Project Proposal

Overview

Student

- Brian Scott Bohme (A0144436H)

Supervisor:

- Professor Bruno Bodin

Title:

- **Natural User Interfaces for SLAMBench2 Artifacts: Benchmarking Tools, Information Visualization, and Cloud-Based Services on Mobile Platforms**

Subject Areas:

- Natural User Interfaces
    - User Experience
    - Platform Development
        - Mobile
- Designing Interactive Systems
    - Prototyping
    - Iterating
    - User Research
- Benchmarking Tools
    - Optimization
- Data Visualization
- Hardware
    - GPU/Device Performance
- Simulataneous Localization and Mapping
    - Computer Vision

Scope

SLAMBench is a SLAM (Simulatenous Localization and Mapping) performance benchmark that combines a framework for quantifying quality-of-result with instrumentation of accuracy, execution time, memory usage and energy consumption. The goal of this project is to meaningfully contribute to SLAMBench by developing interfaces for the benchmarking platform that enchances the user experience, deepens the user's understanding of meaningful results, and serves as an industry-standard for benchmarking suites. Keen consideration will be placed on creating a natural user interface, robust information visualization, and integration of metrics with cloud-based services (multiple algorithms/datasets without the need to hardcode them into the tool itself). The value of the project is in lessening the effort required to obtain results and insights for a major benchmarking tool that is a key component of robotics and AR systems, while crafting a cutting-edge interface for exploring and utilizing benchmarking artifacts. The final product will be a mobile tool built for SLAMBench2 that integrates multiple algorithms and datasets and outputs relevant data artifacts/visualizations in an intuitive, precise and meaningful manner.

The main scope of this project will involve:

- Understanding the SLAMBench2 tool and the needs of its users
- Developing multiple in-depth prototypes
- Evaluating these prototypes (as per Designing Interactive Systems specification) and iterating upon collected quantitiative/qualitative data
    - This will be the most intensive part, which includes multiple lab tests, interviews, experiments, observations, etc.
- Implementing the interface(s), with keen considerations of platform being developed for (i.e. mobile)

Main considerations to ensure the scope of the project is robust enough include:

- Efficiency/Performance
- Design
- Intuitiveness (natural user interface research)
- Intrinsic to industry standards of benchmarking platforms.

The project will also serve as an opportunity to deepen understanding of SLAM/computer vision concepts and algorithms. A large component of the capstone will be a deep exploration into tools such as SLAM, other benchmarking platforms and collaterals such as CMake. Personally, this will be an opportunity to build new skills as per the nature of knowledge and inquiry demanded by the Capstone project.

Challenges

- Fully understanding the SLAMBench2 system with current level knowledge of simulateneous localization and mapping/computer vision
    - Working in new languages and environments
- Highlighting depth of work required for this type of interface design
- Alloting enough time to thoroughly work through iterative, agile-based process
- Communicating the higher-level aspects (theory and practice of benchmarking, natural user interface design, etc.) clearly and confidently.

Considerations

While developing the capstone plan, a couple important considerations were already taken into account:

- Platform
    - When deciding between working on a mobile or web platform, the differences in purposes were highlighted
        - The web SLAMBench2 platform is more robust, meant for professional usage (emphasizes calculations, metrics, etc.)
        - On the other hand, the mobile platform is more visually-enticing, meant to "show off" results such that you could quickly show others what the tool is doing in an impressive and intuitive manner
    - Because my summer internship fell in the latter category of developing natural, beautiful and robust interfaces for the everyday user, I chose to pursue implementing mobile interfaces
- Operating System
    - Android provides the clearest/most ensured path because code already exists. While I will still explore the possibilty of implementing an iOS tool (at the very least, document the steps necessary for doing so for future work beyond the scope of the Capstone), I don't want to derail/diverge from the main scope of the project by being hindered by bridging/compatibility issues that cannot be circumvented, or spend long portions of time learning how to bridge/compile native code. Hence, from the beginning we will stick with an Android focus, which in turn gives opportunity to explore a new environment and platform for my own sake.
- Device
    - Because a large part of SLAMBench is measuring artifacts such as execution time and energy consumption, the hardware/device used will be of core relevance. Part of the initial research will be on different devices/GPUs, but it was decided that a cutting-edge device would be a meaningful way to "push the limits" of the tool and truly see the capabilities, hence a new Huawei device will be utilized (ideally with the Kirin 970-980 system-on-a-chip). Using such a device allows the scope of the research to be pushed towards the future, rather than risking the findings/implementations becoming stale with newer technology arriving.

Time Allocation

September:

- Submit proposal **(September 7th)**
- Solidify knowledge of SLAM and SLAMBench2 alongside elementary readings of CMake *(at least 2 weeks)*
- Collect and document existing state of benchmarking research (for example, current/future benchmarking technologies/interfaces/designs, etc.). Understand breadth and depth of benchmarking tools and identify key components that will be actionable/implementable in the Capstone project *(up to 1 week)*
    - This should include some research and insight into hardware's role (devices, GPUs, etc.)
- Begin natural user interface research and highlight specifics that I want to focus on/implement in the frontend of the Capstone project *(up to 1 week)*
- Important: Document all initial research, plans and resources extensively.

October:

- Begin work with requisite tools *(at least 3 weeks)*
    - Intensive learning of CMake and native Android environment/development *(at least 2 weeks)*
    - Hands-on work with the build system of SLAMBench2
    - Exploration of possible iOS development *(No more than 1 week)*
        - Will it be possible to bridge? Will it require cross compilation?
- Submit milestone 1 **(October 19)**
    - Implement modular command line-based interface that highlights basic goal of being able to pick and choose algorithm/dataset, which then outputs results
        - This can start with implementation of Kinect Vision algorithm, then expanding to show that the tool is modular *(3 weeks, to be done alongside tasks above)*
    - Showcase start of compatibility testing with Android.

November:

- Continue exploration of tools *(No more than 2 weeks)*
    - Hands-on work with Android device
    - Tackling collaterals from CMake/Android (C#/C, etc.)
- Submit milestone 2 **(November 16)**
    - First frontend prototype(s) exists, with acknowledgement of implementation plans *(at least 2 weeks)*
        - Loading datasets/algorithms libraries
            - Key question: What does this look/feel like?
    - Key considerations of design should be highlighted. This includes the naturalism of the user interface (i.e. a non-expert should be able to use intuitively with little to no instructions) and the mobile design thoughtfully considered
        - The mobile platform emphasizes the ability to “show and tell” your results in a fancy, visually-interesting manner, as compared to the desktop version which is more data/analytically driven for professional use.

December:

- Connecting SLAMBench2 to frontend
    - This is the core component of the Capstone and should therefore take the most amount of time
    - Focus on Kinect Vision algorithm implementation, then expand to show modularity of interface *(2 weeks)*
    - Different datasets should be able to be processed
    - After implementing one algorithm, focus should be put on making sure the product is stable and usable (while meeting most design specifications) before moving on to implementing more algorithms.
- Begin iteration/variation process of frontend

January/February:

- Continue implementation process of frontend *(on-going as needed)*
- Garner and implement feedback

    - Feedback from other SLAMBench2 researchers/members or public users *(at least 2 weeks)*
    - Iterate by implementing feedback and gathering more feedback data *(1-2 weeks)*
- Bug testing, and polishing project for symposium showcase and initial thesis submission.

March:

- Submit inital thesis **(March 22, before 5pm)**

Expectations

The main expectations are highlighted in the milestones above. There are two submitted milestones, and two personal milestones:

1. Command-line based implementation (submittable report)
2. First frontend prototypes exist (submittable report)
3. Variations of interfaces done (personal milestone)
4. Feedback received and implemented (personal milestone).

The expectation is that SLAM/SLAMBench is fully realized and understood, and there is a solid, usable interface implemented (i.e. it is better to have a working interface with some clunky elements than an incomplete project).

Fallback Plans

In the event that connecting SLAMBench2 to the frontend (planned for December) is taking longer than expected, the number of iterations/variations on the prototype should be lowered to 1-2 maximum. The amount of feedback gathered and implemented should also be adjusted; priority should be given to receiving feedback from experienced team members/SLAMBench2 researchers over public feedback. Emphasis should be placed on creating a usable interface that highlights good DIS specifications, some natural user interface elements, and strong understanding of benchmarking tools. Extras, such as information visualization and more features, would be ideal, but are discretionary.