Final Project

Image Processing with CUDA

Professor:

Benjamín Valdés Aguirre

May 25th, 2021

Creating image filters and modifications using CUDA and C++

Carlos Miguel Negrete Barrientos A012087833

Table of Contents

Abstract **3**

Theoretical Background **4**

Parallelization 4

CUDA Parallelization 4

Image 1. CPUs cores vs. GPUs cores 5

Image processing 5

Image 2. Image pixels 6

Image 3. Pixel zoom in 6

Implementation **6**

Program 6

Image 4. Program running: step 1 7

Image 5. Program running: image preview 8

Image 6. Program running: output 8

Image 7. Memory allocation in the GPU 9

Filter 1: Black to Red 9

Image 8. Dark color 10

Filter 2: Black lines 10

Filter 3: Deletes the background 10

Tests **10**

Image 1, group of testing 1 11

Table 1. Group of testing 1 11

Image 2, group of testing 2 11

Table 2. Group of testing 2 12

Conclusions **13**

Appendixes **14**

Appendix 1: Dependencies of the project 14

GPU used 14

Libraries 15

Appendix 2: Group of testing 1 19

Appendix 3: Group of testing 2 21

Abstract

In recent years the usage of GPUs has became popular, mainly because they are capable of being used for crypto mining. GPUs are extremely powerful and can make a lot of operations in just a few seconds or less. But, GPUs are not popular only in that segment, gaming is another aspect that really needs excellent performance in order to have the best experience with modern games. During this project, the power of a GPU can be seen by image processing. The task of changing some color in a photo can be difficult to perform if the file is really big, but for CUDA and the GPU it is a really quick activity.

Theoretical Background

Parallelization

It is not a secret that day by day computers are being more and more powerful. And there are no reasons for stopping the progress companies are making on this. Nowadays, people want to perform more complex processes and computer activities. One of the possible solutions for handling complex computational operations can be found in the concept of parallelization, this way of thinking when programming in union with the technical specifications of modern devices may help programmers to divide huge and expensive processes in subtasks. The most common way of parallelization is with the CPU. Modern hardware can handle multiple activities at the same time; parallelization is something similar to an operation of a big company. Probably they need to build a single house, obviously this task can't be performed by only one person. Therefore, a certain number of people is needed. There will be a person focused on designing the blueprints, another one will be buying the material needed and other persons will be performing some other subtasks; all of them at the same time. Something similar happens with the parallelization on computers.

Modern CPUs are capable of calling some "workers'' in order to perform multiple subtasks of a master task. Probably the last example of a house can be done with six or ten persons. However, what happens if the company needs to build a very huge skyscraper? Those six or ten persons won't be enough for the task, the company must call more workers. Nevertheless, this can't be done in the computer world, because CPUs can't handle large numbers of "people". The solution to this problem is the GPU. [2][3]

CUDA Parallelization

GPUs are dedicated units to process graphics. The architecture of GPUs are thinked specially for parallel processing and their cores are optimized for this, while CPUs only have some cores capable of this. Since GPUs have more cores than CPUs, it can be understood that they can handle more "people'' and more efficiently than CPUs. It can be more easy to understand with image 1 extracted from Nvidia documentation. [4]

CPUs Cores vs GPUs Cores

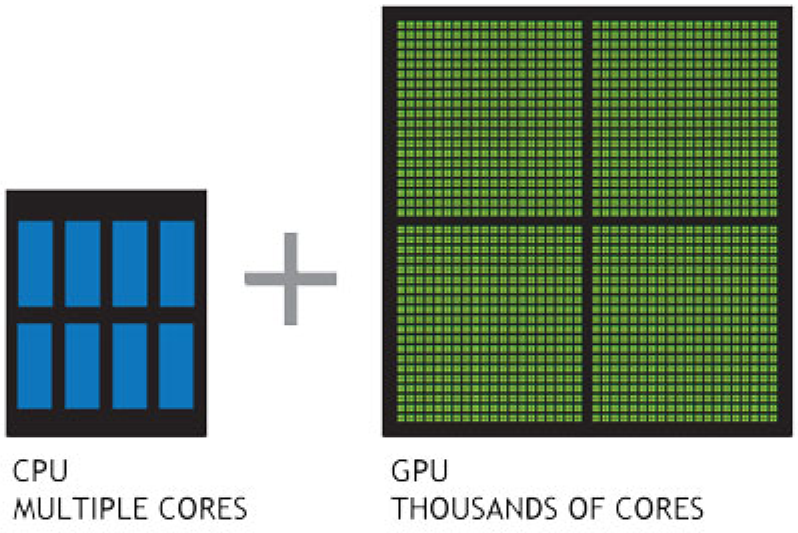


Image 1. CPUs cores vs. GPUs cores

GPUs are really good at gaming and scientific research and there are a lot of GPUs out in the market. For this report purposes, the focus will be on NVIDIA GPUs, since the work presented is developed using CUDA.

CUDA is a way of parallel programming. As it was mentioned before, dividing in subtasks a more complex task. CUDA can be seen as a complement for other languages, however the most common one is C/C++. CUDA mustn't be confused with another language or any kind of API or library. It is a platform and a model of programming. Something to take in consideration is that CUDA only works with NVIDIA GPUs.

The key point of using CUDA is the capacity the programmer has to execute some of the most demanding tasks for a common CPU directly on the GPU. That is the reason why image processing is the perfect way to experiment with CUDA, this can be performed very quickly with CUDA.

Image processing

An image is a file that has a certain number of pixels, in other words some spaces where certain information is stored, such as the color. Each space can be seen by the user as a colored square, as it can be seen on image 2.

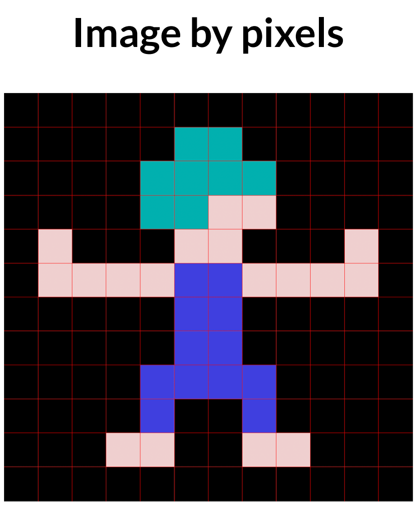
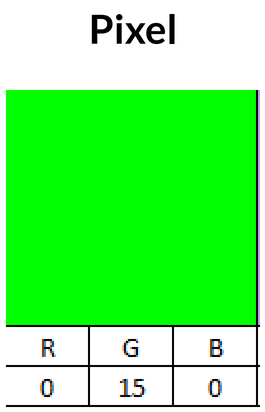


Image 2. Image pixels

The previous image is 12 x 12 pixels, each pixel has the information of what color needs to show. Each pixel has three or 4 channels that can be modified. On normal images, jpg for example, has only three channels; red, green and blue. Each channel can have a value between 0 and 255. Graphically, the pixel can be seen as in image 3.



R	G	B
0	15	0

Image 3. Pixel zoom in

The displayed color on the pixel will depend on the values given to each channel. Then, the principle of applying some changes on the colors of an image are being set. There needs to be an image and there needs to be some access in order to modify each pixel channel's value.

Implementation

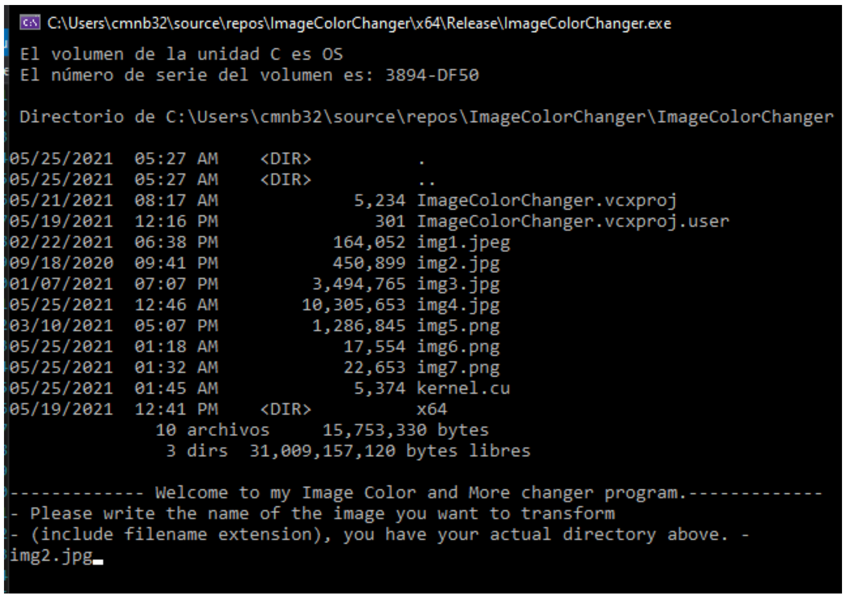
Program

An important consideration is that in this project there is a library as a dependency for reading, writing and showing the images. See appendix 1 in order to see how to install it.

Moving forward, the program presented will let the user do the following:

- Choose an image from the computer using the terminal. (the program automatically displays the actual directory files)
    - The user must specify the name of the image to be modified, including its filename extension.

Program Running: Step 1



```
C:\Users\cmnb32\source\repos\ImageColorChanger\x64\Release\ImageColorChanger.exe
 El volumen de la unidad C es OS
 El número de serie del volumen es: 3894-DF50

 Directorio de C:\Users\cmnb32\source\repos\ImageColorChanger\ImageColorChanger

05/25/2021  05:27 AM    <DIR>          .
05/25/2021  05:27 AM    <DIR>          ..
05/21/2021  08:17 AM             5,234 ImageColorChanger.vcxproj
05/19/2021  12:16 PM               301 ImageColorChanger.vcxproj.user
02/22/2021  06:38 PM           164,052 img1.jpeg
09/18/2020  09:41 PM           450,899 img2.jpg
01/07/2021  07:07 PM         3,494,765 img3.jpg
05/25/2021  12:46 AM        10,305,653 img4.jpg
03/10/2021  05:07 PM         1,286,845 img5.png
05/25/2021  01:18 AM            17,554 img6.png
05/25/2021  01:32 AM            22,653 img7.png
05/25/2021  01:45 AM             5,374 kernel.cu
05/19/2021  12:41 PM    <DIR>          x64
              10 archivos     15,753,330 bytes
               3 dirs  31,009,157,120 bytes libres

------------ Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img2.jpg
```

Image 4. Program running: step 1

- View on independent windows some images, such as:
    - The original input image
    - The image, but some dark color are changed into red tones
    - The image, but some black vertical lines were added, if the image has transparency capability there is a chess pattern instead of vertical lines.
    - The image, but if the input image was a png, all white colors will be transformed into transparent color. (This effect can only be seen on the exported image)

Program Running: Output

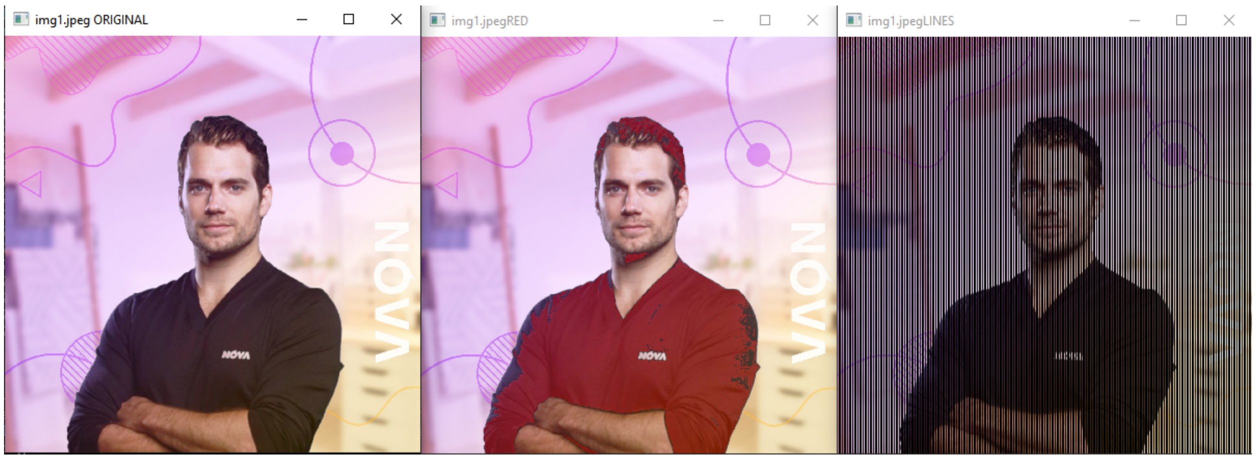
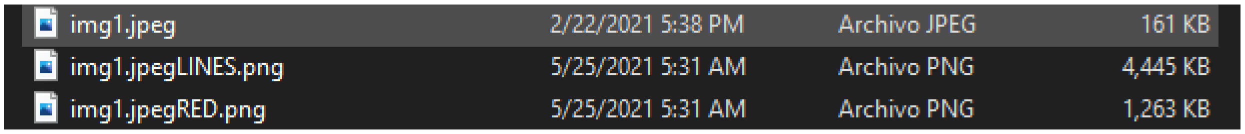


Image 5. Program running: image preview

- The program will write each of the previously listed modified images on the actual directory.

Program Running: Image Preview



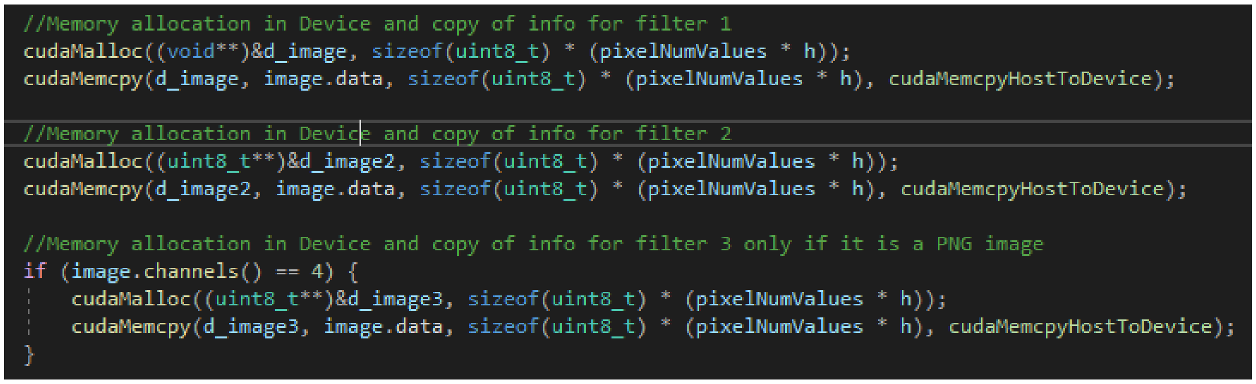
```
img1.jpeg              2/22/2021 5:38 PM    Archivo JPEG      161 KB
img1.jpegLINES.png     5/25/2021 5:31 AM    Archivo PNG     4,445 KB
img1.jpegRED.png       5/25/2021 5:31 AM    Archivo PNG     1,263 KB
```

Image 6. Program running: output

However, how CUDA is being used in the program? The first image is not that big in resolution, so it does not have a lot of pixels in it, but it can be used to show something visually. In CUDA C/C++, the programmer needs to take some things into consideration in order to use CUDA for image processing. The first of them is to think of images as matrixes or tables. Then, the code must open the image and consider the information inside each cell for memory purposes. The intention of this report is not to explain the code itself, but some concepts should be clarified.

In this case, opencv2 is being used to read image files. By default, the image will be opened taking in consideration the original channels. That means that if each pixel has values for Red, Green and Blue it means that it has three channels, but if it has values for Red, Green, Blue and Transparency it will be an image of four channels. This information can be obtained by writing *image.channels()*. Another important text that can be found in the code is *image.step*, which obtains the number of bytes each matrix row occupies. These information is important for assigning space in the memory of the GPU.

Memory Allocation in the GPU



```
//Memory allocation in Device and copy of info for filter 1
cudaMalloc((void**)&d_image, sizeof(uint8_t) * (pixelNumValues * h));
cudaMemcpy(d_image, image.data, sizeof(uint8_t) * (pixelNumValues * h), cudaMemcpyHostToDevice);

//Memory allocation in Device and copy of info for filter 2
cudaMalloc((uint8_t**)&d_image2, sizeof(uint8_t) * (pixelNumValues * h));
cudaMemcpy(d_image2, image.data, sizeof(uint8_t) * (pixelNumValues * h), cudaMemcpyHostToDevice);

//Memory allocation in Device and copy of info for filter 3 only if it is a PNG image
if (image.channels() == 4) {
    cudaMalloc((uint8_t**)&d_image3, sizeof(uint8_t) * (pixelNumValues * h));
    cudaMemcpy(d_image3, image.data, sizeof(uint8_t) * (pixelNumValues * h), cudaMemcpyHostToDevice);
}
```

Image 7. Memory allocation in the GPU

In image 7 there are some memory allocation in the GPU and the information of the image is being copied from the CPU to the GPU in order to perform the task there. The *pixelNumValues* was obtained by calculating the *image.step*. Therefore, the number of bytes horizontally is being multiplied by the height of the image, obviously measured in pixels.

After assigning the blocks and threads that are going to be used, which at the moment has been fixed to *height / 32* and *32* respectively, the first image modification is finally called.

Filter 1: Black to Red

The first filter applies the same concept presented in image 3. At the moment, with img1.jpeg of 1280 x 1280 pixels there are 32 threads per block. Which means a total of 1280 threads. That amount of threads is more than the number of threads that can be used optimally with CPU.

There are 40 blocks and each block has 32 threads.

The image should be linearized in order to work easier with the values. Nevertheless, it is important to consider that each pixel has three or four values previously calculated with the channels, then it needs to be specified in the code.

*int x = threadIdx.x + blockIdx.x * blockDim.x;*

*int y = threadIdx.y + blockIdx.y * blockDim.y;*

*int ind = (x + y * gridDim.x * blockDim.x) * Channels;*

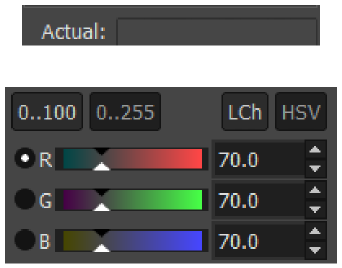
Now the *ind* has a unique value for each channel and each channel can be modified. The steps are simple. The channels are ordered in the following format:

- B - stands for blue
- G - stands for green

- R - stands for red

Then, if the values at *ind[0]*, *ind[1] and ind[2]* of each pixel is less or equal to 70 it is a similar color or darker than the one shown in image 8.

Dark Color



R 70.0
G 70.0
B 70.0

Image 8. Dark color

Then the values at the Red channel (*ind[2]*) are changed by adding to the current value 80. At the end it is exported as a png image and can be watched as a standalone file.

Filter 2: Black lines

The second filter applies the same concept presented in the previous filter, but the changed values are different. At the moment, with img2.jpeg of 4000 x 2667pixels there are 32 threads per block. Which means a total of 332,000 threads. Which is an insane amount of threads.

As the previous filter, the linearization of the pixels and threads is done. Therefore, if the pixel in the x axis is an even number the pixel will be colored in black. However, if the image has four channels, including the alpha one, the pattern will be different. The criteria for coloring it is if the pixel is multiple of three.

Filter 3: Deletes the background

For this last filter I wanted to experiment with the alpha channel, so this one finds white colors and every pixel that matches with the criteria is changed by setting the value of the alpha channel to 0.

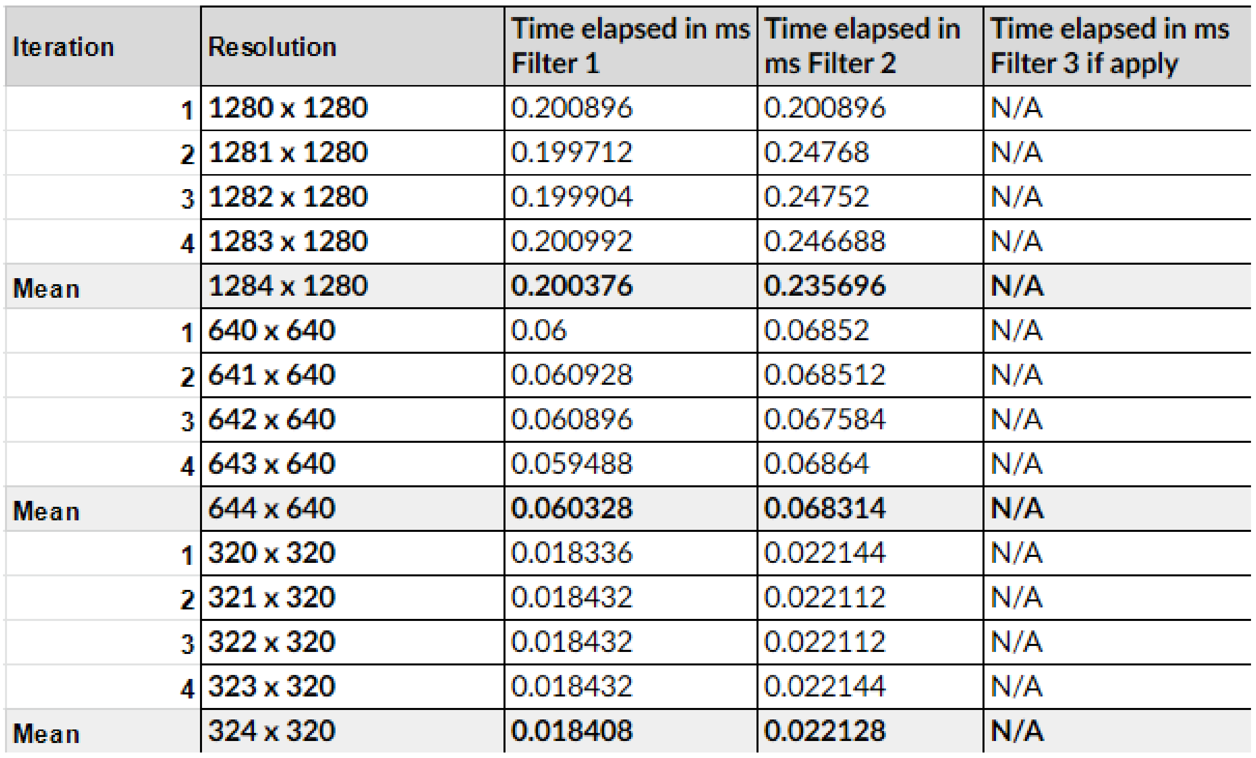
Tests

For the tests I will be using the same photography, but with different sizes. Different resolution. The test was executed four times each one. After that a mean of the times measured in milliseconds were calculated.

Image 1, group of testing 1

The images used were img1.jpeg, img1v2.jpeg and img1v3.jpeg. For screenshots of the results see appendix 2.

Group of Testing 1



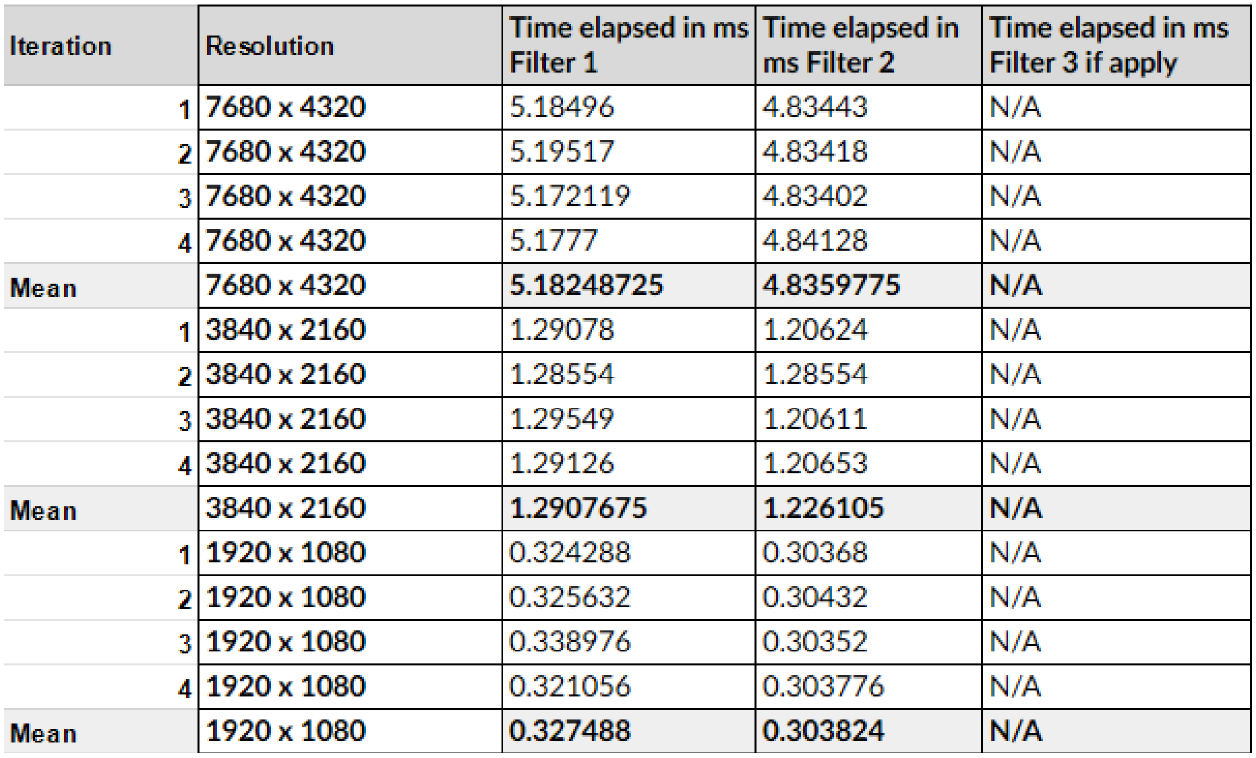
Iteration	Resolution	Time elapsed in ms Filter 1	Time elapsed in ms Filter 2	Time elapsed in ms Filter 3 if apply
1	1280 x 1280	0.200896	0.200896	N/A
2	1281 x 1280	0.199712	0.24768	N/A
3	1282 x 1280	0.199904	0.24752	N/A
4	1283 x 1280	0.200992	0.246688	N/A
Mean	1284 x 1280	**0.200376**	**0.235696**	**N/A**
1	640 x 640	0.06	0.06852	N/A
2	641 x 640	0.060928	0.068512	N/A
3	642 x 640	0.060896	0.067584	N/A
4	643 x 640	0.059488	0.06864	N/A
Mean	644 x 640	**0.060328**	**0.068314**	**N/A**
1	320 x 320	0.018336	0.022144	N/A
2	321 x 320	0.018432	0.022112	N/A
3	322 x 320	0.018432	0.022112	N/A
4	323 x 320	0.018432	0.022144	N/A
Mean	324 x 320	**0.018408**	**0.022128**	**N/A**

Table 1. Group of testing 1

Image 2, group of testing 2

The images used were img4.jpeg, img4v2.jpeg and img4v3.jpeg. For screenshots of the results see appendix 3.

Group of Testing 2



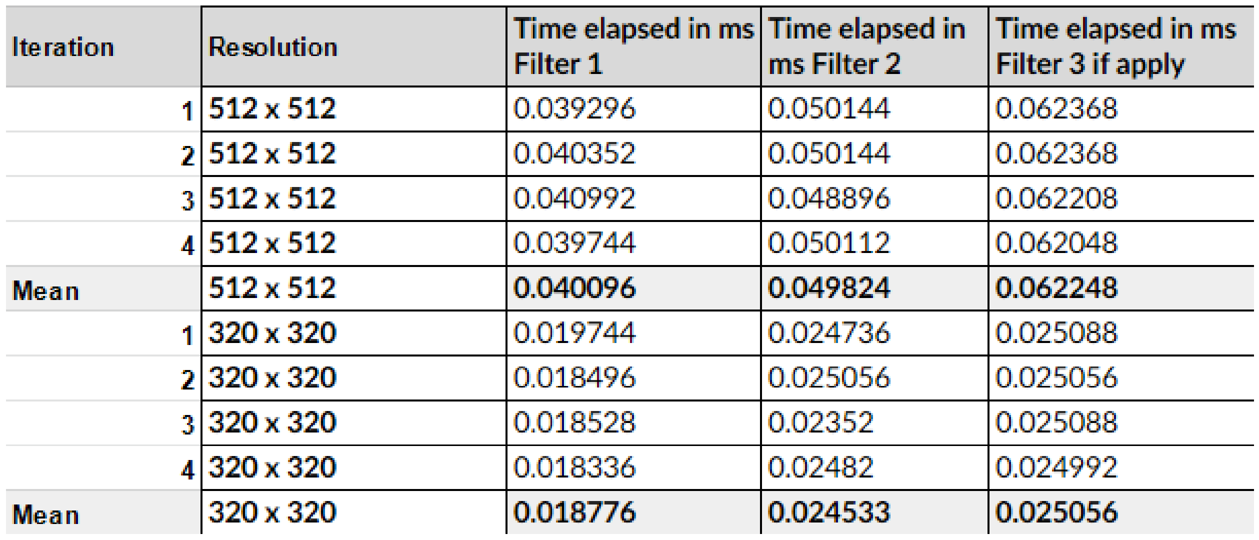
Iteration	Resolution	Time elapsed in ms Filter 1	Time elapsed in ms Filter 2	Time elapsed in ms Filter 3 if apply
1	**7680 x 4320**	5.18496	4.83443	N/A
2	**7680 x 4320**	5.19517	4.83418	N/A
3	**7680 x 4320**	5.172119	4.83402	N/A
4	**7680 x 4320**	5.1777	4.84128	N/A
Mean	**7680 x 4320**	**5.18248725**	**4.8359775**	**N/A**
1	**3840 x 2160**	1.29078	1.20624	N/A
2	**3840 x 2160**	1.28554	1.28554	N/A
3	**3840 x 2160**	1.29549	1.20611	N/A
4	**3840 x 2160**	1.29126	1.20653	N/A
Mean	**3840 x 2160**	**1.2907675**	**1.226105**	**N/A**
1	**1920 x 1080**	0.324288	0.30368	N/A
2	**1920 x 1080**	0.325632	0.30432	N/A
3	**1920 x 1080**	0.338976	0.30352	N/A
4	**1920 x 1080**	0.321056	0.303776	N/A
Mean	**1920 x 1080**	**0.327488**	**0.303824**	**N/A**

Table 2. Group of testing 2

Image 3, group of testing 3

The images used were img7.png and img7v2.png. For screenshots of the results see appendix 3.

Group of Testing 3



Iteration	Resolution	Time elapsed in ms Filter 1	Time elapsed in ms Filter 2	Time elapsed in ms Filter 3 if apply
1	512 x 512	0.039296	0.050144	0.062368
2	512 x 512	0.040352	0.050144	0.062368
3	512 x 512	0.040992	0.048896	0.062208
4	512 x 512	0.039744	0.050112	0.062048
Mean	512 x 512	**0.040096**	**0.049824**	**0.062248**
1	320 x 320	0.019744	0.024736	0.025088
2	320 x 320	0.018496	0.025056	0.025056
3	320 x 320	0.018528	0.02352	0.025088
4	320 x 320	0.018336	0.02482	0.024992
Mean	320 x 320	**0.018776**	**0.024533**	**0.025056**

Table 3. Group of testing 3

The most interesting part of these tests is when we camper with the first group of tests. There is an image with the same size (320 x 320 pixels) and the difference between them is almost imperceptible. However this can be because the images are too small, so the differences are to small. However, it can be seen that images with four channels are slower when image processing, since they have more information on each pixel.

Conclusions

Out in the world each of the consumers of computers wants the most powerful hardware for computational processing. Cryptomining is something popular and can be done using GPUs, Gamers want better and better performance on their games year by year. It was demonstrated that CUDA can be very useful and can have an awesome performance. Of course, during this text it was tested only with image processing, but certainly it demonstrated that handling with very large files, such as 8k photos, was possible to change aspects of the input image in about 5 seconds. At the moment, managing those values is not a standard in the industry, so using CUDA for image processing with regular full HD resolutions may be enough at the moment; also taking in consideration that the GPU used was released six years ago. Actual GPUs may be quicker and more powerful. It can also be seen that there are almost no differences in time between execution and execution of the process. This means that Cuda is reliable and constant with the results.

Parallelization is a very useful tool, and taking to the next level by using the amazing characteristics of the GPUs is perfect. For example, during the research for this work, I noticed that implementing Ray Tracing using CUDA is possible. For those who enjoy the most beautiful experience in games can play a little bit with that. I think everyone should learn CUDA parallelization. Additionally, I noticed that parallelizing with CUDA is way much quicker that any other parallelization method I had already used. However, each method has its pros and cons, it really depends on the task the programer wants to do and the language.

Sources

[1]J. Sanders and E. Kandrot, *CUDA by example*, 1st ed. Upper Saddle River, N.J.: Addison-Wesley, 2010, pp. 1, 7, 8, 9, 10, 11,.

[2] Ian Fooster. “7.1 Data Parallelism“. In URL:

https://www.mcs.anl.gov/~itf/dbpp/text/node83.html

[3] W. Richard Stevens, Stephen A. Rago. Advanced Programming in the UNIX Environment: Second Edition, 2015

[4] Nvidia. “¿QUÉ ES LA COMPUTACIÓN ACELERADA POR GPU?” In URL: https://www.nvidia.com/es-la/drivers/what-is-gpu-computing/

Extra Sources

[5] **Michael Romero, Rodrigo Urra. “CUDA Programming” In URL:** http://cuda.ce.rit.edu/about/about.htm

[6] Mark Harris. “How to implement Performance Metrics in CUDA C?C++” In URL: https://developer.nvidia.com/blog/how-implement-performance-metrics-cuda-cc/

Appendixes

Appendix 1: Dependencies of the project

SO: Windows 10

IDE: Visual Studio Community 2019

GPU used

```
Detected 1 CUDA Capable device(s)
          Device 0: "NVIDIA GeForce GTX 960M"

          CUDA Driver Version / Runtime Version        11.3 / 11.3

          CUDA Capability Major/Minor version number:          5.0

          Total amount of global memory:                 2048 MBytes (2147483648 bytes)

          (005) Multiprocessors, (128) CUDA Cores/MP:          640 CUDA Cores                              GPU
Max Clock rate:                        1176 MHz (1.18 GHz)

          Memory Clock rate:                        2505 Mhz

          Memory Bus Width:                        128-bit                                          L2

          Cache Size:                        2097152 bytes

          Maximum Texture Dimension Size (x,y,z)        1D=(65536), 2D=(65536, 65536), 3D=(4096, 4096, 4096)


          Maximum Layered 1D Texture Size, (num) layers  1D=(16384), 2048 layers
Maximum Layered 2D Texture Size, (num) layers  2D=(16384, 16384), 2048 layers

          Total amount of constant memory:               65536 bytes

          Total amount of shared memory per block:     49152 bytes

          Total shared memory per multiprocessor:       65536 bytes

          Total number of registers available per block: 65536

          Warp size:                        32

          Maximum number of threads per multiprocessor:  2048

          Maximum number of threads per block:         1024

          Max dimension size of a thread block (x,y,z): (1024, 1024, 64)

          Max dimension size of a grid size      (x,y,z): (2147483647, 65535, 65535)
```

```
Maximum memory pitch:                          2147483647 bytes

Texture alignment:                   512 bytes

Concurrent copy and kernel execution:          Yes with 4 copy engine(s)

Run time limit on kernels:           Yes

Integrated GPU sharing Host Memory:            No

Support host page-locked memory mapping:                 Yes

Alignment requirement for Surfaces:            Yes

Device has ECC support:              Disabled

CUDA Device Driver Mode (TCC or WDDM):                   WDDM (Windows Display Driver Model)
Device supports Unified Addressing (UVA):      Yes

Device supports Managed Memory:                Yes

Device supports Compute Preemption:            No

Supports Cooperative Kernel Launch:            No

Supports MultiDevice Co-op Kernel Launch:                No

Device PCI Domain ID / Bus ID / location ID:   0 / 1 / 0
```

Libraries

Opencv 2 version 4.5.2 was used.

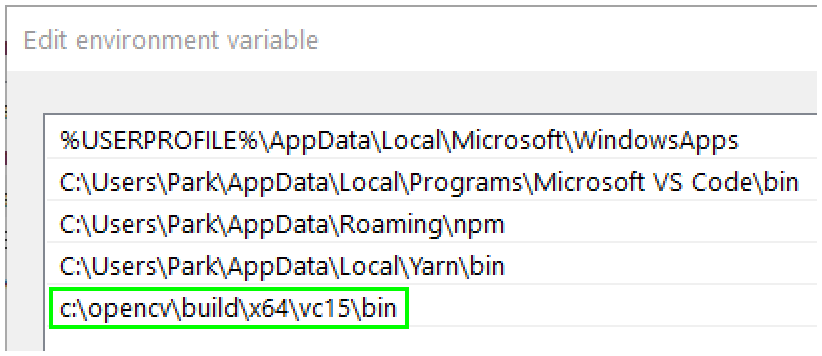
Can be downloaded from: https://opencv.org/releases/

Step 1

The first step for installing opencv2 on VisualSrudio 2019 is to extract the downloaded files. I used the C:// folder. By running the .exe file downloaded.

Step 2

Add the path to the Windows configuration.



```
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\Park\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\Park\AppData\Roaming\npm
C:\Users\Park\AppData\Local\Yarn\bin
c:\opencv\build\x64\vc15\bin
```

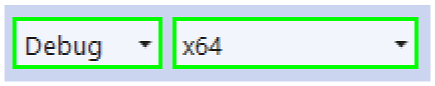
Add to path

Step 3

Open a new CUDA project or, in this case, open the project.

Step 4

If there is any problem or you are creating a new project you should set the platform as the following.

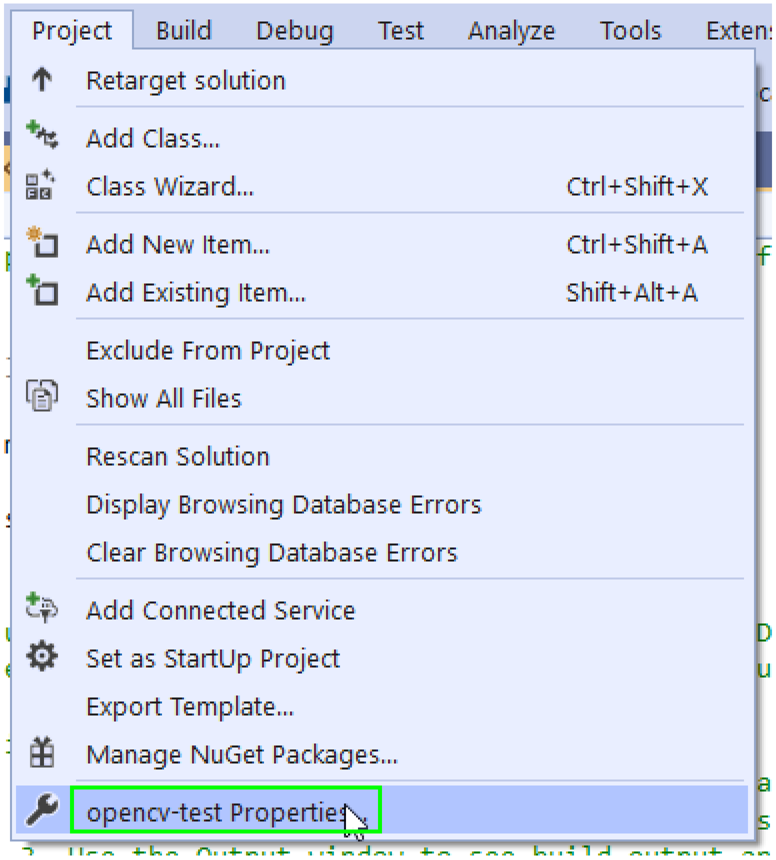


Debug | x64

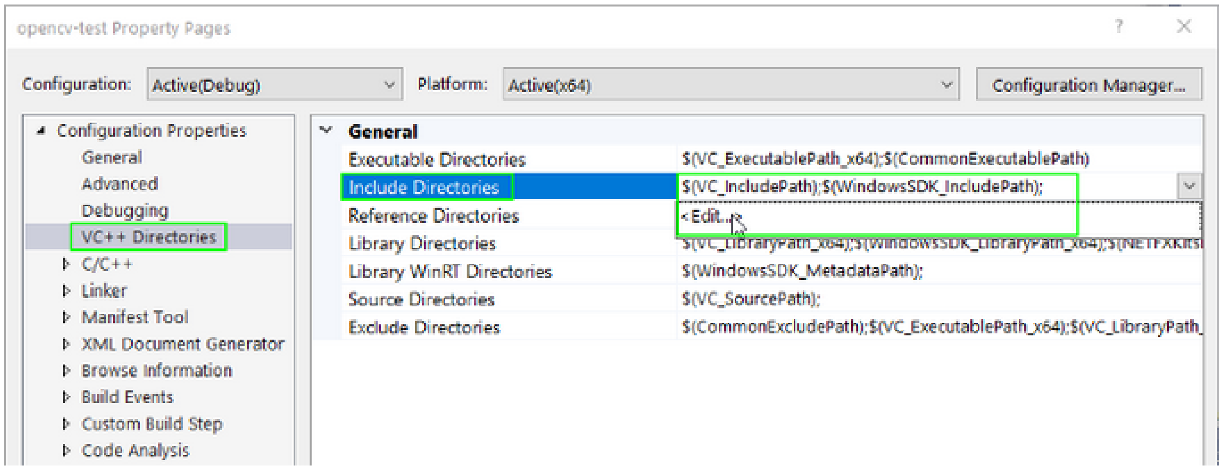
Set platform target to x64

Then, you should add the dependencies.

Directly click on the project button in the menu and then click on properties.



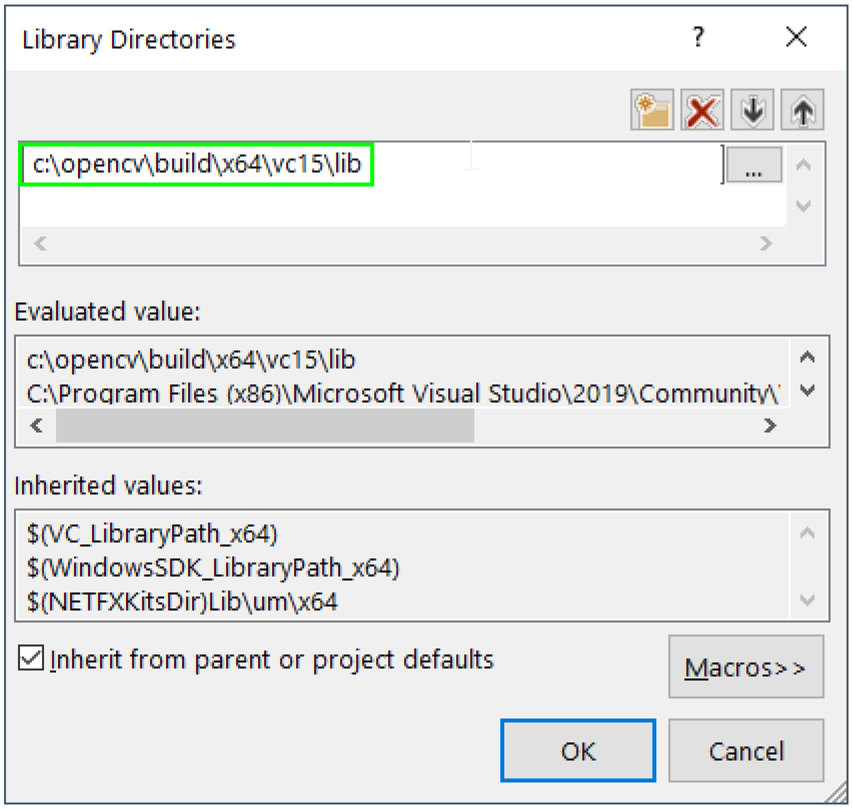
On the following window click on the Include Directories > Edit… option to add the following line. c://opencv/build/include



Executable Directories	$(VC_ExecutablePath_x64);$(CommonExecutablePath)
Include Directories	$(VC_IncludePath);$(WindowsSDK_IncludePath);
Reference Directories	
Library Directories	$(VC_LibraryPath_x64);$(WindowsSDK_LibraryPath_x64);$(NETFXKitsl
Library WinRT Directories	$(WindowsSDK_MetadataPath);
Source Directories	$(VC_SourcePath);
Exclude Directories	$(CommonExcludePath);$(VC_ExecutablePath_x64);$(VC_LibraryPath_

On Library directories you should also edit the entry. You should add the following line.

c:/opencv/build/x64/vc15/lib



Library Directories

c:\opencv\build\x64\vc15\lib

Evaluated value:

```
c:\opencv\build\x64\vc15\lib
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\
```

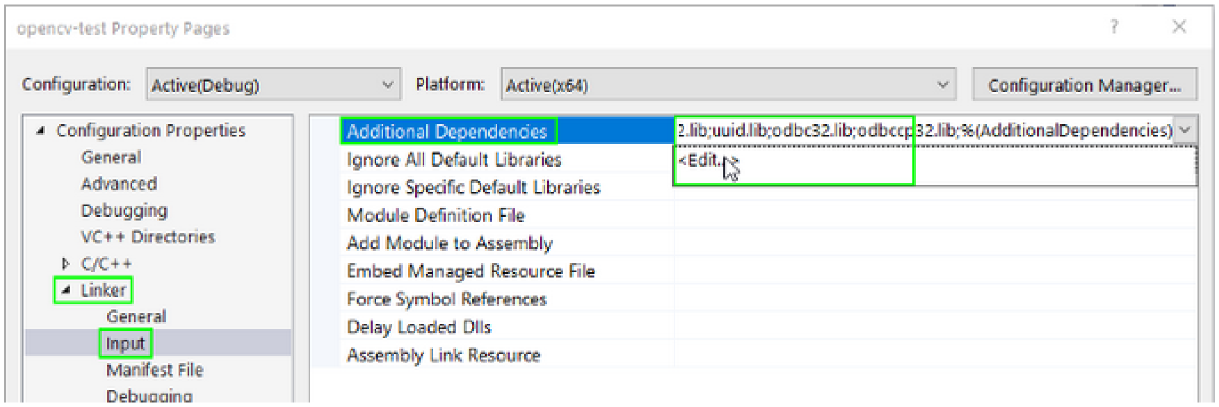
Inherited values:

```
$(VC_LibraryPath_x64)
$(WindowsSDK_LibraryPath_x64)
$(NETFXKitsDir)Lib\um\x64
```

☑ Inherit from parent or project defaults

Add **c:\opencv\build\x64\vc15\lib**

Step 5



opencv-test Property Pages

Configuration: Active(Debug) Platform: Active(x64)

Additional Dependencies	2.lib;uuid.lib;odbc32.lib;odbccp32.lib;%(AdditionalDependencies)
Ignore All Default Libraries	<Edit...>
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Linker — Input — Additional Dependencies

On Linker > Input > Additional Dependencies > Edit… write the following…

opencv_world452d.lib

opencv_world452.lib

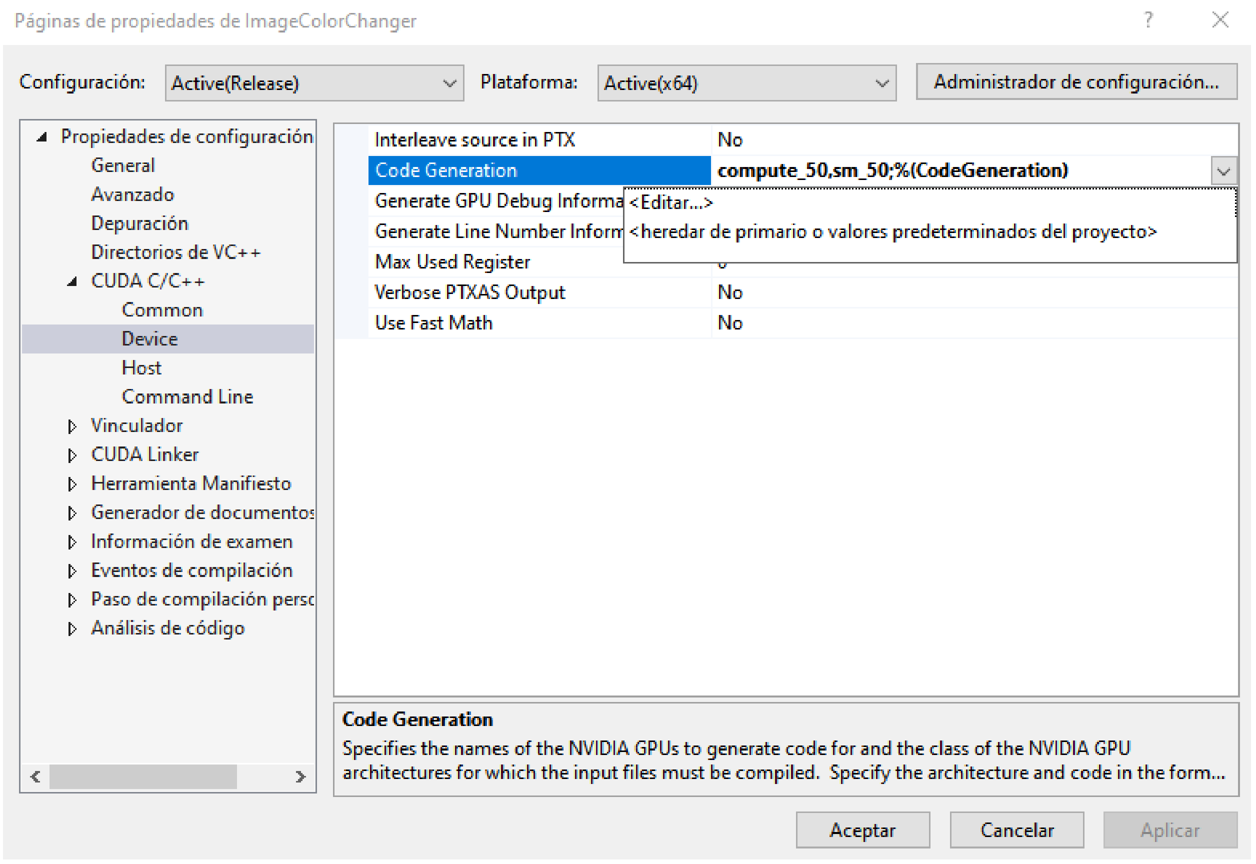
Step 6

If you are using an old GPU add the following into CUDA C/C++ > Device > Code Genereation > Edit…

compute_35,sm_35

compute_37,sm_37

compute_50,sm_50


Páginas de propiedades de ImageColorChanger

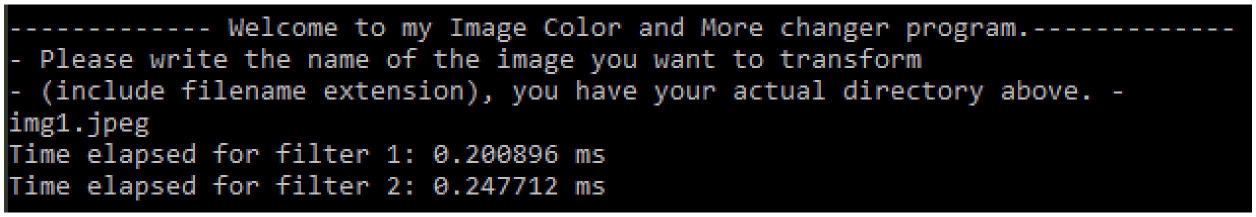
Configuración: Active(Release) Plataforma: Active(x64) Administrador de configuración...

Interleave source in PTX	No
Code Generation	compute_50,sm_50;%(CodeGeneration)
Generate GPU Debug Informa	<Editar...>
Generate Line Number Inform	<heredar de primario o valores predeterminados del proyecto>
Max Used Register	0
Verbose PTXAS Output	No
Use Fast Math	No

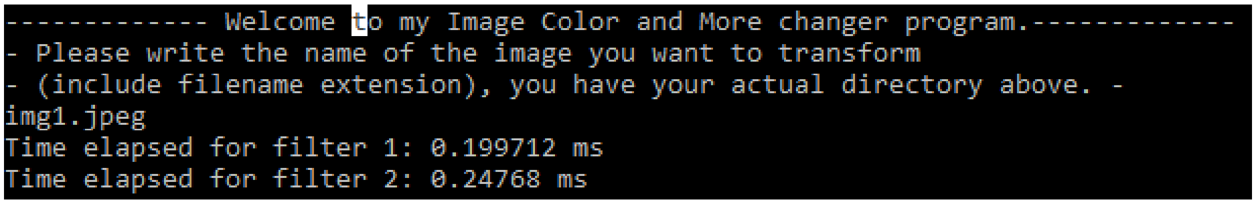
Code Generation
Specifies the names of the NVIDIA GPUs to generate code for and the class of the NVIDIA GPU architectures for which the input files must be compiled. Specify the architecture and code in the form...

Appendix 2: Group of testing 1

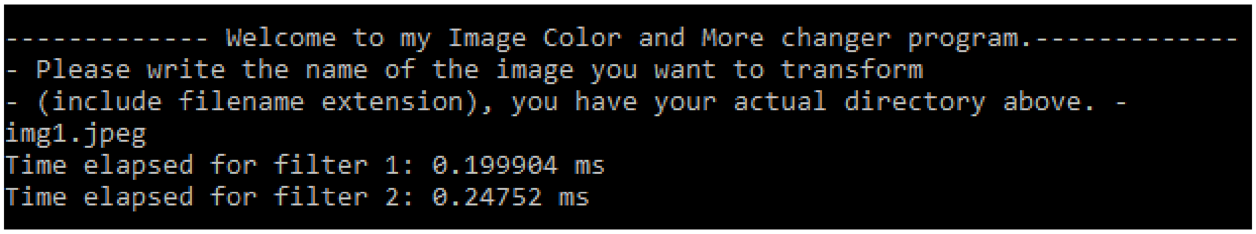
1280 x 1280



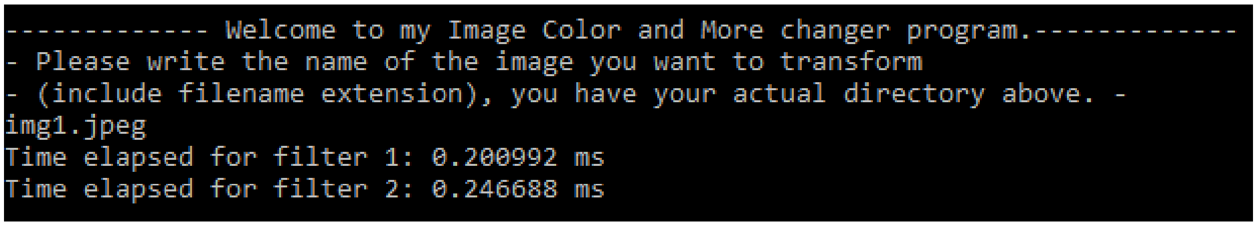
```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1.jpeg
Time elapsed for filter 1: 0.200896 ms
Time elapsed for filter 2: 0.247712 ms
```



```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1.jpeg
Time elapsed for filter 1: 0.199712 ms
Time elapsed for filter 2: 0.24768 ms
```

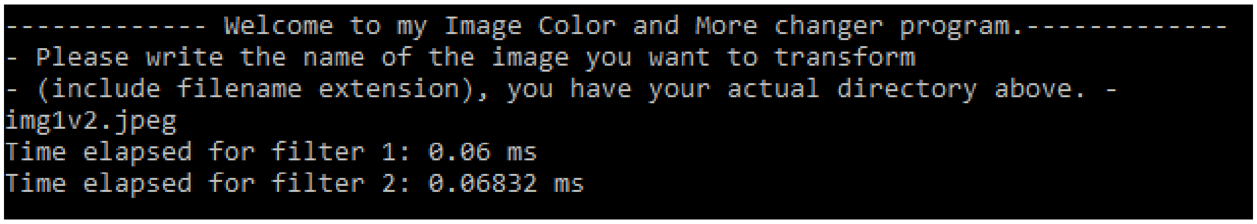


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1.jpeg
Time elapsed for filter 1: 0.199904 ms
Time elapsed for filter 2: 0.24752 ms
```

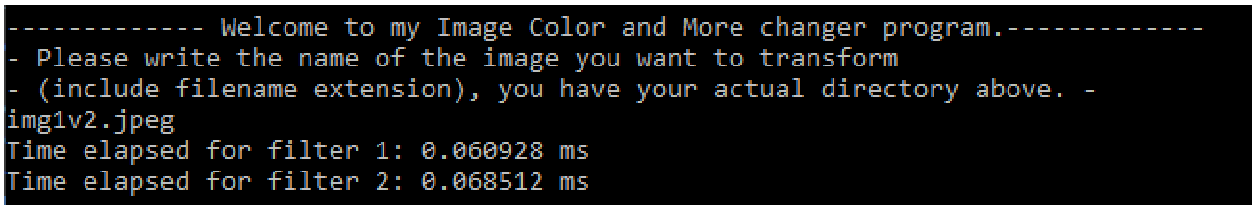


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1.jpeg
Time elapsed for filter 1: 0.200992 ms
Time elapsed for filter 2: 0.246688 ms
```

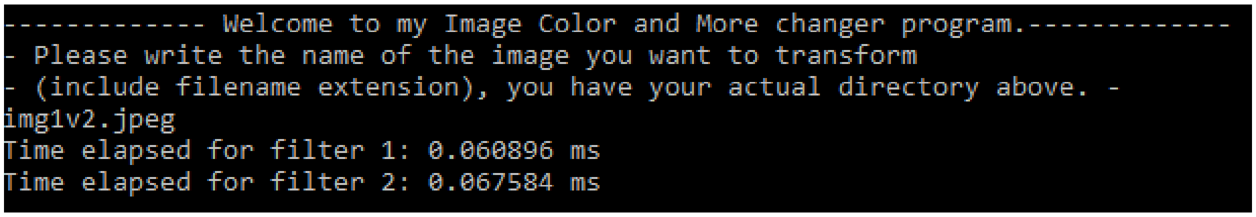
640 x 640

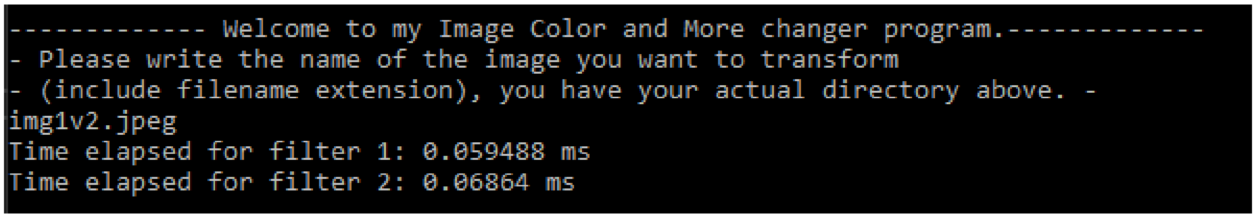


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1v2.jpeg
Time elapsed for filter 1: 0.06 ms
Time elapsed for filter 2: 0.06832 ms
```

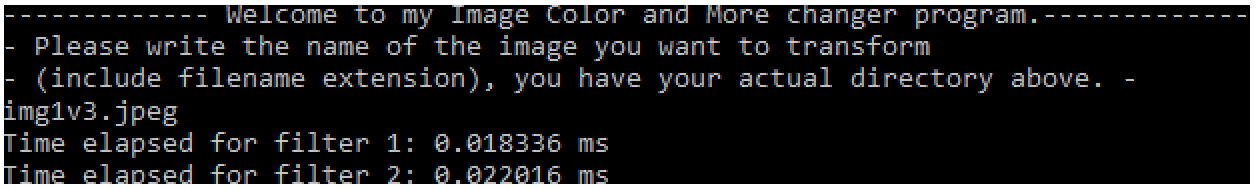


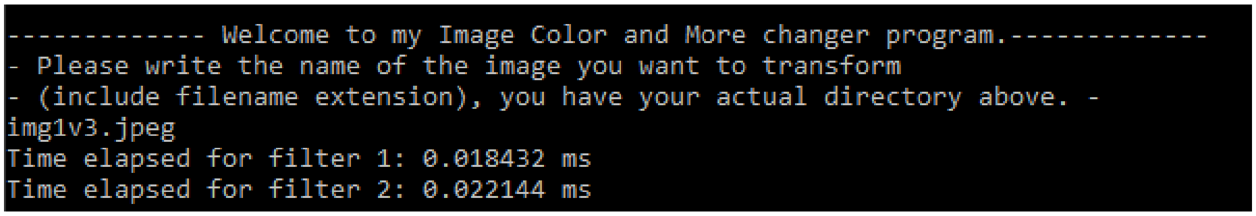
```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1v2.jpeg
Time elapsed for filter 1: 0.060928 ms
Time elapsed for filter 2: 0.068512 ms
```

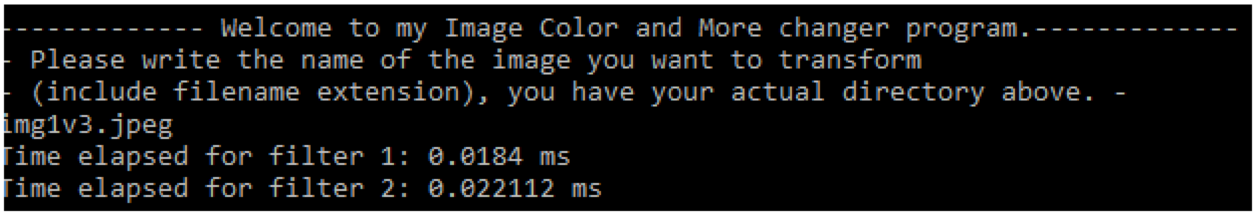

```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1v2.jpeg
Time elapsed for filter 1: 0.060896 ms
Time elapsed for filter 2: 0.067584 ms
```

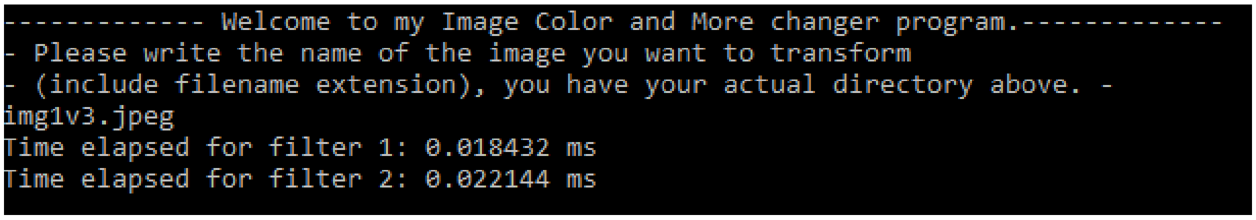

```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1v2.jpeg
Time elapsed for filter 1: 0.059488 ms
Time elapsed for filter 2: 0.06864 ms
```

320 x 320


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1v3.jpeg
Time elapsed for filter 1: 0.018336 ms
Time elapsed for filter 2: 0.022016 ms
```

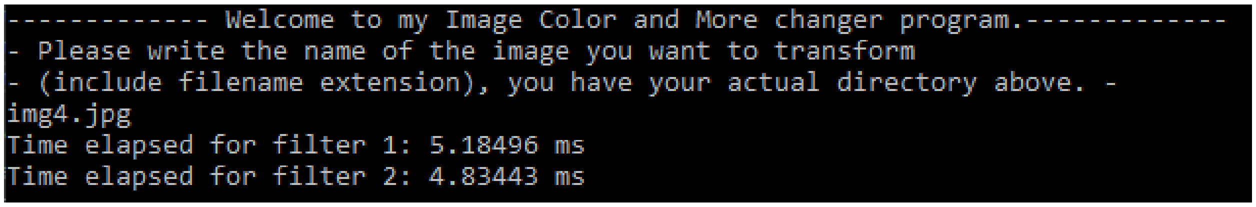

```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1v3.jpeg
Time elapsed for filter 1: 0.018432 ms
Time elapsed for filter 2: 0.022144 ms
```


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1v3.jpeg
Time elapsed for filter 1: 0.0184 ms
Time elapsed for filter 2: 0.022112 ms
```

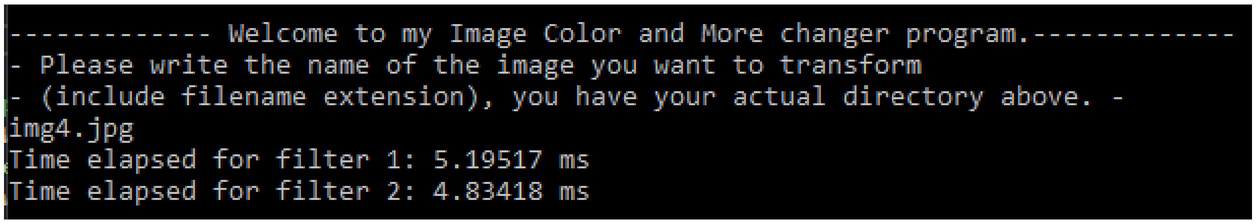

```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img1v3.jpeg
Time elapsed for filter 1: 0.018432 ms
Time elapsed for filter 2: 0.022144 ms
```

Appendix 3: Group of testing 2

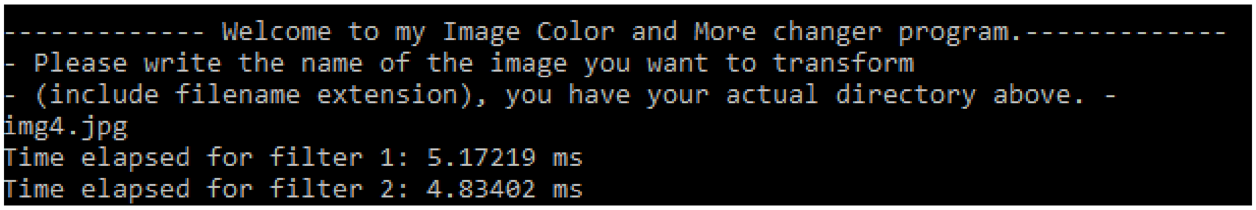
7680 x 4320



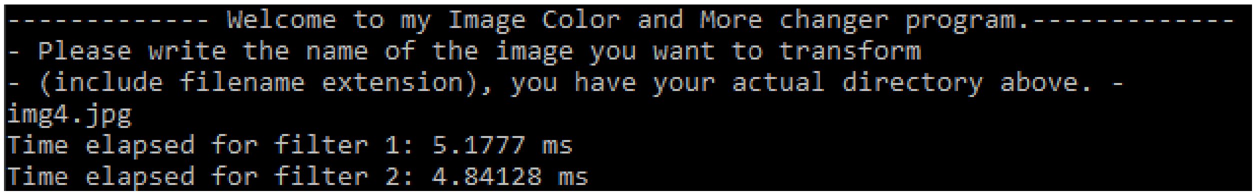
```
------------ Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4.jpg
Time elapsed for filter 1: 5.18496 ms
Time elapsed for filter 2: 4.83443 ms
```



```
------------ Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4.jpg
Time elapsed for filter 1: 5.19517 ms
Time elapsed for filter 2: 4.83418 ms
```

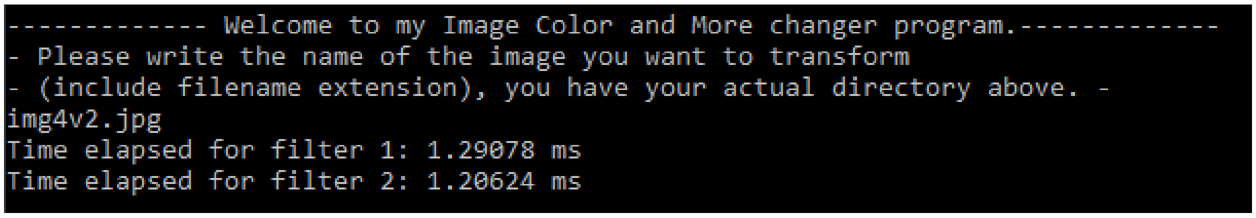


```
------------ Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4.jpg
Time elapsed for filter 1: 5.17219 ms
Time elapsed for filter 2: 4.83402 ms
```

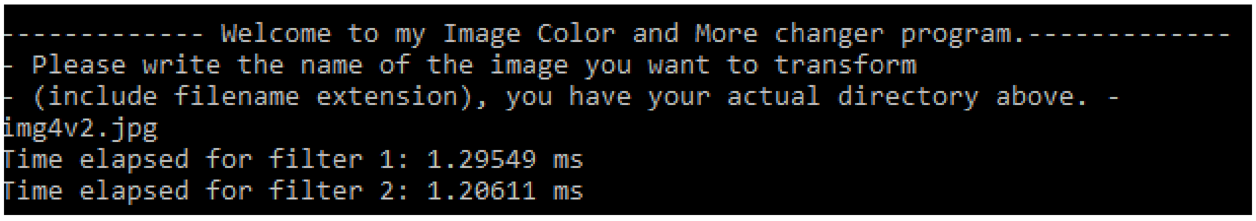


```
------------ Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4.jpg
Time elapsed for filter 1: 5.1777 ms
Time elapsed for filter 2: 4.84128 ms
```

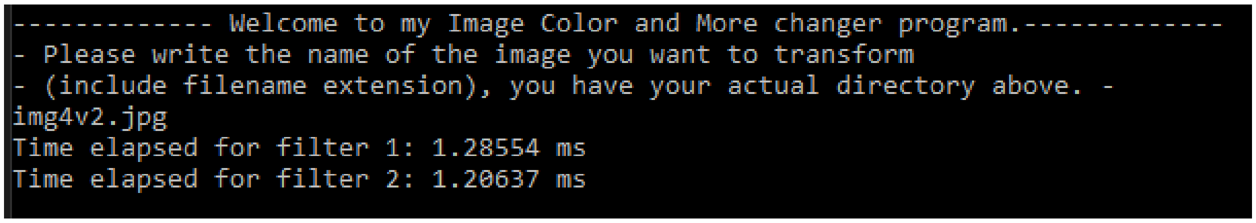
3840 x 2160



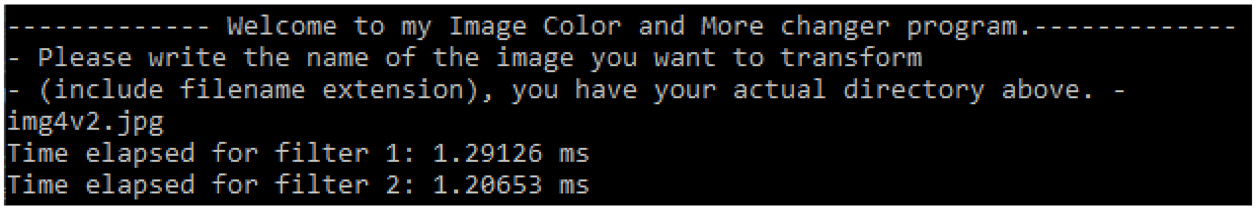
```
------------ Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4v2.jpg
Time elapsed for filter 1: 1.29078 ms
Time elapsed for filter 2: 1.20624 ms
```



```
------------ Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4v2.jpg
Time elapsed for filter 1: 1.29549 ms
Time elapsed for filter 2: 1.20611 ms
```

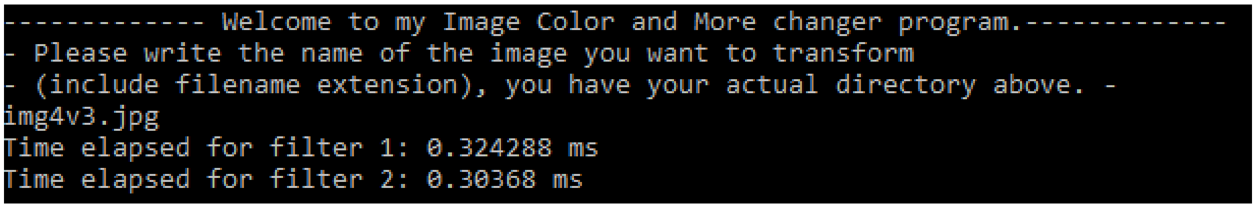


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4v2.jpg
Time elapsed for filter 1: 1.28554 ms
Time elapsed for filter 2: 1.20637 ms
```

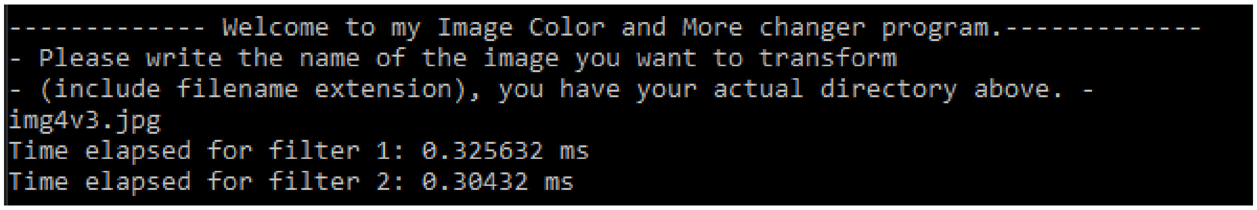


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4v2.jpg
Time elapsed for filter 1: 1.29126 ms
Time elapsed for filter 2: 1.20653 ms
```

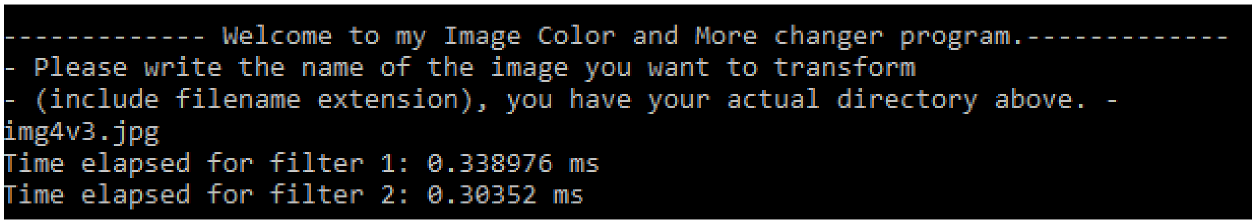
1920 x 1080



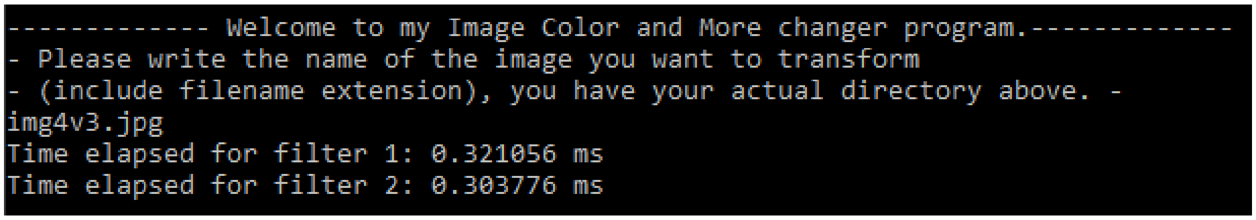
```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4v3.jpg
Time elapsed for filter 1: 0.324288 ms
Time elapsed for filter 2: 0.30368 ms
```



```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4v3.jpg
Time elapsed for filter 1: 0.325632 ms
Time elapsed for filter 2: 0.30432 ms
```



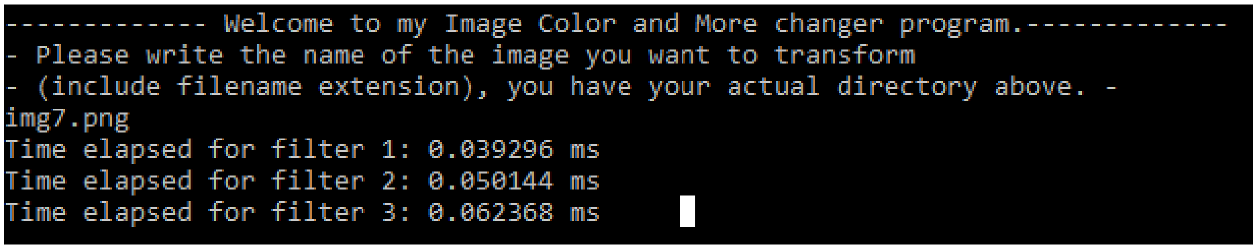
```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4v3.jpg
Time elapsed for filter 1: 0.338976 ms
Time elapsed for filter 2: 0.30352 ms
```



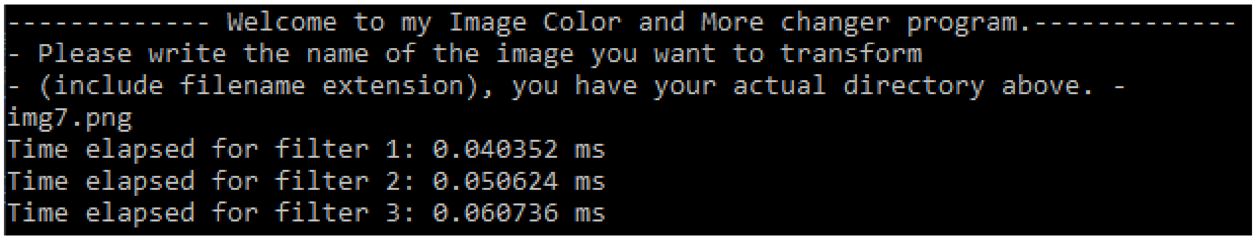
```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img4v3.jpg
Time elapsed for filter 1: 0.321056 ms
Time elapsed for filter 2: 0.303776 ms
```

Appendix 4: Group of testing 3

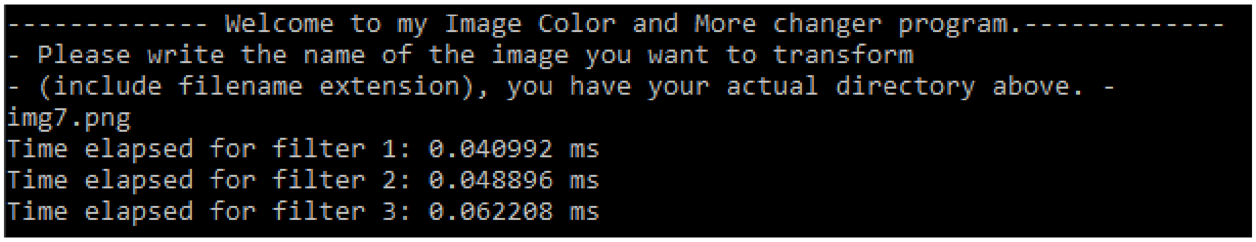
512 x 512



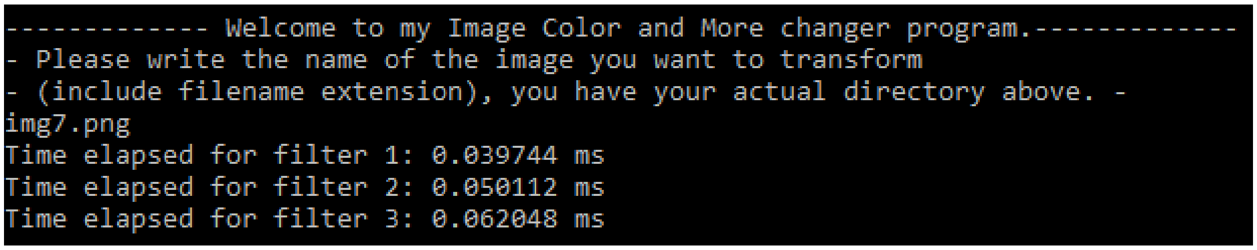
```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img7.png
Time elapsed for filter 1: 0.039296 ms
Time elapsed for filter 2: 0.050144 ms
Time elapsed for filter 3: 0.062368 ms
```



```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img7.png
Time elapsed for filter 1: 0.040352 ms
Time elapsed for filter 2: 0.050624 ms
Time elapsed for filter 3: 0.060736 ms
```

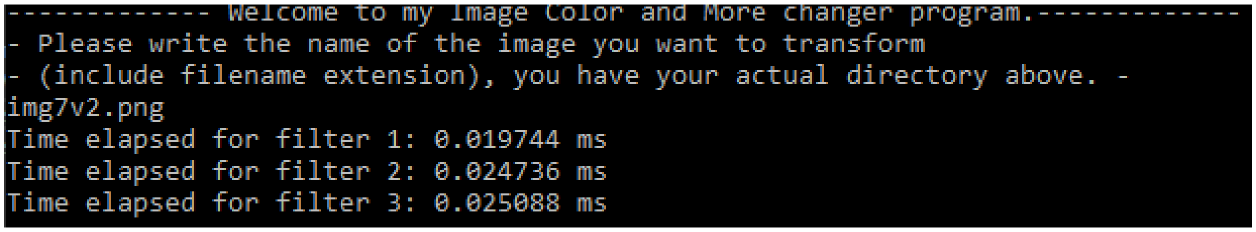


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img7.png
Time elapsed for filter 1: 0.040992 ms
Time elapsed for filter 2: 0.048896 ms
Time elapsed for filter 3: 0.062208 ms
```

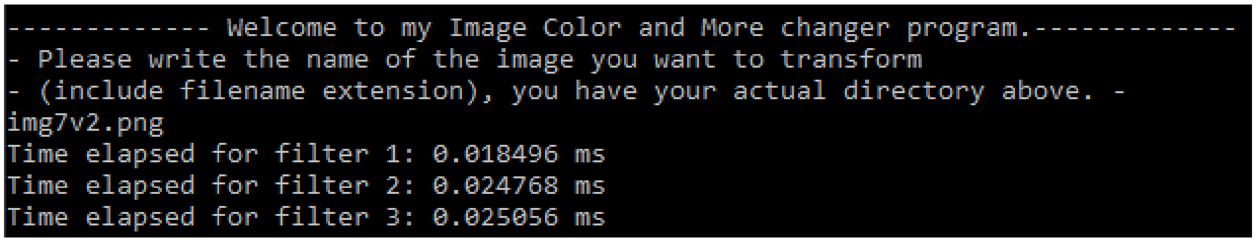


```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img7.png
Time elapsed for filter 1: 0.039744 ms
Time elapsed for filter 2: 0.050112 ms
Time elapsed for filter 3: 0.062048 ms
```

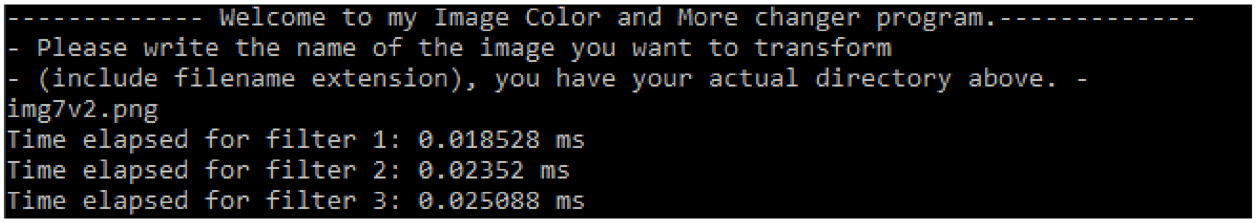
320 x 320



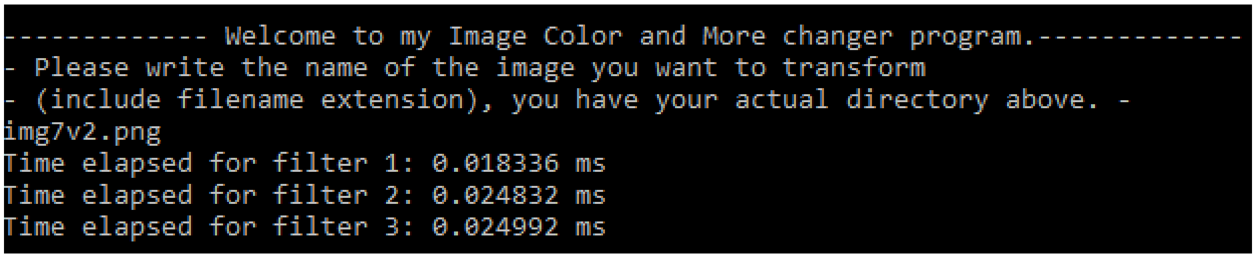
```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img7v2.png
Time elapsed for filter 1: 0.019744 ms
Time elapsed for filter 2: 0.024736 ms
Time elapsed for filter 3: 0.025088 ms
```



```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img7v2.png
Time elapsed for filter 1: 0.018496 ms
Time elapsed for filter 2: 0.024768 ms
Time elapsed for filter 3: 0.025056 ms
```



```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img7v2.png
Time elapsed for filter 1: 0.018528 ms
Time elapsed for filter 2: 0.02352 ms
Time elapsed for filter 3: 0.025088 ms
```



```
------------- Welcome to my Image Color and More changer program.-------------
- Please write the name of the image you want to transform
- (include filename extension), you have your actual directory above. -
img7v2.png
Time elapsed for filter 1: 0.018336 ms
Time elapsed for filter 2: 0.024832 ms
Time elapsed for filter 3: 0.024992 ms
```