Project 3 - Visual Odometry - MC907

LUIZ EDUARDO CARTOLANO *

*Computer Engineering - Undergraduate
E-mail: 1183012@dac.unicamp.br

***Abstract* – In robotics and computer vision, visual odometry is the process of determining the position and orientation of a robot by analyzing the associated camera images. It has been used in a wide variety of robotic applications, such as on the Mars Exploration Rovers. The classical approaches for solve the VO problems are usually based on the kinematics of the robot, so they can be easily induced to fail when either the motion of the robot or the environment are too challenging. In this way, novel VO methods use deep neural networks or attention models to improve or even replace the entire algorithm pipeline. In this project, we aim to brief introduce both algorithms and given sample results of the frameworks.**

***Keywords* – Visual Odometry - Neural Networks - Attentional Models**

I. INTRODUCTION

Visual Odometry (VO) is one of the most essencial techniques for pose estimation and robot localization now a days. The classical approach for VO systems is shown in Figure 1, which typically consists of camera calibration, feature detection, feature matching, outlier rejection, scale estimation and local optimisation, has been developed and broadly recognised as a golden rule to follow. Altough the state-of-the-art algorithms based on this pipeline have shown excellent performance, they are usually hard-coded, where each module of the pipeline is adapted to the hardware in order to achieve an amazing performance.

In parallel, in the last years, Deep Learning (DL) [1], has been dominating many computer vision tasks with spectacular results. Unfortunately, for the VO problem this has not fully arrived yet, walking in short steps. In this work, we are going to present some new approachs that uses DL for the VO problem. Most of them consist at a Deep Recurrent Convolutional Neural Networks to solve the problem. One of the first articles that addresses the solution is shown at [2] and a pipeline is shown at Figure 2.

Another approach that has been gaining attention, is the use of a visual saliency map in order to increase the performance of the state-of-the-art algorithms. The concept is quite simple, as we are going to futher explain, humans perform the task of mapping very differently from how it has been usually done at the classical approaches at robotics. A classical paper, that we are going to further discuss is shown at [3].

II. DISCUSSION

For this article we will analyze three main configurations of Visual Odometry (VO), an overview of classical approach, inspired in [4], a Deep Learning one [2] and, lastly, an attention one [3].

A. *Classical Approach*

Most of the state-of-the-art algorithms of VO involving geometry based methods. Some of the most relevant ones are described next.

*1) **ORB-SLAM**:* ORB-Slam[5] is the most complete feature-based monocular VSLAM system. In basic lines, we can describe the system as follow.

The map is optimized in the background using traditional bundle adjustment, while new frames are tracked in real-time using model-based tracking. The system also include re-localization, loop closure detection, 7DoF pose-graph optimization and can handle large maps, using a double optimization strategy. An overview of the system is shown at Figure 3.

Some results of the framework are shown at Figure 4, as we can see in the figure, the obtained results for the ORB-SLAM with the 7DoF alignment are amazingly good and, they get even better when used along with the bundle adjustment.

In conclusion, the accuracy of the system is typically below one (1) centimeter in small indoor scenarios and of a few meters in large outdoor scenarios. Also it were able to demonstrate that ORB features have enough recognition power to enable place recognition from severe viewpoint change. Moreover, they are so fast to extract and match that enable real-time accurate tracking and mapping.

Even so, the system still has improvement points, like improved incorporating points at infinity in the tracking. Other is to upgrade the sparse map of the system to a denser and more useful reconstruction.

*2) **Direct Sparse Odometry (DSO)**:* Direct Sparse Odometry, proposed in [6], combines a fully direct probabilistic model with consistent, joint optimization of all models parameters, including geometry and camera motion. The algorithm divides the input image into several blocks, and then, it selects high-intensity points as reconstruction candidates.

Since the method does not depend on keypoint detectors or descriptors, it can naturally sample pixels from across all image regions that have intensity gradient, including edges or smooth intensity variations on essentially featureless walls. The model also integrates a full photometric calibration, accounting for exposure time, lens vignetting, and non-linear response functions.

The model were evaluate on three different datasets comprising several hours of video, a sample of the results is shown at Figure 5. The experiments show that approach significantly outperforms state-of-the-art direct and indirect methods in

a variety of real-world settings, both in terms of tracking accuracy and robustness.

As we are going to further see, there are ways to make the model even better.

*3) **Kalman Filter**:* The Multi-State Constraint Kalman Filter, introduced in [7], is a sparse VO, filtering-based, and feature-based method for monocular vision. The framework is formulated as an extended Kalman Filter, keeping as state a sliding windown of new camera frames, 3D points are triangulated and added as observation only once they leave the field of view of the camera.

Some results shown at Figure 6 demonstrate that the algorithm is capable of operating in a real- world environment, and producing very accurate pose estimates in real-time. Also the algorithm is able to discard the outliers which arise from visual features detected on these objects, using a simple Mahalanobis distance test. It's important to add that the described method can be used either as a stand-alone pose estimation algorithm, or combined with additional sensing modalities to provide increased accuracy.

*4) **Dense Tracking and Mapping**:* Dense Tracking and Mapping (DTAM) is a direct SLAM method proposed in [8]. DTAM was one of the first dense and direct SLAM methods that operate on a single monocular camera. This algorithm performs tracking by comparing the input image with synthetic-view images generated from the reconstructed map, and the mapping uses multi-baseline stereo. Then, it optimizes the map by considering space continuity so that 3D coordinates of all pixels can be computed.

*5) **Large-Scale Direct Slam**:* Another direct SLAM algorithm is called Large-scale Direct SLAM (LSD-SLAM), presented in [9]. In this method, geometry is represented in the form of semi-dense inverse depth maps for selected keyframes, containing depth values for all pixels with sufficient intensity gradient. Random values work as initial depth values for each pixel, and then, these values are optimized based on photometric consistency.

*6) **Semi-Direct Visual Odometry**:* The Semi-Direct Visual Odometry (SVO), introduced in [10], is a sparse semi-direct approach. The algorithm operates directly on pixel intensities, which results in subpixel precision at high frame-rates. Point depths are initially estimated using a direct formulation, optimizing the photometric error between the observed image and the reference patch for the given point. Subsequently, the algorithm "relax" the epipolar constraint, effectively fixing the established correspondences and converting the formulation to an indirect one, to optimize the system jointly. Effectively, this means that a direct formulation is used to obtain robust and outlier-free initializations for the underlying indirect model.

*7) **Final Observations**:* A summary of the classification of the most important geometric VO/VSLAM algorithms is shown at Figure 7.

B. Deep VO

The paper [2] presents a novel end-to-end framework for monocular VO by using deep Recurrent Convolutional Neural Networks (RCNNs) [11]. Since it is trained and deployed in an end-to-end manner, it infers poses directly from a sequence of raw RGB images (videos) without adopting any module in the conventional VO pipeline. Based on the RCNNs, it not only automatically learns effective feature representation for the VO problem through Convolutional Neural Networks, but also implicitly models sequential dynamics and relations using deep Recurrent Neural Networks. The framework has been tested under the Kitti Dataset [1], showing great results.

The model is mainly composed of CNN based feature extraction, as previous presented in [12], and RNN based sequential modelling, as shown in [13]. The architecture of the proposed VO system is shown in Figure 8, it takes a monocular image sequence as input. Two consecutive images are stacked together to form a tensor for the deep RCNN to learn how to extract motion information and estimate poses. The VO system develops over time and estimates new poses as images are captured.

The image feature extraction of is done by a CNN, which configuration is shown at Figure 9. The CNN takes raw RGB images instead of pre-processed counterparts, such as optical flow or depth images, as input because the network is trained to learn an efficient feature representation with reduced dimensionality for the VO.

Following the CNN, a deep RNN is designed to conduct sequential learning, i.e., to model dynamics and relations among a sequence of CNN features. Since the RNN is capable of modelling dependencies in a sequence, it is well suited to the VO problem which involves temporal model (motion model) and sequential data (image sequence).

To learn the hyperparameters θ of the DNNs, the Euclidean distance between the ground truth pose (p_k ,φ_k) at time k and its estimated one ($\hat{\mathrm{p}}_k$,$\hat{\varphi_k}$) is minimised. The loss function is composed of Mean Square Error (MSE) of all positions p and orientations :

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{t} \|\hat{p_k} - p_k\|_2^2 + k \cdot \|\hat{\varphi_k} - \varphi_k\|_2^2 \quad (1)$$

In Figure 10, the losses results of the models are given for both training and validate datasets. As shown in the fig., the loss significantly reduces with time, and, once the training and validation losses shown a similar behavior we can also say that is free of overfiting, so, the model is well-fit.

The Figures 11 and 12 show us the obtained maps for the framework for both training and test datasets. As we can see in the figures, the obtained odometry from the system is very close to the ground truth position, showing how great the system performs. The system is analysed according to the KITTI VO/SLAM evaluation metrics, i.e., averaged Root Mean Square Errors (RMSEs) of the translational and rotational errors for all subsequences of lengths ranging. The average RMSEs of the estimated VO are given in Figure 13. Al- though the result of the DeepVO is worst than that of

[1] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

the stereo VISO2 (VISO2_S), it is consistently better than the monocular VISO2 (VISO2_M).

The paper presents a novel end-to-end monocular VO algorithm based on Deep Learning, that does not depend on any module in the conventional VO algorithms (even camera calibration) for pose estimation and it is trained in an end-to-end manner, there is no need to carefully tune the parameters of the VO system. Based on the KITTI VO benchmark, it is verified that it can produce accurate VO results with precise scales and work well in completely new scenarios.

Although the proposed DL based VO method presents some good results, it is not expected as a replacement to the classic geometry based approach.

C. *Salient DSO*

The paper [3], present a way to incorporate semantic information in the form of visual saliency to Direct Sparse Odometry – a highly successful direct sparse VO algorithm. The framework, SalientDSO, relies on the widely successful deep learning based approaches for visual saliency and scene parsing which drives the feature selection for obtaining highly-accurate and robust VO even in the presence of as few as 40 point features per frame. The system has been tested under the *ICL-NUIM*[2] and *TUM monoVO*[3] datasets and outperform *DSO*[6] and *ORB-SLAM*[5] – two very popular state-of-the-art approaches in the literature.

SalientDSO's framework is composed of a preprocessing step and a VO backbone. The VO backbone is responsible for initializing and tracking camera pose and optimizing all model parameters. The pre-processing step involves the saliency prediction and scene parsing using deep Convolutional Neural Networks (CNNs), as shown in [12], and later using these outputs to select features and points. Figure 14 shows the algorithmic overview of the system.

The DSO, mentioned before at Section II-A2, is used as the backbone VO in SalientDSO. In brief, it proposed a direct sparse model to jointly optimize all parameters (camera intrinsics, camera extrinsics, and inverse-depth values for feature points) and perform windowed bundle adjustment.

The main feature of this new approach is the visual saliency prediction. Visual saliency is defined as the amount of attention a human would give to each pixel in an image. This is quantitatively measured as the average time a person's gaze rests on each pixel in the image. The mentioned system adopt SalGAN [14] for saliency prediction, in brief, contains a generator and a discriminator. The generator is a deep CNN [11] trained on adversarial loss,

$$L_{GAN} = \alpha \cdot L_{BCE} - log(D(I, \hat{S}))$$

which includes Binary Cross-Entropy loss,

$$L_{BCE} = -\frac{1}{N}\sum_{j=1}^{N} S_j log(\hat{S}_j) + (1 - S_j) log(1 - \hat{S}_j)$$

[2] https://www.doc.ic.ac.uk/~ahanda/VaFRIC/iclnuim.html

[3] https://vision.in.tum.de/data/datasets/mono-dataset

to produce a down-sampled saliency map, and the discriminator is a shallower network as compared to the generator, this is trained to solve binary classification between saliency map produced by generator and the groundtruth. The saliency obtained from the system is shown at Figure 15.

Some results of the system, and a comparison with the *DSO* can be seen at Figure 16. The proposed framework achieve similar or better performance on most sequences when compared to the *DSO* and to the *ORB-SLAM* (it's important to note that the *ORB* used for comparison didn't have loop-closure detection and re-localization).

The paper introduce the philosophy of attention and fixation to visual odometry. Based on this philosophy, it improves the robustness and accuracy of the state-of-the-art algorithms. The system, however, has two main problems, it has not been tested on outside enviroments and, more important, it wasn't tested under a robot's hardware, i.e., no one knows how will it work on "normal" situations.

III. Conclusion

The ability to know its localization in an environment is an essential task for mobile robots, and it has been a subject of research in robotics for decades. Traditional odometry and SLAM processes depend on the kinematics model of a robot. And, traditional models like *ORB-SLAM* and *DSO* are, nowadays, the best algorithms for such tasks, being considered, as we can say, the state-of-the-art algorithms.

Even so, a lot of new approaches has emerged in order to solve the problem in more robust, efficient and less "hard-coded" way. Approaches like the DeepVO, who uses an end-to-end architecture to preview the bose based on input images, and the Salient DSO, who uses an attention model in order to extract better features from images are amazing examples of that. Both of them preseting very good results, and even were able to perform better than the state-of-the-art algorithms under some circunstances, but, in a general purpose, they are not ready, yet, to assume their place in the odometry problem.

+————————-+

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015. 1

[2] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2043–2050. 1, 2

[3] H.-J. Liang, N. J. Sanket, C. Fermüller, and Y. Aloimonos, "Salientdso: Bringing attention to direct sparse odometry," *IEEE Transactions on Automation Science and Engineering*, 2019. 1, 3

[4] H. M. S. Bruno, "End-to-end visual odometry for mobile robots," 2018. 1

[5] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015. 1, 3

[6] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017. 1, 3

[7] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 3565–3572. 2

[8] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327. 2

[9] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1935–1942. 2

[10] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 15–22. 2

[11] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3367–3375. 2, 3

[12] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016. 2, 3

[13] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems*, 2015, pp. 2980–2988. 2

[14] J. Pan, C. C. Ferrer, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv preprint arXiv:1701.01081*, 2017. 3

ATTACHMENTS

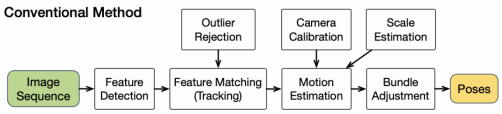


Figure 1. Classical pipeline for Visual Odometry.

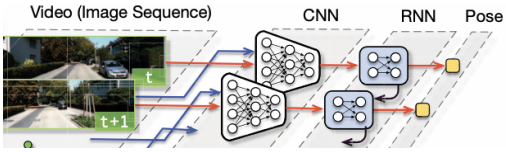


Figure 2. DeepVO pipeline for Visual Odometry.

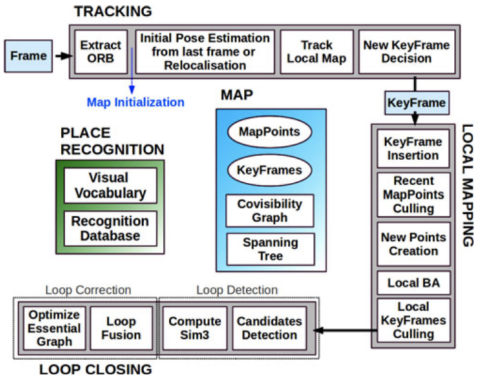


Figure 3. ORB-SLAM system overview for VO.

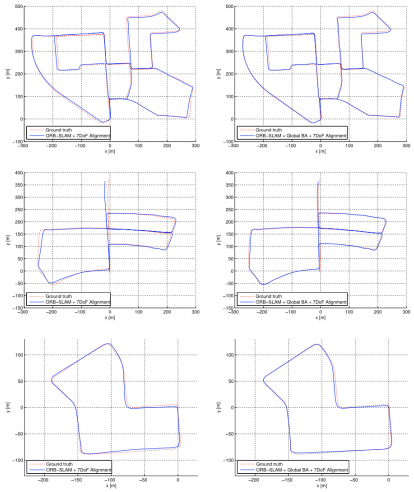


Figure 4. ORB-SLAM results for NewCollege maps.

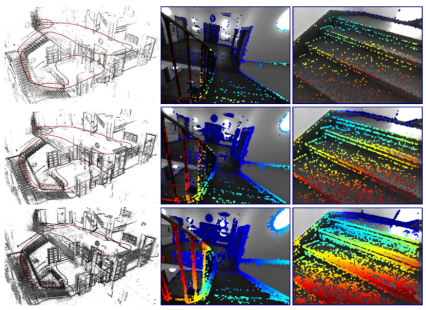


Figure 5. Sample DSO results.

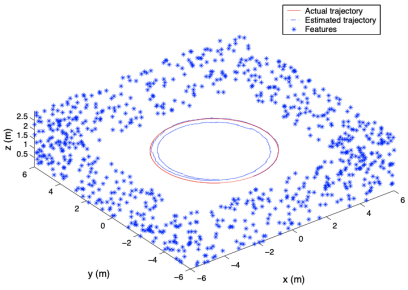


Figure 6. Sample Kalman Filter results.

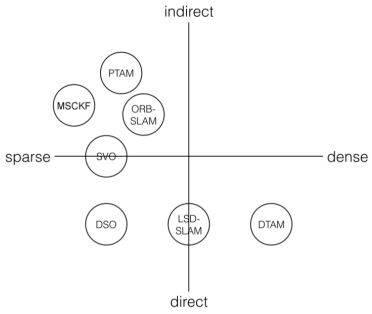


Figure 7. Geometric VOVSLAM algorithms.

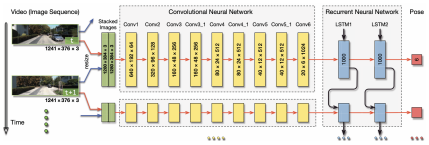


Figure 8. Architecture of the RCNN monocular VO system.

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

Figure 9. Configuration of the CNN for monocular VO system.

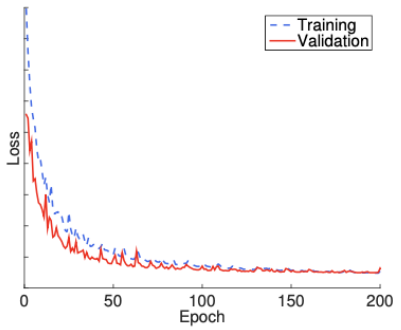


Figure 10. Training losses for the monocular VO system.

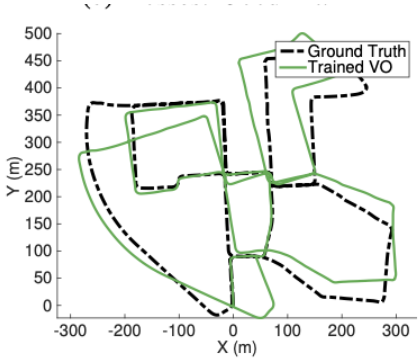


Figure 11. Obtained map for the monocular VO system under the training dataset.

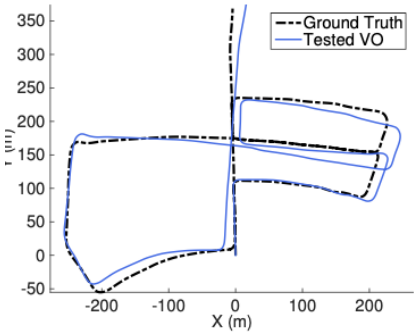
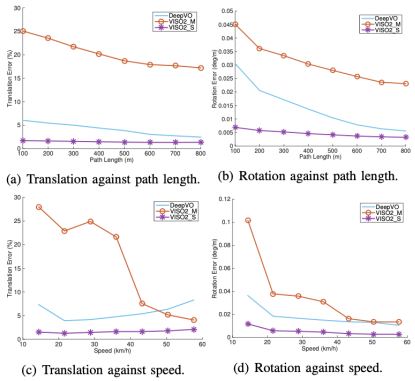


Figure 12. Obtained map for the monocular VO system under the test dataset.



(a) Translation against path length. (b) Rotation against path length.

(c) Translation against speed. (d) Rotation against speed.

Figure 13. Average errors on translation and rotation against different path lengths and speeds.

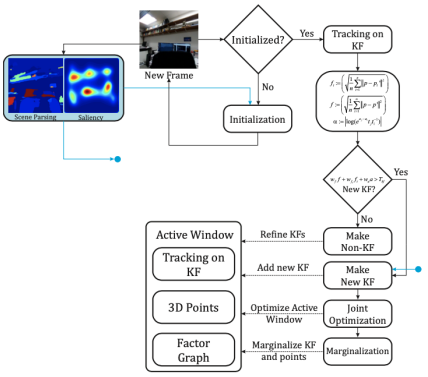


Figure 14. Algorithmic overview of SalientDSO.

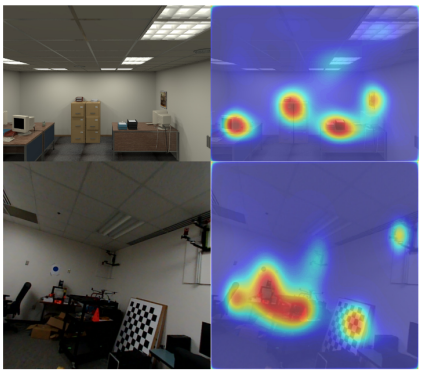


Figure 15. Left column: Input image, Right column: Saliency overlayed on input image.

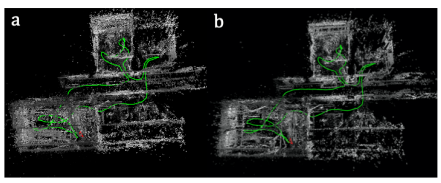


Figure 16. Sample outputs for TUM sequence 1. (a) DSO, (b) SalientDSO.