Overview

In this project I am going to implement efficient key point detection and description algorithm and use it in an application to trackusing CUDA. Thera are a numerous methods to implement feature matching like SIFT, SURF, HOG, BRIEF. I am going to use BRIEF as a feature descriptor beacuse it is the fastest among all the descriptors. Here keypoints are also referred to as corners. The steps to match feature are corner detection, feaure description, and matching keypoint between the images by calculating smallest Hamming distance to all keypoints in second image In this project, I am going to implement my own version of feature matching algorithm on GPUs using CUDA and going to compare it with OpenCV's own highly optimized implementation.

The other part of this project deals with the implementaion of an object tracking algorithm. As in feature matching, there also exist numerous methods that implement object tracking. Some of those are Boosting, MIL, KCF, Medianflow. In this part of the project I am going to deal with implementing one of these algorithms and try to beat OpenCV's FPS benchmark. Object Tracking involves the process of object initialization, apperance modelling, motion estimation and object localization. The majority of this part deals with finding out which is the slowest part in the OpenCV's implementation and coming up with ways to overcome those bounds through parallelization.

Goals

- Implement the Feature Matching algorithm:
    - Setup OpenCV and run its optimized CPU version.
    - Write the GPU version and beat OpenCV
    - Use shared memory and avoid global memory.
    - Use other optimizations like loop unrolling, thread workload balancing.
- Implement the Object Tracking algorithm:
    - Setup OpenCV and run its optimized CPU version.
    - Perform analysis on the steps involved in OpenCV version.
    - Write slow parts of the code on GPU.
    - Beat OpenCV's baseline.

References

1. https://www.learnopencv.com/object-tracking-using-opencv-cpp-python/
2. http://ieeexplore.ieee.org/document/7284011/
3. http://ieeexplore.ieee.org/document/6935620/