ROS 기반 2륜 차량 시스템을 위한 딥러닝 기반 Monocular Visual Odometry 적용

최병찬, 남해운

한양대학교

luwis93@hanyang.ac.kr, hnam@hanyang.ac.kr

Implementation of Deep Learning-based Monocular Visual Odometry on ROS-based 2WD system

Byung Chan Choi, Haewoon Nam

Hanyang University

요 약

본 논문은 Recurrent Neural Network (RNN) 기반 Monocular Visual Odometry 기법인 DeepVO를 ROS 기반 2륜 차량 시스템과 실내 주행 상황에 적용하기 위한 구현 방법을 제안한다. RNN 기반 딥러닝 네트워크인 DeepVO에 Monocular Visual Odometry를 학습시키기 위해 딥러닝용 GPU 서버 컴퓨터에서 KITTI 데이터셋을 사용하여 학습을 진행하였다. 학습된 네트워크를 별도의 ROS Host PC에 탑재한 후 ROS 기반 2WD 차량에서 전송되는 연속되는 실내 주행 이미지로부터 Odometry 연산을 수행하여 차량의 실내 위치 추정을 수행하였다. 본 논문은 주행 데이터셋에서 좋은 성능을 내는 딥러닝 기반 Visual Odometry 기법을 실제 주행 시스템에 적용한 결과와 구현 과정에서 발생하는 문제점을 파악하는 것을 목표로 하였다.

Ⅰ. 서 론

최근 딥러닝 기술의 급격한 발전으로 차량 및 로봇 위치 추정의 핵심 기술인 Visual SLAM과 Visual Odometry에 딥러닝을 적용하려는 연구가 활발히 진행되고 있다.[1][2][3] 이러한 딥러닝 기반 기법에 대한 활발한 연구는 딥러닝의 일반화 특성에 대한 높은 기대치에서 비롯된 것이라 할 수 있다.

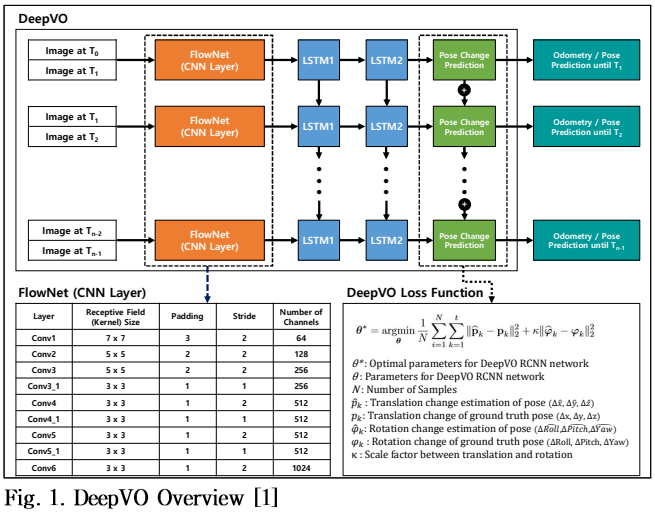
고전적인 Visual Odometry는 Feature Tracking, Point Cloud Tracking, Multi-view Geometry를 통해 카메라의 Pose 변화를 추정하고 이를 누적하여 이동 경로를 예측했다.[4][5][6] 하지만 기존의 방식은 Feature 추출 결과에 매우 의존적이다. 이로 인해 주행 환경 변화에 따른 Parameter Fine Tuning이 요구된다. Bundle Adjustment와 같은 최적화 기법을 사용하여 평균 Pose 추정 오차를 최소화시키지만 여러 주행 환경과 시나리오를 위한 일반화 특성을 확보하는 데에는 제한적이다.

딥러닝 기반 Visual Odometry는 기존의 방식과 달리 대용량의 주행 데이터셋에서 전방 이미지와 위치 정보 사이의 관계를 표현할 수 있는 모델을 Deep Neural Network 학습을 통해 얻어냄으로서 Visual Odometry와 관련 기능을 구현한다.[1][2][3] 다양한 광원 변화, 물체 이동, 회전 시나리오 등을 포함한 데이터셋에서 주행 이미지와 위치 정보 간의 관계를 표현할 수 있는 일반화된 모델을 도출하기 때문에 Robustness가 고전적인 방식보다 좋은 편이다.

대부분의 연구는 KITTI, nuScenes, CARLA와 같은 주행 데이터셋 내에서 학습 및 성능 평가에 집중하고 있다. 그러나 학습된 딥러닝 네트워크를 실제 로봇 또는 차량에 탑재하여 실 주행 상황에서의 성능 평가는 상대적으로 저조한 편이다. 본 논문에서는 [1]에 제시된 방법으로 RNN 기반의 Monocular Visual Odometry 네트워크 DeepVO를 구현하고 실제 2WD 소형 차량 Odometry에 탑재하기 위한 구현 과정, 실내 주행 성능 및 제한사항을 파악하는 것에 중점을 두었다.

Ⅱ. DeepVO 개요

DeepVO는 2017년 IEEE International Conference on Robotics and Automation (ICRA) 에서 발표된 딥러닝 기반 Visual Odometry 기법이다.[1] DeepVO는 연속적으로 입력되는 이미지 데이터를 시계열 데이터 (Sequential Data)로 취급하여, Monocular Visual Odometry 문제를 RNN 네트워크를 통해 해결하였다는 점이 특징이다. 연속적인 이미지로부터 Optical Flow를 추정할 수 있는 FlowNet과 시계열 데이터를 처리할 수 있는 RNN를 합친 Deep RCNN을 제시하여 KITTI 데이터셋에서 고전적인 Visual Odometry 기법보다 Robust하고 정확한 결과를 제시하였다.[1][2]



FlowNet (CNN Layer)

Layer	Receptive Field (Kernel) Size	Padding	Stride	Number of Channels
Conv1	7 x 7	3	2	64
Conv2	5 x 5	2	2	128
Conv3	5 x 5	2	2	256
Conv3_1	3 x 3	1	1	256
Conv4	3 x 3	1	2	512
Conv4_1	3 x 3	1	1	512
Conv5	3 x 3	1	2	512
Conv5_1	3 x 3	1	1	512
Conv6	3 x 3	1	2	1024

DeepVO Loss Function

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{t} \|\hat{p}_k - p_k\|_2^2 + \kappa \|\hat{\varphi}_k - \varphi_k\|_2^2$$

θ^*: Optimal parameters for DeepVO RCNN network
θ: Parameters for DeepVO RCNN network
N: Number of Samples
$\hat{p}_k$: Translation change estimation of pose ($\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{z}$)
p_k: Translation change of ground truth pose ($\Delta x, \Delta y, \Delta z$)
$\hat{\varphi}_k$: Rotation change estimation of pose ($\Delta\hat{Roll}, \Delta\hat{Pitch}, \Delta\hat{Yaw}$)
φ_k : Rotation change of ground truth pose (ΔRoll, ΔPitch, ΔYaw)
κ : Scale factor between translation and rotation

Fig. 1. DeepVO Overview [1]

DeepVO는 연속되는 이미지 2개를 쌓아서 네트워크에 입력하고, 목표값으로 각 이미지에 해당되는 시간의 Pose 변화량을 학습하게 한다. 그리고 Long Short Term Memory (LSTM)를 통해 현재 시간대에서 추정한 정보를 다음 시간대에 전달하여 연속적인 Pose 변화량 추정을 할 수 있도록 한다. Pose에서 Translation 변화량인 Δx, Δy, Δz와 Rotation 변화량인 ΔRoll, ΔPitch, ΔYaw는 서로 단위와 범위가 다른 것을 감안하여 Loss Function을 Translation Error, Rotation Error를 분리하고 Weight를 적용한 Weighted Loss Function을 사용한다.

Ⅲ. 주행 시나리오 데이터셋을 이용한 RNN 학습

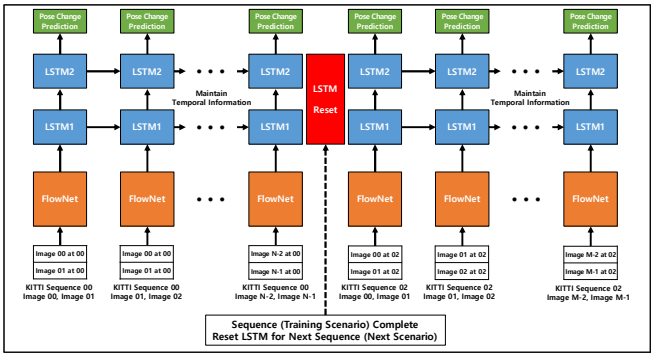


Fig. 2. Stateful RNN Training for DeepVO

DeepVO와 같은 RNN 구조의 딥러닝 네트워크를 학습시키기 위해서는 해당 RNN 문제가 Stateless case인지 Stateful case인지 확인해야한다. Stateless case의 경우 각 Batch가 독립적인 시나리오기이기 때문에 다음 Batch에 영향을 주지 않는다. Stateless RNN을 학습할 때에는 Batch마다 LSTM을 초기화하며 데이터셋을 Shuffling한다. [7] 그러나 Stateful case의 경우에는 각 Batch가 전체 시나리오의 일부이기에 다음 Batch에 영향을 준다. 이로 인해 각 Batch에 할당된 시나리오가 독립적이지 못하다. Stateful RNN을 학습할 때에는 이전 Batch에서 배운 정보의 흐름을 유지하기 위해 시나리오 기준으로 LSTM을 초기화하고 데이터셋을 Shuffling하지 않고 순차적으로 제공한다.[7]

KITTI와 같은 주행 데이터셋에서 학습할 경우 Batch 단위로 가져오는 데이터는 각 주행 시나리오의 일부이기 때문에 하나의 주행 시나리오가 끝나기까지 Stateful RNN case로 취급하여 DeepVO를 학습해야한다. 본 논문에서는 LSTM이 KITTI 데이터셋 시나리오 1개 완료마다 초기화되게 학습하였다.

Ⅳ. DeepVO ROS 기반 시스템 탑재

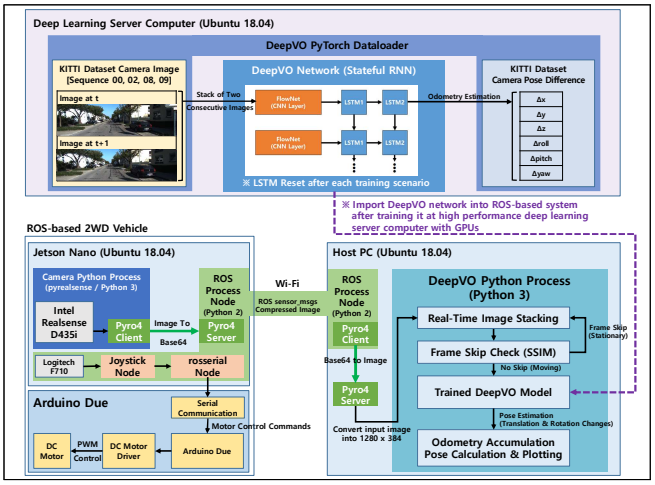


Fig. 3. System Architecture for DeepVO on ROS-based 2WD System

본 논문에서는 Nvidia 2080 Ti가 장착된 딥러닝용 GPU 서버 컴퓨터에서 KITTI 데이터셋과 PyTorch를 이용하여 DeepVO 네트워크를 학습시킨 후 ROS Host PC에 탑재하였다. ROS Host PC는 ROS 2WD 차량에서 연속적으로 전달하는 이미지를 입력받아 학습된 네트워크에 전달하여 Pose 변화량을 추정하고, 결과값을 누적해서 차량의 이동거리와 위치를 추정한다. Python3 기반 딥러닝 프로세스와 Python2 기반 ROS1 사이의 데이터 교환을 구현하기 위해 프로세스간 데이터 교환 라이브러리인 Pyro4를 사용하여 ROS 네트워크와 병행으로 작동하는 Python 프로세스 네트워크를 구성하였다.

DeepVO만 사용해서 위치를 추정할 경우 정지 상황과 후진 상황에서 위치 추정 결과가 발산하거나 추정이 틀리는 것을 볼 수 있다. 왜냐하면 학습에 사용한 KITTI 데이터셋에는 후진과 정지 상황이 전진 주행 보다 적게 배정되어있기 때문이다.

정지 상황에서 위치 추정이 발산하는 것을 막기 위해서 Structural Similarity Index Measure (SSIM)을 이용하여 연속된 이미지가 동일한지 여부를 파악하고 동일할 경우 정지 상태인 것을 판단하는 Frame Skip을 도입해야한다. Frame Skip 기능을 도입함으로서 정지 상태에서 불필요한 Pose 추정 오차가 누적되는 것을 방지하여 전체 시스템을 안정화 시킬 수 있다. 후진 상황을 학습하기 위해서는 데이터셋을 반대로 실행하여 학습시키는 방법이 있으나 학습 소요시간이 증가한다는 단점을 내포하고 있다.

Ⅴ. 최종 구현 결과 및 실험 결과

KITTI 데이터셋에서 장거리 데이터셋 00, 02, 08, 09를 Training Set으로 설정하여 전진, 우회전, 좌회전 등 다양한 주행 시나리오를 딥러닝 네트워크에 학습시켰다. 그리고 Validation으로 비교적 단거리 데이터셋인 01, 03, 04, 05, 06, 07, 10을 선정하였다. 사용된 이미지 데이터의 비율은 Training 67.59%, Validation 32.41%이다. 네트워크 학습을 위해 Learning Rate는 0.001로 설정했으며, Adagrad Optimizer를 사용하였다. Pre-trained된 FlowNet 대신 학습되지 않은 초기 CNN 모델을 사용하여 학습의 전 과정을 관찰하였다. 학습을 통해 생성된 DeepVO 모델을 ROS 기반 2륜 차량의 Host PC에 탑재하였으며, 실내 주행하는 2륜 차량의 Odometry 추정을 수행하였다.

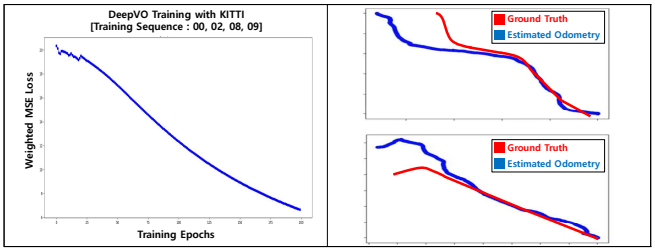


Fig. 4. Training Loss (Left) / Indoor Driving Test (Right)

Pre-trained된 FlowNet이 아닌 순정 상태의 초기 CNN 네트워크를 사용하기 때문에 Loss가 완만하게 감소하면서 Pose 변화량 추정이 학습되는 것을 볼 수 있다. 실내 주행 결과 Pose 추정 결과가 Ground Truth와 전반적으로 형태는 유사하나 4륜 Ackermann 구조의 차량 조향을 기준으로 학습된 모델이 2륜 차량의 Rotation Estimation과 Scale Estimation에 대해 부족한 면모를 보이고 있다. 그리고 4륜 Ackermann 구조 차량은 회전을 위해 전진이 병행되기에 데이터셋에서도 대부분 Pose 변화량에 전진이 반영되어있고 그에 맞춰서 네트워크가 학습되었다. 이로 인해 Pose 변화량을 4륜 차량 구조와 같이 전진이 병행된 형태로 추정하려는 경향이 있다.

Ⅵ. 결론

딥러닝은 여러 연구에서 다양한 주행 시나리오 학습을 통한 일반화된 Visual Odometry 모델을 도출하여 데이터셋에서 고전적인 Visual Odometry 기법보다 Robust한 결과를 보여주었다. 그러나 딥러닝 학습은 데이터셋의 구성에 의존적이기 때문에 데이터셋에 포함되지 않은 시나리오에 대해 취약한 점을 보인다. 그리고 딥러닝 기반 Visual Odometry는 장착한 차량 또는 로봇의 Motion Model, 사용하는 카메라의 Instrinsic Parameter 등을 암묵적으로 학습하기 때문에 사용 HW나 카메라 해상도가 변경될 경우 성능이 저하될 여지가 있다. 이를 보완하기 위해 SSIM와 같은 고전적인 영상처리 기법을 딥러닝 네트워크와 연계하여 시스템을 구현할 수 있다.

ACKNOWLEDGMENT

이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2019M3F6A1106108).

참 고 문 헌

[1] S. Wang, R. Clark, H. Wen and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks," 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 2043-2050

[2] A. Dosovitskiy et al., "FlowNet: Learning Optical Flow with Convolutional Networks," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 2758-2766

[3] R. Wang, S. M. Pizer and J. Frahm, "Recurrent Neural Network for (Un-)Supervised Learning of Monocular Video Visual Odometry and Depth," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019, pp. 5550-5559

[4] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," in IEEE Robotics & Automation Magazine, vol. 18, no. 4, pp. 80-92, Dec. 2011

[5] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015

[6] M. Labbé, F. Michaud. "RTAB Map as an Open Source Lidar and Visual Simultaneous Localization and Mapping Library for Large-Scale and Long-Term Online Operation," Journal of Field Robotics, vol. 36, no. 2, pp. 416 - 446, Mar. 2019

[7] M. A. Yilmaz and A. Murat Tekalp, "Effect of Architectures and Training Methods on the Performance of Learned Video Frame Prediction," 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 2019, pp. 4210-4214