Optimizing Bundle Adjustment In Structure From Motion Using GPUs

Proud Heng, Aishika Kumar, Rene Sanchez
Advisors: Quentin Gautier, Alric Althoff, Ryan Kastner

Motivation

Structure From Motion (SFM) is the process of producing a 3D reconstruction of an object from 2D images at different views. There are many applications of SFM such as archaeology, where creating a model using SFM is much faster than drawing by hand.

Many Images

Detect Good Features

Feature Matching

Outlier Rejection

3D Point Cloud

Bundle Adjustment

Triangulation

The step by step process of SFM; left shows the time spent in each part of the algorithm.

Background

After triangulation, we are left with a point cloud that is not completely accurate. Bundle Adjustment (BA) refines the accuracy of the point cloud using matrix multiplications.

OpenCV: An API for computer vision, which processes matrix operations on the CPU.

ArrayFire: An API used for parallelizing matrix operations on the GPU.

Matrix multiplications are time intensive as matrix size grows, so using the multi-core efficiency of Graphical Processing Units (GPUs) will optimize them. Currently, they are run with OpenCV, so we will parallelize and run them on the GPU with ArrayFire.

Our Design

We focused on optimizing the initialization of Bundle Adjustment since it contained a large number of matrix multiplications. Each matrix multiplication was assigned to a thread, and we launched all the threads at once. The matrix multiplications were then converted from OpenCV to ArrayFire so that they could be run on the GPU. Upon finishing initialization, we resumed the original SFM pipeline.

Changes we made to the Bundle Adjustment pipeline. Our multi-threaded approach is in red block; our ArrayFire approach is in red + green blocks.

Simple Structure Test

We tested both approaches on a simple 24-image dataset of a computer generated 3D model of a monkey’s head. The ArrayFire implementation ran 0.5 seconds faster than the multi-threaded one. However, both approaches were slower than the baseline code.

This was due to not fully optimizing the thread creation and handling algorithm, as well as the overhead of converting from OpenCV to ArrayFire data-structures.

Our Prototype

We tested two versions of our prototype on the same dataset.

1. OpenCV
2. ArrayFire

The second test consisted of 14 images of a Kermit the Frog puppet sitting in a detailed environment.

The results were similar to the first test on the simple structure.

Complex Structure Test

This result is informative, since ArrayFire was expected to perform better with more features to process.

Conclusion

The performance boost between our multi-threaded and multi-threaded with ArrayFire designs show that GPU-based parallel operations are a good approach for optimizing the runtime of SFM. Our results also show that the current OpenCV modularity restricted the performance boost from GPU parallelization.

Future Work

- It became clear that the bottleneck in BA is actually the OpenCV function used to minimize the reprojection error.
- Parallelizing said function would improve SFM’s runtime.
- Can improve the runtime of thread creation and handling.

Acknowledgements

We would like to thank Christine Alvarado and Michael Barrow for all of their guidance, the ERSP program for providing us this research opportunity, and Ryan Kastner’s research team for supporting us. This material is based upon work supported by the National Science Foundation under Grant No. CNS-1339335.