Machine Vision and Applications (2023) 34:18
https://doi.org/10.1007/s00138-022-01364-0

ORIGINAL PAPER

Improved deep depth estimation for environments with sparse visual cues

Niclas Joswig[1] · Juuso Autiosalo[2] · Laura Ruotsalainen[1]

Received: 23 February 2022 / Revised: 4 October 2022 / Accepted: 8 December 2022 / Published online: 4 January 2023

Abstract
Most deep learning-based depth estimation models that learn scene structure self-supervised from monocular video base their estimation on visual cues such as vanishing points. In the established depth estimation benchmarks depicting, for example, street navigation or indoor offices, these cues can be found consistently, which enables neural networks to predict depth maps from single images. In this work, we are addressing the challenge of depth estimation from a real-world bird's-eye perspective in an industry environment which contains, conditioned by its special geometry, a minimal amount of visual cues and, hence, requires incorporation of the temporal domain for structure from motion estimation. To enable the system to incorporate structure from motion from pixel translation when facing context-sparse, i.e., visual cue sparse, scenery, we propose a novel architecture built upon the structure from motion learner, which uses temporal pairs of jointly unrotated and stacked images for depth prediction. In order to increase the overall performance and to avoid blurred depth edges that lie in between the edges of the two input images, we integrate a geometric consistency loss into our pipeline. We assess the model's ability to learn structure from motion by introducing a novel industry dataset whose perspective, orthogonal to the floor, contains only minimal visual cues. Through the evaluation with ground truth depth, we show that our proposed method outperforms the state of the art in difficult context-sparse environments.

Keywords Monocular depth · Deep learning · Visual SLAM · Computer vision

1 Introduction

Computing the scenery depth and ego motion of the camera is an integral part of any vision-based navigation system. The depth is essential, e.g., for obstacle detection and scene understanding which, in combination with the camera's ego motion, enables orientation in an unknown world. The objective of joint estimation of these two is referred to as *visual simultaneous localization and mapping* (SLAM), an area that has been dominated by structure from motion-based approaches, the most notable being *ORB-SLAM* [1].

Regarding the depth estimation, the fundamental difference between algorithmic frameworks such as ORB-SLAM and deep learning-based depth estimation is the underlying image structures the approaches utilize to compute their estimations. Established SLAM frameworks generate their sparse depth measures by analyzing the translational shift of a sparse set of points in consecutive images, while jointly estimating the camera pose [2]. Therefore, the algorithm is dependent on the availability of the underlying geometric features (e.g., ORB [3], BRISK [4], SIFT [5]), but independent of the higher-level scene structure. Most deep learning frameworks and the recent innovation of the *structure from motion learner (SfML)* framework [6] instead utilize only one single image to estimate dense depth, which, by definition, excludes the possibility of incorporating the temporal shift of pixels, i.e., structure from motion. Instead, deep learning-based methods can generalize over higher-level visual cues such as vanishing points or the relationship between a pixel's depth and its y-axis position [7,8].

✉ Niclas Joswig
niclas.joswig@helsinki.fi

Juuso Autiosalo
juuso.autiosalo@aalto.fi

Laura Ruotsalainen
laura.ruotsalainen@helsinki.fi

[1] Department of Computer Science, University of Helsinki, Yliopistonkatu 4, 00100 Helsinki, Uusima, Finland

[2] Department of Electronics and Nanoengineering, Aalto University, Otakaari 24, 02150 Espoo, Uusima, Finland

In this work, we aim at predicting dense depth in a novel industry environment that is characterized by a sparsity of different types of visual cues caused by the bird's-eye camera angle from an overhead crane. The existing single-image depth estimation methods which usually excel in visual cue rich scenarios such as outdoors [9] are prone to fail in this scenario as previous research has shown that these methods rely on visual cues such as vanishing points or the relationship of depth and the horizontal pixel position [7,8].

In order to improve the performance under the given circumstances posed by the bird's-eye view, we seek to regain the previously described ability to predict depth from motion with a self-supervised deep learning pipeline. Dual-image input methods such as [9], which aim at predicting depth from pixel displacement instead of visual cues, are characterized by issues with the predicted depth along edges as the model tends to mix the locations of the edges in both images. To provide supervision that counteracts this phenomena, we propose a new pipeline that integrates a geometric consistency loss into the training process.

We conducted extensive experiments on four datasets in order to validate the performance of our proposed model in different environments and to investigate whether the performance increases can be traced back to the utilization of structure from motion inside the model. To benchmark and compare our method to the reference frameworks, we created a novel *Industry Dataset* containing training images as well as test images with annotated depth. The novelty of this dataset is the camera angle, which is exactly orthogonal to the warehouse floor and to most industry indoor surfaces like tables or boxes. The images orthogonal to the ground from the crane perspective lack mentioned visual cues that are commonly found in most areas, which manifests in the difficulty of this benchmark. To our knowledge, this is the first published training and benchmark dataset that contains real-world data which depicts a bird's-eye view with ground truth depth annotations and is recorded with a high frame rate of around 10Hz. As it is the first real data for that case, the results we observe are crucial to determine the direction of the research field of monocular depth in visual cue sparse environments. We conducted experiments by training different models on the artificial dataset StillBox [10] and the established indoor depth benchmark NYU Depth V2 [11], Furthermore, we benchmark our proposed model on the *WildUAV* dataset [12] which depicts bird's-eye view shots in rural outdoor environments and contains dense annotated depth to check the generalization abilities of our model.

In summary, our main contributions presented in this paper are:

- A novel SfM learner pipeline based on an aligned image pair as DepthNet input with added scale consistency aimed at estimating depth from motion
- An industry dataset that measures any model's ability to estimate structure from object displacement in a visual cue sparse environment

2 Related work

While geometry-based approaches like ORB-SLAM [1] render sparse depth maps, learning-based methods can generate dense depth maps instead. With the rise of convolutional neural networks (CNNs), a broad spectrum of different solutions emerged to detect monocular depth in the form of *strong fusion*, which describes the joint estimation of depth from all available depth cues (e.g., texture, stereo, or kinetic depth effect) [13]. Initially, the learning thereof was facilitated in a supervised way requiring ground truth depth maps as labels [14–18]. In practice, these labels are expensive to obtain and, thus, limit the data quantity and thereby the application of deep learning methods. To cope with the given data limitations, one possibility is to generate artificial datasets [10,19], but the transfer from synthetic datasets to reality is still accompanied by a significant decrease in performance. The supervised learning approaches are overturned by frameworks that build on a novel view synthesis method based on the left–right consistency of stereo images [20–22]. If the baseline between the two cameras is known, it is possible to predict a dense depth map and subsequently warp the 3D points of one camera's image frame to the image frame of the second camera. If the estimated 3D structure is in accordance with reality, the synthesized image should resemble the real right image as close as possible, which allows it to frame the process as a stereo supervised image restoration problem.

The self-supervised SfML [6] architecture exploits the same 3D warping technique as in the stereo case, but it drops the necessity of requiring stereo recordings at train time. Instead of relating spatial stereo images with a known baseline, it predicts the camera movement between temporal images through a second neural network and subsequently warps the estimated 3D structure based on two estimated parameters—pose and depth—from one image frame to another. Based on the *Spatial Transformer* architecture [23], the whole process of warping a 3D structure became differentiable which enables CNNs to learn the 3D structure and camera movement through pure monocular supervision. In contrast to dense depth labels or stereo images, monocular video is a widely available resource which contributed to a much easier access to individual training of the SfML model in novel environments.

In both supervised and self-supervised models, the default mode is to predict depth from a single color image. The authors of [9] put forth a novel architecture based on the SfML where the *DepthNet* makes its prediction based on the input of two temporally related images. Their central aim

was, similar to ours, to establish a depth network that can incorporate structure from motion in its prediction, instead of relying only on structure from scene geometry as the single-frame approaches do. In practice, [9] beat the basic SfML framework only on the artificial *StillBox* dataset [10] which is showing random shapes and textures in a 3D space. Despite a superior performance on StillBox, the results on the autonomous driving benchmark KITTI [24] were similar and in part worse than the baseline architecture. The authors concluded from their results on KITTI and self-recorded UAV videos without depth annotation that their framework is likely to utilize both motion as well as scene structure to infer dense depth maps from image pairs.

The inherent blackbox property of neural networks makes it challenging to evaluate and reason about which image properties, e.g., scene structure through vanishing points or structure form motion through pixel displacement, guided the neural network in its current decision. Still, these insights into the networks' decision-making process are aiding in understanding the limits of generalization and possible future improvements. Various methods were developed to determine and visualize the driving factors that influence the depth networks' decisions [7,8]. The most relevant findings are that for the KITTI dataset [24], the most important visual cue is the y-axis position of any pixel inside the given image. Outdoors most scenes follow the rule that the higher up the y-axis a pixel lies, the farther away it is from the camera. Other relevant visual cues are vanishing points, which often characterize a region toward which depth converges. These insights into the blackbox processes of networks are important as they directly sketch out the generalization limits of a given model, which can be largely defined through the visual cues the model depends upon and which can be found in the given environment.

Predictions are often blurred around edges due to the model not consistently predicting the depth of the first image of an image pair, but rather a combination of both, apparent, for example, using the model presented in [9]. To tackle that problem arising with multi-image input systems, we propose to integrate a geometric consistency loss into the computational pipeline which is improving the pipeline's efficiency and consistency significantly. Despite the reported performance loss in outdoor environment, we, still, advance a system based on stacked image input as we are aiming at improving the depth prediction from an especially difficult bird's-eye viewpoint that is characterized by, for example, having no vanishing points. Using our proposed real-world industry dataset, we are conducting extensive experiments to line out which image properties, i.e., structure from motion or structure from scenery, our model utilizes to improve its accuracy compared to the reference method.

3 Methodology

We propose a novel framework that builds on the self-supervised *scale-consistent SfML* [25]. Figure 1 depicts how we altered the depth prediction pipeline as part of the geometric consistency loss. To enable depth estimation based on object motion, we augmented the depth prediction input to incorporate two images aligned by an *unrotation* mechanism [9] based on the PoseNet's rotation estimation. The estimated depth and pose are subsequently used to compute a geometric consistency loss.

3.1 Structure from motion learner

It is possible to learn structure and motion jointly self-supervised from monocular image data. The Camera's displacement between consecutive frames and the underlying scene depth can be estimated by an architecture based on two neural networks called *DepthNet* and *PoseNet* [6]. The *DepthNet* is a fully convolutional ResNet based on an autoencoder architecture that maps an image I to a dense depth map D of the same size. The second network called *PoseNet* has the same structure in its encoder part, but, instead of upscaling, convolutions are subsequently applied. Then, the spatial dimensions are averaged to generate a pose vector P consisting of six output values addressing the six-degree-of-freedom movement.

The relationship between two temporal frames can be parameterized by the scene depth and the camera's ego motion between two time epochs of taking the images. Mathematically, the geometric relationship between the image points p in the image frame of time step $t-1$ (*source image* or I_s) and $\hat{p}$, the same real-world points in the image at time step t (*target image* or I_t), can be described by function ω as follows:

$$\hat{p} = \omega(p;\, D_t;\, P_{t\to s}) = K\, P_{t\to s}\, D_t\, K^{-1}\, p. \tag{1}$$

After using deep learning to predict the parameters D_t and $P_{t\to s}$, it is possible to relate both images by the *inverse warping* function ω describing the transformation of image points from one image frame into another. First, $2D$ target image points p are projected into $3D$ space using the inverse camera calibration matrix K^{-1} and the estimated dense depth map D_t. Then, a $3D$ point cloud is generated in the target camera frame and subsequently transferred into the second camera frame by a matrix multiplication with the predicted camera pose $P_{t\to s}$. Finally, points are backprojected into the source image's $2D$ space using the intrinsic matrix K.

The resulting $2D$ points $\hat{p}$ require *bilinear sampling* to evaluate the accuracy of the networks' estimate for depth and pose with an RGB-based *photometric error*. Bilinear

sampling is defined as the process of sampling RGB values from an image at some specified image points, in this case the warped image points $\hat{p}$. As these points usually do not lie exactly on the image grid, the RGB values are bilinear interpolated to fit into the discrete image pixel grid [2]. This step has been made differentiable only recently through the introduction of *spatial transformers* [23] which is a technique to apply dynamic spatial transformations, like affirmative transformations, inside a differentiable network pipeline. In this case, it enables the pipeline to sample the RGB values at the computed coordinates and interpolating them while being differentiable. To improve the quality of the networks' prediction of depth and pose, the sampled RGB image can be made subject to a photometric error term comparing the intensities pixel wise using the source image's pixels as labels. This error term, which is minimized during backpropagation, is denoted as L_{photo} in formula 2 where $I(p)$ describes bilinear sampling of the points p from Image I and V is the total set of pixels in the target image.

$$L_{photo} = \frac{1}{|V|} \sum_{p \in V} \frac{1}{2} \| I_t(p) - I_s(\hat{p}) \|^2. \tag{2}$$

The downside of unsupervised training is that the scale factor s which is used to transform the output into metric units is by default unknown. To evaluate our depth predictions, a technique called *median alignment* is used defining the scale factor between prediction and ground truth as $s = median(D_{pred}/D_{GT})$ [6,25].

3.2 Multi-image input

In comparison with established visual SLAM frameworks *ORB-SLAM* [1] or *Direct Sparse Odometry* [26], the SfML framework extracts the depth from a single monocular image. Mathematically, the neural network represents a function with the target image as input: $D_t = DepthNet(I_t)$. This design implies that the network is learning depth from scene structure, i.e., cues that relate directly to the underlying depth, and not from the temporal motion of pixels.

As the goal of this work is to estimate the depth from motion in scenes where visual cues are not sufficiently existing, we incorporated the time domain into our architecture. The time domain theoretically contains full information for pixels that are visible in multiple consecutive frames. For including the temporal domain to the depth computation, *recurrent neural networks* in the form of *LSTM networks* are generally the first choice [27,28]. However, in this concrete use case it was important to exclude the possibility that the system estimates depth from scene structure in some places like the outer image areas and then simply propagates that knowledge across time. Such strategy could be effective depending on the camera trajectory, but the strategy would also be prone to error in the simple case of an object never being seen in an angle that allows precise understanding from scene geometry.

As the basis for our work, we chose the network structures and loss functions from the *scale-consistent SfML* (SC-SfML) [25], which constitutes the current state of the art in deep learning-based monocular SLAM. Based on the previous reasoning, we augmented the DepthNet to have two input images, instead of one. The dual-image input is shown in Fig. 1.

In comparison with single-image methods, our method uses more information in the form of two images separated by rotation and translation which, in theory, means that a dual-image approach should always outperform the single-image method. The authors of [9] have demonstrated on the KITTI dataset [24] that this is not the case as they report worse results with a dual-image method than with the single-image baseline. One reason for this phenomena is the rotational movement that acts as pure noise in the depth prediction task from two images and makes it harder to access the movement's translational component. This is addressed with an *unrotation* process explained in the following section, which is fundamentally limited by the PoseNet's performance. Another issue visible in the estimated depth maps is that the estimated depth is a blurred mixture of the depth predictions of both single images [9]. Therefore, predicted edges often lie in between the color edges of the two images and not precisely at the location of the target image edge. In this work, we are trying to overcome this challenge through the integration of a geometric consistency loss described in the latter part of this chapter.

3.3 Unrotation of consecutive images

The rotational camera movement acts as noise for the depth prediction [9,29] and should be filtered out. The observation of the scene depth is independent of the camera rotation and depends only on camera translation [2]. When feeding two images into any depth algorithm, implicit and explicit options exist to cope with rotational noise. The implicit option does joint optimization of rotation and depth, which means that the depth network itself extracts information about the rotational movement and takes that into account. The explicit alternative has an external instance to calculate the rotational parameters. Before predicting the depth, these parameters can be applied in order to align the camera angle so that, with optimal prediction, only translation separates the image pair. Addressing the rotation in an explicit step [9] gives not only the possibility to reason about the accuracy thereof, but also the opportunity to use, e.g., artificial datasets with ground truth poses to test the DepthNet's performance under ideal conditions [9].

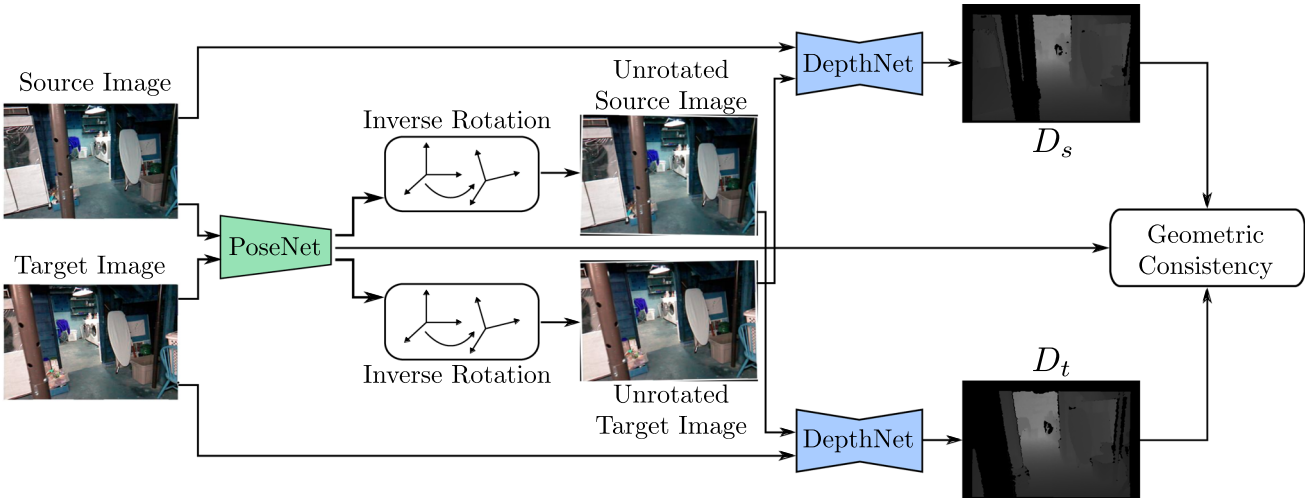

Fig. 1 Architecture overview simplified with one source image

As previously stated, the rotational relationship between two images is independent of the underlying scene depth. Therefore, it is possible to apply a 3D rotation matrix to an image without knowledge of the underlying scene structure. The central difference to the inverse warping process described above is that we do not multiply the 3D points, generated by projecting the 2D points x to 3D via the inverse calibration matrix K^{-1}, with the depth map D and, instead, use the normalized 3D points where all points have a constant depth of 1. The whole unrotation process from 2D to 3D back to 2D is depicted in Eq. 3. Because the depth is set to be planar with a constant value of 1, the relationship between image points in two images $x = (u_1, v_1)$ and $x' = (u_2, v_2)$ is defined by a 2D projective transformation, a homography. In the special case when the camera is experiencing only rotational movement, the homography matrix H containing the homography parameters is a $3x3$ matrix product of the camera calibration matrix K, its inverse K^{-1} and the rotation matrix R [2].

$$x' = KRK^{-1}x = Hx. \tag{3}$$

The unrotated images have some missing points along the borders as some pixels have been rotated out of the image frame. Padding is used to fill missing pixel values (e.g., zeros, border values). Figure 2 shows an example rotation of an image from the NYU dataset [11] with zero padding applied. When the image is rotated and the free space filled with zeros, it becomes clear that for some of the outer pixels from the target image the corresponding pixel in the source image lies outside of the boundaries. Figure 1 shows that the rotated image always acts as complementary information to the stock image whose depth should be predicted. Hence, all pixels in the current target are visible for depth prediction, but some pixels do not have a corresponding pixel in the rotated auxiliary image. Logically, it is physically impossible to estimate the depth of such pixels based solely on the pixel displacement. This phenomena acts as noise for depth from motion and could be a factor that disincentives the network to learn and converge toward structure from motion. However, in the industry use case and also in an indoor pedestrian case the ego motion is generally slow compared to, for example, autonomous driving due to safety and physical limitations. As a result of the slow motion, the total amount of pixels that are missing a correspondence is small compared to the total amount of pixels. Furthermore, in small areas where no correspondence is found, the network can still apply the smoothness prior to interpolate those areas.

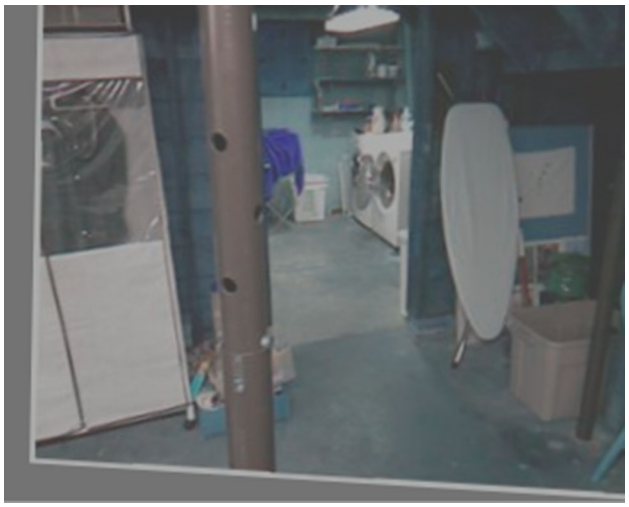

Fig. 2 Example image from the NYU dataset [11] with applied roll, yaw and zero padding

3.4 Scale consistency with multi-image input

Because rotation constitutes noise for the depth prediction, we align the camera and the adjacent image perspectives using unrotation. Unrotation is applied before estimating depth maps for target as well as source images in order to apply the geometric loss [25] for enforced geometric pose and depth consistency. The geometric loss computation is visualized in Fig. 1. First, the ego motion between the target and source images ($P_{t\to s}$), as well as the inverse ($P_{s\to t}$), is obtained by feeding them into the PoseNet. To estimate the target's depth D_t, one random source image is chosen and subsequently inversely rotated into I'_s with the rotational component of the pose $P_{s\to t}$. This pair of images, I_t and I'_s is used as input for the DepthNet computing the dense depth map D_t. To estimate a depth map for each source image, the process is inverted by unrotating poses from target to source $P_{t\to s}$ perspective. Then, the respective pairs of images can be feed into the DepthNet to acquire a depth map D_s for every source image in the minibatch.

After the depth estimation and corresponding poses are obtained for every image in the sequence by a feed-forward process, the target image points p are warped with the inverse warping function ω from Sect. 3.1 to get points $\hat{p}$. $\hat{p}$ should describe the same real-world points as p in the target image frame. Unlike in the photometric error calculation, the sampling takes place from the source image's depth map D_s instead of the RGB image, in order to compare the result to the interpolated depth map $D'_t(p)$ which results in D_{diff}.

$$D_{diff}(p) = \frac{\mid D_s(\hat{p}) - D'_t(p) \mid}{D_s(\hat{p}) + D'_t(p)}. \tag{4}$$

This term is subsequently subject to loss minimization in the backpropagation process:

$$L_{geo} = \frac{1}{|V|} \sum_{p \in V} D_{diff}(p). \tag{5}$$

Asides from the advantage of forcing the network into predicting scale-consistent outputs, the geometric consistency also provides an additional supervision signal to the network ensuring the prediction of a depth map for one image (at position 1) instead of a combination of both images, which would result in blurred textures in places where the two images differ. Furthermore, by being able to compute the geometric consistency loss, we are able to generate a geometric consistency mask $M = 1 - D_{diff}$ which stabilizes the model's training in dynamic scenarios.

4 Experiments

To evaluate our proposed method, we conducted experiments on four datasets assessing different qualities of our depth prediction model. Here we give an overview of the error measures assessing our models and compare results to the state-of-the-art methods.

4.1 Implementation details

For our proposed pipeline, we choose to use the PoseNet from [6] without the uncertainty prediction branch. The fully convolutional network receives multiple images stacked along the color axis as input and generates one vector per source image that contains the predicted 6DOF camera movement between target and source image. For the depth prediction, we used the *DispResNet* from [30] as DepthNet. We augmented the input layer to take in two stacked images of the format $height \times width \times 6$ instead 3 channels. The input is fed through a fully convolutional ResNet [31] encoder and decoder with skip layers to transform the input into one dense depth map of size $height \times width$ in a single-scale design [25].

Our framework builds upon the code of [25] which is based on the PyTorch Library [32]. For the training process, we group the data into sequences of 3 consecutive images where the middle frame is assigned to be the target image. As the sequences contain two source images each, the feed-forward process depicted in Fig. 1 is carried out once for each source image. To optimize the architecture further, we adopted the mechanisms of [25] to compute the photometric and geometric loss in both directions, i.e., once warp the target image to the source image and then the other way around. We also activated the stationary pixel mask from [33] in every training run.

All training runs that are not explicitly pretrained use weights generated through training on the ImageNet [34] dataset. For optimization, we use *Adam* [35] with a learning rate of 0.0001, batch size of 4 and the parameters $\alpha = 1.0$, $\beta = 0.1$ and $\gamma = 0.5$ from [25]. The total loss function to be optimized is weighted as follows:

$$L_{total} = 0.9 \cdot L_{photo} + 0.5 \cdot L_{geo} + 0.1 \cdot L_{smooth}. \tag{6}$$

Therein the terms L_{photo} and L_{geo} refer to the photometric and geometric loss described in chapter 3 and L_{smooth} refers to the edge-aware smooth loss from [25].

4.2 Datasets

StillBox

The artificially created *StillBox* dataset [10] demonstrates an environment where estimating structure from a single image

should be close to impossible. This is done by generating sequences of images which contain random 3D shapes with random colors, and a fraction of the visible shapes are covered by randomly chosen images from the *flickr* platform. The random size and arrangement of objects make it impossible to rely on the size and position thereof. When the system is trying to infer depth from observable textures, the surfaces covered by images will cause problems as the visible image textures are likely to depict a non-planar scene, while the actual shape is plain. In this work, we use the StillBox dataset only for pretraining our models as it does not sufficiently represent a real-world use case and evaluation would make little sense.

NYU Depth V2

NYU Depth V2 [11] dataset consists of 512 indoor scenes recorded with a *Microsoft Kinect* and encompasses densely labeled ground truth depth maps for validation and testing. We choose to entail this dataset because it resembles the indoor crane environment closely and, additionally, contains comparatively much rotational movement [29] in comparison with, e.g., outdoor datasets. For training, we used the same train/validation split as [29]. All images were undistorted and the total number of frames was downsampled by a factor of 10. For evaluation, we used the official test set encompassing 562 images from the labeled subset of the NYU dataset, which are not included in either training nor validation data.

Industry Dataset

The StillBox [10] dataset has been created to train unsupervised models to estimate structure from motion instead of visual cues. As it is artificially generated, it includes specific scenes that are specifically difficult for single-image estimation and, hence, encourage structure from motion-based prediction. While it is a valuable benchmark, the scenes are highly unrealistic and do not really represent the real world.

To bridge the gap in depth from motion evaluation between artificial datasets and real-world environments, we propose a new *industry dataset* containing images from the perspective of an indoor warehouse crane with a bird's-eye view. The cameras were mounted orthogonal and with a maximum distance to the floor of around $3.5m$. The only natural geometric constraint for the scene structure is that the depth is capped to a certain distance by the floor and everything that may not be part of the floor is closer to the camera. Therefore, visual cues like a horizon line or vanishing points cannot contribute to the estimation. Furthermore, the depth is, unlike to most common environments, completely unconstrained from the y-axis position of any given pixel.

The dataset consists of two types of sequences recorded at different times in the *Ilmatar Innovation Environment* [36]. All RGB images have been recorded with a *Realsense D435* camera. First, three sequences covering the whole area were recorded when the premise was only sparsely filled with objects and in daily use. In this case, the camera was mounted to the hook of a Konecranes CTX crane that enabled translation into all three dimensions and rotation as the hook is swinging during movement. The second part of the dataset, 20 sequences, was recorded in the same premise when under construction. Here the camera was fixed to the overhead crane alongside an *Ouster OS0* Lidar for the ground truth depth. The sequences contain two calibration sequences: eight sequences that depict mostly construction site objects and 10 sequences with a flat wooden board in varying positions. By positioning the board at different heights and angles, we constructed a scenery where the board depth can only be estimated by its motion w.r.t. to the background. From the perspective orthogonal to the board, the montage equipment are not visible and, therefore, the scene context does not provide the board's height over the ground.

Due to the overall small size of samples, we only fine-tuned our model on this data, instead of training a model from scratch. To avoid overfitting through the overall small data variance constrained by the single environment, we capped the training at 5 epochs with 1000 batches of 4 samples. To evaluate the models' performances, we generated a test set which only contains images that were not used in training. The test set's images cover all combinations of absolute and relative board positions, i.e., board cutoff by the left/right image border, in the lower/higher image area and in the middle.

WildUAV

The WildUAV dataset [12] is recorded in an outdoor area characterized by fields, trees and streets by an UAV with a camera mounted with an orthogonal view to the ground similar to the crane perspective. One main difference to the industry environment is the much larger distance toward objects which lies around 50 m on average. This is an important factor for this work as with increased depth the object displacement between two images decreases and the differences in displacement are much more subtle than in a close up environment. Another significant aspect about this dataset is the comparatively low frame rate of around 1Hz, which creates large distances between images and smaller image overlaps. Especially when the UAV is rotating in place, the low frame rate results in more than 90 degrees of roll angle between consecutive images with no translational movement. The single-image-based methods are having an advantage in this regard as they are invariant to the overlap of consecutive images caused by large motion or low frame rates. Our method based on stacking consecutive images, on the other hand, is directly affected by this properties, especially because the PoseNet used to unrotate the image is trained on higher frame rates with correspondingly lower translations and rotations and, hence, fails to predict such large magnitudes of rotation needed for proper rotational align-

Table 1 Results from NYU dataset. Methods have been trained on the train/val split and tested as described in 4.2. The pretraining is indicated by the *Training* column. The two letter code (e.g., S-D) stands for the pretraining dataset (*S*—StillBox, *N*—NYU) and the second letter for the transferred network (*D*—DepthNet, *P*—PoseNet). All results are aligned with the ground truth by median alignment. Arrows indicate whether high or small values are better and the best solutions are bolded. In the runs marked by a *, the source image has been replaced by an only zeroes array

Method	Pretraining	AbsDiff⇓	AbsRel⇓	SqRel⇓	RMS⇓	LogRMS⇓	AbsLog⇓	δ_1 ⇑	δ_2 ⇑	δ_3 ⇑
SC-SfM learner [25]	–	–	0.166	–	0.621	–	–	0.755	0.934	0.981
Rectified SC-SfM learner [29]	–	–	**0.147**	–	**0.536**	–	–	**0.804**	**0.950**	**0.986**
Ours	–	0.5704	0.1750	0.1837	0.7682	0.2178	0.1680	0.7469	0.9289	0.9793
Ours	S-D,N-P	0.5475	0.1654	0.1670	0.7430	0.2083	0.1605	0.7617	0.9386	0.9838
Ours	S-D	0.5847	0.1798	0.1910	0.7849	0.2222	0.1718	0.7338	0.9266	0.9787
Ours*	S-D,N-P	0.5496	0.1667	0.1689	0.7445	0.2093	0.1612	0.7599	0.9377	0.9834

Bold indicates the best result in the indicated metric

ment. The authors provide 4 sequences in their *Mapping Set* that contain dense depth annotations. We noticed in all sequences that they are not continuous and have consecutive images that have no overlapping area. As previously stated, this unwanted behavior would make no difference for the single-image methods, but constitutes a big challenge for multi-image-based methods. To address this issue, we manually split all 4 sequences into subsequences where two consecutive images have no overlapping area. After this modification, the dataset contains in total 1424 three-image batches for testing.

4.3 Results

NYU Depth V2

The results we obtained while benchmarking the SC-SfML and our method on the industry dataset are aggregated in table 1.

The baseline framework of the SC-SfML [25] performed better than our method when trained only on the NYU dataset. SC-SfML outperforming our method could be because of two reasons: Firstly, it is possible that in the dual method to estimate depth from structure the second input image produces merely noise [9]. Secondly, the network learned to estimate structure from motion, but, because of the models' *or* environment properties, the performance is inferior to the baseline. Pretraining the DepthNet on the StillBox dataset, which primes the network already for structure from motion before the training begins, could also not increase the performance. This result is an indicator that the first reason is more likely, as the structure from motion performance should be improved by this pretraining.

Finally, we transferred weights from two different training runs. We utilized a DepthNet trained on StillBox as described previously. Additionally, we trained a PoseNet with the s-o-t-a SC-SfML architecture, which is outperforming our method when trained on only the NYU dataset. Then, we initialized the PoseNet of our method with the weights obtained from the SC-SfML run on NYU and the DepthNet with the pretrained StillBox weights of our method.

In summary, the training starts with a DepthNet that is primed to estimate structure from motion and a PoseNet that performs rotation estimation on the current s-o-t-a level. In this setting, our method outperformed the single-image baseline of the SC-SfML by a slim margin of *absolute relative error* (AbsRel) 0.004, which is so small, and it could likely be caused by noise or non-deterministic processes inside the training.

Despite our architecture being build for difficult structure free scenarios, the observed performances indicate that the proposed method performs on par with the s-o-t-a indoors. To test whether the performance is achieved through incorporating structure from motion into prediction, we replaced the source image with a plane array of zeros. The result in the last row of Table 1 is comparable to the one with a proper source image in the 4th row, which is a strong indication that, despite the architectural changes and specific pretraining, the source image does not affect the performance significantly in the indoor environment case.

Industry Dataset

Our industry dataset is used as a benchmark to evaluate monocular depth prediction in difficult scenarios, differing bird's-eye viewpoints, minimal visual cues and physical limitations. The results on the industry benchmark are aggregated in Table 2. Our method pretrained on the StillBox dataset outperformed the baseline of SC-SfML [25] by an AbsRel margin of 0.0078, which is, considering the overall smaller differences in depth compared to, e.g., NYU evaluations, a significant improvement. When pretrained first on the StillBox dataset and subsequently on the NYU dataset, the performance of 0.0490 AbsRel error lies much closer to the baseline solution at 0.0505 AbsRel error.

To analyze whether our model utilizes the additional input image, we, again, tested the model with an array of zeros as replacement for the source image. The results of this

Table 2 Results on the Industry dataset benchmark. All models have been fine-tuned on the train data of the industry dataset. N = NYU depth [11], S = StillBox [10]. A + indicates sequential training on two datasets. Arrows indicate whether high or small values are better and the best solutions are bolded. The entries with a *zeros* annotation are the evaluations where a flat array of zeros replaced the source image and the $2 \times tgt$ annotations refers to the benchmark with two times the target image stacked

Method	Pretraining	AbsDiff⇓	AbsRel⇓	SqRel⇓	RMS⇓	LogRMS⇓	AbsLog⇓	δ_1 ⇑	δ_2 ⇑	δ_3 ⇑
SC-SfM learner [25]	N	0.1169	0.0505	0.0217	0.1594	0.0670	0.0466	0.9759	0.9956	0.9979
Ours	S	**0.0991**	**0.0427**	**0.0162**	**0.1385**	**0.0582**	**0.0396**	0.9730	**0.9975**	0.9980
Ours	S+N	0.1122	0.0490	0.0217	0.1554	0.0653	0.0447	**0.9779**	0.9965	**0.9981**
Ours*	S	0.2203	0.0907	0.0473	0.2804	0.1116	0.0848	0.9050	0.9947	0.9978
Ours*	S+N	0.1274	0.0547	0.0270	0.1750	0.0726	0.0503	0.9698	0.9950	0.9979
$Ours^{2\times tgt}$	S	0.1039	0.0445	0.0170	0.1441	0.0606	0.0413	0.9754	0.9977	0.9980

Bold indicates the best result in the indicated metric

experiment, marked with a * in the bottom part of Table 2, show that the best model pretrained on StillBox doubles its AbsRel error if exposed to a missing context image. The observed performance decrease is a first indicator that the model incorporates the temporal relationship of consecutive frames in its prediction. Replacing the source image when our model is pretrained on StillBox and NYU dataset decreases the AbsRel error only by a slim margin of 0.0057, which, again, highlights that the NYU-pretrained model does make almost no meaningful use of the additional input image or the temporal domain.

To investigate the model pretrained on the StillBox dataset even further, we conducted another benchmark where we input twice the same target image stacked instead of target and source frames. The results in the last row of Table 2 show a performance decrease compared to the benchmark with a valid reference image. Still, the results with two target images are better than the single-image method which leads to the conclusion that the displacement between the images plays a role in the depth estimation, but does not account for the total performance improvement in relation to the single-image method. In total, we can conclude that the performance improvement originates partly from the model's ability to utilize the additional information in the form of pixel displacement.

Example predictions of the models are depicted in Fig. 3. For visualization purpose, the sparse Lidar point cloud has been interpolated to present a dense ground truth depth image in the far right column. The two top rows show the critical case of the wooden board laying directly on the floor. The board is characterized by a different texture than the floor, and therefore, a model predicting depth from textures might estimate the plate as an object that is closer to the camera than the floor. This phenomenon is observable in the baseline prediction on the left. It wrongly identifies the plate as being closer than the floor, while our method correctly recognizes the depth of the plate to be similar to the background. As it can be seen in the third row and, with the one exception depicted in the last row, throughout the whole test set, the board position is estimated correctly when it is lifted in the air. This behavior of misclassifying the grounded board and correctly estimating the lifted ones indicates that, based on the missing visual cues, the textures became the guiding principle for the single-image method. Using only texture information trivially leads to the phenomena observed in the upper images where a specific texture appears at different positions and receives the same (in this case wrong) prediction. Our method instead was able to recognize the different positions of the board by incorporating the time domain alongside the texture information.

The lower two rows show observed error cases of both our models and the baseline model. The second last test image depicts a floor that is textured by vertical lines on the right image side and two plain surfaces of different colors on the left side. All models show large artifacts of misprediction, which may be due to the interpretation of the plain surfaces as objects closer to the camera. Finally, the last row covers the case of the board being lifted in the air and located at the top right corner of the image. It can be seen that no model got the depth fully accurate, but the baseline does recognize more board pixels as closer than our model.

Overall, the observed error values in combination with the visualized results show that our method, pretrained on the StillBox dataset, makes use of the temporal image domain. This ability, to incorporate structure from motion, enables our method to outperform the s-o-t-a method in difficult scenarios that offer minimal visual cues. In practice, our method faces the limitation of being reliant on two images separated by translational movement of the camera. If the crane and, therefore, the camera are in a static position, an input pair lacks the information of the temporal domain and outputs might be inferior to those obtained from a single-image method. Also, if the camera experiences only rotational movement, the same limitation applies, as the rotation does not contain meaningful information about the underlying depth [2]. In the specific case of an indoor crane environment, this would not cause practical problems as the scenery usually does not

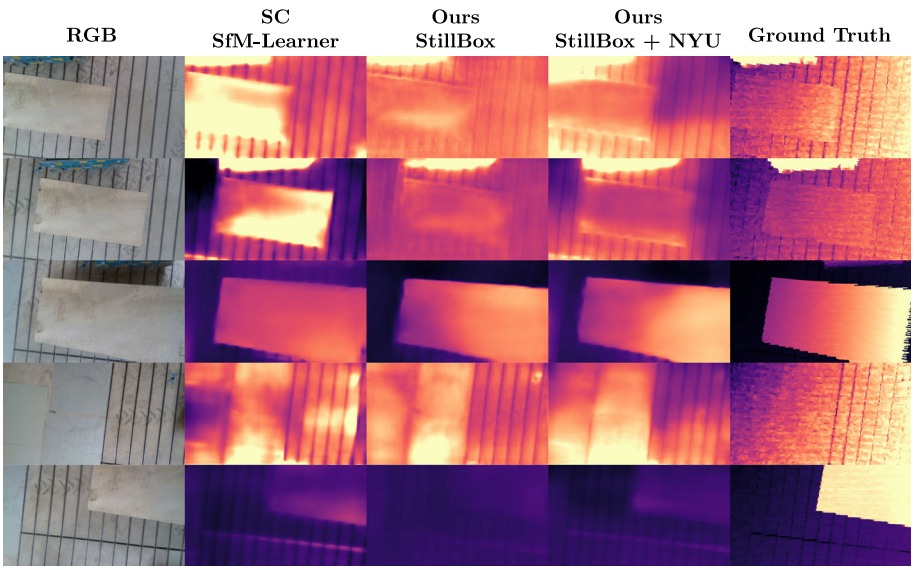

Fig. 3 Example results from the industry dataset benchmarks. Every row depicts one RGB image on the left next to the different predictions and the ground truth on the right

Table 3 Results on the WildUAV dataset benchmark [12]. For our method and the SC-SfML as baseline, we used the best performing models on the industry dataset for benchmarking. The *Training* column highlights that for our method we used a model that was trained on StillBox (S) and the Industry dataset (I) and for the reference solution we used the best model trained on NYU (N) and Industry (I) dataset

Method	Training	AbsDiff⇓	AbsRel⇓	SqRel⇓	RMS⇓	LogRMS⇓	AbsLog⇓	δ_1⇑	δ_2⇑	δ_3⇑
Ours	S+I	**4.8126**	**0.1095**	**0.9921**	**6.0671**	**0.1366**	**0.1098**	0.8654	**0.9896**	0.9992
SC-SfM learner [25]	N+I	4.8244	0.1108	0.9965	6.2864	0.1413	0.1110	**0.8669**	0.9874	**0.9994**

Bold indicates the best result in the indicated metric

change significantly without crane and camera movement. The results further show that the proposed method outperforms the baseline in depth cue sparse environments, but it is likely to under perform in areas where a plethora of cues are visible, as reported in [9] for the KITTI [24] dataset.

UAV Dataset

Asides from our proposed industry dataset, we benchmark our solution on a public UAV dataset whose viewpoint resembles the targeted bird's-eye view. As mentioned in previous chapter 4.2, the WildUAV [12] dataset was recorded with a comparatively low fps, which makes an unsupervised training on this dataset difficult to impossible as the PoseNet cannot learn the contained large-scale translation and rotation from such a small amount of data, which renders the photometric error useless. Hence, we used the best models trained on the industry dataset for both our proposed method and the reference SC-SfML. The results based on the provided ground truth depth maps are depicted in Table 3.

The results show a similar picture to the results observed in the crane environment where our method outperforms the single-image baseline. In this dataset, the improvement in especially the relative errors such as the absolute relative error (AbsRel) is smaller compared to the one observed in the industry dataset. We argue that the less significant performance improvement is due to the small overlapping areas in consecutive images, i.e., large-scale translations and pure rotations, that persist throughout the dataset due to the complicated application area.

5 Conclusion

In this paper, we have presented a new SfML-based architecture that aims at incorporating the temporal domain of image sequences for monocular depth prediction in a difficult industry environment characterized by sparse visual cues. Our pipeline aligns the camera angle of consecutive image pairs, before predicting each images' depth on the basis of a two frame input to finally compute a geometric consistency loss. To validate out method, we proposed a novel industry dataset recorded from the perspective of an indoor crane and show that our method outperforms the current s-o-t-a methods in such environments characterized by minimal visual cues such as vanishing points. In the future, we would like to investigate both networks' architecture more closely for a possible way to further improve the use of the temporal domain and to incorporate the translation prediction explicitly into the depth estimation process.

Acknowledgements This work has been supported by a donation from Konecranes, Finnish Center for Artificial Intelligence (FCAI), the University of Helsinki and Aalto University.

Funding Open Access funding provided by University of Helsinki including Helsinki University Central Hospital.

References

1. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: a versatile and accurate monocular slam system. IEEE Trans. Rob. **31**, 1147–1163 (2015)
2. Hartley, R. I., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, ISBN: 0521540518, second ed., (2004)
3. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf, in ICCV, pp. 2564–2571 (2011)
4. Leutenegger, S., Chli, M., Siegwart, R.: Brisk: Binary robust invariant scalable keypoints. In: IEEE ICCV, pp. 2548–2555, 11 (2011)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**, 91–110 (2004)
6. Zhou, T., Brown, M., Snavely, N., Lowe, D.: Unsupervised learning of depth and ego-motion from video, in CVPR, (2017)
7. Van Dijk, T., De Croon, G.: How do neural networks see depth in single images?, In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 2183–2191 (2019)
8. Hu, J., Zhang, Y., Okatani, T.: Visualization of convolutional neural networks for monocular depth estimation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3868–3877 (2019)
9. Pinard, C., Chevalley, L., Manzanera, A., Filliat, D.: Learning structure-from-motion from motion. In: ECCV, (2018)
10. Pinard, C., Chevalley, L., Manzanera, A., Filliat, D.: End-to-end depth from motion with stabilized monocular videos. ISPRS Annals **4**, 67–74 (2017)
11. Silberman, P. K. Nathan, Hoiem, Derek, Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV, (2012)
12. Florea, H., Miclea, V. -C., Nedevschi, S.: Wilduav: Monocular uav dataset for depth estimation tasks. In: 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP), (2021)
13. Landy, M.S., Maloney, L.T., Johnston, E.B., Young, M.: Measurement and modeling of depth cue combination: in defense of weak fusion. Vision. Res. **35**(3), 389–412 (1995)
14. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Vol 2, NIPS'14, (Cambridge, MA, USA), pp. 2366-2374, MIT Press, (2014)
15. Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N.: "Deeper depth prediction with fully convolutional residual networks," in 2016 Fourth International Conference on 3D Vision (3DV), pp. 239–248, (2016)
16. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2002–2011, (2018)
17. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: Proceedings of the 27th International Conference on Neural Information Processing Systems - Vol 2, NIPS'14, (Cambridge, MA, USA), pp. 2366-2374, MIT Press, (2014)
18. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2650–2658, (2015)
19. Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., Brox, T.: What makes good synthetic training data for learning disparity and optical flow estimation? Int. J. Comput. Vis. **126**, 942–960 (2018)
20. Godard, C., Aodha, O. M., Brostow, G. J.: Unsupervised monocular depth estimation with left-right consistency. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6602–6611, (2017)
21. Amiri, A. J., Loo, S. Yan, Zhang, H.: Semi-supervised monocular depth estimation with left-right consistency using deep neural network. In: 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 602–607, (2019)
22. Yang, N., Wang, R., Stuckler, J., Cremers, D.: Deep virtual stereo odometry: leveraging deep depth prediction for monocular direct sparse odometry. In: European Conference on Computer Vision (ECCV), (2018)
23. Jaderberg, M., Simonyan, K., Zisserman, A., kavukcuoglu, k.: Spatial transformer networks. In: Advances in Neural Information Processing Systems (C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, eds.), vol. 28, Curran Associates, Inc., (2015)
24. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The kitti vision benchmark suite. In: CVPR, (2012)
25. Bian, J.-W., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M. -M., Reid, I.: Unsupervised scale-consistent depth and ego-motion learning from monocular video. In: NeurIPS, (2019)
26. Engel, J., Koltun, V., Cremers, D.: Direct sparse odometry. IEEE Trans. Pattern Anal. Mach. Intell. **40**(3), 611–625 (2018)
27. Zhang, L., Li, G., Li, T. H.: Temporal-aware sfm-learner: unsupervised learning monocular depth and motion from stereo video clips. In: IEEE MIPR, pp. 253–258, (2020)
28. Wang, R., Pizer, S. M., Frahm, J.-M.: Recurrent neural network for (un-)supervised learning of monocular video visual odometry and depth. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5550–5559, (2019)
29. Bian, J. -W., Zhan, H., Wang, N., Chin, T. -J., Shen, C., Reid, I.: Unsupervised depth learning in challenging indoor video: Weak rectification to rescue. arXiv preprint arXiv:2006.02708, (2020)
30. Ranjan, A., Jampani, V., Balles, L., Kim, K., Sun, D., Wulff, J., Black, M. J.: Competitive collaboration: joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12232–12241, (2019)
31. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, (2016)
32. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L. and Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Inf. Processing Systems 32, pp. 8024–8035, Curran Associates, Inc., 2019

33. Godard, C., Aodha, O. Mac, Firman, M., Brostow, G. J.: Digging into self-supervised monocular depth prediction. In: ICCV, (2019)
34. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE CVPR, pp. 248–255, IEEE, (2009)
35. Kingma, D. P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (Y. Bengio and Y. LeCun, eds.), (2015)
36. Autiosalo, J.: Platform for industrial internet and digital twin focused education, research, and innovation: ilmatar the overhead crane. In: IEEE WF-IoT, pp. 241–244, (2018)

Niclas Joswig University of Helsinki, Helsinki, Finland Niclas Joswig received the M.Sc. degree from the Department of Computer Science, University of Helsinki, in 2021. He is pursuing the PhD degree with University of Helsinki in the Spatiotemporal Data Analysis group at the Faculty of Computer Science. His current research interests include vision-based navigation methods in challenging indoor environments and deep learning-based depth estimation from monocular images.

Juuso Autiosaalo Aalto University, Espoo, Finland Juuso Autiosalo received the B.Sc., M.Sc., and D.Sc. degrees from Aalto University, Espoo, Finland, in 2015, 2017, and 2021, respectively, where he is currently working as a Postdoctoral Researcher. He was Visiting Research Scholar at Michigan State University in 2020. His research focuses on digital twins, and he is currently working to enable the creation of a global network of digital twins. Autiosalo has instructed two bachelor's and eight master's theses from three disciplines and performed teaching activities at three university courses. He was the project manager for the DigiTwin project which was executed in tight collaboration with industry partners. Juuso's background is in mechanical engineering, and he likes to get his hands dirty.

Laura Ruotsalainen University of Helsinki, Helsinki, Finland Laura Ruotsalainen is Associate Professor in computer science. She leads a research group in spatiotemporal data analysis for sustainability science, which performs research on estimation and ML methods using spatiotemporal data. She has a long research career in the navigation field including computer vision and sensor fusion in navigation and deep learning methods for GNSS interference characterization and mitigation. She is a Member of the steering group of the Finnish Center for AI and the Chair to a Sub-Commission "Emerging Positioning Technologies and GNSS Augmentations" in the International Association of Geodesy.