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

Anurag Ranjan[1] **Varun Jampani**[2]

Kihwan Kim [2] **Deqing Sun** [2] **Jonas Wulff** [1] **Michael J. Black**[1]

[1]Max Planck Institute for Intelligent Systems
[2]NVIDIA Research
{aranjan, 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 and, consequently, learning to solve them together simplifies the problem because the solutions can reinforce each other by exploiting known geometric constraints. In order to model geometric constraints, we introduce *Competitive Collaboration*, a framework that facilitates competition and collaboration between neural networks. We go beyond previous work by exploiting geometry more explicitly and segmenting the scene into static and moving regions. Competitive Collaboration works much like expectation-maximization but with neural networks that act as adversaries, competing 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 results amongst unsupervised methods.

1 Introduction

Deep learning methods have achieved state of the art results on computer vision problems with supervision using large amounts of data [10, 17, 19]. However, for many vision problems which require dense continuous-valued outputs, it is either impractical or tedious to gather ground truth data [6]. We consider four such problems in the rest of the paper: single view depth prediction, camera motion estimation, optical flow, and motion segmentation. Previous works have tried to approach these problems by 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 is generally obtained using LIDAR [6] and is sparse. Furthermore, there are no sensors that provide ground truth optical flow, so all existing datasets with real imagery are limited or approximate [2, 6, 12]. Motion segmentation ground truth requires manual labeling of all pixels in an image [23].

Preprint. Work in progress.

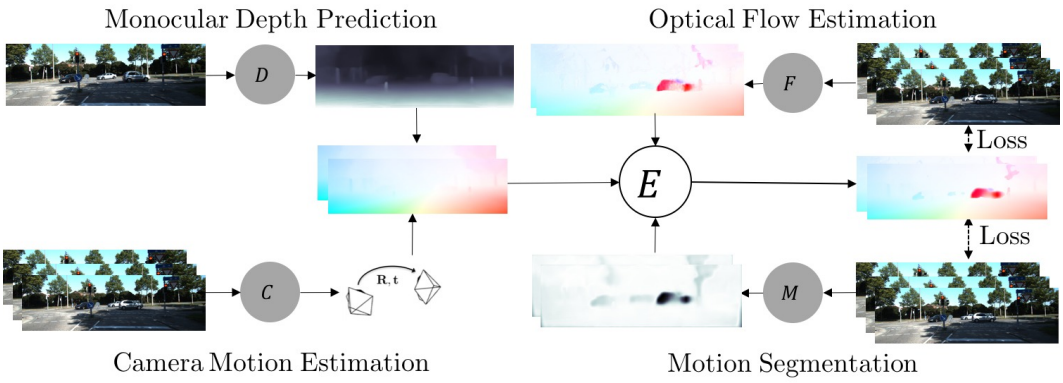

Figure 1: The network $R = (D, C)$ reasons about the scene by estimating optical flow over static regions. The optical flow network F estimates flow over the whole image. The motion segmentation M masks out static scene pixels from F to produce composite optical flow over full image. A loss E using the composite flow, applied over neighboring frames, trains these models jointly.

Recent work has tried to address the problem of limited training data using unsupervised learning [13, 22]. 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 geometry relating static scenes, camera motion, and optical flow can be exploited. For example, unsupervised learning of depth and camera motion has been coupled in [20, 33]. They use an explainability mask to mask out evidence that cannot be explained by the static scene assumption. Yin et al.[32] extend this to estimate optical flow as well and use forward-backward consistency to reason about unexplained pixels. These methods perform poorly on depth [33] and optical flow [32] benchmarks. A key reason is that the constraints applied here do not distinguish or segment objects that move independently like people and cars. Another reason is that commonly not all the data in the unlabeled training set conforms 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. 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 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 useful 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. Competitive Collaboration is a three player game consisting of two adversaries competing for a resource that is regulated by a moderator. As shown in Figure 1, we introduce two adversaries in our framework, the static scene reconstructor $R = (D, C)$ that reasons about the static scene pixels using depth and camera motion; and a moving region reconstructor F that reasons about pixels in the independently moving regions. These two adversaries 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 adversaries. However, the moderator also needs training to ensure a fair competition. Therefore, the adversaries, R, 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 as follows: 1) We introduce *Competitive Collaboration*, an unsupervised learning framework where networks can act as adversaries and collaborators to reach specific goals. 2) We show that jointly training networks in 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, camera motion estimation and optical flow estimation amongst unsupervised methods. We even outperform competing methods that use much larger networks [32] and multiple refinement steps such as network cascading [22]. Our models and code are available at `https://github.com/anuragranj/ac`.

2 Related Work

A generative adversarial network (GAN) [7] can be viewed as a two player game, where networks act as adversaries and train each other. In W-GAN [1], one of the adversaries acts like a critic. In contrast, our method is a three player game, consisting of two adversaries and a moderator, where the moderator takes the role of a critic and two adversaries 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 [1, 7, 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 [15] 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, 21]. Unsupervised methods have tried to independently solve for optical flow [13, 22] by minimizing a photometric loss. This is highly underconstrained and thus the methods perform poorly.

More recent works [20, 28, 29, 32, 33] have approached estimation of these problems by coupling two or more problems together in an unsupervised learning framework. Zhou et al. [33] 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 occlusion at boundaries, and are hence not useful for motion segmentation. Mahjourian et al. [20] use a more explicit geometric loss to jointly learn depth and camera motion for rigid scenes. Yin et al. [32] add a refinement network to [33] 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 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 [20, 32, 33] or on optical flow [22], but not on both. 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 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 adversary 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" [14] and was then widely applied to other vision problems.

3 Competitive Collaboration

Competitive Collaboration is a three player game consisting of two adversaries competing for a resource that is regulated by a moderator as illustrated in Figure 2. Consider an unlabeled training dataset $\mathcal{D}$ which can be partitioned into two disjoint sets. Two adversaries $\{R, F\}$ compete to obtain this data as a resource, and each adversary tries to partition $\mathcal{D}$ to minimize its loss. Each adversary $\{R, F\}$ competes for a partition $\{\mathcal{D}_r, \mathcal{D} \setminus \mathcal{D}_r\}$ and $\{\mathcal{D} \setminus \mathcal{D}_f, \mathcal{D}_f\}$ respectively that is optimal for itself, but not for the group. To resolve this problem, our training cycle consists of two phases. In the first phase, a moderator M assigns the data samples to adversaries while trying to ensure a fair competition. It partitions the data into $\{\mathcal{D}_r^*, \mathcal{D}_f^*\}$ such that $\mathcal{D}_r \approx \mathcal{D}_r^*$ and $\mathcal{D}_f \approx \mathcal{D}_f^*$. However, the moderator M also needs to be trained. This happens in the second phase of the training cycle. The adversaries $\{R, F\}$ collaborate to form a consensus and train the moderator M such that it correctly distributes the data in the next phase of the training cycle.

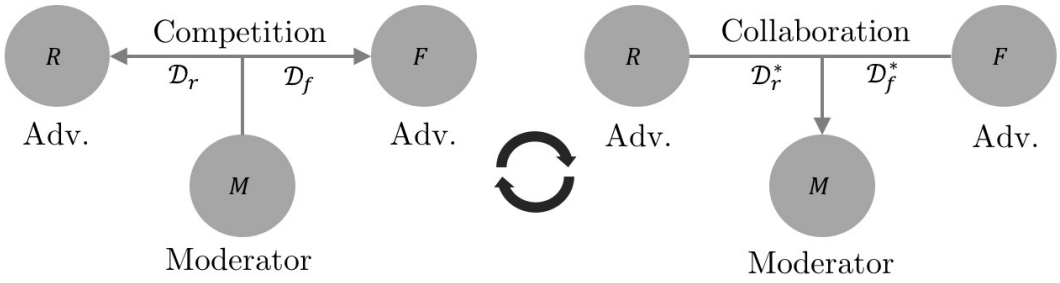

Figure 2: Training cycle of Competitive Collaboration: The moderator M drives competition between two adversaries $\{R, F\}$ (first phase, left). Later, the adversaries collaborate to train the moderator to ensure fair competition in the next iteration (second phase, right).

In our implementation, $R = (D, C)$ denotes the depth and camera motion networks that reason about the static regions in the scene. The adversary F is the optical flow network that reasons about the moving regions. For training the adversaries, the motion segmentation or mask 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 any moving pixels from corrupting its training. Similarly, it prevents any static pixels to appear in the training loss of F, thereby improving its performance in the moving regions. In the second phase of the training cycle, the adversaries (D, C) and F now collaborate to reason about static scene and moving regions by forming a consensus which is used as a loss for training the moderator, M. In the rest of the section, we formulate the joint unsupervised estimation of depth, camera motion, optical flow and motion segmentation under 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 at several places for brevity. Consider an image sequence I_-, I, I_+ with target frame I, and 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. For simplicity, we use 3 frames to describe our approach. We estimate depth of the target frame as

$$d = D_\theta(I). \tag{1}$$

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

$$e_-, e_+ = C_\phi(I_-, I, I_+). \tag{2}$$

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

$$m_+, m_- = M_\chi(I_-, I, I_+), \tag{3}$$

where $m_+, m_- \in [0, 1]$ are the probability of regions being static. Finally, the network F_ψ estimates the optical flow for the moving regions in the scene. F_ψ works with 2 images at a time, and its weights are shared while estimating u_-, u_+, the backward and forward optical flow respectively.

$$u_- = F_\psi(I, I_-), \qquad u_+ = F_\psi(I, I_+). \tag{4}$$

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

$$E = \lambda_R E_R + \lambda_F E_F + \lambda_M E_M + \lambda_C E_C + \lambda_S E_S, \tag{5}$$

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

$$E_R = \sum_\Omega \rho\Big(I, w_c(I_+, e_+, d)\Big) \cdot m_+ + \rho\Big(I, w_c(I_-, e_-, d)\Big) \cdot m_- \tag{6}$$

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

$$E_F = \sum_{\Omega} \rho\Big(I, w_f(I_+, u_+)\Big) \cdot (1 - m_+) + \rho\Big(I, w_f(I_-, u_-)\Big) \cdot (1 - m_-) \tag{7}$$

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

$$\rho(x, y) = \lambda_\rho \sqrt{(x - y)^2 + \epsilon^2} + (1 - \lambda_\rho) \left[1 - \frac{(2\mu_x\mu_y + c_1)(2\mu_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x + \sigma_y + c_2)}\right] \tag{8}$$

where λ_ρ is a fixed constant and $\epsilon = 0.01$. The second term is also known as structure similarity loss [30] that has been used in previous works [20, 32], and μ_x, σ_x are local mean and variance over the pixel neighborhood with $c_1 = 0.01^2$ and $c_2 = 0.03^2$. The loss E_M encourages the competition for training data among $R = (D_\theta, C_\phi)$ and F_ψ. To optimize for an optimal partition $\{\mathcal{D}_r^*, \mathcal{D}_f^*\}$, it minimizes the cross entropy H between the masks and a unit tensor regulated by λ_M,

$$E_M = \sum_{\Omega} H(\mathbf{1}, m_-) + H(\mathbf{1}, m_+). \tag{9}$$

A larger λ_M gives preference to static scene reconstructor R, biasing the scene towards being static. Let ν represent a geometric transformation that takes camera motion and depth and transforms them into flow vectors as defined 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

$$E_C = \sum_{\Omega} H\Big(\mathbf{I}_{||\nu(e_+,d) - u_+|| < \lambda_c}, m_+\Big) + H\Big(\mathbf{I}_{||\nu(e_-,d) - u_-|| < \lambda_c}, m_-\Big), \tag{10}$$

where $\mathbf{I} \in \{0, 1\}$ is an indicator function and equals 1 if the condition in the subscript is true. The threshold λ_c forces $\mathbf{I} = 1$, if the static scene flow $\nu(e, d)$ is close to the optical flow u, indicating a static scene. Finally, the smoothness term E_S acts as a regularizer on depth, segmentations and flow,

$$E_S = \sum_{\Omega} ||\lambda_e \nabla d||^2 + ||\lambda_e \nabla u_-||^2 + ||\lambda_e \nabla u_+||^2 + ||\lambda_e \nabla m_-||^2 + ||\lambda_e \nabla m_+||^2, \tag{11}$$

where $\lambda_e = e^{-\nabla I}$ and ∇ is the first derivative along spatial directions. 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 mask network M_χ and taking a consensus between static flow, and optical flow estimates from F_χ. It is given by

$$m^* = \mathbf{I}_{m_+ \cdot m_- > 0.5} \cdot \mathbf{I}_{||\nu(e_+,d) - u_+|| < \lambda_c}. \tag{12}$$

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 for the masks. The final masks are obtained by taking the intersection of both terms. Finally, the full optical flow, u^* between (I, I_+) is a composite of optical flows from static scene and moving regions given by

$$u^* = \mathbf{I}_{m^* > 0.5} \cdot \nu(e_+, d) + \mathbf{I}_{m^* \leq 0.5} \cdot u_+. \tag{13}$$

The loss in Eq. (5) is formulated to minimize the reconstruction error of the neighboring frames. Two adversaries compete to minimize this loss, the static scene reconstructor $R = (D_\theta, C_\phi)$ and moving region reconstructor F_ψ. The reconstructor R reasons about the static scene using Eq. (6) and the reconstructor F_ψ reasons about the moving regions using Eq. (7). The moderation is achieved by the mask network, M_χ using Eq. (9). Furthermore, the collaboration between R, F is driven using Eq. (10) to train 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 always wrong in the moving regions of the scene, and

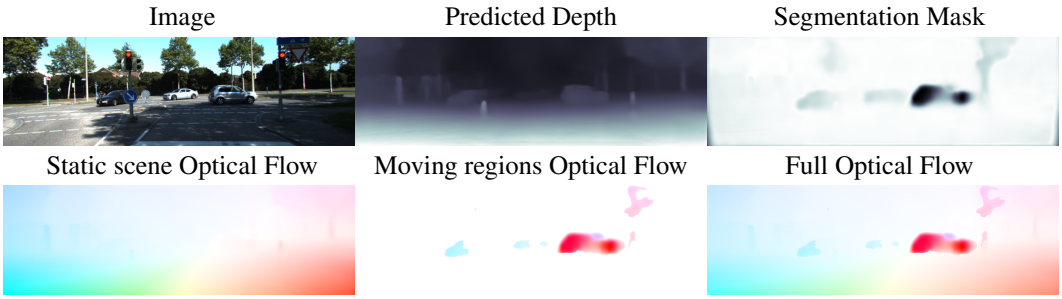

Figure 3: Top row, left to right, we show image, estimated depth map, soft masks representing motion segmentation. Bottom row, left to right, we show static scene optical flow, segmented flow in the moving regions and full optical flow.

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 about reasoning static and moving parts on the scene, as seen in Eq. (10). 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. (6, 7). In the second phase, the adversaries (D_θ, C_ϕ) and F_ψ collaborate together to train the moderator M_χ using Eqs. (9,10).

4 Experiments

Network Architecture. We use a U-Net architecture [4] for the Depth network D_θ. Our network is similar to DispNetS [33] with a few modifications. The network D_θ takes a single RGB image as input and outputs depth. The flow network, F_ψ follows the architecture of FlowNetC [4]. 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 convolutional layers. The decoder 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 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 and exclude complete sequences of test data consistent with previous works [5, 18, 20, 32, 33]. We train the networks with learning rate of 10^{-4} using ADAM[16]. 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

Result: Trained Network Parameters, $(\theta, \phi, \psi, \chi)$
Define $\lambda = (\lambda_R, \lambda_F, \lambda_M, \lambda_C, \lambda_\rho)$;
Initialize (θ, ϕ) by jointly training (D_θ, C_ϕ) with $\lambda = (1.0, 0.0, 0.0, 0.0, 0.005)$;
Initialize ψ by training F_ψ with $\lambda = (0.0, 1.0, 0.0, 0.0, 0.2)$;
Initialize χ by jointly training $(D_\theta, C_\phi, F_\psi, M_\chi)$ with $\lambda = (1.0, 0.5, 0.0, 0.0, 0.0)$;
Loop
  Competition Step
    Update θ, ϕ by jointly training $(D_\theta, C_\phi, F_\psi, M_\chi)$ with $\lambda = (1.0, 0.5, 0.05, 0, 0.005)$;
    Update ψ by jointly training $(D_\theta, C_\phi, F_\psi, M_\chi)$ with $\lambda = (0.0, 1.0, 0.005, 0, 0)$;
  Collaboration Step
    Update χ by jointly training $(D_\theta, C_\phi, F_\psi, M_\chi)$ with $\lambda = (1.0, 0.5, 0.005, 0.3, 0)$;
EndLoop

Algorithm 1: Network Training Algorithm

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

		Error				Accuracy, δ		
Method	Data	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. [18]	k	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhou et al.[33]	cs+k	0.198	1.836	6.565	0.275	0.718	0.901	0.960
Mahjourian et al.[20]	cs+k	0.159	1.231	5.912	0.243	0.784	0.923	0.970
Geonet-Resnet[32]	cs+k	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Zhou et al.*[33]	k	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Mahjourian et al.[20]	k	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Geonet-VGG[32]	k	0.164	1.303	6.090	0.247	0.765	0.919	0.968
Geonet-Resnet[32]	k	0.155	1.296	5.857	0.233	0.793	0.931	**0.973**
Ours (basic)	k	0.184	1.476	6.325	0.259	0.732	0.910	0.967
Ours (basic + ssim)	k	0.168	1.396	6.176	0.244	0.767	0.922	0.971
Ours (CC + ssim)	k	**0.148**	**1.149**	**5.464**	**0.226**	**0.815**	**0.935**	**0.973**

Table 2: Results on Optical Flow (a) and Camera Motion Estimation (b). (a): SP, MP refer to static scene and moving region pixels. We also compare with supervised methods, FlowNet2 and SpyNet that are not fine tuned on KITTI ground truth flow.

(a)

	Average EPE		
Method	SP	MP	Total
FlowNet2[11]	-	-	10.06
SPyNet[24]	-	-	20.26
UnFlow-C[22]	-	-	8.80
UnFlow-CSS[22]	-	-	8.10
Geonet [32]	-	-	10.81
Ours (basic, R)	7.51	32.75	13.54
Ours (basic, F)	15.32	6.20	14.68
Ours (CC)	**6.35**	**6.16**	**7.76**

(b)

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.[33]	0.021 ± 0.017	0.020 ± 0.015
Zhou et al.*[33]	0.016 ± 0.009	0.013 ± 0.009
Mahjourian et al.[20]	0.013 ± 0.010	0.012 ± 0.011
Geonet[32]	$\mathbf{0.012 \pm 0.007}$	0.012 ± 0.009
Ours (basic)	0.022 ± 0.010	0.018 ± 0.011
Ours (basic + ssim)	0.017 ± 0.009	0.015 ± 0.009
Ours (CC + ssim)	$\mathbf{0.012 \pm 0.007}$	$\mathbf{0.012 \pm 0.008}$

experiments. We then learn M_χ using the joint loss. We use $\lambda_R = 1.0, \lambda_F = 0.5$ for joint training because static scene reconstructor R uses 4 reference frames in its loss, whereas 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 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 $\lambda_S = 0.005, \lambda_c = 0.001$. The constant λ_ρ regulates the SSIM loss and is chosen empirically. We iteratively train the adversaries $(D_\theta, C_\phi), F_\psi$ and moderator M_χ for 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 Table 1, 2b. The depth is evaluated on Eigen et al.'s [5] split of raw KITTI dataset [6] and camera motion is evaluated on KITTI Odometry dataset [6]. These evaluation frameworks are consistent with previous works [5, 18, 20, 32]. All depth maps are capped at 80 meters. Our network architecture for depth and camera motion estimation is most similar to [33] and this is reflected in the performance of our basic model. We get some performance improvements by adding the SSIM loss [30]. However, we observe that using Competitive Collaboration (CC) framework with a joint loss results in larger performance gains in both tasks. Geonet [32] is the next best method on unsupervised depth and camera motion estimation, however, it uses a much bigger Resnet [9] and is trained on a larger dataset. In comparison, Geonet-VGG [32] that uses VGG network [25] for depth estimation, which is more similar to our architecture performs significantly worse. In summary, we show that jointly training the networks using CC is very useful in boosting performance of single image depth prediction and camera motion

Table 3: Motion Segmentation Results. Intersection Over Union (IoU) scores on KITTI2015 training dataset images computed over car pixels.

	Moving Car	**Static Car**	**Overall**
MaskNet	54.17	43.65	48.91
Consensus	53.96	49.00	51.48
Joint	**56.44**	**61.03**	**58.74**

estimation tasks. We show qualitative results in Figure 3. In the Appendix A.5, we show more qualitative results and comparisons with other methods.

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 works [22, 32]. We obtain state of the art performance among unsupervised methods as shown in Table 2a. UnFlow-CSS [22] uses 3 cascaded networks to refine optical flow at each stage. Geonet is more similar to our architecture but uses a larger ResNet-50 architecture. We observe that by using static scene reconstructor, R or moving region reconstructor F independently leads to worse performance because they are specialized in reasoning out particular regions in the image. By using them together and compositing optical flow from both the reconstructors as shown in Eq. (13) leads to large improvement in performance. Qualitative results are shown in Figure 3 and in the Appendix A.5.

Motion Segmentation. We evaluate the estimated motion segmentations using KITTI 2015 training set [6] that has 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 3, 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 is referred to as 'Joint' (Eq. 12). IoU results indicate big IoU improvements with 'Joint' masks compared to both 'MaskNet' and 'Consensus' masks showcasing the complementary nature of different masks. Qualitative results are shown in Figure 3 and in the Appendix A.5.

5 Discussion

Learning to infer depth from a single image typically requires training images with ground truth depth scans. Learning to compute optical flow typically relies on synthetic data, which may not generalize to real image sequences. For static scenes, 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 (5 in our case) while training. This is particularly critical for unsupervised training since all of these problems are highly ambiguous. 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 make this work, we introduce Competitive Collaboration in which networks both compete and cooperate. Coupling adversaries in a GAN has been widely adopted in learning representations. In our framework, the adversaries not only compete, thereby improving the performance of each other, but also collaborate to improve other models by forming consensus among each other. We demonstrate that this results in top performance among unsupervised methods for all subproblems. For example, our single-view depth estimates are both accurate and highly detailed. Additionally, the moderator learns to segment the scene into static and moving regions without any direct supervision.

Future Work. Our unsupervised approach could be trained from much more data such as the Cityscapes dataset [3], which does not include ground truth depth or flow. In terms of metric flow accuracy, fully unsupervised methods like ours are still not competitive on standard benchmarks. Future work will look at combining our unsupervised method with some flow supervision. We can add smaller amounts of supervised training, with which we expect to significantly boost performance on benchmarks, cf. [22]. 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 and state-of-the art flow network [27], together with semantic segmentation should improve non-rigid segmentation. Additionally, the segmentation masks should be consistent across time [26]; this provides an additional strong constraint for unsupervised learning. 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, the key ideas here apply to most scenes and not just the automotive case [31] and we should be able to train this method to work with generic scenes and camera motions.

Acknowledgements

We are grateful to Clément Pinard for his github repository. We use it as our initial code base. We thank Georgios Pavlakos for helping us with several revisions of the paper. We thank Joel Janai for preparing optical flow visualizations.

References

[1] Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223.

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

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

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

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

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

[7] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

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

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

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

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

[12] Janai, J., Güney, F., Wulff, J., Black, M., and Geiger, A. (2017). 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. IEEE.

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

[14] Jepson, A. D. and Black, M. J. (1993). *Mixture models for optical flow computation*. Department of Computer Science, University of Toronto.

[15] Kendall, A., Grimes, M., and Cipolla, R. (2015). PoseNet: a convolutional network for real-time 6-dof camera relocalization.

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

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

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

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

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

[21] Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. (2016). 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.

[22] Meister, S., Hur, J., and Roth, S. (2017). UnFlow: Unsupervised learning of optical flow with a bidirectional census loss. *arXiv preprint arXiv:1711.07837*.

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

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

[25] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[26] Sun, D., Sudderth, E. B., and Black, M. J. (2010). Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *Advances in Neural Information Processing Systems*, pages 2226–2234.

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

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

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

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

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

[32] Yin, Z. and Shi, J. (2018). Geonet: Unsupervised learning of dense depth, optical flow and camera pose. *arXiv preprint arXiv:1803.02276*.

[33] Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017). Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7.

A Appendix

A.1 A note on competing methods

The first release by Zhou el at. [33] showed that their model performs better by training with cityscapes [3] and KITTI [6] datasets. However, the latest results were updated on their github page and show that adding augmentations to KITTI dataset produces even better results. Our model is consistent with these augmentations, and therefore we use only KITTI for training. All the comparisons in our paper reflect the performance of Zhou et al.'s latest model. The other competing methods Mahjourian et at.[20] and Geonet [32] have not been published at the time of writing this paper, and we rely on their arxiv manuscript.

A.2 The camera warping function w_c and static flow transformer ν

The network C predicts camera motion that consist of camera rotations $sin\alpha, sin\beta, sin\gamma$, and translations t_x, t_y, t_z. Thus $e = (sin\alpha, sin\beta, sin\gamma, t_x, t_y, t_z)$. Given camera motion and depth d, we transform the image coordinates (x, y) into world coordinates (X, Y, Z).

$$X = \tfrac{d}{f}(x - c_x) \tag{14}$$

$$Y = \tfrac{d}{f}(y - c_y) \tag{15}$$

$$Z = d \tag{16}$$

where (c_x, c_y, f) constitute the camera intrinsics. We now transform the world coordinates given the camera rotation and translation.

$$\mathbf{X}' = R_x R_y R_z \mathbf{X} + t$$

where $(R_x R_y R_z, t) \in SE3$ denote 3D rotation and translation, and $\mathbf{X} = [X, Y, Z]^T$. Hence, in image coordinates

$$x' = \frac{f}{Z} + c_x \tag{17}$$

$$y' = \frac{f}{z} + c_y \tag{18}$$

We can now apply the warping as,

$$w_c\big(I(x, y), e, d\big) = I(x', y'). \tag{19}$$

The static flow transformer is defined as,

$$\nu(e, d) = (x' - x, y' - y) \tag{20}$$

A.3 The flow warping function, w_f

The flow warping function w_f is given by

$$w_f\big(I(x, y), u_x, u_y\big) = I(x + u_x, y + u_y) \tag{21}$$

where, (u_x, u_y) is the optical flow, and (x, y) is the spatial coordinate system.

A.4 Network Architectures

We briefly describe the network architectures below. For details, please refer to Figure 4.

Depth Network D_θ. Our depth network is similar to DispNetS [21] and outputs depths at 6 different scales. Each convolution and upconvolution is followed by a ReLU except the prediction layers. The prediction layer at each scale has a non-linearity given by $1/(\alpha \text{ sigmoid}(x) + \beta)$.

Camera Motion Network C_ϕ. The camera motion network consists of 8 convolutional layers, each of stride 2 followed by a ReLU activation. This is followed by a convolutional layer of stride 1, whose feature maps are averaged together to get the camera motion.

Flow Network F_ψ. The optical flow network is the same as FlowNetC [4] with 6 output scales of flow and is shown in Figure 4. All convolutional and upconvolutional layers are followed by a ReLU except prediction layers. The prediction layers have no activations.

Mask Network M_χ. The mask network has a U-Net [4] architecture. The encoder is similar to the camera motion with 6 convolutional layers. The decoder has 6 upconvolutional layers. Each of these layers have ReLU activations. The prediction layers use a sigmoid.

A.5 Qualitative Results

The qualitative results of the predictions are shown in Figure 5. We would like to point out that, our method is able to segment the moving car, and not the parked cars on the roads. In addition, it segments other moving objects, such as the bicyclist.

We compare the qualitative results for single image depth prediction in Figure 6. We also contrast our results with basic models that were trained independently without a joint loss. We observe that our model produces better results, capturing moving objects such as cars and bikes, as well as surface edges of trees, pavements and buildings.

We compare the qualitative results for optical flow estimation in Figure 7. We show that our method performs better than UnFlow [22], the current state of the art on unsupervised optical flow. Our flow estimations are better at the boundaries of the cars and pavements. In contrast, UnFlow produces blurry flow fields.

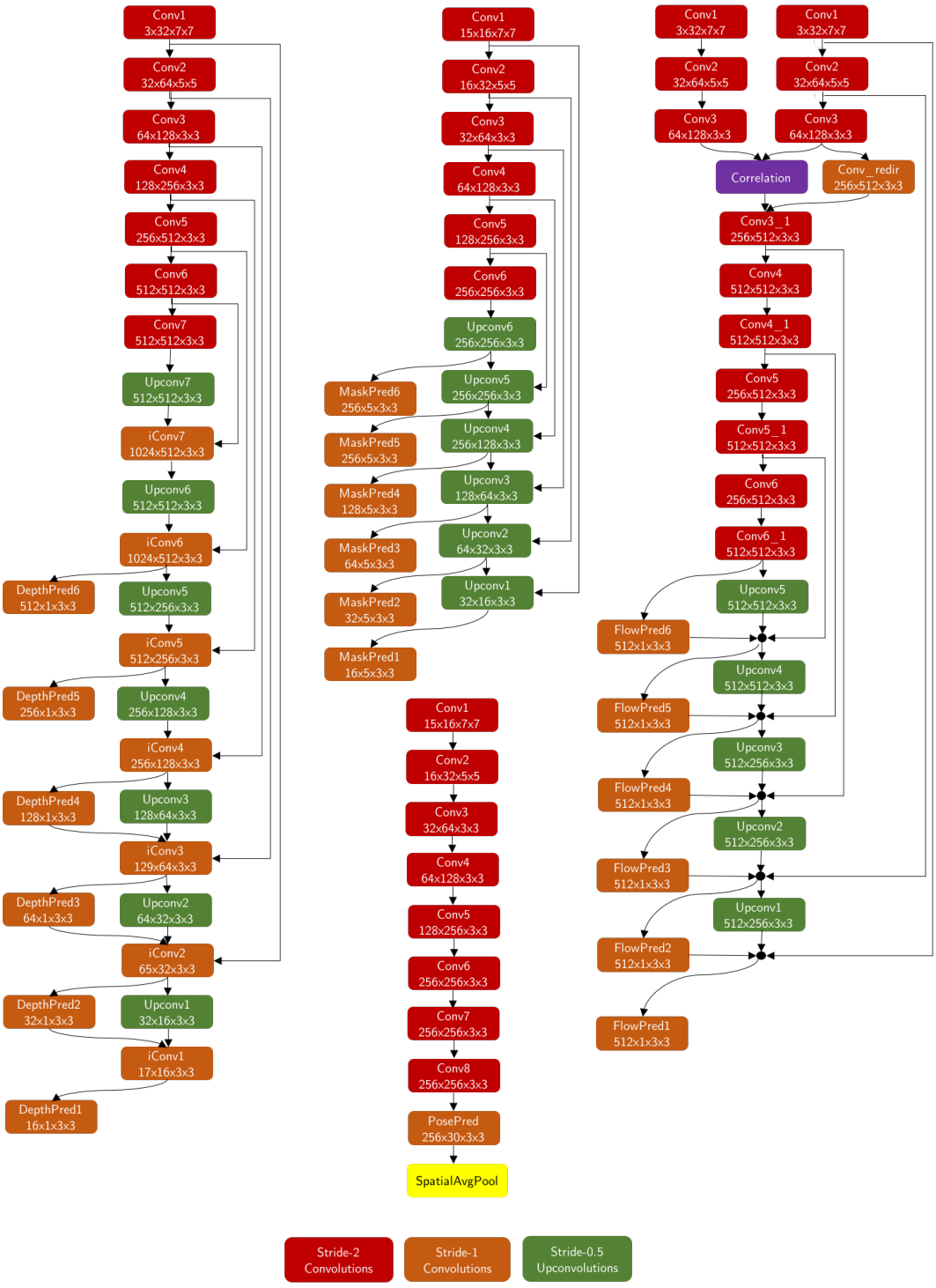

Figure 4: Architecture of the Depth Network (left), Mask Network (center-top), Flow Network (right) and Camera Motion Network (center-bottom). Convolutional layers are red (stride 2) and orange (stride 1) and upconvolution layers are green (stride 2). Other colors refer to special layers. Each layer is followed by ReLU, except prediction layers.

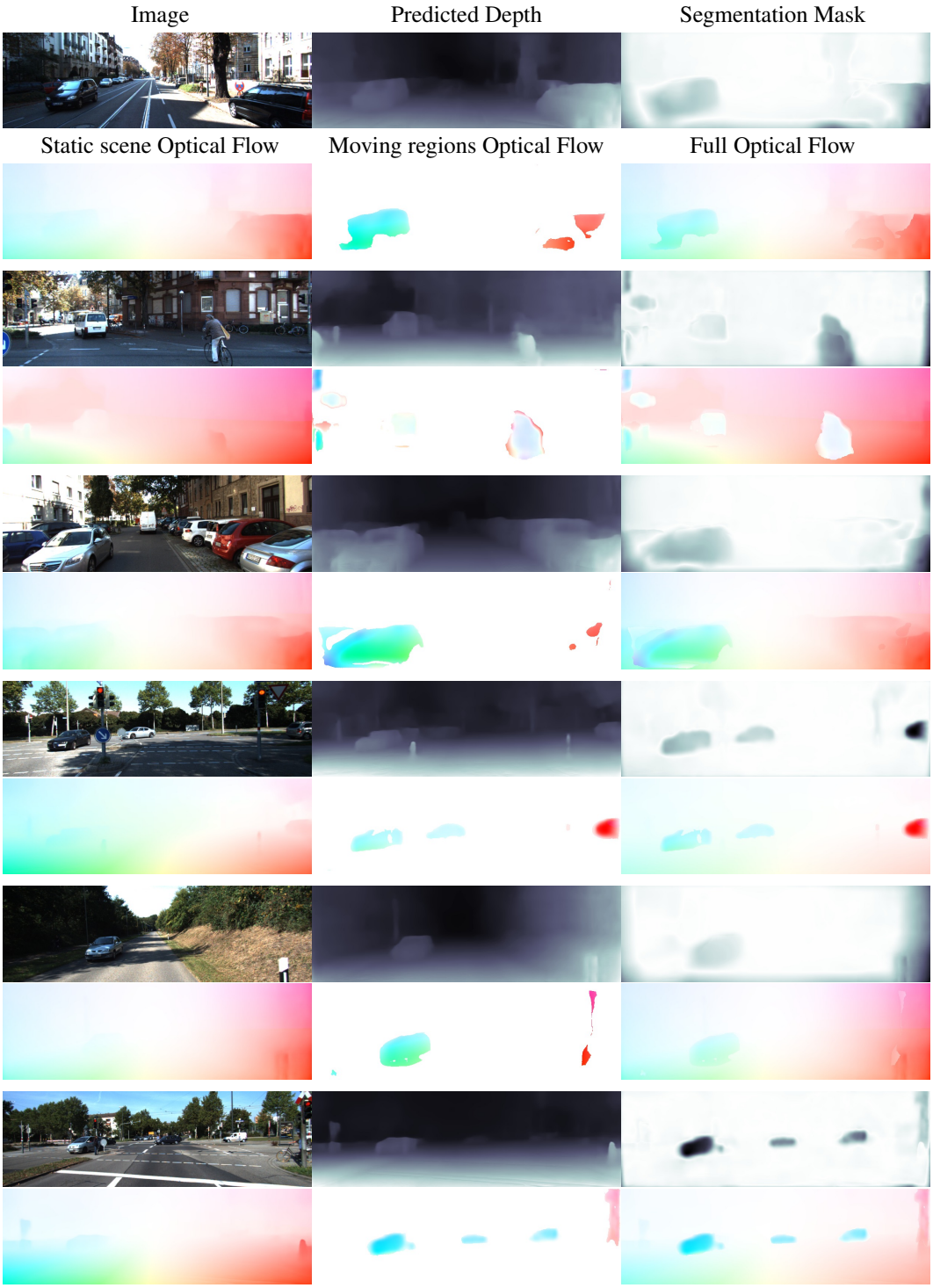

Figure 5: Network Predictions. Top row: we show image, predicted depth, segmentation masks. Bottom row: we show static scene optical flow, segmented flow in the moving regions and full optical flow.

Figure 6: Qualitative results on single view depth prediction. Top row: we show image, interpolated ground truth depths, Zhou et al. [33] results. Bottom row: we show results using our basic model, after adding SSIM loss, and joint model.

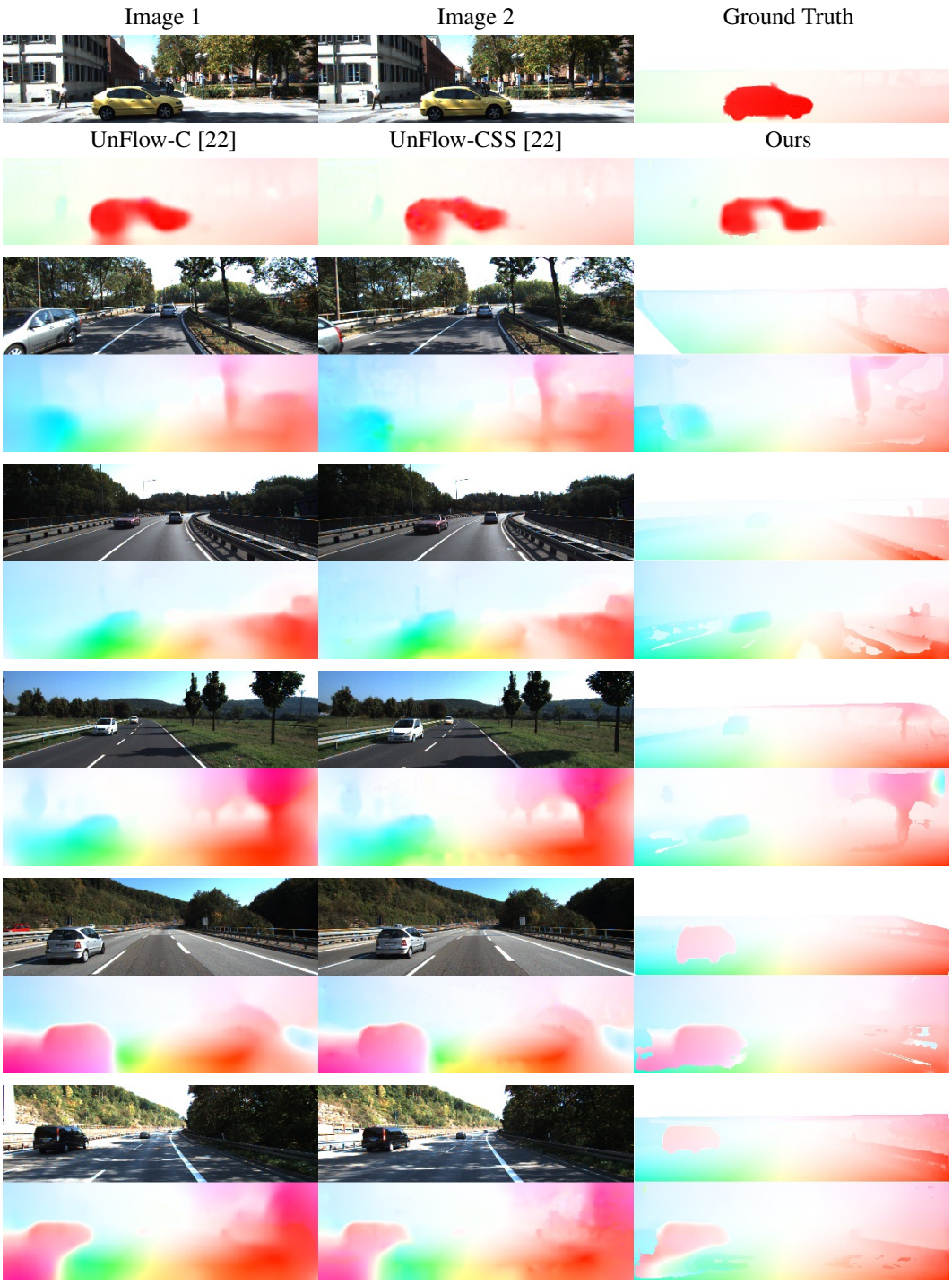

Figure 7: Qualitative results on Optical Flow estimation. Top row: we show image 1, image 2 and ground truth flow. Bottom row: we show results from UnFlow-C, UnFlow-CSS and our model.