USegScene: Unsupervised Learning of Depth, Optical Flow and Ego-Motion with Semantic Guidance and Coupled Networks

Johan Vertens Wolfram Burgard

Abstract—In this paper we propose USegScene, a framework for semantically guided unsupervised learning of depth, optical flow and ego-motion estimation for stereo camera images using convolutional neural networks. Our framework leverages semantic information for improved regularization of depth and optical flow maps, multimodal fusion and occlusion filling considering dynamic rigid object motions as independent SE(3) transformations. Furthermore, complementary to pure photometric matching, we propose matching of semantic features, pixel-wise classes and object instance borders between the consecutive images. In contrast to previous methods, we propose a network architecture that jointly predicts all outputs using shared encoders and allows passing information across the task-domains, e.g the prediction of optical flow can benefit from the prediction of the depth. Furthermore, we explicitly learn the depth and optical flow occlusion maps inside the network, which are leveraged in order to improve the predictions in the respective regions. We present results on the popular KITTI dataset and show that our approach outperforms previous methods by a large margin.

I. INTRODUCTION

The estimation of optical flow, depth and ego-motion from stereo-images are fundamental computer vision problems with an enormous amount of robotic applications. In the field of autonomous driving it is used for solving problems like mapping [1], object tracking [2], [3], or localization [4], enabling the systems to reason about their surrounded environment.

In previous years researchers mainly approached the tasks with supervised learning, leveraging datasets like KITTI [5] or synthetic datasets such as Synthia [6] or FlyingChairs [7]. However, real world datasets with a high amount of ground-truth data are still rare, since expensive laserscanners are required and complicated procedures need to be carried out to gather such data. Furthermore, networks trained for depth, optical flow, and visual odometry on these datasets tend to overfit to the datasets specific camera-setup and thus have limited flexibility.

Recent works propose to tackle the individual problems with unsupervised methods using photometric consistency to trigger the learning [8], [9], [10], [11]. Due to the pure usage of photometric cues these approaches have problems with largely untextured regions or significant lighting changes between the consecutive frames.

Furthermore, most of the previous works struggle with occlusions between the consecutive images and use simple edge-based smoothing techniques that do not involve context information.

Other works [12], [13], [14], [15], [16] couple the tasks of estimating depth, optical flow and ego-motion by exploiting problem-dependencies within the loss function. This way the tasks are not longer learned independently but are constrained by each other in the learning process. However, these proposed approaches do not leverage context information, such as semantic segmentation that we show can help the network to improve the predictions. Additionally, these approaches tackle the individual tasks with independent and largely decoupled networks, making the complete system less efficient, and do not handle occlusions that occur due to dynamic objects.

In this paper, we propose an unsupervised learning framework for semantic aware estimation of depth, optical flow and ego-motion using stereo cameras. We jointly infer all predictions using two consecutive stereo-pairs and their corresponding semantic features in a coupled network with shared encoders. The unified nature of the network enables leveraging complementary information across the output-domains, e.g the ego-motion estimation can profit from the
estimation of depth and optical flow.

Complementarily to direct photometric losses we propose semantic feature matching that matches feature maps from a semantic segmentation network, improving predictions on textureless regions. This pre-trained segmentation network is trained to predict the class-wise semantic segmentation and additionally regresses the borders of distinct rigid instances of objects in the scene.

For flow and disparity regularization we propose a novel weighted semantic patch smoothing loss (WSPS) which smooths the optical flow or disparity map only within certain semantic classes, considering the output of the semantic network.

To cope with occlusions we further propose a method that fills these regions with information from estimated SE(3) transformations of the rigid ego-motion and independent moving rigid objects. In contrast to previous works, our network also predicts the occluded regions directly without the need of multiple inference passes and predicts correction for these regions.

We evaluate our method on the popular real world dataset KITTI [17], achieving state-of-the-art accuracies.

II. RELATED WORK

As our method comprises of the estimation of disparity, ego-motion and optical flow we list related work for each field of research individually. Finally we give an overview on previous works that focus on the combination of these different tasks. We mainly focus on works that leverage convolutional neural networks.

A. Disparity Estimation

Traditionally, disparity estimation algorithms follow the paradigm of computing costs, cost aggregation, global optimization and depth refinement [18]. Published works propose to learn parts of this pipeline with neural networks. LeCun et al. [19] propose to learn the matching costs by training a siamese-network to differentiate between small similar and dissimilar image patches. This training is done in a supervised fashion where a binary classification dataset yields the basis as training data. Similarly Shaked et al. [20] learn the matching costs with a new highway network architecture that leverages multilevel residual shortcut connections. Additionally, they propose a global disparity network to estimate the disparity confidence. As more real world [17], [21] and synthetic datasets [7], [22], [6] for depth estimation became available, many approaches have been proposed to tackle the problem in an end-to-end way. DispNet [23] learns to predict the disparity directly using a two-stream network with explicit correlation layers.

In the work from Kendall et al. [24] a new way to construct the cost volume, where all pixel-wise costs according to a specific disparity value are aggregated and then convolved with 3D convolutions.

Chang et al. [25] present a network architecture that aggregates global context information via a spatial pyramid pooling module for improving the disparity accuracy in regions with difficult appearance or lighting.

Other works [10], [26] learn to predict depth in an unsupervised fashion. They use photometric cues and forward-backward warping in order to train the network without the need of any labeled data.

Additionally, some methods [27], [28] have been proposed that solve disparity estimation and semantic segmentation within a single network. The authors show that the learned semantic embedding helps to improve the accuracy.

B. Optical Flow Estimation

In recent years CNNs became increasingly popular in the field of optical flow estimation, superseding previous methods. The authors of FlowNet [29] propose to learn pixel-wise optical flow using a convolutional neural network and synthetic data. Further works present cascaded [30] and pyramid network [31] architectures that make the predictions more accurate and the system more efficient.

Hui et al. [32] propose a fast network and uses flow regularization for further speeding up inference time while maintaining accurate and sharp predictions.

In order to get rid of the dependence of ground-truth data, Yu et al. [8] propose to learn optical flow in an unsupervised fashion using photometric losses and smoothing terms. The authors of UnFlow [9] further propose a loss that is robust to illumination changes across the consecutive images.

Recent advances [33], [34], [35] in unsupervised optical flow estimation cope with the problem of occlusion. They mask out faulty photometric losses [33], or handle occlusions by training a student network from a teacher network that has more information about occluded regions [34].

C. Odometry Estimation

Visual odometry is a well studied field of computer vision where many efficient and robust methods exist [1], [36], [37]. However, researchers [38], [39] have been investigating learning visual odometry with neural networks in order to overcome problems with texture-less environments or changing lighting conditions. Additionally these methods do not rely on apriori handcrafted feature descriptors but learn them themselves.

In recent years many works [40], [11], [41] learn visual odometry in an unsupervised way. Beside an odometry network, a depth estimator is trained that predicts the scene geometry. The predicted geometry and the ego-motion estimate can then be used to synthesize adjacent images and a photometric loss is applied to train the networks jointly.

D. Combined Approaches

Leveraging all advances in the individual fields of disparity, optical flow and ego-motion estimation, many works emerged that combine all tasks in a single unsupervised framework [13], [16], [15], [42], [14]. In GeoNet [13] firstly monocular depth and the ego-motion is learned. A rigid flow field is then computed by shifting the 3D geometry by the predicted ego-motion and projecting the motion vectors on the image plane. Since this flow prediction is only valid for static parts of the scene, they also propose a residual optical flow module that corrects for the dynamic regions. In the
work of Zou et al. [16] the rigid optical flow constraints the learning of a pixel-wise optical flow module. The constraint is not applied on dynamic regions, which they detect with forward-backward flow consistency checking. The authors of EPC++ [15] additionally add consistency constraints between consecutive depth maps and segment dynamic regions using scene flow vectors. Wang et al. [14] present a unified system for learning optical flow, visual odometry and stereo depth estimation. Instead of directly using the ego-motion estimate from a network they refine the ego-motion using the flow and depth cues.

All combined approaches exploit the relations between the individual tasks and show that joint training of all depth, optical flow and visual odometry yields overall significantly better results in each domain. However, previous works do not couple the tasks for dynamic regions while exploiting semantic cues. Additionally, none of the previous works couple all problems in the network structure directly but rather use stand-alone networks for each individual task independently. Furthermore, all previous methods do not handle regions that are occluded and dynamic. We, instead propose a system which improves the prediction quality in these regions using rigid object-instance level SE(3) transforms.

III. TECHNICAL APPROACH

In the following we describe our approach for unsupervised learning of depth, optical flow and ego-motion. We first present our unified network architectures that solves all tasks within a single network which gets additional semantic features as input. We then present semantic feature matching which enforces semantic consistency across the consecutive frames. Furthermore, we present a loss for semantic aware regularization which guides the network to predict smooth output maps within equal classes, while being sharp at class boundaries. Using a similar technique as for the semantic aware regularization we propose to constrain occluded depth regions by neighboring non-occluded regions with the same semantic class. We also present a method to distinguish between static and dynamic objects and propose novel 3D consistency losses which couples optical flow, disparity and ego-motion directly in the 3D space. Finally we propose a loss that enforces the network to fill occluded regions in optical flow maps using ego-motion and independent motions of rigid dynamic objects.

A. Unified 3D Geometry and Motion Network

While previous methods predict optical flow, disparity and ego-motion with independent networks we present a neural network architecture that predict these outputs jointly. In general, our architecture can be divided into seven different modules namely a semantic segmentation network, a shared feature encoder, a disparity estimator, an optical flow estimator, an occlusion refinement module, an ego-motion estimator and a context-module.

Our network takes advantage from the fact that the predictions of optical flow, disparity and ego-motion individually but computes them once and feeds it to the modules in a shared manner. In contrast to other works, the output of the disparity module gets passed to the optical flow and the ego-motion module, while the ego-motion module also benefits from the output of the optical flow module.

To cope with texture-less regions and occlusions, we additionally feed semantic cues from a semantic segmentation network into each of the sub-modules, making our modules context aware. The high-level semantic information is then used in the matching process, making the predictions more robust.

Our network is structured as a multi-scale pyramid neural network similar to PWCNet [31], following a course-to-fine scheme. In total we use 5 different levels $k_1, ... k_5 \in K$ where in each level a optical flow map, a disparity map and the ego-motion gets estimated. After the last level $k_5$ we further pass the disparity and optical flow predictions through a context-network.

1) Feature Encoder: We adapt the feature encoder from PWCNet [31] and add a SPP module [25] to it, which showed significant improvements in disparity estimation. Our feature encoder extracts features $f_2, f_{b_1}, f_r$ and $f_{r_1}$ corresponding to the consecutive input stereo image pairs $(l_t, l_r), (l_{t+1}, l_{r+1})$. The feature maps are predicted for the different scales $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{8}$ and $\frac{1}{16}$ and then fed into the 5 levels of the depth, optical flow and ego-motion estimators.

2) Semantic Segmentation Network: We use a ResNext50 [43] architecture with ASPP [44] and skip connections in order to predict pixel-wise semantic segmentation maps. We choose the classes in a way that they represent classes with distinct geometric complexity and rigidity characteristics. Our classes are: Road, Sidewalk, Building, Construction, Fence, Poles, Vegetation, Various Terrain, Sky, Pedestrian/Rider, Vehicle and Background. Beside the segmentation map $S$ our semantic network also predicts a regression of borders of rigid object instances $B = e^{-\frac{d}{\sigma}}$ where $d$ is the distance to the instance borders in pixels and $\sigma$ a shaping constant for the exponential, which we set to 15 during all experiments. Fig. 3 shows an example output of our segmentation network. We feed both, the segmentation maps $S_t$, $S_{b_1}$ and the border regression maps $B_t$, $B_{b_1}$, corresponding to the consecutive left images $l_t$, $l_{b_1}$, into every level of our disparity, optical flow and ego-motion modules. In order to be compatible to the corresponding scales for each level we simply re-size the segmentation and border maps bi-linearly to match with the input resolution of the respective modules.

3) Depth Estimator: In order to predict the 3D geometry as two consecutive disparity maps $D_t^l$ and $D_{b_1}^l$, we adapt the estimator of PWCNet [31] with dense blocks that was originally designed to predict optical flow. In contrast to previous works such as [25], which builds a memory expensive cost volume and use 3D convolutions, this estimator only uses 2D convolutions and fast correlation modules, resulting in a faster network with smaller memory footprint. In addition to the features of the left and right image $f^l_t$ and $f^l_{r_1}$ from the shared feature encoder of the respective level and the
FIG. 2. Overview of our USegScene architecture. Using two consecutive stereo image-pairs we predict optical flow $F$, two consecutive disparity maps $D$, ego-motion $T_{ego}$ and occlusions $O$ in respective estimators at multiple levels $k_1, k_2, k_3, k_4, k_5$ with different resolutions. Each estimator additionally leverages complementary information from a semantic segmentation network, which predicts pixel-wise class labels $S$ and a regression of instance-borders $B$. Previous disparity estimate $D^{k-1}_b$ from the previous level, we concatenate the re-sized segmentation maps $S_b^k, S_{r_b}^k$ and border maps $B_b^k, B_{r_b}^k$ to the features after the correlation layer, thus making our disparity estimator semantic aware.

4) Optical Flow Estimator: Similar to the depth estimator our optical flow estimator follows the estimator architecture of PWCNet [31], but further leverages complementary geometric and semantic cues from the depth estimators, the semantic segmentation network and the ego-motion estimators. More precisely, our optical flow estimator at level $k \in K$ takes the feature maps $f^k_{\text{lt}}$ and $f^k_{\text{lt}+1}$ from the shared feature encoder, the disparity maps $D^k_{\text{lt}}, D^k_{\text{lt}+1}$, the semantic and border maps $S^k_{\text{lt}}, S^k_{\text{lt}+1}, B^k_{\text{lt}}, B^k_{\text{lt}+1}$ and the optical flow estimate from the previous level $F^{k-1}_{\text{lt}}$ in order to predict the next optical flow estimate $F^k_{\text{lt}}$. Furthermore, using the disparity $D^k_{\text{lt}}$ of the current level and the ego-motion estimate $T^{k-1}_{\text{ego}_{t-1} \rightarrow t}$ of the previous level, the rigid flow $F^k_{\text{rt}}$, which refers to an optical flow map that is valid only for the static regions, is calculated and additionally concatenated to the inputs of the optical flow estimator. For more information on how to render the rigid flow please refer to chapter III-B. Thus, our multi-modal optical flow branch is able to learn to match consecutive features of the consecutive frames but also 3D geometry and semantic information. Since we pass the rigid flow to the estimator as prior information about the optical flow in static regions, the full prediction of the optical flow is eased, which we show in our experiments in section IV-D.

5) Occlusion Refinement Network: Since pixel-wise disparity and optical flow estimation is dependent on consistently observable image features, handling the occluded pixels in the consecutive stereo image pairs plays an important role. We therefore propose a novel module in our network that is capable to learn to detect and correct occluded regions in the disparity and optical flow maps. First, features $f^k_b$ or $f^k_D$ and the corresponding initial optical flow $\bar{D}^k_b$ or disparity map $\bar{F}^k$ are passed into a 4-layer convolutional occlusion detection module that predicts the likelihood of a pixel being occluded or not. Here, $f^k_b$ and $f^k_D$ are the features taken from before the last layer of the respective optical flow or disparity estimator modules at level $k$. We use a sigmoid function at the end of the occlusion detection module in order to have a...
Fig. 4. This figure illustrates the module GM-Net, which comprises of the depth estimator, optical flow estimator and ego-motion estimator. The GM-Net is applied to every level within the USegScene architecture predicting the disparities $D^k_l$, $D^k_{l+1}$, optical flow $F^k$ and ego-motion $T^k_{ego}$ for the current level, given the previous predictions $D^k_l$, $D^k_{l+1}$, $F^k_l$, $T^k_{ego}$ and the features from the shared encoder $f^k_l$, $f^k_{l+1}$; $f^k_0$, $f^k_{l+1}$. In the GM-Net all sub-modules leverage the predicted consecutive semantic class and instance-border information $S$, $B$. Additionally we apply our OCC-Net to each disparity and optical flow estimator, which estimates the respective occlusion maps $O^k_l$, $O^k_{l+1}$, $O^k_{R}$ and applies corrections in those regions. In our architecture the optical flow estimator can benefit from the consecutive disparity maps and the rigid flow. Complementary to the features from the shared encoder the ego-motion estimator also takes the consecutive disparity maps $D^k_l$, $D^k_{l+1}$ and the predicted optical flow $F^k$ as input.

output range between 0 and 1, where 0 means that a pixel is occluded and 1 means that the pixel is not occluded. Second, an occlusion correction network takes $f^k_l$ or $f^k_{l+1}$, the estimated occlusion map $O^k_l$ or $O^k_{l+1}$, and the optical flow/disparity map $D^k$ or $F^k$ in order to predict corrections $R^k$ for the occluded parts of the input maps. Finally, the corrections are gated by the estimated occlusion map and added to the input optical flow/disparity map resulting in the final maps $F^k$ or $D^k$. Fig. 5 shows the schematic of the module, which we refer to as OCC-Net. We apply OCC-Net after each disparity and optical flow estimator.

6) Ego-Motion Estimator: In previous works [12], [13], [11] for unsupervised learning of ego-motion the problem is approached with dense photometric matching while the respective networks do not use 3D information in their inputs. We propose a CNN architecture that additionally leverages 3D geometry given by the consecutive disparity maps $D^k_l$ and $D^k_{l+1}$. In each level $k$ we use $D^k_l$ and the ego-motion of the previous level $T^k_{ego_{k-1}}$ to warp $f^k_l$ to the reference of the first frame $I^1_l$. We then correlate the warped features $j^k_l$ with $j^k_R$ and feed the correlation result together with the semantic maps $S^k_l$, $S^k_{l+1}$, border maps $B^k_l$, $B^k_{l+1}$, optical flow $F^k$ and the consecutive disparity maps $D^k_l$, $D^k_{l+1}$ in a convolutional network block. Considering a ego-motion estimator at level $k$ our estimator consists of $5 + (k-1)$ layers, where each additional layer has a stride of two, so that its output of every level has the same dimension as the lowest level $k_1$. Similar to previous works [12] we apply global average pooling at the output of each ego-motion estimator that predicts the rigid body transformation $T_{ego_{k-1}} = \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix}$ where $R \in SO(3)$ and $t \in \mathbb{R}^3$. We predict our transformation in lie-algebraic exponential coordinates $\xi = (v^T \omega^T) \in so(3)$ and use the exponential map with small-angle approximations [45] to map from $so(3)$ to $SE(3)$. In every level we add the previous lie-algebraic exponential coordinates from the previous level to the new result. Thus, higher levels only predict residuals that further refine the ego-motion prediction. We illustrate the ego-motion estimator architecture on Fig. 5.

The complete network architecture is illustrated in Fig.
we use the depth I pixel correspondences in motion in the lowest level scheme of our network. Following PWCNet [31], as there are flow, disparity and odometry in each level. For brevity, we do average pooling and adding the ego-motion of the previous level. The result is fed together with the disparities, disparity change, optical flow, are correlated with the features of the previous left image. The correlation result is fed together with the disparities, disparity change, optical flow, rigid optical flow, semantic information and learned high-level features to the SE(3) estimator. The final ego-motion is obtained by applying global average pooling and adding the ego-motion of the previous level.

Fig. 6. This figure illustrates our ego-motion estimator. First, we compile the rigid flow using the disparity and the previous ego-motion estimate. We then use the predicted rigid flow to warp features related to the next left image to the view of the previous left image. Following, the warped features are correlated with the features of the previous left image. The correlation result is fed together with the disparities, disparity change, optical flow, rigid optical flow, semantic information and learned high-level features to the SE(3) estimator. The final ego-motion is obtained by applying global average pooling and adding the ego-motion of the previous level.

where GM-Net denotes the network branches for optical flow, disparity and odometry in each level. For brevity, we do not present all architecture parameters, but depict the general scheme of our network. Following PWCNet [31], as there are no previous predictions for optical flow, disparity or ego-motion in the lowest level $k_1$ we directly feed the feature representations into the network without having a correlation before.

B. Unsupervised Training

For training our network architecture we formulate the optical flow, disparity and ego-motion estimation as a photometric optimization problem similar to previous works [14], [15], [13]. Given the optical flow $F_T$ or disparity $D_T$ corresponding to a target image $I_T$, we can warp a source Image $I_S$ to $I_T$ by using spatial transformer networks [46].

For synthesizing $I_T$ from $I_S$ given the ego-motion $T_{egomotion}$, we use the depth $Z_T = f_h b/T$, where $f$ is the camera focal-length and $b$ is the baseline of the stereo rig, and project the pixel locations of $I_T$ to 3D. In order to get the pixel correspondences in $I_S$ we then transform the points by $T_{egomotion}$ and project them back on the image plane using the camera intrinsic matrix $K$, resulting in the corresponding pixel coordinates of $I_T$ in $I_S$.

The pixel correspondences for the synthesizing operations can be found with:

\[ p_T^F = p_T + F_T(p_T) \]
\[ p_T^D = p_T - D_T(p_T) \]
\[ p_S^F = \phi(K T_{egomotion}^{-1} \gamma(p_T | K_DT)) \]

where $p_S^F$ denote the pixel-locations in $I_S$ corresponding to $p_T$ mapped by the optical flow $F_T$, $p_S^D$ denote the pixel-locations in $I_S$ corresponding to $p_T$ mapped by the disparity $D_T$, and $p_S^F$ denote the pixel-locations in $I_S$ corresponding to $p_T$ mapped by the disparity $D$ and the ego-motion $T_{egomotion}$. Here, $\gamma(p_T | K_DT) = D_T(p_T)K^{-1}h(p_T)$ means the back-projection of pixel coordinate $p_T$ to the 3D space, where $h(p_T)$ converts $p_T$ into the homogeneous vector form. The function $\phi(x)$ divides a vector $x$ by it’s last element.

Leveraging the calculated target pixel-locations we denote a synthesized target image as $\hat{I}_T = W(I_S, M_T)$, where $W$ is the warping function that synthesizes the target bi-linearly using a displacement map $M$, such as the optical flow or disparity map.

We further denote the pixel-displacement vectors between $p_S^F$ and $p_T$ as the rigid optical flow map $F_T^S = p_S^F - p_T$, which is valid in completely static scenes.

Given the pixel correspondences, we define a loss function that calculates direct photometric differences and the structure similarity [47] between the synthesized target image $\hat{I}_T$ and the observed target image $I_T$:

\[ L_p^{OSSIM} = \alpha p \left| I_T - W(I_S, M_T) \right| + \beta_p \text{SSIM}(I_T, W(I_S, M_T)) \]

Here, $O_{M_T}$ denotes an occlusion mask that indicates which pixels are matchable between the corresponding frames, the weights $\alpha_p$ and $\beta_p$ are empirically set to 0.6 and 0.4 respectively.

1) Estimating Occlusions: We follow [13] to estimate occluded regions $O_M$, corresponding to any flow or disparity map $M$, using forward-backward consistency checks on the disparity and optical flow maps. Pixels where forward and backward optical flow/disparity contradict are assumed to be occluded.

Specifically, while training, the optical flow and disparity modules infer the output in both directions, meaning a flow or disparity map gets calculated that maps from a source image $I_S$ to a target image $I_T$ and also from $I_T$ to $I_S$.

We define the binary occlusion map as:

\[ O_{M_{fb}} = (|| \Delta M \||_2 \leq \max \{ \alpha, \beta \ || \Delta M \||_2 \} \]

where $\Delta M$ is the difference between the aligned forward and backward inference of $M$: $\Delta M = M_T - W(M_S, M_T)$.

We use the calculated binary occlusion maps in turn for self-supervised training of the occlusion detectors within the OCC-Nets. Since the forward-backward consistency checks are only needed for training, at testing time our network is able to predict occlusions without the need of running the optical flow or disparity estimators in forward and backward mode, saving computation time. While training the occlusion detectors the loss function is simply:
\[ \mathcal{L}_{\text{occ}}^M = ||O_M - O_{M_0}||_2 \] (4)

Here, \( O_M \) is the predicted occlusion map generated by the respective OCC-Net. We set \( \alpha = 0.5 \) and \( \beta = 2e^{-3} \) for inferring optical flow occlusions and \( \alpha = 1.0 \) and \( \beta = 4e^{-2} \) for inferring disparity occlusions respectively.

2) **Semantic Feature Matching:** Since pure photometric losses do not deliver meaningful information in ill posed conditions, such as low-lighting, textureless regions, transparent areas or reflections, we propose a loss function that matches complementary semantic feature representations from our semantic segmentation network. Specifically, we match source softmax segmentation values \( f_s \), the border regression map \( B \), as well as features \( f_{\text{seg}} \) from the encoder of the segmentation network, to the target view \( T \). In our case, \( f_s \) correspond to features that are inferred from consecutive images \( I_S \) and \( I_T \) and are matched by \( M_T \). As semantic features \( f_{\text{seg}} \) we take the feature maps after the second block of the ResNext-architecture of our semantic segmentation network and apply a sigmoid activation function to squeeze the values to the range between 0 and 1.

We formulate our semantic feature matching loss as:

\[ \mathcal{L}_{\text{sm}}^{M_T} = \sum_{f \in (f_s, B, f_{\text{seg}})} \theta(f_T - W(f_S, M_T)) \] (5)

An example of each semantic modality \( S, B, f_{\text{seg}} \) is shown in Fig. 3.

3) **Weighted Semantic Patch Smoothing (WSPS):** In image regions where neither photometric nor semantic cues exist, we guide the optical flow and disparity estimation by information from neighboring pixels. Many works [10], [15], [48] use smoothing techniques, which minimize the local second derivative while down-weighting the smoothness effect at positions with high gradient in the corresponding color image. Instead, we propose a smoothing method that considers larger neighborhoods and additional semantic cues.

For every pixel location \( p_m \in P_M \) of a displacement map \( M \) we define \( \mathcal{N}(p_m) \) as the set of pixel locations within a quadratic neighborhood around the location \( p_m \) where the semantic class at location \( p_m \) is equal to \( \delta(p_m) \). e.g \( \delta(p_m) = S(p_m) \). Within this neighborhood, we minimize the difference of the gradient of the flow and disparity map at the center pixel \( \nabla M(p_m) \) with all other gradients of other pixels \( \nabla M(p_n) \) of the same class.

We formulate our weighted semantic patch smoothing WSPS loss as follows:

\[ \mathcal{L}_{\text{WSPS}}^M = \sum_{p_m \in P_M} \frac{1}{N} \alpha_c \sum_{p_n \in \mathcal{N}(p_m)} \omega(p_n, p_m, Z(p_m)) \cdot \rho(\| \nabla M(p_n) - \nabla M(p_m) \|) \] (6)

where:

\[ \omega(u_n, u_m, z) = e^{-\frac{||u_n - u_m||_2^2}{4\theta(z)^2}} \] (7)

\[ \theta(z) = \frac{1}{1 + \frac{z^2}{\beta^2}} \] (8)

The function \( \omega \) weights the differences inversely proportional to the euclidean distance between the locations \( p_n \) and \( p_m \) and the depth of the center pixel. Thus, differences of gradients that are far away from each other are less penalized. Further, if the center pixel has far depth \( z \) this effect gets strengthened since the covered real area of the neighborhood becomes bigger with high distances to the camera, due to the projection-characteristics. The regularization is weighted class-dependently by \( \alpha_c \), which we empirically set to 0.1 for classes with high structural variance: *Construction, Fence, Poles, Vegetation, Various Terrain, Pedestrian/Rider*, 1.0 for classes: *Building, Various Terrain, Sky* and 8.0 for occluded pixels or classes with low structural variance or less textural cues: *Vehicle, Road, Sidewalk*. The parameter \( \eta \) defines how much pixels \( p_n \) that are far away from \( p_m \) are influenced by the loss. We set \( \eta \) empirically to 10 for all experiments.

To avoid oversmoothing between distinct object instances, we enhance our semantic segmentation map with instance information of the types *Vehicle* and *Pedestrian/Rider* using Mask-RCNN [49]. Note that Mask-RCNN is however not needed at testing time. Consequently, we consider every object instance as a individual class focusing the smoothing inside the instance. Additionally, by using a robust loss function \( \rho \) [50] (with parameters: \( \alpha = 0.5, c = 0.1 \)) we allow outliers within a class patch up to a certain level, accounting for intra-class breaks.

In comparison to standard local smoothing [10], our loss improves predictions in regions with challenging conditions, such as difficult lighting or overexposure. Pixels that lie in ill-posed regions are not just coupled to their direct neighbors, but over a large patch, regularizing over longer distances. Instead of just down-weighting the loss at pixels with high color gradient, our loss explicitly avoids oversmoothing between different semantic objects. Furthermore, our loss offers the possibility to adjust the smoothing strength over the different classes, e.g allowing less smoothing on classes with high structural variance, such as vegetation.

We illustrate WSPS on Fig. 4.
4) Semantic Depth Occlusion Filling: In previous works [28], occluded disparity values get post-processed by assigning the nearest not occluded value to the occluded ones. This method, however, does not consider the context and thus fails if the nearest valid values are not belonging to the same object that is further or nearer.

Instead, we guide the disparity predictions inside occluded areas using the semantic information at training time, similar to WSPS. Thus, we do not rely on any post-processing at testing time. We formulate our loss for guiding occluded depth pixels as:

$$
\mathcal{L}_{\text{D,occ}} = \sum_{p_m \in P_{D,oc}} \alpha c \sum_{p_n \in S(p_m)} \omega(p_n, p_m, Z(p_m)) \rho(|D(p_n) - D(p_m)|)
$$

where:

$$\omega(p_n, p_m, z) = e^{-\frac{|p_m - p_n|}{\eta m(z)}}$$

For every pixel location $p_m \in P_{D,oc}$ that is occluded by $O_{D,oc}$ we minimize the absolute difference between $D(p_m)$ and the value $D(p_n)$, where $p_n$ is a pixel location that is in $S(p_m)$, which is the set of pixel locations within a square neighborhood around $p_m$ that have the same semantic class as $p_m$ and that are not occluded. Since all stereo pairs are rectified, the stereo matches are located along the horizontal axes. We therefore down-weight the differences between $p_m$ and $p_n$ by the distance of their y-coordinates $p_{my}$ and $p_{ny}$. Further, we down-weight the influence of the loss proportional to the depth $z$ at $p_m$ and a constant factor $\eta$ similar to WSPS. As in WSPS, we leverage the robust loss function $\rho$ [50], where we set $\alpha$ to 0.5 and $c$ to 0.1. We illustrate the method in Fig. 8.

5) Sky Regularization: Since sky has infinite depth and weak visual features, previous methods often predict unreasonable optical flow and disparity values in these regions or foreground-predictions ‘bleed’ into the sky regions. To overcome this problem, we propose to minimize the optical flow or disparity in regions that belong to the class sky using the segmentation map from our semantic network. Specifically we formulate our loss as:

$$\mathcal{L}_{\text{sky}} = \frac{1}{N} \sum_{p_m | M(p_m) = \text{sky}} || M(p_m) ||_2$$

where $N$ is the number of pixels belonging to the class sky.

6) 3D Coupling: While other works [14], [16], [15] propose a minimization of the difference between the rigid flow and the optical flow within rigid parts of the image, we instead constrain the network directly in 3D space.

First, a source and a target depth map $Z_S$ and $Z_T$ are back-projected to 3D, resulting in the point-clouds $P_S$ and $P_T$. Second, we synthesize $P_T$ using the predicted optical flow $F_T$ and $P_S$. Third, we transform the synthetized pointcloud $P_T$ by the ego-motion $T_{ego_T \rightarrow S}$ in order to align it with the target pointcloud $P_T$. We then formulate a loss function that minimizes the euclidean distance between the two pointclouds for the static parts of the scene, which are not occluded by $O_{F_T}$.

We define our loss as:

$$\mathcal{L}_{\text{3D,rigid}} = \theta(Z_T) O_{F_T}(1 - C_T)(|| P_T - T_{ego_T \rightarrow S} \hat{P}_T ||_2)$$

Here, $C_T$ denotes a map which is 1 at locations that are dynamic and 0 otherwise. For far points the depth tends to be inaccurate due to the limited resolution of the cameras. Therefore, with $\theta(Z) = \frac{Z}{Z_{\text{median}}}$ we weight the euclidean loss inversely proportional to an approximated quadratic depth error as in Eq. 8 making our loss function numerically stable.

7) Dynamic Detection and Instance 3D Loss: We segment dynamic objects using the predicted disparities $D_h, D_i, D_t$ and semantic instances $I_h, I_i, I_t$ by combining the predicted ego-motion $T_{ego_S \rightarrow T}$ to the corresponding instance pointcloud $P_{it}$ by predicting the ego-motion $T_{ego_S \rightarrow T}$ with the dynamic motion of the individual object $T_{\hat{S} \rightarrow \hat{T}}$.

We define the instance-object motions as an optimization problem as follows:

$$T_{\hat{S} \rightarrow \hat{T}} = \arg\min_{T} \frac{|| TT_{ego_S \rightarrow T} \hat{P}_{\hat{T}} - T_{\hat{S} \rightarrow \hat{T}} ||_2}{T}$$

where:

$$Q = e^{-|| \hat{P}_{\hat{T}} - P_T ||_2 - \text{median}(|| \hat{P}_{\hat{T}} - P_T ||_2)}$$

Here, $Q$ is a weighting term reduces the influence of outliers. We solve the optimization problem with a differentiable SVD [51].
Using the estimated motion of the rigid instances we then present a loss function that complements our 3D coupling loss for rigid regions with additional constraints for dynamic rigid instances:

\[ \mathcal{L}_{3D_{bn}}^{S \rightarrow T} = \sum_{i \in T} \theta(D_{i}^{T}) \| T_{a}^{i} P_{T_{i}}^{T} - P_{T_{i}}^{i} \|_{2} \]  

where \( D_{i}^{T} \) refers to the average disparity of object instance \( i \) and \( T_{a}^{i} = T_{a}^{i} \) \( T_{ego}^{S \rightarrow T} \) is the apparent instance motion corresponding to instance \( i \). For solving the data association of the target and source instance masks, we use the optical flow predictions to warp the source instance masks to the target instance masks and calculate the IoU score for all combinations between source and target instances. Finally we use the hungarian algorithm [52] to assign the best association. Associations with an IoU score under 0.5 are not considered for the loss function.

Given all instance transformations \( T_{a}^{i} \), we compile the dynamic object segmentation map \( C_{T} \) where all pixels are set to 1 that belong to instances that have a speed over 15 \( km/h \). All other pixels are set to 0. Objects that belong to a non-rigid class such as person are set always to dynamic during training.

Combining both 3D losses for rigid and dynamic regions we define the overall 3D loss as:

\[ \mathcal{L}_{3D}^{S \rightarrow T} = \mathcal{L}_{3D_{rigid}}^{S \rightarrow T} + \mathcal{L}_{3D_{bn}}^{S \rightarrow T} \]  

8) Occlusion Filling within Optical Flow Maps using 3D Geometry and Motion: For filling occluded areas in the optical flow maps previous works [15] propose to fill occluded pixels with the information of the rigid flow \( F^{R} \). This method however, does handle partially occluded dynamic moving rigid objects. We propose an enhanced version that leverages all predicted instance motions \( T^{3} \) to create rigid flow maps for all pixels. We formulate our occlusion loss as:

\[ \mathcal{L}_{F_{occ}}^{S \rightarrow T} = \gamma(1 - O_{F_{T}})(1 - C_{T}) \| F_{T} - F_{R}^{T} \| + (1 - O_{F_{T}})C_{T} \sum_{i \in T} \sigma(T_{a}^{i} P_{T_{i}}^{T}, P_{i}^{T}) \| (K T_{a}^{i} P_{T_{i}}^{T} - K P_{T_{i}}^{i}) - F_{T} \|_{1} \]  

where \( \sigma(P_{S}, P_{T}) = e^{-||P_{S} - P_{T}||^{2}} \) denotes the reconstruction error between the transformed source pointcloud and the target pointcloud and serves as a measure of the quality of \( T_{a}^{i} \). The constant factor \( \gamma \) is set to 0.1 in all experiments. We present a schematic of our flow occlusion filling loss on Fig. 9.

9) Total Loss: Our final loss function is composed by a weighted combination of all previously presented losses:

\[ \mathcal{L} = \lambda_{1} \mathcal{L}_{p}^{M_{1}} + \lambda_{2} \mathcal{L}_{s}^{M_{1}} + \lambda_{3} \mathcal{L}_{WSPS}^{M_{1}} + \lambda_{4} \mathcal{L}_{sky}^{M_{1}} + \lambda_{5} \mathcal{L}_{Docc}^{M_{1}} + \lambda_{6} (\mathcal{L}_{3D}^{(r \rightarrow (r+1))} + \mathcal{L}_{3D}^{(r+1) \rightarrow r}) + \lambda_{7} (\mathcal{L}_{F_{occ}}^{(r \rightarrow (r+1))} + \mathcal{L}_{F_{occ}}^{(r+1) \rightarrow r}) + \lambda_{8} \mathcal{L}_{D_{occ}}^{M_{1}} \]  

Here, \( M_{1} \) means that the loss is applied with \( F_{t}, F_{t_{(i+1)}}, F_{i}, F_{i_{(i+1)}} \), \( D_{t} \) and \( D_{n} \), while \( M_{2} \) means that the loss applied on \( D_{t} \) and \( D_{n} \). We apply \( \mathcal{L}_{p}, \mathcal{L}_{s} \) and \( \mathcal{L}_{sky} \) on all available levels. For computational efficiency the smoothing loss \( \mathcal{L}_{WSPS} \) and depth occlusion guidance loss \( \mathcal{L}_{Docc} \) are only applied to the last two levels 4 and 5, while the 3D reconstruction loss \( \mathcal{L}_{3D} \) as well as the flow occlusion filling loss \( \mathcal{L}_{F_{occ}} \) are applied only on the last level 5. We weight the individual loss terms by the hyperparameters \( \lambda_{1} \ldots \lambda_{8} \).

IV. EXPERIMENTAL RESULTS

A. Implementation Details

We implement our method using the pytorch [53] framework. We train our network in two stages with two distinct sets of hyperparameters \( \lambda_{1} \ldots \lambda_{8} \). First, we train our network for 50k iterations with \( [10, 1, 0.001, 0.1, 0.4, 0.10] \) and an initial learning rate \( 5e^{-5} \). In the second stage we increase the weight of the WSPS, the depth occlusion filling and the flow occlusion filling terms, which yield a total parameter set as \( [10, 0.1, 0.1, 0.1, 0.5, 4, 2, 10] \). We train the second stage for 150k iterations with an initial learning rate of \( 1e^{-5} \). During all trainings we set the batch size to 2 and half the learning rate every 25k iterations. Adam [54] is used as the optimizer.

We apply random cropping as data augmentation and set the crop size to 1024x256. Note that our odometry module needs the full and non-cropped image to work. We thus pass the cropped and the full input in two copies of the network, resulting in two different outputs. We compute all losses on the outputs of the network with cropped input, but take the ego-motion estimate of the network with full input. We fix all parameters except the ones of the odometry estimator modules in the network with full-input. Thus, all training of the feature encoder, depth estimator and optical flow estimator is done on with cropped input, while optimizing the parameters of the ego-motion estimator is done with full image input.

B. Datasets

We use the KITTI raw dataset [5] to train our network for the disparity, optical flow and visual odometry tasks and adopt the corresponding training and testing sets for evaluation. In order to be comparable to other works [14],
DispNetC [23] and compare it to various state-of-the-art methods. It can be seen that our method outperforms other unsupervised methods UnOS, DispSegNet and EPC++, and to the supervised methods DispNetC and FlowNet2.

![Figure 10](image)

Table I: Comparison of disparity estimation performance with state-of-the-art approaches on the KITTI 2015 dataset.

| Type         | Approach     | Test split D1- bg | D1- fg | D1- all | Train split D1-all |
|--------------|--------------|-------------------|--------|---------|-------------------|
|              |              | 4.32              | 4.41   | 4.34    |                   |
|              | Godard et al. [10] | 5.10              | 14.551 | 6.67    | 9.19              |
|              | UnOS [14]    |                   | 16.97  | 6.33    | 6.32              |
|              | DispSegNet [28] |                   |        |         | 10.03             |
|              | SegStereo [27] |                   |        |         |                   |
|              | USegScene (ours) | 4.12              | 6.58   | 4.53    | 4.50              |

C. Disparity Evaluation

In order to evaluate the disparity estimations we train our network on the KITTI raw dataset and remove all scenes that contain images of the KITTI 2012 or 2015 optical flow/disparity sets, following [14], [28], [15]. In Table I we show the D1 error [17] on the training and testing splits of KITTI and compare it to various state-of-the-art methods. It can be seen that our method outperforms other unsupervised approaches by a large margin. Our approach performs on par with the fully supervised method DispNetC [23], which is trained on the ground-truth data from KITTI 2015. Fig. 10 illustrates multiple comparisons to other unsupervised and supervised methods and reveals that our network predicts significantly smoother disparity maps while being sharp at depth boundaries. Additionally, our disparity predictions are significantly less blurred out in occluded.
TABLE II
MODEL ABLATION STUDY FOR OPTICAL FLOW AND DEPTH ON THE KITTI 2015 DATASET. ALL MODELS ARE TRAINED ON THE MULTI-VIEW AND EVALUATED ON THE TRAINING SPLIT.

| Nr. | Depth Input | Semantic Input | Semantic Matching | WPS | 3D Coupling | Rigid Occlusion Filling | Flow All | Noc | Depth All | D1 |
|-----|-------------|----------------|--------------------|-----|-------------|------------------------|---------|-----|-----------|----|
| 1   | ✓           | ✓              | x                  | x   | x           | x                      | 85.81   | 4.10 | 74.99     | 10.69 | 9.66 |
| 2   | ✓           | ✓              | ✓                  | x   | x           | x                      | 86.56   | 3.85 | 75.55     | 10.54 | 9.50 |
| 3   | ✓           | ✓              | ✓                  | ✓   | ✓           | ✓                      | 88.28   | 3.26 | 77.49     | 9.52  | 6.71 |
| 4   | ✓           | ✓              | ✓                  | ✓   | ✓           | ✓                      | 88.37   | 3.26 | 77.54     | 9.41  | 5.84 |
| 5   | x           | ✓              | ✓                  | ✓   | ✓           | ✓                      | 88.93   | 2.29 | 82.29     | 3.80  | 6.00 |
| 6   | ✓           | x              | ✓                  | ✓   | ✓           | ✓                      | 89.30   | 2.34 | 86.96     | 3.15  | 6.30 |
| 7   | ✓           | ✓              | ✓                  | ✓   | ✓           | ✓                      | 90.62   | 2.03 | 88.56     | 2.79  | 5.73 |

| Nr. | Qualitative Flow | Qualitative Depth |
|-----|------------------|--------------------|
| 1   | ![Flow Image 1](image1) | ![Depth Image 1](image2) |
| 2   | ![Flow Image 2](image3) | ![Depth Image 2](image4) |
| 3   | ![Flow Image 3](image5) | ![Depth Image 3](image6) |
| 4   | ![Flow Image 4](image7) | ![Depth Image 4](image8) |
| 5   | ![Flow Image 5](image9) | ![Depth Image 5](image10) |
| 6   | ![Flow Image 6](image11) | ![Depth Image 6](image12) |
| 7   | ![Flow Image 7](image13) | ![Depth Image 7](image14) |

regions. Since we regularize the disparity predictions in regions that belong to sky, we do not get artifacts that occur in previous works. In contrast to other works, our network directly predicts the occlusions. An example of a occlusion prediction can be seen in Fig. 1.

In Table II we further present different configurations of our approach. We observe that the all presented losses consistently improve the results, while the WPS and 3D coupling losses yield the highest gains of 41.6% and 14.9% respectively. If the semantic inputs are neglected in the disparity estimator modules the performs drops by 9.9%, showing the benefit of complementary semantic inputs.

D. Optical Flow Evaluation

As in the disparity evaluation we train our network on the raw dataset without the scenes that hold images of the 2015/2012 optical flow training/testing sets that we use for evaluation. This enables us to be comparable to UnOS [14], EPC++ [15], DF-Net [16], Competitive-Collaboration [48], GeoNet [13] and UnFlow [9].

Since most of the dynamic object instances of the KITTI dataset are contained in the corresponding scenes of the optical flow/disparity training/testing sets, many recent works [34], [35], [57], [33], [56] leverage the multi-view extensions of those sets for training. Here, all images with ground-truth maps, and their neighbors, are removed for training.

We train our model on both sets and name the sets ‘raw’ and ‘mv’ respectively. In Table III, we compare our optical flow predictions to all other methods using the average end-point error as well as the F1 error [5].

We found that training on the multi-view dataset yields consistently better results, which may be due to higher occurrences of dynamic objects in the dataset.

Our method outperforms other methods by a significant margin.

The qualitative results, shown in Fig. 10 reveal that our method shows especially great performance in occluded regions and regions with high displacement vectors.

As in the disparity evaluation section, we also present an ablation study for the optical flow estimation in Fig. II. We observe that semantic feature matching improves the EPE in the non-occluded benchmark by 6.5%, WPS
TABLE III
COMPARISON OF OPTICAL FLOW ESTIMATION PERFORMANCE WITH STATE-OF-THE-ART APPROACHES ON THE KITTI 2015 DATASET.

| Approach                  | Split       | Test split | Train split |
|---------------------------|-------------|------------|-------------|
|                           |             | bg-all     | fg-all      | all         | EPE | all | all |
| UnFlow-CSS [9]            | raw         | -          | -           | -           | 8.10 | -   |
| GeoNet [13]               | raw         | -          | -           | -           | 10.81 | -   |
| DF-Net [16]               | raw         | -          | -           | -           | 25.70 | 8.98 |
| Competitive-Collaboration-utf [48] | raw     | -          | -           | -           | 25.27 | 5.66 |
| EPC++ (stereo) [15]       | raw         | 19.24      | 26.93       | 20.52       | 5.43 | -   |
| UnOS [14]                 | raw         | 16.93      | 23.34       | 18.00       | 5.58 | -   |
| Multi-frame [56]          | mv          | 22.67      | 24.27       | 22.94       | 6.59 | -   |
| DSTFlow [57]              | mv          | -          | -           | 39.00       | 16.79 | -   |
| OceAwareFlow [33]         | mv          | -          | -           | 31.20       | 8.88 | -   |
| DDFlow [35]               | mv          | 13.08      | 20.40       | 14.29       | 5.72 | -   |
| SelFlow [34]              | mv          | 12.68      | 21.74       | 14.19       | 4.84 | -   |
| USegScene (ours)          | raw         | 11.69      | 20.23       | 13.11       | 2.79 | 11.97 |
| USegScene (ours)          | mv          | 11.69      | 20.23       | 13.11       | 2.79 | 11.97 |

Fig. 11. This figure shows the utility of the optical flow rigid occlusion filling loss. It can be observed that the model, which is trained with rigid occlusion filling predicts much more accurate optical flow in occluded static as well as dynamic regions. Top: consecutive rgb images. Middle: optical flow prediction and corresponding error-map without rigid flow occlusion filling. Bottom: optical flow prediction and corresponding error-map with rigid flow occlusion filling.

by another 18.1% and rigid occlusion filling by further 60.5%. In contrast to the disparity case we see only little improvements when utilizing the 3D coupling loss. When looking at the EPE errors in occluded and non-occluded regions one can note that our rigid occlusion filling loss improves the EPE from 9.41 to 2.79.

In order to better understand the influence of our novel network architecture we additionally conduct experiments where we neglect the semantic inputs S and B to all modules. We can note that removing the semantic inputs yields to 1.8% performance decrease. Furthermore, we note a 7.6% decrease in the performance of the optical flow prediction when neglecting the depth and depth-dependent inputs Dl and FlR at the input of the optical flow estimator modules. This confirms that geometric inputs ease the optical flow estimation tasks.

In Fig. 11 we show an example optical flow prediction with and without rigid optical flow occlusion filling. Here, rigid optical flow occlusion filling not only drastically reduces the error in occluded static regions but also in occluded rigid dynamic regions, as we guide the prediction by individual SE(3) transformations of the dynamic rigid object.

E. Ego-Motion Estimation

We follow [15], [14] and use the KITTI odometry sequences 00-08 as training and sequences 09, 10 as validation datasets. Table IV shows a comparison to other unsupervised state-of-the-art methods using the metrics described in [17]. Note that we only use 2-frames to predict the ego-motion, while others [12], [13], [58] require more. The numbers suggest that our method outperforms other approaches by a large margin, showing the effectiveness of our method.

V. Conclusion

In this paper we propose a novel unsupervised framework for learning depth, optical flow and ego-motion. We propose a new architecture that solves all individual tasks jointly, using shared network components and leveraging semantic maps as complementary information. We present novel loss functions for complementary semantic feature matching, semantic regularization, 3D-constraints and occlusion handling. To quantify our work, we present state-of-the-art quantitative and qualitative results on the KITTI benchmark [17].

REFERENCES

[1] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. IEEE transactions on robotics, 31(5):1147–1163, 2015.
[2] Thao Dang, Christian Hoffmann, and Christoph Stiller. Fusing optical flow and stereo disparity for object tracking. In Proceedings. The IEEE 5th International Conference on Intelligent Transportation Systems, pages 112–117. IEEE, 2002.

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3404–4048, 2016.

[4] Yuval Golan, Sotiris Carlino, Abhishek Kar, and Matthew O.缰. Learning from the ground up: Supervised learning of optical flow. In Proceedings of the IEEE International Conference on Computer Vision, pages 2043–2050. IEEE, 2017.

[5] Asad Rizvi and Christian Rother. Supervision transfer for optical flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 66–75, 2017.

[6] Jia-Bin Shen, Shih-En Wei, and Jiang-Chung Chen. Pyramid stereo matching network. CoRR, abs/1803.08669, 2018.

[7] Xingzheng Cheng, Yuchao Dai, and Hongdong Li. Supervised learning for stereo matching with self-improving ability. arXiv preprint arXiv:1709.00930, 2017.

[8] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 636–651, 2018.

[9] Junming Zhang, Katherine A Skinner, Ram Vasudevan, and Matthew Johnson-Roberson. Dispsegnet: Leveraging semantics for end-to-end learning of disparity estimation from stereo imagery. IEEE Robotics and Automation Letters, 4(2):1162–1169, 2019.

[10] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philipp Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international conference on computer vision, pages 2758–2766, 2015.

[11] E. Ilg et al. Flownet 2.0: Evolution of optical flow estimation with deep networks. arXiv:1612.01925, Dec 2016.

[12] Dqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. CoRR, abs/1709.02371, 2017.

[13] Tak-Wai Hui, Xiaouu Tang, and Chen Change Loy. A lightweight optical flow cnn-revisiting data fidelity and regularization. arXiv preprint arXiv:1903.07414, 2019.

[14] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4884–4893, 2018.

[15] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selfflow: Self-supervised learning of optical flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4571–4580, 2019.

[16] Pengpeng Liu, Michael Lyu, Daniel Cremers, and Irwin King. Selfsupervised learning of optical flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4571–4580, 2019.

[17] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 8041–8047. IEEE, 2017.

[18] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. The International Journal of Robotics Research, 37(4-5):513–542, 2018.

[19] 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.

[20] Ganesh Iyer, J Krishna Murthy, Gunshi Gupta, Madhava Krishna, and Liam Paull. Geometric consistency for self-supervised end-to-end visual odometry. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 267–275, 2018.

[21] R Alcubierre, Abhishek Kar, Christian Haue, and Jitendra Malik. Learning independent object motion from unlabeled stereoscopic videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5594–5603, 2019.

[22] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5681–5690, 2017.

[23] Nikolaus Mayer, Eddy Ilg, Philipp Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4040–4048, 2016.

[24] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In Proceedings of the IEEE International Conference on Computer Vision, pages 66–75, 2017.

[25] Jia-Bin Shen and Yufeng Chen. Pyramid stereo matching network. CoRR, abs/1803.08669, 2018.
Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1492–1500, 2017.

[44] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587, 2017.

[45] Ethan Eade. Lie groups for 2d and 3d transformations.

[46] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, Advances in Neural Information Processing Systems 28, pages 2017–2025. Curran Associates, Inc., 2015.

[47] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004.

[48] 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.

[49] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[50] Jonathan T Barron. A general and adaptive robust loss function. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4331–4339, 2019.

[51] Mike Giles. An extended collection of matrix derivative results for forward and reverse mode automatic differentiation, 2008.

[52] Harold W Kuhn. The hungarian method for the assignment problem. Naval research logistics quarterly, 2(1-2):83–97, 1955.

[53] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In NIPS Autodiff Workshop, 2017.

[54] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[55] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In International Conference on Computer Vision (ICCV), 2017.

[56] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In Proceedings of the European Conference on Computer Vision (ECCV), pages 690–706, 2018.

[57] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[58] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5667–5675, 2018.