SfM-Net: Learning of Structure and Motion from Video

Sudheendra Vijayanarasimhan* svnaras@google.com
Susanna Ricco* ricco@google.com
Cordelia Schmid†* cordelia.schmid@inria.fr
Rahul Sukthankar* sukthankar@google.com
Katerina Fragkiadaki‡ katef@cs.cmu.edu

Abstract

We propose SfM-Net, a geometry-aware neural network for motion estimation in videos that decomposes frame-to-frame pixel motion in terms of scene and object depth, camera motion and 3D object rotations and translations. Given a sequence of frames, SfM-Net predicts depth, segmentation, camera and rigid object motions, converts those into a dense frame-to-frame motion field (optical flow), differentiably warps frames in time to match pixels and back-propagates. The model can be trained with various degrees of supervision: 1) self-supervised by the reprojection photometric error (completely unsupervised), 2) supervised by ego-motion (camera motion), or 3) supervised by depth (e.g., as provided by RGBD sensors). SfM-Net extracts meaningful depth estimates and successfully estimates frame-to-frame camera rotations and translations. It often successfully segments the moving objects in the scene, even though such supervision is never provided.

1. Introduction

We propose SfM-Net, a neural network that is trained to extract 3D structure, ego-motion, segmentation, object rotations and translations in an end-to-end fashion in videos, by exploiting the geometry of image formation. Given a pair of frames and camera intrinsics, SfM-Net, depicted in Figure 1, computes depth, 3D camera motion, a set of 3D rotations and translations for the dynamic objects in the scene, and corresponding pixel assignment masks. Those in turn provide a geometrically meaningful motion field (optical flow) that is used to differentiably warp each frame to the next. Pixel matching across consecutive frames, constrained by forward-backward consistency on the computed motion and 3D structure, provides gradients during training in the case of self-supervision. SfM-Net can take advantage of varying levels of supervision, as demonstrated in our experiments: completely unsupervised (self-supervised), supervised by camera motion, or supervised by depth (from Kinect).

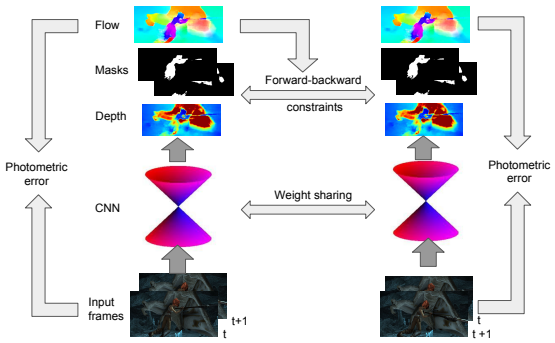

Figure 1. SfM-Net: Given a pair of frames as input, our model decomposes frame-to-frame pixel motion into 3D scene depth, 3D camera rotation and translation, a set of motion masks and corresponding 3D rigid rotations and translations. It backprojects the resulting 3D scene flow into 2D optical flow and warps accordingly to match pixels from one frame to the next. Forward-backward consistency checks constrain the estimated depth.

SfM-Net is inspired by works that impose geometric constraints on optical flow, exploiting rigidity of the visual scene, such as early low-parametric optical flow methods [2, 19, 23] or the so-called direct methods for visual SLAM (Simultaneous Localization and Mapping) that perform dense pixel matching from frame to frame while estimating a camera trajectory and depth of the pixels in the scene [9, 26]. In contrast to those, instead of optimizing directly over optical flow vectors, 3D point coordinates or camera rotation and translation, our model optimizes over neural network weights that, given a pair of frames, produce such 3D structure and motion. In this way, our method learns to estimate structure and motion, and can in principle improve as it processes more videos, in contrast to non-learning based alternatives. It can thus be made robust to lack of texture, degenerate camera motion trajectories or

*Google Research
†Inria, Grenoble, France
‡Carnegie Mellon University

dynamic objects (our model explicitly accounts for those), by providing appropriate supervision. Our work is also inspired and builds upon recent works on learning geometrically interpretable optical flow fields for point cloud prediction in time [5] and backpropagating through camera projection for 3D human pose estimation [33] or single-view depth estimation [11, 35].

In summary, our contributions are:

- A method for self-supervised learning in videos *in-the-wild*, through explicit modeling of the geometry of scene motion and image formation.
- A deep network that predicts pixel-wise depth from a single frame along with camera motion, object motion, and object masks directly from a pair of frames.
- Forward-backward constraints for learning a consistent 3D structure from frame to frame and better exploit self-supervision, extending left-right consistency constraints of [13].

We show results of our approach on KITTI [12, 21], MoSeg [4], and RGB-D SLAM [27] benchmarks under different levels of supervision. SfM-Net learns to predict structure, object, and camera motion by training on realistic video sequences using limited ground-truth annotations.

2. Related work

Back-propagating through warps and camera projection. Differentiable warping [16] has been used to learn end-to-end unsupervised optical flow [34], disparity flow in a stereo rig [13] and video prediction [24]. The closest previous works to ours are SE3-Nets [5], 3D image interpreter [33], and Garg et al.'s depth CNN [11]. SE3-Nets [5] use an actuation force from a robot and an input point cloud to forecast a set of 3D rigid object motions (rotation and translations) and corresponding pixel motion assignment masks under a static camera assumption. Our work uses similar representation of pixel motion masks and 3D motions to capture the dynamic objects in the scene. However, our work differs in that 1) we predict depth and camera motion while SE3-Nets operate on given point clouds and assume no camera motion, 2) SE3-Nets are supervised with pre-recorded 3D optical flow, while this work admits diverse and much weaker supervision, as well as complete lack of supervision, 3) SE3-Nets consider one frame and an action as input to predict the future motion, while our model uses pairs of frames as input to estimate the intra-frame motion, and 4) SE3-Nets are applied to toy or lab-like setups whereas we show results on real videos.

Wu et al. [33] learn 3D sparse landmark positions of chairs and human body joints from a single image by computing a simplified camera model and minimizing a camera re-projection error of the landmark positions. They use synthetic data to pre-train the 2D to 3D mapping of their network. Our work considers dense structure estimation and uses videos to obtain the necessary self-supervision, instead of static images. Garg et al. [11] also predict depth from a single image, supervised by photometric error. However, they do not infer camera motion or object motion, instead requiring stereo pairs with known baseline during training.

Concurrent work to ours [35] removes the constraint that the ground-truth pose of the camera be known at training time, and instead estimates the camera motion between frames using another neural network. Our approach tackles the more challenging problem of simultaneously estimating both camera and object motion.

Geometry-aware motion estimation. Motion estimation methods that exploit rigidity of the video scene and the geometry of image formation to impose constraints on optical flow fields have a long history in computer vision [2, 3, 19]. Instead of non-parametric dense flow fields [14] researchers have proposed affine or projective transformations that better exploit the low dimensionality of rigid object motion [23]. When depth information is available, motions are rigid rotations and translations [15]. Similarly, direct methods for visual SLAM having RGB [26] or RGBD [17] video as input, perform dense pixel matching from frame to frame while estimating a camera trajectory and depth of the pixels in the scene with impressive 3D point cloud reconstructions.

These works typically make a static world assumption, which makes them susceptible to the presence of moving objects in the scene. Instead, SfM-Net explicitly accounts for moving objects using motion masks and 3D translation and rotation prediction.

Learning-based motion estimation. Recent works [7, 20, 29] propose learning frame-to-frame motion fields with deep neural networks supervised with ground-truth motion obtained from simulation or synthetic movies. This enables efficient motion estimation that learns to deal with lack of texture using training examples rather than relying only on smoothness constraints of the motion field, as previous optimization methods [28]. Instead of directly optimizing over unknown motion parameters, such approaches optimize neural network weights that allow motion prediction in the presence of ambiguities in the given pair of frames.

Unsupervised learning in videos. Video holds a great potential towards learning semantically meaningful visual representations under weak supervision. Recent works have explored this direction by using videos to propagate in time semantic labels using motion constraints [25], impose temporal coherence (slowness) on the learnt visual feature [32], predict temporal evolution [30], learn temporal instance

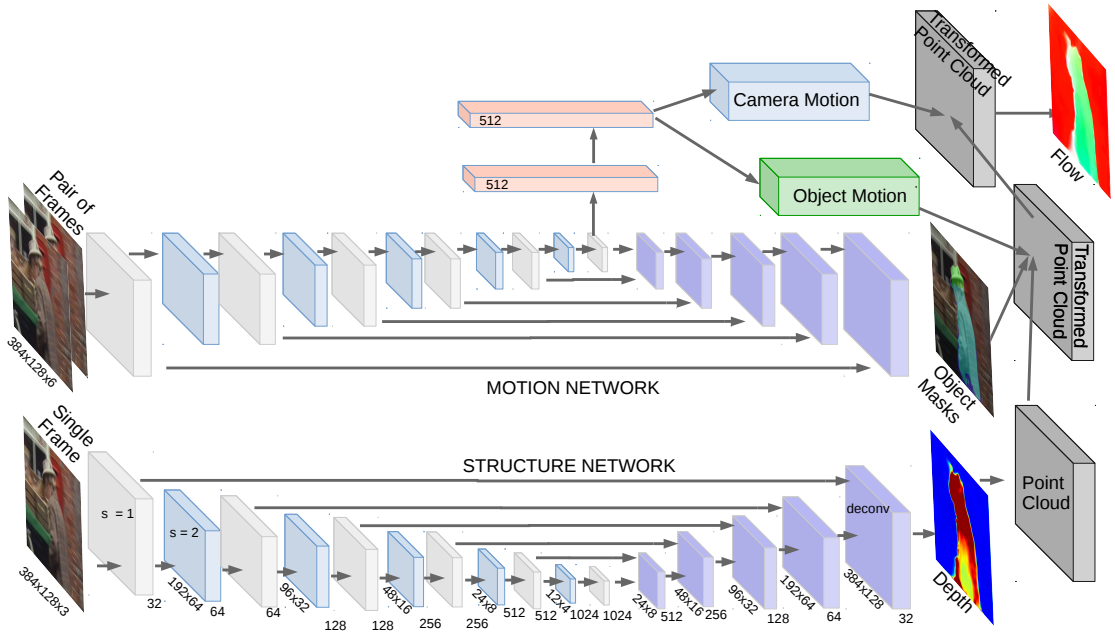

Figure 2. SfM-Net architecture. For each pair of consecutive frames I_t, I_{t+1}, a conv/deconv sub-network predicts depth d_t while another predicts a set of K segmentation masks m_t. The coarsest feature maps of the motion-mask encoder are further decoded through fully connected layers towards 3D rotations and translations for the camera and the K segmentations. The predicted depth is converted into a per frame point-cloud using estimated or known camera intrinsics. Then, it is transformed according to the predicted 3D scene flow, as composed by the 3D camera motion and independent 3D mask motions. Transformed 3D depth is projected back to the 2D next frame, and thus provides corresponding 2D optical flow fields. Differentiable backward warping maps frame I_{t+1} to I_t, and gradients are computed based on pixel errors. Forward-backward constraints are imposed by repeating this process for the inverted frame pair I_{t+1}, I_t and constraining the depths d_t and d_{t+1} to be consistent through the estimated scene motion.

level associations [31], predict temporal ordering of video frames [22], etc.

Most of those unsupervised methods are shown to be good pre-training mechanisms for object detection or classification, as done in [22, 30, 31]. In contrast and complementary to the works above, our model extracts fine-grained 3D structure and 3D motion from monocular videos with weak supervision, instead of semantic feature representations.

3. Learning SfM

3.1. SfM-Net architecture.

Our model is shown in Figure 2. Given frames I_t, $I_{t+1} \in \Re^{w \times h}$, we predict frame depth $\mathrm{d}_t \in [0, \infty)^{w \times h}$, camera rotation and translation $\{R_t^c, t_t^c\} \in SE3$, and a set of K motion masks $\mathrm{m}_t^k \in [0, 1]^{w \times h}, k \in 1, \ldots, \mathrm{K}$ that denote membership of each pixel to K corresponding rigid object motions $\{R_t^k, t_t^k\} \in SE3, k \in \{1, \ldots, \mathrm{K}\}$. Note that a pixel may be assigned to none of the motion masks, denoting that it is a background pixel and part of the static world. Using the above estimates, optical flow is computed by first generating the 3D point cloud corresponding to the image pixels using the depth map and camera intrinsics, transforming the point cloud based on camera and object rigid transformations, and back projecting the transformed 3D coordinates to the image plane. Then, given the optical flow field between initial and projected pixel coordinates, differentiable backward warping is used to map frame I_{t+1} to I_t. Forward-backward constraints are imposed by repeating this process from frame I_{t+1} to I_t and constraining the depths d_t and d_{t+1} to be consistent through the estimated scene motion. We provide details of each of these components below.

Depth and per-frame point clouds. We compute per frame depth using a standard conv/deconv subnetwork operating on a single frame (the structure network in Figure 2). We use a RELU activation at our final layer, since depth values are non-negative. Given depth d_t, we obtain the 3D point cloud $\mathbf{X}_t^i = (X_t^i, Y_t^i, Z_t^i), i \in 1, \ldots, w \times h$ corresponding to the pixels in the scene using a pinhole camera model. Let (x_t^i, y_t^i) be the column and row positions of the i^{th} pixel in frame I_t and let (c_x, c_y, f) be the camera intrinsics, then

$$\mathbf{X}_t^i = \begin{bmatrix} X_t^i \\ Y_t^i \\ Z_t^i \end{bmatrix} = \frac{d_t^i}{f} \begin{bmatrix} \frac{x_t^i}{w} - c_x \\ \frac{y_t^i}{h} - c_y \\ f \end{bmatrix} \quad (1)$$

where d_t^i denotes the depth value of the ith pixel. We use the camera intrinsics when available and revert to default values of $(0.5, 0.5, 1.0)$ otherwise. Therefore, the predicted depth will only be correct up to a scalar multiplier.

Scene motion. We compute the motion of the camera and of independently moving objects in the scene using a conv/deconv subnetwork that operates on a pair of images (the motion network in Figure 2). We depth-concatenate the pair of frames and use a series of convolutional layers to produce an embedding layer. We use two fully-connected layers to predict the motion of the camera between the frames and a predefined number K of rigid body motions that explain moving objects in the scene.

Let $\{R_t^c, t_t^c\} \in SE3$ denote the 3D rotation and translation of the camera from frame I_t to frame I_{t+1} (*relative* camera pose across consecutive frames). We represent R_t^c using an Euler angle representation as $R_t^{cx}(\alpha)R_t^{cy}(\beta)R_t^{cz}(\gamma)$ where

$$R_t^{cx}(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$R_t^{cy}(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix},$$

$$R_t^{cz}(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix},$$

and α, β, γ are the angles of rotation about the x, y, z-axes respectively. The fully-connected layers are used to predict translation parameters t^c, the pivot points of the camera rotation $p_c \in \mathbb{R}^3$ as in [5], and $\sin\alpha, \sin\beta, \sin\gamma$. These last three parameters are constrained

to be in the interval $[-1, 1]$ by using RELU activation and the minimum function.

Let $\{R_t^k, t_t^k\} \in SE3, k \in \{1, ..., \mathrm{K}\}$ denote the 3D rigid motions of up to K objects in the scene. We use similar representations as for camera motion and predict parameters using fully-connected layers on top of the same embedding E. While camera motion is a global transformation applied to all the pixels in the scene, the object motion transforms are weighted by the predicted membership probability of each pixel to each rigid motion, $\mathrm{m}_t^k \in [0, 1]^{(h \times w)}, k \in \{1, \ldots, \mathrm{K}\}$. These masks are produced by feeding the embedding layer through a deconvolutional tower. We use *sigmoid* activations at the last layer instead of *softmax* in order to allow each pixel to belong to any number of rigid body motions. When a pixel has zero activation across all K maps it is assigned to the static background whose motion is a function of the global camera motion alone. We allow a pixel to belong to multiple rigid body transforms in order to capture composition of motions, e.g., through kinematic chains, such as articulated bodies. Learning the required number of motions for a sequence is an interesting open problem. We found that we could fix $\mathrm{K} = 3$ for all experiments presented here. Note that our method can learn to ignore unnecessary object motions in a sequence by assigning no pixels to the corresponding mask.

Optical flow. We obtain optical flow by first transforming the point cloud obtained in Equation 1 using the camera and object motion rigid body transformations followed by projecting the 3D point on to the image plane using the camera intrinsics. In the following, we drop the pixel superscript i from the 3D coordinates, since it is clear we are referring to the motion transformation of the ith pixel of the tth frame. We first apply the object transformations:

$\mathbf{X}'_t = \mathbf{X}_t + \sum_{k=1}^{\mathrm{K}} \mathrm{m}_t^k(i)(R_t^k(\mathbf{X}_t - p_k) + t_t^k - \mathbf{X}_t)$. We then apply the camera transformation:

$\mathbf{X}''_t = R_t^c(\mathbf{X}'_t - p_t^c) + t_t^c$.

Finally we obtain the row and column position of the pixel in the second frame (x_{t+1}^i, y_{t+1}^i) by projecting the corresponding 3D point $\mathbf{X}''_t = (X''_t, Y''_t, Z''_t)$ back to the image plane as follows:

$$\begin{bmatrix} \frac{x_{t+1}^i}{w} \\ \frac{y_{t+1}^i}{h} \end{bmatrix} = \frac{f}{Z''_t} \begin{bmatrix} X''_t \\ Y''_t \\ f \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}$$

The flow U, V between the two frames at pixel i is then $(U_t(i), V_t(i)) = (x_{t+1}^i - x_t^i, y_{t+1}^i - y_t^i)$.

3.2. Supervision

SfM-Net inverts the image formation and extracts depth, camera and object motions that gave rise to the observed temporal differences, similar to previous SfM works [1, 6]. Such inverse problems are ill-posed as many solutions of depth, camera and object motion can give rise to the same observed frame-to-frame pixel values. A learning-based solution, as opposed to direct optimization, has the advantage of *learning to handle such ambiguities* through partial supervision of their weights or appropriate pre-training, or simply because the same coefficients (network weights) need to explain a large abundance of video data consistently. We detail the various supervision modes below and explore a subset of them in the experimental section.

Self-Supervision. Given unconstrained video, without accompanying ground-truth structure or motion information, our model is trained to minimize the photometric error

between the first frame and the second frame warped towards the first according to the predicted motion field, based on well-known brightness constancy assumptions [14]:

$$\mathcal{L}_t^{\text{color}} = \frac{1}{w\,h}\sum_{x,y} \|I_t(x,y) - I_{t+1}(x',y')\|_1$$

where $x' = x + U_t(x,y)$ and $y' = y + V_t(x,y)$. We use differentiable image warping proposed in the spatial transformer work [16] and compute color constancy loss in a fully differentiable manner.

Spatial smoothness priors. When our network is self-supervised, we add robust spatial smoothness penalties on the optical flow field, the depth, and the inferred motion maps, by penalizing the L1 norm of the gradients across adjacent pixels, as usually done in previous works [18]. For depth prediction, we penalize the norm of second order gradients in order to encourage not constant but rather smoothly changing depth values.

Forward-backward consistency constraints. We incorporate forward-backward consistency constraints between inferred scene depth in different frames as follows. Given inferred depth d_t from frame pair I_t, I_{t+1} and d_{t+1} from frame pair I_{t+1}, I_t, we ask for those to be consistent under the inferred scene motion, that is:

$$\begin{aligned}\mathcal{L}_t^{FB} = \quad & \tfrac{1}{w\,h}\sum_{x,y} |\,(\mathrm{d}_t(x,y) + W_t(x,y)) - \\ & \mathrm{d}_{t+1}(x + U_t(x,y), y + V_t(x,y))|\end{aligned}$$

where $W_t(x,y)$ is the Z component of the scene flow obtained from the point cloud transformation. Composing scene flow forward and backward across consecutive frames allows us to impose such forward-backward consistency cycles across more than one frame gaps, however, we have not yet seen empirical gain from doing so.

Supervising depth. If depth is available on parts of the input image, such as with video sequences captured by a Kinect sensor, we can use depth supervision in the form of robust depth regression:

$$\mathcal{L}_t^{depth} = \frac{1}{w\,h}\sum_{x,y} \mathrm{dmask}_t^{GT}(x,y)\cdot\|\mathrm{d}_t(x,y) - \mathrm{d}_t^{GT}(x,y)\|_1,$$

where dmask_t^{GT} denotes a binary image that signals presence of ground-truth depth.

Supervising camera motion. If ground-truth camera pose trajectories are available, we can supervise our model by computing corresponding ground-truth camera rotation and translation R_t^{c-GT}, t_t^{c-GT} from frame to frame, and constrain our camera motion predictions accordingly. Specifically, we compute the relative transformation between predicted and ground-truth camera motion $\{t_t^{\text{err}} = \text{inv}(R_t^c)(\mathrm{t}_t^{c-GT} - \mathrm{t}_t^c), R_t^{\text{err}} = \text{inv}(R_t^c)R_t^{c-GT})\}$ and minimize its rotation angle and translation norm [27]:

$$\begin{aligned}\mathcal{L}_t^{c_{\text{trans}}} &= \|t_t^{\text{err}}\|_2 \\ \mathcal{L}_t^{c_{\text{rot}}} &= \arccos\left(\min\left(1, \max\left(-1, \tfrac{\text{trace}(R_t^{err})-1}{2}\right)\right)\right)\end{aligned} \quad (2)$$

Supervising optical flow and object motion. Ground-truth optical flow, object masks, or object motions require expensive human annotation on real videos. However, these signals are available in recent synthetic datasets [20]. In such cases, our model could be trained to minimize, for example, an L1 regression loss between predicted $\{U(x,y), V(x,y)\}$ and ground-truth $\{U^{GT}(x,y), V^{GT}(x,y)\}$ flow vectors.

3.3. Implementation details

Our depth-predicting structure and object-mask-predicting motion conv/deconv networks share similar architectures but use independent weights. Each consist of a series of 3×3 convolutional layers alternating between stride 1 and stride 2 followed by deconvolutional operations consisting of a depth-to-space upsampling, concatentation with corresponding feature maps from the convolutional portion, and a 3×3 convolutional layer. Batch normalization is applied to all convolutional layer outputs. The structure network takes a single frame as input, while the motion network takes a pair of frames. We predict depth values using a 1×1 convolutional layer on top of the image-sized feature map. We use RELU activations because depths are positive and a bias of 1 to prevent small depth values. The maximum predicted depth value is further clipped at 100 to prevent large gradients. We predict object masks from the image-sized feature map of the motion network using a 1×1 convolutional layer with sigmoid activations. To encourage sharp masks we multiply the logits of the masks by a parameter that is a function of the number of step for which the network has been trained. The pivot variables are predicted as heat maps using a softmax function over all the locations in the image followed by a weighted average of the pixel locations.

4. Experimental results

The main contribution of SfM-Net is the ability to explicitly model both camera and object motion in a sequence, allowing us to train on unrestricted videos containing moving

objects. To demonstrate this, we trained self-supervised networks (using zero ground-truth supervision) on the KITTI datasets [12, 21] and on the MoSeg dataset [4]. KITTI contains pairs of frames captured from a moving vehicle in which other independently moving vehicles are visible. MoSeg contains sequences with challenging object motion, including articulated motions from moving people and animals.

KITTI. Our first experiment validates that explicitly modeling object motion is necessary to effectively learn from unconstrained videos. We evaluate unsupervised depth prediction using our models on the KITTI 2012 and KITTI 2015 datasets which contain close to 200 frame sequence and stereo pairs. We use a scale-invariant error metric (log RMSE) proposed in [8] due to the global scale ambiguitiy in monocular setups which is defined as

$$\mathcal{E}_{\text{scaleinv}} = \frac{1}{N}\sum_{x,y}\|\bar{d}(x,y)\|_2 - \left(\frac{1}{N}\sum_{x,y}\|\bar{d}(x,y)\|_1\right)^2,$$

where N is the number of pixels and $\bar{d} = (\log(\mathrm{d}) - \log(\mathrm{d}^{GT}))$ denotes the difference between the log of ground-truth and predicted depth maps. We pre-train the our unsupervised depth prediction models using adjacent frame pairs on the raw KITTI dataset which contains $\sim 42,000$ frames and train and evaluate on KITTI 2012 and 2015 which have depth ground truth.

We compare the the results of Garg et al. [11] who use stereo pairs to estimate depth. Their approach assumes the camera pose between the frames is a known constant (stereo baseline) and optimize the photometric error in order to estimate the depth. In contrast, our model considers a more challenging "in the wild" setting where we are only given sequences of frames from a video and camera pose, depth and object motion are all estimated without any form of supervision. Garg et al. report a log RMSE of 0.273 on a subset of the KITTI dataset. To compare with our approach on the full set we emulate the model of Garg et al. using our architecture by removing object masks from our network and using stereo pairs with photometric error. We also evaluate our full model on frame sequence pairs with camera motion estimation both with and without explicit object motion estimation.

Table 1 shows the log RMSE error between the ground-truth depth and the three approaches. When using stereo pairs we obtain a value of 0.31 which is on par with existing results on the KITTI benchmark (see [11]). When using frame sequence pairs instead of calibrated stereo pairs the problem becomes more difficult, as we must now infer the unknown camera and object motion between the two frames. As expected, the depth estimates learned in this scenario are less accurate, but performance is much worse when no motion masks are used. The gap between the two approaches is wider on the KITTI 2015 dataset which contains more moving objects. This shows that it is important to account for moving objects when training on videos in the wild.

Approach	Log RMSE	
	KITTI 2012	KITTI 2015
with stereo pairs	0.31	0.34
seq. with motion masks	0.45	0.41
seq. without motion masks	0.77	1.25

Table 1. RMSE of Log depth with respect to ground truth for our model with stereo pairs and with and without motion masks on sequences in KITTI 2012 and 2015 datasets. When using stereo pairs the camera pose between the frames is fixed and the model is equivalent to the approach of Garg et al. [11]. Motion masks help improve the error on both datasets but more so on the KITTI 2015 dataset which contains more moving objects.

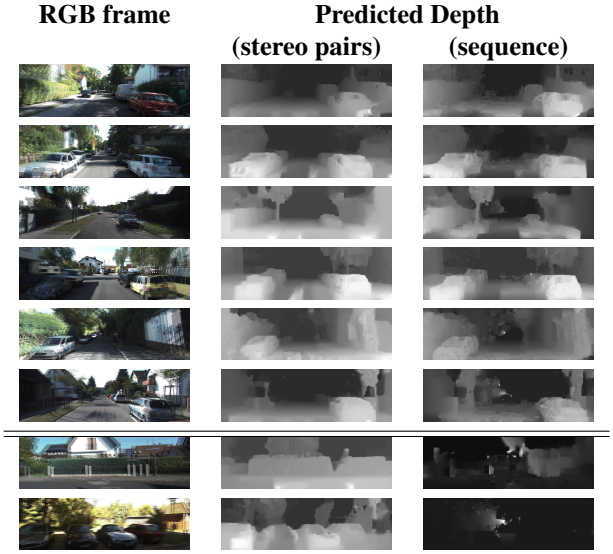

Figure 3. Qualitative comparison of the estimated depth using our unsupervised model on sequences versus using stereo pairs in the KITTI 2012 benchmark. When using stereo pairs the camera pose between the pair is constant and hence the model is equivalent to the approach of Garg et al. [11]. For sequences, our model needs to additionally predict camera rotation and translation between the two frames. The first six rows show successful predictions even without camera pose information and the last two illustrate failure cases. The failure cases show that when there is no translation between the two frames depth estimation fails whereas when using stereo pairs there is always a constant offset between the frames.

Figure 3 shows qualitative examples comparing the depth obtained when using stereo pairs with a fixed baseline and when using frame sequences without camera pose information. When there is large translation between the frames, depth estimation without camera pose information is as good as using stereo pairs. The failure cases in the last two rows show that the network did not learn to accurately

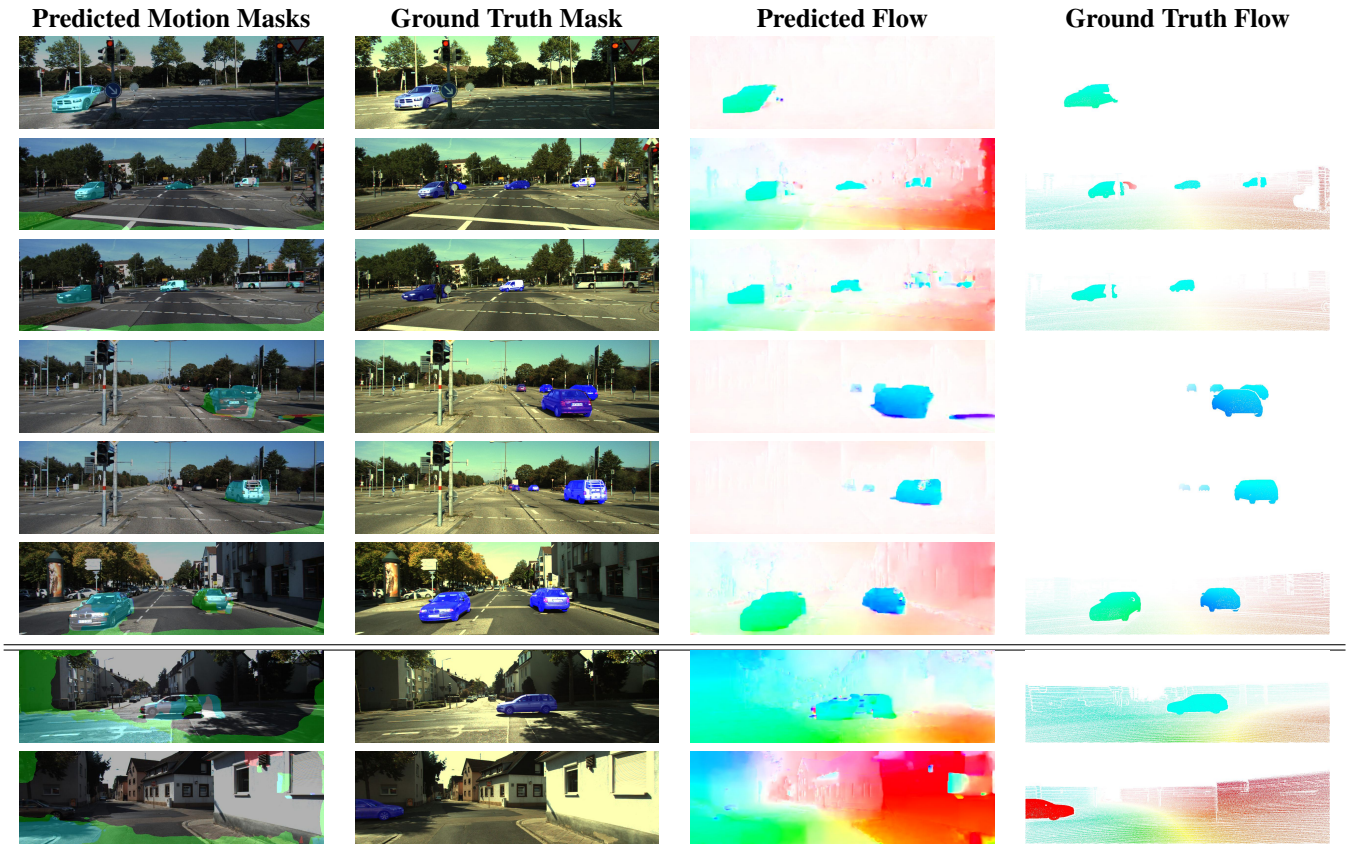

Figure 4. Ground truth segmentation and flow compared to predicted motion masks and flow from SfM-Net in KITTI 2015. The model was trained in a fully unsupervised manner. The top six rows show successful prediction and the last two show typical failure cases.

predict depth for scenes where it saw little or no translation between the frames during training. This is not the case when using stereo pairs as there is always a constant offset between the frames. Using more data could help here because it increases the likelihood of generic scenes appearing in a sequence containing interesting camera motion.

Figure 4 provides qualitative examples of the predicted motion masks and flow fields along with the ground-truth in the KITTI 2015 dataset. Often, the predicted motion masks are fairly close to the ground truth and help explain part of the motion in the scene. We notice that object masks tended to miss very small, distant moving objects. This may be due to the fact that these objects and their motions are too small to be separated from the background. The bottom two rows show cases where the predicted masks do not correspond to moving objects. In the first example, although the mask is not semantically meaningful, note that the estimated flow field is reasonable, with some mistakes in the region occluded by the moving car. In the second failure case, the moving car on the left is completely missed but the motion of the static background is well captured. This is a particularly difficult example for the self-supervised photometric loss because the moving object appears in heavy shadow.

Analysis of our failure cases suggest possible directions for improvement. Moving objects introduce significant occlusions, which should be handled carefully. Because our network has no direct supervision on object masks or object motion, it does not necessarily learn that object and camera motions should be different. These priors could be built into our loss or learned directly if some ground-truth masks or object motions are provided as explicit supervision.

MoSeg. The moving objects in KITTI are primarily vehicles, which undergo rigid-body transformations, making it a good match for our model. To verify that our network can still learn in the presence of non-rigid motion, we retrained it from scratch under self-supervision on the MoSeg dataset, using frames from all sequences. Because each motion mask corresponds to a rigid 3D rotation and translation, we do not expect a single motion mask to capture a deformable object. Instead, different rigidly moving object parts will be assigned to different masks. This is not a problem from the perspective of accurate camera motion estimation, where the important issue is distinguishing pixels whose motion is caused by the camera pose transformation directly from those whose motion is affected by indepen-

Figure 5. Motion segments computed from SfM-Net in MoSeg [4]. The model was trained in a fully unsupervised manner.

dent object motions in the scene.

Qualitative results on sampled frames from the dataset are shown in Fig. 5. Because MoSeg only contains ground-truth annotations for segmentation, we cannot quantitatively evaluate the estimated depth, camera trajectories, or optical flow fields. However, we did evaluate the quality of the object motion masks by computing Intersection over Union (IoU) for each ground-truth segmentation mask against the best matching motion mask and its complement (a total of six proposed segments in each frame, two from each of the three motion masks), averaging across frames and ground-truth objects. We obtain an IoU of 0.29 which is similar to previous unsupervised approaches for the small number of segmentation proposals we use per frame. See, for example, the last column of Figure 5 from [10], whose proposed methods for moving object proposals achieve IoU around 0.3 with four proposals. They require more than 800 proposals to reach an IoU above 0.57.

Kinect depth supervision. While the fully unsupervised results show promise, our network can benefit from extra supervision of depth or camera motion when available. The improved depth prediction given ground truth camera poses on KITTI stereo demonstrate some gain. We also experimented with adding depth supervision to improve camera motion estimation using the RGB-D SLAM dataset [27]. Given ground-truth camera pose trajectories, we estimated relative camera pose (camera motion) from each frame to the next and compare with the predicted camera motion from our model, by measuring translation and rotation error of their relative transformation, as done in the corresponding evaluation script for relative camera pose error and detailed in Eq. 2. We report camera rotation and translation error in Table 2 for each of the Freiburg1 sequences compared to the error in the benchmark's baseline trajectories. Our model was trained from scratch for each sequence and used the focal length value provided with the dataset. We observe that our results better estimate the frame-to-frame translation and are comparable for rotation.

Seq.	transl [27]	rot [27]	transl. ours	rot ours
360	0.099	0.474	0.009	1.123
plant	0.016	1.053	0.011	0.796
teddy	0.020	1.14	0.0123	0.877
desk	0.008	0.495	0.012	0.848
desk2	0.099	0.61	0.012	0.974

Table 2. Camera pose relative error from frame to frame for various video sequences of Freiburg RGBD-SLAM benchmark.

5. Conclusion

Current geometric SLAM methods obtain excellent ego-motion and rigid 3D reconstruction results, but often come at a price of extensive engineering, low tolerance to moving objects — which are treated as noise during reconstruction — and sensitivity to camera calibration. Furthermore, matching and reconstruction are difficult in low textured regions. Incorporating learning into depth reconstruction, camera motion prediction and object segmentation, while still preserving the constraints of image formation, is a promising way to robustify SLAM and visual odometry even further. However, the exact training scenario required to solve this more difficult inference problem remains an open question. Exploiting long history and far in time forward-backward constraints with visibility reasoning is an important future direction. Further, exploiting a small amount of annotated videos for object segmentation, depth, and camera motion, and combining those with an abundance of self-supervised videos, could help initialize the network weights in the right regime and facilitate learning. Many other curriculum learning regimes, including those that incorporate synthetic datasets, can also be considered.

Acknowledgements. We thank our colleagues Tinghui Zhou, Matthew Brown, Noah Snavely, and David Lowe for their advice and Bryan Seybold for his work generating synthetic datasets for our initial experiments.

References

[1] I. Akhter, Y. A. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *NIPS*, 2008.

[2] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, 1992.

[3] M. Black, Y. Yacoob, A. Jepson, and D. Fleet. Learning parameterized models of image motion. In *CVPR*, 1997.

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*. 2010.

[5] A. Byravan and D. Fox. SE3-Nets: Learning rigid body motion using deep neural networks. *CoRR*, abs/1606.02378, 2016.

[6] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. *ICCV*, 1995.

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[9] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *ECCV*, 2014.

[10] K. Fragkiadaki, P. A. Arbeláez, P. Felsen, and J. Malik. Spatio-temporal moving object proposals. *CoRR*, abs/1412.6504, 2014.

[11] R. Garg, B. V. Kumar, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[12] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012.

[13] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CoRR*, abs/1609.03677, 2016.

[14] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17, 1981.

[15] M. Hornacek, A. Fitzgibbon, and C. Rother. SphereFlow: 6 DoF scene flow from RGB-D pairs. In *CVPR*, 2014.

[16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015.

[17] C. Kerl, J. Sturm, and D. Cremers. Dense visual SLAM for RGB-D cameras. In *IROS*, 2013.

[18] N. Kong and M. J. Black. Intrinsic depth: Improving depth transfer with intrinsic images. In *ICCV*, 2015.

[19] L. Z. Manor and M. Irani. Multi-Frame Estimation of Planar Motion. *PAMI*, 22(10):1105–1116, 2000.

[20] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

[21] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[22] I. Misra, C. L. Zitnick, and M. Hebert. Unsupervised learning using sequential verification for action recognition. In *ECCV*, 2016.

[23] T. Nir, A. Bruckstein, and R. Kimmel. Over-Parameterized Variational Optical Flow. *IJCV*, 76(2):205–216, 2008.

[24] V. Patraucean, A. Handa, and R. Cipolla. Spatio-temporal video autoencoder with differentiable memory. *CoRR*, abs/1511.06309, 2015.

[25] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[26] T. Schöps, J. Engel, and D. Cremers. Semi-dense visual odometry for AR on a smartphone. In *ISMAR*, 2014.

[27] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012.

[28] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. In *CVPR*, 2010.

[29] J. Thewlis, S. Zheng, P. H. Torr, and A. Vedaldi. Fully-trainable deep matching. In *BMVC*, 2016.

[30] J. Walker, C. Doersch, A. Gupta, and M. Hebert. An uncertain future: Forecasting from static images using variational autoencoders. In *ECCV*, 2016.

[31] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015.

[32] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.*, 14(4):715–770, 2002.

[33] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3D interpreter network. In *ECCV*, 2016.

[34] J. J. Yu, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV*, 2016.

[35] T. Zhou, M. Brown, N. Snavely, and D. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR 2017*.