LS-VO: Learning Dense Optical Subspace for Robust Visual Odometry Estimation*

Gabriele Costante†,1 and Thomas A. Ciarfuglia†,1

Abstract—This work proposes a novel deep network architecture to solve the camera Ego-Motion estimation problem. A motion estimation network generally learns features similar to Optical Flow (OF) fields starting from sequences of images. This OF can be described by a lower dimensional latent space. Previous research has shown how to find linear approximations of this space. We propose to use an Auto-Encoder network to find a non-linear representation of the OF manifold. In addition, we propose to learn the latent space jointly with the estimation task, so that the learned OF features become a more robust description of the OF input. We call this novel architecture LS-VO. The experiments show that LS-VO achieves a considerable increase in performances in respect to baselines, while the number of parameters of the estimation network only slightly increases.

I. INTRODUCTION

Learning based Visual Odometry (L-VO) in the last few years has seen an increasing attention of the robotics community because of its desirable properties of robustness to image noise and camera calibration independence [1], mostly thanks to Convolutional Neural Networks (CNNs) representational power, which can complement current geometric solutions [2]. While current results are very promising, making these solutions easily applicable to different environments still presents challenges. One of them is that most of the approaches so far explored have not shown strong domain independence and suffer from high dataset bias, i.e. the performances considerably degrade when tested on sequences with motion dynamics and scene depth significantly different from the training data [3]. In the context of L-VO this bias is expressed in different Optical Flow (OF) field distribution in training and test data, due to differences in scene depth and general motion of the camera sensor. One possible explanation for the poor performances of learned methods on unseen contexts is that most current learning architectures try to extract both visual features and motion estimate as a single training problem, coupling the appearance and scene depth with the actual camera motion information contained in the OF input. Some works have addressed the problem with an unsupervised, or semi-supervised approach, trying to learn directly the motion representation and scene depth from some kind of frame-to-frame photometric error [4] [5] [6]. While very promising, these approaches are mainly devised for scene depth estimation and still fall short in terms of general performances on Ego-Motion estimation.

At the same time, previous research has shown how OF fields have a bilinear dependence on motion and inverse scene depth [7]. We suggest that this is the main reason for the low generalization properties shown by learned algorithms so far. Past research has shown that the high dimensional OF field, when scene depth can be considered locally constant, can be projected on a much lower dimensional linear space [8] [9]. However, when these conditions do not hold, the OF field subspace exists but is highly non-linear.

In this work we propose to exploit this knowledge, estimating the latent OF representation using an Auto-Encoder (AE) Neural Network architecture as a non-linear subspace approximator. AE networks are able to extract latent variable representation of high dimensional inputs. Since our aim is to make the Ego-Motion estimation more robust to OF fields that show high variability in their distribution, we do not simply use this subspace to directly produce motion prediction. Instead, we propose a novel architecture that jointly trains the
subspace estimation and Ego-Motion estimation so that the two network tasks are mutually reinforcing and at the same time able to better generalize OF field representation. To demonstrate the increased performances and reduced dataset bias with respect to high dynamical variation of the OF field, we test the proposed approach on a challenging scenario. We sub-sample the KITTI dataset [10], producing sequences that simulate high speed variations, then we train and test on sequences that are both different in appearance and subsampling rate.

II. RELATED WORKS

A. Ego-Motion estimation

1) Geometric Visual Odometry: G-VO has a long history of solutions. While the first approaches were based on sparse feature tracking, mainly for computational reasons, nowadays direct or semi-direct approaches are preferred. These approaches use the photometric error as an optimization objective. Research on this topic is very active. Engel at al. developed one of the most successful direct approaches, LSD SLAM, both for monocular and stereoscopic cameras [11], [12]. Forster et al. developed the Semi-Direct VO (SVO) [13] and its more recent update [14], which is a direct method but tracks only a subset of features on the image and runs at very high frame rate compared to full direct methods. Even if direct methods have gained most of the attention in the last few years, the ORB-SLAM algorithm by Mur-Artal et al. [15] reverted to sparse feature tracking and reached impressive robustness and accuracy comparable with direct approaches.

2) Learned Visual Odometry: Learned approaches go back to the early explorations by Roberts et al. [8], [16], Guizilini et al. [17], [18], and Ciarfuglia et al. [19]. As for the geometric case, the initial proposal focused on sparse OF features that, faithful to the there’s no free lunch theorem, explored the performances of different learning algorithms such as SVMs, Gaussian Processes and others. While these early approaches already showed some of the strengths of L-VO, it was only more recently, when Costante et al. [1] introduced the use of CNNs for feature extraction from dense optical flow, that the learned methods started to attract more interest. Since then a couple of methods have been proposed. Muller and Savakis [20] added the FlowNet architecture to the estimation network, producing one of the first end-to-end approaches. Clark et al. [21] proposed an end-to-end approach that merged camera inputs with IMU readings using an LSTM network. Through this sensor fusion, the resulting algorithm is able to give good results but requires sensors other than a single monocular camera. The use of LSTM is further explored by Wang et al. in [22], this time without any sensor fusion. The resulting architecture gives again good performances on KITTI sequences but does not show any experiments on environments with different appearance from the training sequences. On a different track is the work of Pillai et al. [23], that, like [18], looked at the problem as a generative probabilistic problem. Pillai proposes an architecture based on an MDN network and a Variational Auto-Encoder (VAE) to estimate the motion density given the OF inputs \( q(\mathbf{y} | \mathbf{x}) \) as a GMM. While Frame to Frame (F2F) performances are on a par with other approaches, they also introduce a loss term on the whole trajectory that mimics the bundle optimization that is often used in G-VO. The results of the complete system are thus very good. However, they use as input sparse KLT optical flow, since the joint density estimation for dense OF would become computationally intractable, meaning that they could be more prone to OF noise than dense methods.

Most of the described approaches claim independence from camera parameters. While this is true, we note that this is more an intrinsic feature of the learning approach than the merit of a particular architecture. The learned model implicitly learns also the camera parameters, but then it fails on images collected with other camera optics. This parameter generalization issue remains an open problem for L-VO.

B. Semi-supervised Approaches

Since dataset bias and domain independence are critical challenges for L-VO, it is not surprising that a number of unsupervised and semi-supervised methods have been recently proposed. However, all the architectures have been proposed as a way of solving the more general problem of joint scene depth and motion estimation, and motion estimation is considered more as a way of improving depth estimation. Konda and Memisevich [24] used a stereo pair to learn VO but the architecture was conceived only for stereo cameras. Ummenhofer and Zhou [4] propose the DeMoN architecture, a solution for F2F Structure from Motion (SfM) that trains a network end-to-end on image pairs, leveraging motion parallax. Their architecture is iterative, in the sense that they feed the last computed prediction as input to the next. Zhou et al. [5] proposed an end-to-end unsupervised system based on a loss that minimizes image warping error from one frame to the next. A similar approach is used by Vijayanarasimhan et al. [6] with their SfM-Net. SfM-Net has been devised to be trained with different degrees of supervision but the self-supervised one is similar to [5], since it is based on the minimization of the F2F photometric reprojection error. All these approaches are devised mainly for depth estimation and the authors give little or no attention to the performances on VO tasks. Nonetheless, the semi-supervised approach is one of the more relevant future directions for achieving domain independence for L-VO, and we expect that this approach will be integrated in the current research on this topic.

C. Optical Flow Latent Space Estimation

The semi-supervised approaches described in Section II-B make evident an intrinsic aspect of monocular camera motion estimation, that is, even when the scene is static, the OF field depends both on camera motion and scene depth. This relationship between inverse depth and motion is bilinear and well known [25] and is at the root of scale ambiguity in monocular VO. However, locally and under certain hypothesis of depth regularity, it is possible to express...
the OF field in terms of a linear subspace of OF basis vectors. Roberts et al. [16] used Probabilistic-PCA to learn a lower dimensional dense OF subspace without supervision, then used it to compute dense OF templates starting from sparse optical flow. They then used it to compute Ego-Motion. Herdtweck and Cristóbal extended the result and used Expert Systems to estimate motion [26]. More recently, a similar approach to OF field computation was proposed by Wulff and Black [9] that complemented the PCA with MRF, while Ochs et al. [27] did the same by including prior knowledge with an MAP approach. These methods suggest that OF field, which is an intrinsically high dimensional space generated from a non-linear process, lies on an ideal lower dimensional manifold that sometimes can be linearly locally approximated. However, modern deep networks are able to find latent representation of high dimensional image inputs, and in this work we use this intuition to explore this OF latent space estimation.

III. CONTRIBUTION

Inspired by the early work of Roberts on OF subspaces [7], and by recent advances in deep latent space learning [28], we propose a network architecture that jointly estimates a low dimensional representation of dense OF field using an Auto-Encoder (AE) and at the same time computes the camera Ego-Motion estimate with a standard Convolutional network, as in [1]. The two networks share the feature representation in the decoder part of the AE, and this constrains the training process to learn features that are compatible with a general latent subspace. We show through experiments that this joint training increases the Ego-Motion estimation performances and generalization properties. In particular, we show that learning the latent space and concatenating it to the feature vector makes the resulting estimation considerably more robust to domain change, both in appearance and in OF field dynamical range and distribution.

We train our network both in an end-to-end version, using deep OF estimation, and with standard OF field input, in order to explore the relative advantages and weaknesses. We show that while the end-to-end approach is more general, precomputed OF still has some performance advantages.

In summary our contributions are:

- A novel end-to-end architecture to jointly learn the OF latent space and camera Ego-Motion estimation is proposed. We call this architecture Latent Space-VO (LS-VO).
- The strength of the proposed architecture is demonstrated experimentally, both for appearance changes, blur, and large camera speed changes.
- Effects of geometrically computed OF fields are compared to end-to-end architectures in all cases.
- The adaptability of the proposed approach to other end-to-end architectures is demonstrated, without increasing the chances of overfitting them, due to parameters increase.

IV. LEARNING DEEP OF SUBSPACES

Given an optical flow vector \( \mathbf{u} = (u_x^T, u_y^T)^T \) from a given OF field \( \mathbf{x} \), [7] [9] approximate it with a linear relationship:

\[
\mathbf{u} \approx Wz = \sum_{i=1}^{l} z_i \mathbf{w}_i
\]

where the columns of \( W \) are the basis vectors that form the OF linear subspace and \( z \) is a vector of latent variables. This approximation is valid only if there are some regularities of scene depth and is applicable only to local patches in the image. The real subspace is non-linear in nature and, in this work, we express it as a generic function \( \mathbf{u} = D(z) \) that we learn from data by using the architecture described in the following.

A. Latent Space Estimation with Auto-Encoder Networks

Let \( y \in \mathbb{R}^6 \) be the camera motion vector and \( \mathbf{x} \in \mathbb{R}^{w \times h} \) the input OF field, computed with some dense method, where \( x_{i,j} = u_{i,j} \) is a 2-dimensional vector of the field. Both can be viewed as random variables with their own distributions. In particular, we make the hypothesis that the input images lie on a lower dimensional manifold, as in [29], and thus also the OF field lies on a lower dimensional space \( \mathcal{O} \) with a distance function \( S(x_1, x_2) \), where \( x_i \in \mathcal{O} \). The true manifold is very difficult to compute, so we look for an estimate \( \hat{\mathcal{O}} \approx \mathcal{O} \) using the model extracted by an encoding neural network.

Let \( z \in \mathbb{Z} \subset \mathbb{R}^l, l < w \times h \) be a vector of latent random variables that encodes the variabilities of OF field that lies on this approximate space. The decoder part of the AE can be seen as a function

\[
D(z; \theta_d) = D(W_d z + b_d)
\]

where \( \theta_d = (W_d, b_d) \) is the set of learnable parameters of the network that is able to generate a dense optical flow from a vector of latent variables \( z \). Note that the AE works similarly to a non-linear version of PCA [28]. We define the set \( \hat{\mathcal{O}} = \{ D(z; \theta_d) \mid z \in \mathbb{Z} \} \) as our approximation of the OF field manifold and use the logarithmic Euclidean distance (as described in Section IV-B as a loss function) as an approximation of \( S(D(z_1), D(z_2)) \). Using this framework the problem of estimating the latent space is carried out by the AE network, where the Encoder part can be defined as the function \( z = E(\mathbf{x}; \theta_e) \).

While in [23] the AE is used to estimate motion, and \( z \) are the camera translation and rotations, here we follow a different strategy. We compute the latent space for a two-fold purpose: we use the latent variables as an input feature to the motion estimation network and we learn this latent space together with the estimator, thus forcing the estimator to learn features compatible with the encoder representation. Together these two aspects make the representation more robust to domain changes.

B. Network Architecture

The LS-VO network architecture in its end-to-end form is shown in Figure 2. It is composed of two main branches,
one is the AE network and the other is the convolutional network that computes the regression of motion vector \( y \). The OF extraction section is Flownet [30], for which we use the pre-trained weights. We experimented with the fine-tuning of this part of the network on KITTI dataset, but the result was a degraded performance due to overfitting.

The next layers are convolutions that extract features from the computed OF field. After the first convolutional layers (conv1, conv2 and conv3), the network splits into the AE network and the estimation network. The two branches share part of the feature extraction convolutions, so the entire network is constrained in learning a general representation that is good for estimation and latent variable extraction. The Encoder is completed by another convolutional layer, that brings the input \( x \) to the desired representation \( z \), and its output is fed both in the Decoder and concatenated to the feature extracted before. The resulting feature vector, composed of latent variables and convolutional features is fed into a fully connected network that performs motion estimation. The details of the network are summarized in Table I.

The AE is trained with a pixel-wise squared Root Mean Squared Log Error (RMSLE) loss:

\[
L_{AE} = \sum_i \left( \log(\hat{u}(i) + 1) - \log(u(i) + 1) \right)^2
\]  

where \( \hat{u}(i) \) is the predicted OF vector for the i-th pixel, and \( u(i) \) is the corresponding input to the network, and the logarithm is intended as an element-wise operation. The loss for the Ego-Motion estimation branch is:

\[
L_{EM} = \sum_i \left( \|\hat{\tau} - \tau\|^2 + \beta \|\hat{\theta} - \theta\|^2 \right)
\]  

where the \( \tau \) is camera translation vector, \( \theta \) is the rotation vector in Euler notation, and \( \beta \) is a scale factor that has been cross-validated (\( \beta = 20 \) for our experiments).

The networks are implemented in Keras/Tensorflow, using an Nvidia Titan Xp. The number of network parameters to train are more than the parameters used in [1], but, since the estimation network receives only the low dimensional latent vector (in our experiments \( l = 64 \)), the increase in the number of parameters to learn in the estimation branch is negligible. We therefore increment the expressiveness of the network, without increasing the chances of overfitting the data.

C. OF field distribution

As mentioned in Section IV-A, the OF field has a probability distribution that lies on a manifold with lower dimensionality than the number of pixels of the image. We can argue that the actual density depends on the motion of the camera as much as the scene depth of the images collected. In this work, we test generalization properties of the network for both aspects:

- For the appearance we use the standard approach to test on completely different sequences than the ones used in training.
- For the motion dynamics, we sub-sample the sequences, thus multiplying the OF dynamics by the same factor.
- To further test OF distribution robustness, we also test the architecture on blurred images, as in [1].

Examples of the resulting OF field are shown in Figure 3, while an example of a blurred OF field is shown in Figure 4. In both images there are evident differences both in hue and saturation, meaning that both modulus and phase of the OF vectors change.
TABLE I: LS-VO network architecture

| Layer name | Kernel size | Stride | output size | activation | Notes |
|------------|-------------|--------|-------------|------------|-------|
| Input      | -           | -      | -           | (94, 300, 2)| -     |
| conv1      | 7 × 7 × 64  | 2 × 2  | (47, 150, 64)| ReLU       |       |
| conv2      | 5 × 5 × 64  | 1 × 1  | (47, 150, 64)| ReLU       |       |
| conv3      | 3 × 3 × 64  | 4 × 4  | (12, 38, 64)| ReLU       |       |
| conv4      | 3 × 3 × 64  | 1 × 1  | (12, 38, 64)| ReLU       | latent variables |
| upconv1    | -           | -      | (48, 192, 6)| ReLU       |       |
| crop       | -           | -      | (47, 190, 6)| -          |       |
| upconv2    | 1 × 1 × 6   | 1 × 1  | (94, 300, 2)| Linear     |       |
| maxpool    | 2 × 2       | 2 × 2  | (6, 19, 64) | -          | then flattened in 7296 vector |
| concat     | -           | -      | (36480)     | -          | concat of maxpool and conv3 features |
| dense1     | -           | -      | (1000)      | ReLU       |       |
| dense1     | -           | -      | (1000)      | ReLU       |       |
| dense1     | -           | -      | (6)         | Linear     | output layer |

![Fig. 3: Examples of the OF field intensity due to different sub-sampling rates of the original sequences. In the left are the OF field extracted with Brox algorithm (BF) [31], while on the right the ones extracted with Flownet [30]. While the BF fields look more crisp, they require parameter tuning, while the Flownet version is non-parametric at test time.](image)

(a) no sub-sampling - 10Hz  
(b) sub-sampling 2 - 5Hz  
(c) sub-sampling 3 - 3.33Hz

![Fig. 4: Examples of OF fields obtained applying gaussian blur to image sequences. (a) The image and its blurred variant is shown, with blur radius 10. (b) The corresponding OF fields. Note the huge change in OF distribution.](image)

(a) Standard and blurred image  
(b) Standard and blurred OF field

V. EXPERIMENTAL RESULTS

A. Data and Experiments set-up

To train and test the proposed network we used the KITTI dataset [10], using the sequences 00 to 07 for training and the 08, 09 and 10 for test, as is common practice. The images are all around 1240 × 350, and we resize them to 300 × 94. The original sequences show some variability in speed, since the car travels at speeds of up to 60km/h, but the distribution of OF field is still limited since the possible combinations of linear and rotational speeds are limited in this dataset. As mentioned in Section IV-C we extend the variability of OF field distribution performing some data augmentation. Firstly, we sub-sample the sequences by 2 and 3 times, to generate virtual sequences that have OF vectors with very different intensity. In Figure 3 an example of the different dynamics is shown. We indicate the standard sequences by the \(d_1\) subscript, and the sequences sub-sampled by 2 and 3 times by \(d_2\) and \(d_3\), respectively. In addition to this, we generate blurred versions of the test sequences, with gaussian blur, as in [1]. Then we perform three kinds of experiment and compare the results. The first is a standard training and test on \(d_1\) sequences. This kind of test explores the generalization properties on appearance changes alone. In the second kind of experiment we train all the networks on the sequences \(d_1\) and \(d_3\) and test on \(d_2\). This helps us to understand how the networks perform when both appearance and OF dynamics change. The third experiment is training on \(d_1\) sequences, and testing on the on the blurred versions of the test set.

The proposed architecture is end-to-end, since it computes the OF field through a Flownet network. However, as a baseline, we decided to test the performances of all the architecture on a standard geometrical OF input, computed as in [31], and indicated as BF in the following. We note that BF input requires parameter tuning, while Flownet input does not. Even if in the following experiments BF often performs better, this is expected, since Flownet features are not fine-tuned for this dataset. This makes the comparison more informative in the sense of generalization performances of the end-to-end network.

In addition, for both kind of networks, we tried to train on floating point OF field values and on RGB quantized version. We found that BF performed better when quantized, while Flownet performed better on floating point values. We therefore use these parametrizations and use RGB quantization for visualization purposes in both cases.

B. Discussion

The experiments described in Section V-A have been evaluated with KITTI devkit [10], and the output plots have been reported in Figures 5 and 6. In both Figures, the...
Fig. 5: Comparison between the four network architectures, with no sequence sub-sampling. It is evident that the LS-VO network outperforms the ST equivalent, and in the case of the BF OF inputs it is almost always better by a large margin.

![Graphs showing performance comparisons](image)

Fig. 6: Comparison between the four network architectures on the sequences sub-sampled by 2 (5Hz). Again, the LS-VO architecture outperforms the other, except for speed around 60Km/h.

|                  | Single Task (ST) | Multi Task (LS-VO) |
|------------------|------------------|--------------------|
| **BF**           | 8.06%            | 6.98%              |
| **Rot.**         | 0.0205           | 0.0199             |
| **Flownet**      | 13.34%           | 10.79%             |
| **Rot.**         | 0.0346           | 0.0330             |

**TABLE II:** Average performances on standard sequences (no sub-sampling)

|                  | Single Task (ST) | Multi Task (LS-VO) |
|------------------|------------------|--------------------|
| **BF**           | 9.43%            | 7.13%              |
| **Rot.**         | 0.0360           | 0.0296             |
| **Flownet**      | 12.31%           | 10.53%             |
| **Rot.**         | 0.0384           | 0.0321             |

**TABLE III:** Average performances on sub-sampled sequences (d2 - 5Hz)

Fig. 7: Trajectories computed on the sub-sampled sequences for all architectures (d2 - 5Hz).

upper sub-plots, (a) and (b), represent the translational and rotational errors averaged on sub-sequences of length 100m up to 800m. The lower plots represent the same errors, but averaged on vehicle speed (Km/h). The horizontal axis limits for the lower plots, in Figures 5 and 6 are different, since the sub-sampling is seen by the evaluation software...
as an increase in vehicle speed. In Tables II and III, the total average translational and rotational errors are reported, respectively for the standard case and for the sub-sampled case.

From Figures 5a and 5b, we observe that the BF-fed architectures outperform the Flownet-fed networks by a good margin. This is expected, since BF OF fields have been tuned on the dataset to be usable, while Flownet has not been fine-tuned on KITTI sequences. In addition, the LS-VO networks perform almost always better than, or on a par with, the corresponding ST networks. When we consider Figures 5c and 5d, we observe that the increase in performance from ST to LS-VO appears to be slight, except in the rotational errors for the Flownet architecture. However, the difference between the length errors and the speed errors is coherent if we consider that the errors are averaged. Therefore, the speed values that are less represented in the dataset are probably the ones that are more difficult to estimate, but at the same time their effect on the general trajectory estimation is consequently less important. Table II summarizes what we noticed here, that the LS-VO always gives a good increase in general performances over the ST networks with both input types.

When we consider the second type of experiment, we expect that the general performances of all the architectures should decrease, since the task is more challenging. At the same time we are interested in probing the robustness and generalization properties of the LS-VO architectures over the ST ones. From Figures 6a and 6b, we notice that, while all the average errors for each length increase, they increase much more for the two ST variants. This demonstrates that the LS-VO networks are much more able to learn a representation that is less prone to this kind of input domain change. If we consider the errors depicted in Figures 6c and 6d, we observe that the LS-VO networks perform better than the ST ones, except on speed around 60Km/h, where they are on par. This is understandable, since the networks have been trained on $d_1$ and $d_3$, that correspond to very low and very high speeds, so the OF in between them are the less represented in the training set. However, the most important consideration here is that the Flownet LS-VO architecture in all cases performs better than in the experiments depicted in Figure 5. For this reason, on the third kind of experiment, we focus only on the end-to-end architecture. The results are shown in Figure 8 and Table IV. We know already that the blur on the images is enough to cause failures in most G-VO algorithms, and that CNN-VO systems still perform well on this scenario. Still, in Figure 8, we observe that the LS-VO again give a huge improvement to the performances when compared to the standard network.

This experiments demonstrate that the LS-VO architecture is particularly apt to help end-to-end networks in extracting a robust OF representation. This is an important result, since this architecture can be easily included in other end-to-end approaches, increasing the estimation performances by a good margin, but without significantly increasing the number of parameters for the estimation task, making it more robust to overfitting, as mentioned in Section IV-B.

VI. CONCLUSIONS

This work presented LS-VO, a novel network architecture for estimating monocular camera Ego-Motion. The architecture is composed by two branches that jointly learn a latent space representation of the input OF field, and the camera motion estimate. The joint training allows for the learning of OF features that take into account the underlying structure of a lower dimensional OF manifold. The proposed architecture has been tested on KITTI dataset and on modified version of it, in order to test the robustness to domain variability in both appearance and OF dynamic range. LS-VO network outperformed the single branch network on most benchmarks, and in the others performed at the same level. The new architecture is lean and easy to train and shows good generalization performances. The results provided here are promising and encourage further exploration of OF field latent space learning for the purpose of estimating camera Ego-Motion. All the code, datasets and trained models are made available online [32].

REFERENCES

[1] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, “Exploring Representation Learning with CNNs for Frame-to-Frame Ego-Motion Estimation,” Robotics and Automation Letters, IEEE, vol. 1, no. 1, pp. 18–25, 2016.

[2] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza, “Learning-based image enhancement for visual odometry in challenging hdr environments,” arXiv preprint arXiv:1707.01274, 2017.
[3] T. Tommasi, N. Patricia, B. Caputo, and T. Tuytelaars, “A deeper look at dataset bias,” in Pattern Recognition: 37th German Conference, GCPR 2015, Aachen, Germany, October 7-10, 2015, Proceedings. Springer International Publishing, 2015, pp. 504–516.

[4] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, “DeMoN: Depth and Motion Network for Learning Monocular Stereo,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[5] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[6] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, “SIM-Net: Learning of structure and motion from video,” arXiv preprint arXiv:1704.07804, 2017.

[7] R. J. W. Roberts, “Optical flow templates for mobile robot environment understanding,” Ph.D. dissertation, Georgia Institute of Technology, 2014.

[8] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. R. Balch, “Memory-based learning for visual odometry,” in 2008 IEEE International Conference on Robotics and Automation (ICRA), 2008, pp. 47–52.

[9] J. Wulf and M. J. Black, “Efficient sparse-to-dense optical flow estimation using a learned basis and layers,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The KITTI dataset,” The International Journal of Robotics Research, p. 0278364913491297, 2013.

[11] J. Engel, T. Schöps, and D. Cremers, “LSD-SLAM: Large-scale direct monocular SLAM,” in European Conference on Computer Vision (ECCV). Springer, 2014, pp. 834–849.

[12] J. Engel, J. Stücker, 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.

[13] 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.

[14] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “SVO: Semidirect visual odometry for monocular and multicamera systems,” IEEE Transactions on Robotics, vol. 33, no. 2, pp. 249–265, 2017.

[15] 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.

[16] R. Roberts, C. Potthast, and P. Dellaert, “Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 57–64.

[17] T. Guizilini and P. Ramos, “Visual odometry learning for unmanned aerial vehicles,” in 2011 IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 6213–6220.

[18] ———, “Semi-parametric models for visual odometry,” in 2012 IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 3482–3489.

[19] T. A. Cliartuglia, 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.

[20] P. Muller and A. Savakis, “Flowdometry: An optical flow and deep learning based approach to visual odometry,” in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), March 2017, pp. 624–631.

[21] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, “ViNet: Visual-inertial odometry as a sequence-to-sequence learning problem,” in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., 2017, pp. 3995–4001.

[22] 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), May 2017, pp. 2043–2050.

[23] S. Pillai and J. J. Leonard, “Towards visual ego-motion learning in robots,” arXiv preprint arXiv:1705.10279, 2017.

[24] K. R. Konda and R. Memisevic, “Learning visual odometry with a convolutional network,” in International Conference on Computer Vision Theory and Applications (VISAPP), 2015, pp. 486–490.

[25] D. J. Heeger and A. D. Jepson, “Subspace methods for recovering rigid motion I: Algorithm and implementation,” International Journal of Computer Vision, vol. 7, no. 2, pp. 95–117, 1992.

[26] C. Herdtweck and C. Curio, “Experts of probabilistic flow subspaces for robust monocular odometry in urban areas,” in 2012 IEEE Intelligent Vehicles Symposium, 2012 IEEE, June 2012, pp. 661–667.

[27] M. Ochs, H. Bradler, and R. Mester, “Learning rank reduced interpolation with Principal Component Analysis,” in Intelligent Vehicles Symposium (IV), 2017 IEEE, June 2017, pp. 1126–1133.

[28] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, http://www.deeplearningbook.org.

[29] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros, “Generative visual manipulation on the natural image manifold,” in European Conference on Computer Vision (ECCV). Springer, 2016, pp. 597–613.

[30] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, “FlowNet: Learning optical flow with convolutional networks,” in 2015 IEEE International Conference on Computer Vision (ICCV), Dec 2015, pp. 2758–2766.

[31] T. Brox, A.Bruhn, N. Papenberg, and J. Weickert, “High accuracy optical flow estimation based on a theory for warping,” in European Conference on Computer Vision (ECCV). Springer, 2004, pp. 25–36.

[32] Isarlab @ unipg website. [Online]. Available: http://isar.unipg.it/