

Baboons on the Move

Oral Project Update

Debaditya Basu, Sananya Majumder, Zhi Wang, Baiqiang Zhao

Project Overview

“The behavior of monkeys and apes has always held great fascination for [humans]”.

– Washburn and DeVore, 1961, "The Social Life of Baboons."



© Love Nature

Can you find the baboons?



“Using Cutting Edge Computer Vision to Invade Baboons' Privacy Since 2018”



Overview

- Key problem: The baboon-tracking algorithms takes a *huge* amount of time for post-processing in Python.

Solution: Implement the algorithms in C++, and potentially CUDA where we can take advantage of parallel execution.



Python



C++/CUDA

Project Objectives – Milestones

- Port the algorithms to C++
- Performance benchmarking
- Refactoring existing implemented code
- CUDA implementation of the algorithms that can be ported

Project Objectives – Deliverables

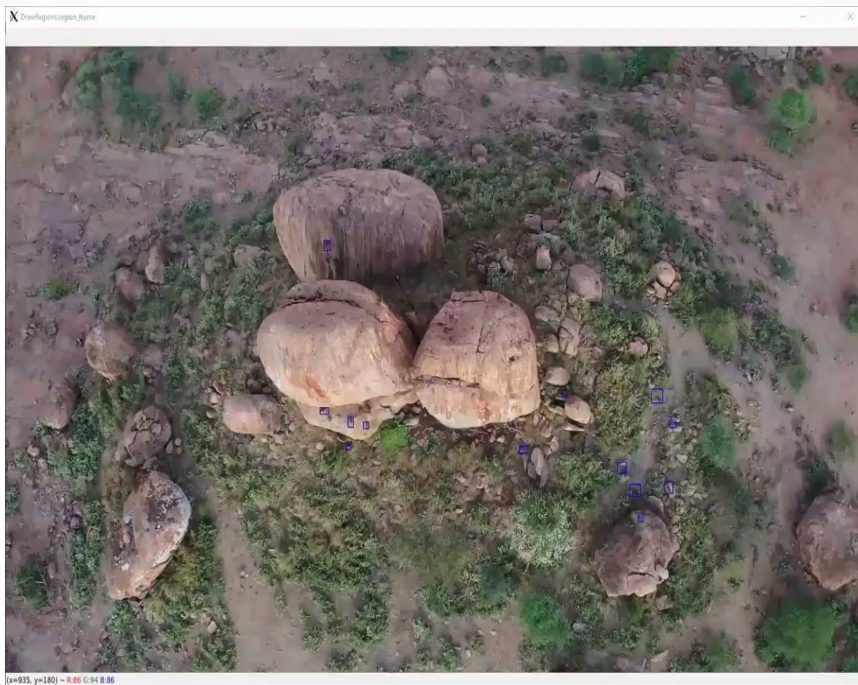
- Ensure the functionality of the C++ algorithms match the Python algorithms
- Show at least 10% improvement in performance in terms of runtime in C++
- Optimize some of the stages in the existing code implemented in C++
- Feasibility study and improvement in performance in terms of runtime in CUDA

Accomplishment/Demos

Stage	Time (ms)		👍 ?
	Python	C++	
Get frame	57	1	98% ✓
Preprocessing	25	2	92% ✓
Store History Frame	0.1	0.1	–
Compute Transformation Matrices	4826	1981	59% ✓
Transformed Frames	714	407	43% ✓
Quantize History Frames	305	145	52% ✓

Stage	Time (ms)		👍 ?
	Python	C++	
Generate Weights	198	30	85% ✓
Generate Mask Subcomponents	632	112	82% ✓
Compute Moving Foreground	47	83	-77% ✗
Apply Masks	15	8	47% ✓
Erode Dilation	134	18	86% ✓
Detect blobs	45	1	98% ✓

Accomplishment/Demos

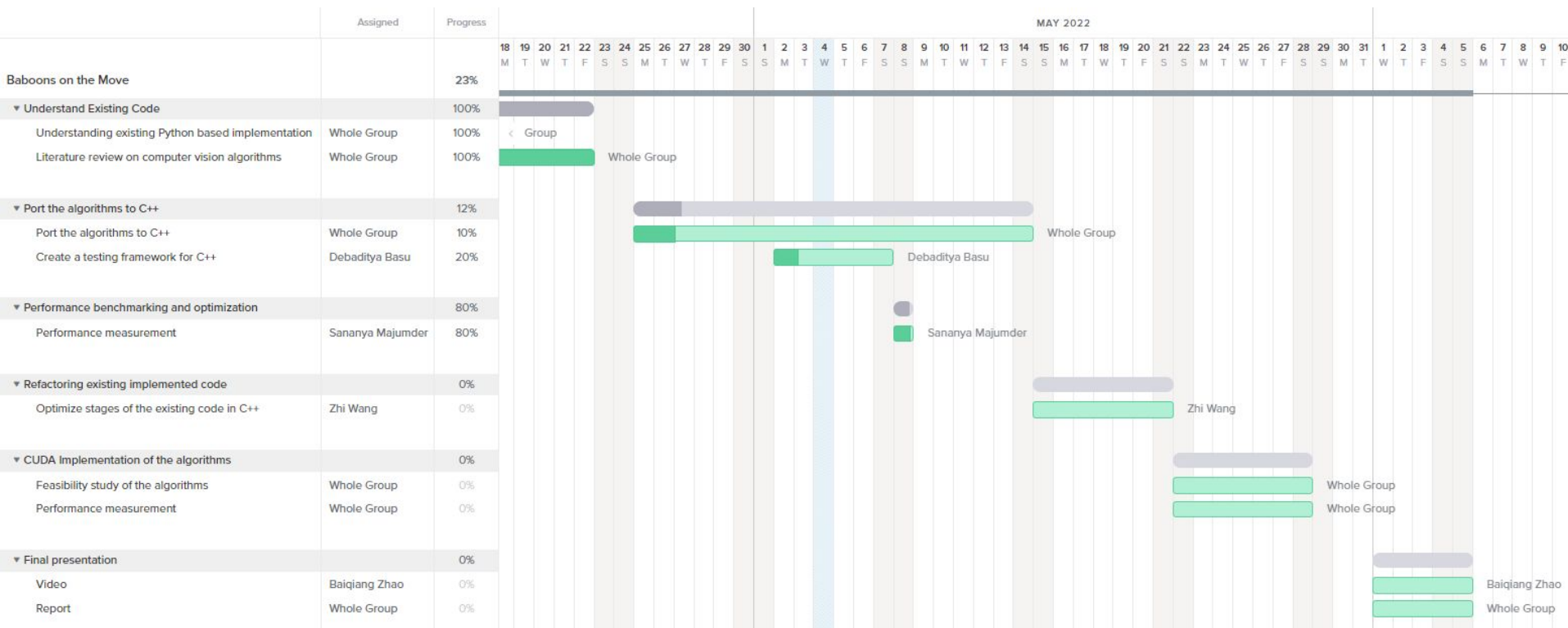


Python codebase



C++ codebase

Quarter Plan



Conclusion

- **Objective:** Improve the performance of baboon tracking algorithms by reimplementing them in C++ and CUDA
- **What we have done so far:** Ensured functionality of various C++ implemented stages, benchmarked the performance - 62% overall runtime improvement ✓
- **What's next?** Further optimize the runtime of certain stages in the C++ implementation, explore the possibility of parallelizing algorithms in CUDA



© Francois Gohier/ardea.com