Harnessing the power of RNNs for Visual Odometery

Aman Chulawala
Carnegie Mellon University
Pittsburgh, PA
achulawa@andrew.cmu.edu

Rohan Chandrasekar
Carnegie Mellon University
Pittsburgh, PA
rohancha@andrew.cmu.edu

Mansi Sarawata
Carnegie Mellon University
Pittsburgh, PA
msarawat@andrew.cmu.edu

Vigknesh Rajan
Carnegie Mellon University
Pittsburgh, PA
vvaithil@andrew.cmu.edu

Abstract

One of the most difficult challenges for mobile robots is to precisely localize the position of a vehicle. Such a robot must constantly be aware of its location in order to perform autonomous navigation, motion tracking, and obstacle recognition and avoidance. For this purpose, a reliable technique is vision-based odometry. This report studies monocular visual odometry(VO). The standard pipeline for performing visual odometry includes feature extraction, camera calibration,local optimisation etc. Thus some prior knowledge of system is required to recover absolute trajectory. However,a RNN+CNN model can be used to infer poses directly without this prior knowledge. This report presents comparison between the conventional method (geometry-based odometry) used for monocular visual odometry with an end-to-end trained RNN+CNN model for trajectory estimation and verifies the viability of the end-to-end model over traditional visual odometry systems.

1. Introduction

In mobile robot applications, precise vehicle localisation is a major difficulty. To perform autonomous navigation, a robot must continuously keep track of its location. As a result, researchers and engineers have created a variety of sensors, techniques, and systems for mobile robot positioning, including wheel odometry, laser/ultrasonic odometry, the global position system (GPS), the global navigation satellite system (GNSS), the inertial navigation system (INS), and visual odometry (VO) [1]. Every method, however, has flaws of its own like the commercial GPS estimates position with errors in the order of meters. This error is considered too large for precise applications that require accuracy in centimeters, such as autonomous parking. Whereas with a relative position inaccuracy of between 0.1-2%, VO is a less expensive alternative odometry approach that is more accurate than traditional methods like GPS, INS, wheel odometry, and sonar localization systems [2].

1.1. Background

Stereo VO can degenerate to the monocular situation when the distance between the scene and the stereo camera is substantially more than the stereo baseline, rendering stereo vision useless.In monocular VO, the 3D structure and relative motion are calculated from the 2D bearing information. For the purpose of rejecting outliers, monocular VO employs the feature tracking technique and random sample consensus (RANSAC). Through the estimation of the 3D to 2D camera position, the new forthcoming camera pose was calculated. With a little modification to the motion estimation phase, the created technique, which consists of three stages (feature detection, feature tracking, and motion estimation), can be used with either monocular or stereo vision systems. Each image frame's corners are first extracted by the algorithm, which then tracks the features it has found between frames. To properly track features from one image to the next, a matching criterion is used.The motion estimation phase is then carried out. A five-point posture algorithm is used in the motion estimation phase of a monocular vision system to determine the pose for each monitored feature. The first and last photos taken are used to determine the 3D position of each feature that was detected. The estimation of the camera's 3D posture is then done using 3D point data. In a stereo vision system, the stereo matching of the features between the two images captured by each camera yields the 3D position of each extracted feature.The features from the ground plane are used in a hybrid technique that combines feature- and appearance-based VO in a monocu-

0	1	2	3	4	5	6	7	8	9	10	11	12
P0:	718.856000	0.000000	607.192800	0.000000	0.000000	718.856000	185.215700	0.000000	0.000000	0.000000	1.000000	0.000000
P1:	718.856000	0.000000	607.192800	-386.144800	0.000000	718.856000	185.215700	0.000000	0.000000	0.000000	1.000000	0.000000
P2:	718.856000	0.000000	607.192800	45.382250	0.000000	718.856000	185.215700	-0.113089	0.000000	0.000000	1.000000	0.003780
P3:	718.856000	0.000000	607.192800	-337.287700	0.000000	718.856000	185.215700	2.369057	0.000000	0.000000	1.000000	0.004915
Tr:	0.000428	-0.999967	-0.008084	-0.011985	-0.007211	0.008081	-0.999941	-0.054040	0.999974	0.000486	-0.007207	-0.292197

Figure 1. Sequences in KITTI Dataset

lar omnidirectional configuration. The translation and absolute scale were estimated using these features by tracking scale-invariant feature transform (SIFT) points. The spin of the vehicle was calculated using an image appearance visual compass. Because the appearance-based method is susceptible to obstacles, the feature-based approach was also used to identify its shortcomings.

1.2. The KITTI Dataset

The Odometry Benchmark dataset of colored images created by the Karlsruhe Institute of Technology, Germany was used for this project. This dataset is commonly referred to as the KITTI Dataset. The odometry benchmark consists of 22 stereo sequences, saved in loss less png format: We provide 11 sequences (00-10) with ground truth trajectories for training and 11 sequences (11-21) without ground truth for evaluation. From all the test sequences, our evaluation computes translational and rotational errors for the first set of subsequences. The data sequences of all 4 cameras can be seen in Figure 1.

1.3. Methodology

This section reviews earlier work on the monocular VO and discusses various methods and how they differ from one another. There are essentially two sorts of algorithms in terms of the technique and framework adopted: geometry based and learning based methods.

Methods Based on Geometry: Geometry-based approaches, which predominate in VO and are theoretically grounded on geometric theory, use geometric constraints taken from pictures to estimate motion. Since they are derived from elegant and proven principles and have been widely researched, most of state-of-the-art VO algorithms come into this family. They can also be separated into direct approaches and sparse feature based methods.

1) Sparse Feature Based Methods: After extracting and matching (or tracking) salient feature points from a series of images, sparse feature-based algorithms use multi-view geometry to determine motion. However, due to the presence of outliers, disturbances, etc., all VO algorithms suffer from drifts over time. Visual SLAM (simultaneous localization and mapping) or SfM (structure from motion) can be used to maintain a feature map for drift correction together with posture estimation in order to alleviate this issue. Examples include keyframe basis PTAM and ORB-SLAM.

2) Direct Methods: The computational cost of feature extraction and matching for sparse feature-based approaches is high. More importantly, they ignore the wealth of information included in the entire image and just use the key characteristics. On the contraray, direct approaches can, if photometric consistency is assumed, utilize every pixel in a series of photos to estimate a pose. Recently, semi-direct methods for the monocular VO have been devised that achieve improved performance. Direct approaches are progressively becoming more popular since, in general, they are more accurate than feature-based ones and can function better in situations without textures.

Methods Based on Learning: Machine learning is used in learning-based techniques, which are data-driven, to infer VO from sensor measurements and train motion models without specifically referencing geometric theory. Using optical flow, the regression techniques K Nearest Neighbor (KNN), Gaussian Process (GP), and Support Vec- tor Machines (SVM) are trained for the monocular VO. Few studies on learning-based systems have been conducted, and no one has yet directly dealt with raw RGB images.
Traditional machine learning methods have been shown to be ineffective when dealing with huge or highly non-linear, high-dimensional data, such as RGB images. DL, which automatically learns proper feature representation from large-scale datasets, offers an alternate solution to the VO problem.

Deep Learning Based Methods: DL has achieved promising results on some localisation related applications. The features of CNNs, for instance, have been utilised for appearance based place recognition. Unfortunately, there is little work on VO or pose estimation. To our knowledge, firstly realises deep learning based VO through synchrony detection between image sequences and features. After estimating depth from stereo images, the CNN predicts the discretized changes of direction and velocity by

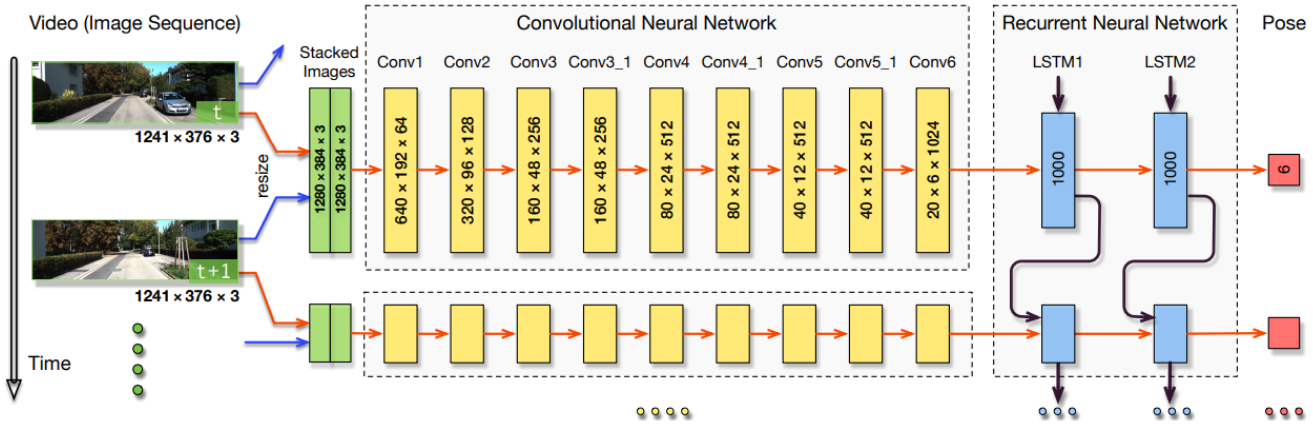


Figure 2. Model Architecture [2]

the softmax function. Although this work provides a feasible scheme for deep learning based stereo VO, it inherently formulates the VO as a classification problem rather than pose regression. Camera relocalization using a single image is solved in by fine-tuning images of a specific scene with CNNs. It suggests to label these images by SfM, which is time-consuming and labour-intensive for large-scale scenarios. Because a trained CNN model serves as an appearance "map" of the scene, it needs to be re-trained or at least fine-tuned for a new environment. This seriously hampers the technique for widespread usage, which is also one of the biggest difficulties when applying DL for VO. To overcome this problem, the CNNs are provided with dense optical flow instead of RGB images for motion estimation. Three different architectures of CNNs are developed to learn appropriate features for VO, achieving robust VO even with blurred and under-exposured images. However, the proposed CNNs require pre-processed dense optical flow as input, which cannot benefit from the end-to-end learning and may be inappropriate to real-time applications. Because the CNNs are incapable of modelling sequential information, none of the previous work considers image sequences or videos for sequential learning. In this work, we tackle this by leveraging the RNNs.

2. Experiments

In this section, we discuss the experimental results of the monocular visual odometry and the proposed RNNs for visual odometry on the well-known KITTI dataset. Since most of existing monocular VO algorithms do not estimate an absolute scale, their localisation results have to be manually aligned with ground truth.

2.1. Training and Tests

The data set was split for training and testing purposes. There were a total of 11 path sequences out of which we trained the model on 7 of them and tested on 5.

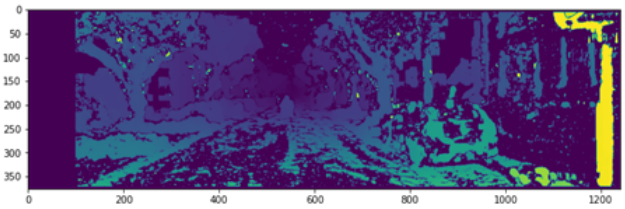


Figure 3. Disparity map using the StereoBM method

Training sequences - 01,02,06,08,09,10,11
Validation sequences- 03,05
Testing sequences - 07,03,04,05
The sequences were decided based on the length of the path. The one with relatively longer path was used a training set and the rest as test.

The model was trained using NVIDIA CUDA for 100 epochs with a learning rate of 1e-3. Dropout and early stopping techniques were introduced to prevent the models from over fitting. In order to reduce both the training time and data required to converge, the CNN is based on a pre-trained FlowNet model.

2.2. Stereo Semi-Global Block Matching (SGBM)

We used OpenCV to apply stereo depth estimation and multi-view geometry to attempt to track vehicle position through a sequence.

We first compute the disparity map using the StereoBM method. The StereoBM method is used to compute stereo correspondence using the block matching algorithm.

The output from the StereoBM method is then compared with the StereoSGBM method. In this method, the disparity of a pixel is calculated by considering a smaller block

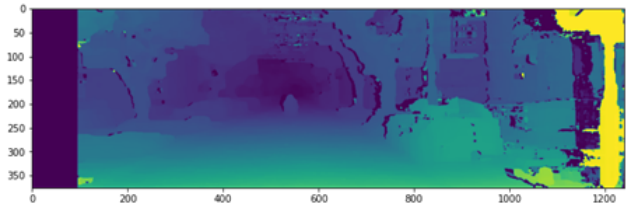


Figure 4. Disparity map using the StereoSGBM method

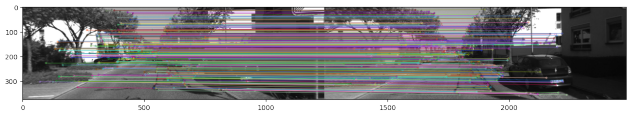


Figure 5. Feature Matching using SIFT Descriptors using the StereoSGBM matcher

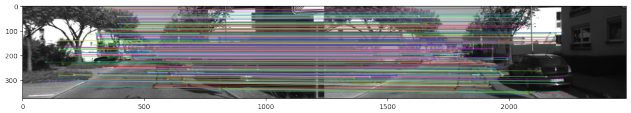


Figure 6. Visualizing the Lidar pointcloud

of pixels for ease of computation. Thus, the Semi-Global Block Matching (SGBM) algorithm uses block-based cost matching that is smoothed by path-wise information from multiple directions.

This is then followed by finding the width of in order to create a mask to prevent the feature detector from searching insignificant areas for features on every frame. Next, we detect and match features between two images using the SIFT descriptor.The algorithm consists of:

1.Peak selection

2,Keypoint Localization

3.Orientation Assignment

4.Keypoint Descriptor and Keypoint Matching

Feature matching using StereoSGBM and StereoBM as matchers are depicted in figures 5 and 6.

We then visualize the Lidar pointcloud using matplotlib.

2.3. CNN+RNN

The performance of the trained VO models is analysed according to the KITTI VO/SLAM evaluation metrics, i.e., averaged Root Mean Square Error (RMSEs) of the translational and rotational errors for all subsequences of lengths ranging from 100 to 800 meters and different speeds (the range of speeds varies in different sequences).

parameters: epochs=100 batch size= 5 learning rate= 0.001

The model maps the roll, pitch and yaw of the vehicle at every instance. This can be easily learnt to model by the

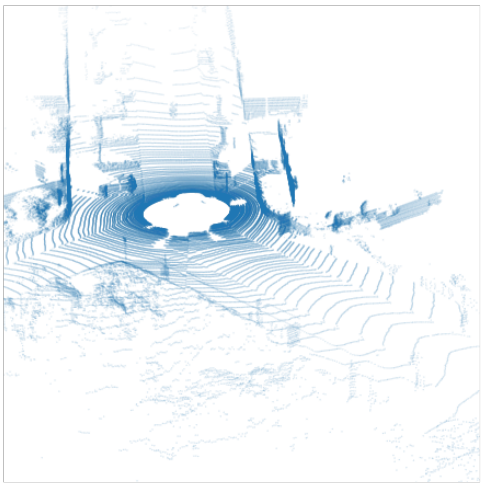


Figure 7. Feature Matching using SIFT Descriptors using the StereoBM matcher

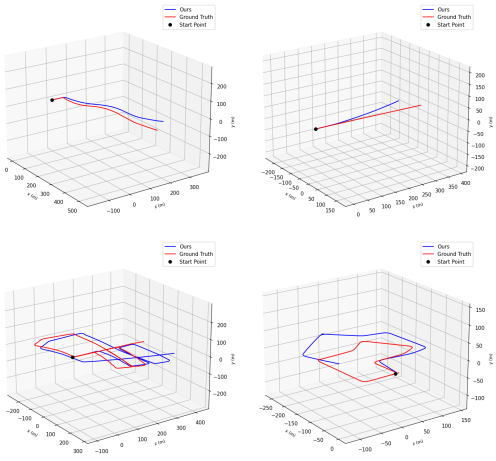


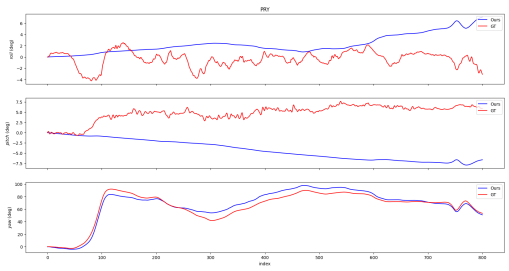
Figure 8. Predicted 3D trajectories of sequences 03 04 05 07

RNN in terms of orientation.

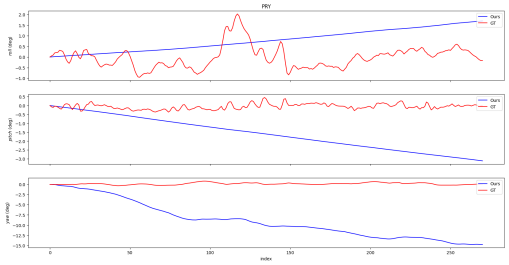
In Figure 8. we have mapped the translation and rotation error of the model as it progresses through the epochs. The errors decreases and plateaus towards the end.

3. Conclusion

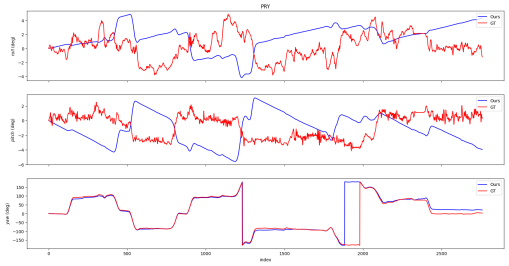
We first implemented visual odometry using geometry-based methods.Next, we developed a CNN+RNN model to predict the same trajectory.The disadvantage of the geometry-based visual odometry is that some prior knowledge of system is required to recover absolute trajectory.



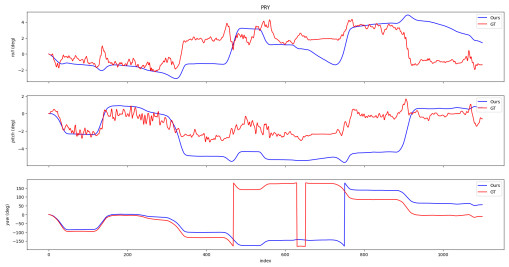
Sequence 03



Sequence 04



Sequence 05



Sequence 07

Figure 9. Roll, pitch and yaw prediction

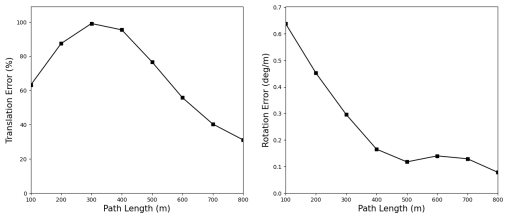


Figure 10. Translational and rotation error for sequence 06

However,a RNN+CNN model omits this and can be used to predict poses directly. Both the methods perform comparably well.

4. References

[1] M.O.A.Aqel, M.H. Marhaban, M. Iqbal, Napsiah Bt. Ismail,"Review of visual odometry:types,approaches,challenges,and applications",Springer,2016

[2] Sen Wang,Ronald Clark,Hongkai Wen,and Nikki Trigoni,"DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks",IEEE International Conference on Robotics and Automation (ICRA),2017

[3] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3D reconstruction in real-time," in Intelligent Vehicles Symposium (IV), 2011.

[4] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 1052–1067, 2007.

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR). IEEE, 2007, pp. 225–234.

[6] R. Mur-Artal, J. Montiel, and J. D. Tardos, "ORB-SLAM: a versa- ´ tile and accurate monocular SLAM system," IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147–1163, 2015.

[7] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in Proceedings of IEEE International Conference on Computer Vision (ICCV). IEEE, 2011, pp. 2320–2327.

[8] J. Engel, J. Sturm, and D. Cremers, "Semi-dense visual odometry for a monocular camera," in Proceedings of IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1449–1456.

[9] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in Proceedings of IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 15–22.

[10] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," in arXiv:1607.02565, July 2016.

[11] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. Balch, "Memorybased learning for visual odometry," in Proceedings of IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2008, pp. 47–52.

[12] V. Guizilini and F. Ramos, "Semi-parametric learning for visual odometry," The International Journal of Robotics Research, vol. 32, no. 5, pp. 526–546, 2013.

[13] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "Evaluation of non-geometric methods for visual odometry," Robotics and Autonomous Systems, vol. 62, no. 12, pp. 1717–1730, 2014.

[14] N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, ¨ B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in Proceedings of Robotics: Science and Systems (RSS), 2015.

[15] K. Konda and R. Memisevic, "Learning visual odometry with a convolutional network," in Proceedings of International Conference on Computer Vision Theory and Applications, 2015.

[16] A. Kendall, M. Grimes, and R. Cipolla, "Convolutional networks for real-time 6-DoF camera relocalization," in Proceedings of International Conference on Computer Vision (ICCV), 2015.

[17] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5162–5170, 2015.

[18] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. IEEE transactions on pattern analysis and machine intelligence, 38(10):2024–2039, 2016.

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015.

[20] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 234–241. Springer, 2015.

[21] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In Computer Vision (ICCV), 2011 IEEE international conference on, pages 2564–2571. IEEE, 2011.