Self-Supervised Monocular Depth Estimation on Unseen Synthetic Cameras

Cecilia Diana Albelda$^{1\,[0009-0009-9210-0853]}$, Juan Ignacio Bravo Pérez-Villar$^{1,2\,[0000-0003-3199-5779]}$, Javier Montalvo$^{1\,[0000-0001-6610-1566]}$, Álvaro García Martín$^{1\,[0000-0002-1705-3972]}$, and Jesús Bescós Cano$^{1\,[0000-0001-6238-6859]}$

[1] Video Processing and Understanding Lab, Univ. Autónoma de Madrid, 28049 Madrid, Spain {`cecilia.diana,juanignacio.bravo, javier.montalvor`}`@estudiante.uam.es` {`alvaro.garcia,j.bescos`}`@uam.es`
[2] Deimos Space, 28760 Madrid, Spain `juan-ignacio.bravo@deimos-space.com`

Abstract. Monocular depth estimation is a critical task in computer vision, and self-supervised deep learning methods have achieved remarkable results in recent years. However, these models often struggle on camera generalization, i.e. at sequences captured by unseen cameras. To address this challenge, we present a new public custom dataset created using the CARLA simulator (4), consisting of three video sequences recorded by five different cameras with varying focal distances. This dataset has been created due to the absence of public datasets containing identical sequences captured by different cameras. Additionally, it is proposed in this paper the use of adversarial training to improve the models' robustness to intrinsic camera parameter changes, enabling accurate depth estimation regardless of the recording camera. The results of our proposed architecture are compared with a baseline model, hence being evaluated the effectiveness of adversarial training and demonstrating its potential benefits both on our synthetic dataset and on the KITTI benchmark (8) as the reference dataset to evaluate depth estimation.

Keywords: Monocular Depth Estimation · Computer Vision · Self-Supervised Learning · Camera Generalization · Custom Synthetic Dataset · Adversarial Training.

1 Introduction

Monocular depth estimation is the process in which a depth map is obtained from an RGB image. This is a fundamental task in computer vision, in particular for autonomous driving (9), computational medicine (14) and many others (1; 21).

Traditional methods typically use multi-view techniques through epipolar geometry or feature matching to perform depth estimation. However, these approaches have a major limitation in that they assume that the objects in the scene are rigid (10; 32), which does not take into account deformable objects. Moreover, these methods cannot recover dense depth maps.

Deep learning (DL) is considered as a solution to overcome these limitations. Nevertheless, it requires training on very large datasets. To address this, self-supervised DL can be used to generate a supervisory signal from unlabeled data by exploiting the underlying structure on it through image reconstruction (9). In addition, these models can be trained solely with monocular RGB sequences.

Self-supervised DL models for monocular depth estimation consist on performing feature extraction from a target frame and nearby ones. Then, a depth map from the target frame is predicted, as well as the relative pose changes from each pair of consecutive frames. Using this information, a reconstruction of the target frame is generated and used as the supervisory signal.

Even though these models show good performance on predicting depth maps from unlabeled data, they also show dependency to the intrinsic camera calibration parameters used during training, leading to a degradation of the results when applied to sequences recorded with different cameras (23; 10).

To help overcome this limitation, we propose a method to increase the generalisation ability of the models to different cameras. We predict the camera that has captured each frame and we include it in the system in an adversarial manner, as we can see in Figure 1. By doing this, we achieve a feature extraction process that encourages invariance to changes in the camera intrinsic parameters.

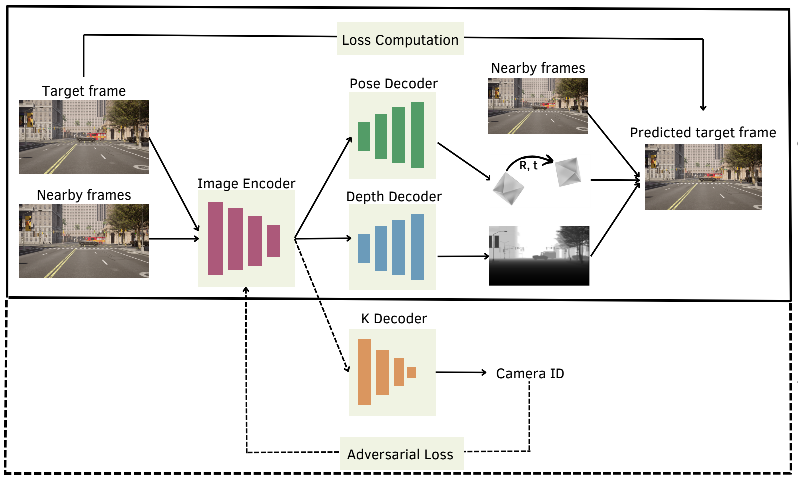


Fig. 1: Overview of self-supervised monocular depth estimation and the inclusion of adversarial training as our proposal. First, feature extraction is performed for a target frame and nearby ones (Image Encoder). Second, the pose change between each pair of consecutive frames is obtained (Pose Decoder), as well as the depth map of the target frame (Depth Decoder). Then, a reconstruction of the target frame is generated and used as the supervisory signal. Proposal: adding a classifier for the camera (K Decoder) in an adversarial manner to achieve features invariant to camera changes.

To support our research, we need a dataset containing identical sequences recorded by different cameras. As there is currently no public dataset that meets these characteristics, we have created a synthetic public dataset in which we can compare the same sequence taken by different cameras, thus allowing a fair comparison for the models.

2 Related Work

In this section, we are going to describe the concept of self-supervised monocular depth estimation and its main restrictions. Then, we particularise on the limitations introduced by the camera intrinsics.

2.1 Self-Supervised Monocular Depth Estimation

Self-supervised monocular depth estimation allows inference of depth at the pixel level from images captured by a single camera, without the need of annotated data. It employs deep Convolutional Neural Networks (CNNs) to learn the relationship between input images and their corresponding depth maps. These models are trained in a self-supervised manner, comparing each image with a reconstruction of it as the supervisory signal.

2.2 Restrictions of Existing Methods

Restrictions for self-supervised monocular depth estimation methods can be categorized into three main areas: scale, photometric consistency, and camera model.

Scale Scale consistency in depth estimation is assumed to ensure accurate results. However, it is inherently ambiguous in monocular depth estimation as these methods use a single image of the scene, thus posing a major challenge. Different approaches have been proposed to address this challenge, such as minimizing the difference between target and source depth (20), using inverse depth terms (29), or aligning depth estimations with sparse depth points (31).

Photometric Consistency Self-supervised monocular depth estimation assumes static scenes for image reconstruction, where all pixels move relative to the camera's ego-motion. This assumption simplifies the depth estimation process but is limiting in dynamic scenes with moving objects or occlusions. Various techniques have been proposed to address this, as using a pose-explainability network (32), optical flow-based masking (28; 17; 12), depth difference measurements for occlusion handling (10), and weak supervision based on epipolar geometry (19).

Camera Existing methods in this task assume constant intrinsic camera parameters, during both training and testing. While this is usually true in controlled environments, it often leads to a decrease in the generalization capability when using a different camera, thus limiting the applicability.

2.3 Approaches to Address Camera Restrictions

Several methods can be applied to deal with camera restrictions, which are mainly based on: adding camera information explicitly, continuous learning, fusion of multi-view geometry and DL or mixture of datasets.

Adding Camera Information Explicitly One approach to address camera assumptions is by explicitly incorporating camera information into the learning process. This can be done: at the data level, by using synthetic and diverse datasets that simulate different camera parameters and situations (23); and at the architecture level, by making changes to neural networks to adapt the convolution filters and introduce attention layers based on camera calibration (6). These modifications improve the consistency and accuracy of depth estimation.

Continuous Learning Continuous learning involves updating and adapting the model as new data is collected to overcome the limitations of static camera assumptions. This can be done, for example, by employing Bayesian inference and scene-independent geometric computations (13), which incorporate both optical flow and depth estimations. This approach enables fast adaptation to unseen scenes, but it requires online optimization and a big hardware support.

Fusion of Multi-View Geometry and DL The fusion of multi-view geometry and DL (24) combines the spatial structure of the scene from multiple images with deep neural networks. By leveraging information from different viewpoints, more accurate and consistent depth estimations can be obtained. This approach uses traditional multi-view geometry methods to resolve camera variability issues and achieves end-to-end learning of 3D geometry.

Mixture of Datasets The mixture of datasets strategy aims to address camera assumptions by combining diverse datasets, which can be mainly done by Multi-objective learning or through adversarial training. Multi-objective learning involves training the model using multiple datasets with variations in capture conditions, thus improving the generalization and accuracy of depth estimations (16). Alternatively, adversarial training introduces a discriminator to identify the camera source to generate robust depth estimations across different cameras. This approach enables the system to learn invariant features and patterns with respect to the cameras, enhancing the generalization ability to changes in the intrinsic calibration parameters.

This paper focuses its research line on the use of adversarial training to mitigate the dependence of self-supervised monocular depth estimation models to changes in the intrinsic camera parameters.

3 Method

In order to have a reference of the validity of our work, we implement a baseline algorithm. Then, we describe our proposal, that is based on the inclusion of adversarial training to achieve feature invariance to the camera intrinsics.

3.1 Baseline Algorithm

Network Architecture The architecture of this algorithm can be seen in Figure 2. Firstly, there is an encoder which performs feature extraction. We follow the literature (9) and use a triplet of frames to ensure consistency: I_{t-1}, I_t and I_{t+1}. Subsequently, these features are used as input in two different decoders; one estimates the depth of I_t, D_t, and the other one predicts the pose changes between both pairs (I_t, I_{t-1}) and (I_t, I_{t+1}), representing them with transformation matrices, $T_{t \rightarrow t-1}$ and $T_{t \rightarrow t+1}$ respectively, where $T = [R|t]$, being R a rotation matrix and t a translation vector.

The encoder used is a standard Residual Neural Network, ResNet18 (11). On the other hand, the Depth decoder is a U-Net (18) alike network, as it receives features at different resolution levels. Finally, the Pose decoder is a CNN that predicts rotation and translation from 2 feature vectors concatenated.

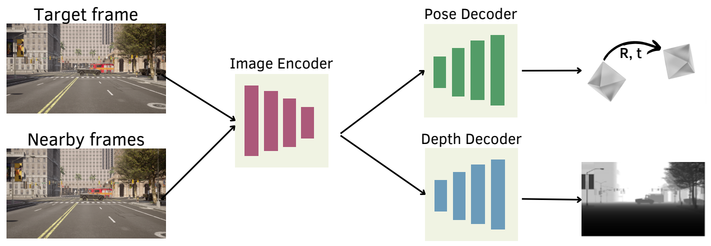


Fig. 2: Baseline Architecture. 'Image Encoder' performs feature extraction from a target frame (I_t) and two nearby ones (I_{t-1} and I_{t+1}). Then, from this features, 'Pose Decoder' predicts the relative pose change between each pair of consecutive frames, while 'Depth Decoder' estimates the depth map of I_t.

Bilinear Warping We generate two reconstructions of the I_t frame, $\hat{\text{I}}_t$, one of them combining I_{t-1}, $T_{t \rightarrow t-1}$, the camera calibration matrix, K, and D_t, and the other one combining I_{t+1}, $T_{t \rightarrow t+1}$, K and D_t.

This computation is done according to the following formula (32):

$$p_a = K T_{t \rightarrow a} D_t K^{-1} p_t, \quad (1)$$

being p_t the pixels of the target frame and p_a the pixels of the corresponding adjacent frame in each case. Intuitively, we first project p_t in the 3D world with K^{-1}. However, as K projects the pixel up to a scale factor, we obtain this pixel in the 3D world through D_t. Then, we use the transformation matrix, $T_{t \rightarrow a}$, which contains the estimated translation vector t and rotation matrix R from I_t to I_a, to transform the pixel to the position of the adjacent frame. Finally, we use K to project it back to the camera dimension, thus obtaining p_a.

Loss Computation The model computes the difference between the target frame and both reconstructions of it as the supervision signal.

This difference is computed by combining the L1 loss (30), the structural similarity index measure, SSIM, (25) and the smoothness loss (22). This combination is done to ensure a better result both numerically and visually (26).

It is expressed as follows:

$$Loss = (1-\alpha)L_1 + \alpha L_{ssim} + \beta L_{smooth}, \quad (2)$$

being $\alpha = 0.85$ and $\beta = 0.01$, as suggested in the literature (9).

Then, we average both losses, the one that extracts $\hat{\text{I}}_t$ from I_{t-1} and the one that uses I_{t+1} instead.

Auto-Masking Stationary Pixels Self-supervised monocular depth estimation operates under the assumption of a moving camera and a static scene. When this assumption breaks down, performance can suffer greatly, even causing 'holes' of infinite depth to appear in the predicted depth maps (15). To mitigate this problem, we use a simple auto-masking method (9) that filters out pixels which do not change appearance from one frame to the next one. This has the effect of letting the network ignore pixels that break photometric consistency assumptions, such as moving objects, occlusions, and even to ignore whole frames when the camera stops moving.

We perform this auto-masking method through the following equation:

$$Loss^{i,j} \quad = \quad min\,(pe(I_t, I_{a \to t})^{i,j},\, pe(I_t, I_a)^{i,j}), \tag{3}$$

where pe is the projection error defined by Equation 2, $I_{a \to t}$ is the reconstruction of I_t from an adjacent frame I_a, and the terms i, j represent the pixel coordinates of the image.

We compare for each pixel of I_t what gives us the smallest error: using its corresponding value in a nearby frame or using its warped value. As this loss is a tensor of the same shape of the images, i.e. provides an error for each RGB pixel, we first compute the mean per channel and, then, global mean. Thus, a single error value per image is obtained.

3.2 Baseline Algorithm + Adversarial training

In this section we describe our main contribution, which is based on adding robustness to the models so that the features extracted from each frame are invariant to changes in the intrinsic parameters of the camera that captured it. To do this, a K Decoder is added into the model architecture, as shown in Figure 3, to classify the camera that has recorded each of the images. The proposed method involves training the architecture in an adversarial manner in which the Image Encoder acts as a feature generator and competes with the K Decoder as a discriminator, hence achieving a feature representation that remains independent of the camera model used. Apart from that, the same warping process is performed with respect to the basline algorithm, as well as the same loss computation for the common part of the networks in Figure 2.

The K Decoder is designed to be small, thus encouraging changes in the features. It consists of convolutional layers followed by LeakyReLU activation functions (27) and a final layer that generates the classifications.

As K is included in an adversarial manner in the algorithm, there are two different optimizers. The first one optimises the Image Encoder, Pose Decoder and Depth Decoder, whose aim is to obtain accurate depth and pose values while simultaneously maximizing the K Decoder loss. By doing this, the features extracted become indistinguishable between different cameras.

This optimization process is done according to the following equation:

$$L_{opt1} \quad = \quad (1-\alpha)L_1 \,+\, \alpha L_{ssim} \,+\, \beta L_{smooth} \,-\, \gamma L_k, \tag{4}$$

being $\alpha = 0.85$ and $\beta = 0.01$, as suggested in the literature (9). We take $\gamma = 0.001$ by experimental procedure, being L_k the Cross Entropy loss function.

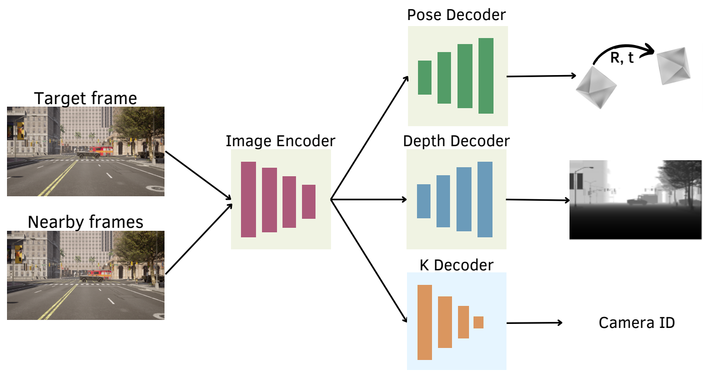


Fig. 3: Adversarial Training Architecture. The baseline algorithm for self-supervised monocular depth estimation contains an image encoder, a pose decoder and a depth decoder, as presented in Figure 2. In this architecture there is an extra decoder, K, that is in charge of classifying the camera that recorded each frame. This decoder is included in an adversarial manner to achieve a feature extraction process invariant to changes in the camera parameters.

On the other hand, the second optimizer improves the prediction of the K decoder enhancing its ability to correctly identify the camera associated with each input image. Its objective is to optimize the network's proficiency in camera classification by:

$$L_{opt2} = L_k. \tag{5}$$

4 Experiments

4.1 Setup

Three models have been trained and evaluated: Base, Mult, and Multiseq-adv, the latter being our proposal.

Base model uses the baseline algorithm explained in Section 3.1 and has been trained with images of equal focal distance. Mult model also uses the Baseline architecture, but it has been trained with images of three different focal distances, hence capturing more camera variability. Finally, Multiseq-adv employs the Adversarial architecture proposed in Section 3.2 to enhance robustness to camera parameter changes, and it has been trained using the same images as Mult model, where three different focal distances are represented. The learning process is described in Algorithm 1.

We have trained all three models with Adam optimizer (lr=1e-5), a batch size of 5, min_depth = 0.1 and max_depth = 100, for 300 epochs. These values have been selected according to the evaluation framework from (9).

Dataset Our method requires a dataset in which the same sequence is available for cameras with different intrinsic parameters. However, since there is no public

```
**Input** : Dataset images, K
**Output:** Improved pose and depth estimators
foreach iteration do
    Randomly select $I_t$, $I_{t-1}$ and $I_{t+1}$;
    Introduce Gaussian noise into the images;
    Extract features from images;
    if Multiseq-adv model then
        Estimate the camera ID;
    end
    Estimate $T_{t \rightarrow t-1}$ and $T_{t \rightarrow t+1}$;
    Estimate $D_t$ and scale it;
    $p_{t-1}^{(1)} = KT_{t \rightarrow t-1} D_t K^{-1} p_t$;
    $p_{t+1}^{(2)} = KT_{t \rightarrow t+1} D_t K^{-1} p_t$;
    $Loss^{i,j} = min\,(pe(I_t, I_{a \rightarrow t})^{i,j},\, pe(I_t, I_a)^{i,j})$;
    Update model parameters ;
end
```

Algorithm 1: Iterative Learning Algorithm.

dataset that meets these conditions, we have generated a custom public dataset (available in (3)) that includes both RGB and depth images using CARLA simulator (4) through Unreal Engine 4 (7). It comprises 6000 synthetic training images distributed across 3 video sequences, each captured by 5 different cameras with Field Of View (FOV) values of 40, 60, 80, 100, and 120. Therefore, we have 30000 images for the training phase.

On the other hand, there are 2 test sequences of 1200 images each, also captured by 5 different cameras, thus obtaining a total of 12000 test images.

It is important to note that Test Set 1 is captured in a complex urban environment with significant presence of buildings and traffic elements, while Test Set 2 takes place in a simpler rural environment, as we can see in Figure 4. As a result, Test Set 1 can be considered more challenging due to its higher complexity, potentially yielding higher error values compared to Test Set 2.

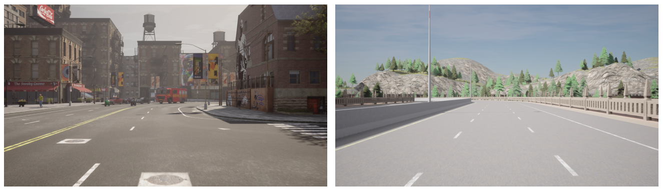


Fig. 4: Images from Test Set 1 and Test Set 2 as urban and rural environments respectively, both belonging to the synthetic custom dataset created to enable comparisons of the same sequence captured by different cameras.

Evaluation Metrics The metrics used to evaluate depth estimation are: Absolute Relative Error, RSE, RMSE, Log Scale Invariant RMSE (5) and Accuracy under a threshold (2).

4.2 Analysis of the results

We validate that -1- models show a degradation of the performance when FOV value changes, and -2- adversarial training mitigates this effect. We evaluate our models with both test sequences of the custom created dataset in terms of depth estimation. Moreover, we also evaluate their performance on KITTI dataset (8).

Synthetic Dataset Results Note that 'Cam' column in all figures and tables represents the camera used to record the test sequence that is being evaluated.

Table 1 shows the results of the models for each camera in Test Set 1.

Table 1: Results of the three models for depth estimation over Test Set 1. (KEY: Base model has been trained only with images taken from Cam 1; Mult model's training involves images from Cam 2, Cam 3 and Cam 4; Multiseq-adv model is our proposal, as it has been trained using images from Cam 2, Cam 3 and Cam 4 and it includes adversarial training for camera generalization).

Cam	**FOV**	**Model**	**Abs Rel**$\downarrow$	**Sq Rel**$\downarrow$	**RMSE**$\downarrow$	**RMSE log**$\downarrow$	$\delta < 1.25\uparrow$	$\delta < 1.25^2\uparrow$	$\delta < 1.25^3\uparrow$
		Base	**0.22**	**3.44**	**10.45**	**0.27**	**0.69**	**0.91**	**0.97**
1	40	Mult	0.28	5.71	12.3	0.33	0.63	0.87	0.94
		Multiseq-adv	0.25	4.75	11.82	0.3	0.66	0.88	0.95
		Base	0.26	**3.18**	**10.26**	0.3	0.6	**0.89**	**0.96**
2	60	Mult	0.28	5.67	11.0	0.33	0.66	0.88	0.94
		Multiseq-adv	**0.24**	3.9	10.48	**0.29**	**0.67**	**0.89**	0.95
		Base	0.3	**3.35**	10.42	0.35	0.51	0.83	**0.95**
3	80	Mult	**0.26**	3.86	**9,47**	**0.31**	**0.69**	0.88	0.94
		Multiseq-adv	**0.26**	3.9	9.72	**0.31**	0.67	**0.89**	**0.95**
		Base	0.37	3.79	10.62	0.41	0.43	0.74	0.89
4	100	Mult	0.26	2.96	**8.46**	0.32	**0.68**	**0.87**	**0.94**
		Multiseq-adv	**0.24**	**2.83**	8.5	**0.31**	**0.68**	**0.87**	**0.94**
		Base	0.39	3.64	11.45	0.48	0.33	0.66	0.84
5	120	Mult	0.25	2.42	8.71	0.34	0.63	0.84	0.93
		Multiseq-adv	**0.23**	**2.32**	**8.63**	**0.32**	**0.65**	**0.86**	**0.94**

Base model exhibits superior performance in Cam 1, suggesting its advantage in maintaining consistency between train and test images regarding intrinsic camera parameters. However, Base error values progressively rise as the camera changes, thus evidencing dependence on the parameters of the training camera.

In contrast, Mult model shows a better consistency in the result as the FOV changes due to the higher variability in its training images. Nevertheless, Multiseq-adv outperforms the other two models in most error measures, also displaying enhanced generalization to the FOV value. In this case, error values remain consistent across different cameras, showcasing the benefits of adversarial training in improving the model's robustness to unseen cameras. Furthermore, Multiseq-adv achieves the best results on Cam 5, which is the only unknown camera for all the models.

Alternatively, Figure 5 presents the depth predictions generated by the models for a specific RGB image captured by the lowest and highest FOV cameras.

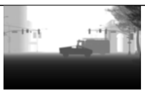
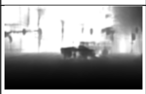
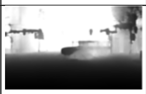
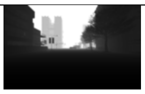
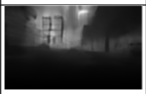
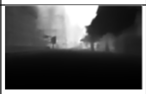
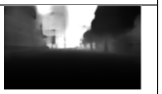
Cam	FOV	RGB	Ground truth	Base	Mult	Multiseq-adv
1	40					
5	120					

Fig. 5: Depth maps extracted from the predictions of the three models for a specific RGB image of Test Set 1. Note that darker pixels correspond to closer distances while brighter pixels are farther. (KEY: Base model has been trained only with images taken from Cam 1; Mult model's training involves images from Cam 2, Cam 3 and Cam 4; Multiseq-adv model is our proposal, as it has been trained using images from Cam 2, Cam 3 and Cam 4 and it includes adversarial training for camera generalization).

As we can observe, Base model performs well on Cam 1 but shows considerable deterioration on Cam 5, resulting in blurred predictions. Alternatively, Mult model improves upon Base on Cam 5 but still exhibits some deterioration, specially on Cam 1. Finally, Multiseq-adv shows reduced blurring effects, outperforming both models for Cam 5 and also Mult model on Cam 1, accurately capturing nearby and further objects on unseen cameras.

Overall, after examining Table 1 and Figure 5, Multiseq-adv model demonstrates the best performance in terms of depth prediction for Test Set 1.

Table 2: Results of the three models for depth estimation over Test Set 2. (KEY: Base model has been trained only with images taken from Cam 1; Mult model's training involves images from Cam 2, Cam 3 and Cam 4; Multiseq-adv model is our proposal, as it has been trained using images from Cam 2, Cam 3 and Cam 4 and it includes adversarial training for camera generalization).

Cam	FOV	Model	Abs Rel↓	Sq Rel↓	RMSE↓	RMSE log↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑
1	40	Base	**0.16**	**2.26**	**9.7**	**0.22**	0.74	**0.92**	**0.98**
		Mult	0.21	4.15	11.69	0.26	0.75	0.89	0.96
		Multiseq-adv	0.18	3.19	10.95	0.23	**0.79**	**0.92**	**0.98**
2	60	Base	0.18	**2.15**	10.02	0.26	0.72	0.89	0.96
		Mult	0.18	3.47	10.44	0.26	0.78	0.89	0.96
		Multiseq-adv	**0.16**	2.98	**9.86**	**0.23**	**0.8**	**0.92**	**0.97**
3	80	Base	0.2	2.57	10.81	0.34	0.7	0.85	0.91
		Mult	0.17	2.53	8.69	0.25	0.79	0.91	**0.97**
		Multiseq-adv	**0.16**	**2.45**	**8.6**	**0.24**	**0.8**	**0.92**	**0.97**
4	100	Base	0.22	2.93	11.56	0.42	0.67	0.81	0.88
		Mult	0.2	2.46	8.44	0.3	0.74	0.86	0.92
		Multiseq-adv	**0.16**	**2.03**	**8.22**	**0.26**	**0.79**	**0.91**	**0.96**
5	120	Base	0.25	3.9	13.47	0.54	0.62	0.77	0.83
		Mult	0.21	2.36	9.29	0.34	0.71	0.84	0.9
		Multiseq-adv	**0.17**	**2.06**	**9.11**	**0.3**	**0.75**	**0.88**	**0.93**

Table 2 shows the error values obtained by each model in Test Set 2. In general, there is a significant reduction in errors compared to those achieved on Test Set 1, due to the lower complexity of this sequence.

Similar patterns to those from the Test Set 1 results are observed: Base performs the best on Cam 1, while Multiseq-adv outperforms the rest of the cases, showing lower errors on both seen and unseen cameras.

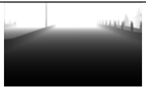
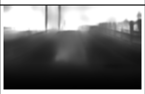
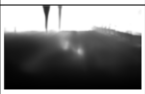
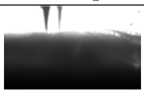
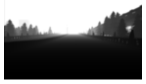
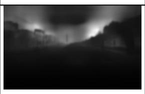
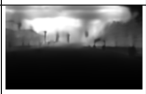
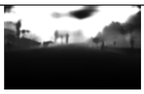
Cam	**FOV**	**RGB**	**Ground truth**	**Base**	**Mult**	**Multiseq-adv**
1	40					
5	120					

Fig. 6: Depth maps extracted from the predictions of the three models for a specific RGB image of Test Set 2. Note that darker pixels correspond to closer distances while brighter pixels are farther. (KEY: Base model has been trained only with images taken from Cam 1; Mult model's training involves images from Cam 2, Cam 3 and Cam 4; Multiseq-adv model is our proposal, as it has been trained using images from Cam 2, Cam 3 and Cam 4 and it includes adversarial training for camera generalization).

Figure 6 displays the predictions of the models over an RGB image of Test Set 2. Base model performs well for Cam 1 but gets deteriorated results on Cam 5. Mult model shows a slight improvement but still struggles with generalization to unseen cameras, especially Cam 5. Finally, Multiseq-adv model provides the best depth predictions, even though some artifacts are observed in the top of the Cam 5 image, potentially caused by the presence of clouds.

KITTI Dataset Results We have evaluated the monocular depth estimation capacity of the three models created using images from the KITTI dataset (8).

In this case, it should be noted that the model taken as a reference, Base, is a model that uses the Baseline algorithm detailed in Section 3.1 and that has been trained only with images taken by Cam 5, i.e. with FOV = 120.

This change in the training design for Base model is due to the fact that the KITTI dataset images have a focal length very close to 40, which is the one used by Cam 1 in the synthetic dataset, so using a model trained only with images from this camera would not be representative for this study.

Table 3: Results of the three models for depth estimation over KITTI evaluation dataset. (KEY: Base model has been trained only with images taken from Cam 5; Mult model's training involves images from Cam 2, Cam 3 and Cam 4; Multiseq-adv model is our proposal, as it has been trained using images from Cam 2, Cam 3 and Cam 4 and it includes adversarial training for camera generalization).

Model	**Abs Rel**$\downarrow$	**Sq Rel**$\downarrow$	**RMSE**$\downarrow$	**RMSE log**$\downarrow$	$\delta < 1.25\uparrow$	$\delta < 1.25^2\uparrow$	$\delta < 1.25^3\uparrow$
Base	0.579	8.338	11.039	0.641	0.235	0.474	0.678
Mult	0.442	4.525	9.500	0.555	0.273	0.541	0.749
Multiseq-adv	**0.414**	**4.200**	**9.373**	**0.516**	**0.283**	**0.575**	**0.799**

As we can see in Table 3, the model that has been trained with images from a single camera, Base, gets significantly the worst results. On the other hand, Mult model improves considerably the error values, suggesting a higher generalisation capacity as it has been trained with three different cameras. Finally, Multiseq-adv model provides the best results, demonstrating the potential benefits of the use of adversarial training to mitigate the dependence of these models on the camera used during training.

On the other hand, Figure 7 shows the visual predictions of the three models for an image of the KITTI evaluation set. As we can see, the depth map obtained with Base model is quite noisy, as we cannot intuit the structures present in the image. Mult model improves these results detecting parts of the car that is in front of the camera, although it still does not provide a good view of the depth information. Finally, Multiseq-adv model allows us to intuit the correct depth along the road, as well as the car and other structures in the frame.

Although Multiseq-adv model achieves the best predictions both numerically and visually, these is also a domain shift between the synthetic training images and the real evaluation ones. Nevertheless, it is demonstrated in this study that the inclusion of adversarial training is a potential improvement for the generalisation of models to unseen cameras in this task.

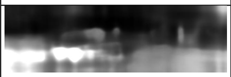
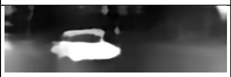
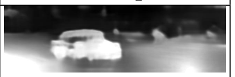
RGB	**Base**	**Mult**	**Multiseq-adv**
			

Fig. 7: Depth maps extracted from the predictions of the three models for a specific RGB image of KITTI Dataset. Note that the colors in the predictions of the models have been inverted for a better visualization i.e. closer pixels are whiter while further pixels are darker. (KEY: Base model has been trained only with images taken from Cam 5; Mult model's training involves images from Cam 2, Cam 3 and Cam 4; Multiseq-adv model is our proposal, as it has been trained using images from Cam 2, Cam 3 and Cam 4 and it includes adversarial training for camera generalization).

5 Conclusion/Discussion

In this work, the challenge of mitigating camera dependence on self-supervised monocular depth estimation models has been addressed. For this purpose, a synthetic public dataset has been created that allows a fair analysis of the effect that changes on the intrinsic camera parameters have on the models, which has been crucial for the development and evaluation of the project.

It has been observed evidence of degradation in the results when changing intrinsic camera parameters on models that are trained solely on one camera, highlighting the importance of achieving robustness of the models to these variations. Furthermore, the results obtained suggest improved performance when incorporating adversarial training into the model architecture.

In future work, we would like to evaluate the errors associated to the extracted pose values both on the created synthetic dataset and on real images.

Acknowledgments. This work is part of the HVD (PID2021-125051OB-I00), IND2020/TIC-17515 and SEGA-CV (TED2021-131643A-I00) projects, funded by the Ministerio de Ciencia e Innovacion of the Spanish Government.

References

[1] Antensteiner, D., Štolc, S., Huber-Mörk, R.: Depth estimation with light field and photometric stereo data using energy minimization. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 21st Iberoamerican Congress, CIARP 2016, Lima, Peru, November 8–11, 2016, Proceedings 21. pp. 175–183. Springer (2017)

[2] Cadena, C., Latif, Y., Reid, I.D.: Measuring the performance of single image depth estimation methods. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4150–4157. IEEE (2016)

[3] Cecilia Diana, Juan Ignacio Bravo, J.M.G.J.B.: Unsyn-mf dataset: Unified synthetic multiple fov. `http://www-vpu.eps.uam.es/publications/UnSyn-MF_Dataset/`, 2023

[4] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning. vol. 78, pp. 1–16 (2017)

[5] Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. Advances in neural information processing systems **27** (2014)

[6] Facil, J.M., Ummenhofer, B., Zhou, H., Montesano, L., Brox, T., Civera, J.: Cam-convs: Camera-aware multi-scale convolutions for single-view depth. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11826–11835 (2019)

[7] Games, E.: Unreal engine 4. `https://www.unrealengine.com`, 2019

[8] Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)

[9] Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3828–3838 (2019)

[10] Gordon, A., Li, H., Jonschkowski, R., Angelova, A.: Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8977–8986 (2019)

[11] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

[12] Li, H., Gordon, A., Zhao, H., Casser, V., Angelova, A.: Unsupervised monocular depth learning in dynamic scenes. In: Conference on Robot Learning. pp. 1908–1917. PMLR (2021)

[13] Li, S., Wu, X., Cao, Y., Zha, H.: Generalizing to the open world: Deep visual odometry with online adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13184–13193 (2021)

[14] Liu, X., Sinha, A., Ishii, M., Hager, G.D., Reiter, A., Taylor, R.H., Unberath, M.: Dense depth estimation in monocular endoscopy with self-supervised learning methods. IEEE transactions on medical imaging **39**(5), 1438–1447 (2019)

[15] Luo, C., Yang, Z., Wang, P., Wang, Y., Xu, W., Nevatia, R., Yuille, A.: Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. IEEE transactions on pattern analysis and machine intelligence **42**(10), 2624–2641 (2019)

[16] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE transactions on pattern analysis and machine intelligence **44**(3), 1623–1637 (2020)

[17] 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12240–12249 (2019)

[18] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)

[19] Shen, T., Luo, Z., Zhou, L., Deng, H., Zhang, R., Fang, T., Quan, L.: Beyond photometric loss for self-supervised ego-motion estimation. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 6359–6365. IEEE (2019)

[20] Tang, J., Ambrus, R., Guizilini, V., Pillai, S., Kim, H., Jensfelt, P., Gaidon, A.: Self-supervised 3d keypoint learning for ego-motion estimation. In: Conference on Robot Learning. pp. 2085–2103. PMLR (2021)

[21] Vieira, A.W., Nascimento, E.R., Oliveira, G.L., Liu, Z., Campos, M.F.: Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 17th Iberoamerican Congress, CIARP 2012, Buenos Aires, Argentina, September 3-6, 2012. Proceedings 17. pp. 252–259. Springer (2012)

[22] Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S.: Learning depth from monocular videos using direct methods. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2022–2030 (2018)

[23] Wang, W., Hu, Y., Scherer, S.: Tartanvo: A generalizable learning-based vo. In: Conference on Robot Learning. pp. 1761–1772. PMLR (2021)

[24] Wang, Y., Luo, K., Chen, Z., Ju, L., Guan, T.: Deepfusion: A simple way to improve traditional multi-view stereo methods using deep learning. Knowledge-Based Systems **221**, 106968 (2021)

[25] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

[26] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

[27] Xu, J., Li, Z., Du, B., Zhang, M., Liu, J.: Reluplex made more practical: Leaky relu. In: 2020 IEEE Symposium on Computers and communications (ISCC). pp. 1–7. IEEE (2020)

[28] Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1983–1992 (2018)

[29] Zhan, H., Weerasekera, C.S., Garg, R., Reid, I.: Self-supervised learning for single view depth and surface normal estimation. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 4811–4817. IEEE (2019)

[30] Zhao, H., Gallo, O., Frosio, I., Kautz, J.: Loss functions for image restoration with neural networks. IEEE Transactions on computational imaging **3**(1), 47–57 (2016)

[31] Zhao, W., Liu, S., Shu, Y., Liu, Y.J.: Towards better generalization: Joint depth-pose learning without posenet. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9151–9161 (2020)

[32] Zhou, T., Brown, M., Snavely, N., Lowe, D.G.: Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1851–1858 (2017)