Erosion and Dilation GPU

Claus Marion,
Delassus Hadrien,
Yi Seungme

I. Subject
II. Algorithm
III. Technical point
IV. Benchmarks
V. Conclusion

I. Subject

II. Algorithm

III. Technical point

IV. Benchmarks

V. Conclusion

Subject

Two common morphology operators:

- Dilation
- Erosion

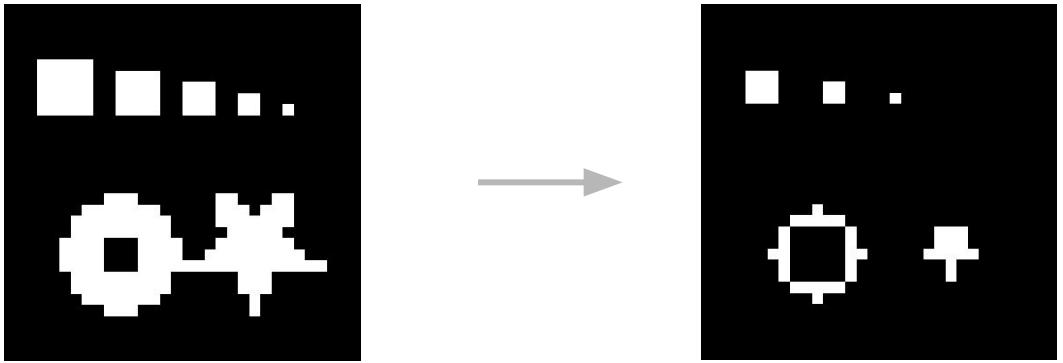
GPU programming language used:

- CUDA

Erosion

$$\varepsilon_B(X) = \{ x \mid B_x \subset X \}$$

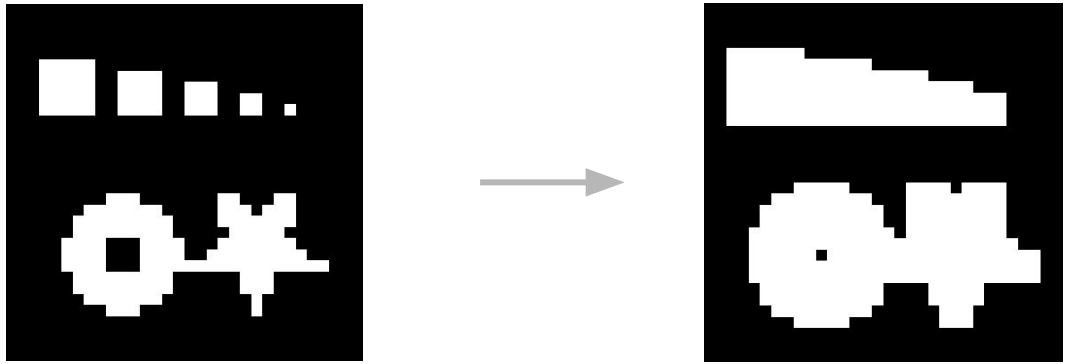
X = image
B = structuring element
x = pixel of image



Dilation

$$\delta_B(X) = \{ x \mid B_x \cap X \neq \varnothing \}$$

X = image
B = structuring element
x = pixel of image



Cuda

- 2007
- NVIDIA
- GPU programming language
- C, C++, Fortran

I. Subject

II. Algorithm

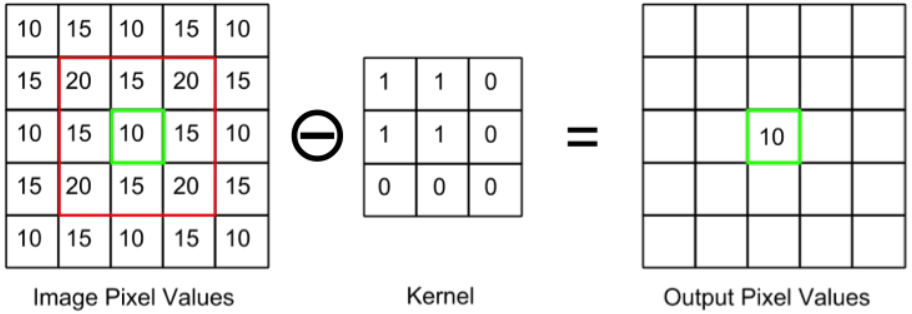
III. Technical point

IV. Benchmarks

V. Conclusion

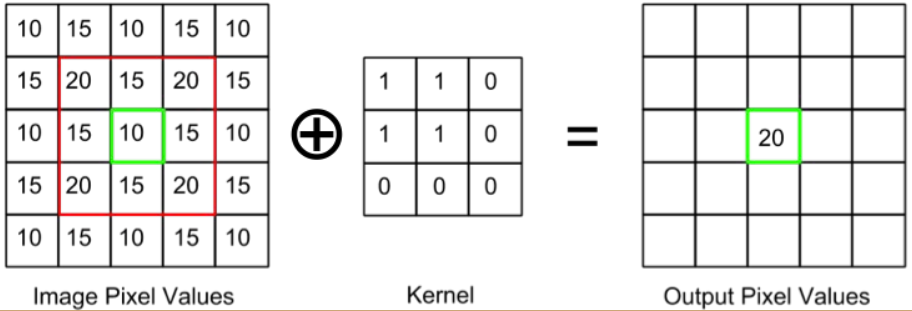
Algorithm

- Based on the convolution algorithm but non-linear



Erosion

Select minimum value



Dilation

Select maximum value

CPU version

- Iterate through the image
- For each pixel:
  - Iterate on the kernel
- Not multithreaded nor vectorized

CPU vectorized version

- Same algorithm as CPU version
- Compiled with flags :
  - `-pthread -m64 -march=native -fopt-info-vec-optimized -O3`
- Most of the loops are vectorized

GPU version

- Addition of padding to the image
- Each pixel is computed in parallel
- Reduced checks for bounds

I. Subject

II. Algorithm

III. Technical point

IV. Benchmarks

V. Conclusion

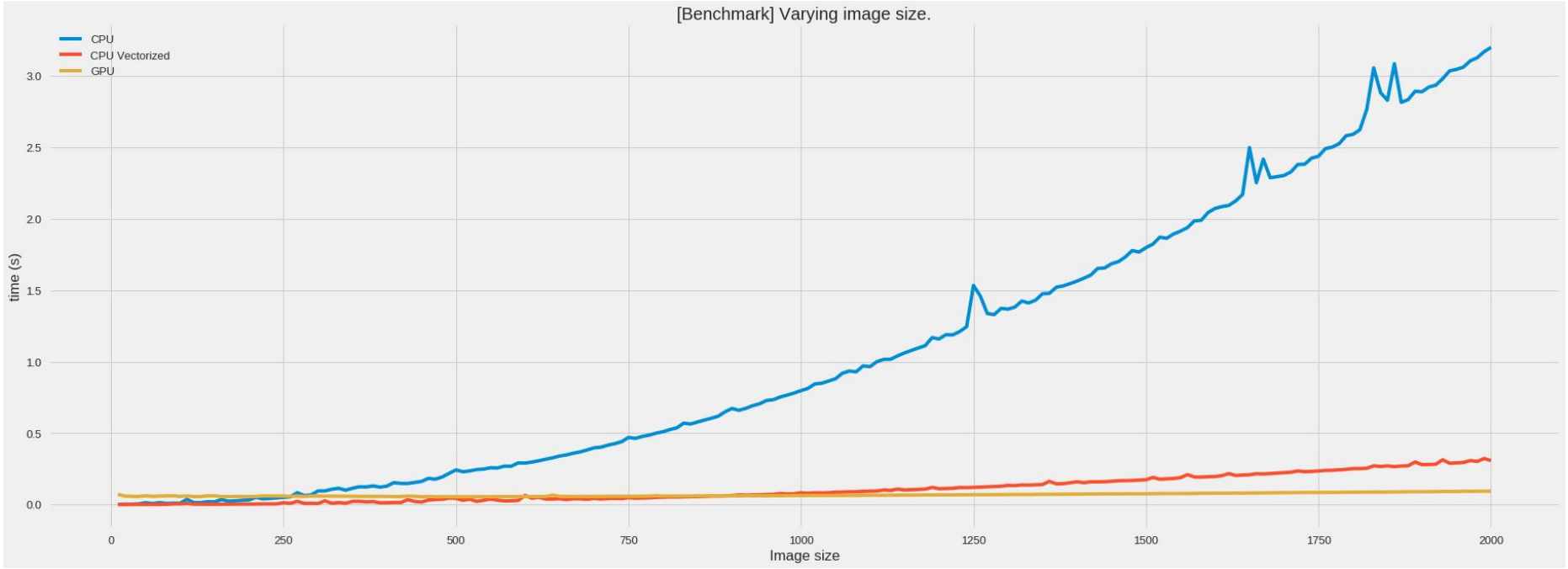
Technical Point

- `Erosion<<<numBlocks, threadsPerBlock>>>(...);`
  - `threadsPerBlock: dim3(32, 32)`
  - `numBlocks: dim3(width / 32, height / 32)`

- Std::chrono used to measure execution time

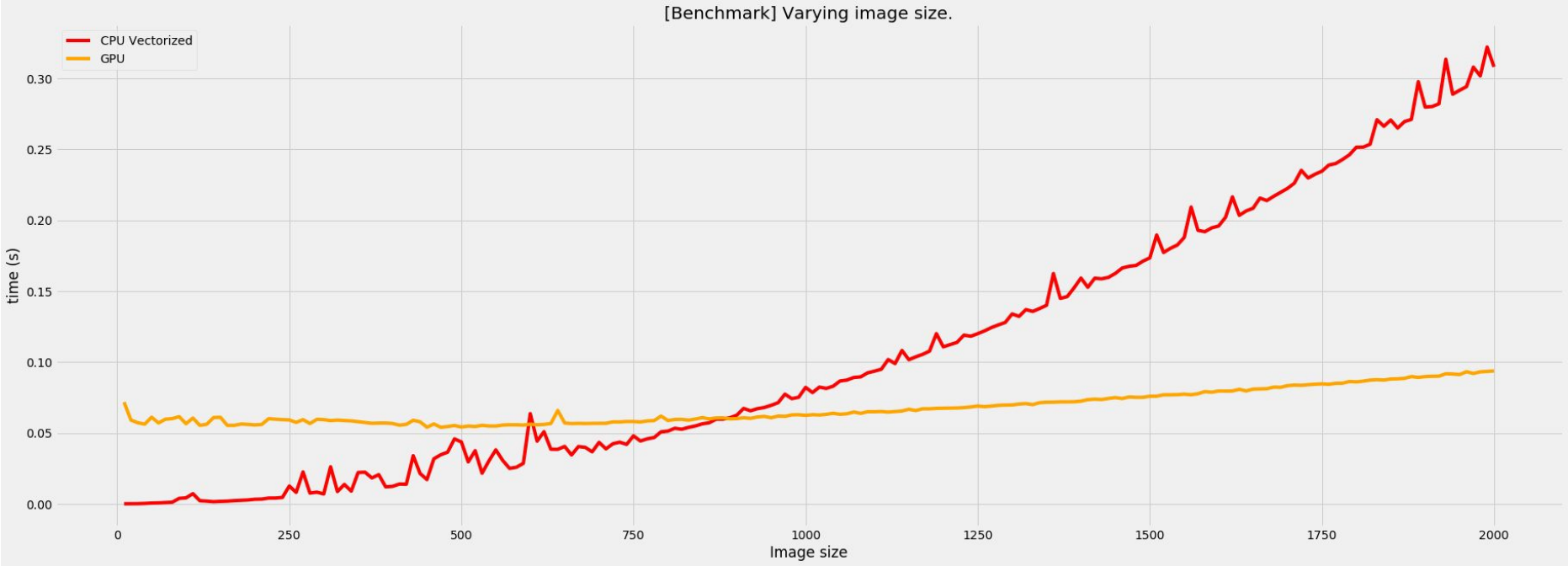
I. Subject
II. Algorithm
III. Technical point
IV. Benchmarks
V. Conclusion

Benchmark



- GPU and vectorized CPU outperform the naive version of CPU

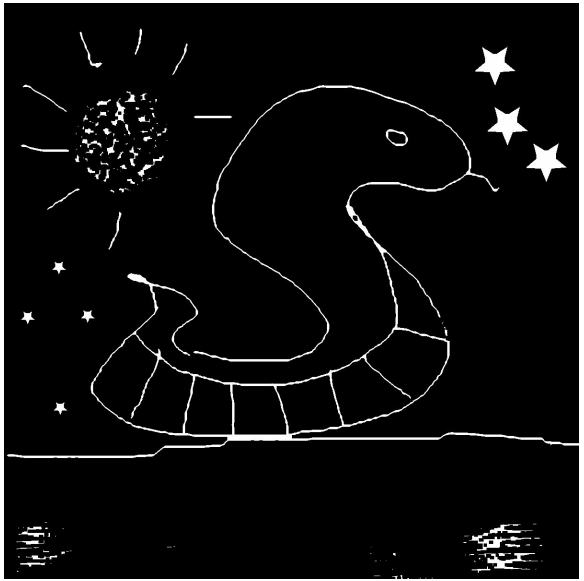
Benchmark

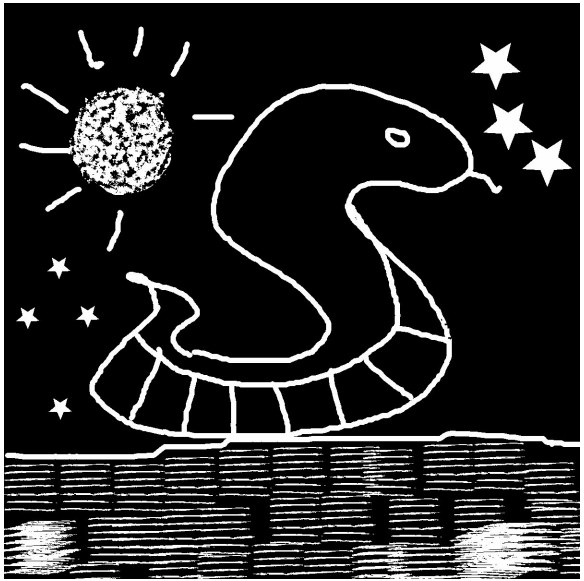


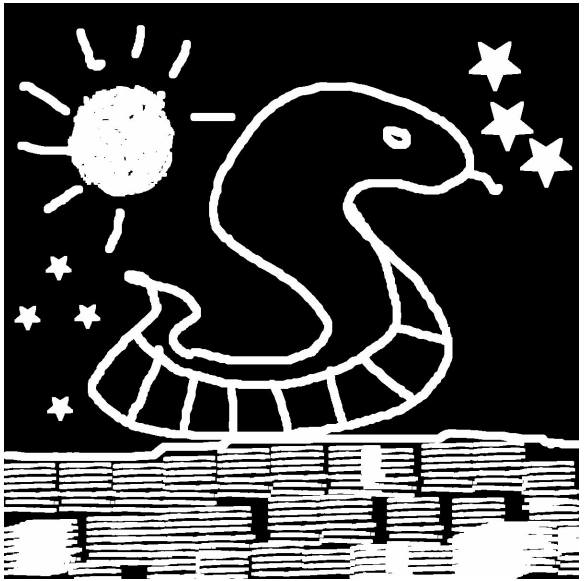
- GPU outperforms with the exception at rates lower than 900*900 image size

Demo

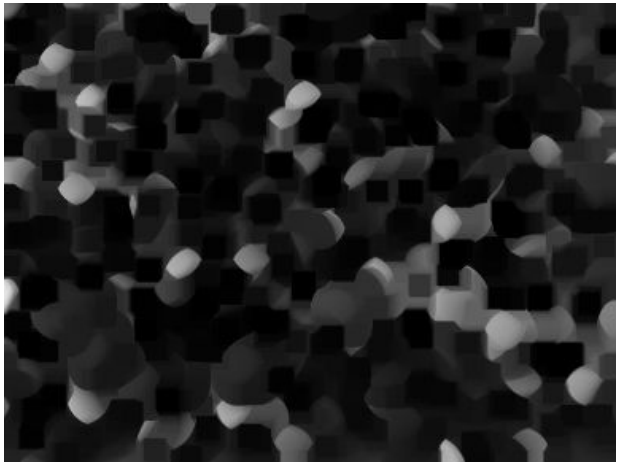
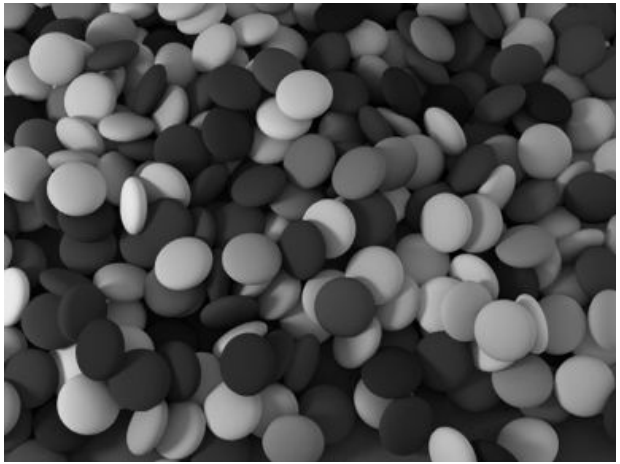
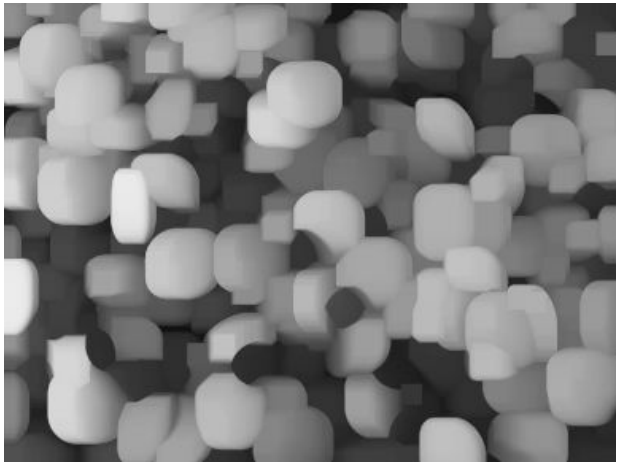
Results







Results

I. Subject
II. Algorithm
III. Technical point
IV. Benchmarks
V. Conclusion

Conclusion

- Learn CUDA
- GPU version faster depending on image size

Any Questions?