Learning-based Pose Estimation for Visual Odometry

Jingtian Yan
Electrical and Computer Engineering Department
jingtiay@andrew.cmu.edu

Troy Gu
Mechanical Engineering Department
zgu2@andrew.cmu.edu

Abdullatif Al Alsheikh
Mechanical Engineering Department
aalsheik@andrew.cmu.edu

1 Problem Statement

The project explores deep learning applications in the field of structure from motion(SfM) and visual odometry(VO). Specifically, the proposed solution is expected to estimate the relative motion of the robot from images obtained from an on-board camera, using a CNN-based architecture.

Traditionally such a problem is solved by detecting and matching features (using methods like SIFT [1] and SURF [2]), from which correspondences are created and then used to interpret motion of the camera. However, the performance of such methods usually suffer from lack of feature correspondences due to the presence of repetitive structures, texture-less objects, etc [3].

In recent years, Convolutional Neural Networks (CNNs) have been proved to be able to outperform conventional methods in completing various computer vision tasks such as image classification and object recognition. In this project, we will be implementing a CNN-based network to learn ego-motion from a sequence of frames.

2 Data

The project plans to use open-source datasets collected from the real world, such as the KITTI odometry dataset [4]. This dataset contains 22 stereo sequences that were collected from a moving car in the urban area. The images are available in lossless png format. The training data (sequence 00-10) is labeled with ground-truth trajectories.

Furthermore, there are other open source datasets to explore that offer various contexts which can be used to test visual odometry methods. The context here can refer to the different places where the datasets were collected (urban areas, indoors, etc) or the vehicle/device used to collect the data (Drones, rovers, etc), amongst other various settings that might provide a specific case of interest.

3 Method

The proposed implementation [5] includes two subnetworks for estimating depth and motion respectively. The depth estimation network is comprised of a standard conv/deconv network with RELU activation on the last layer, operating on a single image frame. It will first extract the features from the images using the convolutional neural network. Then, it will use a decoder to decode those features into the depth of each pixel. By using the intrinsic parameters of the camera, those pixels can be project into points cloud.

The motion network consists of a similar conv/deconv network structure, but with an input of two consecutive frame and independent weights. The coarsest feature map from the motion network will be fed through two fully connected neural network layers for motion estimation. The network

can be trained either in a self-supervised fashion, by minimizing photometric error between the first frame and the second frame warped onto the first, or in a supervised way given ground-truth camera trajectories.

4 Plan

The project aims for a successful implementation of the proposed algorithm/network. The majority of the project will be written using Python, with the help of packages such as PyTorch/TensorFlow to realize the CNN structures. The program will be first tested on the dataset used in the paper to validate its performance. Testing on other datasets mentioned above is also possible if time permits.

References

[1] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, **60**(2), 91-110.

[2] Bay, H., Tuytelaars, T., & Gool, L. V. (2006, May). Surf: Speeded up robust features. *In European conference on computer vision* (pp. 404-417). Springer, Berlin, Heidelberg.

[3] Melekhov, I., Ylioinas, J., Kannala, J., & Rahtu, E. (2017, September). Relative camera pose estimation using convolutional neural networks. *In International Conference on Advanced Concepts for Intelligent Vision Systems* (pp. 675-687). Springer, Cham.

[4] Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2015). The KITTI vision benchmark suite. URL http://www. cvlibs. net/datasets/kitti, 2.

[5] Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., & Fragkiadaki, K. (2017). Sfm-net: Learning of structure and motion from video. *arXiv preprint* arXiv:1704.07804.