Self-supervised Relative Pose with Homography Model-fitting in the Loop

Bruce R. Muller and William A. P. Smith
Department of Computer Science
University of York, UK
{brm512, william.smith} @york.ac.uk

Abstract

We propose a self-supervised method for relative pose estimation for road scenes. By exploiting the approximate planarity of the local ground plane, we can extract a self-supervision signal via cross-projection between images using a homography derived from estimated ground-relative pose. We augment cross-projected perceptual loss by including classical image alignment in the network training loop. We use pretrained semantic segmentation and optical flow to extract ground plane correspondences between approximately aligned images and RANSAC to find the best fitting homography. By decomposing to ground-relative pose, we obtain pseudo labels that can be used for direct supervision. We show that this extremely simple geometric model is competitive for visual odometry with much more complex self-supervised methods that must learn depth estimation in conjunction with relative pose. Code and result videos: github.com/brucemuller/homographyVO.

1. Introduction

Estimation of the relative pose between two images is important for applications in computer vision including visual odometry (VO), image stitching, structure-from-motion, change detection and augmented reality. The rapid development in autonomous vehicle technology has brought particular focus on VO. While a classical problem in vision, VO methods based on local feature extraction and matching can be fragile, fail for textureless scenes and slow.

On the other hand, over the past 5 years deep learning based methods have shown themselves to be robust and provide fast inference. However, since these methods are only trained to be optimal in aggregate over a training set, they do not necessarily provide the optimal solution for a given image pair and therefore lack the precision of classical methods that can exactly align features that were correctly matched. In addition, most learning based VO techniques rely on ground truth relative pose labels for supervised learning. These labels are difficult to collect, suffer from inconsistent coverage and must be synchronised and geometrically calibrated relative to the cameras.

Self-supervised methods provide an alternative approach that can exploit the vast amounts of unlabelled driving video. Most commonly, these methods simultaneously learn depth and relative pose estimation such that a supervision signal can be obtained via cross-projection of one image into the other [13]. In many cases, relative pose estimation is simply a byproduct of seeking to learn depth estimation. While depth is very useful in its own right, it entails estimating potentially millions of depth values for each image. This is an ill-posed problem for which learning-based methods tend to overfit the biases in their training data [9, 23]. Errors in depth will influence the accuracy of relative pose and vice versa, thus this approach is not necessarily optimal if the goal is only to estimate relative pose.

We propose a method for relative pose estimation that combines self-supervised learning with classical feature matching and alignment. We leverage the fact that, for autonomous driving applications, the scene contents (i.e. road scenes) contain significant regions of approximately flat ground plane. This enables us to cross-project between images (and hence obtain a supervision signal) using only a homography. By explicitly enforcing the planar nature of road scenes, we dramatically simplify the task that the network must solve, while retaining the benefits of self-supervision. In addition, we refine and provide an alternate supervision signal by using classical image alignment within the network training loop. Specifically, we make the following contributions:

1. We regress 9D ground-relative pose using a geometric matching network that can handle arbitrary pose changes on overlapping image pairs.

2. An appearance loss is provided via differentiable cross-projection using the estimated homography.

3. We compute a refined homography by applying a non-differentiable optical flow plus RANSAC procedure to regions of the image labelled as ground plane by a se-

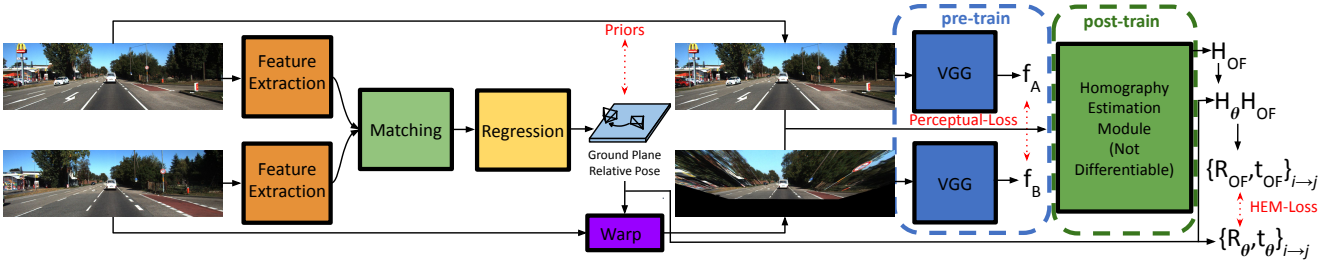

Figure 1. We estimate ground-relative pose using a geometric matching network trained without labels and without estimating depth. By assuming a locally planar scene, we can compute the homography between views and use this to provide a self-supervision signal by cross-projection. We train in two phases: 1. using a perceptual loss based on deep features provided by a pre-trained VGG [26] network, 2. via a Homography Estimation Module (HEM) which harnesses model fitting (optical flow + RANSAC) to fit a homography which we decompose to camera-relative pose for supervision. The HEM can also be utilised at test-time to improve performance.

mantic segmentation network. This provides pseudo-labels and, hence, another source of self-supervision.

4. At inference time, the optical flow based refinement can be applied on top of network output for improved accuracy for tasks such as VO.

To the best of our knowledge, the 9D parameterisation as a network output for self-supervised VO is novel. No other pose estimation works utilise this method, and generally adopt a 6 DoF camera-relative pose, choosing to learn scene regularity, rather than utilising it. Furthermore, keeping a geometric parameterisation general is more powerful as we can extract from it multiple useful transformations.

Further, the estimated relative poses from our method can be used for trajectory estimation by applying transformation synchronisation across all overlapping image pairs in the sequence. We use this approach to evaluate our method on the KITTI VO benchmark. Self-supervision provided by our simple geometric model and optical flow based refinement is highly competitive with state-of-the-art self-supervised methods that require dense depth estimation.

2. Related Work

Self-Supervised Relative Pose Most self-supervised VO methods parameterise network outputs with dense depth and 6 DoF camera-relative pose [5, 7, 11, 13, 14, 19, 33, 35], allowing for cross-projecting one image into the perspective of another, to be then directly compared to form a training loss. D3VO [30] is the most competitive purely monocular method on the KITTI odometry benchmark. They utilise pose-depth networks with illumination transformations and estimated uncertainty maps to provide an improved self-supervised training loss similar to [13]. However, they use the depth, pose and uncertainty map predictions solely to perform an offline, nonlinear bundle adjustment over the entire sequence, which makes it akin to classical optimisation-based methods and not directly comparable to much faster direct regression methods. LT-MVO [37] achieves the best VO results for self-supervised methods by using a recurrent CNN to temporally constrain the trajectory but also rely on pose-depth networks with a 6 DoF camera-relative pose. Highly competitive self-supervised approaches [11, 14, 27, 28] are reliant on dense depth estimation. Recently, methods use dense optical flow [20,24,34,38] with camera-relative pose or depth estimation to form a self-supervised signal.

Parameterising as a camera-relative pose and dense depth estimation task tends to limit estimation to adjacent or temporally close video frames. Further, estimating many thousands of parameters for depth or flow is a demanding and ill-posed task which is hard to train. For example, Monodepth2 [13] performs very well with depth estimation, but significantly less so with pose estimation. This implies that methods using a pose and depth network are prone to the issue of one network influencing the accuracy of the other. Tiwari et al. [27] attempt to remedy this issue but rely on classical SLAM and potentially expensive optimisation routines such as bundle adjustment and loop closure.

While there are works tackling direct homography estimation [8, 25, 29], we specifically tackle road-scene relative pose estimation and thus do not compare to these methods.

None of these approaches utilise the basic known geometry in road-scenes: the ground is approximately planar. We propose instead to parameterise with respect to the ground plane, cross-projecting via that known geometry to form the training loss, avoiding the requirement of estimating dense depth with a second network entirely. Note that a useful consequence of our parameterisation is that we can obtain road depth from our ground-relative poses. Further, our method, while constrained to a planar model, is highly flexible as it allows for estimating arbitrary relative poses.

While classical approaches like ORB-SLAM2 [22] are a powerful approach, they often fail with slightly larger pose variations (which our parameterisation is robust to) and usually rely on intensive bundle adjustment and loop-closure.

Road scenes are highly regular, but Dijk *et al.* show that

common road depth networks simply utilise the vertical image position of objects, rather than overall size. Further, they show generalised depth accuracy depends on the presence of accompanying features for objects (e.g. shadows). Such behaviour is common when forcing networks to learn without reason on large datasets in a black-box fashion. We model the regularity of the road plane explicitly, helping to avoid this over-fitting.

Architectural Considerations Many methods concatenate pose network input images, assuming that the receptive field of convolutions will be sufficient to capture the local variations in features for accurate pose, but this favours only little variation in relative pose between frames.

Rocco et al. [25] use a geometric matching architecture for directly estimating a geometric transformation to synthetically warp object instances into a similar perspective. Inspired by traditional feature matching pipelines, their architecture consists of separate feature extraction branches with shared weights, and a novel matching layer, essentially allowing regression based on putative feature matches between both images. We chose their architecture due to its effectiveness of capturing correspondences which accurately convey geometric perspective, and avoiding use of input concatenation which aligns with our thesis of arbitrary pose estimates. Work by [10] uses this network [25] to estimate a thin plate spline directly for their human-pose system for trying on clothing. To the best of our knowledge, we are the first to use the geometric matching architecture [25] for the task of 3D relative pose estimation.

Perceptual Loss and Model Fitting Popularised by work in style transfer and image denoising [17, 31], we choose to train initially using a perceptual loss instead of a per-pixel loss for the image difference, which provides a wider basin of convergence. This avoids problems with illumination assumptions required for pixel-level loss, which often requires adding more regularisation terms. To the best of our knowledge we are the first to use a perceptual loss with a primary focus on VO evaluation, and *also the first to parameterise deep-pose in terms of the local geometry surrounding a camera-pair*. Inspired by Kolotouros et al. [18] we chose to use a model-fitting in the loop approach to help further refine our learned model and allow for inference time refinement. While other works such as [6] have also been inspired by [18], as far as we know, we are the first to apply the concept to motion estimation setting with homographies.

3. Two View Ground-Relative Geometry

We propose to predict the *positioning of two views relative to their local ground plane*. Our novel parametisation is ground-relative and illustrated in Fig. 2. The parameterisation has 9 degrees of freedom: 3 to define the plane relative to the first camera, and 6 to define the second camera relative to the first. In this section we detail our ground-relative parameterisation, how we extract camera-relative pose from this, and how we compute a homographic cross-projection from our parameterisation for mapping road plane pixels between two cameras. Lastly, we explain handling the scale ambiguity present in planar cross-projection.

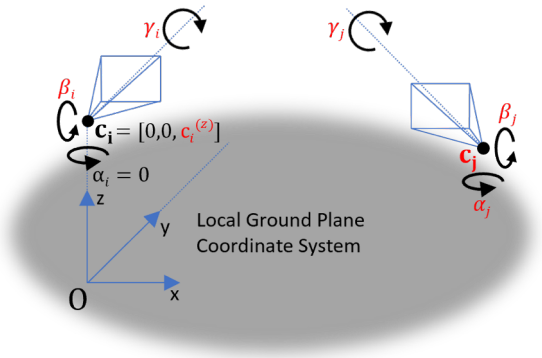

Figure 2. Our ground-relative coordinate system (9D parameterisation - red) comprises four translational and five rotational parameters for the two cameras i and j of the network input-pair. Specifically, we predict the two camera heights, planar position for camera j, and roll and pitch for both cameras, all relative to an origin defined to be on the ground-plane directly under camera i.

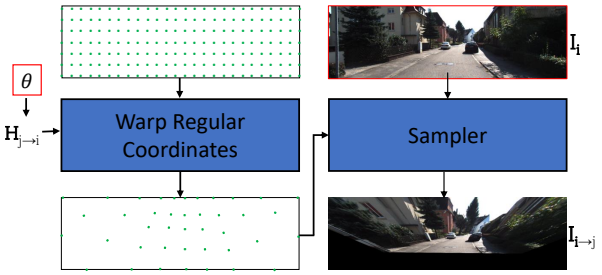

Figure 3. The local road scene planar geometry allows for differentiable cross-projection via backwards warping with a homography. We transform a regular grid of points with a homography computed from ground-relative pose θ, to sample network input. Red boundaries represent input to the warping module in Fig 1.

Parameterisation Using a 9D form $\theta \in \mathbb{R}^9$ (see Fig. 2), we write the ground-relative pose of cameras i and j as:

$$\theta = (c_i^{(z)}, \gamma_i, \beta_i, c_j^{(x)}, c_j^{(y)}, c_j^{(z)}, \gamma_j, \beta_j, \alpha_j). \quad (1)$$

As shown, we define a local coordinate frame where camera i is positioned directly above the origin, the optical axis is aligned with the y-axis, and the ground plane coincides with $z = 0$. Therefore, there exist three degrees of freedom for camera i: roll (γ_i) and pitch (β_i) relative to the local orientation of the ground, and its distance above the ground ($c_i^{(z)}$). Camera j is specified with six parameters: a position $\mathbf{c}_j = [c_j^{(x)}, c_j^{(y)}, c_j^{(z)}]$ in the local coordinate system and a rotation defined by roll, pitch and yaw (γ_j, β_j and α_j).

We use Tait–Bryan angles to parameterise rotation as vehicular motions is represented naturally in this way and

priors on each parameter is simplified. For example, over unbiased motion sequences, a camera facing forwards with optical axis aligned parallel to the ground plane will have zero mean pitch and roll. We emphasise that our representation describes the position of both cameras relative to the local ground plane. This is entirely a local parameterisation where it does not imply that the local ground plane aligns with the global $z = 0$ plane, i.e. the direction of gravity is not necessarily aligned with the z-axis. Therefore, under the assumption that small motions can be approximated as planar motion, we can describe non-planar motion sequences.

Relative Pose from Parameterisation Camera-relative pose is computed from our ground-relative pose, which is used later for estimating absolute pose trajectories. Moreover, it is important for the second stage of self-supervision we propose in Section 5. Using camera angles and centres we may compute world-to-camera rotation and translation:

$$\begin{aligned}\mathbf{R}(\gamma,\beta,\alpha) &= \mathbf{R}_z(\gamma)\mathbf{R}_x(\beta)\mathbf{R}_y(\alpha)\mathbf{R}_x(90^\circ)\\ \mathbf{t}(\mathbf{c},\gamma,\beta,\alpha) &= -\mathbf{R}(\gamma,\beta,\alpha)\mathbf{c}\end{aligned} \tag{2}$$

World coordinates (z up) are converted to camera coordinates (z aligned with optical axis) via the fixed rotation $\mathbf{R}_x(90^\circ)$. World-to-camera transforms for the two views are computed from our parameterisation (1) with (2) as:

$$\mathbf{R}_i = \mathbf{R}(\gamma_i,\beta_i,0),\ \mathbf{t}_i = \mathbf{t}([0,0,c_i^{(z)}]^\top,\gamma_i,\beta_i,0) \tag{3}$$
$$\mathbf{R}_j = \mathbf{R}(\gamma_j,\beta_j,\alpha_j),\ \mathbf{t}_j = \mathbf{t}([c_j^{(x)},c_j^{(y)},c_j^{(z)}]^\top,\gamma_j,\beta_j,\alpha_j)$$

As shown in Fig. 2, we define camera i to be forward facing ($\alpha_i = 0$) and directly above the local coordinate frame ($c_i^{(x,y)} = 0$). The camera-relative pose for transforming between coordinate systems of camera i to j is given by:

$$\mathbf{R}_{i\to j} = \mathbf{R}_j\mathbf{R}_i^\top\ ,\ \ \mathbf{t}_{i\to j} = \mathbf{t}_j - \mathbf{R}_{i\to j}\mathbf{t}_i \tag{4}$$

Planar Cross-Projection We propose to supervise in an initial stage by cross-projecting one of the input images into the perspective of the other to form an appearance consistency loss. This is straightforward due to our assumption of local planarity. By deriving a homography from our ground-relative representation, the dominant planar part of the scene, namely the road, can be accurately cross-projected. Transformation of a point on the local $z = 0$ ground-plane to a camera k is given by the homography:

$$\mathbf{H}_k(\mathbf{K}_k,\mathbf{R}_k,\mathbf{t}_k) = \mathbf{K}_k[\mathbf{R}_k\mathbf{S}^\top\ \ \mathbf{t}_k],\mathbf{S} = \begin{bmatrix}1&0&0\\0&1&0\end{bmatrix}, \tag{5}$$

where $\mathbf{R}_k$, $\mathbf{t}_k$ are derived from (3) and $\mathbf{K}_k$ are provided camera intrinsics. We may then combine two homographies for two cameras overlooking the same plane. The homography mapping a location in image i to a point on the ground-plane, and to the corresponding position in image j:

$$\mathbf{H}_{i\to j} = \mathbf{H}_j\mathbf{H}_i^{-1}. \tag{6}$$

Scale Ambiguity A homography relates points between two views which are located on the local ground plane with 8 DoF. The parameterisation (1) for our ground-relative pose spans 9 DoF. Scale ambiguity explains the extra dimension, as the homography between two views is invariant to scaling the ground-relative translations (or equivalently camera centres above the ground plane). Therefore, it is not feasible to estimate ground-relative poses at global scale with only the planar correspondences. However, road scene datasets such as KITTI [12] often include calibrations for parameters such as camera height above the local ground plane. Moreover, vehicular motion such as acceleration, cornering, bumps in the road surface etc, can cause variations in the height of a mounted camera. By using the calibrated height as a prior training loss, we softly constrain the mean calibrated height (and hence scale), resolving the unknown scale ambiguity. Further, these priors for camera height, roll and pitch are normally distributed around the calibration values, and thus we can handle small variations.

4. Pre-training: Learning via Perceptual Loss

As indicated in Fig. 1, we train our network in two stages. Here, we describe our self-supervised method for pre-training from scratch using priors and appearance loss.

4.1. Pre-Training Losses

Priors For most road-scene datasets the mean camera height, roll and pitch relative to the road is known. For the KITTI dataset, our motion model assumes that β and γ have a mean of zero degrees and that the calibrated height of the camera above the road plane, $c_{\text{cal}}^{(z)}$, has a mean of 1.65 metres, and that variation is normally distributed. To enforce this motion model, we use the priors loss function:

$$L_{\text{pri}} = (c_i^{(z)} - c_{\text{cal}}^{(z)})^2 + (c_j^{(z)} - c_{\text{cal}}^{(z)})^2 + \gamma_i^2 + \gamma_j^2 + \beta_i^2 + \beta_j^2 \tag{7}$$

where we represent camera height, roll and pitch as $c_{i,j}^{(z)}$, $\gamma_{i,j}$ and $\beta_{i,j}$ respectively for each camera pair (i,j).

Perceptual Loss We cross-project one input image into the perspective of the other (via Eqn. (6)) and form a perceptual loss between them to self-supervise our network initially. We use a symmetric L2 loss between both images, with a sum over 2 scales to improve convergence:

$$\begin{aligned}L_{\text{pe}} = \sum_{s=1}^{2} &\left\|\text{VGG}(\text{ds}(\mathbf{I}_j,s)) - \text{VGG}(\text{ds}(\mathbf{I}_{i\to j},s))\right\|_2 \frac{1}{M_j^{(s)}}\\ &+ \left\|\text{VGG}(\text{ds}(\mathbf{I}_i,s)) - \text{VGG}(\text{ds}(\mathbf{I}_{j\to i},s))\right\|_2 \frac{1}{M_i^{(s)}}\end{aligned} \tag{8}$$

where $\text{ds}(\mathbf{I},s)$ is differentiable downsampling of $\mathbf{I}$ by a factor s, VGG is inference of feature maps from the first seven convolution layers of VGG-16 [26] (ImageNet pre-trained),

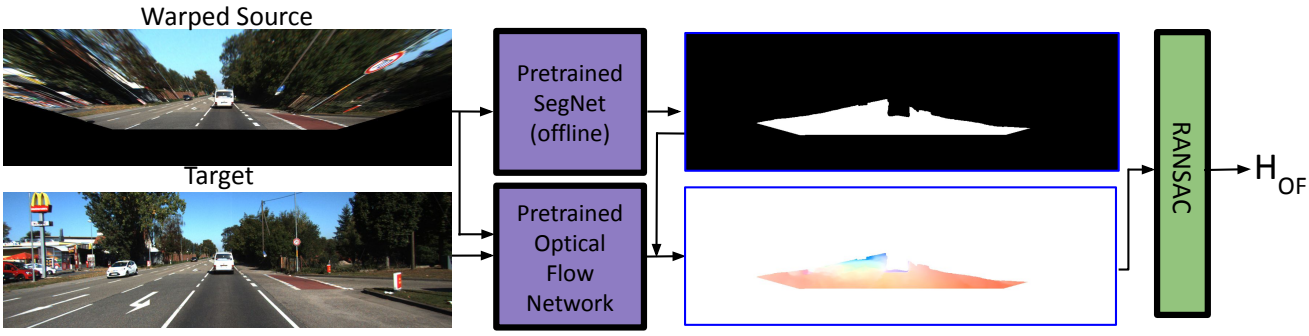

Figure 4. Non-Differentiable Homography Estimation Module (see Fig. 1): we use a pre-trained optical flow network to estimate point correspondences between one network input and the corresponding input transformed via the ground-relative pose output. A pre-trained semantic segmentation network isolates the road plane points so that RANSAC can be used to robustly estimate the road plane homography.

and $M_{i,j}^{(s)}$ is the number of pixels in the cross-projected image that are within warped coordinates for scale s.

Differentiable cross-projection is achieved by following the sampling method used by Spatial Transformer Networks [16] (see Fig. 3). In particular, differentiable bilinear sampling is utilised, where cross-projected image $\mathbf{I}_{i\to j}$ is formed from warping image $\mathbf{I}_i$ from camera i into the perspective of camera j. Firstly, we use the pose parameters (1) from the network output to compute a reverse homography $\mathbf{H}_{j\to i}$ with (5) and (6). Secondly, we form matrix $\mathbf{X} \in \mathbb{R}^{3\times HW}$ of homogeneous coordinates by composing coordinates in a regular grid. By applying $\mathbf{H}_{j\to i}$ with each coordinate we form a grid of transformed coordinates. Finally, we use differentiable bilinear sampling at the warped coordinates to sample image $\mathbf{I}_i$: $\mathbf{I}_{i\to j} = \text{sample}(\mathbf{I}_i, \mathbf{H}_{j\to i}\mathbf{X})$.

The total loss for pre-training our network is formed from the weighted (chosen to balance both terms) sum of the perceptual and prior losses: $L_{\text{total}} = w_1 L_{pe} + w_2 L_{\text{pri}}$, where $w_1 = 1$ and $w_2 = 287000$. See the supplementary material for architecture and further training details.

5. Post-Training: Model-fitting in the Loop

In the previous section we relied on the network to learn an image to homography function based on a perceptual loss where backward gradients must pass coherently through a bilinear sampler. In this section we show that we can extract a homography directly from an image-pair and then, with basic knowledge of the scene, decompose it into camera-relative pose for the purpose of directly supervising the network and for estimating camera-relative pose at test-time.

5.1. Homography Estimation Module (HEM)

Fig. 4 illustrates our method where we use a direct matching method in a non-differentiable module for estimating a homography between $\mathbf{I}_{i\to j}$ and $\mathbf{I}_j$. We form a homography $\mathbf{H}_\theta$ from the network output θ by using Eqn. 6, which is used to warp a source image i into the perspective

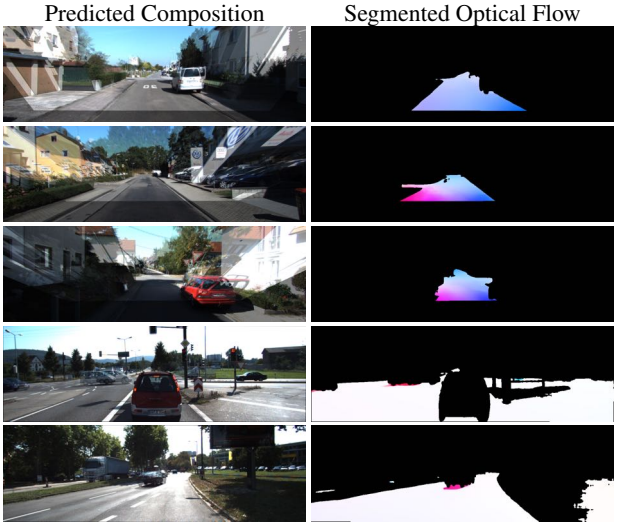

Figure 5. Performance of optical flow and segmentation networks.

of its corresponding target j. For simplicity we chose to use a pre-trained optical flow network (FlowNet2 [15]) to estimate the flow between $\mathbf{I}_j$ and $\mathbf{I}_{i\to j}$, but it is worth noting that other methods for feature matching could be employed. We compute pixel destination points $\mathbf{P}_d$ from a regular grid of source points $\mathbf{P}_s$ as $\mathbf{P}_d = \mathbf{P}_s + OF(\mathbf{I}_{i\to j}, \mathbf{I}_j)$, where OF denotes inference with FlowNet2. Multiple scene parts can contain planarity (e.g. trucks, buildings) which can conflict with the homography estimation from image pairs. We explicitly isolate the road plane by filtering non-road pixels using a pre-trained semantic segmentation network [36]:

$$\mathbf{P}_s^{(road)} = mask_{road}(\mathbf{P}_s),\ \mathbf{P}_d^{(road)} = mask_{road}(\mathbf{P}_d) \tag{9}$$

where $mask_{road}$ denotes filtering out non-road pixels.

Segmentation is computed once (offline) on the original un-warped imagery. Optical flow is applied to the whole warped and target images and subsequently masked. Further, road surfaces are still in a natural perspective after warping and distortion to non-planar regions do not seem

Method	Depth	6DoF	S.Inp	Net+	Staged	Seq. 9			Seq. 10		
						t_{err}	r_{err}	ATE	t_{err}	r_{err}	ATE
LTMVO [37]	✓	✓	✓	✓	✓	**3.49**	*0.010*	**11.30**	*5.81*	*0.018*	*11.80*
TBG [34]	✓	✓	✗	✓	✓	6.93	**0.004**	-	**4.66**	**0.006**	-
CC [24]	✓	✓	✓	✓	✗	6.92	0.018	29.0	7.97	0.031	13.77
GeoNet [32]	✓	✓	✓	✓	✗	28.72	0.098	158.4	23.90	0.090	43.04
SfM [35]	✓	✓	✓	✓	✗	8.28	0.031	24.31	12.20	0.030	20.87
SC-SfM [5]	✓	✓	✗	✗	✗	11.20	0.034	-	10.10	0.050	-
Mono2 [13]	✓	✓	✗	✗	✗	11.47	0.032	55.47	7.73	0.034	20.46
Ours (PLoss mono2-net)	✗	✗	✗	✗	✗	16.69	0.058	58.88	16.72	0.071	32.0
Ours (PLoss)	✗	✗	✗	✗	✗	11.30	0.043	28.68	11.66	0.060	16.48
Ours (HEM Test)	✗	✓	✗	✗	✗	*6.13*	0.017	*15.73*	7.38	0.033	*11.80*
Ours (HEM Train)	✗	✗	✗	✗	✓	7.14	0.023	16.27	8.58	0.031	**11.72**
Ours (HEM Train+Test)	✗	✓	✗	✗	✓	*6.53*	0.018	19.65	7.19	0.037	12.77

Table 1. Visual odometry results on KITTI. Metrics t_{err} (%) and r_{err} ($^\circ/m$) are translation and rotation error respectively.

to disrupt optical flow accuracy. Fig. 5 shows the optical flow and segmentation works well in our case (see the supplementary material for additional results and the flow key).

Optical flow with road plane semantic segmentation allows for estimating many corresponding points but which contain significant noise. Thus, we leverage an OpenCV RANSAC routine [2] to robustly fit a homographic model at training or test-time: $\mathbf{H}_{OF} = RANSAC(\mathbf{P}_s^{(road)}, \mathbf{P}_d^{(road)})$. A drawback here is that we require a reasonable initial homography from the network for a good optical flow between $\mathbf{I}_j$ and $\mathbf{I}_{i\to j}$. This is easily achieved by using our pre-trained network from perceptual loss. The homography $\mathbf{H}_{OF}$ is a transformation representing how we should update the original homography computed from our network $\mathbf{H}_\theta$ (see Eqns. (1) and (6)), which we update as: $\mathbf{H}_{i\to j}^{(OF)} = \mathbf{H}_\theta \mathbf{H}_{OF}$.

5.2. Homographic Decomposition

While it would be possible to compute a loss between $\mathbf{H}_\theta$ and $\mathbf{H}_{i\to j}^{(OF)}$ in order to provide a self-supervision signal to the network, our experience is that it is ineffective. Instead, we find that we achieve improved performance by decomposing $\mathbf{H}_{i\to j}^{(OF)}$ into camera-relative pose parameters that can be used to directly supervise the network output.

In general, any homography can be decomposed into four possible plane-relative poses via a closed form solution using the analytical method of Malis and Vargas [21] as $\mathbf{H}_{i\to j}^{(OF)} \to \{\mathbf{R}_{i\to j}^{(OF)}, \mathbf{t}_{i\to j}^{(OF)}, \mathbf{n}\}_k$, where we have camera-relative rotation and translation $\mathbf{R}_{i\to j}$ and $\mathbf{t}_{i\to j}$ respectively, plane normals $\mathbf{n}$ relevant for the homography $\mathbf{H}_{i\to j}^{(OF)}$, and $k = 0, 1, 2, 3$ which denotes the possible solutions. In practice, we obtain these four possible solutions using the OpenCV implementation [1] of this procedure.

We use domain knowledge to discount three of these four possibilities. Generally two of these normals tend to be negative for the y-component, a physical impossibility. To choose between the remaining two normals we select the normal closest to $(\mathbf{0}, \mathbf{1}, \mathbf{0})^T$ (given that cameras are always travelling approximately perpendicular to the road surface), and take the associated camera-relative poses $\{\mathbf{R}_{i\to j}^{(OF)}, \mathbf{t}_{i\to j}^{(OF)}\}$ as our refined solution. Finally, we can fine-tune our network with the loss:

$$L_{HEM} = \|\mathbf{R}_{i\to j}^{(OF)} \mathbf{R}_{\theta,i\to j}^T - \mathbf{I}\|_2 + \|\mathbf{t}_{i\to j}^{(OF)} - \mathbf{t}_{\theta,i\to j}\|_2 \quad (10)$$

where $\{\mathbf{R}_{\theta,i\to j}, \mathbf{t}_{\theta,i\to j}\}$ is the output pose from Eqn. (4).

6. Experiments

We evaluate our pose estimation pipeline using the KITTI VO dataset [12], and train on the raw dataset, omitting sequences 09 and 10 which are commonly used for testing. Training pairs are shuffled over all sequences. Our training and testing pairs consisted of target I_t and source I_s images which are separated by zero to four adjacent frames.

As outlined in Fig. 1, we train a geometric matching network in two sequential stages. Firstly, we pre-train using the perceptual loss outlined in Section 4 (referred to as *PLoss*). Secondly, we refine the *PLoss* model with the HEM loss described in Section 5 (referred to as *HEM Train*). Additionally, we apply the HEM to *PLoss* at test-time (referred to as *HEM Test*). Lastly, we apply the HEM to the *HEM Train* model at test-time (referred to as *HEM Train+Test*).

Having used these increasingly refined models to infer relative poses on test sequences, we use the method of trans-

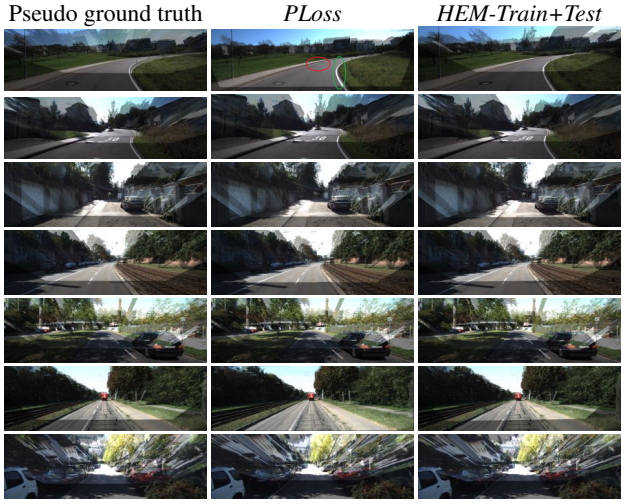

Figure 6. KITTI qualitative results, best viewed zoomed in. Images are a composition of one network input with its warped counterpart. Left: Ground truth where we assume fixed prior values for ground plane cross-projection. Middle: Our full ground-relative pose result with perceptual loss pre-training. Right: Our HEM applied at training and test-time to the *PLoss* pre-trained model.

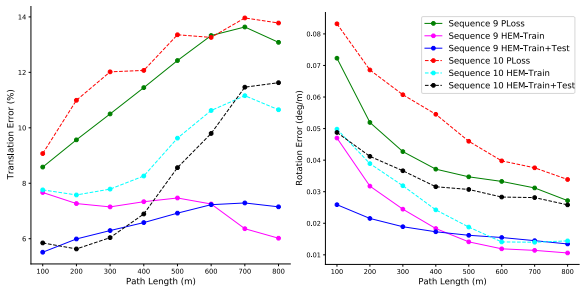

Figure 7. Translation and rotation errors by path length on sequences 09 and 10. We compare errors with pre-training with perceptual loss (*PLoss*), post-training with our HEM (*HEM-Train*) and additionally applied at test-time (*HEM-Train+Test*).

formation syncronisation by Arrigoni et al. [4] (outlined in the supplementary material) to obtain absolute poses $\widehat{\mathbf{R}}_i$ and $\widehat{\mathbf{t}}_i$ for VO evaluation. As our proposed method does not rely on any direct supervision we focus our comparison on leading methods which are fully self-supervised and only rely on a single camera.

In Table 1 we provide VO scores on sequences 09 and 10. We use the KITTI benchmark translation (%), rotation error (deg/m) and absolute trajectory RMSE (m), as in [37], for metrics. The translation and rotation errors are measured as an average positional or rotational error over all possible subsequences (100,...,800) (see [12] for details). We compare against the leading monocular self-supervised methods and show key differences between these methods which encapsulate various levels of constraint and method complexity. From left to right, methods are split between: training of a dense depth network, estimating only a 6 DoF camera-relative pose, requiring adjacent or sequential input for inference or training, training additional network(s) for dense estimation (e.g. optical flow, explainability mask, recurrent modules), and requiring a staged training process. Though we do use pre-trained networks for perceptual loss, optical flow and segmentation, this is only for inference and not trained. While LTMVO [37] and TBG [34] perform most accurately, they are more restrictive and complex in their approach. Results indicate that training with our HEM can significantly improve performance of our network and can be further refined with its application at inference time. Our method is highly competitive with leading self-supervised approaches, while remaining flexible and unconstrained. Moreover, our method is easy to train, and easy to use, and we can handle arbitrary pose changes (e.g. at opposing ends of a junction). Additionally we show that using our PLoss method with the Monodepth2 [13] pose network produces significantly worse results than the matching network we use. Additionally we note that use of a standard pixel-wise loss was difficult to train.

Close competitors (LTMVO, TBG, and CC) attempt to learn robust features for dense depth, optical flow and pose networks simultaneously to estimate *100s of thousands* of parameters - we use a *single* network to estimate only *nine* parameters (dramatically simplifying training) while *outperforming or performing very competitively*. Further, the LTMVO LSTM modules are easy to overfit, sensitive to weight initialisation, memory intensive, and can require extended training time. TBG [34] and CC [24] rely heavily on training multiple networks for scene and motion reconstruction which is challenging to train accurately.

We have evaluated trajectories (height vs horizontal distance) at specific parts of the test sequences where gradient changes more rapidly in Fig. 9, showing accurate estimation where road scenes slope strongly. For each image pair we assume the road surface is *locally* planar - we can still handle scenes with changes in gradient, and effectively are approximating a curved surface by a series of planar patches. In practice, guidance [3] for safe construction of roads with adequate camber for drainage seem to be *for the most part* a road will slope smoothly, without exceeding a maximum gradient of 1 in 12. Further, any outliers would be handled by the robust transformation syncronisation algorithm [4] which, additionally, is unreliant on scene assumptions, and handles non-planar absolute trajectories accurately.

Qualitative results are shown in Fig. 6. We use pseudo ground truth as we use camera-relative ground truth and transform it to ground-relative using assumed fixed priors for camera height and rotations. In the first example the ground truth performs poorly (perhaps due to the unknown roll relative to the ground) and in our PLoss version, features such as road lines (green) align but other features misalign globally (red), though our HEM method significantly

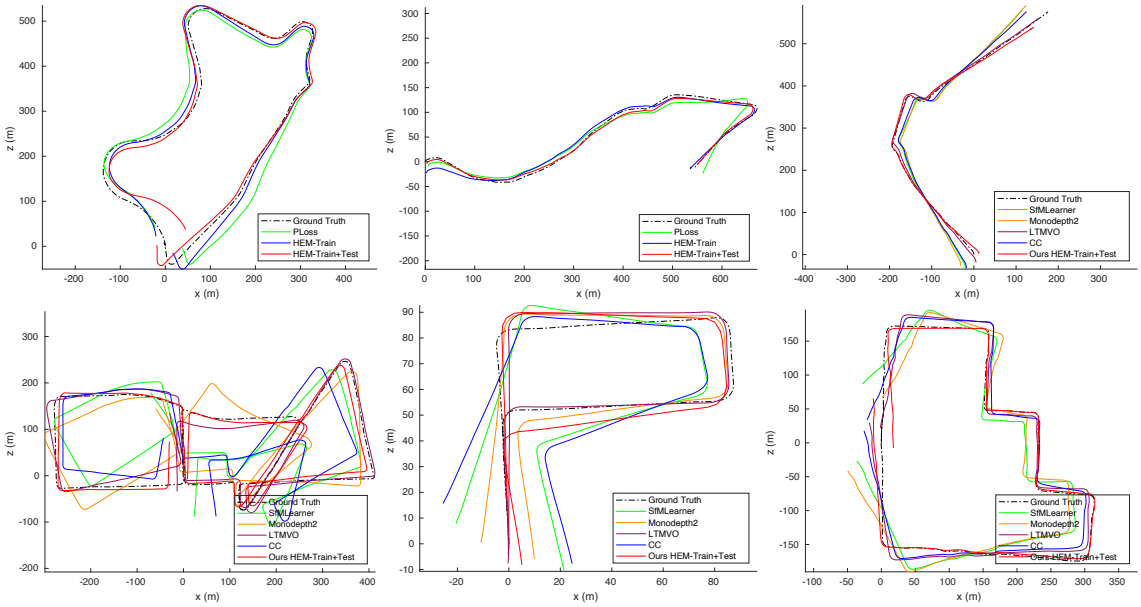

Figure 8. KITTI visual odometry trajectories for sequences 09, 10, 11, 13, 14 and 15. We compare with leading self-supervised methods for sequences 11, 13, 14 and 15 achieving very competitive performance with applying our homography estimation module at training and test-time (*HEM-Train+Test*). For sequences 09 and 10 we compare between our training methods: pre-trained perceptual loss alone (*PLoss*), and post-training with our homography estimation module (HEM) at training (*HEM-Train*) and test-time (*HEM-Train+Test*).

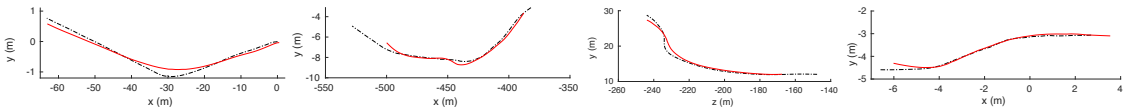

Figure 9. Vertical trajectory remains robust where gradients change rapidly. We effectively fit a series of planes to a curved road surface.

corrects these errors. Examples display increasing refinement, particularly in the penultimate example where though our *PLoss* has found a suitable rotation and failed with estimating an accurate translation, our HEM is able to correctly recover an accurate transformation. The last example shows a fail case where our HEM method is unable to achieve alignment (see manhole cover), possibly due to excessive glare in the road plane, resulting in high translational error. In summary, the *PLoss* model performs visually very well but is inclined to misaligning features in one direction, which is likely due to cases where it converges to a local minimum. The HEM refinement is able to correct these errors but can be prone to illumination issues such as dynamic shadows (e.g. see the final two examples of Fig. 5 for miscorrespondence due to shadows from moving vehicles).

In Fig. 7 we show how translation and rotation errors vary with trajectory path length on sequences 09 and 10. Generally errors are refined with each method but interestingly we observe rotation error on sequence 10 is higher after applying HEM at test-time. In Fig. 8. we show predictions for three trajectories on the benchmark test set and also sequences 09 and 10. For the benchmark sequences we compare with leading self-supervised methods. Our method is very competitive, particularly on Sequence 14 which contains imagery very different to the rest of the test sequences, and on sequence 13 which is very challenging with significant cornering and height variation.

7. Conclusions

We proposed to harness the known locally planar geometry of road-scenes with a 9D ground-relative pose to greatly simplify the learning process and enabling two novel supervision signals. We illustrated an initial appearance loss supervision from cross-projecting imagery via the ground-plane. Further, to the best of our knowledge, we are the first to employ non-differentiable in-the-loop homography refinement as a source for self-supervision for relative pose estimation. Further, we show that fitting and decomposing a homographic model directly to the road plane can generate pose pseudo-labels during training and, furthermore, at inference time this allows for additional refinement independent of the network, tackling dataset bias. We evaluated our method on the KITTI VO dataset and show very competitive results against leading self-supervised approaches which rely heavily over parameterised learning for dense depth or optical flow. For future work we plan to extend the planar constraint of our method to a more complex geometrical model and to utilise richer semantic understanding.

References

[1] OpenCV: decomposeHomographyMat. https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga7f60bdff78833d1e3fd6d9d0fd538d92. Accessed: 19-10-2022.

[2] OpenCV: findHomography. https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html#ga4abc2ece9fab9398f2e560d53c8c9780. Accessed: 19-10-2022.

[3] Roadways / site traffic control / immobilisation of vehicles. https://www.hse.gov.uk/comah/sragtech/techmeastraffic.htm. Accessed: 2022-08-26.

[4] Federica Arrigoni, Beatrice Rossi, and Andrea Fusiello. Spectral synchronization of multiple views in SE(3). *SIAM Journal on Imaging Sciences*, 9(4):1963–1990, 2016.

[5] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32:35–45, 2019.

[6] Benjamin Biggs, Oliver Boyne, James Charles, Andrew Fitzgibbon, and Roberto Cipolla. Who left the dogs out? 3D animal reconstruction with expectation maximization in the loop. In *European Conference on Computer Vision*, pages 195–211. Springer, 2020.

[7] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8001–8008, 2019.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016.

[9] Dijk and Croon. How do neural networks see depth in single images? In *ICCV*, pages 2183–2191, 2019.

[10] Haoye Dong, Xiaodan Liang, Xiaohui Shen, Bochao Wang, Hanjiang Lai, Jia Zhu, Zhiting Hu, and Jian Yin. Towards multi-pose guided virtual try-on network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9026–9035, 2019.

[11] Tuo Feng and Dongbing Gu. SGANVO: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks. *IEEE Robotics and Automation Letters*, 4(4):4431–4437, 2019.

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[13] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3838, 2019.

[14] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia Angelova. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8977–8986, 2019.

[15] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[16] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025, 2015.

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[18] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2252–2261, 2019.

[19] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. UndeepVO: Monocular visual odometry through unsupervised deep learning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7286–7291. IEEE, 2018.

[20] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. 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.

[21] Ezio Malis and Manuel Vargas. *Deeper understanding of the homography decomposition for vision-based control*. PhD thesis, INRIA, 2007.

[22] Raul Mur-Artal and Juan D Tardós. ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[23] Rene Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–1, 2020.

[24] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 12240–12249, 2019.

[25] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157, 2017.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. Pseudo RGB-D for self-improving monocular slam and depth prediction. In *European Conference on Computer Vision*, pages 437–455. Springer, 2020.

[28] Anjie Wang, Zhijun Fang, Yongbin Gao, Songchao Tan, Shanshe Wang, Siwei Ma, and Jenq-Neng Hwang. Adversarial learning for joint optimization of depth and ego-motion. *IEEE Transactions on Image Processing*, 29:4130–4142, 2020.

[29] Chen Wang, Xiang Wang, Xiao Bai, Yun Liu, and Jun Zhou. Self-supervised deep homography estimation with invertibility constraints. *Pattern Recognition Letters*, 128:355–360, 2019.

[30] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3VO: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.

[31] Qingsong Yang, Pingkun Yan, Yanbo Zhang, Hengyong Yu, Yongyi Shi, Xuanqin Mou, Mannudeep K Kalra, Yi Zhang, Ling Sun, and Ge Wang. Low-dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss. *IEEE transactions on medical imaging*, 37(6):1348–1357, 2018.

[32] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018.

[33] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 340–349, 2018.

[34] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9151–9161, 2020.

[35] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[36] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019.

[37] Yuliang Zou, Pan Ji, Quoc-Huy Tran, Jia-Bin Huang, and Manmohan Chandraker. Learning monocular visual odometry via self-supervised long-term modeling. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 710–727. Springer, 2020.

[38] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 36–53, 2018.