

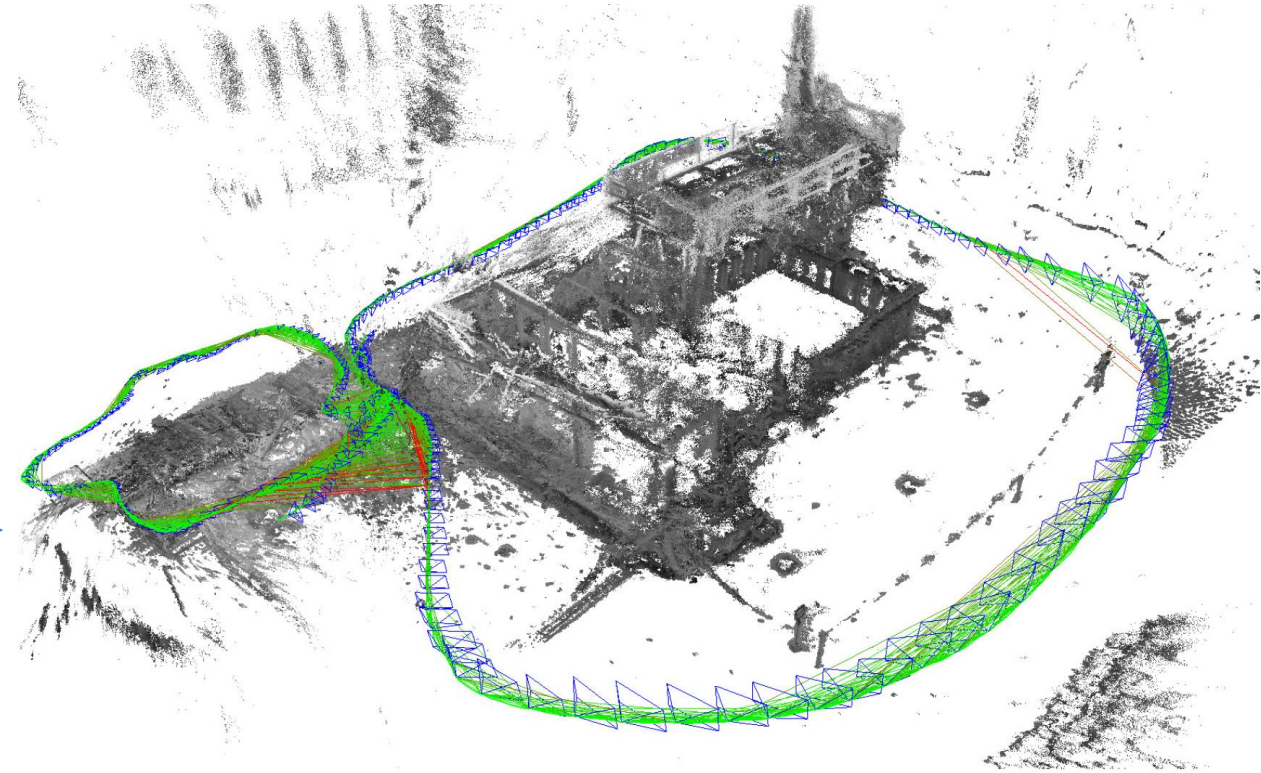
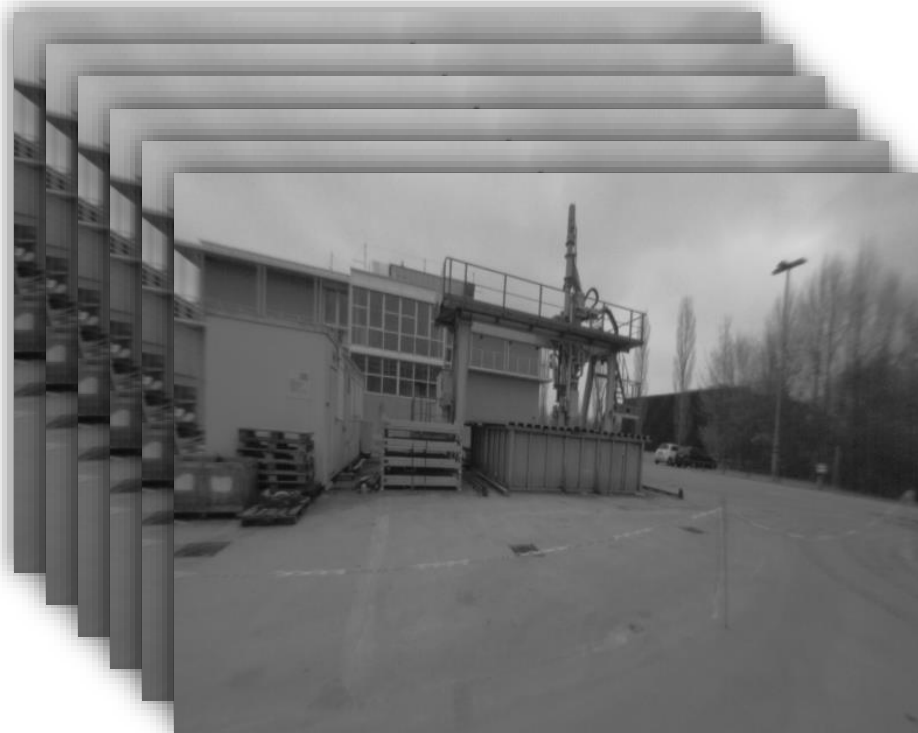
Semi-dense SLAM on an FPGA-SoC

Konstantinos Boikos, Christos-Savvas Bouganis
k.boikos14@imperial.ac.uk

FPL 2016, Lausanne
Sept 2016

What is SLAM?

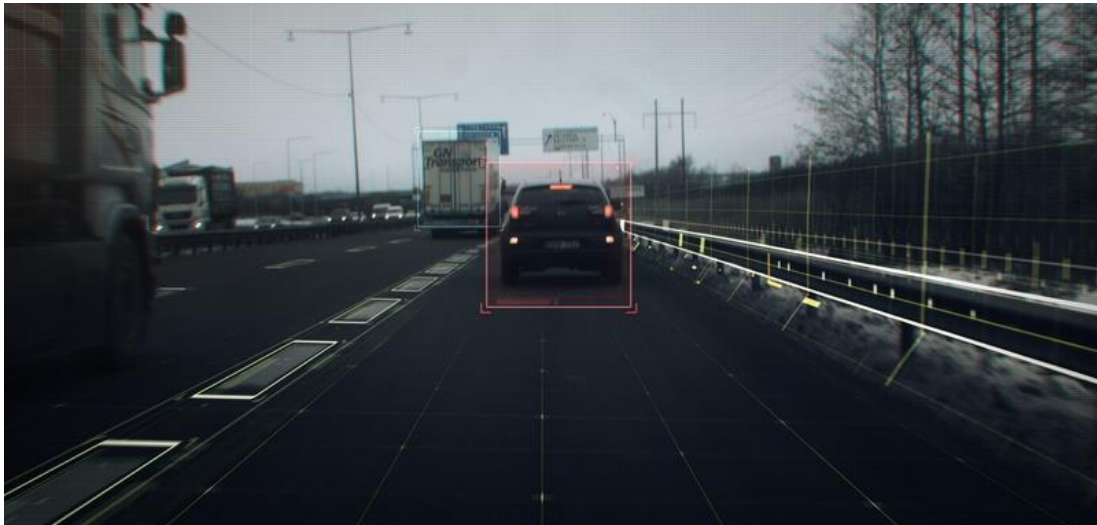
- Simultaneous Localisation and Mapping
- Use a series of observations to:
- Reconstruct an environment (map)
 - Track the observer's trajectory in it



Why is it Important?

Advanced SLAM algorithms are fundamental in autonomous robotics and emerging applications

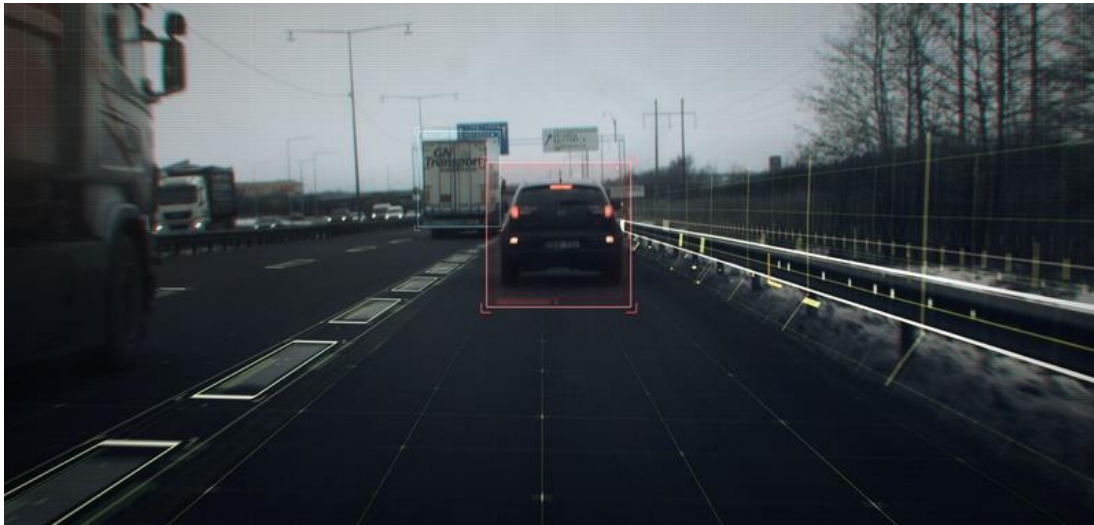
➤ Self-driving Cars



Why is it Important?

Advanced SLAM algorithms are fundamental in autonomous robotics and emerging applications

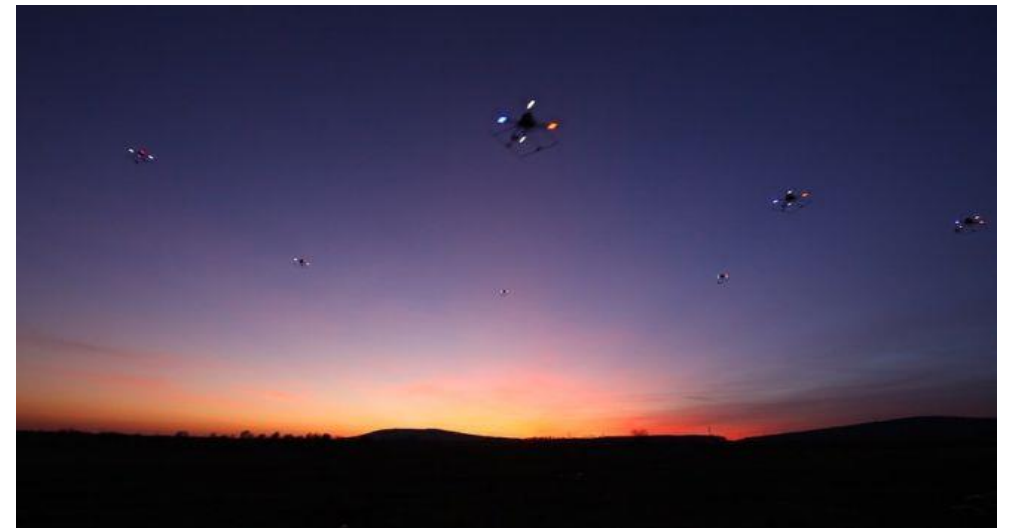
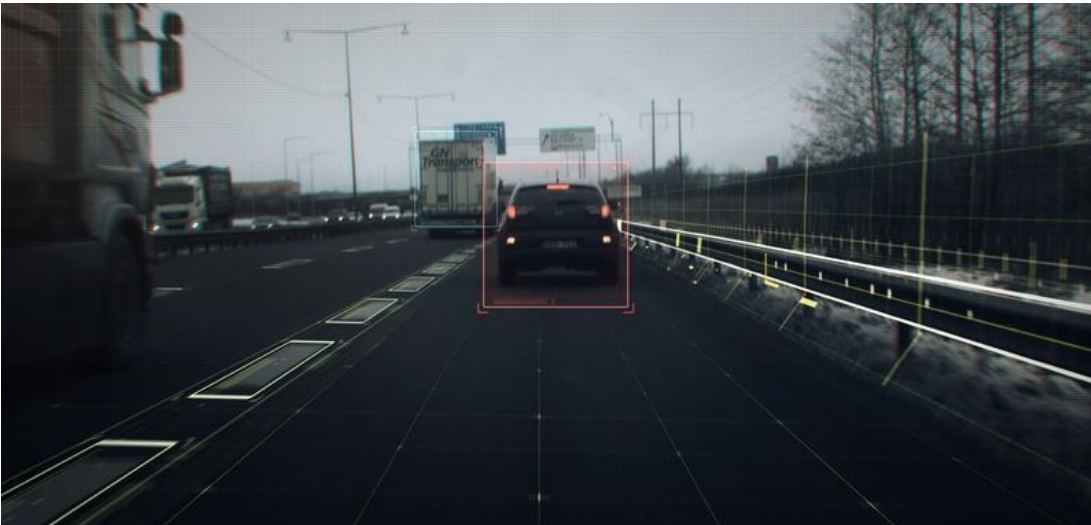
- Self-driving Cars
- Precision Agriculture



Why is it Important?

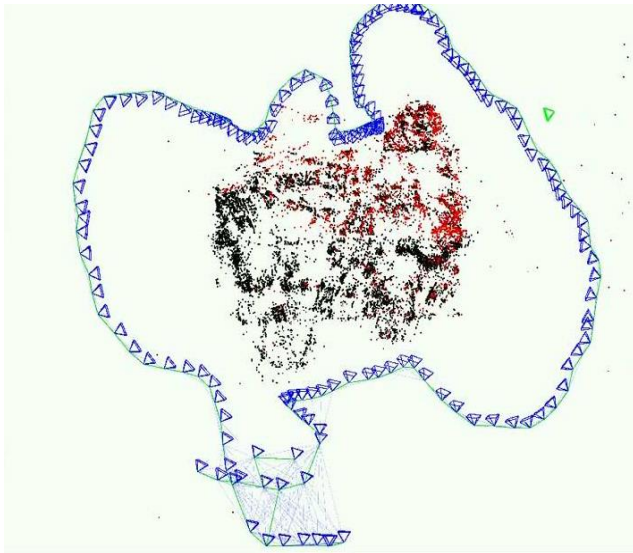
Advanced SLAM algorithms are fundamental in autonomous robotics and emerging applications

- Self-driving Cars
- Precision Agriculture
- Rapid environment exploration
- Much more effective search and rescue operations

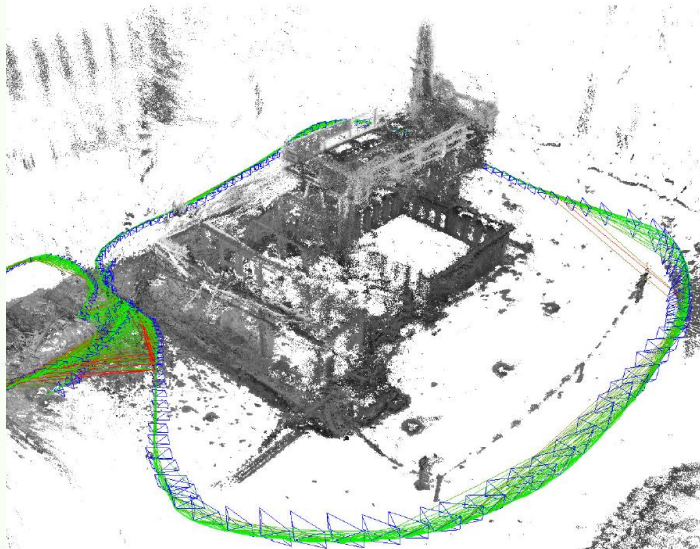


Challenges in Embedded SLAM

- High complexity and bandwidth requirements
- Low latency and high framerate crucial to keep track of a fast moving robot
- Emerging applications require an unprecedented richness and accuracy



Sparse



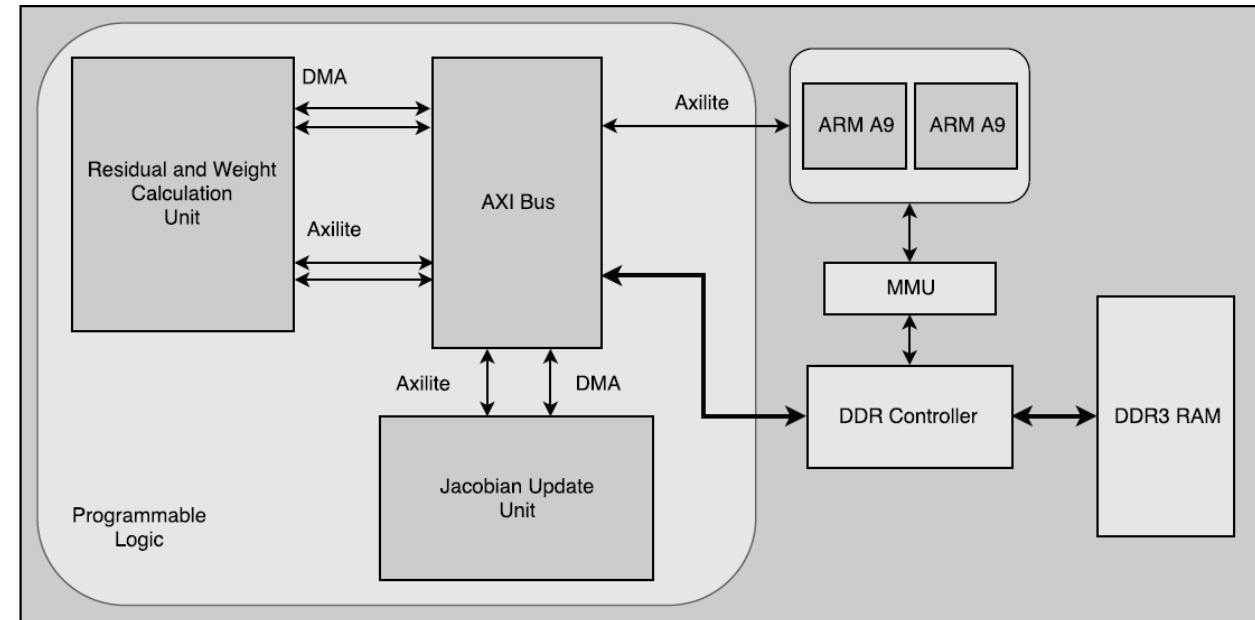
Semi-Dense



Dense

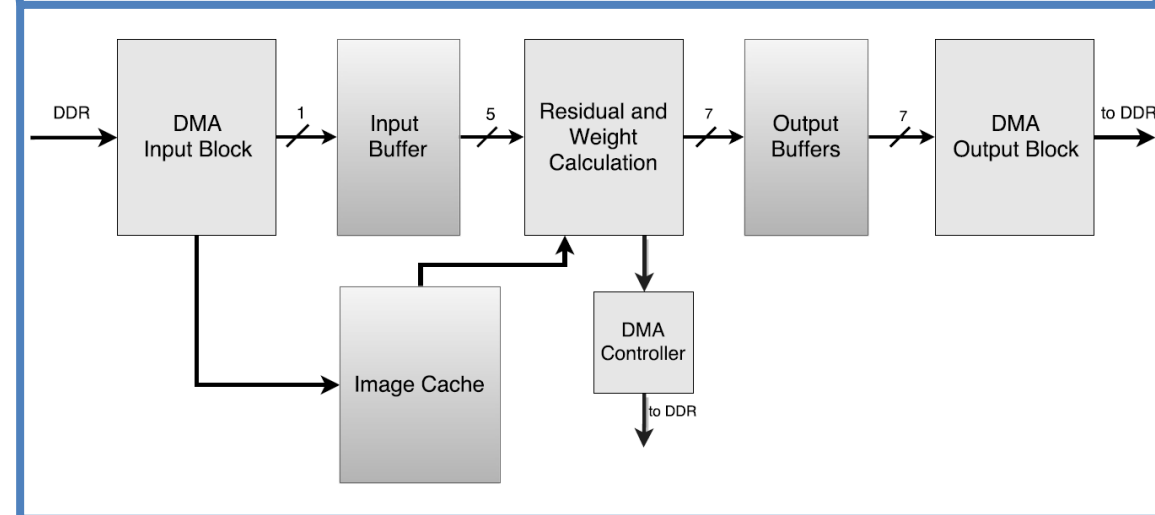
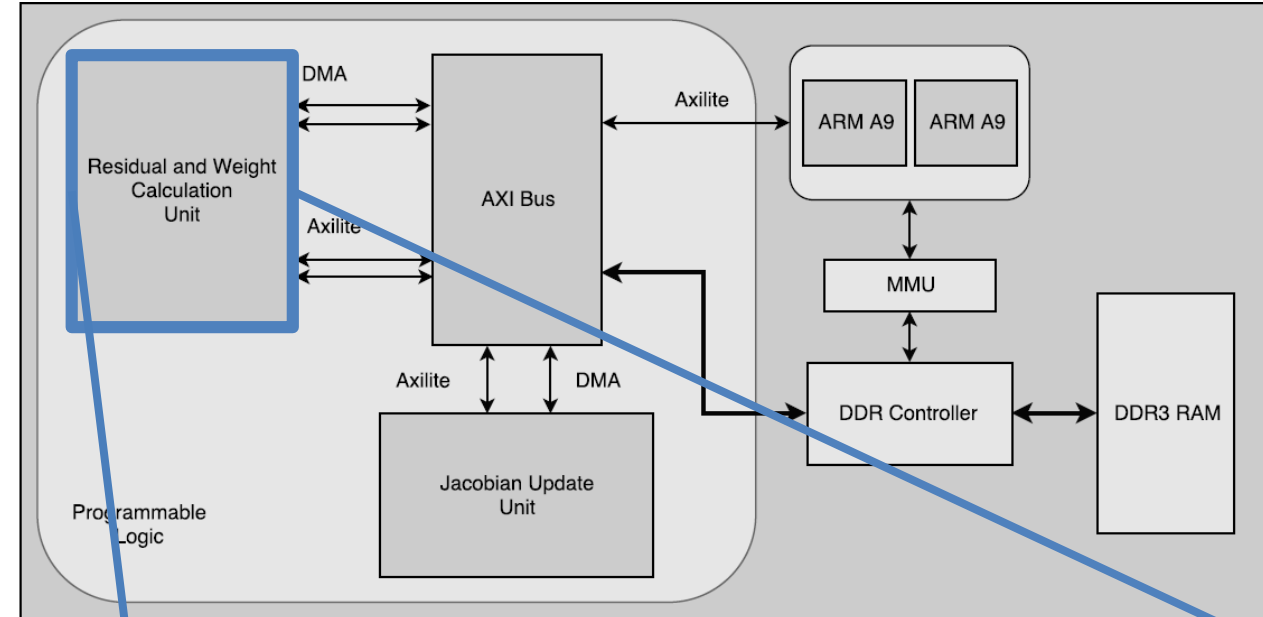
Proposed Solution

- Based on LSD-SLAM, a state of the art semi-dense SLAM algorithm
- Design custom hardware to offload tracking (pose estimation) with a significantly higher performance-per-watt
- Dual-core ARM processor handles mapping and complex control flow



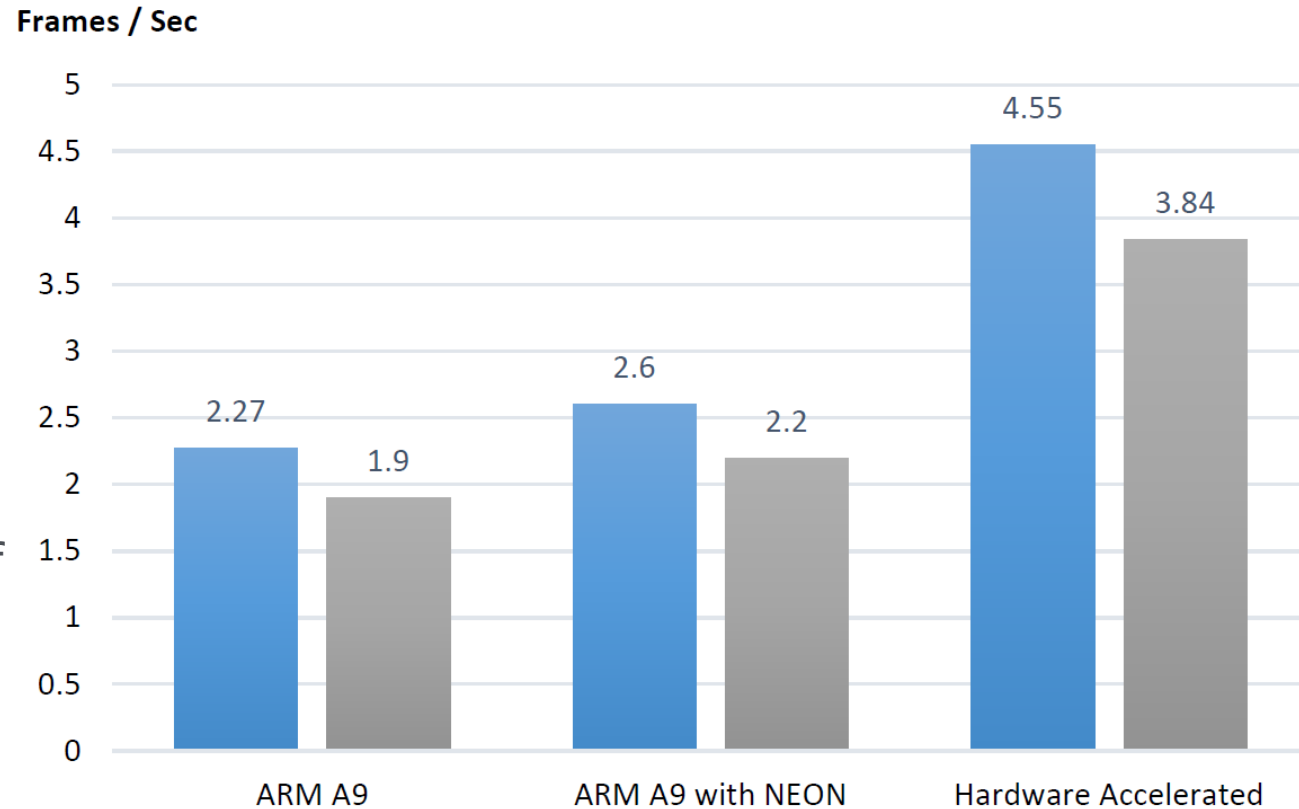
Proposed Solution – Pt. 2

- Prefetching the current frame in a local cache
- Interleaved all other input data in the DDR memory for optimised read access
- Reads and writes in batches to take advantage of AXI4 burst transfers
- Accelerator units and ARM cores operate in a common memory space



Evaluation


- Full SLAM system running on a Zynq-7020
- Overall, achieved a 2X speedup and more than 4.3X the energy efficiency
- Speedup is consistent for simpler (blue) to more complicated scenes with a larger number of textures (gray)
- Graph highlights sensitivity to the amount of information we want to recover



Poster Session

➤ If you are interested to learn more and discuss my work, look for my poster in the afternoon session that starts at **3.15pm**.

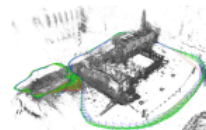
Semi-Dense SLAM on an FPGA-SoC
 Konstantinos Boikos and Christos-Savvas Bougias
 Department of Electrical and Electronic Engineering, Imperial College London
 { k.boikos14, christos.savvas.bougias@imperial.ac.uk }



Introduction

- Simultaneous Localisation and Mapping
- Integrate a series of observations to reconstruct an environment
- Simultaneously track the camera's position and trajectory
- Fundamental to Autonomous Robotics
- Has to be on board the device with high computational complexity
- Emerging applications require unprecedented runtime and accuracy embedded CPU/CPU+camera solutions
- Current Solutions reduce quality and infrastructure runtime, or use offboard processing

LSD-SLAM



How LSD-SLAM

- Direct Photometric Tracking by whole-image alignment
- Semi-dense Reconstruction and Global Graph based optimization
- To recover pose, the photometric error for pose ξ_w is minimised [2]

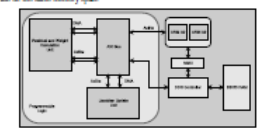
$$R(\xi_w) = \sum_{p \in C_{t-1}} \sum_{p' \in C_t} \frac{I(p, \xi_w) - I(p', \xi_w)}{I(p, \xi_w) + I(p', \xi_w)}$$

is the photometric residual and σ^2 the residual's variance

System Architecture

Thinking how to be fast and accurate for robot operation

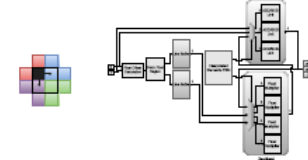
- FPGA: Vision tracking task was offloaded to two custom hardware units
- CPU: Handle mapping, global optimization and control flow
- Both operate on the same memory space



Residual Calculation Unit - Part 2

Gradient Interpolation Unit

- Calculate gradients for a pixel region and interpolate both the gradient and the intensity value
- Each pixel is an 8 bit value
- Cache has a line with a size of 256 bits to track appearance of pixels with a high throughput

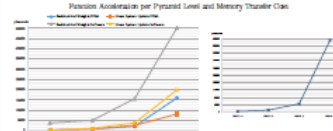
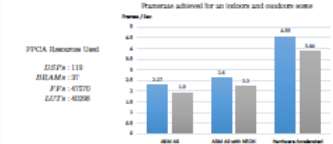


Jacobian Calculation

- One called when an optimization step is accepted
- Calculate Jacobian and Hessian to constrain the Linear System to solve
- Use a similar architecture as the other other two blocks. Separated because it is called less frequently

Evaluation

- Both using Virtuo 11120 on a Zynq 1010 SoC and run using a 100MHz
- Achieved 2.3 x improvement in performance and more than 2x in energy over an optimized Software only version running on a dual-core ARM A9
- Estimated CPU power under 50% load is 1.53 Watts, FPGA power is 0.11 Watts

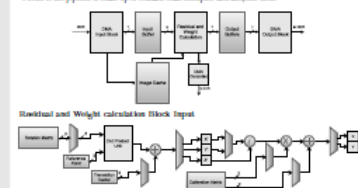



FPGA Resource Used

EEFPA - 119
EEAMA - 20
FFA - 41520
EEFA - 60288

Residual Calculation Unit - Part 1

- Handle batches of input and output as a time to hide memory access latency
- Memory Streams Interleaved → Higher performance and energy efficiency
- Most of the pipeline is made up of reusable flow multipliers and add/sub units



Row/col and Weight interleaved Block Input

Results & Conclusions

- The work demonstrated that an FPGA SoC is capable of bringing advanced state-of-the-art SLAM algorithms to the embedded space
- More than 4x performance on average as a resolution of 300 x 240, with an estimated power of 2.5 Watts
- Current bottleneck is memory bandwidth and latency
- Future work should focus on an algorithm redesign to allow a remaining paradigm for the hardware

References

[1] J. Engel, T. Schar, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in European Conference on Computer Vision (ECCV), September 2014.

Thank you!