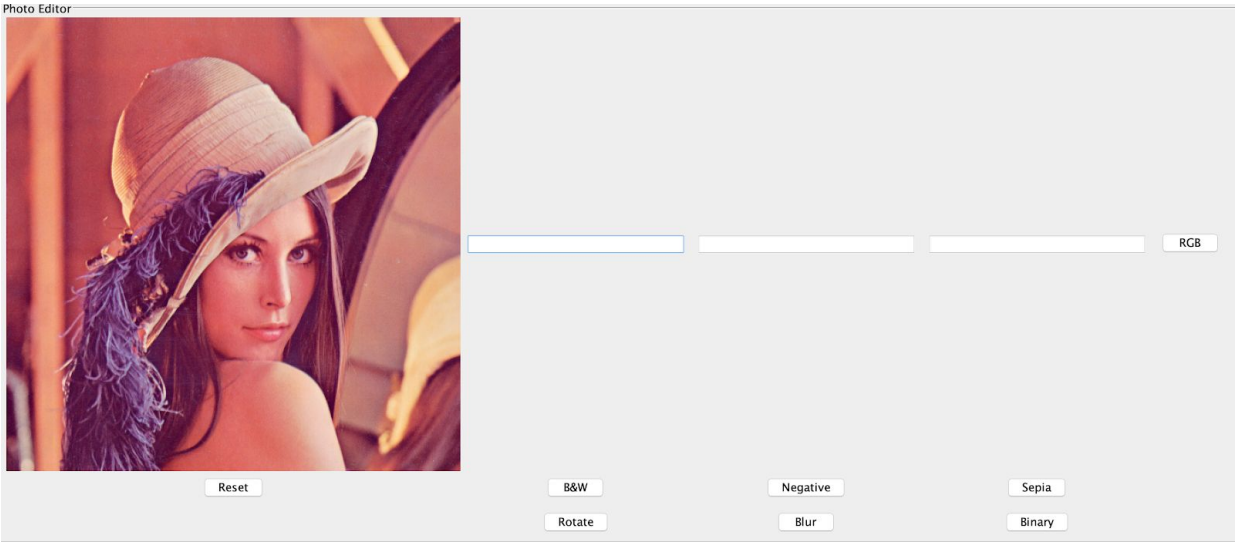


Fig. 3 [LENA: Photo Editor GUI]

Fig. 3 shows the GUI that allows the user to edit a square photo such as those in Instagram to apply different filters to it. It allows the user to write and set the Red Green and Blue values of the image and even do some other preset filters such as Rotate or Binary. Thanks to the usage of a kernel and a Java Convolution operation the Blur filter is achieved.

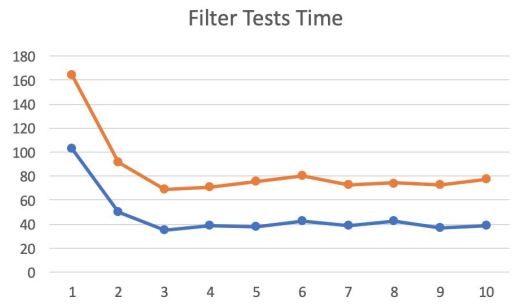
When it comes to the why of using Parallelism as a solution to tackle the problem, the reason is that parallel computing has the capacity of dividing a task in a way that the task executes with maximum efficiency in minimum time.

4. Results

The filtering was successful and time efficient. In order to prove if the parallel approach was indeed faster than the sequential approach, 10 filtering tests were made. Each test used both the sequential and parallel approach:

Image Processing Time Tests			
Test	**Sequential Time (ms)**	**Parallel Time (ms)**	**Comparison Rate**
1	**103**	**61**	**40.78%**
2	**50**	**41**	**18.00%**
3	**35**	**34**	**2.86%**
4	**39**	**32**	**17.95%**
5	**38**	**37**	**2.63%**
6	**42**	**38**	**9.52%**
7	**39**	**34**	**12.82%**
8	**42**	**32**	**23.81%**
9	**37**	**36**	**2.70%**
10	**39**	**38**	**2.56%**

Table 1 [Time Tests]



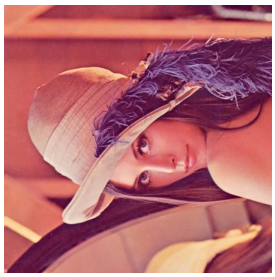
Graph 1 [Parallel (blue) vs Sequential (red) Comparison]

Graph 1 is the graphical representation of Table 1; it shows for each task (X-Axis of the graph) a clear representation of how the time (Y-Axis of the graph) it takes for the parallel tasks to complete is considerably smaller than the time it takes for the sequential tasks to finish.

The following images are some of the results of the processed photo:

RGB (Yellow)

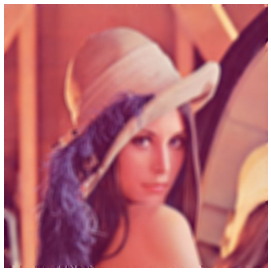
Negative



Rotated

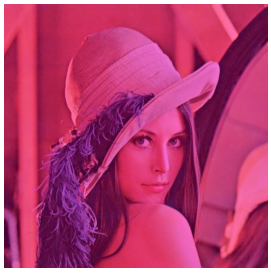
Sepia

Gray



Blurry

Binary



RGB (Red)

5. Conclusions

Concurrency and Parallelism are two very useful paradigms that can help improve the efficiency of relevant applications used by society on a daily basis and image processing is one of them. Therefore it's the perfect playground for improving existing image processing methods and even studying new possibilities thanks to other technologies such as machine learning.

6. Evidence & references

a. Source Code GitHub Repository: https://github.com/filyv/Programming-Languages/tree/master/Final%20Project/PL_Project

b. Image Processing. Clinical Engineering, Ganney, 2014: https://www.sciencedirect.com/topics/engineering/image-processing

c. Digital Image Processing. University of Tartu, 2014: https://sisu.ut.ee/imageprocessing/book/1

d. Parallel Image Processing Techniques, Benefits and Limitations. Saxena, Sharma, 2016:

e. https://www.researchgate.net/publication/297629054_Parallel_Image_Processing_Techniques_Benefits_and_Limitations

f. The Art of Image Processing with Java. Hunt, 2016: https://books.google.com.mx/books?id=AKzMBQAAQBAJ&pg=PA321&lpg=PA321&dq=CONCURRENCY+IMAGE+PROCESSING&source=bl&ots=m0yaiVgViZ&sig=ACfU3U24qoTd316xSXeCHqHkCGQ9DKK16A&hl=en&sa=X&ved=2ahUKEwiz2aS99_zlAhVF-6wKHdlTAHkQ6AEwB3oECCoQAQ#v=onepage&q&f=false

g. Asynchronous Programming in Java. Karunakaran, 2016: https://coding2fun.wordpress.com/2016/08/09/asynchronous-programming-in-java/

h. Concurrency vs. Parallelism. Jenkov, 2015: http://tutorials.jenkov.com/java-concurrency/concurrency-vs-parallelism.html