Rebalancing gradient to improve self-supervised co-training of depth, odometry and optical flow predictions

Marwane Hariat, Antoine Manzanera, David Filliat

Rebalancing gradient to improve self-supervised co-training of depth, odometry and optical flow predictions

Marwane Hariat[1], Antoine Manzanera[1], David Filliat [1, 2]

[1]U2IS, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France
[2]INRIA FLOWERS
{marwane.hariat, antoine.manzanera, david.filliat}@ensta-paris.fr

Abstract

*We present **CoopNet**, an approach that improves the cooperation of co-trained networks by dynamically adapting the apportionment of gradient, to ensure equitable learning progress. It is applied to motion-aware self-supervised prediction of depth maps, by introducing a new hybrid loss, based on a distribution model of photo-metric reconstruction errors made by, on the one hand the depth + odometry paired networks, and on the other hand the optical flow network. This model essentially assumes that the pixels from moving objects (that must be discarded for training depth and odometry), correspond to those where the two reconstructions strongly disagree. We justify this model by theoretical considerations and experimental evidences. A comparative evaluation on KITTI and CityScapes datasets shows that CoopNet improves or is comparable to the state-of-the-art in depth, odometry and optical flow predictions. Our code is available here: **https://github.com/mhariat/CoopNet**.*

1. Introduction

Humans are amazingly competent at inferring 3D structures of a scene from monocular images. This ability is acquired from the very first day of their lives, when infants learn to understand the geometric properties of their environment and its regularities. Then, they learn to interpret 2D images as 3D scenes by making their visual perception consistent with their inner understanding of the world.

This mechanism can be emulated with self-supervised learning. To do so, intermediate visual tasks such as depth, optical flow and camera pose estimations are performed by deep neural networks to reproduce a scene from different viewpoints. This whole pipeline can be trained in an end-to-end manner, using the consistency between the observed images and synthesised views as the supervisory signal [43, 8]. It will only perform well if the intermediate estimations are close enough to their ground truth.

Self-supervision has the advantage that the underlying process producing the visual tasks is more robust and can generalise better to new unknown data compared to direct supervision setting with available ground-truth data [6], where it can be hard to avoid over-fitting due to the lack of constraints. Self-supervised networks have to develop both geometric and contextual reasoning skills, attributes that are far less dataset dependent, to correct the inconsistencies of the view reconstruction. Self-supervision, on top of its healthy training conditions, brings a lot of flexibility. It allows to learn from a much larger scope of data as ground-truth data are not required. Fine-tuning can additionally be stacked within an incremental learning strategy with only minor manageable time and memory increases.

In our work, we are particularly interested in depth estimation. However, other intermediate visual estimation such as odometry or optical flow will also be considered and assessed with appropriate metrics. We will use only monocular images in order to force visual task estimations to leverage contextual information as much as possible to solve ambiguities, and because it only requires a cheap and ubiquitous monocular camera.

The view synthesis training strategy requires to face situations such as texture changes, light reflections and occlusions amongst others. But the most challenging issue would certainly be to deal with moving objects, since the warping transformation assumes the scene to be static, and moving regions can pollute the learning process with misleading high reconstruction errors.

Our contribution is to propose a new strategy relying on the cooperation between the Optical Flow, the Depth and the Pose networks during the learning process. Basically, regions for which these networks disagree on their view

syntheses are removed from the training samples when necessary. Our completely self-supervised training strategy is assessed on KITTI [9] and Cityscapes [4]. Although simple, our method outperforms the current state-of-the art unsupervised training strategies dealing with moving objects by a substantial margin. It also competes with methods that make use of semantic information coming from off-the-shelf algorithms.

2. Related Work

Self-supervised Learning framework. Recently, many research works on unsupervised monocular depth prediction have emerged with the willingness to reduce the gap with fully-supervised methods. The principle is based on the warping image transformation procedure [8, 17]. A target view at time t is reconstructed from a source view at time s of the same scene by calculating a warped image $\hat{I}_s$. The chosen sources timestamps s are surrounding the target one t, and generally set to $\left\{t-1, t+1\right\}$. The supervision signal used to train the neural network is:

$$\mathcal{L} = \sum_{\sigma} \sum_{s} \sum_{p_\sigma} \Phi\Big(I_t^\sigma(p_\sigma), \hat{I_s^\sigma}(p_\sigma)\Big) \tag{1}$$

With the *photo-metric error function* $\mathbf{\Phi}$ defined as:

$$\Phi(x,y) = \alpha \frac{1 - SSIM(x,y)}{2} + (1-\alpha)|x-y| \tag{2}$$

where *SSIM* is the structural-similarity [38] and σ is a scale index, since intermediate downscale estimations are also considered in the process to address the *gradient locality problem* caused by the bilinear interpolation [17]. Here I^σ refers to the resized version of image I with a downscale factor of $\frac{1}{2^\sigma}$, and p_σ is the pixel index of images resized at scale σ. In the remainder of the paper, we will drop the σ for better readability.

Now, depending on the objective, the warped image $\hat{I}_s$ can be obtained in two different ways. One introduced by [43] and using the combination of a depth network $\mathcal{D}_\theta$ and a camera pose network $\mathcal{T}_\alpha$, to apply the re-projection formula:

$$\begin{aligned} \hat{I}_s^{\theta,\alpha}(p) &= I_s\left(K\hat{T}_{t\to s}\hat{D}_t(p)K^{-1}p\right) \\ \hat{T}_{t\to s} &= \Big[\mathcal{R}, t\Big] \in \mathcal{SE}\,(3) \end{aligned} \tag{3}$$

where K is the calibration matrix, $\hat{T}$ is the displacement matrix predicted by $\mathcal{T}_\alpha$, and $\hat{D}$ is the depth map predicted by $\mathcal{D}_\theta$.
And another one [31, 42] using an *optical flow* network $\mathcal{F}_\delta$ that directly predicts the displacement vector F_δ:

$$\hat{I}_s^\delta(p) = I_s\left(p + F_\delta(p)\right) \tag{4}$$

Accounting for Motion. Unlike the warping of Eq. 4, which does not care for the origin of motion, the warping of Eq. 3 is no longer valid in moving regions, corresponding to objects that have a displacement on their own. It then makes sense to define the *rigid flow* as the apparent motion flow induced by the camera motion only, under rigid assumption, and calculated as:

$$F_{\theta,\alpha}(p) = K\hat{T}_{t\to s}\hat{D}_t(p)K^{-1}p - p \tag{5}$$

Inside moving object regions, even though depth and pose predictions are correct, the *photo-metric Loss* $\mathbf{\Phi}$ will render wrong values and disrupt the back-propagation process within pose and depth networks. There are two ways to fix this issue. Either one adds a residual correction to $\hat{T}_{t\to s}$ in order to account for potentially moving objects as done by [26] and [11]. Or one can also decide to remove moving object pixels from the loss $\mathcal{L}$ in Eq. 1. This is the strategy that we decide to follow in this paper.

Being able to detect the moving regions of an image is a real challenge. Hence, several methods [2, 24] chose to rely on an off-the-shelf instance segmentation algorithms [14] to get rid of potential moving objects. A strong limitations here is the lack of generalisation. Indeed, these off-the-shelf algorithms are trained on different datasets [29], as the mainstream ones used in monocular depth estimation don't offer enough annotated ground-truth data. Some works tried to overcome this issue either by incorporating the instance segmentation part into the learning pipeline [23], with the off-the-shelf algorithm predictions used as the ground-truth data. Or by using the feature maps of the off-the-shelf network to drive [28, 13] the different visual task networks. In both cases the issue still remains. Besides, these methods are not compliant with our fully self-supervised learning setting. We want to be able to keep learning on the fly and benefit from a large scope of data.

Closely related to our work, [3] warps an image using the two orthogonal ways from Eq. 3 and Eq. 4. The supervision signal is then modified as:

$$\mathcal{L}_{GLNet} = \sum_{s,p} \Psi\left(I_t(p), \hat{I}_s^{\theta,\alpha}(p), \hat{I}_s^{\delta}(p)\right) \tag{6}$$

with $\hat{I}_s^{\theta,\alpha}$ and $\hat{I}_s^{\delta}$ respectively given by Eq. 3 and Eq 4, and Ψ is the *adaptive photo-metric loss*:

$$\Psi(x,y,z) = \min\left(\Phi(x,y), \Phi(x,z)\right) \tag{7}$$

This approach therefore tries to detect moving pixels by the difference between the optical flow and rigid flow predictions, assuming a worse prediction by the rigid flow. Other approaches such as [40, 41, 30, 27, 25, 7] propose to infer a moving object mask using a pre-determined metric

related to the geometric inconsistency between the optical flow and the rigid flow.

Following the idea of [3], our contribution, rather, incorporates a loss-oriented component as part of the decision on moving pixels, while taking care of the different progression speeds between networks to make them benefit from each other in the best way. Additionally, we continuously adapt our decision criterion along the training process using a quantile based approach.

3. Limits of the adaptive photo-metric loss

3.1. Instability

The goal of the *adaptive photo-metric loss* of [3] is to co-train, on the one hand the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ and, on the other hand $\mathcal{F}_\delta$. Since the loss distributes the pixel errors to both networks, according to $\arg\min_{y,z}(\Phi(x,y), \Phi(x,z))$, the networks are actually competing against each other. All things being equal, the optical flow $\mathcal{F}_\delta$ is intrinsically better at learning from the photo-metric loss than $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$, as the re-projection (Eq. 3) is more constrained compared to Eq. 4. We will call this property the *intrinsic bias* throughout the paper. This unbalanced learning capacity between the two contestants is worsen over the training epochs. Indeed, the $\mathbf{\Psi}$ operator splits the set of pixels in two parts based on the sign of the random variable $\mathbf{\Delta}$ defined as:

$$\Delta(p) = \Phi\Big(I_t(p), \hat{I}_s^{\theta,\alpha}(p)\Big) - \Phi\Big(I_t(p), \hat{I}_s^{\delta}(p)\Big) \quad (8)$$

The probability density function f_Δ is approximately Gaussian. In [3], pixels for which $\mathbf{\Delta}$ has non-zero negative values are used to train the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$, whereas the rest of the pixels train $\mathcal{F}_\delta$.

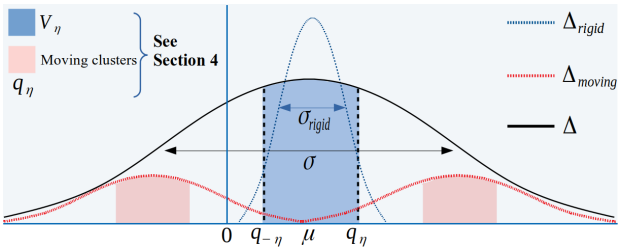


Figure 1: Density models of Δ used in our work, for all the pixels (black), rigid pixels (blue dashed), and mobile pixels (red dashed). This is the result of the statistical analysis of Δ on all the images of KITTI and highlights the intrinsic bias of the Gaussian distribution, the moving pixels following a bimodal distribution centred on both sides of the tails and the rigid pixels located around the mean value. Note that since rigid pixels are the vast majority, $\mu_{\text{rigid}} \approxeq \mu$.

Over the training iterations, $\mathcal{F}_\delta$ takes advantage of its better learning abilities over $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$, shifting f_Δ to the right as shown in Fig. 1, thus creating an imbalance on the number of pixels allocated to each contestant. It benefits the optical flow network, which gets even better at the expense of the depth and pose networks. The resulting sequence of mean values $(\mu_n = \mathbb{E}[\Delta_n])_{n\in\mathbb{N}}$, where n refers to the training iteration index, has an upward trend that needs to be kept under control to avoid a degenerative state where the optical flow is too good and prevents the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ from learning anything. The criterion used to study stability is the convergence of the sequence $\big(\theta_n = 1/\mathbb{P}(\Delta_n < 0)\big)_{n\in\mathbb{N}}$, with $\mathbb{P}$ the probability measure. We provide in supplementary material a proof that the operator $\mathbf{\Psi}$ can make θ_n diverge if the intrinsic power of $\mathcal{F}_\delta$ is not taken care of. Practically, the procedure is very sensitive to small changes, especially when it advantages the optical flow. For the same hyper-parameter settings, the depth network can, depending on the initialized weights, either give good predictions or produce bad map estimations as displayed in Fig. 2.

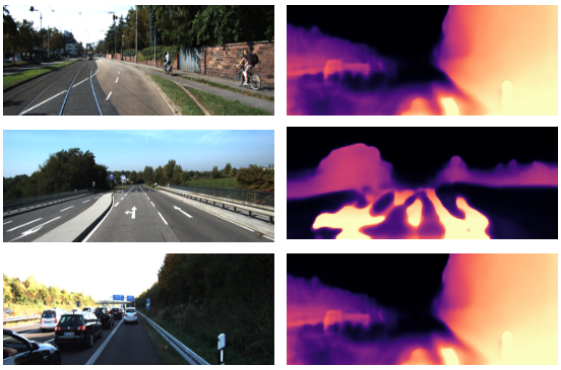


Figure 2: Degenerate cases with black stains (corresponding to infinite depths) spreading all over the image.

3.2. Fundamental issue

As illustrated in Fig. 1, values of moving pixels $\mathbf{\Delta}_{\text{moving}}$ are particularly found in the tails of the distribution. Values in the right part of the tail are due to the systemic inability of the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ to account for any moving displacement. Values in the left part of the tail often corresponds to moving objects for which the optical flow faces smoothing issues as pictured in Fig. 3, rather than a much better prediction of the depth and pose networks. Together, they are responsible for a great part of the variation of Δ and thus lead to $\sigma_{\text{rigid}} < \sigma$. Values of rigid pixels $\mathbf{\Delta}_{\text{rigid}}$, rather, are mostly located in a close neighbourhood of μ. Intuitively, both the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ and $\mathcal{F}_\delta$ ***have a consistent understanding*** for static regions of a scene. Hence, the resulting values taken by $\mathbf{\Delta}$ are more stationary and fairly close, neglecting the intrinsic bias.

As mentioned previously, the idea of [3] is to consider pixels p for which $\Delta(p)$ has negative values to train the pair

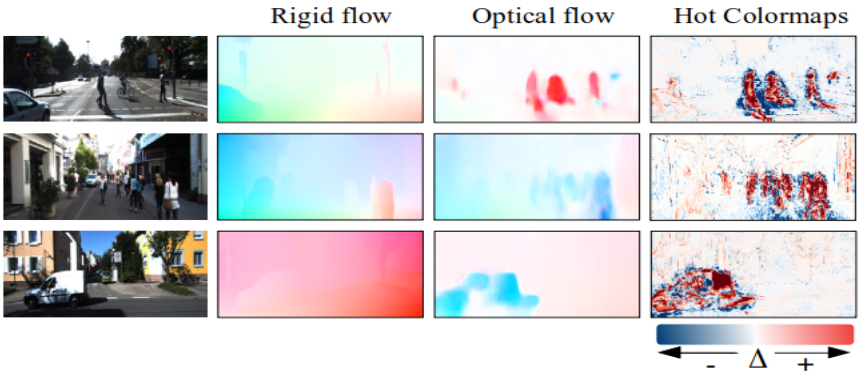


Figure 3: Smoothing issue around moving objects. The *Hot Colormaps* images aim at representing both the sign and absolute value of the Δ function (see colour bar).

$(\mathcal{D}_\theta, \mathcal{T}_\alpha)$. However, by doing so, not only does it throw away a substantial number of rigid pixels, as the interval $[\mu - \sigma_{\text{rigid}}, \mu + \sigma_{\text{rigid}}]$, which covers the values of about 68% of the rigid pixels, satisfies:

$$\begin{aligned}]-\infty, 0[\cap [\mu, \mu + \sigma_{\text{rigid}}] &= \emptyset \\ ]-\infty, 0[\cap [\mu - \sigma_{\text{rigid}}, \mu] &= \emptyset \text{ if } \sigma_{\text{rigid}} < \mu \end{aligned} \tag{9}$$

but more importantly, it wrongly takes into account the moving pixels in the *left* part of the tails (see Fig. 1).

4. Method

The main purpose of our work is to offer a healthier learning protocol to co-train $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ and $\mathcal{F}_\delta$. Inspired by the different issues raised in the previous section we propose a ***Quantile Based Split*** of the probability density function f_Δ in order to:

- Only train the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ on a tight neighbourhood of μ, and stop considering pixels with values in the left tail, thus focusing better on static pixels.
- Train the flow on the whole set of pixels with a larger weight on pixels with $\boldsymbol{\Delta}$ values in the tails of the distribution.

The diagram of our method, named **CoopNet**, is given in Fig. 4. Unlike [3], which makes the networks compete against one another, our approach is based on ***cooperation***. In the same spirit as ***teacher - student techniques***, the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ wait for the approval of the stronger network $\mathcal{F}_\delta$ to select pixels to be trained on. If the optical flow $\mathcal{F}_\delta$ agrees with the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ on the displacement of a given pixel p, then this pixel p can safely be considered as rigid and used to feed the loss $\mathcal{L}_{\theta,\alpha}$ defined in Eq. 10 below.

4.1. Description

For $\eta \in [0, 0.5]$, let us denote q_η the $(0.5 + \eta)$-quantile of the probability density function f_Δ. Let us also define $\mathcal{V}_\eta = [q_{-\eta}, q_\eta]$ a neighbourhood of μ.

As said in the previous section, most of the rigid pixels have a Δ value close to μ. That's why in our approach, the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ is only trained on pixels belonging to $\Delta^{-1}(\mathcal{V}_\eta)$ (see Fig. 1), with Δ^{-1} the inverse image. The larger the interval $\mathcal{V}_\eta$, the closer to the tails and the more likely the pollution by large absolute values $|\boldsymbol{\Delta}(\boldsymbol{p})|$ of moving pixels, which is not desirable. On the contrary, a small $\mathcal{V}_\eta$ will filter many pixels and the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ may not learn anything, as the back-propagation needs enough samples to work. Hence, the hyper-parameter η has to be adjusted to find the best trade-off.

Different training strategies were experimented to find the best way to train the optical flow network, and we found out that learning from all the pixels was the most effective manner, with a weighted sum advantaging moving pixels. Pixels p corresponding to values $\Delta(p)$ in the tails, specifically the η-quantile and the $(1-\eta)$-quantile have more weights (Eq. 11).

These two ideas lead to a split of the loss $\mathcal{L}$ (eq 1), into two terms, as follows:

$$\begin{aligned} \mathcal{L}_\delta &= \sum_{p \in \mathcal{P}} w(p)\, \Phi\Big(I_t(p), \hat{I}_s^\delta(p)\Big) \\ \mathcal{L}_{\theta,\alpha} &= \sum_{p \in \Delta^{-1}(\mathcal{V}_\eta)} \Phi\Big(I_t(p), \hat{I}_s^{\theta,\alpha}(p)\Big) \\ \mathcal{L}_{\text{CoopNet}} &= \mathcal{L}_{\theta,\alpha} + \mathcal{L}_\delta \end{aligned} \tag{10}$$

with $\mathcal{P}$ the set of all pixels and w defined as:

$$w(p) = \begin{cases} \frac{|\mathcal{P}|}{|\Delta^{-1}(\Gamma)|} & \text{if } p \in \Delta^{-1}(\Gamma) \\ \frac{|\mathcal{P}|}{|\Delta^{-1}(\overline{\Gamma})|} & \text{otherwise} \end{cases} \tag{11}$$

$$\Gamma =]-\infty, q_{-\eta}[\cup]q_\eta, +\infty[$$

The losses $\mathcal{L}_{\theta,\alpha}$ and $\mathcal{L}_\delta$ are used to train respectively the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ and $\mathcal{F}_\delta$.

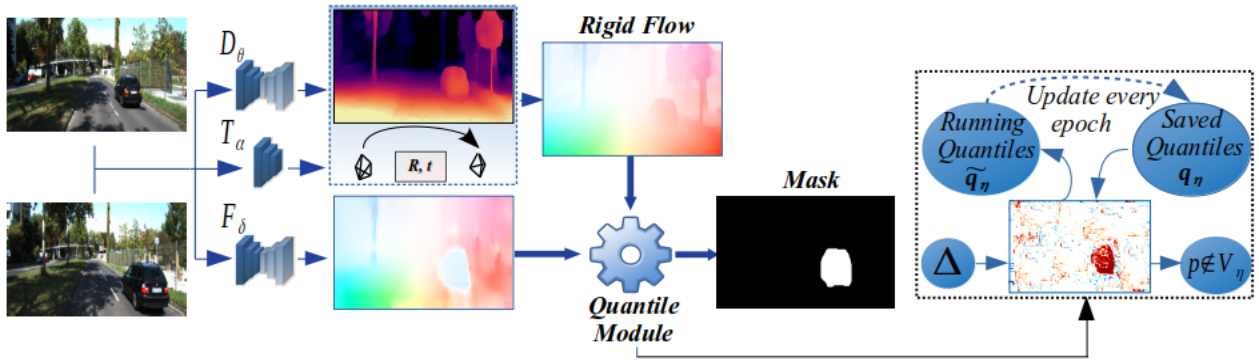


Figure 4: Diagram depicting **CoopNet**. The ***Quantile Module*** takes as input the rigid flow inferred by the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ and the flow produced by $\mathcal{F}_\delta$ to compute $\boldsymbol{\Delta}$. The running values $(\widetilde{q_{-\eta}}, \widetilde{q_\eta})$ are updated with the $\boldsymbol{P^2}$ algorithm [18] to be used at the next epoch. The current values $(q_{-\eta}, q_\eta)$ determine the neighbourhood $\boldsymbol{\mathcal{V}_\eta}$ to induce a mask map $\{\boldsymbol{p \notin \mathcal{V}_\eta}\}$

.

4.2. Advantages over the adaptive photo-metric loss

Training the optical flow on a different set, the complement $\overline{\mathcal{V}_\eta}$ of the one used by the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ for instance, as done by [3], would be sub-optimal. The performances of the optical flow network on rigid regions would be very poor, close to a random prediction. As a consequence, rigid pixels p_{rigid} would have significant negative $\boldsymbol{\Delta}\,(\boldsymbol{p}_{\text{rigid}})$ values and would be mixed with the large negative values of moving pixels that $\mathcal{F}_\delta$ failed to predict correctly. Although the most important is to distinguish rigid pixels from moving pixels correctly predicted by $\mathcal{F}_\delta$ which are in much greater numbers, this is not ideal. With our approach, rather, the probability density function f_Δ has a clumped dispersion pattern as illustrated in Fig. 1, with three clusters sharply delimited. And the intersection between the rigid cluster and each of the two other moving ones is greatly limited. Keeping all that has been said so far in mind, one can legitimately wonder how the distribution of $\boldsymbol{\Delta}$ can stay clamped with three clusters in the approach of [3], as stated in the previous section. The optical flow is trained on $\Delta^{-1}\left(\right] - \infty, 0 \left[\right)$ a subset which is, luckily, composed of enough rigid pixels for $\mathcal{F}_\delta$ to be decent on the static regions. Unfortunately this property is not taken advantage of thereafter.

Finally, let us define:

$$\begin{aligned} L_1 &= \mathbb{E}\left[\Phi\left(I_t(p), \hat{I}_s^{\theta,\alpha}(p)\right) \middle| p \in \mathcal{P}, \Delta(p) \in \mathcal{V}_\eta\right] \\ L_2 &= \mathbb{E}\left[\Phi\left(I_t(p), \hat{I}_s^{\theta,\alpha}(p)\right) \middle| p \in \mathcal{P}, \Delta(p) < 0\right] \end{aligned} \tag{12}$$

We give in supplementary material the mathematical proof that $L_1 < L_2$. This inequality demonstrates theoretically the benefits of introducing the neighbourhood $\mathcal{V}_\eta$ over using the sign of $\Delta(p)$ like [3].

4.3. Regularisation set

We observe experimentally that too many moving pixels can still pollute the neighbourhood $\mathcal{V}_\eta$, even when the hyper-parameter η is set to be very selective. This is due to the well known weakness of the *photo-metric function* Φ (Eq. 4): Because Φ compares images based on colour similarities, it has difficulties to be discriminative in homogeneous areas. Hence, although the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ and $\mathcal{F}_\delta$ disagree on the displacement to be made to warp a moving pixel p, the value $\Delta(p)$ might still be similar to the ones taken by rigid pixels and thus fall into $\mathcal{V}_\eta$. For this reason, we propose to add a new constraint on the agreement of both networks on the pixels displacement.

We introduce Δ_{flow} defined as:

$$\Delta_{\text{flow}}(p) = \frac{F_{\theta,\alpha}(p) - F_\delta(p)}{\left\|F_{\theta,\alpha}(p)\right\|_2 + \left\|F_\delta(p)\right\|_2} \tag{13}$$

where $F_{\theta,\alpha}(p)$ is the image displacement of a pixel produced by the image warping. As for Δ, the closer the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ and $\mathcal{F}_\delta$, the smaller Δ_{flow}. However, the flow values $F_{\theta,\alpha}(p)$ and $F_\delta(p)$ are vectors with two components, then Δ_{flow} is a 2d random vector. As the flow value has a strong dependency on the position in the image (close pixels tend to have higher flow magnitudes than far pixels), a normalisation term is added in the denominator in Eq. 13. The random variable Δ_{flow} doesn't take into account colour intensities and is thus insensitive to the homogeneous issue raised previously. Besides, it has no intrinsic bias which exists with Δ because of the operator Φ.

Finally, the neighbourhood $\mathcal{V}_\eta$ chosen to compute $\mathcal{L}_{\theta,\alpha}$ in Eq. 10 can be replaced by $\mathcal{V}$:

$$\begin{aligned} \mathcal{V} &= \mathcal{V}_\eta \cap \mathcal{V}_\zeta \\ \mathcal{V}_\zeta &= \mathcal{V}_\zeta^{\text{flow, x}} \cap \mathcal{V}_\zeta^{\text{flow, y}} \end{aligned} \tag{14}$$

with $\mathcal{V}_\zeta^{\text{flow, x}}$ and $\mathcal{V}_\zeta^{\text{flow, y}}$ defined in the same way as $\mathcal{V}_\eta$ using respectively Δ_{flow}^x and Δ_{flow}^y. The loss-oriented neighbourhood $\mathcal{V}_\eta$ remains the main actor of the decision process, while $\mathcal{V}_\zeta$ can be seen as a **prior with a regularisation** effect to solve the homogeneous issue. One might well ask why not consider the magnitude of the flow differences instead. The reason is that this choice makes all that had been demonstrated with Δ (that assumes signed values and a normally distributed random variable) to remain true.

4.4. Implementation details

Additional Losses On top of $\mathcal{L}_{\text{CoopNet}}$ defined in Eq. 10, we also take into account the following subsidiary losses:

- The geometric consistency loss $\mathcal{L}_{\text{gc}}$ proposed by [1].
- A forward-backward consistency check $\mathcal{L}_{\text{fwd,bwd}}$ of the optical flow $\mathcal{F}_\delta$ as done in [31, 44].
- The standard edge-aware smoothness loss $\mathcal{L}_s$ for both depth and flow maps. The normalised disparity is used here as proposed by [36] to avoid divergence.
- The epipolar constraint $\mathcal{L}_{\text{ep}}$ with different version of the one proposed in [3] (see supplementary materials).
- The inverse of the variance of depth maps $\mathcal{L}_{\text{var}}$ mentioned in [21] in order to stabilise the training process.

The final loss is:

$$\begin{aligned}\mathcal{L}_{\text{final}} = \mathcal{L}_{\text{CoopNet}} + \lambda_{gc}\mathcal{L}_{gc} + \lambda_{\text{fwd,bwd}}\mathcal{L}_{\text{fwd, bwd}} \\ +\lambda_s\mathcal{L}_s + \lambda_{\text{ep}}\mathcal{L}_{\text{ep}} + \lambda_{\text{var}}\mathcal{L}_{\text{var}}\end{aligned} \tag{15}$$

Network Architectures. Our focus in this work is to promote our ***cooperation*** learning protocol and to see how it compares with the other well-established **self-supervised depth estimation** training strategies [10, 43, 3, 26]. For a better comparison, we decided to use the same standard networks as those methods. In particular, for both the depth and flow networks, we adopt a UNet structure with four intermediate multi-scale predictions as proposed by [43]. The Pose networks is based on a ResNet encoder at the end of which a 6-DoF vectors is predicted. For the depth network we use the specific DispResNet architecture of [10]. For the flow network we implement the ResFlowNet of [41, 25]. Both the depth and pose networks have a ResNet18 backbone while the flow network uses a ResNet50 encoder. More efficient networks could of course improve performances even more. For instance, regarding the depth network, PackNet [12], architectures using attention [20, 7, 19, 27] and/or cost-volume [39, 20] are performing well, just like FlowNet [5] and PwC-Net [15, 34] for the flow prediction.

Occlusions are dealt with in two ways. The *warping module* of [37] is used to mask occluded pixels in $\mathcal{L}_\delta$ with a hard occlusion threshold set to 0.2. While for $\mathcal{L}_{\theta,\alpha}$, occluded pixels are handled thanks to the standard *minimum re-projection* of [10].

Parameter settings. Our method is implemented in PyTorch. Training is done using the optimiser Adam [22] with $\beta_1 = 0.99$ and $\beta_2 = 0.999$. ResNet backbones are initialised using ImageNet [33] pretrained weights. Networks are trained for 30 epochs with a batch size of 4. The initial learning is set to 10^{-4} and decreased to 10^{-5} after 20 epochs. Standard data-augmentation is performed including horizontal flips, random contrast, saturation, hue and brightness jitters. A burning step of 5 epochs is used during which the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ is trained with [10]. Quantiles are computed on the fly based on the algorithm of [18]. The neighbourhood $\mathcal{V}_\eta$ is determined using quantile values of the previous epoch (see Fig. 4) with $\eta = 0.15$ and $\zeta = 0.25$. The loss weights were determined with a grid-search and finally set to $\lambda_{\text{gc}} = 0.001$, $\lambda_{\text{fwd, bwd}} = 0.001$, $\lambda_s = 0.01$, $\lambda_{ep} = 0.001$ and $\lambda_{\text{var}} = 10^{-6}$. We employ a single NVIDIA GTX 1080 Ti GPU. Training time takes 12 hours.

5. Experiments

We conducted extensive experiments on depth, camera pose and optical flow estimations in order to validate our method. We present the results obtained on two datasets:

KITTI [9] is the most popular benchmark to evaluate depth and ego-motion estimations. It consists of urban, rural and highway images, captured by driving around the city of *Karlsruhe*. We use the standard evaluation protocol to retrieve the ground truth depth values from the LIDAR sensor data, and we follow the standard data split proposed by [6] with 22 600 training pair images and 697 test pair images.

Cityscapes [4] is also composed of urban images but it contains a greater variety of situations with images coming from more than 50 European cities and is challenging because it contains more scenes with moving objects. Following the protocol of [26] training is done on 22 973 image-pairs obtained by completing the usual 2975 training images with the 19 998 extra-training images. For evaluation, we use the 1 525 test images.

5.1. Depth

Quantitative results are given in Tab. 1. **CoopNet** outperforms already very effective methods with a substantial margin in almost all of the different metrics. Qualitative results are presented in Fig. 5. Overall, depth maps yielded by **CoopNet** are sharpest and achieve better predictions in challenging situation such as thin objects, high

Set	Method	Size	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
K	Li *et al.* [26]	128×416	0.130	**0.950**	5.138	0.209	0.843	0.948	0.978
	DLNet[19] ‡	128×416	0.128	0.979	**5.033**	0.202	.851	0.954	0.980
	CoopNet	128×416	0.126	1.014	5.091	0.204	0.856	0.954	**0.980**
	CoopNet R50	128×416	**0.121**	0.971	5.055	**0.199**	**0.863**	**0.955**	**0.980**
	Monodepth2[10]	192×640	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	SGDepth [23] †	192×640	0.117	0.907	**4.693**	0.191	**0.879**	**0.961**	0.981
	Tosi *et al.* [35] †	192×640	0.126	**0.835**	4.937	0.199	0.844	0.953	0.982
	CoopNet	192×640	**0.113**	0.872	4.824	**0.190**	0.878	0.959	**0.982**
	Insta-DM [24] †	256×832	0.112	0.777	4.772	0.191	0.872	0.959	0.982
CS	Struct2Depth[2]†	128×416	0.145	1.737	7.28	0.205	0.813	0.942	0.978
	Gordon[11] †	128×416	0.127	1.330	6.96	0.195	0.830	0.947	0.981
	Li *et al.*[26]	128×416	**0.119**	**1.29**	**6.98**	**0.190**	**0.846**	**0.952**	**0.982**
	CoopNet	128×416	0.121	1.443	7.01	**0.190**	**0.846**	0.951	0.980

Table 1: **Results of depth estimations**. We only compare our methods to the most recent and competitive algorithms. All results here are presented for different image sizes. For each metric the best result is displayed in bold and the second one is underlined. The depth cutoff is set to 80m. For red metrics, lower is better. For blue metrics, higher is better. † - Use of an off-the-shelf semantic algorithms. ‡ - Use of a transformer depth network. **K**: trained and evaluated on KITTI. **CS**: trained and evaluated on Cityscapes. **R50:** Use a ResNet50 backbone instead of ResNet18 for the depth network.

Methods	Seq. 09		Seq. 10	
	t_{err} (%)	r_{err} ($^\circ$/100m)	t_{err} (%)	r_{err} ($^\circ$/100m)
ORB[32]	15.30	0.26	3.68	0.48
Zhou[43]	17.84	6.78	37.91	17.78
Bian[1]	11.2	3.35	10.1	4.96
CoopNet	**8.42**	**2.66**	**7.29**	**2.14**

Table 2: **Odometry**: Average Translation and Rotation errors for sequence 09 and 10 of the KITTI Odometry Dataset.

Method	Noc	All
FlowNetS[5]	8.12	14.19
FlowNet2[16]	4.93	10.06
GeoNet[41]	8.05	10.81
GLNet[3]	**4.86**	**8.35**
CoopNet	5.10	9.43

Table 3: **Optical Flow:** Average end point error (in pixels) for non occluded (Noc) and for all (All) pixels on the KITTI 2015 flow dataset.

$\mathcal{L}_{gc}$	$\mathcal{L}_{ep}$	$\mathcal{L}_{s}$	$\mathcal{L}_{fwd,bwd}$	$\mathcal{L}_{var}$	$\mathcal{L}_{photo}$			Abs Rel	APE
					$\mathcal{L}_{baseline}$	$\mathcal{L}_{apc}$	$\mathcal{L}_{CoopNet}$		
		✓			✓			0.157	11.93
		✓				✓		0.144	12.26
		✓					✓	0.130	9.74
		✓	✓				✓	0.130	9.21
		✓	✓	✓			✓	0.130	9.16
✓		✓	✓				✓	0.128	9.27
✓	✓	✓	✓	✓			✓	0.126	9.43
✓	✓	✓	✓			✓		0.135	8.35

Table 4: **Ablation study on absolute relative error (depth) and average end point error (flow).** Resolution size: 128×416. In our baseline, both the optical flow $\mathcal{F}_\delta$ and the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ are trained using the standard *photometric* losses $\mathcal{L}$ (see equations 1 to 4) computed on all of the pixels. As common practice, the smoothness loss $\mathcal{L}_s$ is used in all experiments. Last row corresponds to GLNet [3].

texture and moving regions. An ablation study is also presented in Tab. 4. Note the great improvements brought by $\mathcal{L}_{\text{CoopNet}}$ (line 3) compared to the two other types of photometric loss (line 1-2). As displayed in the second part of the ablation study, the subsidiary loss benefits are marginal as compared to $\mathcal{L}_{\text{CoopNet}}$.

5.2. Optical flow and Odometry

To assess optical flow we use the KITTI 2015 flow dataset containing 200 annotated training images as test images. Tab. 3 shows that **CoopNet** gives close results to GLNet[3] while outperforming all the other methods.
The results of our camera-pose estimations trained on KITTI are shown in Tab. 2. Again, our method achieves significant gains over the presented methods. We chose to compare specifically to [1], as their *scale-consistent* approach is particularly centred on the odometry. The results are however still below classical approaches [32].

5.3. Visual analysis of Δ

Fig. 6 shows some example of Δ values. We observe experimentally that rigid pixels p_{rigid} for which $\boldsymbol{\Delta}\,(\boldsymbol{p}_{\text{rigid}})$ differs the most from μ (blue and red values) correspond to

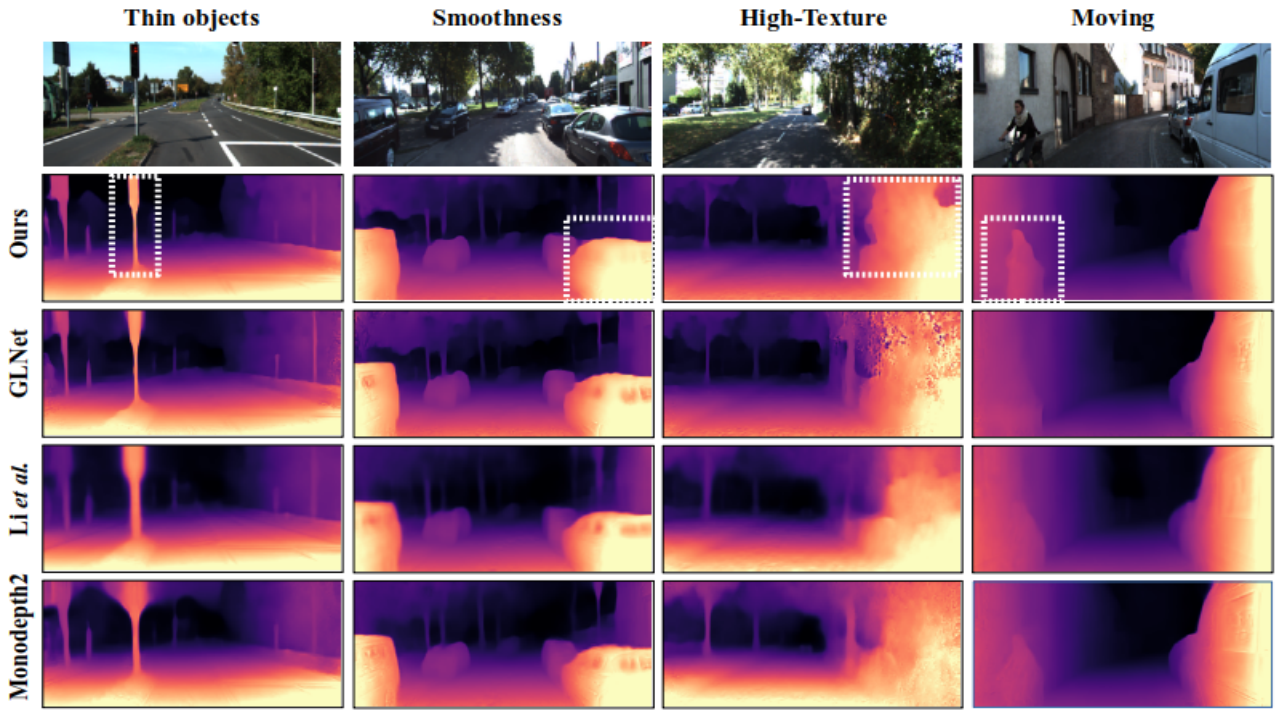
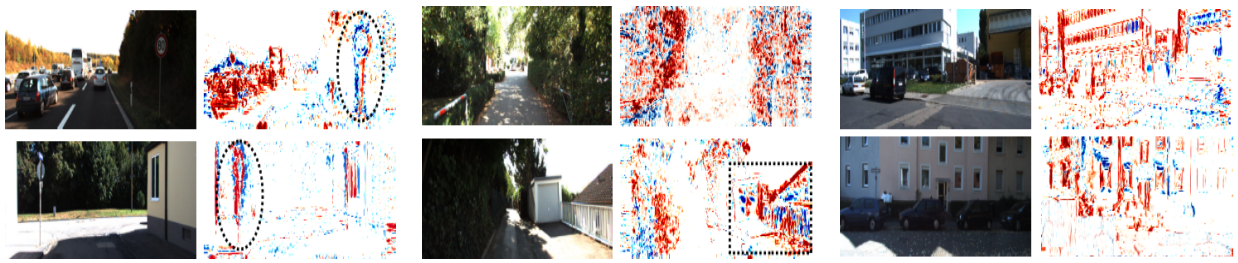


Figure 5: Comparison of depth map estimation algorithms in challenging situations. White Dashed rectangles target the improvement brought by our method.



(a) Thin objects (b) High-Texture regions (c) Edges

Figure 6: Illustration of the large variations of $\mathbf{\Delta}$ between positive and negative values in challenging cases indicative of a **strong ambiguity**. Also note the dominance of the red colour due to the intrinsic bias mentioned in Sec. 3.1.

tricky cases where it's quite difficult, even for a human, to determine the flow displacement: for instance pixels at the edges, in high-texture areas or around thin objects. Conversely, values nearby μ (white values) come from rigid pixels that are easy to infer. This supports our claim that when η is low enough, the neighbourhood $\mathcal{V}_\eta$ can be seen as an *agreement area*. In other words, both the pair $(\mathcal{D}_\theta, \mathcal{T}_\alpha)$ and $\mathcal{F}_\delta$ *share the same understanding on the mechanism that governs the displacement* of pixels from $\Delta^{-1}(\mathcal{V}_\eta)$.

6. Conclusion

We have presented **CoopNet**, a training strategy that achieves competitive performances in depth, ego-motion and optical flow estimations using unsupervised training. It relies on a healthy cooperation between different visual tasks so that each one can benefit from the others, relying on the fact that networks should agree on their warping displacement prediction for a pixel to be considered as rigid.

This idea could be further improved by combining it with an explicit residual correction of the ego-motion [26]. In this case $\mathcal{V}_\eta$ no longer represents rigid pixels exclusively, but can still be seen as an ***agreement area*** from which one can take advantage to emphasise the training process on pixels for which networks disagree.

References

[1] J.W. Bian, Z. Li, N. Wang, H. Zhan, C.Shen, M.M Cheng, and I. Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *International Conference on Neural Information Processing Systems (NIPS)*, page 35–45, 2019.

[2] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 8001–8008, 2018.

[3] Y. Chen, C. Schmid, and C. Sminchiescu. Self-supervised learning with geometric constraints in monocular video. In *ICCV*, pages 7063–7072, 2019.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016.

[5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015.

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *International Conference on Neural Information Processing Systems (NIPS)*, page 2366–2374, 2014.

[7] F .Gao, J. Yu, H. Shen, Y. Wang, and H. Yang. Attentional separation-and-aggregation network for self-supervised depth-pose learning in dynamic scenes. In *Conference on Robot Learning (CoRL 2020)*, Cambridge MA, 2020.

[8] R. Garg, V. Kumar, B.G. Gustavo, and I. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *ECCV*, pages 740–756. Springer International Publishing, 2016.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[10] C. Godard, O.M. Aodha, M. Firman, and G. Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pages 3828–3838, 2019.

[11] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *ICCV*, pages 8977–8986, 2019.

[12] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon. 3d packing for self-supervised monocular depth estimation. In *CVPR*, pages 2485–2494, 2020.

[13] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon. Semantically guided representation learning for self-supervised monocular depth. In *International Conference on Learning Representations*, 2020.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *ICCV*, pages 2961–2969, 2017.

[15] J. Hur and S. Roth. Self-supervised monocular scene flow estimation. In *CVPR*, 2020.

[16] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, P. Van der Smagt, D. Cremers, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, pages 2462–2470, 2017.

[17] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *International Conference on Neural Information Processing Systems (NIPS)*, 2016.

[18] R. Jain and I. Chlamtak. The P2 algorithm for dynamic statistical computing calculation of quantiles and histograms without storing observations. *Communications of The ACM - CACM*, 28, 1985.

[19] S. Jia, X.Pei, W. Yao, and S.C Wong. Self-supervised depth estimation leveraging global perception and geometric smoothness using on-board videos. *CoRR*, abs/2106.03505, 2021.

[20] A. Johnston and G. Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *CVPR*, pages 4756–4765, 2020.

[21] U.H. Kim and J.H. Kim. Revisiting self-supervised monocular depth estimation. *Robot Intelligence Technology and Applications*, 2022.

[22] D.P Kingma, J. Ba, N. Snavely, and D.G. Lowe. ADAM: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diego, CA, 2015.

[23] M. Klingner, J.A Termöhlen, J. Mikolajczyk, and T. Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. In *ECCV*, pages 582–600. Springer International Publishing, 2020.

[24] S. Lee, S. Im, S. Lin, and I.S Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency. In *AAAI Conference on Artificial Intelligence*, 2021.

[25] S. Lee, S. Im, S. Lin, and S. Kweon. Learning residual tease flow as dynamic motion from stereo videos. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[26] H. Li, A. Gordon, H. Zhao, V. Casser, and A. Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning (PMLR)*, pages 1908–1917, 2020.

[27] J. Li, J. Zhao, S. Song, and T. Feng. Unsupervised joint learning of depth, optical flow, ego-motion from video. *CoRR*, abs/2105.14520, 2021.

[28] R. Li, X. He, D. Xue, S. Su, Q. Mao, Y. Zhu, J. Sun, and Zhang Y. Learning depth via leveraging semantics: Self-supervised monocular depth estimation with both implicit and explicit semantic guidance. *CoRR*, abs/2102.06685, 2021.

[29] T.Y Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. Lawrence Zitnick, and P. Dollár. Microsoft COCo: Common objects in context. In *ECCV*, pages 740–755. Springer International Publishing, 2015.

[30] L. Liu, G. Zhai, W. Ye, and Y. Liu. Unsupervised learning of scene flow estimation fusing with local rigidity. In

Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pages 876–882. International Joint Conferences on Artificial Intelligence Organization, 2019.

[31] S. Meister, J. Hur, and S. Roth. Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, New Orleans, Louisiana, Feb. 2018.

[32] R. Mur-Artal, J.M.M Montiel, and J. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31:1147 – 1163, 10 2015.

[33] O. Russakovsky, J. Deng, H. Yao, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, page 211–252, 2015.

[34] D. Sun, X. Yang, M.Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8943–8943, 2018.

[35] F. Tosi, F. Aleotti, P.Z Ramirez, M. Poggi, S. Salti, L.D Stefano, and S. Mattoccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of IEEE CVPR*, page 4654–4665, 2020.

[36] C. Wang, J.M Buenaposada, R. Zhu, and S. Lucey. Learning depth from monocular videos using direct methods. In *CVPR*, 2018.

[37] Y. Wang, Y. Yang, Z. Yang, L.Zhao, P.Wang, and W.Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, 2018.

[38] Z. Wang, A.C Bovik, H.R Sheikh, and E.P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, pages 600–612, 2004.

[39] J. Watson, O.M. Aodha, V. Prisacariu, G. Brostow, and M. Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *CVPR*, pages 1164–1174, 2021.

[40] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding. In *ECCV 2018 Workshops*, 2018.

[41] Z. Yin and J. Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, pages 1983–1992, 2018.

[42] J.J. Yu, A.W. Harley, and K.G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV 2016 Workshops*, pages 3–10. Springer International Publishing, 2016.

[43] T. Zhou, M. Brown, N. Snavely, and D.G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, pages 1851–1860, 2017.

[44] Y. Zou, Z. Luo, and J.B. Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision*, 2018.