A SLAM Algorithm Based on Multi-Constraint Learning Model

Haibin Shi*, Menghao Guo, Yuanbin Zou and Zhi Xu

College of Information Science and Engineering, Northeastern University

*E-mail: shihaibin@ise.neu.edu.cn

Abstract. Unsupervised Learning based SLAM algorithm has lately drawn significant attention for its potential in label-free leaning ability and robustness to camera parameters and environmental variations. In order to achieve better robustness and accuracy, a multi-constraint learning model is proposed. In contrast to traditional geometry-based methods, multi-constraint unsupervised learning models optimize the photometric consistency over image sequences by warping one view into another; make the Network learning more geometrically information. A lot of experiments on the KITTI data set show that our model is superior to previous unsupervised methods and has comparable results with the supervised method.

1. Introduction

Simultaneous Localization and Mapping (SLAM) [1] is a robot equipped with vision, laser, visual odometry and other sensors, to achieve self-localization process of constructing a map of an unknown environment at the same time, play a key role in autonomous robot navigation tasks. Although the sensors used by SLAM have many types such as LIDAR and vision, the processing generally consists of two parts: front-end inter-frame estimation and back-end optimization. The front-end inter-frame estimation solves the motion estimation of the robot in the time interval of acquiring the two frames of sensor information before and after [2, 3], while the back-end optimization solves the optimization problem of the historical trajectory after the robot detects the path closed loop [4].

Due to the different sensors, the SLAM algorithm is divided into LIDAR SLAM and visual SLAM. Among them, with the rapid development of visual algorithms and hardware, visual SLAM has received wide attention because of the large amount of information received by visual sensors and the wide range of applications.

The visual SLAM algorithm can be reclassified into two groups based on the technology and framework [5]: geometry-based and learning-based methods. Geometry-based methods are usually solved by shallow features or luminosity re-projection and online error minimization. Learning-based method has lately drawn significant attention for its potential in label-free leaning ability and robustness to camera parameters and environmental variations.

The traditional visual SLAM algorithm, for example, ORB-SLAM [6] uses ORB features, and feature matching and relocation based on ORB descriptions have better viewing angle invariance. In addition, the feature matching of the new 3D points is more efficient, so the scene can be expanded more timely. But the constructed map is a sparse point cloud map. Only a part of the feature points in the image is retained as a key point, and it is fixed in the space for positioning, and it is difficult to depict the existence of obstacles in the map.

Kendall et al. [7] first use the deep learning method on the SLAM algorithm directly regress the
camera’s world pose from RGB images with the convolutional neural network. However, the cost of collecting ground truth poses limits the application of such methods.

The first architecture that achieved unsupervised learning of ego-motion from the video is SfMLearner proposed by Zhou et al [8]. SfMLearner takes consecutive temporal images to predict both depth and ego-motion with view synthesis as supervision. However, similar to geometric approaches, SfMLearner can only observe ego-motion in a relative scale from monocular image.

The main idea behind them is to optimize the photometric consistency over image sequences by warping one view into another. It is a well-known fact that simply minimizing such an error might not effectively capture the relation between pixels. Further, minimizing the photometric loss does not ensure proper pixel correspondences, which is a key factor for accurate depth and pose estimations. In contrast, we present a novel multi-constraint learning SLAM algorithm with depth warp constraints loss based on unsupervised deep convolutional neural networks. We trained two convolutional network end-to-end to calculate depth and ego-motion from a continuous, unlabeled pair of images. The ego-motion is estimated using image projection constraints and depth warp constraints as supervisory information.

2. Multi-constraint learning SLAM algorithm

The basis of our approach is based on the nature of the 3D scene geometry. The intuitive explanation is that most natural scenes consist of rigid static surfaces, i.e. roads, houses, trees, etc. The 2D image motion project between video frames can be determined entirely by depth structure and camera motion. Traditional methods have been able to calculate depth from binocular images and then use 2D image projection to construct VO. Therefore, we use a deep convolution network to grasp the above intuition, build our deep learning network. Our deep learning network consists of three parts:

1) By constructing a convolutional neural network called Depth net, left and right photometric consistency is used to determine the left and right photometric twist error loss function, called Spatial Photometric Loss. At the same time, a corresponding Depth Map is obtained from the left and right image.

2) By constructing a convolutional neural network called Pose net, 2D image projection constraints of two consecutive frames is used to determine Temporal Photometric Loss, and at the same time, ego-motion is learned from the Pose net.

3) Based on the convolutional neural network of Depth net and Pose net, the depth images of two consecutive frames predicted by part 1 is used to determine a Depth Warp Loss by depth warp Constraints to assist the constraints Spatial Photometric Loss and Temporal Photometric Loss.

In summary, we make the following contributions:

(1) An unsupervised two-part cascade network structure to learn the depth map and ego-motion separately without being affected by scale ambiguities.

(2) Multi-constraint learning SLAM algorithm using all the constraints available in spatial and temporal image pairs to improve the accuracy of predicts ego-motion.

(3) A novel depth warp constraints is proposed, to adaptively solve the rigid flow and object motion of the scene, which significantly improves depth and visual odometry accuracy.

3. Three partial constraints

This section introduce the Network Architecture of SLAM algorithm based on multi-constraint (shown in Figure. 1), which learning $T_{s1} \in SE3$ from two consecutive frames and learning the depth image $d_1$ and $d_2$ from the left and right images of the binocular camera at the same time. Our example uses a divide-and-conquer strategy. A new cascade structure consisting of three partial constraints is designed to adaptively solve the rigid flow and object motion of the scene. Therefore, the global constraints can be gradually improved, making our complete learning process a decomposed and easy to learn way.
3.1. Spatial Photometric Loss
The image reconstruction loss between the warp view $I'_{R,1}$ and the real view $I_{R,1}$ is calculated as a supervised signal for training Depth net. The image construction loss called Spatial Photometric Loss is represented by the following formula:

$$I_{SP} = \sum |I_{R,1} - I'_{R,1}| + |I_{R,2} - I'_{R,2}|$$  \hspace{1cm} (1)

We can use polar line geometry to obtain the projected coordinates from the reference view $I_{l,1}$ warp to the view $I'_{R,1}$. $p_{l,1}$ is the homogeneous coordinate of the pixel in the reference view $I_{l,1}$. Warp view $I'_{R,1}$ can be synthesized from the reference view $I_{l,1}$ using the differentiable bilinear interpolation mechanism (warping) proposed in [9].

$$p'_{R,1} = (K \ast (T_{lR} \ast (D_1 \ast K^{-1})) \ast p_{l,1}$$  \hspace{1cm} (2)

$$p'_{R,2} = (K \ast (T_{lR} \ast (D_2 \ast K^{-1})) \ast p_{l,2}$$  \hspace{1cm} (3)

3.2. Temporal Photometric Loss
The image reconstruction loss between the warp view $I'_{R,2}$ and the real view $I_{R,2}$ is calculated as a supervised signal for training Depth net and Pose net. The image construction loss called Temporal Photometric Loss is represented by:

$$I_{TP} = \sum |I_{R,2} - I'_{R,2}|$$  \hspace{1cm} (4)

We can use the direct method to obtain the projected coordinates $p'_{R,2}$ from the reference view $I_{R,1}$ to the warp view $I'_{R,2}$. $p_{R,2}$ is the homogeneous coordinate of the pixel in the reference view $I_{R,2}$. Warp view $I'_{R,2}$ can be synthesized from the reference view $I_{R,1}$ using the differentiable bilinear interpolation mechanism (warping) proposed in [9].

$$p'_{R,2} = (K \ast (T_{R2} \ast (D_2 \ast K^{-1})) \ast p_{R,2}$$  \hspace{1cm} (5)
3.3. Depth Warp Loss
The image reconstruction loss between the warp view $D'_w$ and the real view $D'_2$ is calculated as a supervised signal for training Depth net and Pose net. The image construction loss called Depth Warp Loss is represented by:

$$L_{DW} = \sum |D'_w - D'_2|$$  \hspace{1cm} (6)

We can use the constraint formula [10] between the depth maps corresponding with two consecutive frames to obtain the projected coordinates from the reference view $D'_1$ warp to the view $D'_2$.

3.4. Finally Loss
In summary, finally loss function contains two distinguishable operations that allow for gradient propagation for Depth net and Pose net training. These two operations are polar line geometry transformations and warp. The former defines the correspondence between pixels in two views, while the latter synthesizes the image by distorting the real-time view. The final loss function becomes:

$$L = \lambda_{SP} L_{SP} + \lambda_{TP} L_{TP} + \lambda_{DW} L_{DW} + \lambda_{ds} L_{ds}$$  \hspace{1cm} (7)

Where, $\lambda$ is the loss weight of each loss item and is obtained through training and fine tuning. In order to obtain smooth depth prediction, following the method adopted by Zhan et al. [11], we encourage depth to be localized by introducing edge-aware smoothness. Edge-aware smoothing loss is the formula:

$$L_{eA} = \sum_{x,y} \left( \frac{1}{w_h} \sum_{m,n} D_{x,y} \left| \frac{\partial}{\partial x} D_{x,y} \right| + \frac{1}{w_v} \sum_{m,n} D_{x,y} \left| \frac{\partial}{\partial y} D_{x,y} \right| \right)$$  \hspace{1cm} (8)

4. Experiments and Results
In this section, we present a wide range of experiments for evaluating the performance of our proposed framework. We compare our approach in the KITTI dataset [12, 13] with the state of the art SLAM algorithm.

4.1. Experiments
The network models were implemented with the Caffe [14] framework and trained with NVIDIA TITAN Xp GPUs and Intel Core i7 2.7GHz CPU. Adam optimizer [15] was employed to train the networks for up to 20000 epochs with parameter $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The initial learning rate for all trained networks is 0.001, and we will manually reduce when training losses converge. Our work does not involve any data expansion. In the loss weights in our final loss function, we have empirically found that the main parameters in stability training are:

$$\begin{bmatrix} \lambda_{SP} \\ \lambda_{TP} \\ \lambda_{DW} \\ \lambda_{ds} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0.01 \\ 10 \end{bmatrix}$$  \hspace{1cm} (9)

Since our network only needs paired binocular camera data for training, and the test uses the image of the monocular camera for testing, we use the raw data part of the KITTI dataset, the original image size is 1242x375 pixels. However, because the picture quality is too high, increasing the training cost, we adjust the Caffe input data parameter during training. The input image size is 608x160 pixels.

As mentioned in the previous paragraph, our framework uses monocular data for testing. To evaluate our SLAM algorithm performance and compare it with previous method [8, 11, 16], we
follow [8] by test the Pose net on the official KITTI Odometry training set. The details of the test set are: The KITTI Odometry Split [12, 13] contains 11 driving sequences with publicly available ground truth camera poses. We trained our system on the Odometry Split in accordance with [8] (no fine-tuning of Eigen Split). The split of sequence 00-08 (sequence 03 is not available in KITTI raw data) is used for training, while 09-10 is used for evaluation. The training set consists of 8 sequences with 19,600 time stereo pairs.

4.2. Result and Analysis
Our unsupervised method is superior to previous unsupervised methods and has comparable results with the supervised method, which reflects the effectiveness and advantages of our approach. The large error in our frame-to-frame rotation estimation results in a larger gradual drift, which should be fixed by the beam. A visual comparison of the estimated trajectories of our method can be seen in Figure 2, Figure 3.

![Figure 2. Figure with Seq. 09](image1.png)

![Figure 3. Figure with Seq. 10](image2.png)

5. Conclusion
We propose an unsupervised learning SLAM algorithm based on multi-constraint learning model, the depth map is predicted by the Depth net based on Spatial Photometric constraints in the first part, and the ego-motion is predicted by the Pose net based on Temporal Photometric constraints in the second part. We innovatively add depth warp constraints on the two-part cascade network, which further enables the cascade network to capture the relation between pixels, thereby improving the accuracy of predicting ego-motion. Our unsupervised method is superior to previous unsupervised methods and has comparable results with the supervised method, which reflects the effectiveness and advantages of our approach.

But there are still many challenges for our framework that we need to solve. The experimental results also illustrate the dependence of the proposed algorithm on the training data. Especially when the inter-frame speed of the image sequence is too fast, the algorithm error is large. The reason is that the lack of high-speed training samples in the training set causes the estimated rotation error to be large.
Acknowledgments
This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61873306.

References
[1] J. J. Leonard and H. F. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” IEEE Transactions on robotics and Automation, vol. 7, no. 3, pp. 376–382, 1991.
[2] Scaramuzza D, Fraundorfer F. Visual odometry: Part I: the first 30 years and fundamentals. IEEE Robotics and Automation Magazine, 2011, 18(4): 80–92
[3] Fraundorfer F, Scaramuzza D. Visual odometry: Part II: matching, robustness, optimization, and applications. IEEE Robotics and Automation Magazine, 2012, 19(2):78–90
[4] Kummerle R, Grisetti G, Strasdat H, et al. g2o: A general frame work for graph optimization [C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2011: 3607-3613.
[5] Tomasz M. (2016, January 13). The Future of Real-Time SLAM and Deep Learning vs SLAM. Retrieved September 4, 2018, from http://www.computervisionblog.com/2016/01/why-slam-mattersfuture-of-real-time.html
[6] Mur-Artal, R.,Montiel,J.M.M.,&Tardos,J.D.(2015).ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Transactions on Robotics, 31(5), 1147-1163.
[7] Kendall, A., Grimes, M., & Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE international conference on computer vision (pp. 2938-2946).
[8] Zhou, T., Brown, M., Snavely, N., & Lowe, D. G. (2017, July). Unsupervised learning of depth and ego-motion from video. In CVPR (Vol. 2, No. 6, p. 7).
[9] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In Advances in Neural Information Processing Systems, pages 2017–2025, 2015.
[10] Fang Z, Scherer S. Real-time onboard 6DoF localization of an indoor MAV in degraded visual environments using a RGB-D camera[C]// 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015.
[11] Zhan H, Garg R, Weerasekera C S, et al. Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction [J]. 2018.
[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The kitti vision benchmark suite. In Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
[13] A. Geiger, P. Lenz, C. Stiller, and R.Urtasun. Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR), 2013.
[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia, pages 675–678. ACM, 2014.
[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv: 1412.6980, 2014.
[16] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. IEEE Transactions on Robotics, 31(5):1147–1163, 2015.