Fast Stereo Visual Odometry Techniques for Practical Applications

Achyut Boggaram
Indiana University Bloomington
IN, USA
achbogga@indiana.edu

Abstract

The stereo visual odometry has many vital applications in robotics and other fields as well. We investigate various techniques including CNN based approaches and check the feasibility of such methods with respect to real time application. Further, we have specified some useful modifications in the current state of the art pipeline to incorporate faster performance without compromising the accuracy. The modifications include forward motion significance, inlier detection instead of outlier detection, raw corner location information instead of feature descriptors. We have found out that CNN approaches in their current state may not be a good choice given their performance. We have also found out that segmentation information may not help in increasing the system performance. We have proven that the performance of traditional approaches increase substantially on real world data with our modifications. We have performed all of our experiments on publicly available KITTI data set [10].

1. Introduction and Related Work

Stereo Visual odometry is the task of estimating the current pose and trajectory of a robot given the video from two cameras mounted on the robot and their calibration parameters. It is essentially a 3D reconstruction of the world given the point/feature/disparity correspondences across the frames. It is very useful in GPS denied environments and in visual SLAM. The traditional approaches like 3D reconstruction are geometry, optics dependent. Some people have attempted deep learning, both end to end and partially into the traditional pipeline to increase the performance as well as automate the whole pipeline. We will talk about the related traditional approaches in subsection 1.1 and related modern deep learning approaches in subsection 1.2. Then we will briefly talk about our proposed modifications and results in subsection 1.3. The section 2 discusses our approach followed by results in section 3. The conclusion and future directions will follow in section 4.

1.1. Traditional Geometry approaches

There is very good amount of research in the areas of monocular visual odometry by Scaramuzza *et al*. [26, 27] and odometry in general [29]. The previous work in this area can be broadly classified into two categories. Feature based Stereo Visual Odometry and Appearance based Stereo Visual Odometry and of course, there are other hybrid approaches as well [3]. The feature based approaches like that of Nister *et al*. [23] and Howard *et al*. [14] were one of the first to develop such robust approaches which can be applied in real time. There are also many other modifications to such works by using several other features, outlier rejection schemes, etc. [5, 11, 18, 24]. These state of the art SOFT, BRISK, FAST Corner, etc., feature based stereo VO approaches are being widely used in robotics because of their robustness and performance. Although these approaches are very popular, they are very geometry dependent and hence very much subjective to the camera sensor calibration data, environment and many other handcrafted features.

1.2. Deep Learning Approaches

As we have been witnessing tremendous popularity in the application of end to end Deep Learning techniques vs the traditional handcrafted to solve many vision problems objectively, it is natural to investigate the feasibility of deep learning based approaches.Arandjelovic*et al*. [4] and Hou *et al*. [13] used unsupervised deep learning for loop detection as an enhancement in order to eradicate the drift or cumulative errors. There are other deep learning approaches which have concentrated on developing robust or alternative features by using unsupervised learning approaches [8]. Coming to the end to end learning specifically for the task of Visual Odometry, Konda *et al*. [16] attempted to tackle the problem with something called Sychrony detection. Agarwal *et al*. [1] and others who tackled this problem with Siamese style CNN networks called KittiNet. There are others who used deep learning along with monocular cameras and Laser sensors who achieved good performance [15, 22]. While this direction of using deep learning shows promise,

we have a long way to go before they can be practically applied as the results obtained are not comparable to those of traditional geometry based approaches.

1.3. Attempted Experiments and Modifications

We investigate the feasibility of automation using end to end CNNs very similar to that of AlexNet [17] keeping in mind the real time performance. Then we investigate the use of fusing foreground and background semantic segmentation information with existing feature descriptors. Later, we attempted inlier detection posed as a clique problem [2] as suggested by Avi Singh [28]. Further, we have attempted to alter the acceptance step of rotation and translation matrices only when they imply a significant forward motion. This step along with the inlier detection step show a significant rise (18 percent) in the performance of the system for a minor decrease in accuracy (6 percent) when tested on Howard's approach. *et al*. [14] with KITTI Visual Odometry grayscale data [10]. The details of the approaches and experiments will be discussed in the following sections 2 and 3.

2. Experiments

2.1. Setup

All the experiments are performed using the publicly available rectified and undistorted KITT Vision benchmark suite's Visual Odometry Grayscale stereo dataset [10]. which contains sequences of outdoor drives captured from the two cameras on a car. Please refer their website for additional information. The system used to conduct experiments is a third generation i5 with 2.4GHz and 8GB RAM. The code is written in MATLAB on Linux machine. The experiments involving CNNs were performed on Bigred2 with a dedicated NVIDIA K20 Tesla GPU and were coded using Torch7.

2.2. AlexNet Experiment

We have adapted AlexNet in the experiment except, we made it to predict continuous values rather than classify. We simply removed the ReLU activation functions and used linear activations for the output layer and used MSE (Mean squared Error as the cost function criterion). We have fed initially the net with KITTI sequences 00 to 05 for training and check how it is performing. The cost kept on oscillating and was maybe appeared to be going down but very slowly when we tried it on the Bigred2. We have decided to move on to investigate traditional approaches given the lack of resources required to play with larger nets and complex architectures.

We have an LSTM based [12] Siamese style [7] hierarchical approach in mind that I would love to test out soon. I will discuss more in the future direction section.

Approach	time	overallVO-err/400m
RANSAC, SIFT	22.197674s	0.25%
RANSAC, FAST	16.549902s	0.28%
CLIQUE, FAST	11.734424s	0.26%

Table 1. Average VO time per frame over KITTI grayscale data

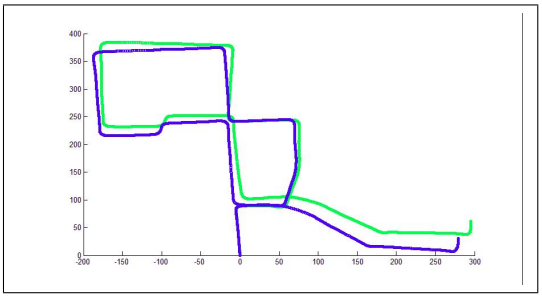

Figure 1. CLIQUE, FAST approach run on KITTI sequence 00

2.3. Segmentation Experiment

Intuitively when trying to match correspondences, we have a tendency to think that what if we can eliminate some mismatches directly by telling them from foreground or background. As it turned out the task of pixel wise semantic segmentation using CNNs [19] was not simply not fast enough to be implemented in real time. Although, it might give the inlier detection or outlier detection steps a boost in speed.

2.4. Getting rid of Feature Descriptors

The problem is with the feature descriptor itself. We have seen that some of the real time implementations of Visual odometry such as that of Howard*et al*. [14] were using FAST corner locations directly to match correspondences rather than using feature descriptors. This approach although reduces accuracy in the number of features detected per frame, it gives a lot of performance improvement. On KITTI data we have seen an average performance increase of 18 percent just using these FAST-ER [25] corners and tracking them using KLT tracker [21].

2.5. Inlier Detection

Traditionally all the visual odometry estimation approaches use RANSAC [9] for rotation and translation model fitting given the point correspondences. We use a different approach as suggested by Avi Singh [28] for faster performance along with FAST corners. We noticed a significant performance increase for estimation over just one frame.

3. Results

By incorporating the above discussed best practices we have seen an avg VO calculation speed increase of almost forty five percent in comparison to the traditional RANSAC [9] and SIFT [20] feature matching. Refer to the table 1 for exact avg times of rotation and translation estimation for just one frame over the entire sequences in KITTI grayscale odometry data. The accuracy seems to be almost the same for each of the above approaches.

Please refer to the Figure 1 for the Full VO estimation trajectory vs ground truth trajectory image for the sequence zero zero of the KITTI Grayscale stereo odometry data.

The code used for this project can be found publicly on my Git-hub Repository [6]. I will be updating the same with time.

4. Conclusion and Future direction

Given the fact that we have implemented VO in MATLAB, our software cannot be directly applied to practical applications. Although, if the same approach implemented using C or Cpp would be faster than real time. We are providing a comparison over existing approaches and best practices. Also, We have not tested our assumptions with the new state of the art methods like Kinect RGBD, ORB,etc., One would look at his resources and proceed given the literature available to him/her. Also, to avoid going through such hassles, we would like to further investigate the modelling of the Visual Odometry estimation as dependent on disparity map information given the stereo data, dependent on the optical flow as it has to model the motion estimation and finally remember the whole path to make the approach robust. Therefore given the spatial and temporal dependencies, we would like to use a hierarchical approach combining Siamese style networks to account for optical flow, raw stereo RGB images and disparity maps. Then on top of these CNN features apply LSTMs over time so that the deep networks remember the previous configuration as well accounting for a smooth flow of trajectory given sufficient training data. Such network is a very complex one and one would need all architectural tricks such as inception v3, resnets, etc., Given the limited amount of time and resources we could not design and test such networks. Though, we believe that these type of networks may handle such continuous geometrical estimation problems as well as facial expression estimation problems robustly without much human handcrafted inputs.

5. Problems encountered

The torch broke after sometime while using Bigred2 which I am still trying to figure out why. Very less time in learning about traditional Visual odometry approaches and understanding the publicly available code bases. Installation if different dependencies into servers, especially I had a lot of trouble identifying different binary locations and creating local environments. Very less time and resources in understanding the complex networks such as LSTMs and Siamese style networks. Segmentation approach did not work as it was taking much longer time than expected. We were better off with normal pipeline.

I have learned so much about Visual SLAM and Visual Odometry through the course of this project regardless of the difficulties.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. *CoRR*, abs/1505.01596, 2015.

[2] Anonymous. Clique problem.

[3] M. O. A. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, 5(1):1897, 2016.

[4] R. Arandjelovic, P. Gronát, A. Torii, T. Pajdla, and J. Sivic. Netvlad: CNN architecture for weakly supervised place recognition. *CoRR*, abs/1511.07247, 2015.

[5] H. Azartash, N. Banai, and T. Q. Nguyen. An integrated stereo visual odometry for robotic navigation. *Robot Auton Syst*, 62, 2014.

[6] A. Boggaram. achbogga/vo_fall_16, Dec 2016.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[8] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to SIFT. *CoRR*, abs/1405.5769, 2014.

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] 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.

[11] Golban. Golban c, istvan s, nedevschi s (2012) stereo based visual odometry in difficult traffic scenes. in: Anonymous intelligent vehicles symposium (iv), 2012 ieee. ieee, piscataway, p 736–741, 2012.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] Y. Hou, H. Zhang, and S. Zhou. Convolutional neural network-based image representation for visual loop closure detection. *CoRR*, abs/1504.05241, 2015.

[14] A. Howard. Howard a (2008) real-time stereo visual odometry for autonomous ground vehicles. in: Anonymous 2008 ieee/rsj international conference on intelligent robots and systems. p 3946–3952, 2008.

[15] D. K. Kim and T. Chen. Deep neural network for real-time autonomous indoor navigation. *CoRR*, abs/1511.04668, 2015.

[16] K. Konda and R. Memisevic. Learning visual odometry with a convolutional network. In *Proceedings of the 10th International Conference on Computer Vision Theory and Applications - Volume 1: VISAPP, (VISIGRAPP 2015)*, pages 486–490, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[18] L. H. Lin, P. D. Lawrence, and R. Hall. Robust outdoor stereo vision slam for heavy machine rotation sensing. *Mach Vis Appl*, 24, 2013.

[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] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[21] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *IJCAI*, volume 81, pages 674–679, 1981.

[22] V. Mohanty, S. Agrawal, S. Datta, A. Ghosh, V. D. Sharma, and D. Chakravarty. Deepvo: A deep learning approach for monocular visual odometry. *CoRR*, abs/1611.06069, 2016.

[23] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *J Field Robot*, 23, 2006.

[24] N. Nourani-Vatani and P. V. K. Borges. Correlation-based visual odometry for ground vehicles. *J Field Robot*, 28, 2011.

[25] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE transactions on pattern analysis and machine intelligence*, 32(1):105–119, 2010.

[26] D. Scaramuzza and F. Fraundorfer. Tutorial: visual odometry. *IEEE Robot Autom Mag*, 18, 2011.

[27] D. Scaramuzza and R. Siegwart. Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Trans Robot*, 24, 2008.

[28] A. Singh. Visual odmetry from scratch - a tutorial for beginners.

[29] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad. An overview to visual odometry and visual slam: Applications to mobile robotics. *Intelligent Industrial Systems*, 1(4):289–311, 2015.