HPC Project Report

# Accelerating KLT Algorithm

**Version 1.0: Profiling**

M. Abdullah Siddiqui | 23I-0617

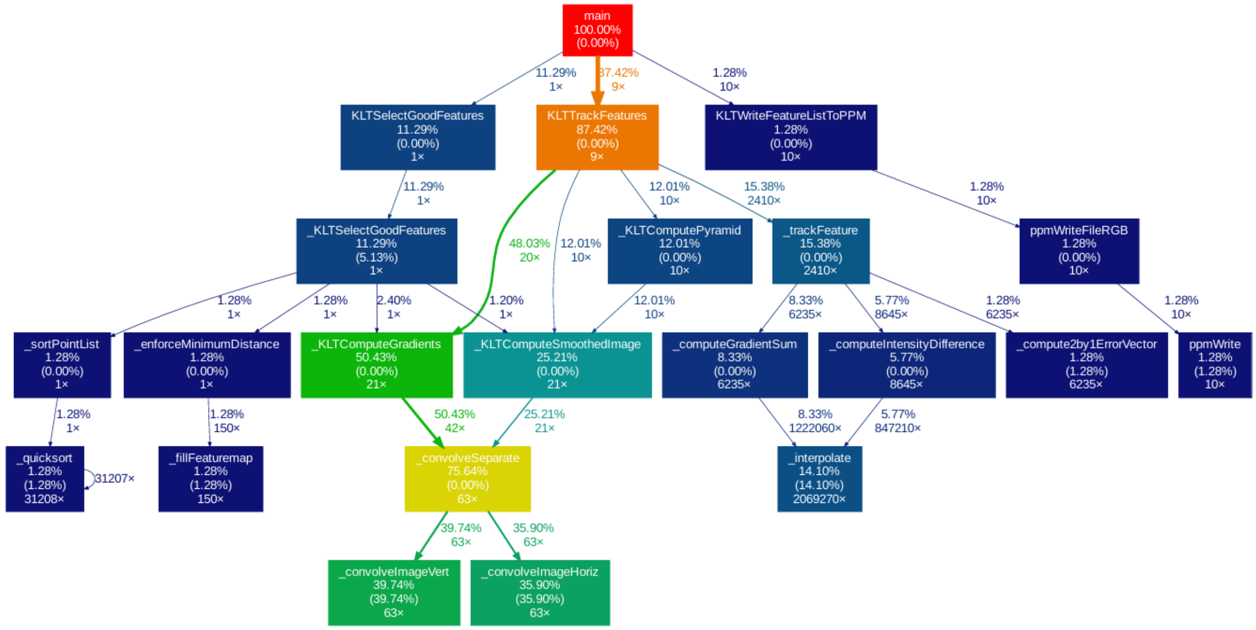
M. Daniyal | 23I-0579

Moiz Ansari | 23I-0523

# Introduction

This version is focused on the profiling of the KLT algorithm, identifying the hotspots and understanding the theoretical performance that can be achieved with the acceleration.

# Identifying GPU-Suitable Functions


main 100.00% (0.00%)
11.29% 1×
87.42% 9×
1.28% 10×
KLTSelectGoodFeatures 11.29% (0.00%) 1×
KLTTrackFeatures 87.42% (0.00%) 9×
KLTWriteFeatureListToPPM 1.28% (0.00%) 10×
11.29% 1×
48.03% 20×
12.01% 10×
12.01% 10×
15.38% 2410×
1.28% 10×
_KLTSelectGoodFeatures 11.29% (5.13%) 1×
_KLTComputePyramid 12.01% (0.00%) 10×
_trackFeature 15.38% (0.00%) 2410×
ppmWriteFileRGB 1.28% (0.00%) 10×
1.28% 1×
1.28% 1×
2.40% 1×
1.20% 1×
12.01% 10×
8.33% 6235×
5.77% 8645×
1.28% 6235×
1.28% 10×
_sortPointList 1.28% (0.00%) 1×
_enforceMinimumDistance 1.28% (0.00%) 1×
_KLTComputeGradients 50.43% (0.00%) 21×
_KLTComputeSmoothedImage 25.21% (0.00%) 21×
_computeGradientSum 8.33% (0.00%) 6235×
_computeIntensityDifference 5.77% (0.00%) 8645×
_compute2by1ErrorVector 1.28% (1.28%) 6235×
ppmWrite 1.28% (1.28%) 10×
1.28% 1×
1.28% 150×
50.43% 42×
25.21% 21×
8.33% 1222060×
5.77% 847210×
_quicksort 1.28% (1.28%) 31208×
31207×
_fillFeaturemap 1.28% (1.28%) 150×
_convolveSeparate 75.64% (0.00%) 63×
_interpolate 14.10% (14.10%) 2069270×
39.74% 63×
35.90% 63×
_convolveImageVert 39.74% (39.74%) 63×
_convolveImageHoriz 35.90% (35.90%) 63×

According to the profiling data, three functions are suitable candidates for GPU acceleration:

1. **_convolveImageVert()** : 39.74% of runtime
2. **_convolveImageHoriz()** : 35.90% of runtime
3. **_interpolate()** : 14.10% of runtime Combined, these represent **89.74%** of total execution time.

These three functions are chosen because:

1. Parallel running ability on the GPU
2. No dependency
3. High runtime, parallelizing will provide better optimization

## Theoretical performance measurement using Amdahl's law

### Single functions: (∞ speedup)

1. Only _convolveImageVert

P = 0.3974 => Speedup = 1 / (1 - 0.3974) = 1 / 0.6026 = 1.66×

2. Only _convolveImageHoriz

P = 0.3590 => Speedup = 1 / (1 - 0.3590) = 1 / 0.6410 = 1.56×

3. Only _interpolate

P = 0.1410 => Speedup = 1 / (1 - 0.1410) = 1 / 0.8590= 1.16×

### Two Functions: (∞ speedup each)

4. Both Convolutions

P = 0.3974 + 0.3590 = 0.7564 => Speedup = 1 / (1 - 0.7564) = 1 / 0.2436 = 4.11×

5. _convolveImageVert + _interpolate

P = 0.3974 + 0.1410 = 0.5384 => Speedup = 1 / (1 - 0.5384) = 1 / 0.4616 = 2.17×

6. _convolveImageHoriz + _interpolate

P = 0.3590 + 0.1410 = 0.5000 => Speedup = 1 / (1 - 0.5000) = 1 / 0.5000 = 2.00×

### All Three Functions: (∞ speedup)

7. All Three Functions

P = 0.3974 + 0.3590 + 0.1410 = 0.8974 => Speedup = 1 / (1 - 0.8974) = 1 / 0.1026 = 9.75×

So, in conclusion if we shift all three functions to GPU it will theoretically increase our performance 9.75x.

## GitHub Private Repository Link

https://github.com/MuhammaDaniyal/Complex_Computing_Problem