Data-Efficient and Hardware Decentralized Visual SLAM

Jincheng Yu[1] and Feng Gao[1]

Abstract— Decentralized simultaneous localization and mapping (DSLAM) is essential to a multi-robot system, especially in environments lacking absolute positioning equipments like GPS. Visual based SLAM is a widely adopted solution in industry for its low cost and high flexibility. There are two essential components need to be efficiently deplyed on each agent: 1) Visual Odometry(VO) and 2) Place Recognition. However, both of these components require intensive computation and storage on embedded system. The place recognition task is usually done with CNN based methods. We adopt CNN as the VO to provide 6-D pose between different frames, for both intra-robot or inter-robot. Thus we can use the CNN accelerator based on FPGA to execute these two components. In this work, we propose a hardware-software co-design DSLAM framwork and use embedded FPGA to accelerator these two components.

We evaluate our framework on the hardware platform Xilinx ZU9 SoC and we can perform DSLAM in real time on each agent. We also evaluate our system on publicly available dataset.

I. Introduction

In recent years, with the development of the hardware and algorithms, the capabilities of a single agent have been greatly improved. To further expand the capabilities of intelligent robots, using several robots can accelerate many tasks, such as localization, exploration, and mapping. As simultaneous localization and mapping(SLAM) is an essential component in many tasks, it is important to do SLAM across different robots in many multi-agent applications. The camera is a widely used snesor in SLAM for its rich information and low cost. However, in many scenarios, communication is limited, so that there is no server or an agent can stably collect all of the visual data from each robot.

Therefore, to reduce communication requirements, the previous work [?] proposes a data-efficient decentralized SLAM(DSLAM) system. The DSLAM frame in [?] is illustrated in fig 1(a). It makes improvements in three typical components in the DSLAM system: 1) Using ORB-SLAM [?] in stereo configuration as the visual odometry (VO) algorithm which provides basic intra-robot pose estimation. 2) Using NetVLAD [?] algorithm to do place recognition which relates the current observation to previous scenes and other robots. 3) Using distributed Gauss-Seidel algorithm [?] as the optimization back-end which optimizes the intra-robot position and fuses the inter-robot locations and maps. Each agent executes the ORB-SLAM which contains three steps for each input frame: feature extraction, feature matching and RANSAC. The NetVLAD method can encode the camera frame to a short vector which can be transformed to the server or a central agent with low communication cost.

However, both ORB-SLAM and NetVLAD require tremendous computation and storage resources, and thus, the deployment of DSLAM on embedded system is challaged by the limited resources and power supply.

Though NetVLAD consumes huge computation, with the development of FPGA accelerators, we use the embedded CNN accelerator on FPGA [?] to perform NetVLAD for each frame. We also notice that there are also some previous works regression the 6-D pose directly from the input setero camera [?] or monocular camera [?], [?], With the development of CNN. We adopt Depth-VO-Feat [?] in DSLAM system to estimate the pose from the input monocular camera. Because Depth-VO-Feat is trained with stereo input frames and inferenced with monocular camera, the CNN method can provide absolute scale from monocular camera, and also be accelerated with our CNN accelerator. Thus we do not need to execute ORB-SLAM on embedded CPUs.

The proposed DSLAM framework is illustrated in fig 1(b). To make the DSLAM system more energy efficient and hardware friendly, we propose a novel hardware-software co-design DSLAM framework with the following contributions:

- We implement NetVLAD on an embedded SoC platform with CPU cores and FPGA fabric.
- We use an end-to-end CNN based method to estimate the 6-DoF pose between intra-robot successive frames and matched scenes between different robots.
- We demonstrate that our proposed hardware-software co-design decentralized SLAM system can achieve a similar accuracy with the current state-of-the-art DSLAM system without increase of communication.

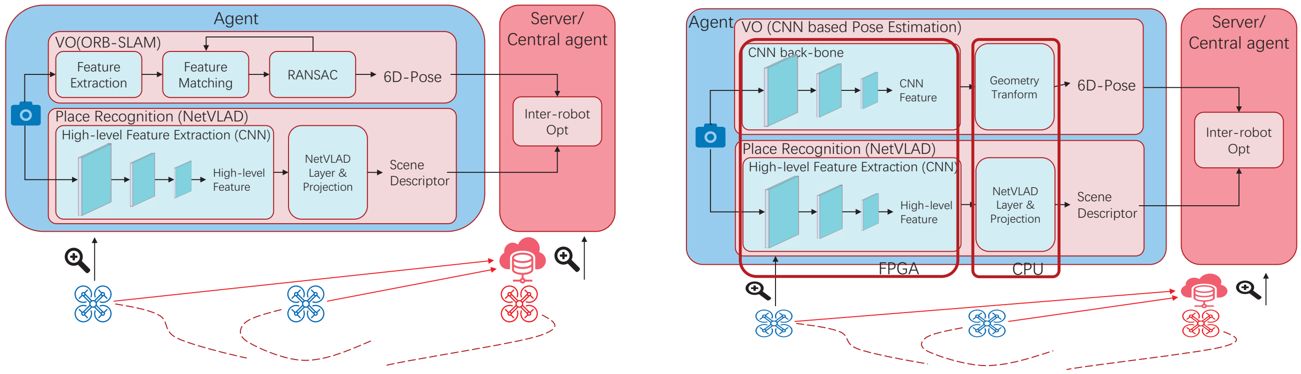
The rest part of this article is orgnized as follows. Section II will give the basic idea of CNN based methods and the hardware architecture of embedded FPGA. Section III will detail the implementation of our hardware-software co-design DSLAM system. The experiment result will be given in Section IV. Section V will conclude this paper.

II. Background and Motivation

A. CNN based methods in DSLAM

As described before, there are two essential components on each agent: 1) Visual Odometry (VO) and 2)

[1]Electronic Engineering Department, Tsinghua University, Beijing, China yjc16@mails.tsinghua.edu.cn



(a) DSLAM in [?]

(b) Our hardware-software co-design DSLAM.

Fig. 1. Overview of the DSLAM in [?] and our hardware-software co-design DSLAM. Each agent (blue drones) will send the result of 6-D pose estimation and scene descriptor to a server or a central agent (red server or drone in figure) to do inter-robot place recognition and optimization. We use CNN instead of feature points to do pose estimation so that we can use CNN accelerator to speed up the whole process.

Place Recognition.

1) Visual Odometry (VO): Visual odometry estimation is the task to infer ego-motion from a sequence of images, and is an essential component in SLAM system. Some feature-based SLAM systems have enjoyed great success, like ORB-SLAM[?] and ORB-SLAM2[?]. Recently, several studies have shown that these feature-based SLAM systems require high computing resources. Fang et al.[?] shows that the feature extraction stage is the most computation-intensive, consuming over 50% of the CPU resources. As FPGA is one of the most promising platforms as accelerator, the SLAM system on FPGAs has become a hot research topic. However, FPGA-accelerated feature extraction still consumes a lot of time and computing resources, which cannot be deployed simultaneously with an FPGA-accelerated neural network.

2) Place Recognition: The goal of place recognition is to calculate a given frame into a limited set of places. Each places can be encoded as a very short code which can be easy tranformed with low communication cost. Tranditional place recognition method usually translate the input frame as the aggregation of handcrafted feature point and local descriptors, like SIFT [?] or ORB [?], using vectorization techniques like bag-of-words (BoW) [?] or vector of locally aggregated descriptors(VLAD) [?].

Recent advances in the deep learning and the convolution neural network (CNN) enable powerful end-to-end mode for place recognition [?], [?], and the NetVLAD method is one of the most accurate method based on CNN. The NetVLAD algorithm based on VGG-16 model [?] consumes more than $80G$ operations for a single 300×300 input image (each operation means an addition or a multiplication). It is very challaging to deploy the NetVLAD on a tranditional embedded hardware platform.

B. Hardware architecture of Zync MPSoC

The Xilinx Zync MPSoC is a chip with ARM cores and FPGA fabric. The system is illustrated in section II-B. The ARM cores with an embedded Linux operation system are called Processing System (PS). The FPGA fabric is called Programmable Logic (PL). The peripherals like camera and communication unit (WiFi or others) are accessable with PS. The high-bandwidth on-chip AXI interface is used to communicate between PS and PL. PS and PL can also share the DDR to transfer large volume of data such as each frame of camera. Deephi CNN accelerator [?] is one of the state-of-the-art accelerators and is famous for high energy efficiency on various of CNN structure. We deploy the accelerator on the PL side of Zynq SoC.

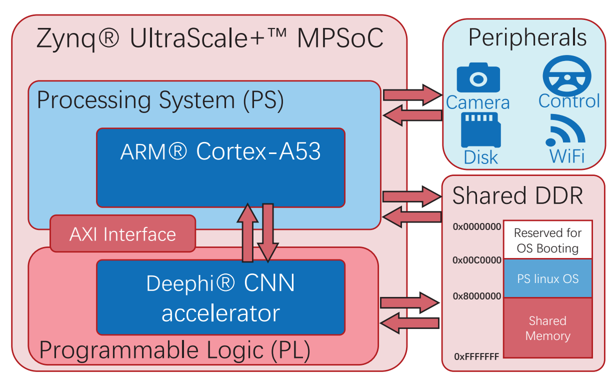


Fig. 2. Hardware architecture of Zynq SoC

Though FPGA can greatly improve the performance and energy efficiency of CNN inference, FPGA cannot efficiently calculate float-point number and requires fixed-point parameters and intermediate data in CNN.

C. Motivation

Though previous work [?] proposes data-efficient DSLAM system, it is difficult to implement the two essential components, VO and place recognition simultaneously on a communication-limited and energy-constrained embedded hardware platform on a real rebot. We propose this hardware-software co-design DSLAM system to use Xilinx Zynq MPSoC and Deephi accelerator to execute these two components on real system.

III. Hardware-Software Co-design DSLAM

Our hardware-software co-design DSLAM system contains two essential improvements in the pose estimation and the place recognition tasks. As illustrate in fig 1(b), both of these two components are divide into two stages: 1) CNN front end to extact features which is deployed to the CNN acclerator on PL and 2) geometric operations to present final results which is depoyed on the PS ARM core. To make full use of the Zynq MPSoC (illustrated in section II-B), we optimize the data follow for both of these components.

A. Pose Estimation

We adopt Depth-VO-Feat [?] in DSLAM system to estimate the pose from the input monocular camera. Monocular visual SLAM is a key issue in the field of robotics, while there are two challenging problems: 1) it's difficult and expensive to obtain accurate labeled data. 2) the methods that use monocular sequences in training always suffer from the scale-ambiguity problem, i.e. the actual scale of translations is missing and only direction is learned. In Depth-VO-Feat [?], we use image reconstruction loss as a self-supervised signal to train the convolutional neural networks, and jointly train two networks for depth and odometry estimation without external supervision, which can be used independently in testing phase. Besides, to fix scale-ambiguity issue, we use stereo sequences in training phase and monocular sequences in testing phase. With the known spatial relationship between the left and right cameras, our neural networks can learn the real world scale. Feature reconstrution loss is an additional supervision signal, used to improve the robustness of this framework. And we use depth smoothness loss to encourage the predicted depth to be smooth, which demonstrated success in prior works. Then the final loss becomes

$$L = \lambda_{ir}L_{ir} + \lambda_{fr}L_{fr} + \lambda_{ds}L_{ds}$$

, where L_{ir}, L_{fr} and L_{ds} are image reconstruction loss, feature reconstruction loss and depth smoothness loss respectively, λ_{ir}, λ_{fr} and λ_{ds} are the loss weightings for each loss term. The training framework is illustrated in section III-A.

In order to run efficiently on the FPGA platform, we use fixed-point arithmetic units in the hardware to replace the floating-point number format in GPU and CPU. Many previous works have shown that 8-bit quantization for weights and featuremaps can make the networks run faster on FGPA. Here we adopt the fixed-point finetune method in [?], in that we use the fixed-point number representation in the feed forward phase and keep floating-point number representation for backpropagation, and both weights and data will be re-quantized after each backpropagation. As the fixed-point method can leads to the accuracy loss of the model, We attempt several different quantization strategies to balance speed and accuracy, which will be shown in detail in section IV.

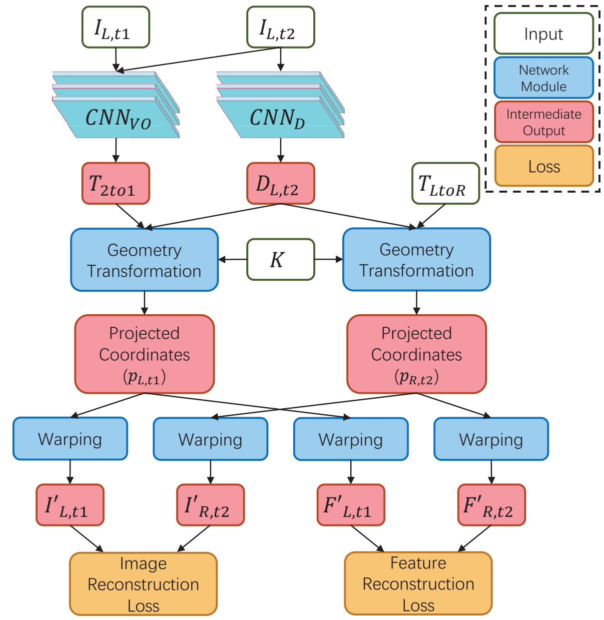


Fig. 3. Illustration of Depth-VO-Feat framework in training phase, where T_LtoR is the relative camera pose transformations between right and left views, and K denotes the known camera intrinsic matrix. CNN_{VO} and CNN_D can be used independently in testing phase.

B. Place Recognition

As described in section II, CNN has achieved great improvements in place recognition tasks.

IV. Experiments

The experiment results shows our proposed DSLAM system can perform in real time and achieves similar accuracy with previous work.

V. Conclusion

Acknowledgment

This work is not supported by any fund.