Multithreaded Image Processing Tool

ECE 5772 - High Performance Embed Programing

Oakland University
Fall 2023

Team: Mazen Albarazi
Martin Shoraji

December 14th, 2023

Application

To create a multithreaded image processing tool that the user can use for many of the popular filtering and morphological image operations and compare the processing times for the different operations in respect to a sequential and a parallel approach, as well as to different image sizes and running it on different machines. This project could even be used to create a GUI for the user, similar to a MATLAB tool that can do different operations on an input image.

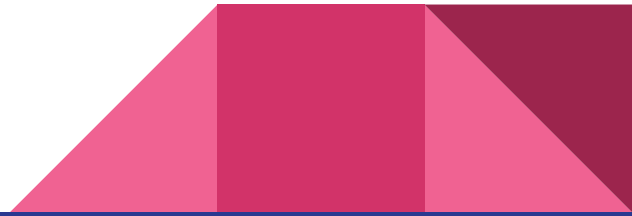


Image Processing Operations

Figure 1. Gaussian Blur

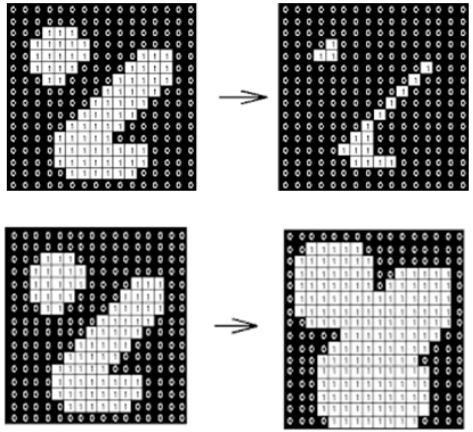


Figure 3. Erosion and Dilation

Figure 2. Edge Detection

Figure 4. Gamma Correction

Algorithm

- Blur and Edge Detection - Filtering using convolution
- Dilation and Erosion - Morphological operations by adding to or subtracting a kernel from the image
- Gamma - Changing luminance values according to the following formula and a factor $Y = 0.6$

$$OM = round\left(\left(\frac{IM}{256}\right)^{\gamma} \times 256\right)$$

Kernels

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

Dilation and Erosion

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

3x3 Average Blur

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

5x5 Average Blur

-3	-3	6
-3	6	-3
4	-5	1

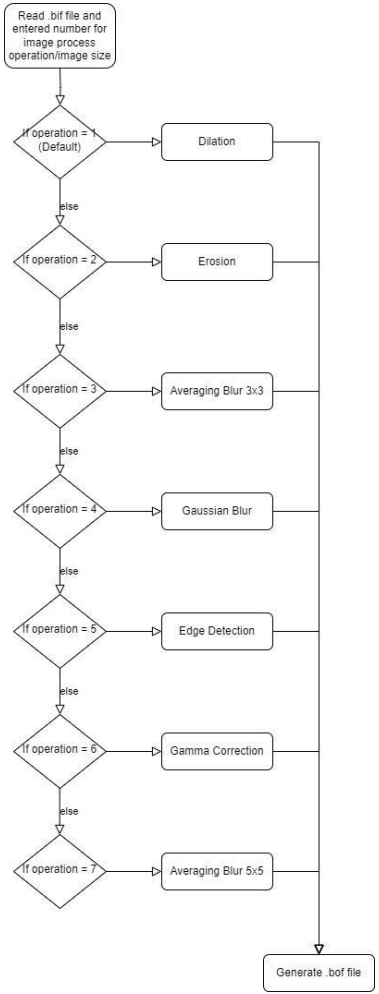
Edge Detection

1	4	6	4	1
4	16	24	16	4
6	24	36	24	6
4	16	24	16	4
1	4	6	4	1

Gaussian Blur

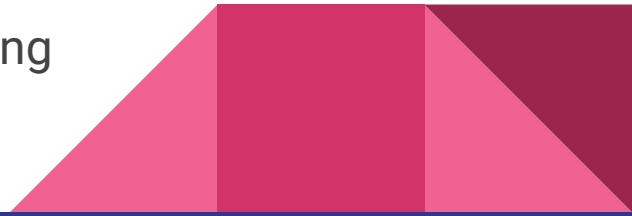
Flowchart

- Read .bif file and operation number/image size
- Run function for desired image processing operation
- Output .bof file



Parallelization Strategy

- Parallel_for loop was used since an operation had to be performed on each individual pixel, which meant multiple pixels could have the same task performed in parallel
- Every iteration is independent
- The number of iterations is known due to the dimensions of the image being known in advance
- Every computation depends particularly on the number of iterations performed and the input data uses the iteration count as an index for the operation
- PThreads was not used due to the program functioning with multiple image sizes, TBB would take care of assigning the number of threads more efficiently



Examples

Original

Greyscale

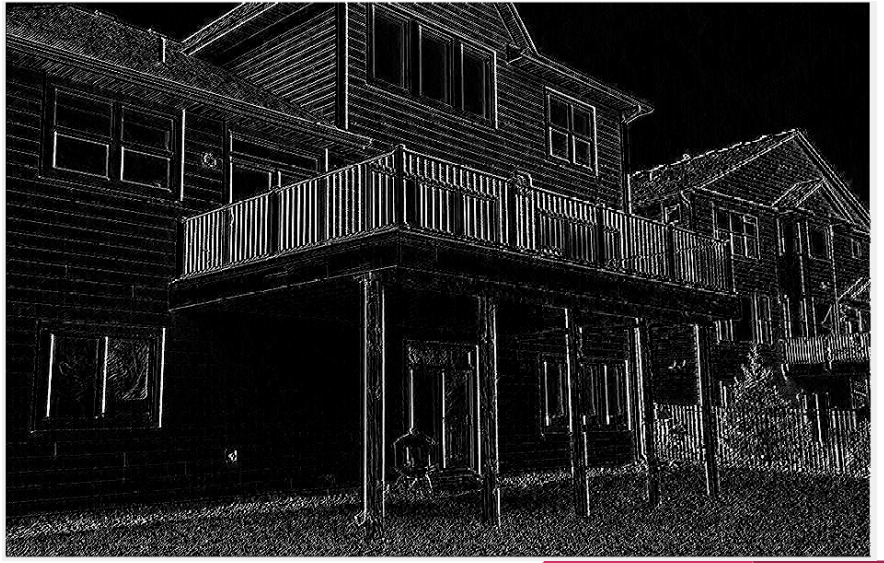
Examples

Dilation

Erosion

Examples

Gamma Correction



Edge Detection

Examples

Averaging Blur 3x3

Averaging Blur 5x5

Gaussian Blur

Data

480 x 307 Image (147,360 elements)

Intel Atom Dual Core Processor N2600 @1.6GHz (Terasic DE2i-150 board)

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	51700	50551	85983	45041	36565	60768	69335
TBB (µs)	39357	48011	66159	36504	28333	49396	50005

Intel Core i5-6200U 2-Core Processor @2.30GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	12331	7470	10197	6360	6149	9809	11508
TBB (µs)	10618	5848	5017	5554	5710	7564	8799

AMD Ryzen 9 5900X 12-Core Processor CPU @3.70GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	2266	3208	1624	1917	1859	2610	2762
TBB (µs)	2069	2288	1371	1685	1465	2120	2161

Data

640 x 410 Image (262,400 elements)

Intel Atom Dual Core Processor N2600 @1.6GHz (Terasic DE2i-150 board)

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	90465	85517	146165	74248	75777	100853	110525
TBB (µs)	78079	66223	106778	63271	64053	98834	103006

Intel Core i5-6200U 2-Core Processor @2.30GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	14426	16097	11604	12222	10274	17111	16025
TBB (µs)	9836	11911	9025	8333	7716	12507	12567

AMD Ryzen 9 5900X 12-Core Processor CPU @3.70GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	4004	5659	2800	3025	2470	4727	4820
TBB (µs)	2319	2363	2337	2596	2266	2265	2409

Data

940 x 602 Image (565,880 elements)

Intel Atom Dual Core Processor N2600 @1.6GHz (Terasic DE2i-150 board)

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	177845	191801	328897	140795	147901	243157	283595
TBB (µs)	128292	120701	192432	107801	114298	178997	200896

Intel Core i5-6200U 2-Core Processor @2.30GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	27416	30031	17240	17475	17134	28852	33812
TBB (µs)	19353	17440	10458	15182	14148	24364	27536

AMD Ryzen 9 5900X 12-Core Processor CPU @3.70GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	8518	12086	6033	6235	4992	9565	10336
TBB (µs)	3057	3572	2630	4437	2939	3542	3631

Data

1280 x 820 Image (1,049,600 elements)

Intel Atom Dual Core Processor N2600 @1.6GHz (Terasic DE2i-150 board)

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	349611	319996	581107	273305	260267	410755	398574
TBB (µs)	158223	155130	344743	250129	256296	260917	297251

Intel Core i5-6200U 2-Core Processor @2.30GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	53153	57552	38139	34844	36914	57270	55719
TBB (µs)	31935	45404	19288	29905	26942	46209	45081

AMD Ryzen 9 5900X 12-Core Processor CPU @3.70GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	15635	22347	11271	10946	8898	17344	17437
TBB (µs)	4365	4644	3171	5837	4034	4753	5325

Data

1920 x 1230 Image (2,361,600 elements)

Intel Atom Dual Core Processor N2600 @1.6GHz (Terasic DE2i-150 board)

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	745256	697655	1207770	523887	515310	908358	939032
TBB (µs)	355341	357881	524633	443125	416930	595624	625761

Intel Core i5-6200U 2-Core Processor @2.30GHz

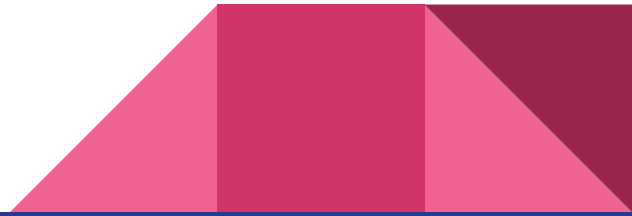
Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	112791	140199	74927	70594	71998	127611	140309
TBB (µs)	64426	82672	50029	54101	52634	94642	109570

AMD Ryzen 9 5900X 12-Core Processor CPU @3.70GHz

Operation	Dilation	Erosion	Gamma	Edge Detection	3x3 Blur	5x5 Blur	Gaussian Blur
Sequential (µs)	35190	50627	25496	25992	20943	37405	41061
TBB (µs)	7074	7298	52634	5076	6847	8619	8852

Results and Conclusions

- TBB is faster/better than sequential in image processing operations
- As image size or number of elements increase, processing time increases and TBB becomes much more efficient than sequential
- As the processor core count and speed increase, processing time decreases



Future Improvements

- Other image processing operations can be implemented
- A MATLAB script to convert an image back from the final .bof back to a .bif so it can be re-processed with another operation
- Parallel_pipeline could be used to assist in indexing the pixels concurrently, which would not speed up the calculation time, but could improve loading time for very large images
- A GUI can be implemented for a more user-friendly image processing tool

DEMO

https://drive.google.com/file/d/1clxwDLuGyqClcZWv
bTiekF1EsrIbVVY/view?usp=drive_link

Thank You!

Questions?