Exploiting the GPU

For cross-correlation calculation between 2 images

Anders Handler, s052838, DTU

Summary

To calculate a cross-correlation between two images is an expensive operation. Exploiting the capabilities of the GPU to calculate the discrete Fourier transform can efficiently speedup the cross-correlation calculation. This project presents 3 different GPU implementations of the fast Fourier transform (FFT) and compares them.

The Algorithm

The cross-correlation calculation is sketched on figure 1. The FFT is calculated 3 times (marked). The FFT operation is made in blocks of 16x16 pixels of a 1280x1024 pixels image. The result is shown in figure 2.

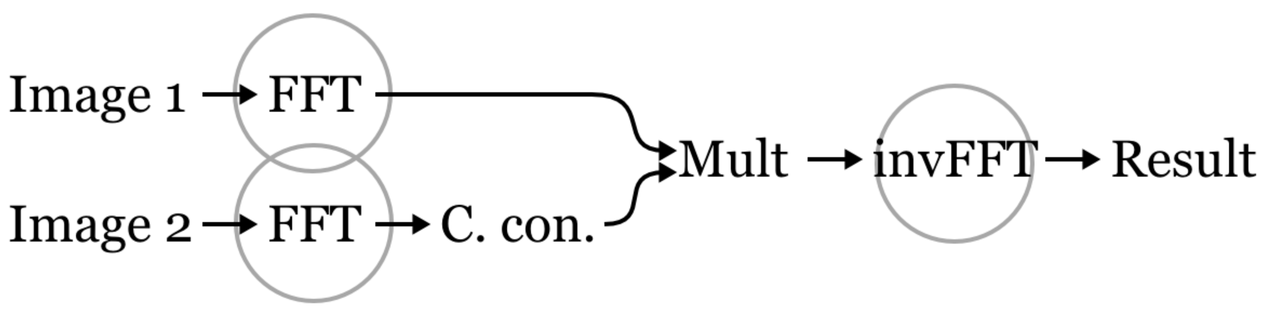

Figure 1: Sketch of the cross-correlation calculation algorithm.

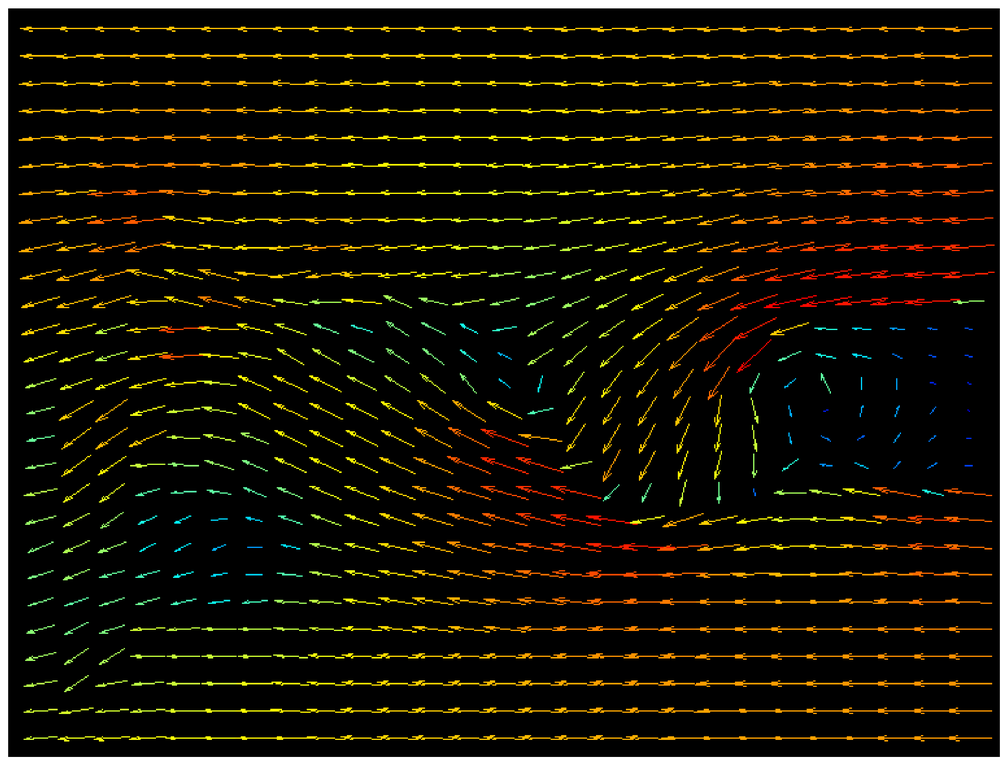

Figure 2: Example result.

Approach 1: CUFFT 2D

Directly using the built-in CUDA CUFFT library. The algorithm is applied on each of the 16x16 pixel blocks as shown in figure 3.

- Good: Easy to implement.
- Bad: Small batch size.

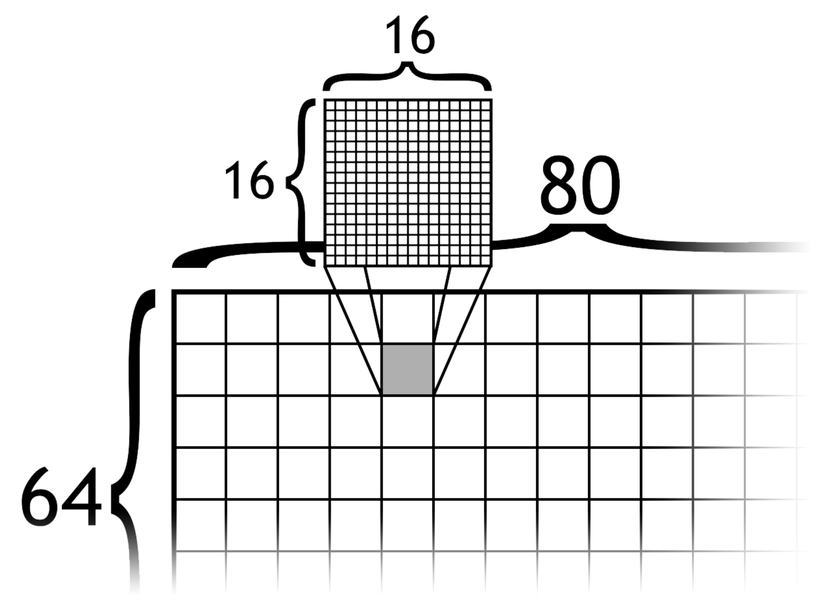

Figure 3: The structure of the image data.

Approach 2: CUFFT 1D x 2

Using the FFT 1D transform two times gives the same result although extra transpose operations is needed as shown in figure 4.

- Good: Batch processing.
- Bad: Extra transpose operations.

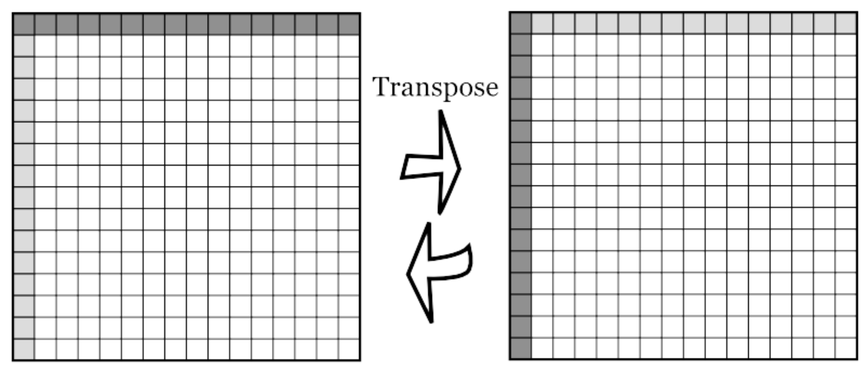

Figure 4: Creating 2D FFT using two times 1D FFT.

Approach 3: Volkov

Same method as in approach 2, but 1D FFT implementation by Vasily Volkov instead of the CUDA CUFFT library.

- Good: Optimized for batch processing.
- Bad: Further data rearrangement needed.

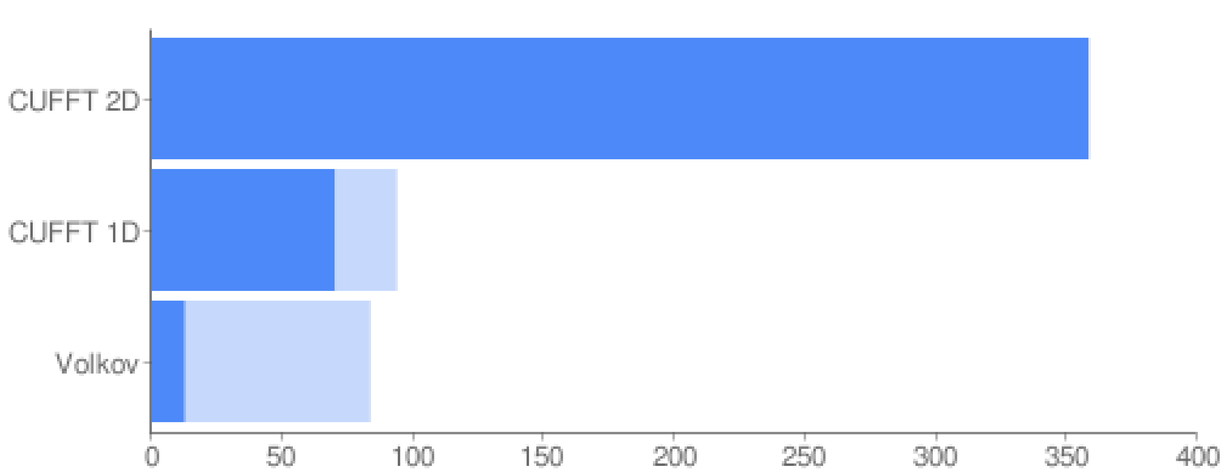

Figure 5: Results. Blue: FFT only, Light blue: Transpose/rearrangement code.

Results

Test was carried out on a NVIDIA Geforce 8400GS card. Results is shown in figure 5. The Volkov implementation is much faster, but rearranging the data eliminates the speed advantage. Using the CUFFT library makes implementation simple, which makes the CUFFT 1Dx2 a good tradeoff between speed and coding complexity.

References

“NVIDIA CUDA: Compute Unified Device Architecture”, 2. Ed., NVIDIA Corp., 2008.

“CUDA: CUFFT Library”, NVIDIA Corp., 2008.

Vasily Volkov, “My speedy FFT, 3x faster than CUFFT”, http://forums.nvidia.com/index.php?showtopic=69801&st=0