“Driving” Innovation on Arm

Lessons learned porting open source autonomous driving stack

Liyou Zhou
Sep 2019
Welcome to the Autoware Foundation.

The Autoware Foundation is a non-profit organization supporting open-source projects enabling self-driving mobility. The Autoware Foundation creates synergies between corporate development and academic research, enabling autonomous driving technology for everyone. Your contribution is essential.
Components in Autoware

Analysis and Diagram produced by Jiang Xiang (Arm) jiang.xiang@arm.com
Components of a AD system

The main functions and algorithms

Localization

• Lidar localization
  • NDT matching
• Visual-inertial localisation

Perception

• Visual Object Detection
  • CNN
• Lidar Object Detection
  • Euclidean Clustering
• Traffic Light Detection
  • CNN
• Lane detection
  • Computer Vision

Planning

• Mission Planning
• Motion Planning
  • A* or State Lattice
Perception: Visual Object/Traffic Light Detection

Convolutional Neural Networks

- **Algorithm**
  - Large number of floating point/integer arithmetic operations
  - Fixed latency

- **Hardware**
  - Cortex-A ArmV8 CPU with NEON
  - Mali GPU with OpenCL acceleration
  - Arm Machine Learning Processor
    - 8 bit integer accelerator
    - CNN and RNN graph runner

- **Software**
  - ArmNN
  - Arm Compute Library
  - Mali SDK

Autoware Object detection node running on Arm target. Using ArmNN and OpenCL backend for acceleration. This work is carried out by Eliot Lim (Arm). See the demo later in the demo area.
Perception: Object Detection Node

Yolo-tiny running in a ROS node using ArmNN inference engine
Perception: Traffic Light Detection

The current approach

HD map approach

• Rely on HD map to provide exact location of light bulbs
• Manually calibrate against sensor intrinsic and extrinsic

HD map + CNN

• Rely on HD map to provide Region of Interests
• Use a CNN to do detection and classification within the RoI

Disadvantages

• Not robust against the posture of vehicle/camera
• Cannot deal with temporary lights or outdated map
• Cannot deal with varying shapes and sizes of traffic lights
Perception: Traffic Light Detection
Proposal for Improvement

New flexible architecture

• Support different implementation in one architecture
• Allow maximum code reuse

Traffic light detection and classification

• Traffic light recognition and classification in a single CNN
• Maximum hardware acceleration
• Minimum latency due to transferring of data between stages
• Arm specific optimizations
  • Tflite compatible
  • 8-bit quantization
Perception: Euclidian Clustering
Grouping of points that are close together - Classification

- Down-sampling of input data
- K-d tree nearest neighbour search
  - implemented in FLANN library

Analysis carried out by Anouk Van Laer (Arm) anouk.vanlaer@arm.com
Localization: NDT Matching

Point cloud manipulation

- Single Thread on Arm
- CUDA acceleration available
- OpenMP implementation is not up to date
Future Work

Performance optimization

- Hardware Acceleration
  - SVE/SVE2
  - OpenCL
  - ML Processor
- Memory/Latency optimisation

New Algorithms

- CNN based lane detection
- LSTM for object detection
- Visual-inertial localisation

Simulation

- Enable running Autoware distributed on multiple pieces of hardware
- Enable full stack working with simulators

Arm Targets

- Maintain Autoware build for Arm targets, ensure all functionality work and continue to work.
- Maintain cross compilation
- Support new Automotive SoCs and reference platforms
Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה