Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation

Anurag Ranjan[1] Varun Jampani[2] Lukas Balles[1]
Kihwan Kim [2] Deqing Sun [2] Jonas Wulff [1,3] Michael J. Black[1]
[1]Max Planck Institute for Intelligent Systems [2]NVIDIA [3]MIT
{aranjan, lballes, jwulff, black}@tuebingen.mpg.de
{vjampani, kihwank, deqings}@nvidia.com

Abstract

We address the unsupervised learning of several interconnected problems in low-level vision: single view depth prediction, camera motion estimation, optical flow, and segmentation of a video into the static scene and moving regions. Our key insight is that these four fundamental vision problems are coupled through geometric constraints. Consequently, learning to solve them together simplifies the problem because the solutions can reinforce each other. We go beyond previous work by exploiting geometry more explicitly and segmenting the scene into static and moving regions. To that end, we introduce Competitive Collaboration, *a framework that facilitates the coordinated training of multiple specialized neural networks to solve complex problems. Competitive Collaboration works much like expectation-maximization, but with neural networks that act as both competitors to explain pixels that correspond to static or moving regions, and as collaborators through a moderator that assigns pixels to be either static or independently moving. Our novel method integrates all these problems in a common framework and simultaneously reasons about the segmentation of the scene into moving objects and the static background, the camera motion, depth of the static scene structure, and the optical flow of moving objects. Our model is trained without any supervision and achieves state-of-the-art performance among joint unsupervised methods on all sub-problems.* .

1. Introduction

Deep learning methods have achieved state-of-the-art results on computer vision problems with supervision using large amounts of data [9, 18, 21]. However, for many vision problems requiring dense, continuous-valued outputs, it is either impractical or expensive to gather ground truth data [6]. We consider four such problems in this paper: single view depth prediction, camera motion estimation, optical flow, and motion segmentation. Previous work has approached these problems with supervision using real [5] and synthetic data [4]. However there is always a realism gap between synthetic and real data, and real data is limited or inaccurate. For example, depth ground truth obtained using LIDAR [6] is sparse. Furthermore, there are no sensors that provide ground truth optical flow, so all existing datasets with real imagery are limited or approximate [1, 6, 13]. Motion segmentation ground truth currently requires manual labeling of all pixels in an image [26].

Problem. Recent work has tried to address the problem of limited training data using unsupervised learning [14, 24]. To learn a mapping from pixels to flow, depth, and camera motion without ground truth is challenging because each of these problems is highly ambiguous. To address this, additional constraints are needed and the geometric relations between static scenes, camera motion, and optical flow can be exploited. For example, unsupervised learning of depth

Figure 1: **Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation.** Left, top to bottom: sample image, soft masks representing motion segmentation, estimated depth map. Right, top to bottom: static scene optical flow, segmented flow in the moving regions and combined optical flow.

This project was formerly referred by *Adversarial Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation* and camera motion has been coupled in [38, 22]. They use an explainability mask to exclude evidence that cannot be explained by the static scene assumption. Yin et al. [37] extend this to estimate optical flow as well and use forward-backward consistency to reason about unexplained pixels. These methods perform poorly on depth [38] and optical flow [37] benchmarks. A key reason is that the constraints applied here do not distinguish or segment objects that move independently, such as people and cars. More generally, not all the data in the unlabeled training set will conform to the model assumptions, and some of it might corrupt the network training. For instance, the training data for depth and camera motion should not contain independently moving objects. Similarly, for optical flow, the data should not contain occlusions, which disrupt the commonly used photometric loss.

Idea. A typical real-world scene consists of static regions, which do not move in the physical world, and moving objects [36]. Given depth and camera-motion, we can reason about the static scene in a video sequence. Optical flow, in contrast, reasons about all parts of the scene. Motion segmentation classifies a scene into static and moving regions. Our key insight is that these problems are coupled by the geometry and motion of the scene; therefore solving them jointly is synergistic. We show that by learning jointly from unlabeled data, our coupled networks can partition the dataset and use only the relevant data, resulting in more accurate results than learning without this synergy.

Approach. To address the problem of joint unsupervised learning, we introduce *Competitive Collaboration (CC)*, a generic framework in which networks learn to collaborate and compete, thereby achieving specific goals. In our specific scenario, Competitive Collaboration is a three player game consisting of two players competing for a resource that is regulated by a third player, the moderator. As shown in Figure 2, we introduce two players in our framework, the static scene reconstructor, $R = (D, C)$, that reasons about the static scene pixels using depth, D, and camera motion, C; and a moving region reconstructor, F, that reasons about pixels in the independently moving regions. These two players compete for training data by reasoning about static-scene and moving-region pixels in an image sequence. The competition is moderated by a motion segmentation network, M, that segments the static scene and moving regions, and distributes training data to the players. However, the moderator also needs training to ensure a fair competition. Therefore, the players, R and F, collaborate to train the moderator, M, such that it classifies static and moving regions correctly in alternating phases of the training cycle. This general framework is similar in spirit to expectation-maximization (EM) but is formulated for neural network training.

Contributions. In summary our contributions are: 1) We introduce *Competitive Collaboration*, an unsupervised learning framework where networks act as competitors and collaborators to reach specific goals. 2) We show that jointly training networks with this framework has a synergistic effect on their performance. 3) To our knowledge, our method is the first to use low level information like depth, camera motion and optical flow to solve a segmentation task without any supervision. 4) We achieve state-of-the-art performance on single view depth prediction and camera motion estimation among unsupervised methods. We achieve state of art performance on optical flow among unsupervised methods that reason about the geometry of the scene, and introduce the first baseline for fully unsupervised motion segmentation. We even outperform competing methods that use much larger networks [37] and multiple refinement steps such as network cascading [24]. 5) We analyze the convergence properties of our method and give an intuition of its generalization using mixed domain learning on MNIST [19] and SVHN [25] digits. All our models and code are available at https://github.com/anuragranj/cc.

2. Related Work

Our method is a three-player game, consisting of two competitors and a moderator, where the moderator takes the role of a critic and two competitors collaborate to train the moderator. The idea of collaboration can also be seen as neural expectation maximization [8] where one model is trained to distribute data to other models. For unsupervised learning, these ideas have been mainly used to model the data distribution [8] and have not been applied to unsupervised training of regression or classification problems.

There is significant recent work on supervised training of single image depth prediction [5], camera motion estimation [16] and optical flow estimation [4]. However, as labeling large datasets for continuous-valued regression tasks is not trivial, and the methods often rely on synthetic data [4, 23, 28]. Unsupervised methods have tried to independently solve for optical flow [14, 24, 35] by minimizing a photometric loss. This is highly underconstrained and thus the methods perform poorly.

More recent works [22, 32, 33, 37, 38] have approached estimation of these problems by coupling two or more problems together in an unsupervised learning framework. Zhou et al. [38] introduce joint unsupervised learning of ego-motion and depth from multiple unlabeled frames. To account for moving objects, they learn an explainability mask. However, these masks also capture model failures such as occlusions at depth discontinuities, and are hence not useful for motion segmentation. Mahjourian et al. [22] use a more explicit geometric loss to jointly learn depth and camera motion for rigid scenes. Yin et al. [37] add a refinement network to [38] to also estimate residual optical flow. The estimation of residual flow is designed to account for moving regions, but there is no coupling of the optical flow network with

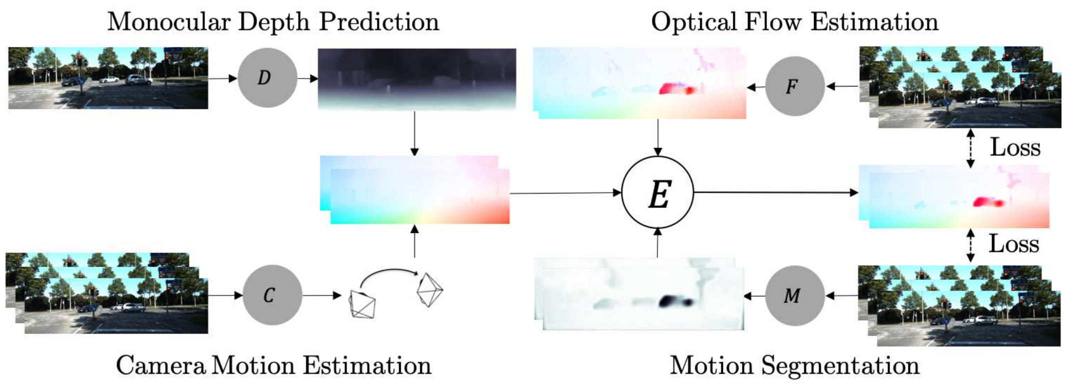

Figure 2: The network $R = (D, C)$ reasons about the scene by estimating optical flow over static regions using depth, D, and camera motion, C. The optical flow network F estimates flow over the whole image. The motion segmentation network, M, masks out static scene pixels from F to produce composite optical flow over the full image. A loss, E, using the composite flow is applied over neighboring frames to train all these models jointly.

the depth and camera motion networks. Residual optical flow is obtained using a cascaded refinement network, thus preventing other networks from using flow information to improve themselves. Therefore, recent works show good performance either on depth and camera motion [22, 37, 38] or on optical flow [24], but not on both. Zou et al. [39] exploit consistency between depth and optical flow to improve performance. The key missing piece that we add is to jointly learn the segmentation of the scene into static and independently-moving regions. This allows the networks to use geometric constraints where they apply and generic flow where they do not. Our work introduces a framework where motion segmentation, flow, depth and camera motion models can be coupled and solved jointly to reason about the complete geometric structure and motion of the scene.

Competitive Collaboration can be generalized to problems in which the models have intersecting goals where they can compete and collaborate. For example, modeling multi-modal distributions can be accomplished using our framework, whereby each competitor learns the distribution over a mode. In fact, the use of expectation-maximization (EM) in computer vision began with the optical flow problem and was used to segment the scene into "layers" [15] and was then widely applied to other vision problems.

3. Competitive Collaboration

In our context, Competitive Collaboration is formulated as a three-player game consisting of two players competing for a resource that is regulated by a moderator as illustrated in Figure 3. Consider an unlabeled training dataset $\mathcal{D} = \{\mathcal{D}_i : i \in \mathbb{N}\}$, which can be partitioned into two disjoint sets. Two players $\{R, F\}$ compete to obtain this data as a resource, and each player tries to partition $\mathcal{D}$ to minimize its loss. The partition is regulated by the moderator's output $m = M(\mathcal{D}_i), m \in [0, 1]^\Omega$, and Ω is the output domain of the competitors. The competing players minimize their loss function L_R, L_F respectively such that each player optimizes for itself but not for the group. To resolve this problem, our training cycle consists of two phases. In the first phase, we train the competitors by fixing the moderator network M and minimizing

$$E_1 = \sum_i \sum_\Omega m \cdot L_R(R(\mathcal{D}_i)) + (1-m) \cdot L_F(F(\mathcal{D}_i)), \quad (1)$$

where $\cdot$ is used to represent elementwise product throughout the paper. However, the moderator M also needs to be trained. This happens in the second phase of the training cycle. The competitors $\{R, F\}$ form a consensus and train the moderator M such that it correctly distributes the data in the next phase of the training cycle. In the collaboration phase, we fix the competitors and train the moderator by minimizing,

$$E_2 = E_1 + \sum_i \sum_\Omega L_M(\mathcal{D}_i, R, F) \quad (2)$$

where L_M is a loss that denotes a consensus between the competitors $\{R, F\}$. Competitive Collaboration can be applied to more general problems of training multiple task specific networks. In the Appendix A.1, we show the generalization of our method using an example of mixed domain learning on MNIST and SVHN digits, and analyze its convergence properties.

In the context of jointly learning depth, camera motion, optical flow and motion segmentation, the first player $R = (D, C)$ consists of the depth and camera motion networks that reason about the static regions in the scene. The second player F is the optical flow network that reasons about the moving regions. For training the competitors, the

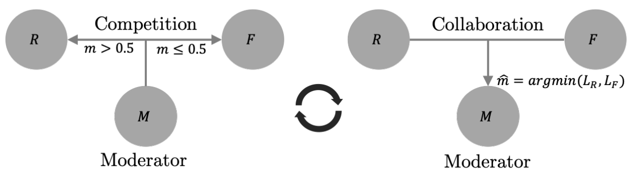

Figure 3: Training cycle of Competitive Collaboration: The moderator M drives two competitors $\{R, F\}$ (first phase, left). Later, the competitors collaborate to train the moderator to ensure fair competition in the next iteration (second phase, right).

motion segmentation network M selects networks (D, C) on pixels that are static and selects F on pixels that belong to moving regions. The competition ensures that (D, C) reasons only about the static parts and prevents moving pixels from corrupting its training. Similarly, it prevents any static pixels from appearing in the training loss of F, thereby improving its performance in the moving regions. In the second phase of the training cycle, the competitors (D, C) and F now collaborate to reason about static scene and moving regions by forming a consensus that is used as a loss for training the moderator, M. In the rest of this section, we formulate the joint unsupervised estimation of depth, camera motion, optical flow and motion segmentation within this framework.

Notation. We use $\{D_\theta, C_\phi, F_\psi, M_\chi\}$, to denote the networks that estimate depth, camera motion, optical flow and motion segmentation respectively. The subscripts $\{\theta, \phi, \psi, \chi\}$ are the network parameters. We will omit the subscripts in several places for brevity. Consider an image sequence I_-, I, I_+ with target frame I and temporally neighboring reference frames I_-, I_+. In general, we can have many neighboring frames. In our implementation, we use 5-frame sequences for C_ϕ and M_χ but for simplicity use 3 frames to describe our approach. We estimate the depth of the target frame as

$$d = D_\theta(I). \tag{3}$$

We estimate the camera motion, e, of each of the reference frames I_-, I_+ w.r.t. the target frame I as

$$e_-, e_+ = C_\phi(I_-, I, I_+). \tag{4}$$

Similarly, we estimate the segmentation of the target image into the static scene and moving regions. The optical flow of the static scene is defined only by the camera motion and depth. This generally refers to the structure of the scene. The moving regions have independent motion w.r.t. the scene. The segmentation masks corresponding to each pair of target and reference image are given by

$$m_-, m_+ = M_\chi(I_-, I, I_+), \tag{5}$$

where $m_-, m_+ \in [0, 1]^\Omega$ represent the probabilities of regions being static in spatial pixel domain, Ω. Finally, the network F_ψ estimates the optical flow. F_ψ works with 2 images at a time, and its weights are shared while estimating u_-, u_+, the backward and forward optical flow[1] respectively.

$$u_- = F_\psi(I, I_-), \qquad u_+ = F_\psi(I, I_+). \tag{6}$$

Loss. We learn the parameters of the networks $\{D_\theta, C_\phi, F_\psi, M_\chi\}$ by jointly minimizing the energy

$$E = \lambda_R E_R + \lambda_F E_F + \lambda_M E_M + \lambda_C E_C + \lambda_S E_S, \tag{7}$$

where $\{\lambda_R, \lambda_F, \lambda_M, \lambda_C, \lambda_S\}$ are the weights on the respective energy terms. The terms E_R and E_F are the objectives that are minimized by the two competitors reconstructing static and moving regions respectively. The competition for data is driven by E_M. A larger weight λ_M will drive more pixels towards the static scene reconstructor. The term E_C drives the collaboration, and E_S is a smoothness regularizer. The static scene term, E_R minimizes the photometric loss on the static scene pixels given by

$$E_R = \sum_{s \in \{+,-\}} \sum_{\Omega} \rho\Big(I, w_c(I_s, e_s, d)\Big) \cdot m_s \tag{8}$$

where Ω is the spatial pixel domain, ρ is a robust error function, and w_c warps the reference frames towards the target frame according to depth d and camera motion e. Similarly, E_F minimizes photometric loss on moving regions

$$E_F = \sum_{s \in \{+,-\}} \sum_{\Omega} \rho\Big(I, w_f(I_s, u_s)\Big) \cdot (1 - m_s) \tag{9}$$

where w_f warps the reference image using flow u. We show the formulations for w_c, w_f in the Appendix A.2 and A.3 respectively. We compute the robust error $\rho(x, y)$ as

$$\rho(x,y) = \lambda_\rho \sqrt{(x-y)^2 + \epsilon^2} + (1-\lambda_\rho)\left[1 - \frac{(2\mu_x\mu_y + c_1)(2\mu_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x + \sigma_y + c_2)}\right] \tag{10}$$

where λ_ρ is a fixed constant and $\epsilon = 0.01$. The second term is also known as the structure similarity loss (SSIM) [34] that has been used in previous work [22, 37], and μ_x, σ_x are the local mean and variance over the pixel neighborhood with $c_1 = 0.01^2$ and $c_2 = 0.03^2$.

The loss E_M minimizes the cross entropy, H, between the masks and a unit tensor regulated by λ_M

$$E_M = \sum_{s \in \{+,-\}} \sum_{\Omega} H(\mathbf{1}, m_s). \tag{11}$$

A larger λ_M gives preference to the static scene reconstructor R, biasing the scene towards being static.

[1] Note that this is different from the forward and backward optical flow in the context of two-frame estimation.

Let $\nu(e, d)$ represent the optical flow induced by camera motion e and depth d, as described in the Appendix A.2. The consensus loss E_C drives the collaboration and constrains the masks to segment moving objects by taking a consensus between flow of the static scene given by $\nu(e, d)$ and optical flow estimates from F_ψ. It is given by

$$E_C = \sum_{s\in\{+,-\}} \sum_{\Omega} H\left(\mathbb{I}_{\rho_R<\rho_F} \vee \mathbb{I}_{||\nu(e_s,d)-u_s||<\lambda_c}, m_s\right) \quad (12)$$

where $\mathbb{I} \in \{0, 1\}$ is an indicator function and equals 1 if the condition in the subscript is true. The first indicator function favors mask assignments to the competitor that achieves lower photometric error on a pixel by comparing $\rho_R = \rho(I, w_c(I_s, e_s, d))$ and $\rho_F = \rho(I, w_f(I_s, u_s))$. In the second indicator function, the threshold λ_c forces $\mathbb{I} = 1$, if the static scene flow $\nu(e, d)$ is close to the optical flow u, indicating a static scene. The symbol $\vee$ denotes logical OR between indicator functions. The consensus loss E_C encourages a pixel to be labeled as static if R has a lower photometric error than F or if the induced flow of R is similar to that of F. Finally, the smoothness term E_S acts as a regularizer on depth, segmentations and flow,

$$E_S = \sum_{\Omega} ||\lambda_e \nabla d||^2 + ||\lambda_e \nabla u_-||^2 + ||\lambda_e \nabla u_+||^2 + ||\lambda_e \nabla m_-||^2 + ||\lambda_e \nabla m_+||^2, \quad (13)$$

where $\lambda_e = e^{-\nabla I}$ (elementwise) and ∇ is the first derivative along spatial directions [29]. The term λ_e ensures that smoothness is guided by edges of the images.

Inference. The depth d and camera motion e are directly inferred from network outputs. The motion segmentation m^* is obtained by the output of mask network M_χ and the consensus between the static flow and optical flow estimates from F_χ. It is given by

$$m^* = \mathbb{I}_{m_+ \cdot m_- > 0.5} \vee \mathbb{I}_{||\nu(e_+,d)-u_+||<\lambda_c}. \quad (14)$$

The first term takes the intersection of mask probabilities inferred by M_χ using forward and backward reference frames. The second term takes a consensus between flow estimated from $R = (D_\theta, C_\phi)$ and F_ψ to reason about the masks. The final masks are obtained by taking the union of both terms. Finally, the full optical flow, u^*, between (I, I_+) is a composite of optical flows from the static scene and the independently moving regions given by

$$u^* = \mathbb{I}_{m^*>0.5} \cdot \nu(e_+, d) + \mathbb{I}_{m^* \leq 0.5} \cdot u_+. \quad (15)$$

The loss in Eq. (7) is formulated to minimize the reconstruction error of the neighboring frames. Two competitors, the static scene reconstructor $R = (D_\theta, C_\phi)$ and moving region reconstructor F_ψ minimize this loss. The reconstructor R reasons about the static scene using Eq. (8) and the reconstructor F_ψ reasons about the moving regions using Eq. (9). The moderation is achieved by the mask network, M_χ using Eq. (11). Furthermore, the collaboration between R, F is driven using Eq. (12) to train the network M_χ.

If the scenes are completely static, and only the camera moves, the mask forces (D_θ, C_ϕ) to reconstruct the whole scene. However, (D_θ, C_ϕ) are wrong in the independently moving regions of the scene, and these regions are reconstructed using F_ψ. The moderator M_χ is trained to segment static and moving regions correctly by taking a consensus from (D_θ, C_ϕ) and F_ψ to reason about static and moving parts on the scene, as seen in Eq. (12). Therefore, our training cycle has two phases. In the first phase, the moderator M_χ drives competition between two models (D_θ, C_ϕ) and F_ψ using Eqs. (8, 9). In the second phase, the competitors (D_θ, C_ϕ) and F_ψ collaborate together to train the moderator M_χ using Eqs. (11,12).

4. Experiments

Network Architecture. For the depth network, we experiment with DispNetS [38] and DispResNet where we replace convolutional blocks with residual blocks [10]. The network D_θ takes a single RGB image as input and outputs depth. For the flow network, F_ψ, we experiment with both FlowNetC [4] and PWC-Net [31]. The PWC-Net uses the multi-frame unsupervised learning framework from Janai et al. [12]. The network F_ψ computes optical flow between a pair of frames. The networks C_ϕ, M_χ take a 5 frame sequence $(I_{--}, I_-, I, I_+, I_{++})$ as input. The mask network M_χ has an encoder-decoder architecture. The encoder consists of stacked residual convolutional layers. The de-

Result: Trained Network Parameters, $(\theta, \phi, \psi, \chi)$
Define $\lambda = (\lambda_R, \lambda_F, \lambda_M, \lambda_C)$;
Randomly initialize $(\theta, \phi, \psi, \chi)$;
Update (θ, ϕ) by jointly training (D_θ, C_ϕ) with $\lambda = (1.0, 0.0, 0.0, 0.0)$;
Update ψ by training F_ψ with $\lambda = (0.0, 1.0, 0.0, 0.0)$;
Update χ by jointly training $(D_\theta, C_\phi, F_\psi, M_\chi)$ with $\lambda = (1.0, 0.5, 0.0, 0.0)$;
Loop
  Competition Step
    Update θ, ϕ by jointly training $(D_\theta, C_\phi, F_\psi, M_\chi)$ with $\lambda = (1.0, 0.5, 0.05, 0)$;
    Update ψ by jointly training $(D_\theta, C_\phi, F_\psi, M_\chi)$ with $\lambda = (0.0, 1.0, 0.005, 0)$;
  Collaboration Step
    Update χ by jointly training $(D_\theta, C_\phi, F_\psi, M_\chi)$ with $\lambda = (1.0, 0.5, 0.005, 0.3)$;
EndLoop

Algorithm 1: Network Training Algorithm

Figure 4: **Visual results.** Top to bottom: Sample image, estimated depth, soft consensus masks, motion segmented optical flow and combined optical flow.

coder has stacked upconvolutional layers to produce masks $(m_{--}, m_{-}, m_{+}, m_{++})$ of the reference frames. The camera motion network C_ϕ consists of stacked convolutions followed by adaptive average pooling of feature maps to get the camera motions $(e_{--}, e_{-}, e_{+}, e_{++})$. The networks D_θ, F_ψ, M_χ output their results at 6 different spatial scales. The predictions at the finest scale are used. The highest scale is of the same resolution as the image, and each lower scale reduces the resolution by a factor of 2. We show the network architecture details in the Appendix A.4.

Network Training. We use raw KITTI sequences [6] for training using Eigen et al.'s split [5] that is consistent across related works [5, 20, 22, 37, 38, 39]. We train the networks with a batch size of 4 and learning rate of 10^{-4} using ADAM [17] optimization. The images are scaled to 256×832 for training. The data is augmented with random scaling, cropping and horizontal flips. We use Algorithm 1 for training. Initially, we train (D_θ, C_ϕ) with only photometric loss over static pixels E_R and smoothness loss E_S while other loss terms are set to zero. Similarly, we train F_ψ independently with photometric loss over all pixels and smoothness losses. The models $(D_\theta, C_\phi), F_\psi$ at this stage are referred to as 'basic' models in our experiments. We then learn M_χ using the joint loss. We use $\lambda_R = 1.0, \lambda_F = 0.5$ for joint training because the static scene reconstructor R uses 4 reference frames in its loss, whereas the optical flow network F uses 2 frames. Hence, these weights normalize the loss per neighboring frame. We iteratively train $(D_\theta, C_\phi), F_\psi, M_\chi$ using the joint loss while keeping the other network weights fixed. The consensus weight $\lambda_C = 0.3$ is used only while training the mask network. Other constants are fixed with $\lambda_S = 0.005$, and threshold in Eq. (14), $\lambda_c = 0.001$. The constant $\lambda_\rho = 0.003$ regulates the SSIM loss and is chosen empirically. We iteratively train the competitors $(D_\theta, C_\phi), F_\psi$ and moderator M_χ for about 100,000 iterations at each step until validation error saturates.

Monocular Depth and Camera Motion Estimation. We obtain state of the art results on single view depth prediction and camera motion estimation as shown in Tables 1 and 3. The depth is evaluated on the Eigen et al. [5] split of the raw KITTI dataset [6] and camera motion is evaluated on the KITTI Odometry dataset [6]. These evaluation frameworks are consistent with previous work [5, 20, 22, 37]. All depth maps are capped at 80 meters. As shown in Table 1, by training our method only on KITTI [6], we get similar or better performance than competing methods like [37, 39] that use a much bigger Resnet-50 architecture [10] and are trained on the larger Cityscapes dataset [3]. Using Cityscapes in our training further improves our performance on depth estimation benchmarks (cs+k in Table 1).

Ablation studies on depth estimation are shown in Table 2. In the basic mode, our network architecture, DispNet for depth and camera motion estimation is most similar to [38] and this is reflected in the performance of our basic model. We get some performance improvements by adding the SSIM loss [34]. However, we observe that using the Competitive Collaboration (CC) framework with a joint loss results in larger performance gains in both tasks. Further improvements are obtained by using a better network architecture, DispResNet. Greater improvements in depth estimation are obtained when we use a better network for flow, which shows that improving on one task improves the performance of the other in the CC framework (row 4 vs 5 in Table 2).

The camera motion estimation also shows similar performance trends as shown in Table 3. Using a basic model, we achieve similar performance as the baseline [38], which improves with the addition of the SSIM loss. Using the CC framework leads to further improvements in performance.

In summary, we show that joint training using CC boosts performance of single view depth prediction and camera motion estimation. We show qualitative results in Figure 4. In the Appendix, we show additional evaluations using Make3D dataset [30] (A.6) and more qualitative results (A.5).

Method	Data	Error				Accuracy, δ		
		AbsRel	SqRel	RMS	RMSlog	<1.25	$<1.25^2$	$<1.25^3$
Eigen et al. [5] coarse	k	0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen et al. [5] fine	k	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu et al. [20]	k	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhou et al. [38]	cs+k	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian et al. [22]	cs+k	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Geonet-Resnet [37]	cs+k	0.153	1.328	5.737	0.232	0.802	0.934	0.972
DF-Net [39]	cs+k	0.146	1.182	5.215	**0.213**	0.818	**0.943**	**0.978**
CC (ours)	cs+k	**0.139**	**1.032**	**5.199**	**0.213**	**0.827**	**0.943**	0.977
Zhou et al.* [38]	k	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Mahjourian et al. [22]	k	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Geonet-VGG [37]	k	0.164	1.303	6.090	0.247	0.765	0.919	0.968
Geonet-Resnet [37]	k	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Godard et el. [7]	k	0.154	1.218	5.699	0.231	0.798	0.932	0.973
DF-Net [39]	k	0.150	1.124	5.507	0.223	0.806	0.933	0.973
CC (ours)	k	**0.140**	**1.070**	**5.326**	**0.217**	**0.826**	**0.941**	**0.975**

Table 1: Results on Depth Estimation. Supervised methods are shown in the first rows. Data refers to the training set: Cityscapes (cs) and KITTI (k). Zhou el al.* shows improved results from their github page.

Method	Data	Net D	Net F	Error				Accuracy, δ		
				AbsRel	SqRel	RMS	RMSlog	<1.25	$<1.25^2$	$<1.25^3$
Basic	k	DispNet	-	0.184	1.476	6.325	0.259	0.732	0.910	0.967
Basic + ssim	k	DispNet	-	0.168	1.396	6.176	0.244	0.767	0.922	0.971
CC + ssim	k	DispNet	FlowNetC	0.148	1.149	5.464	0.226	0.815	0.935	0.973
CC + ssim	k	DispResNet	FlowNetC	0.144	1.284	5.716	0.226	0.822	0.938	0.973
CC + ssim	k	DispResNet	PWC Net	0.140	1.070	5.326	0.217	0.826	0.941	0.975
CC + ssim	cs+k	DispResNet	PWC Net	**0.139**	**1.032**	**5.199**	**0.213**	**0.827**	**0.943**	**0.977**

Table 2: Ablation studies on Depth Estimation. Joint training using Competitive Collaboration and better architectures improve the results. The benefits of CC can be seen when depth improves by using a better network for flow (row 4 vs 5).

Method	Sequence 09	Sequence 10
ORB-SLAM (full)	0.014 ± 0.008	0.012 ± 0.011
ORB-SLAM (short)	0.064 ± 0.141	0.064 ± 0.130
Mean Odometry	0.032 ± 0.026	0.028 ± 0.023
Zhou et al. [38]	0.016 ± 0.009	0.013 ± 0.009
Mahjourian et al. [22]	0.013 ± 0.010	0.012 ± 0.011
Geonet [37]	$\mathbf{0.012 \pm 0.007}$	0.012 ± 0.009
DF-Net [39]	0.017 ± 0.007	0.015 ± 0.009
Basic (ours)	0.022 ± 0.010	0.018 ± 0.011
Basic + ssim (ours)	0.017 ± 0.009	0.015 ± 0.009
CC + ssim (ours)	$\mathbf{0.012 \pm 0.007}$	$\mathbf{0.012 \pm 0.008}$

Table 3: Results on Camera Pose Estimation.

Optical Flow Estimation. We compare the performance of our approach with competing methods using the KITTI 2015 training set [6] to be consistent with previous work [24, 37]. We obtain state of the art performance among joint methods as shown in Table 4. Unsupervised fine tuning (CC-uft) by setting $\lambda_M = 0.02$ gives more improvements than CC as masks now choose the best flow between R and F without being overconstrained to choose R. In contrast, UnFlow-CSS [24] uses 3 cascaded networks to refine optical flow at each stage. Geonet [37] and DF-Net [39] are more similar to our architecture but use a larger ResNet-50 architecture. Back2Future [12] performs better than our method in terms of outlier error, but not in terms of average end point error due to use of additional data.

In Table 5, we observe that training the static scene reconstructor R or moving region reconstructor F independently leads to worse performance. This happens because R can not reason about dynamic moving objects in the scene. Similarly F is not as good as R for reasoning about static parts of the scene, especially in occluded regions. Using them together, and compositing the optical flow from both as shown in Eq. (15) leads to a large improvement in performance. Moreover, using better network architectures further improves the performance under the CC framework. We show qualitative results in Figure 4 and in the Appendix A.5.

Method	Train EPE	Train Fl	Test Fl
FlowNet2 [11]	10.06	30.37 %	-
SPyNet [27]	20.56	44.78%	-
UnFlow-C [24]	8.80	28.94%	29.46%
UnFlow-CSS [24]	8.10	**23.27%**	-
Back2Future [12]	**6.59**	-	**22.94%**
Back2Future* [12]	7.04	24.21%	-
Geonet [37]	10.81	-	-
DF-Net [39]	8.98	26.01%	25.70%
CC (ours)	6.21	26.41%	-
CC-uft (ours)	**5.66**	**20.93%**	**25.27%**

Table 4: Results on Optical Flow. We also compare with supervised methods (top 2 rows) that are trained on synthetic data only; unsupervised methods specialized for optical flow (middle 3 rows) and joint methods that solve more than one task (bottom 4 rows). * refers to our Pytorch implementation used in our framework which gives slightly lower accuracy.

Method	Net D	Net F	Average EPE SP	Average EPE MP	Average EPE Total
R	DispNet	-	7.51	32.75	13.54
F	-	FlowNetC	15.32	6.20	14.68
CC	DispNet	FlowNetC	6.35	6.16	7.76
CC	DispResNet	PWC Net	5.67	5.04	6.21

Table 5: Ablation studies on Flow estimation. SP, MP refer to static scene and moving region pixels. EPE is computed over KITTI 2015 training set. R, F are trained independently without CC.

Motion Segmentation. We evaluate the estimated motion segmentations using the KITTI 2015 training set [6] that provides ground truth segmentation for moving cars. Since our approach does not distinguish between different semantic classes while estimating segmentation, we evaluate segmentations only on car pixels. Specifically, we only consider car pixels and compute Intersection over Union (IoU) scores for moving and static car pixels. In Table 6, we show the IoU scores of the segmentation masks obtained using our technique under different conditions. We refer to the masks obtained with the motion segmentation network $\left(\mathbf{I}_{m_{-}m_{+}>0.5}\right)$ as 'MaskNet' and refer to the masks obtained with flow consensus $\left(\mathbf{I}_{||\nu(e_{+},d)-u_{+}||<\lambda_c}\right)$ as 'Consensus'. The final motion segmentation masks m^* obtained with the intersection of the above two estimates are referred to as 'Joint' (Eq. 14). IoU results indicate substantial IoU improvements with 'Joint' masks compared to both 'MaskNet' and 'Consensus' masks, illustrating the complementary nature of different masks. Qualitative results are shown in Figure 4 and in the Appendix A.5.

We thank Frederik Kunstner for verifying the proofs, Clément Pinard for his code, Georgios Pavlakos for paper revisions, Joel Janai for optical flow visualizations, and Clément Gorard for Make3d evaluation code. MJB is a part-time employee of Amazon; has financial interests in Amazon and

	Overall	Static Car	Moving Car
MaskNet	41.64	30.56	52.71
Consensus	51.52	47.30	55.74
Joint	**56.94**	**55.77**	**58.11**

Table 6: Motion Segmentation Results. Intersection Over Union (IoU) scores on KITTI2015 training dataset images computed over car pixels.

5. Conclusions and Discussion

Typically, learning to infer depth from a single image requires training images with ground truth depth scans, and learning to compute optical flow relies on synthetic data, which may not generalize to real image sequences. For static scenes, observed by a moving camera, these two problems are related by camera motion; depth and camera motion completely determine the 2D optical flow. This holds true over several frames if the scene is static and only the camera moves. Thus by combining depth, camera, and flow estimation, we can learn single-image depth by using information from several frames during training. This is particularly critical for unsupervised training since both depth and optical flow are highly ill-posed. Combining evidence from multiple tasks and multiple frames helps to synergistically constrain the problem. This alone is not enough, however, as real scenes contain multiple moving objects that do not conform to static scene geometry. Consequently, we also learn to segment the scene into static and moving regions without supervision. In the independently moving regions, a generic flow network learns to estimate the optical flow.

To facilitate this process we introduce Competitive Collaboration in which networks both compete and cooperate. We demonstrate that this results in top performance among unsupervised methods for all subproblems. Additionally, the moderator learns to segment the scene into static and moving regions without any direct supervision.

Future Work. We can add small amounts of supervised training, with which we expect to significantly boost performance on benchmarks, cf. [24]. We could use, for example, sparse depth and flow from KITTI and segmentation from Cityscapes to selectively provide ground truth to different networks. A richer segmentation network together with semantic segmentation should improve non-rigid segmentation. For automotive applications, the depth map formulation should be extended to a world coordinate system, which would support the integration of depth information over long image sequences. Finally, as shown in [36], the key ideas of using layers and geometry apply to general scenes beyond the automotive case and we should be able to train this method to work with generic scenes and camera motions.

Meshcapde GmbH; and received research gift funds from Intel, Nvidia, Adobe, Facebook, and Amazon. MJB's research was performed solely at, and funded solely by MPI. This project was supported by NVIDIA grants.

References

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. 1

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, pages 611–625, 2012. 16

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[4] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 1, 2, 5, 14

[5] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 1, 2, 6, 7

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 6, 7, 8

[7] C. Godard, O. Mac Aodha, and G. Brostow. Digging into self-supervised monocular depth estimation. *arXiv preprint arXiv:1806.01260*, 2018. 7, 16

[8] K. Greff, S. van Steenkiste, and J. Schmidhuber. Neural expectation maximization. In *Advances in Neural Information Processing Systems*, pages 6694–6704, 2017. 2

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5, 6, 14

[11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 8

[12] J. Janai, F. Güney, A. Ranjan, M. Black, and A. Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 690–706, 2018. 5, 7, 8, 14

[13] J. Janai, F. Güney, J. Wulff, M. Black, and A. Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE. 1

[14] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016. 1, 2

[15] A. Jepson and M. J. Black. Mixture models for optical flow computation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761. IEEE, 1993. 3

[16] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2, 11

[20] F. Liu, C. Shen, G. Lin, and I. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2016. 6, 7

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[22] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. *arXiv preprint arXiv:1802.05522*, 2018. 2, 3, 4, 6, 7

[23] N. Mayer, E. Ilg, P. Hausser, 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 2, 14

[24] S. Meister, J. Hur, and S. Roth. UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. *arXiv preprint arXiv:1711.07837*, 2017. 1, 2, 3, 7, 8, 14, 20

[25] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011. 2, 11

[26] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 1

[27] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 8

[28] A. Ranjan, J. Romero, and M. J. Black. Learning human optical flow. In *29th British Machine Vision Conference*, Sept. 2018. 2

[29] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics (TOG)*, volume 23, pages 309–314. ACM, 2004. 5

[30] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006. 6, 16

[31] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. *arXiv preprint arXiv:1709.02371*, 2017. 5

[32] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox. Demon: Depth and motion network for learning monocular stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, 2017. 2

[33] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. SfM-Net: learning of structure and motion from video. abs/1704.07804, 2017. 2

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 4, 6

[35] J. Wulff and M. J. Black. Temporal interpolation as an unsupervised pretraining task for optical flow estimation. In *German Conference on Pattern Recognition (GCPR)*, Oct. 2018. 2

[36] J. Wulff, L. Sevilla-Lara, and M. J. Black. Optical flow in mostly rigid scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 8

[37] Z. Yin and J. Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 2, 3, 4, 6, 7, 8, 14, 16, 18, 20

[38] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 2, 3, 5, 6, 7, 16, 18

[39] Y. Zou, Z. Luo, and J.-B. Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *European Conference on Computer Vision*, pages 38–55. Springer, 2018. 3, 6, 7, 8, 14, 16, 18, 20