Implementation of Morphological Filters

Bikash Tripathy - 3440874
Sumit Sukhadev Jadhav - 3438857
Institut for Parallele und Verteile Systeme, IPVS, University of Stuttgart

1. Abstract

Originally, set theory is being considered as the context for deriving 'Morphological Operation'. Morphological filters are one of the great application of it on images. This report gives a detailed information on effects of applying morphological filters on images. The main operation includes *erosion*, *dilation, opening, and closing*. The morphological filters are morphed using the algorithms respectively to generate the desired effect on the images. Erosion and dilation filters also known popularly as minimum and maximum filters. This report also takes a deeper insight into the Gaussian filter and operation on images. Finally, all of this filters being performed on a CPU implementation and also on the GPU, where we expect GPU process to be faster with equivalent output of the morphed images.

2. Introduction

Mathematical morphology is a nonlinear image processing methodology. Mathematical morphology is a set-theory-based method for analyzing geometrical structures or images. Morphology for a binary image has been studied first and foremost. It is now possible to use it on grayscale images. Shape, orientation, and smoothness can be changed in morphological image processing by using structuring elements and morphological operators. The structuring element is a binary matrix made up of 1s and 0s. Shape analysis and edge detection were the primary applications of mathematical morphology. Objects can be thinned and thickened using the morphological operation. The output's sensitivity is determined by the size of the structuring element. Dilation and erosion are the two fundamental morphological operations. These fundamental operations are used in a specific order to perform tasks like opening and closing. Overall, given that they are based on simple mathematical concepts, the accomplishments of morphological operations are impressive.

3. Morphological Operations

Dilation –

One of the morphological operations that adds pixels to the boundary of an objects in an image is dilation. The structuring element, also known as a kernel, is used to expand the shape of an object in an image.

Structure element superimposes on each pixel of an image during dilation operation so that the origin of the structuring element coincides with the pixel of an image. The output pixel takes the maximum value of all pixels in the neighborhood within the kernel's dimension.

Dilation operation makes the object in an image more visible and fills in small holes in the objects.

The dilation is denoted as follows.

$(A \oplus B)$ where A is an image and B is a structuring element.

Erosion –

The erosion operation is exact opposite operation of the dilation. It shrinks the boundary of an objects in the image. Similarly, to the dilation operation, the structuring element superimposes on each pixel of an image such that the origin of the structuring element coincides with the pixel of the image. However, the output pixel takes the minimum value of all pixels in the neighborhood within the kernel's dimension. Erosion operation eliminates the holes and broadens the gap within the object in an image.

An erosion is denoted as follows.

$(A \ominus B)$ where A is an image and B is a structuring element.

Opening –

The opening operation is derived from the above-mentioned fundamental operation. In the opening operation, an erosion operation is performed followed by a dilation operation. The same structuring element is used for both erosion and dilation. This morphological operation removes the small objects in the image while preserving the shape and size of the larger objects.

The opening operation is denoted as follows.

$(A \circ B=(A \ominus B) \oplus B)$ where A is an image and B is a structuring element.

Closing –

The closing operation is also derived from the fundamental operation. In closing, first dilation is performed and then it erodes the image. The same structuring element is used for both dilation and erosion. This morphological operation fills the small holes while preserving the shape and size of the objects in the image.

The closing operation is denoted as follows.

$(A \bullet B=(A \oplus B) \ominus B)$ where A is an image and B is a structuring element

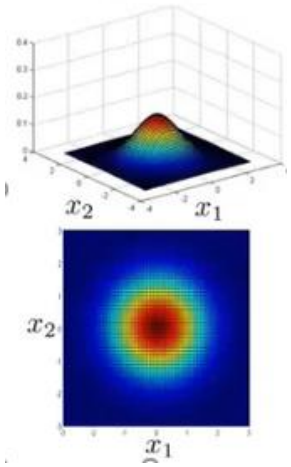
4. Gaussian Blur (Gaussian Filter)

This is one of the most popular image filtering techniques. A Gaussian filter is a liner filter which usually used to blur the image or so as to reduce the noise of the image. The Gaussian blur is also known as the Gaussian smoothing. In a layman sense it can be said that it is equivalent to see a picture through a translucent screen.

Mathematically, to apply this Gaussian filter, it is highly necessary to understand the Gaussian function which helps produces the main effect on the images. The below equation gives a brief idea for two dimensional case, where we can operate pixel wise in case to 2D images. Where x, y presents the directions in the X axis and Y axis respectively. And sigma represents the standard deviation of the Gaussian distribution.

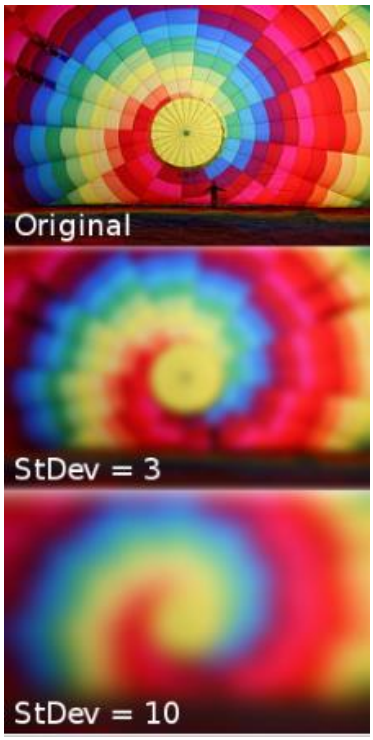
$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

The values from the distribution is calculated and stored. This acts as convolutional matrix for the images we want to operate on. Which helps in giving out the image which is blurred.



The above picture represents the probability distribution of a multivariate Gaussian function, where mu(mean) of both x and y are zeros.

Image Source - https://regenerativetoday.com/univariate-and-bivariate-gaussian-distribution-clear-explanation-with-visuals/



This picture depicts the Gaussian Blur and effect of increasing sigma.

Image Source - https://en.wikipedia.org/wiki/Gaussian_blur

5. Implementation

5.1 Implementation on CPU

The code is implemented in C++, where we have implemented our morphological filters. The basic morphological filters (Opening, closing, erosion and dilation) are implemented based on a structuring element. Which is defined 3X3 matrix. This matrix decides the final output processed imaged/ filtered image when processing is done as per the algorithm.
The structuring element, is iterated over each of the pixel, and values are calculated as per the algorithm (min or max). An output matrix is defined which stores the output in an output buffer. This total execution time is then taken into account for the profiling to check the performance.

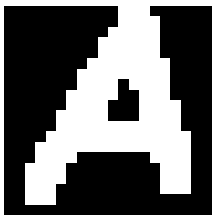
5.2 Implementation on GPU

OpenCL is a framework for writing the logic for multiplatform, GPU, and processor or hardware accelerators. It also provides independent platform for parallel computing. This framework supports multi-threaded heterogeneous computing workloads with tight cooperation between

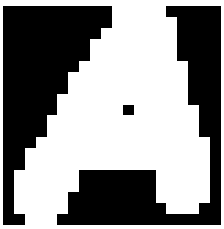
CPU and GPU. Interestingly, it provides the concept of work groups, which is the collection of related work items., that execute on a single computing unit.

6. Results and Analysis

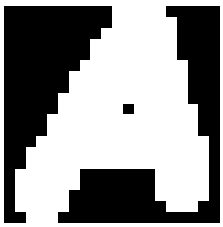
Input Image



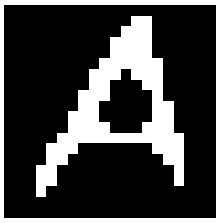
Results of Dilation (CPU)



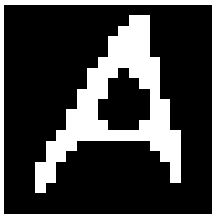
Results of Dilation (GPU)



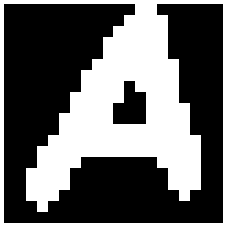
Results of Erosion (CPU)



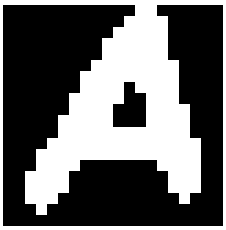
Results of Erosion (GPU)



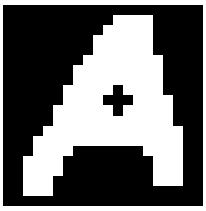
Results of Opening (CPU)



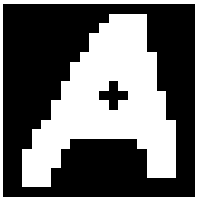
Results of Opening (GPU)



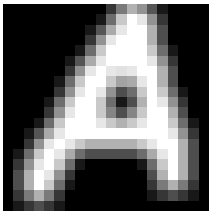
Results of Closing (CPU)



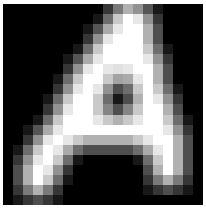
Results of Closing (GPU)



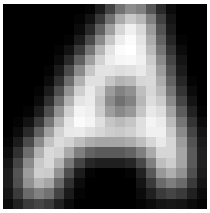
Results of Gaussian Filter(3X3) (CPU)



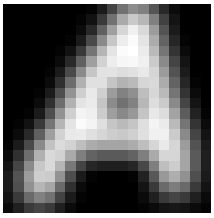
Results of Gaussian Filter(3X3) (GPU)



Results of Gaussian Filter(5X5) (CPU)



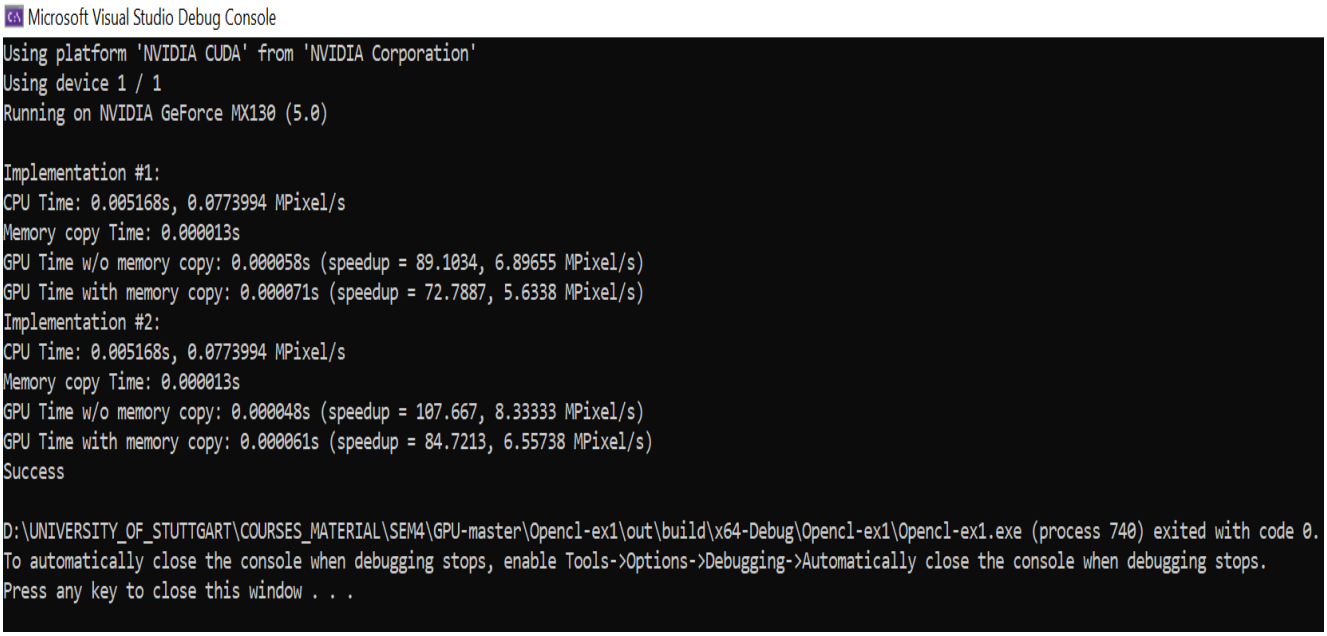
Results of Gaussian Filter(5X5) (GPU)



7. Performance and Evaluation

The main objective of this task is to achieve the filter application on images. However, by performing the task on CPU and GPU makes a significant change on the execution time. This helps in evaluating the performance of the implementation on the host and the GPU. By performing this we have measured the speedup. In general, the multicore architecture makes GPU faster execution to the implementations and makes it obvious for a better performance. And the results of ours also conveys the same very clearly.

This clearly achieves the two main objective of the task given, one where we are able to produce desired results for filters and achieve a better execution speed on the GPU with same effect of filters on the output. The following figure shows the amount of time required for the CPU and GPU to perform the operations. Implementation 1 shows the elapsed time when the 3x3 kernel is used for the Gaussian filter operation, whereas implementation 2 shows the elapsed time when the 5x5 kernel is used for the Gaussian filter operation.



8. Project Execution

Sl No	Work Packages	Ownership
1	Initial Project Plan	Sumit Jadhav, Bikash Tripathy
2	Concept mining and Algorithm Understanding	Sumit Jadhav, Bikash Tripathy
3	CPU Implementation Erosion	Sumit Jadhav ,Bikash Tripathy
4	CPU Implementation Dilation	Sumit Jadhav ,Bikash Tripathy
5	CPU Implementation Opening	Sumit Jadhav
6	CPU Implementation Closing	Sumit Jadhav
7	CPU Implementation Gaussian Filter	Sumit Jadhav, Bikash Tripathy
8	GPU Implementation Erosion	Sumit Jadhav, Bikash Tripathy
9	GPU Implementation Dilation	Sumit Jadhav, Bikash Tripathy
10	GPU Implementation Opening	Sumit Jadhav
11	GPU Implementation Closing	Sumit Jadhav, Bikash Tripathy
12	GPU Implementation Gaussian Filter	Bikash Tripathy
13	Testing & Debugging	Sumit Jadhav, Bikash Tripathy
14	Project Report Compilation	Sumit Jadhav, Bikash Tripathy

9. Reference

1.https://en.wikipedia.org/wiki/Gaussian_blur
2.http://www.theobjects.com/dragonfly/dfhelp/40/Content/05_Image%20Processing/Morphology%20Filters.htm
3.https://towardsdatascience.com/image-processing-class-egbe443-6-morphological-filter-e952c1ec886e