

# Parallel Programming for Efficiency Final Project Proposal

Hannah Atmer and Xiaoyue Chen  
Team 6

October 2021

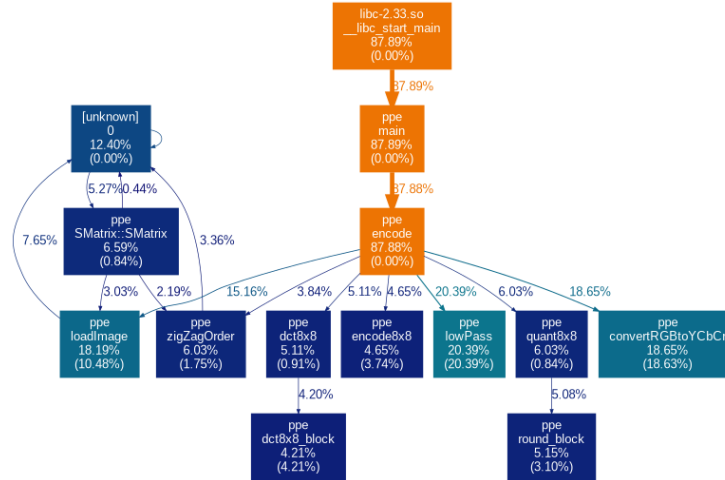


Figure 1: Call-graph of the un-optimized program

## 1 Planned Improvements

Based on the call-graph of the code in figure 1, the functions that are most important to optimise are `loadImage`, `convertRGBtoYCbCr`, and `lowPass`. Therefore, we will use OpenMP and the GPU to optimize these functions. We can use OpenMP to parallelise `loadImage`, and we can use the GPU to parallelise `convertRGBtoYCbCr` and `lowPass`. Once we have loaded the data into GPU memory, we can run the `convertRGBtoYCbCr` kernel first and the `lowPass` kernel second, while keeping the data stored on the GPU, which will further optimise the program run-time.

Additionally, we will use our existing optimisations that we made during the labs, such as treating the image as a 1D array instead of as a 2D array, to improve performance.

We will ensure optimal cache usage in all parts of the program, and use OpenMP to parallelise where possible.

Finally, we will analyze our results and improve depending on what we determine will need to be optimised further. We are confident that we can achieve near-optimal performance with our code.