Algorithm selection

This part cover the motivation for the selected benchmark application used in this thesis. The basic theory will be presented on a selection of candidates, also why they are suitable for benchmarking GPGPU platforms.

# Discrete Fourier Transform

The Fourier transform is of use when analyzing the spectrum of a continuous analog signal. When applying transformation to a signal it is decomposed into the frequencies that makes it up. In digital signal analysis the Discrete Fourier transform (DFT) is the counterpart of the Fourier transform for analog signals. The DFT converts a sequence of finite length into a list of coefficients of a finite combination of complex sinusoids. Given that the sequence is a sampled function from the time or spatial domain it's a conversion to the frequency domain. It is defined in Definition #.#

**Definition #.#** Discrete Fourier Transform

where and coined as the twiddle factor[[1]](#footnote-0) and the name is often used since.

The DFT is used in many practical applications to perform Fourier analysis. In digital signal processing, such as discrete samples of sound waves, radio signal or any continuous signal over a finite time interval. In image processing the samples can by pixels along a row or column. The DFT takes input in complex numbers and outputs in complex coefficients. In practical applications the input is usually in real numbers.

## Fast Fourier Transform

The problem with the DFT is that it has by definition a time complexity of that makes it too slow for some applications. The Fast Fourier Transform (FFT) is one of the most common algorithm used to compute the DFT of a sequence. An FFT computes the transformation by factorizing the transformation matrix of the DFT into a product of mostly zero factors. This reduces the time complexity to .

The FFT was made popular in 1965 by J.W Cooley and J.W. Tukey and it found its way into practical use at the same time and meant a serious breakthrough in digital signal processing[[2]](#footnote-1) [[3]](#footnote-2). However the algorithm was not invented at the time, the history of the Cooley-Tukey FFT algorithm trace back to around 1805 by work of the famous mathematician Carl Friedrich Gauss[[4]](#footnote-3).

The algorithm is a divide and conquer algorithm that relies on recursively dividing the input into sub-blocks and eventually the problem is small enough to be solved and the sub-blocks are combined into the final result.

# Image processing

Image processing consists of a wide range of domains, earlier work involving performance evaluation on GPU[[5]](#footnote-4) tested four major domains (3D shape reconstruction, feature extraction, image compression and computational photography) and compared with result on the CPU. Generally image processing is by nature parallel and one can expect good results on a GPU (compared to that of an efficient CPU-implementation) from algorithms with the right characteristics.

Most of image processing algorithms apply the same computation on a number of pixels and that is a typically data parallel operation. Some algorithms can then be expected to have huge speed up compared to an efficient CPU implementation. A representative task is applying a simple image filter that gathers neighboring pixel-values and compute a new value for a pixel. If done with respect to the underlying structure of the system one can expect a speedup near linear to the number of computational cores used. That is a CPU with four cores can theoretically expect a near four time speed up compared to a single core. This extends to a GPU so a GPU with *n* cores can in ideal cases expect a speedup in the order of *n*. An example of this could be a Gaussian blur (or smoothing) filter.

## Image compression

The image compression standard JPEG2000 offers algorithms with parallelism but is very computationally and memory intensive. The standard aims to improve performance over JPEG but also adding new features. The following sections are part of the JPEG2000 algorithm[[6]](#footnote-5).

1. Color Component transformation
2. Tiling
3. Wavelet transform
4. Quantization
5. Coding

The computation heavy parts can be identified as the Discrete Wavelet Transform (DWT) and the encoding engine using Embedded Block Coding with Optimized Truncation (EBCOT) Tier-1.

The important difference between the older format JPEG compared to JPEG2000 is the use of DWT instead of Discrete Cosine Transform (DCT). In comparison to the DFT, the DCT operates solely on real values but at the same time complexity. DWT's on the other hand uses another representation that allows for a time complexity of .

# Linear algebra

## Matrix transposition

The method of transposing a matrix.

## Matrix multiplication

Multiplication of two matrices.

# Sorting

Sorting algorithms can be parallelized but there are several conditions for each of them that can make them more or less suitable.

## Bitonic sort

Bitonic mergesort is a parallel algorithm for sorting.

## Sample sort

1. Gentleman, W Morven, and Gordon Sande. "Fast Fourier Transforms: for fun and profit." *Proceedings of the November 7-10, 1966, fall joint computer conference* 7 Nov. 1966: 563-578. [↑](#footnote-ref-0)
2. Cooley, James W, and John W Tukey. "An algorithm for the machine calculation of complex Fourier series." *Mathematics of computation* 19.90 (1965): 297-301. [↑](#footnote-ref-1)
3. Brigham, E Oran, and RE Morrow. "The fast Fourier transform." *Spectrum, IEEE* 4.12 (1967): 63-70. [↑](#footnote-ref-2)
4. Heideman, Michael T, Don H Johnson, and C Sidney Burrus. "Gauss and the history of the fast Fourier transform." *Archive for history of exact sciences* 34.3 (1985): 265-277. [↑](#footnote-ref-3)
5. Park, In Kyu et al. "Design and performance evaluation of image processing algorithms on GPUs." *Parallel and Distributed Systems, IEEE Transactions on* 22.1 (2011): 91-104. [↑](#footnote-ref-4)
6. Christopoulos, Charilaos, Athanassios Skodras, and Touradj Ebrahimi. "The JPEG2000 still image coding system: an overview." *Consumer Electronics, IEEE Transactions on* 46.4 (2000): 1103-1127. [↑](#footnote-ref-5)