

Visual Odometry with Depth-Wise Separable Convolution and Quaternion Neural Networks

Giorgio De Magistris¹, Danilo Comminiello², Christian Napoli¹ and Janusz T. Starczewski³

¹*Department of Computer, Control, and Management Engineering
Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy
demagistris@diag.uniroma1.it, cnapoli@diag.uniroma1.it*

²*Department of Information Engineering, Electronics and Telecommunications
Sapienza University of Rome, Via Eudossiana 18, 00184 Rome, Italy
danilo.comminiello@uniroma1.it*

³*Department of Intelligent Computer Systems
Czestochowa University of Technology, al.Armi Krajowej 36, 42-200C Czestochowa, Poland
janusz.starczewski@pcz.pl*

Abstract

Monocular visual odometry is a fundamental problem in computer vision and it was extensively studied in literature. The vast majority of visual odometry algorithms are based on a standard pipeline consisting in feature detection, feature matching, motion estimation and local optimization. Only recently, deep learning approaches have shown cutting-edge performance, replacing the standard pipeline with an end-to-end solution. One of the main advantages of deep learning approaches over the standard methods is the reduced inference time, that is an important requirement for the application of visual odometry in real-time. Less emphasis, however, has been placed on memory requirements and training efficiency. The memory footprint, in particular, is important for real world applications such as robot navigation or autonomous driving, where the devices have limited memory resources. In this paper we tackle both aspects introducing novel architectures based on Depth-Wise Separable Convolutional Neural Network and deep Quaternion Recurrent Convolutional Neural Network. In particular, we obtain equal or better accuracy with respect to the other state-of-the-art methods on the KITTI VO dataset with a reduction of the number of parameters and a speed-up in the inference time.

1. Introduction


Monocular Visual Odometry consists in estimating the trajectory of an agent from a sequence of images acquired at consecutive time instants from a single camera mounted on the agent. Most of the visual odometry (VO) systems are based on a standard pipeline based entirely on geometry. Some implementations achieved excellent results, and a number of them can also run in real-time [1][2], however there is always a trade-off between real-time performances and consistency due to the final local optimization procedure. Moreover, in the case of monocular

✉ demagistris@diag.uniroma1.it (G. De Magistris); danilo.comminiello@uniroma1.it (D. Comminiello); cnapoli@diag.uniroma1.it (C. Napoli); janusz.starczewski@pcz.pl (J. T. Starczewski)

🆔 0000-0002-3076-4509 (G. De Magistris); 0000-0003-4067-4504 (D. Comminiello); 0000-0002-3336-5853 (C. Napoli); 0000-0003-4694-7868 (J. T. Starczewski)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

VO, the global scale is unknown and it must be computed using other measurements. Recently, the first end-to-end deep learning pipeline for monocular VO (DeepVO) was introduced in [3]. The authors used a RCNN [4] in order to automatically extract and match features in consecutive frames. In particular DeepVO is trained to predict the pose and orientation (and also the global scale) directly from a sequence of RGB images. With this work we extend the aforementioned model by introducing some optimizations aimed at reducing the number of parameters and increasing the inference speed. We will achieve this with two important changes. We will replace the standard real value convolution with the convolution in the quaternion domain, that allows to reduce the number of parameters without harming the expressive power. We also introduced Depth-Wise Separable Convolution (DC) both in the real and in the quaternion domain. The DC convolution separates the spatial correlation from the channels correlation reducing drastically the number of parameters and the number of operations. The rest of the paper is structured as follows: section 2 introduces the problem of monocular visual odometries along with related works; section 3 provides the mathematical background and describes in detail the quaternion convolution (section 3.1) and the depth-wise separable convolution (section 3.2); section 5 illustrates the proposed method with the implementation details; section 6 discusses the results and the conclusion is drawn in section 7.

2. Related Works

Visual Odometry has a long history in the computer vision community[5, 6, 7, 8, 9]. However, thanks to the precise and clean geometric formulation of the problem, the vast majority of the state-of-art monocular VO systems are based on the following standard pipeline [10, 11]:

Image sequence: the input of the pipeline is an ordered sequence of images collected by the camera in consecutive instants. In order to match features, it is important that two consecutive images have a sufficient scene overlap.

Feature Detection: in this step salient keypoints are extracted by each image, where keypoints are patterns that are different from their neighbours and can be easily identified in a different pose and orientation. Common algorithms such as SIFT [12] or FAST[13] consist in applying a feature-response function over the entire image and detect features as local maxima.

Feature Description: in which each feature is converted into a compact representation that can be matched with other descriptors.

Feature Matching: the feature descriptors in consecutive images are matched according to a similarity measure.

Motion Estimation: the motion of the consecutive frames is computed using the correspondences between the features descriptors in the two images.

Bundle Adjustment: in this step the result is refined by optimizing the reprojection error on the entire sequence of images. This is the most costly operation.

Even tough applications based on this pipeline achieved excellent results, there is always a trade-off between performance and consistency, and the right solution must be chosen carefully considering both the navigation environment and the requirements. Moreover, monocular VO approaches based on geometry are not able to recover the global scale, that must be recovered

through external measurements. On the other hand, deep learning have been rarely used to tackle VO problems, and some existing approaches require a pre-processed optical-flow as input [14]. DeepVO, introduced in [3], was the first deep learning model to estimate the 6 Dof of the poses directly from a sequence of RGB images. The model is composed of a convolutional neural network (CNN) to extract local descriptors and a recurrent neural network (RNN) to match the extracted features in consecutive frames. DeepVO will be used as a baseline model and its performances will be compared to those of the same model with the addition of quaternion convolution, separable depth-wise convolution and both. Quaternion convolutional networks (QCNN) for image processing were introduced in [15] in order to give colored images a meaningful representation through the quaternion algebra. In particular a pixel is represented as a single quaternion rather than a vector in \mathcal{R}^3 . In this way it is possible to interpret pixel multiplications (with the Hamilton product) as rotations in the color space. The effect of this representation is investigated in [16] where a QCNN based autoencoder, trained only on gray scale images, is able to perfectly reproduce colored images at test time. Depth-wise separable convolution (DC) firstly appeared in [17] and was popularized by its extensive application in the famous Xception[18] and MobileNets[19] architectures. DC allows to drastically reduce the number of parameters and operation without harming the accuracy. More details will be given in section 3.2.

3. Background

This section describes the mathematical foundation of the proposed approach. In particular the first part introduces quaternions and quaternion convolution, while the second part is about the detph-wise separable convolution.

3.1. Quaternions

The quaternion algebra \mathbf{H} was introduced by Hamilton in 1843 as an extension to the complex algebra. A quaternion has a real component and three imaginary components i, j, k , with the property that

$$i^2 = j^2 = k^2 = ijk = -1$$

Given two quaternions

$$Q_1 = r_1 + x_1i + y_1j + z_1k$$

and

$$Q_2 = r_2 + x_2i + y_2j + z_2k$$

, they can be summed, multiplied by a scalar and multiplied by each other according to the following formulae:

$$\begin{aligned} Q_1 + Q_2 &= (r_1 + r_2) + (x_1 + x_2)i + \\ &\quad (y_1 + y_2)j + (z_1 + z_2)k \\ \lambda Q_1 &= \lambda r_1 + \lambda x_1i + \lambda y_1j + \lambda z_1k \\ Q_1 \otimes Q_2 &= (r_1r_2 - x_1x_2 - y_1y_2 - z_1z_2) + \end{aligned} \tag{1}$$

$$\begin{aligned}
& (r_1x_2 + x_1r_2 + y_1z_2 - z_1y_2)i+ \\
& (r_1y_2 - x_1z_2 + y_1r_2 + z_1x_2)j+ \\
& (r_1z_2 + x_1y_2 - y_1x_2 + z_1r_2)k
\end{aligned}$$

where \otimes is the Hamilton product and it is the core of the Quaternion Convolution (QC). In standard convolution (SC) each pixel is represented as a three channels (RGB) feature vector, while in QC a pixel is a single quaternion where the imaginary parts are its RGB components and the real part is the gray scale image.

$$Q(p) = Gray(p) + R(p)i + G(p)j + B(p)k \quad (2)$$

Let $I \in \mathcal{Q}^{N \times N}$ be the image and $W \in \mathcal{Q}^{L \times L}$ the filter, both in the quaternion domain, then the QC can be defined as:

$$I \circledast W[k, k'] = \sum_{l=1}^L \sum_{l'=1}^L W_{l,l'} \otimes I_{k+l, k'+l'} \quad (3)$$

The peculiarity of this operation is that the information about the color space is preserved, whereas in the standard convolution the contributions from the RGB channels are summed. In QC the color space is modeled in the quaternion domain and each pixel, each weight of the network and each element of the intermediate feature maps are represented in this domain. By replacing the SC with the QC, the number of parameters and operations increases by a factor 4. However it was shown [15] that quaternion convolutional networks have good performances even if the number of kernels in each layer is reduced to match those of the real value convolution, hence input and output channels are divided by $\sqrt{4} = 2$. Quaternion convolution acts as a regularizer and reduces the degrees of freedom of the trainable parameters, as explained in [16]. In order to exploit this property, we further reduced the number of parameters dividing both input and output channels by 4 (instead of 2). Thanks to this optimization, the number of parameters of the convolutional networks drops from 1.3M to 416K as shown in table 1.

Model Name	CNN Parameters
DeepVO	14.6M
QDeepVO	3.7M
DeepVO DSC	1.6M
QDeepVO DSC	416K

Table 1

This table compares the number of parameters in the convolutional part of the three proposed models and the baseline

3.2. Depth-Wise Separable Convolution

Depth-Wise Separable Convolution (DC) splits the correlation of the spatial features and the features channels in two separate steps and consequently reducing the number of parameters. In standard convolution, each output channel is the result of the sum of the activations of

N kernels, where N is the number of input channels. Let $o^m[x]$ be the channel *cout* of the convolution at position x where $x \in R^2$. The standard convolution equation can be written as:

$$o^{cout}[x] = \sum_{cin} I * k_{cin} = \sum_{cin} \sum_y I[x+y]k_{cin}^{cout}[y] \quad (4)$$

Hence, a convolution of a feature map with size $N \times N$ with M input channels with K filters (output channels) with size $L \times L$ requires MKL^2 parameters and N^2MKL^2 operations. In DC convolution, first the spatial convolution is computed independently for each input channel:

$$o^{cout}[x] = I * k^{cout} = \sum_y I[x+y]k^{cout}[y] \quad (5)$$

Of course the number of output channels equals the number of input channels. This computation requires ML^2 parameters and N^2ML^2 operations. Then a 1×1 convolution correlates the channels:

$$o^{cout}[x]' = \sum_{cin} o^{cout} * K_{1 \times 1 cin}^{cout} \quad (6)$$

This computation requires MK parameters and N^2MK operations. Hence in DC convolution the total number of parameters is $MK + ML^2$ and the number of operations is $N^2ML^2 + N^2MK$. With simple algebraic manipulations, it can be shown that the reduction factor in both the number of parameters and operations is $\frac{KL^2}{K+L^2}$.

4. Dataset

For training and testing our models we used the famous KITTI dataset [20] and in particular the KITTI VO/SLAM benchmark, containing 22 sequences of RGB images, where the first 11 have the ground truth pose matrix associated to each image in the sequence. Figure 4 shows two consecutive images in a sequence while figure 4 shows a ground truth trajectory computed from the ground truth file.

5. Method

The three architectures presented in this section are derived from the DeepVO network. The network is composed of 9 convolutional blocks, each with 2D convolution, ReLU activation, Batch Normalization and Dropout. After the convolutional blocks there are two stacked LSTM that receive as input the sequence of feature maps extracted by the convolutional layers (more details are reported in table 3).

The input of the network is a sequence of raw RGB images and the output consists in the 3 components of the translation vector expressed in meters and the 3 euler angles expressed in degrees for each image in the sequence. The three variants of DeepVO introduced in this paper are:

Quaternion DeepVO: with Quaternion convolution (as described in section 3.1) instead of standard convolution.



Figure 1: Example of two consecutive images from the KITTI dataset

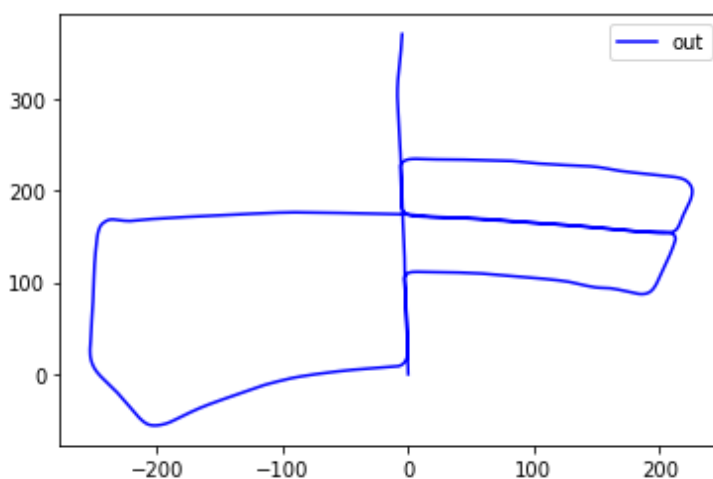


Figure 2: Example of a trajectory from ground truth.

DeepVO DSC: with Depth-Wise Separable convolution instead of standard convolution.

Quaternion DeepVO DSC: with both Quaternion convolution and Depth-Wise Separable convolution. The three networks have the same type and number of layers, but the implementation of the convolution algorithm changes as explained in section 3. The advantage is a considerable reduction in the number of parameters. In particular with Quaternion DeepVO, DeepVO DSC and Quaternion DeepVO DSC we obtained a reduction in the number of parameters of 13M, 10.9M, 14.2M respectively in the three cases (these results are summarized in table 2). For the quaternion implementation we opted for a *full quaternion representation*, where the real part represents the gray scale image and the imaginary parts the RGB components, with respect to a *pure quaternion representation*, in which the real part is set to zero.

Model Name	textbfCNN parameters	Translation loss (%)	Rotational Loss (deg)
DeepVO	14.6M	10.17	8.76
QDeepVO	3.7M	9.48	7.93
DeepVO DSC	1.6M	12.54	10.79
QDeepVO DSC	416K	9.74	7.13

Table 2

This table summarizes the results both in term of model size and translation and rotational losses.

DeepVO Architecture	
input	(bs,3,1280,384)
	kernels = 64, kernel size = 7, stride = 2
conv1	kernels = 128, kernel size = 5, stride = 2
conv2	kernels = 256, kernel size = 3, stride = 2
conv3	kernels = 256, kernel size = 3, stride = 1
conv3_1	kernels = 512, kernel size = 3, stride = 2
conv1	kernels = 512, kernel size = 3, stride = 1
conv4	kernels = 512, kernel size = 3, stride = 2
conv4_1	kernels = 512, kernel size = 3, stride = 1
conv5	kernels = 1024, kernel size = 3, stride = 2
LSTM1	hidden size = 1000
LSTM2	hidden size = 1000
Linear	out features = 6

Table 3

DeepVO architecture. To increase readability, the table did not state that after each convolutional block there is a ReLU activation function, batch normalization and dropout with rate 0.2.

6. Results

The three networks were trained for 200 epochs with learning rate 10^{-3} on 5 of the 11 labelled sequences (in particular on sequences 00, 01, 02, 08, 09), while the remaining 6 labelled sequences are used for testing. The average translational RMSE drift in percentage on lengths of 100m - 800m and the average rotational RMSE drift expressed in $\text{deg}/100m$ on lengths of 100m-800m are reported in table 2. The learning curves of the models are reported in figure 3 while tables 4 and 5 show the true and predicted trajectories in the test sequences. The results show that the best performances are obtained by our Quaternion DeepVO model, while the metrics of our Quaternion DeepVO DSC model are comparable to those of the baseline, but with a significant reduction in the number of parameters and operations.

7. Conclusion

In this paper we presented a novel end-to-end deep learning approach for monocular visual odometry. The proposed method was based on the DeepVO, a state-of-art deep learning architecture for VO, and introduced quaternion convolution and depth-wise separable convolution. We obtained the best result in the quaternion domain, with 10.9M parameter less and a comparable

result with the depth-wise convolution in the quaternion domain, with 14.2M parameter less. However the changes we introduced apply only on the convolutional part. We left as a future work the investigation of other optimization strategies for the recurrent part, and in particular, considering the success of the quaternion convolution, it would be of interest to keep also the LSTM part in the quaternion domain.

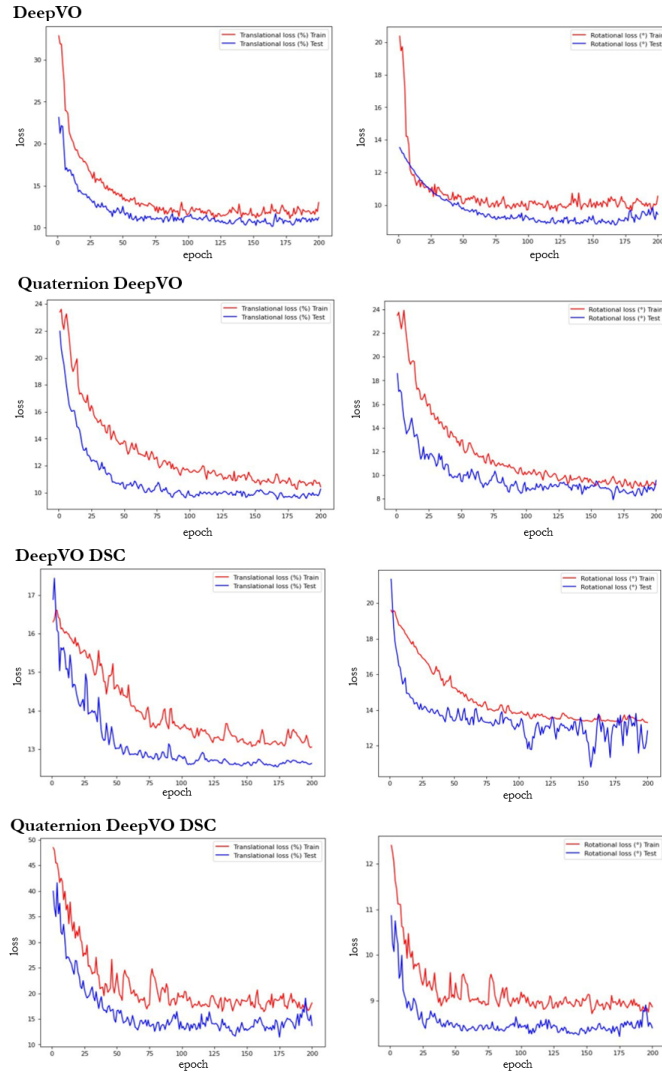
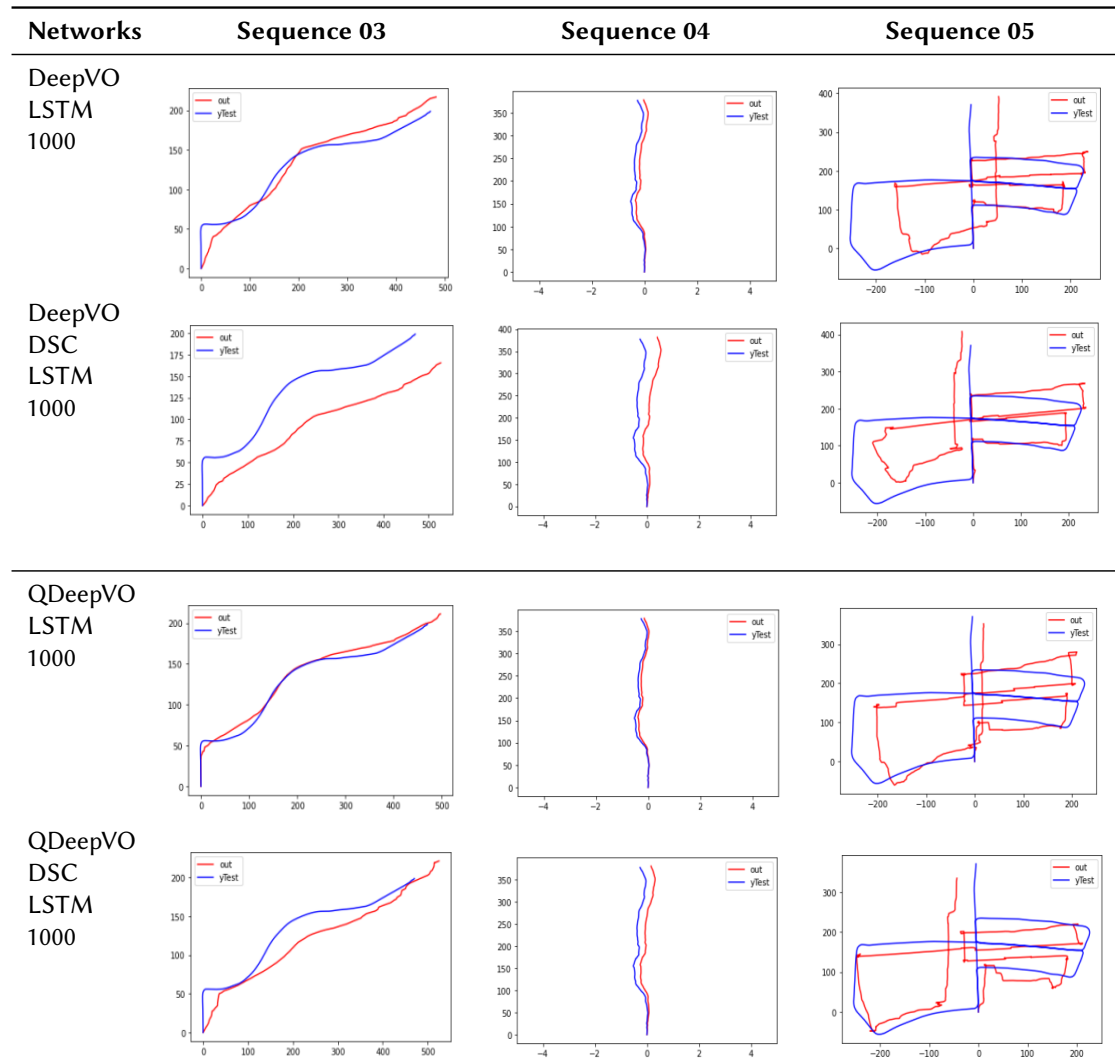


Figure 3: The figure shows the learning curves of the models. The first and second columns represents respectively the translational and rotational losses, where the red and blue curves are the train and test curves.

Table 4

True (blue) and predicted (red) trajectories on the test sequences 03, 04, 05



Acknowledgments

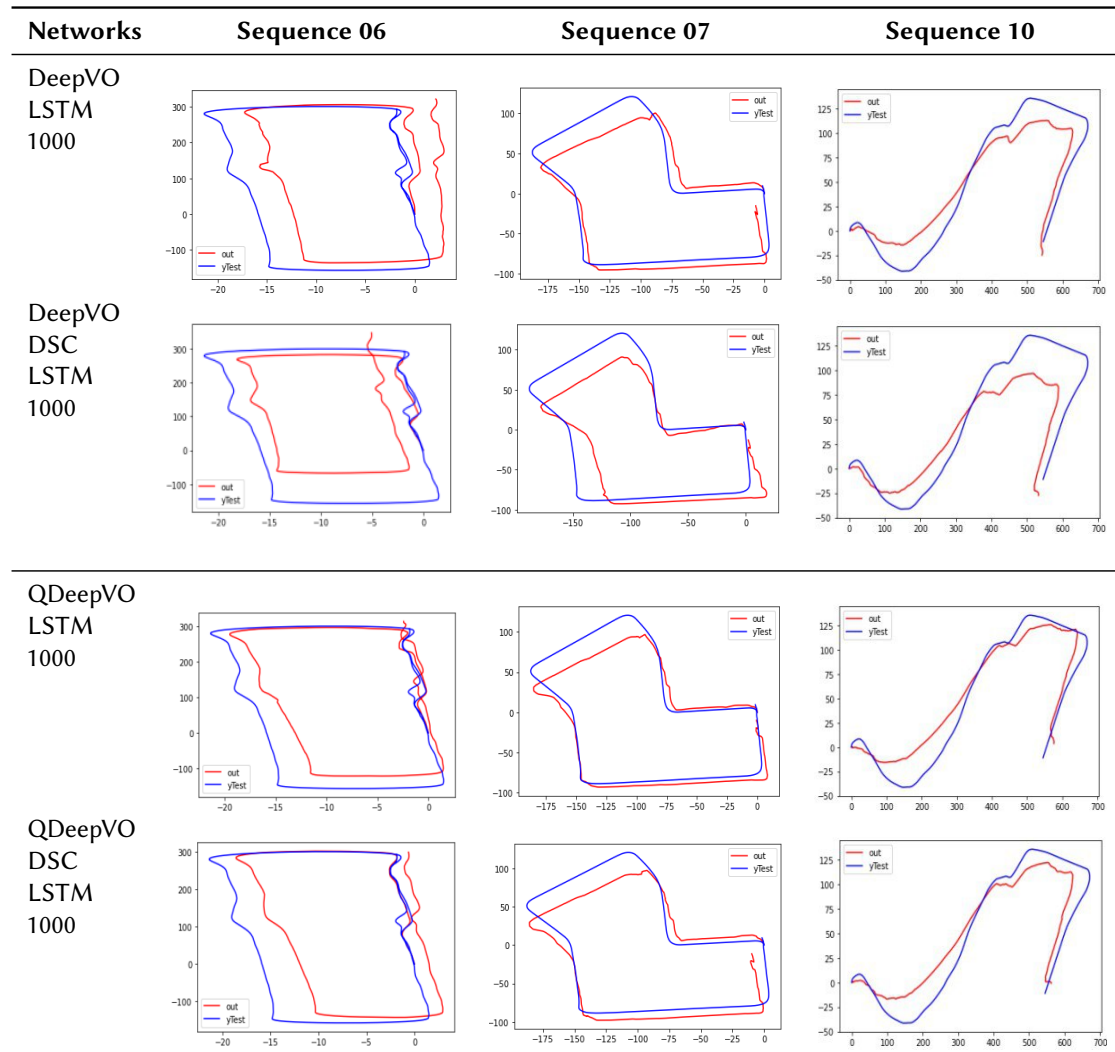
The initial work was carried out with the help of the students Alessandro Lambertini and Denise Landini and supported by the Hermes-WIRED project within the Large Research Projects grant framework 2020 funded by Sapienza University of Rome.

References

- [1] D. Scaramuzza, F. Fraundorfer, R. Siegwart, Real-time monocular visual odometry for on-road vehicles with 1-point ransac, in: 2009 IEEE International conference on robotics

Table 5

True (blue) and predicted (red) trajectories on the test sequences 06, 07, 10



and automation, *Ieee*, 2009, pp. 4293–4299.

- [2] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, F. Moreno-Noguer, Pl-slam: Real-time monocular visual slam with points and lines, in: *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 4503–4508.
- [3] S. Wang, R. Clark, H. Wen, 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.
- [4] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

- [5] M. Srinivasan, S. Zhang, M. Lehrer, T. Collett, Honeybee navigation en route to the goal: visual flight control and odometry, *The Journal of experimental biology* 199 (1996) 237–244.
- [6] G. De Magistris, R. Caprari, G. Castro, S. Russo, L. Iocchi, D. Nardi, C. Napoli, Vision-based holistic scene understanding for context-aware human-robot interaction, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 13196 LNAI (2022) 310 – 325. doi:10.1007/978-3-031-08421-8_21.
- [7] C. F. Olson, L. H. Matthies, M. Schoppers, M. W. Maimone, Stereo ego-motion improvements for robust rover navigation, in: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, IEEE, 2001, pp. 1099–1104.
- [8] N. Brandizzi, S. Russo, G. Galati, C. Napoli, Addressing vehicle sharing through behavioral analysis: A solution to user clustering using recency-frequency-monetary and vehicle relocation based on neighborhood splits, *Information (Switzerland)* 13 (2022). doi:10.3390/info13110511.
- [9] D. Nistér, O. Naroditsky, J. Bergen, Visual odometry, in: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, Ieee, 2004, pp. I–I.
- [10] N. Brandizzi, S. Russo, R. Brociek, A. Wajda, First studies to apply the theory of mind theory to green and smart mobility by using gaussian area clustering, volume 3118, 2021, p. 71 – 76.
- [11] F. Fraundorfer, D. Scaramuzza, Visual odometry: Part i: The first 30 years and fundamentals, *IEEE Robotics and Automation Magazine* 18 (2011) 80–92.
- [12] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International journal of computer vision* 60 (2004) 91–110.
- [13] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: *European conference on computer vision*, Springer, 2006, pp. 430–443.
- [14] G. Costante, M. Mancini, P. Valigi, T. A. Ciarfuglia, Exploring representation learning with cnns for frame-to-frame ego-motion estimation, *IEEE robotics and automation letters* 1 (2015) 18–25.
- [15] X. Zhu, Y. Xu, H. Xu, C. Chen, Quaternion convolutional neural networks, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 631–647.
- [16] T. Parcollet, M. Morchid, G. Linares, Quaternion convolutional neural networks for heterogeneous image processing, in: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 8514–8518.
- [17] L. Sifre, S. Mallat, Rigid-motion scattering for texture classification, *arXiv preprint arXiv:1403.1687* (2014).
- [18] F. Chollet, Xception: Deep learning with depthwise separable convolutions, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.
- [19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, *arXiv preprint arXiv:1704.04861* (2017).
- [20] A. Geiger, P. Lenz, C. Stiller, R. Urtasun, Vision meets robotics: The kitti dataset, *International Journal of Robotics Research (IJRR)* (2013).