The Visual Computer (2023) 39:785–798
https://doi.org/10.1007/s00371-021-02345-6

ORIGINAL ARTICLE

Unsupervised deep learning based ego motion estimation with a downward facing camera

Maximilian Gilles[1] · **Sascha Ibrahimpasic**[2]

Accepted: 24 October 2021 / Published online: 27 November 2021

Abstract
Knowing the robot's pose is a crucial prerequisite for mobile robot tasks such as collision avoidance or autonomous navigation. Using powerful predictive models to estimate transformations for visual odometry via downward facing cameras is an understudied area of research. This work proposes a novel approach based on deep learning for estimating ego motion with a downward looking camera. The network can be trained completely unsupervised and is not restricted to a specific motion model. We propose two neural network architectures based on the Early Fusion and Slow Fusion design principle: "EarlyBird" and "SlowBird". Both networks share a Spatial Transformer layer for image warping and are trained with a modified structural similarity index (SSIM) loss function. Experiments carried out in simulation and for a real world differential drive robot show similar and partially better results of our proposed deep learning based approaches compared to a state-of-the-art method based on fast Fourier transformation.

Keywords Visual odometry · Spatial transformer layer · Unsupervised learning · Ego motion estimation · Downward facing camera

1 Introduction

Many crucial robot tasks such as obstacle detection, mapping or autonomous navigation require accurate pose estimates. Apart from many other sensor types such as laser scanners, GPS and inertial measurement units, optical camera systems have been proven successful for ego motion estimation. They are comparatively cheap and due to their high information density not only restricted to localization tasks.

This paper deals with estimating a mobile robot's ego motion driving on a planar surface, as it is the case in many industrial applications. The image stream of a downward looking camera is used to compute the relative transformation between consecutive camera frames (visual odometry). Knowing the initial pose at the beginning, the robot's path can be fully recovered by concatenating all relative transformations. For an overview over current state-of-the-art visual odometry (VO) systems, we refer to the work of [1]. A more fundamental introduction to the subject can be found in [2]. Generally speaking, images of a planar scene, e.g. a ground plane, are related by a homography transformation. Constraining the motion of the camera, e.g. forcing it downward with constant height above the ground, eliminates degrees of freedom (DoF) in the homography transformation and yields an Euclidean transformation which is defined up to a scale factor by 3 DoF.

We propose a novel, deep learning based, approach to infer the transformation parameters (θ, t_x, t_y) directly as outputs from a convolutional neural network. The two proposed network architectures, in the following referred to as EarlyBird and SlowBird, are inspired by the work of [3] and do not require any ground truth labelling as they are trained in an unsupervised manner. Unlike the proposed homography estimator in [3], we exploit the fact that in our simpler Euclidean case, there is no ambiguity between translational and [2] rotational parameters. This allows us to estimate the parameters of the Euclidean transformation matrix directly from the regression network, no detour via point correspondences and Direct Linear Transform algorithm [3, 4] is needed.

In order to improve our estimation accuracy and obtain a smoother output, we adapt the Slow Fusion design principle presented in [5]. Considering ego motion estimation

✉ Maximilian Gilles
maximilian.gilles@kit.edu

[1] Institute for Material Handling and Logistics, Karlsruher Institute of Technology, KIT, Karlsruhe, Germany

[2] Things Alive Robotics GmbH, Karlsruhe, Germany

with 6 DoF, [5] has shown the potential in learning temporal information over multiple image frames instead of relying solely on consecutive image pairs. However, their networks are trained in a supervised manner. We present an unsupervised version of their Slow Fusion architecture modified to retrieve the relative transformation between every consecutive image frame pair and will investigate if our approach benefits of the spatio-temporal feature extraction as well.

The contribution of this work can be summarized as followed: We present two neural networks architectures called EarlyBird and SlowBird for the general planar incremental ego-motion estimation with a downward facing camera. In contrast to other related work in the field, we do not assume any kind of steering model. Our proposed method can be applied to any vehicle type, e.g. ackermann-steering, differential-drive or omnidirectional. We demonstrate that by constraining the camera motion to face the ground, it is possible to train the networks in an unsupervised manner with unlabeled synthetic and real world data. Both network architectures are evaluated on synthetic data and on a real-world commercial robot system.

2 Related work

Estimating a mobile ground robot's ego motion via a downward facing camera is a rich area of research. Prior work has demonstrated the value of exploiting constrained camera motion introduced simplifications of the homography estimation. The work in this field can be subdivided in systems relying purely on relative transformations between consecutive frames (odometry) and map-based localization approaches. Methods for absolute localization are not prone to drift but need a preceding map building. Closer examination of these methods is out of the scope of this work, and for the interested reader, we refer to the extensive related work section of [6].

Saurer et al. [7] propose different minimal solutions based on point correspondences for the relative pose estimation between two camera frames depending on the prior knowledge about the 3D scene. Qifa and Kanade [8] virtually rotate the camera downward and estimate the ego motion of the camera by minimizing the sum of squared differences of pixel intensities with respect to motion parameters. Kitt et al. [9] use homographies of a virtually downward facing camera to recover the scale of the motion in their visual odometry pipeline. Yu et al. [10] propose a rotated template matching algorithm to compute the translation and rotation between consecutive images.

Other works constrain the problem to car-like vehicles [11–13]. By assuming the Ackermann steering model, the general planar motion can be restricted to 2DoF. Gao et al. [11] propose a keypoint-less method to recover the ego motion of car-like steering models using an optimal image registration technique based on global branch-and-bound optimization. Peng et al. [12] use an event-based camera system to overcome the downsides of a frame-based camera such as latency, motion blur and low dynamic range. In [13], template matching is applied for calculating the image displacement between consecutive frames. While the assumption of an Ackermann-like steering model greatly simplifies the registration problem, it is at the expense of generalization and not suitable for our presented use case.

In [14–16], the image registration problem is transformed into the frequency domain. Ri and Fujimoto [14] use phase-only correlation based on fast Fourier transforms (FFT) to estimate the Euclidean transformation parameters between image pairs. Goecke et al. [15] use the Fourier-Mellin transform to solve for the rotation and translation parameters simultaneously. In order to speed up the estimation process, [16] tackles the ego motion estimation problem by only solving for translational shift between image pairs. They apply FFT to multiple regions on each frame and recover the rotation from the shift between corresponding regions of consecutive image frames.

Downward facing cameras also play an important role in the field of micro aerial vehicles [17, 18]. In [17], an optical flow based approach based on a sparse feature set is applied for estimating the two dimensional pose of the micro aerial vehicles. The height above the ground is measured with an ultrasound sensor. Fu et al. [18] imply a direct approach for estimating the homography between consecutive image frames. An inertial measurement unit and a single beam laser rangefinder are used to make the ego motion estimation of the micro aerial vehicle more salient.

3 Problem formulation

Images taken from different viewpoints ${}^0\boldsymbol{C}$ and ${}^1\boldsymbol{C}$ of the same plane are related by a (planar) homography $\boldsymbol{H}$ (Fig. 1, left). The warping between corresponding points ${}^0\boldsymbol{x}$ and ${}^1\boldsymbol{x}$ is defined as:

$$ {}^0\boldsymbol{x} = \boldsymbol{H}\,{}^1\boldsymbol{x} \tag{1} $$

Knowing the rotation ${}^1_0\boldsymbol{R}$ and translation ${}^1_0\boldsymbol{t}$ from camera frame ${}^1\boldsymbol{C}$ to ${}^0\boldsymbol{C}$, the homography matrix $\boldsymbol{H}$ can be expressed as

$$ \boldsymbol{H} = {}^1_0\boldsymbol{R} - \frac{1}{d}{}^1_0\boldsymbol{t}\boldsymbol{n}^T \tag{2} $$

where d is the distance between ${}^1\boldsymbol{C}$ and the observed plane described by the normal vector $\boldsymbol{n}$ [7]. For the special case

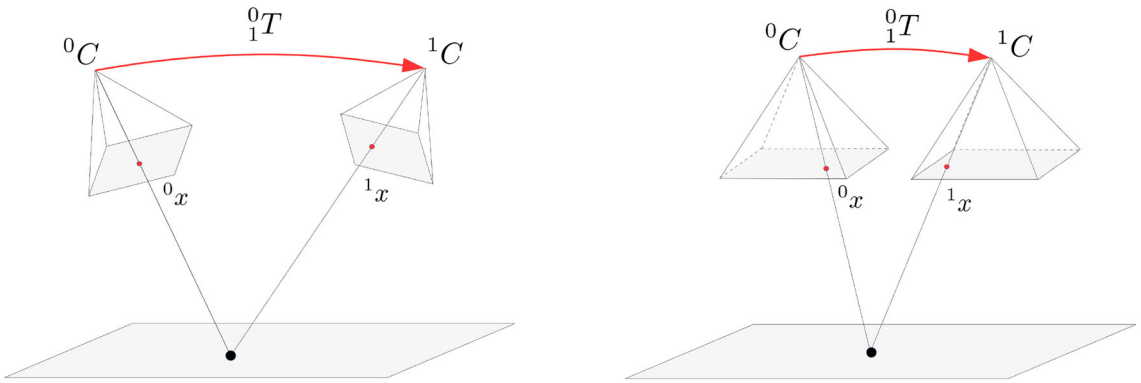

Fig. 1 Two images of a planar scene taken at different viewpoints ${}^0C$ and ${}^1C$ are related by a homography matrix. The left scene describes the general case, whereas the right scene illustrates the downward looking case. Introducing motion constraints simplifies the homography estimation

of a downward looking camera (Fig. 1, right) with constant height d above the ground, Eq. 2 can be expressed as:

$$\begin{aligned} \boldsymbol{H} &= {}^1_0\boldsymbol{R}_z - \begin{bmatrix} \frac{t_x}{d} \\ \frac{t_y}{d} \\ 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}^T \\ &= \begin{bmatrix} \cos\theta & -\sin\theta & t'_x \\ \sin\theta & \cos\theta & t'_y \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3}$$

Equation 3 shows that in this particular case, the image warping from frame ${}^1\boldsymbol{C}$ to ${}^0\boldsymbol{C}$ is described by an Euclidean transformation and there is no ambiguity between rotational and translational parameters [8]. The rotation ${}^0_1\boldsymbol{R}$ and translation ${}^0_1\boldsymbol{t}$ can be retrieved by estimating the warping matrix $\boldsymbol{H}$ between those images and knowing the scale.

The complete robot's path can be recovered by concatenating all relative transformations ${}^{k-1}_k\boldsymbol{T}$ for $k = 1 \ldots n$ with ${}^w_0\boldsymbol{C}$ being the initial camera pose at the start [2]:

$${}^w_n\boldsymbol{C} = {}^w_0\boldsymbol{C} \cdot {}^0_1\boldsymbol{T} \cdot \ldots \cdot {}^{n-2}_{n-1}\boldsymbol{T} \cdot {}^{n-1}_n\boldsymbol{T} \tag{4}$$

4 Neural network architecture

The proposed networks EarlyBird and SlowBird are both trained in the same way, only their respective regression network's input and architecture differ (see Fig. 3). In both cases, consecutive image frames with spatial dimensions of 200×200 pixels are fed into the regression network and the estimated parameters (θ, t_x, t_y) are relative to the previous image $\boldsymbol{I}_{n-1}$. They describe the transformation from image $\boldsymbol{I}_{n-1}$ taken at a previous point in time $t = n - 1$ to the current image in time $\boldsymbol{I}_n$.

In order to train the regression network without any provided ground truth, the error signal is computed by warping an image $\boldsymbol{I}_{n-1}$ taken at a previous point in time $t = n - 1$ accordingly to the network's estimation for (θ, t_x, t_y) and by analyzing how similar the warped image $\boldsymbol{I}_{n-1}(\theta, t_x, t_y)$ and the target image $\boldsymbol{I}_n$ are (see Fig. 2). The warping is done via a layer called Spatial Transformer (STL) [19]. It consists of a sampling grid generator and a bilinear sampler. Both entities are differentiable, so that the regression network can be trained in an end-to-end manner.

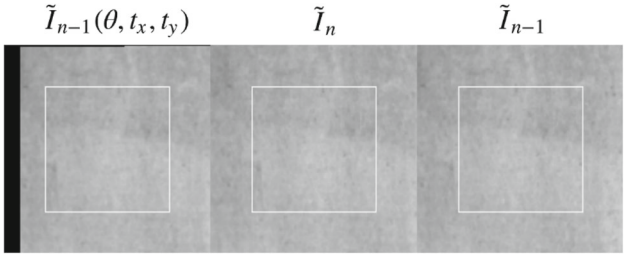

Fig. 2 The unsupervised training of the networks is done via comparing on how similar the warped image $\boldsymbol{I}_{n-1}(\theta, t_x, t_y)$ and the target image $\boldsymbol{I}_n$ are. In order to remove the black areas introduced by the warping procedure, both images are centrally cropped $\tilde{\boldsymbol{I}}_{n-1}(\theta, t_x, t_y)$ and $\tilde{\boldsymbol{I}}_n$. The white box indicates the central crop with $\alpha_c = 0.6$

Our loss function is based on the structural similarity index (SSIM) [20]. The SSIM function takes the warped and the target image as inputs and returns a value between $\in (-1, 1)$. The return value is close to 1 if both images are similar. We modify it by adding a constant of value one and inverting the sign of the SSIM. As a result, our proposed loss function returns a value between $\in (0, 2)$. Since the robot is moving, consecutive image frames $\boldsymbol{I}_{n-1}$ and $\boldsymbol{I}_n$ generally do not contain the same real world scene. Depending on the camera's focus length, the distance to the floor, the frame rate, and the robot's velocity successive frames overlap more or less and black borders are introduced in the warped image. To compensate these artefacts, the warped image $\boldsymbol{I}_{n-1}(\theta, t_x, t_y)$ and the target image $\boldsymbol{I}_n$ are centrally cropped by a factor $\alpha_c \in (0, 1)$. The hyperparameter α_c has to be chosen so that the

cropped image patches $\tilde{\boldsymbol{I}}_{n-1}(\theta, t_x, t_y)$ and $\tilde{\boldsymbol{I}}_n$ theoretically show the same real world scenery.

The resulting loss function is defined as:

$$L = 1 - \text{SSIM}\left(\tilde{\boldsymbol{I}}_{n-1}(\theta, t_x, t_y), \tilde{\boldsymbol{I}}_n\right) \tag{5}$$

In the work of [3], the $l1$ loss function is used; however, in the course of our work, we experienced the SSIM loss function to be easier to train and encourage to experiment with both.

In the following, the two proposed regression network architectures for EarlyBird and SlowBird will be further described, as well as the subsequent warping procedure in the STL.

4.1 EarlyBird

The EarlyBird network (see Fig. 3) takes as input two consecutive grey level images $\boldsymbol{I}_{n-1}$ and $\boldsymbol{I}_n$. They are stacked channel wise (200, 200, 2) and forwarded into the net following the Early Fusion design principle of [21]. The net layout is inspired by the VGGNet architecture of [3]. It has five convolutional blocks, each one consisting of two 2D convolutional layer and a max pooling layer. The filter sizes of the convolutional layers become greater going deeper into the network: 8, 16, 32, 64, and 64. The quadratic kernel size remains the same with (3, 3). The pooling kernel remains (2, 2) throughout the net. The second last fully connected layer has 512 neurons. The final layer consists of three neurons, one for each transformation parameter. Between those two layers there is dropout.

4.2 SlowBird

The SlowBird network (see Fig. 3) has as input five consecutive grey level images $\boldsymbol{I}_{n-4} \ldots \boldsymbol{I}_n$ and follows the Slow Fusion design principle of [5, 21]. Instead of being stacked up channel wise (EarlyBird) and fed as a whole into the net, they are ordered in a 4-dimensional way (200, 200, 5, 1) and convoluted over the spatial dimensions (1st and 2nd dimension) as well as the time (3rd dimension). In order to deal with the four-dimensional shape of the input, 3D convolutional layers [22] are used with the following kernel sizes: (3, 3, 3), (3, 3, 2), (3, 3, 1), and (3, 3, 1). Since the convolutional operation is not only performed over the spatial dimension of the feature map as it is the case with 2D convolutions, the temporal information of the image frames is preserved in its output signals. With every 3D convolutional operation, the time dimension vanishes until it finally disappears in the third 3D convolution. The following 3D convolutional operations can be seen as standard 2D convolutions. After every second convolutional operation, max pooling is applied. This downsampling step only affects the spatial dimension of the feature map, the time dimension is not involved. Just as in the EarlyBird network, high-level reasoning is done via two fully connected layers. They have the same structure and number of neurons as the fully connected neural network in the EarlyBird approach.

4.3 Spatial transformer network

The STL introduced in [19] performs an inverse affine warping of the pixel coordinates of image $\boldsymbol{I}_{n-1}$. It is differentiable and allows backpropagation of the error gradient through the STL into the regression network. First, the regression network's output (θ, t_x, t_y) is converted into an affine transformation matrix A:

$$\boldsymbol{A} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{21} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \end{bmatrix} \tag{6}$$

The following inverse warping of the STL can be subdivided into two steps: 1. generation of the sampling grid and 2. bilinear sampling. The sampling grid generator applies an inverse warping of the regular and normalized target grid $\boldsymbol{G}$. For every target pixel $\boldsymbol{G}_i = \left(x_i^t, y_i^t\right)^T$ of our target grid $\boldsymbol{G} = \{\boldsymbol{G}_i\}$ a source coordinate $\boldsymbol{S}_i = \left(x_i^s, y_i^s\right)^T$ is defined.

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \boldsymbol{A} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{7}$$

To perform the warping of the input image, a differentiable sampler takes the set of source coordinates $\boldsymbol{S} = \{\boldsymbol{S}_i\}$, along with the input image $\boldsymbol{I}_{n-1}$ and produces the sampled output image $\boldsymbol{I}_{n-1}(\theta, t_x, t_y)$. Since the source coordinates $\boldsymbol{S}_i$ are in general not aligned with the grid-like structure of the input image $\boldsymbol{I}_{n-1}$, the STL makes use of a differentiable bilinear interpolation kernel. For every source pixel $\boldsymbol{S}_i = \left(x_i^s, y_i^s\right)^T$, the interpolation is described via

$$V_i = \sum_n^H \sum_m^W I_{n-1}(n, m) \max\left(0, 1 - \left|x_i^s - m\right|\right) \max\left(0, 1 - \left|y_i^s - n\right|\right) \tag{8}$$

where V_i is the sampled pixel value belonging to the target pixel $\boldsymbol{G}_i$ and $I_{n-1}(n, m)$ stands for the pixel value of image $\boldsymbol{I}_{n-1}$ at position (n,m). Height and width normalised coordinates are used, such that $-1 \leq x_i^t, y_i^t, x_i^s, y_i^s \leq 1$, when within the spatial bounds of the input or output image.

5 System setup

Images are recorded with a frame rate of 90 fps. The camera is mounted in front of a differential drive mobile robot

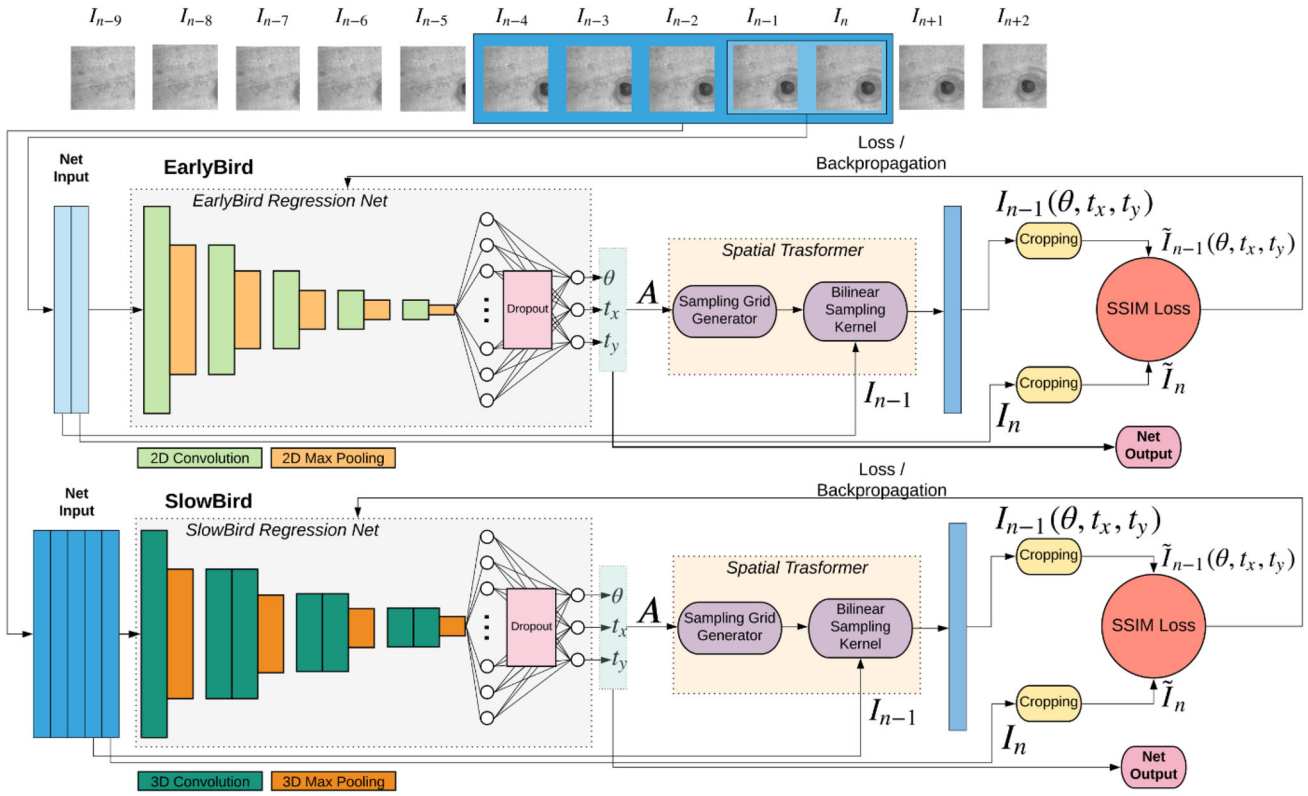

Fig. 3 Overview over proposed neural network architectures: EarlyBird (top) and SlowBird (bottom). Except for the input and the regression part both networks share the same structure. The training is done via warping the previous image in time accordingly to the estimated transformation parameters $\boldsymbol{I}_{n-1}(\theta, t_x, t_y)$ and comparing it against the current image in time $\boldsymbol{I}_n$

and facing downward with an approximately distance to the ground of 0.11 m. To eliminate shadows when driving around and to realize higher shutter speeds, two LED spotlights are illuminating the ground. For later use cases, it would be also possible to mount the camera underneath the robot. In order to accomplish an exact bird's eye vision system, each camera image is warped accordingly to a homography estimated in advance with a calibration chessboard pattern laying on the ground (see Fig. 4). This warping procedure allows us to set the scale arbitrary. It is however to be considered to choose the desired resolution accordingly to the original camera image to minimize interpolation error. In our case, we set the resolution to 8 $\frac{\text{px}}{\text{mm}}$.

The average inference times are 18.31 ms ($\sim$ 54 fps) for the SlowBird network and 19.51 ms ($\sim$ 51 fps) for the EarlyBird network and 14.75 ms ($\sim$ 67 fps) for the FFT method (GeForce GTX 1080 Ti, i7-8700 @3.20 GHz).

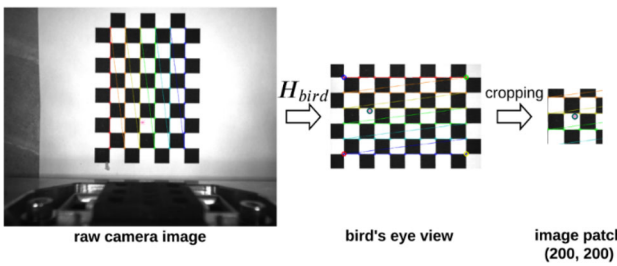

Fig. 4 In order to compensate for mounting inaccuracies of the camera the raw image is warped by a precalibrated homography $\boldsymbol{H}_{bird}$. The warped image is cropped to 200×200 pixels and fed into the network

6 Training

The two networks EarlyBird and SlowBird are pretrained on synthetic datasets and then fine-tuned for the surface texture of the real world testing area.

6.1 Synthetic dataset

The synthetic dataset is generated by laying hand designed trajectories based on Bézier curves upon multiple high resolution images acting as different ground surfaces. A virtual robot is travelling along these trajectories and taking snapshots of the ground images. Its velocity is randomized in order to cover a broader range of relative transformations between consecutive image frames. Though ground truth is available for our synthetic dataset, we do not use it for training. In total, the synthetic dataset consists of 50 trajectories and 25 different ground images resulting in one million training samples.

6.2 Real world dataset

Transfer learning [23] is used to transfer the knowledge gained with the synthetic dataset to the real world domain. Our experience has shown that without pretraining on the synthetic dataset finding the proper hyperparameters to minimize the loss on the real world dataset might become very exhaustive in some cases. Although it is still possible without, we recommend to begin training on the synthetically pretrained models. The real world dataset consists of 730 k training samples. Since no labelling is required, it can be obtained in under 2.5 h of driving around with the robot in our testing area (max. translational velocity of 0.1 $\frac{\mathrm{m}}{\mathrm{s}}$ and max. angular velocity of 10 $\frac{\mathrm{deg}}{\mathrm{s}}$). The networks are trained with data augmentation on a single consumer grade GPU (GTX 1080 Ti) for about one day (including the synthetic dataset). Data augmentation is done by randomly skipping image frames and by feeding them backwards into the net. This simulates different inter frames velocities and generates more reverse training samples. Adam [24] was used for optimization. Our experiments showed that beneath setting the learning rate to $\alpha = 0.0001$, modifying the value of the numerical stability constant $\hat{\varepsilon} = 0.0001$ (default value $\hat{\varepsilon} = 1 \times 10^{-8}$) yielded to good results. To avoid overfitting, the network is evaluated on a separate validation dataset after every 3000 training iterations and the model with the best performance (minimal loss value) is kept.

7 Results and discussion

We evaluated our proposed neural network architectures by comparing them against a state-of-the-art image registration technique[1] based on FFT [14]. Although it is already well known that feature-based methods have difficulties with the fast moving, texture less images of a downward facing camera, we added an ORB feature based approach to our evaluation as a conservative baseline. Experiments are conducted on synthetic and on real world data.

7.1 Synthetic data experiments

The synthetic test dataset consists of eight unseen trajectories on nine different surfaces each. As proposed in [25], we used the root mean square relative pose error (RMS-RPE) and root mean square absolute trajectory error (RMS-ATE) as evaluation metrics. As we are evaluating a visual odometry system based on incremental motion estimation between consecutive frames, we choose for the time parameter $\Delta = 1$. The RMS-RPE accounts for the drift per frame and can be divided in translational (trans) and rotational (rot) components, whereas the RMS-ATE relates to the global consistency of the predicted and the ground truth trajectory. For an in-depth discussion of the used metrics, please refer to [25].

Evaluation on the whole synthetic data (Table 1) shows that EarlyBird has the lowest translational (RMS-RPE trans) as well as rotational relative pose error (RMS-RPE rot), followed by FFT and SlowBird. In terms of ATE SlowBird performed best, followed by EarlyBird and FFT.

In "Fig. 8 in Appendix", the RPE was put in relation to the amount of overlap between image pairs by sorting and dividing the error into three groups of same size and computing the RMS-RPE separately for each tertile. The results show that, independent of the chosen method, the amount of overlap between consecutive image frames has a strong impact on the accuracy of the registration problem and is not distributed equally over all three performance tertiles. This illustrates a short-coming for the usage in incremental motion estimation methods where consistent performance is desired.

An important requirement for a visual odometry system is its functionality on a wide variety of different floor surfaces. Therefore, we analyzed the statistical distribution of RMS-RPE and RMS-ATE according to the different surface images of our test dataset (see Table 2). It can be seen that FFT has the smallest standard deviation (STD) for the translational component of FFT while still performing worse than EarlyBird which has the lowest STD for RMS rot and ATE. An extensive overview over all trajectories can be found in "Table 7 in the Appendix". Looking at the ATE value for the ORB method, it becomes clear that its performance is not very consistent over different surfaces. For feature rich, highly textured surfaces it performs well, but fails when encountering highly illuminated, featureless images (see "Fig. 7 in Appendix" for an overview over the chosen test surface images).

In order to test the sensitivity of our models regarding illumination changes or noise in between consecutive image frames, we augmented each image in the test dataset by a random pixel intensity value of $\beta \in (-10, 10)$ or added Gaussian distributed white noise ($\mu = 0$) for each pixel with

[1] https://github.com/YoshiRi/ImRegPOC under 2-clause BSD License.

Table 1 RPE and ATE for given models evaluated on the synthetic test dataset for different models

	RMS-RPE				RMS-ATE	
	trans [px]	+%[a]	rot [rad]	+%[a]	trans [px]	+%[a]
SlowBird	4.89E−01	6.1	2.22E−03	76.1	1.12E+02	–
EarlyBird	4.61E−01	–	1.26E−03	–	1.13E+02	1.3
FFT	4.83E−01	4.8	1.91E−03	51.0	2.11E+02	88.1
ORB	2.56E+01	5443.7	9.81E−03	677.4	3.03E+02	170.4

[a]Percentage increase w.r.t. to best model

Table 2 Statistical error distribution for RMS-RPE and RMS-ATE divided according to different surface images

	RMS-RPE trans [px]			RMS-RPE rot [rad]			RMS-ATE [px]		
	Mean	RMSE	STD	Mean	RMSE	STD	Mean	RMSE	STD
SlowBird	4.73E−01	4.73E−01	1.29E−02	2.09E−03	2.12E−03	3.92E−04	8.28E+01	8.67E+01	2.73E+01
EarlyBird	4.61E−01	4.61E−01	3.83E−03	1.26E−03	1.26E−03	1.13E−04	1.13E+02	1.13E+02	5.35E+00
FFT	4.83E−01	4.83E−01	6.74E−04	1.90E−03	1.91E−03	1.75E−04	2.10E+02	2.11E+02	1.75E+01
ORB	1.25E+01	2.56E+01	2.37E+01	7.25E−03	9.81E−03	7.01E−03	1.77E+02	3.03E+02	2.61E+02

Table 3 RMS-RPE and RMS-ATE evaluated on test dataset with *augmented illumination changes*

	RMS-RPE				RMS-ATE [px]	
	trans [px]	+%[a]	rot [rad]	+%[a]	trans [px]	+%[a]
SlowBird	9.28E−01	89.9	6.12E−03	175.3	2.32E+02	107.1
EarlyBird	4.90E−01	6.2	1.42E−03	12.8	1.33E+02	16.9
FFT	4.83E−01	0.0	1.90E−03	− 0.1	2.22E+02	5.4
ORB	2.56E+01	0.0	9.71E−03	− 1.0	2.99E+02	− 1.2

[a]Percentage increase w.r.t. to evaluation on non-augmented images

$\sigma^2 = 4$. From Table 3, it can be seen that FFT and EarlyBird performed best when encountering changing illumination in the video stream. On the dataset with added noise, SlowBird had the lowest RPE and ATE, though having an increased error of 54.6%, 207.4% and 111.8% with regard to the original dataset (see Table 4).

7.2 Real world experiments

We recorded 19 test trajectories with a differential drive robot going straight, taking turns, moving backward, driving arbitrary loops or circles (see "Table 8 in Appendix" for more details about the chosen trajectories). The recording of the test dataset has been completely independent from our train and evaluation dataset drive. The ground truth trajectory is recorded with a Valve Index VR kit. In a predefined testing area, the controller is globally tracked and therefore not vulnerable to drift. This makes it to a well suited ground truth sensor for odometry systems. For our real world experiments, we are comparing the translational and rotational ATE at the end of each drive ATE(t_{end}). See Fig. 5 for a visualization over a subset of the test trajectories.

The results on the real world test data (see Table 5) show that SlowBird outperforms FFT, EarlyBird and ORB when considering the ATE(t_{end}) over all test trajectories. The fact that RMSE is very sensitive to outliers suggests that SlowBird is the most consistent method for this dataset. Looking at the high error for the ORB based approach, the results once again

Table 4 RMS-RPE and RMS-ATE evaluated on test dataset with *added Gaussian distributed noise*

	RMS-RPE				RMS-ATE [px]	
	trans [px]	+%[a]	rot [rad]	+%[a]	trans [px]	+%[a]
SlowBird	7.56E−01	54.6	6.83E−03	207.4	2.37E+02	111.8
EarlyBird	1.03E+00	122.7	6.84E−03	442.4	2.95E+02	159.6
FFT	1.90E+00	293.0	1.14E−02	498.1	3.42E+02	62.6
ORB	8.69E+04	339,905.6	4.18E−02	326.5	2.20E+06	726,710.1

[a]Percentage increase w.r.t. to evaluation on non-augmented images

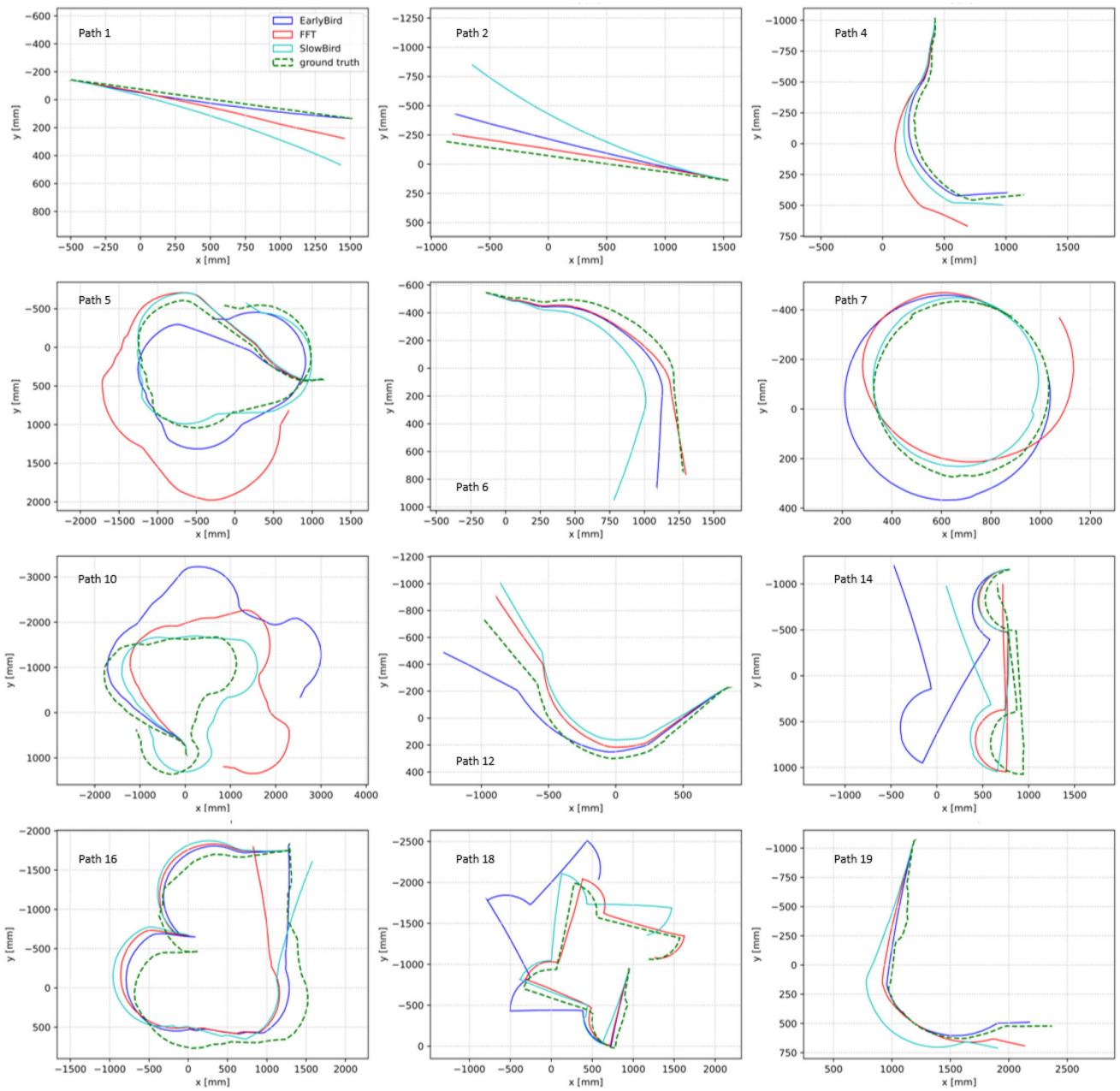

Fig. 5 Visualization over a subset (12 out of 19 trajectories are shown) of all test trajectories for the real world robot. As it is common with all incremental motion estimation methods, rotational errors generally grow unbounded in lateral position. That is why these methods have such a high demand on rotational accuracy

confirm the general consent that feature-based methods are not suitable for this use-case.

Comparing both network architectures, it can be said that our proposed model based on the Early Fusion design principle had the smallest RPE on the synthetic dataset, and our model based on the Slow Fusion design principle performed best considering the absolute trajectory error on synthetic as well as real world data (see Tables 2, 5). Looking at the paths in Fig. 5 and the results in Table 9, it can be observed that FFT is able to predict the correct trajectory very well on a subset of trajectories but does not perform as consistently as SlowBird (higher RMSE) over all trajectories.

Although our proposed networks EarlyBird and SlowBird had the best performance on the synthetic and real world dataset and there is a positive correlation between loss function error and relative pose error ($\rho_{\text{RPEtrans}-\text{SSIM}} = 0.33$ and $\rho_{\text{RPErot}-\text{SSIM}} = 0.42$ evaluated on the synthetic test dataset), it can be observed that very small orientation esti-

Table 5 The translational and rotational ATE over *all* test trajectories

	ATE-trans(t_{end}) [mm]			ATE-rot(t_{end}) [rad]		
	Mean	RMSE	STD	Mean	RMSE	STD
SlowBird	3.91E+02	4.66E+02	2.24E+02	1.42E+00	1.88E+00	1.17E+00
EarlyBird	5.80E+02	1.08E+03	9.14E+02	1.41E+00	1.92E+00	1.25E+00
FFT	4.04E+02	6.83E+02	5.50E+02	1.41E+00	1.97E+00	1.32E+00
ORB	3.21E+07	7.66E+07	6.97E+07	1.28E+00	1.84E+00	1.02E+00

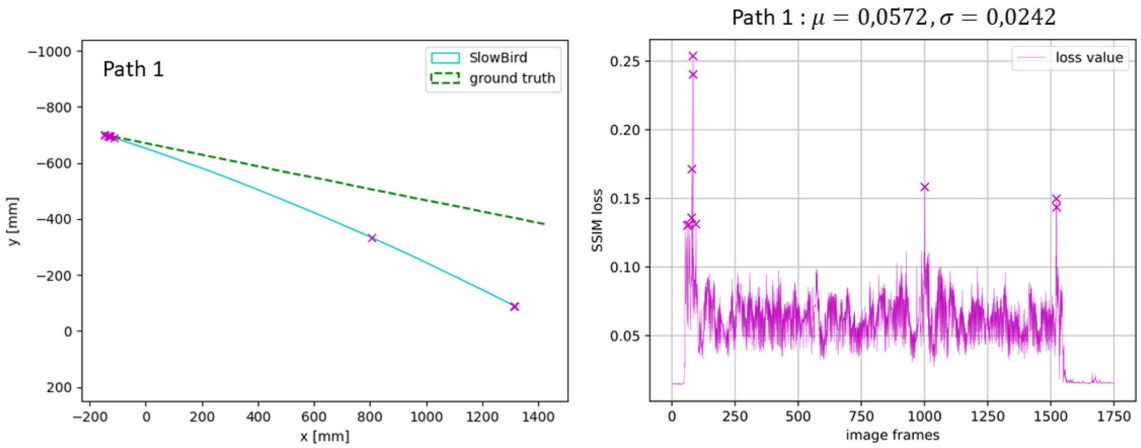

Fig. 6 Estimated SlowBird trajectory for Path 1 (left) and corresponding loss values (right). The crosses illustrate loss outliers of the 3σ-intervall. Drift does not necessarily correspond to outliers in the loss signal (e.g. outlier at $x = 800$ mm in left image)

Table 6 The correlation between RPE and SSIM loss value divided into three error tertiles evaluated on our synthetic dataset

	Mean value		Pearson correlation coefficient ρ	
	RPE—trans [px]	RPE—rot [rad]	RPE trans—SSIM	RPE rot—SSIM
1st tertile	0.12	0.0003	0.1715	0.0032
2nd tertile	0.32	0.0010	0.1812	0.0755
3rd tertile	0.72	0.0029	0.1242	0.4786
All	0.39	0.0014	0.3270	0.4165

mation errors, as they appear in Fig. 6 when driving straight, seem to pass unnoticed by the signal noise (Fig. 6, right) of our proposed SSIM loss function. This may be explained by the weak correlation between rotation error and SSIM value for the first and second tertile. Therefore, for very small rotational errors (mean RPE rot of 1st tertile is 0.0003 rad), it can be assumed that the error cannot be minimized by minimizing the objective function (see Table 6). Since VO systems are in general prone to drift, this illustrates a shortcoming of our currently used loss function.

8 Conclusion

In this work a novel, deep learning based approach to estimate the ego motion of a camera facing the ground is presented. In contrast to many other work, it is independent of the chosen drive type as long as the camera is moving parallel with constant distance to the ground plane. Both proposed network architectures achieved similar and partially better results than the performance of a state-of-the-art FFT image registration method evaluated on synthetic and real world scenarios for a differential drive robot. Especially our proposed SlowBird approach represents a promising alternative to existing methods regarding its performance consistency on both datasets.

Though, in this work, the network weights were fine-tuned for a wooden floor, results on synthetic data demonstrate that both networks can generalize to different floor materials, sudden illumination changes and image noise. With the proposed unsupervised learning strategy, the model weights

can be easily and cheaply adapted to alternative ground surfaces and even drive types, no expensive labels are required. We are aware that the currently used SSIM loss function does have its limitations when it comes to rotational accuracy for small errors. Future work should concentrate on alternative loss functions or additional sensors such as IMUs and compasses to compensate for that. Also for future work, it would be interesting to investigate how an online learning approach would be applicable since no ground truth data is required for training.

Despite the aforementioned limitations, our experiments have shown that a deep learning based image registration method, trained completely unsupervised, is over a long series of images accurate enough to become a viable alternative to existing methods. We are aware that our solution is not yet practically and economically feasible, but with current progress in Edge AI, estimating ego motion unsupervised with downward facing cameras might become an interesting use case.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Appendix

See Figs. 7 and 8, Tables 7, 8 and 9.

Fig. 7 Nine different surfaces were chosen for the synthetic test dataset

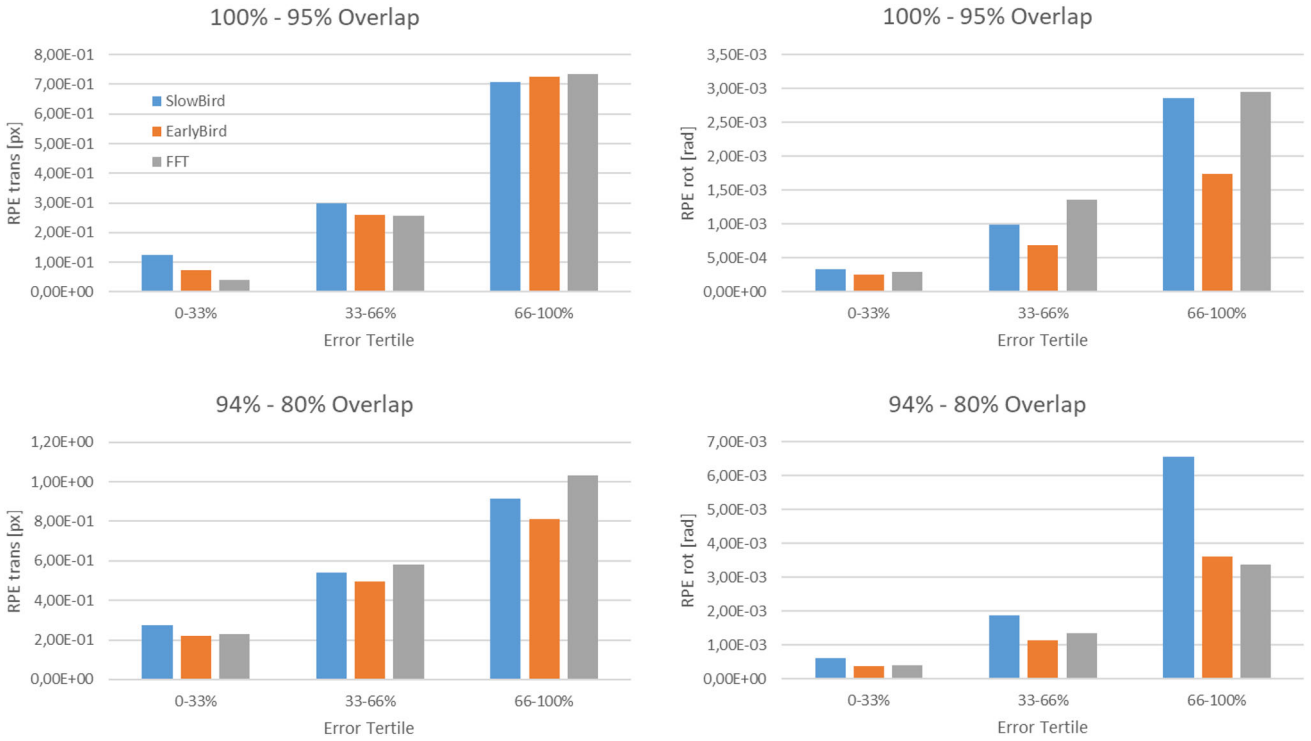

Fig. 8 RPE trans and rot sorted in three groups of same size and ascending order. The amount of overlap between image pairs has a strong influence of the prediction accuracy of the network. The feature based approach is not displayed since its error values are higher by order of magnitude 10 and would distort the scale of the graph

Table 7 RPE and ATE summarized for all trajectories and evaluated separately for each image simulating different surfaces

	Surface 1	Surface 2	Surface 3	Surface 4	Surface 5	Surface 6	Surface 7	Surface 8	Surface 9
RMSE-RPE—trans [px]									
SlowBird	4.78E−01	4.58E−01	4.69E−01	4.66E−01	4.66E−01	4.69E−01	4.95E−01	4.94E−01	4.65E−01
EarlyBird	4.65E−01	4.58E−01	4.56E−01	4.59E−01	4.58E−01	4.60E−01	4.66E−01	4.66E−01	4.59E−01
FFT	4.83E−01	4.83E−01	4.84E−01	4.83E−01	4.82E−01	4.84E−01	4.84E−01	4.84E−01	4.82E−01
ORB	9.40E−01	8.19E−01	4.15E+01	9.14E−01	8.74E−01	9.62E−01	6.44E+01	9.21E−01	9.57E−01
RMSE-RPE—rot [rad]									
SlowBird	2.16E−03	1.49E−03	2.16E−03	1.80E−03	1.86E−03	2.38E−03	2.75E−03	2.42E−03	1.79E−03
EarlyBird	1.32E−03	1.15E−03	1.25E−03	1.21E−03	1.16E−03	1.39E−03	1.47E−03	1.20E−03	1.18E−03
FFT	1.72E−03	1.72E−03	1.97E−03	1.76E−03	1.83E−03	1.97E−03	2.23E−03	2.08E−03	1.81E−03
ORB	4.08E−03	3.43E−03	1.46E−02	3.85E−03	3.68E−03	4.10E−03	2.34E−02	4.00E−03	4.15E−03
RMSE-ATE [px]									
SlowBird	9.80E+01	7.81E+01	1.02E+02	5.59E+01	5.44E+01	7.35E+01	1.41E+02	7.38E+01	6.79E+01
EarlyBird	1.11E+02	1.08E+02	1.08E+02	1.19E+02	1.24E+02	1.14E+02	1.13E+02	1.14E+02	1.10E+02
FFT	1.90E+02	1.93E+02	2.20E+02	1.98E+02	2.03E+02	2.07E+02	2.43E+02	2.29E+02	2.06E+02
ORB	5.43E+01	3.19E+01	6.51E+02	4.27E+01	3.67E+01	5.61E+01	6.22E+02	4.13E+01	5.30E+01

Table 8 Overview over real world test trajectories

	Description[a]	# Images	Length (mm)	Ratio[b]
Path 1	ST–FW	2063	2055	1.00
Path 2	ST–BW	2374	2454	0.97
Path 3	LT–FW	2437	2162	1.13
Path 4	RT–BW	1893	2152	0.88
Path 5	LP–FW	6881	7983	0.86
Path 6	LT–BW	2200	2309	0.95
Path 7	CC–FW	3626	2362	1.54
Path 8	ST–FW	2143	2129	1.01
Path 9	SC–BW	3880	3837	1.01
Path 10	LP–FW	9224	11,024	0.84
Path 11	ST–FW	1754	1617	1.08
Path 12	RT–FW	2311	2534	0.91
Path 13	LT–BW	2206	2560	0.86
Path 14	BF–FW	6275	5025	1.25
Path 15	LP–FW	8124	9230	0.88
Path 16	LP–BW	7597	8941	0.85
Path 17	LP–FW	11,026	11,930	0.92
Path 18	SQ–FW	7357	6135	1.20
Path 19	LT–FW	2407	2927	0.82

[a]*ST* straight, *LT* left turn, *RT* right turn, *CC* circle, *SC* S-curve, *LP* arbitrary loop, *SQ* square, *FW* forward, *BW* backward
[b]Ratio, Ratio between number of images per path and length [1/mm]

Table 9 ATE(t_{end}) evaluated on all trajectories of real world dataset

	ATE trans(t_{end}) [mm]				ATE rot(t_{end}) [rad]			
	SlowBird	EarlyBird	FFT	ORB	SlowBird	EarlyBird	FFT	ORB
Path 1	3.42E+02	1.71E+01	1.57E+02	1.83E+05	2.23E−01	1.27E−01	5.01E−02	1.94E+00
Path 2	6.97E+02	2.53E+02	9.00E+01	4.27E+05	4.26E−01	6.66E−02	2.41E−02	3.11E+00
Path 3	2.20E+02	4.81E+01	2.23E+02	1.03E+04	3.13E+00	2.98E+00	2.90E+00	2.93E+00
Path 4	1.82E+02	1.36E+02	5.19E+02	1.03E+06	2.95E+00	2.81E+00	2.96E+00	3.42E−01
Path 5	3.04E+02	2.36E+02	1.61E+03	2.32E+08	9.15E−01	3.46E−01	2.44E+00	1.74E−01
Path 6	5.39E+02	2.17E+02	3.68E+01	3.13E+09	2.94E+00	2.63E+00	2.38E+00	2.86E+00
Path 7	6.50E+01	3.99E+02	3.19E+02	1.17E+08	8.48E−01	3.96E−01	4.72E−02	2.59E+00
Path 8	4.65E+02	1.13E+01	6.74E+01	8.58E+06	1.81E−01	2.41E−01	1.76E−01	1.51E+00
Path 9	9.96E+02	2.54E+02	1.63E+02	3.85E+07	5.20E−01	9.09E−02	1.62E−01	2.25E−01
Path 10	2.76E+02	3.69E+03	2.08E+03	2.08E+09	8.11E−01	3.13E+00	1.88E+00	2.62E+00
Path 11	3.14E+02	9.21E+01	1.02E+02	3.99E+03	1.77E−01	1.02E−01	1.42E−01	9.07E−01
Path 12	3.00E+02	3.85E+02	1.87E+02	1.78E+03	3.04E+00	2.64E+00	3.11E+00	2.71E+00
Path 13	1.88E+02	1.68E+02	2.57E+02	8.24E+06	1.27E+00	1.16E+00	1.33E+00	2.82E−01
Path 14	5.63E+02	1.14E+03	5.43E+01	4.31E+04	2.78E−01	3.34E−01	2.21E−02	3.11E+00
Path 15	4.81E+02	1.87E+03	9.27E+02	1.39E+07	2.34E+00	1.07E+00	1.41E+00	5.15E−01
Path 16	3.69E+02	1.83E+01	4.59E+02	6.70E+05	2.60E+00	3.01E+00	3.14E+00	8.85E−01
Path 17	7.23E+02	1.33E+03	4.72E+02	7.99E+07	2.29E+00	2.83E+00	3.01E+00	1.32E+00
Path 18	2.93E+02	1.15E+03	7.11E+01	3.92E+07	5.26E−01	1.58E+00	1.04E−01	2.40E+00
Path 19	4.98E+02	1.90E+02	2.86E+02	2.11E+08	2.95E+00	2.68E+00	2.94E+00	1.98E+00

Although ORB had competitive results regarding the rotational error at the end of each path, looking at the translational error one can see that the tracking was lost (exception path 11 and 12)

References

1. He, M., Zhu, C., Huang, Q., Ren, B., Liu, J.: A review of monocular visual odometry. Vis. Comput. (2020). https://doi.org/10.1007/s00371-019-01714-6
2. Scaramuzza, D., Fraundorfer, F.: Visual odometry [Tutorial]. IEEE Robot. Autom. Mag. (2011). https://doi.org/10.1109/MRA.2011.943233
3. Nguyen, T., Chen, S.W., Shivakumar, S.S., Taylor, C.J., Kumar, V.: Unsupervised deep homography: a fast and robust homography estimation model. IEEE Robot. Autom. Lett. (2018). https://doi.org/10.1109/LRA.2018.2809549
4. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision, 2nd edn. Cambridge Univ. Press, Cambridge (2015)
5. Weber, M., Rist, C., Zöllner, J.M.: Learning temporal features with CNNs for monocular visual ego motion estimation. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). pp. 1–6 (2017). https://doi.org/10.1109/ITSC.2017.8317922
6. Fabian Schmid, J., Simon, S.F., Mester, R.: Ground Texture Based Localization Using Compact Binary Descriptors. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 1315–1321 (2020). https://doi.org/10.1109/ICRA40945.2020.9197221
7. Saurer, O., Vasseur, P., Boutteau, R., Demonceaux, C., Pollefeys, M., Fraundorfer, F.: Homography based egomotion estimation with a common direction. IEEE Trans. Pattern Anal. Mach. Intell. (2017). https://doi.org/10.1109/TPAMI.2016.2545663
8. Ke, Q., Kanade, T.: Transforming camera geometry to a virtual downward-looking camera: robust ego-motion estimation and ground-layer detection. In: 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings. p. I (2003). https://doi.org/10.1109/CVPR.2003.1211380
9. Kitt, B., Rehder, J., Chambers, A., Schönbein, M., Lategahn, H., Singh, S.: Monocular Visual Odometry using a Planar Road Model to Solve Scale Ambiguity. In: Proceedings of the 5th European Conference on Mobile Robots (ECMR 2011), Örebro, Sweden, September 7–9, 2011. Ed.: A. J. Lilienthal, 43 (2011)
10. Yu, Y., Pradalier, C., Zong, G.: Appearance-based monocular visual odometry for ground vehicles. In: 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). pp. 862–867 (2011). https://doi.org/10.1109/AIM.2011.6027050
11. Gao, L., Su, J., Cui, J., Zeng, X., Peng, X., Kneip, L.: Efficient Globally-Optimal Correspondence-Less Visual Odometry for Planar Ground Vehicles. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 2696–2702 (2020). https://doi.org/10.1109/ICRA40945.2020.9196595
12. Peng, X., Wang, Y., Gao, L., Kneip, L.: Globally-optimal event camera motion estimation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) Computer Vision: ECCV 2020, Cham, 2020, pp. 51–67. Springer, Cham (2020)
13. Nourani-Vatani, N., Roberts, J., Srinivasan, M.V.: Practical visual odometry for car-like vehicles. In: 2009 IEEE International Conference on Robotics and Automation. pp. 3551–3557 (2009). https://doi.org/10.1109/ROBOT.2009.5152403
14. Ri, Y., Fujimoto, H.: Drift-free motion estimation from video images using phase correlation and linear optimization. In: 2018 IEEE 15th International Workshop on Advanced Motion Control (AMC). pp. 295–300 (2018). https://doi.org/10.1109/AMC.2019.8371106
15. Goecke, R., Asthana, A., Pettersson, N., Petersson, L.: Visual Vehicle Egomotion Estimation using the Fourier-Mellin Transform. In: 2007 IEEE Intelligent Vehicles Symposium. pp. 450–455 (2007). https://doi.org/10.1109/IVS.2007.4290156
16. Birem, M., Kleihorst, R., El-Ghouti, N.: Visual odometry based on the Fourier transform using a monocular ground-facing camera. J. Real-Time Image Proc. (2018). https://doi.org/10.1007/s11554-017-0706-3
17. More, V., Kumar, H., Kaingade, S., Gaidhani, P., Gupta, N.: Visual odometry using optic flow for Unmanned Aerial Vehicles. In: 2015 International Conference on Cognitive Computing and Information Processing (CCIP). pp. 1–6 (2015). https://doi.org/10.1109/CCIP.2015.7100731
18. Fu, B., Shankar, K.S., Michael, N.: RaD-VIO: rangefinder-aided downward visual-inertial odometry. In: 2019 International Conference on Robotics, pp. 1841–1847 (2019)
19. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial Transformer Networks. Adv. Neural Inf. Proc. Syst. **28** (2015)
20. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. Publ. IEEE Signal Process. Soc. (2004). https://doi.org/10.1109/TIP.2003.819861
21. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 2014, pp. 1725–1732. IEEE (2014). https://doi.org/10.1109/CVPR.2014.223
22. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 4489–4497 (2015). https://doi.org/10.1109/ICCV.2015.510
23. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A Comprehensive Survey on Transfer Learning. http://arxiv.org/abs/1911.02685 (2019)
24. Kingma, D.P., Ba, J.L. (eds.): Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings (2015)
25. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A Benchmark for the Evaluation of RGB-D SLAM Systems. In: Proc. of the International Conference on Intelligent Robot Systems (IROS) (2012)

Maximilian Gilles is research assistant of the department for Robotics and Interactive Systems at the Institute for Material Handling and Logistics (IFL) at Karlsruhe Institute of Technology (KIT). His main research focuses on computer vision and pattern recognition for mobile or stationary autonomous robots in the field of intralogistics.

Sascha Ibrahimpasic is working at Things Alive Robotics GmbH, a spin-off from the Institute for Material Handling and Logistics (IFL) at Karlsruhe Institute of Technology (KIT). Things Alive Robotics provides software for mobile robots in industrial areas, from prototype to the final product.