# Parallel image processing report/analysis:

In this report I will track the impact of parallel development on a Simple Image processing program.  
I have set up 3 different kernels: Converting RGB values to xyY, Lowering Y (Luminance), Converting xyY values back to RGB. I will test several different images and sizes.

Measurements done in Microseconds (µs)

1. Barcelona image: (Size: 3072 x 1728 **Low resolution**) -> Average measurement after 5 tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RGB to xyY** | **Luminance** | **xyY to RGB** | **Combined Serial** | **Combined Parallel** |
| 79100 µs | 40200 µs | 51300 µs | 445000 µs | 198500 µs |

1. Barcelona image: (Size: 7712 x 4352 **High resolution**) -> Average measurement after 5 tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RGB to xyY** | **Luminance** | **xyY to RGB** | **Combined Serial** | **Combined Parallel** |
| 412000 µs | 235000 µs | 377000 µs | 2630000 µs | 1011000 µs |

1. Fungus image: (Size: 1728 x 3072 **Low resolution**) -> Average measurement after 5 tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RGB to xyY** | **Luminance** | **xyY to RGB** | **Combined Serial** | **Combined Parallel** |
| 71000 µs | 40500 µs | 52000 µs | 440000 µs | 163900 µs |

1. Fungus image: (Size: 4352 x 7712 **High resolution**) -> Average measurement after 5 tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RGB to xyY** | **Luminance** | **xyY to RGB** | **Combined Serial** | **Combined Parallel** |
| 402000 µs | 267000 µs | 383000 µs | 2635000 µs | 1035000 µs |

1. Test png image: (Size: 2000 x 1500 **low resolution**) -> Average measurement after 5 tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RGB to xyY** | **Luminance** | **xyY to RGB** | **Combined Serial** | **Combined Parallel** |
| 41900 µs | 22100 µs | 38000 µs | 352000 µs | 101000 µs |

1. Index image: (Size: 12570 x 6000 **Ultra resolution**) -> Average measurement after 5 tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RGB to xyY** | **Luminance** | **xyY to RGB** | **Combined Serial** | **Combined Parallel** |
| 1120000 µs | 5040000 µs | 1340000 µs | 6420000 µs | 2950000 µs |

# Conclusion:

It is clear to see that parallel programming has a positive impact on the runtime of our image processing.

But we can also see that the bigger the image resolution in this case, the greater in the improvement in efficiency. For a lower resolution image there is a 55%-60% run time improvement, but for a higher resolution image this is 70%-75%. At the highest resolution image, I have tested the improvement drops back down to ~55%. So, there is still a significant improvement.

There is also Little to no difference when processing a different filetype, in this case between a .png and .jpg .

The use of TBB blocks can clearly improve the efficiency of projects like this. Having the different instructions completed on different threads on the CPU simultaneously clearly decreases the run time of a project.

The short but obvious and quite clear conclusion is that the use of threading building blocks is helpful and improves the time a piece of software needs to complete different tasks on both a lower and larger scale.