Monocular Depth Prediction using Generative Adversarial Networks

Arun CS Kumar Suchendra M. Bhandarkar
The University of Georgia
aruncs@uga.edu suchi@cs.uga.edu

Mukta Prasad
Trinity College Dublin
prasadm@tcd.ie

Abstract

We present a technique for monocular reconstruction, i.e. *depth map and pose prediction from input monocular video sequences, using adversarial learning. We extend current geometry-aware neural network architectures that learn from photoconsistency-based reconstruction loss functions defined over spatially and temporally adjacent images by leveraging recent advances in adversarial learning. We propose a generative adversarial network (GAN) that can learn improved reconstruction models, with flexible loss functions that are less susceptible to adversarial examples, using generic semi-supervised or unsupervised datasets. The generator function in the proposed GAN learns to synthesize neighbouring images to predict a depth map and relative object pose, while the discriminator function learns the distribution of monocular images to correctly classify the authenticity of the synthesized images. A typical photoconsistency-based reconstruction loss function is used to assist the generator function to train well and compete against the discriminator function. We demonstrate the performance of our method on the KITTI dataset in both, depth-supervised and unsupervised settings. The depth prediction results of the proposed GAN are shown to compare favorably with state-of-the-art techniques for monocular reconstruction.*

1. Introduction

As computer vision matures, it is increasingly clear that in addition to recognition and classification, reconstruction and pose estimation are imperative to do well, ideally performed jointly. This goal when achieved, would have wide ranging implications especially in areas *e.g*. robotic navigation and simultaneous localization and mapping (SLAM). However, the problem continues to be a difficult one to solve, the ability to relate information across multiple views, while handling noise, uncertainty and estimating pose and shape (encapsulated by Structure from Motion (SfM) and SLAM) continues to be actively researched and improved, despite much progress. While the geometry of image formation, image features, and hand crafted energies and priors have been well studied, there is significant curiosity and hope for what deep learning progress can bring to the table, especially in terms of discovering features, formulations, exploiting priors and large amounts of data. Naturally, in recent times, various sub-tasks of reconstruction *e.g*. single view reconstruction, optical flow (and scene flow!), pose estimation and joint pose and depth estimation for temporal and stereo setups have seen a flurry of activity using supervised and semi-supervised learning.

In this paper, we advance this state of the art, by harnessing the power of adversarial learning over the existing state of the art in geometry aware neural network based monocular reconstruction. Generative Adversarial Networks (GANs) have shown themselves to be a promising tool; rather than purely loss based learning regulated by some prior, a GAN pits a generative neural network (generating samples of a variable of interest) against a discriminative one (that tests its authenticity). This allows the discriminator to learn more flexible distributions from available data than typical manually defined loss functions, and are shown to tackle underfitting-based issues, work well even without supervised training pairs and tackle confusing, adversarial cases better. We address the following question in monocular reconstruction: given at least 2 contiguous images in time from a monocular camera, can we predict depth and pose (using the generator) such that the predicted images in neighbouring time steps are realistic enough to pass the discriminator's test? The answer, it turns out, is yes. Further, traditional photo-consistency losses help train our system better. Our method, shown in Figure 1, compares favourably against state of the art on the latest benchmarks for reconstruction based evaluations and our results are not only geometrically and photometrically consistent but also better trained against adversarial examples. We now introduce this in the context of related work.

2. Related Work

Reconstruction traditionally adopted approaches motivated by the physics of image formation from 3D either based on finding consistent 2D matches that yield good reconstruction (triangulation) [25, 12, 20] or proposing

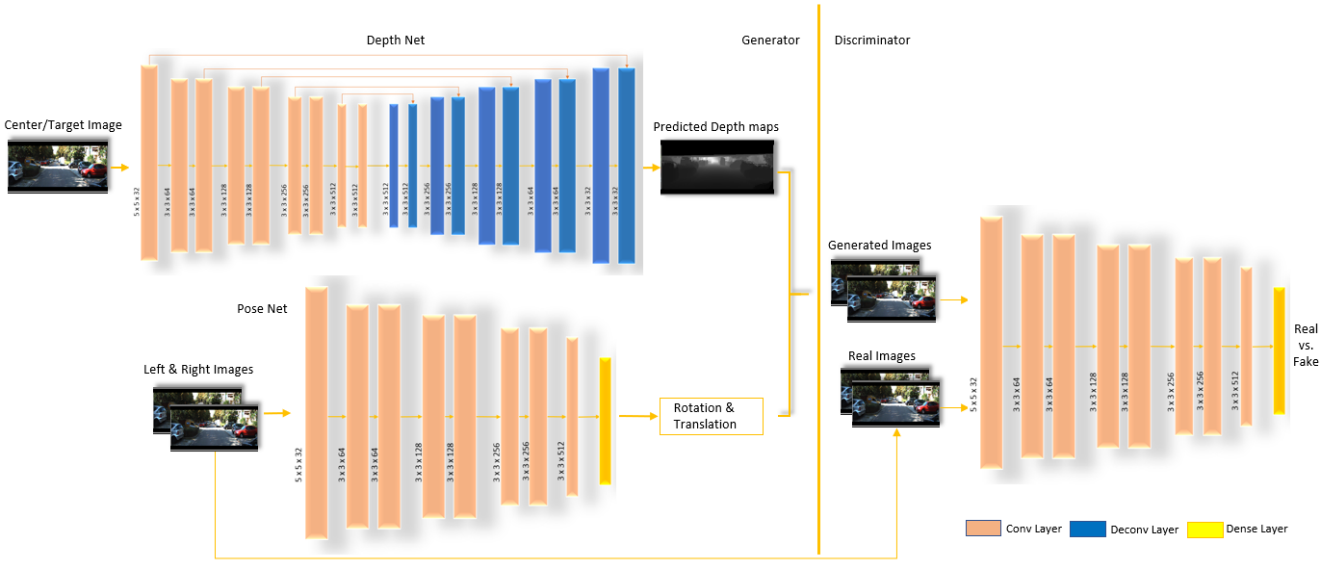

Figure 1: Proposed Framework: The *generator* consists of two subnetworks - the *depth* subnetwork that predicts depth map from the target *(center)* image, and the *pose* subnetwork, that learns to predict pose parameters from image triplets. The image triplets are fed to the *pose* subnetwork, that transforms the source *(left & right)* images to the target *(center)* image, while the *center* image is fed to the *depth* subnetwork that outputs a depth map. Using the estimated depth and pose parameters, the generator transforms the source images, which is then interpolated using Spatial Transformer Networks [17], to output a generated pair of images. The *discriminator* subnetwork then learns to differentiate the real and the generated images. Please refer to section 4 for more information on network architectures.

3D structure and texture that is consistent with image evidence (photo-consistency based methods, or direct methods) [24, 21, 10]. Whether machine learning can learn to reconstruct without depending on hand-crafted features and energies, has been discussion for a long time. Though regression based approaches towards subsets of the problem [27, 22] showed promise, substantial headway was made more recently by deep learning based approaches. The deep learning progress, seen initially in image classification [18], was followed closely by reconstruction based learning. In addition to being an intricate problem, deep learned reconstruction depends on exploiting large amounts of data (often with supervision), which was lacking in reconstruction based benchmarks for a while. A popular workaround was the use of artificial datasets *e.g*. with objects superimposed on artificial backgrounds [7, 19]. But then projects like Kitti [14], CityScapes [3] *etc*. paved the way for larger, more comprehensive benchmarks.

Initial attempts were made on, arguably, sub-parts of the problem like correspondence estimation [26, 30], optical flow or disparity [7]. These approaches slowly consolidated such learning into estimation of scene flow (3D point and velocity estimation implicitly giving flow or disparity) [19] for a small set of frames. Another class of approaches aimed at learning to predict depth maps [9, 8, 16] but learnt this mapping from typical supervised pairs of inputs and outputs while Byravan and Fox [2] learned similar prediction of pose from depth maps. However, supervised data for reconstruction is usually limited, LiDaR scanners are expensive, capture limited depth and have a limited view of the viewing sphere. But the community has been quick to extend such learning where a reconstruction based image-consistency error is used to self-supervise the learning from multiple frames, independent of an explicit depth source, either using stereo or a monocular source over time. The basic principle is that if the correct depth and pose for an image pair are predicted, they should be photo-consistent with the source images. So methods [15, 28, 31] moved towards more effectively predicting both depth and pose with less supervision. While some methods [28] utilize available supervision, others [31] aim to be completely self-supervised. Naturally the problem is still underconstrained and the use of priors helps estimate a sensible solution. Godard *et al*. [15] enforce a prior to encourage the estimated disparity to be smooth. In [28] additional priors on depth maps, motion maps and even the depth gradients are explored.

3. Proposed Approach

The proposed method aims to learn depth and pose parameters via adversarial learning. Given a triplet of images that are adjacent frames of a monocular video sequence, we

feed the *center or target* image to the *depth* subnetwork that learns to generate a depth map, and the image triplet is fed to the *pose* subnetwork, that regresses pose parameters that transforms the source *(left & right)* images to the target *(center)* image. The estimated depth and pose parameters are then used to render a pair of predicted images for the left and right (using the well known interpolation of [17]). The discriminator, a network that learns to discriminate between original and generated images, scores a likelihood of how similar the original and the generated images are.

Model

Like a traditional GAN, our network consists of two adversarial components; the *generator* that predicts images neighbouring a given image and the discriminator which classifies the authenticity of such generated neighbourhood images. Our generator is comprised of two subnetworks, the *depth* subnetwork and the *pose* subnetwork. Similar to [31], the input of the depth subnetwork is the image I_t, for which network predicts a depth map $\mathcal{D}_t$. Images I_{t-1} and I_{t+1} are the *left* and *right* (adjacent) images, in time, of I_t. Given the image triplet $\{I_{t-1}, I_t, I_{t+1}\}$ containing, the *pose* subnetwork predicts two pairs *rotation* and *translation* parameters $\{R_{t,t+1}, \mathfrak{t}_{t,t+1}\}, \{R_{t-1,t}, \mathfrak{t}_{t-1,t}\} \in SE3$, representing the relative transform between the camera at successive instants of time, which is then used with the predicted depth map to generate I_t^1 and I_t^2.

In [31], the generator would be trained on the typical photo-consistency loss (augmented with *left-right* consistency constraints as in [15]), given by:

$$\mathcal{L}_{photo} = \frac{1}{2}\{|I_t - I_t^1| + |I_t - I_t^2|\} \tag{1}$$

where I_t^1 and I_t^2 are the predictions of I_{t+1} and I_{t-1}, are generated by transforming images I_{t+1} and I_{t-1} using predicted depth maps and pose parameters.

Then the objective of the discriminator subnetwork is to learn to classify real images from generated ones $\{I_t^1, I_t^2\}$. It has to be noted that the objective of the discriminator is not to learn explicitly the difference between the instances of each generated and real image pair, instead, to learn to provide a likelihood of how real or fake (or generated) a given image is. With the discriminator as D and the the generator as G, following [5] where a generator is modelled by a context encoder, the adversarial loss can be formulated as,

$$\mathcal{L}_{adv} = \max_{D} \quad \mathbb{E}_{x \in \mathcal{X}}[log(D(x)) + log(1 - D(F(x)))] \tag{2}$$

where $\mathcal{X}$ represents the data distribution. We adapt the formulation of [5], and use a generative modeling of images, where we model our autoencoder as our generator, $G \triangleq F$. Training context encoders as generators by propagating adversarial loss via discriminator have been shown to be successful [5] on problems such as image inpainting.

We train and evaluate our network in both *depth*-supervised and unsupervised mode. Our training objective for the unsupervised setting that minimizes the photoconsistency error between true and transformed image pairs (Equation 1), along with the adversarial loss (Equation 2), is given by:

$$\mathcal{L}_{total}^{u} = \lambda_1 \mathcal{L}_{photo} + \lambda_2 \mathcal{L}_{adv} \tag{3}$$

For depth-supervised learning, for a pixel i in target image I_t, the depth loss $\mathcal{L}_{depth}$ can be formulated as

$$\mathcal{L}_{depth}(\mathcal{D}_{ti}, \mathcal{G}_{ti}) = \frac{1}{n}\sum_{i} |\mathcal{D}_{ti} - \mathcal{G}_{ti}|^2 \tag{4}$$

where $\mathcal{D}_t$ and $\mathcal{G}_t$ are the predicted and ground truth depth map respectively, and n is the total number of pixels. In case of learning with depth supervision, the loss function then becomes:

$$\mathcal{L}_{total}^{s} = \lambda_1 \mathcal{L}_{photo} + \lambda_2 \mathcal{L}_{adv} + \lambda_3 \mathcal{L}_{depth} \tag{5}$$

We tried both L_1 and L_2 norms for computing depth loss while training, but we found the results to be pretty similar.

At initial stages of training, the discriminator learns consistently as images generated are quite distorted, overly smoothed and considerably different from real images, as the pose and depth estimates are highly inaccurate. As training progresses, despite the differences between generated and source images (photoconsistency error) are high, the pose and depth estimates are reasonable enough to render images that *appear* realistic. This is because, the optimization requires the pose and depth parameters to be accurate enough to minimize photoconsistency loss, while the discriminator can be fooled by any reasonable estimates of pose and depth as long as they project to the image plane and the interpolation of the image appear realistic. As the generated images start to appear a bit more realistic, the discriminator, then eventually stops learning half way through training, and reaches equilibrium prematurely. As a means to tackle this problem, we first compute difference images $\delta_t^1 = |I_t - I_t^1|$ and $\delta_t^2 = |I_t - I_t^2|$. We then add the difference images δ_{t-1} and δ_{t+1} to the I_t, to obtain $I_t^{\delta_{t-1}}$ and $I_t^{\delta_{t+1}}$ respectively; this step allows us to externally induce discrepancy to make the image appear fake, generating photoconsistency-aware adversarial examples to confuse the discriminator, and let it learn effectively and continually. The discriminator is fed with these error induced instances of generated images $I_t^{\delta_{t-1}}$ and $I_t^{\delta_{t+1}}$ instead of raw generated images I_t^1 and I_t^2. In reference to the discriminator training, this step reduces generated images from appearing realistic despite inaccurate pose/depth maps, by inducing additional error, which will

allow the discriminator to continue learning than stop half way through training.

The algorithmic implication of this step is that, the discriminator is in fact trying to minimize the photoconsistency loss as well, as the adversarial loss would remain high as long as the photoconsistency is not minimized. Thus the discriminator is forced to work against the generator at the same time minimize the same objective of that of the generator. Eventually, when the photoconsistency loss is substantially minimized, generated adversarial examples would start to appear more realistic as the noise added would approach to zero. Given the generated and difference images, an adversarial example is computed using the formulation: ${}^{adv}I_t^1 = \omega I_t^1 + (1-\omega) I_t^1 \delta_t^1$ and ${}^{adv}I_t^2 = \omega I_t^2 + (1-\omega) I_t^2 \delta_t^2$.

When estimating pose parameters, independent objects, such as a car passing by, or a pedestrian crossing the road, tends to have its own motion, that, in general, does not agree with the actual camera motion. In order to tackle the problem, we use the explainability masks proposed by [31], to ignore or mask the regions with independent motion out, while estimating the pose of the scene.

4. Implementation Details

The **depth subnetwork** follows a traditional encoder-decoder architecture, where the encoding or the contracting phase comprises a series of convolutional layers that transforms an image into a latent representation, which is followed by the decoding or expanding phase that consists of deconvolutional or transpose convolutional layers along with convolutional layers, learns to regress depth maps from the latent representation of encoder. We removed pooling layers and used convolutional layers with alternating strides instead, for downsampling the tensor. The pooling layer provides spatial invariance which aids classification, but for autoencoders, removing pooling layers have been shown to improve performance [5]. The encoder consists of 6 pairs of convolutional layers of alternating strides of 2's and 1's, with $\{32, 64, 128, 256, 512\}$ filters respectively. The *decoder* consists of a series of alternating upconvolutional or transpose convolutional layers and convolutional layers, that maps the latent features of encoding layer into the depth maps. We use skip connections across convolutional and deconvolutional layers as they have been shown improve performance [6], especially for coming across vanishing gradient problems, thus effectively allowing us to explore deeper network architectures. All the convolutional and deconvolutional layers use *relu* as the activation function. The decoder of the depth subnetwork, is affixed with a convolutional layer with 1 filter and a sigmoid activation, that outputs the final depth map.

The **pose subnetwork**, identical to the encoder of *depth subnetwork*, consists of a series of 8 convolutional layers with alternating strides of 2's and 1's, which is then followed by an average pooling layer, with 12 filters $(6 \times (\eta - 1))$ —3 rotation and 3 translation for two transformations, as we use image triplet ($\eta = 3$). The input of the pose subnetwork is the image triplet that is concatenated across the number of channels. We also use explainability mask to reason motion that do not correspond to the estimated comprehensive camera motion of the scene. We employ the explainability mask algorithm proposed by [31], to mask/ignore the regions with independent motion or occlusion out and use the rest of the regions for inferring pose parameters. We also use resize layers towards the end of the generator, to resize the predicted depth maps to correspond to the actual or required sizes as there are often negligible offsets due to padding. While computing the pose matrix, our method uses intrinsic camera parameters when available, else would default to use $\{0.5, 0.5, 1\}$ for $\{c_x, c_y, f\}$ respectively.

The architecture of the **discriminator** subnetwork is similar to that of a traditional - convolutional layers followed by a set of dense layers (with $\{512, 256, 128\}$ filters respectively), and a sigmoid layer with a filter size of 1, that simply outputs a probability; all layers use *leaky relu* [1] activation. Like traditional discriminators, the sigmoid layer outputs a single value, the likelihood of the image being real or fake. For all the above subnetworks, except for the first (convolutional) layer where we use a filter of size 5×5, the filter sizes of all other layers are set to 3×3.

For **Training** the network, we use *Adam* optimizer [4] with an initial learning rate of 0.002 which is periodically adjusted as the training progresses using an exponential decay function, with a rate of decay as 0.95 for every 1500 steps. We gather batches of adjacent images (sequences) in the dataset, as triplets. The *pose* network is fed with the triplets $\{I_{t-1}, I_t, I_{t+1}\}$ of size $B \times H \times W \times (\eta * ch)$, where B is the batch size and ch is the number of channels ($ch = 3$, as we use *rgb* images through out the paper). For all our experiments H and W, the *height* and *width* of images are set to 128 and 384 respectively, and the batch size is set to 32. Also while training, we treat the *left* and *right* stereo pairs of images in KITTI dataset as independent image/video sequences.

The **adversarial examples** are generated as a weighted sum of the generated images and pixel-wise photoconsistency error. We tried various values for weights ω, but we found that ω between 0.90 and 0.95 works better overall. A lesser value for ω implies that the generated adversarial example will be highly noisy, that subsequently leads the adversarial loss to fail to improve as the discriminator accuracy reaches 100 percent hastily, at most times within 3000 iterations. On the other hand, a very high value for ω will introduce too little noise to make a difference in training. Moreover, following [5], we set the hyperparameter (weights) for the adversarial loss λ_2 to be quite low (in comparison to λ_1 or λ_3). For depth supervised learning we set $\lambda_1 + \lambda_3 = 0.995$ and $\lambda_2 = 0.005$, where $\lambda_1 = \lambda_3$, and for

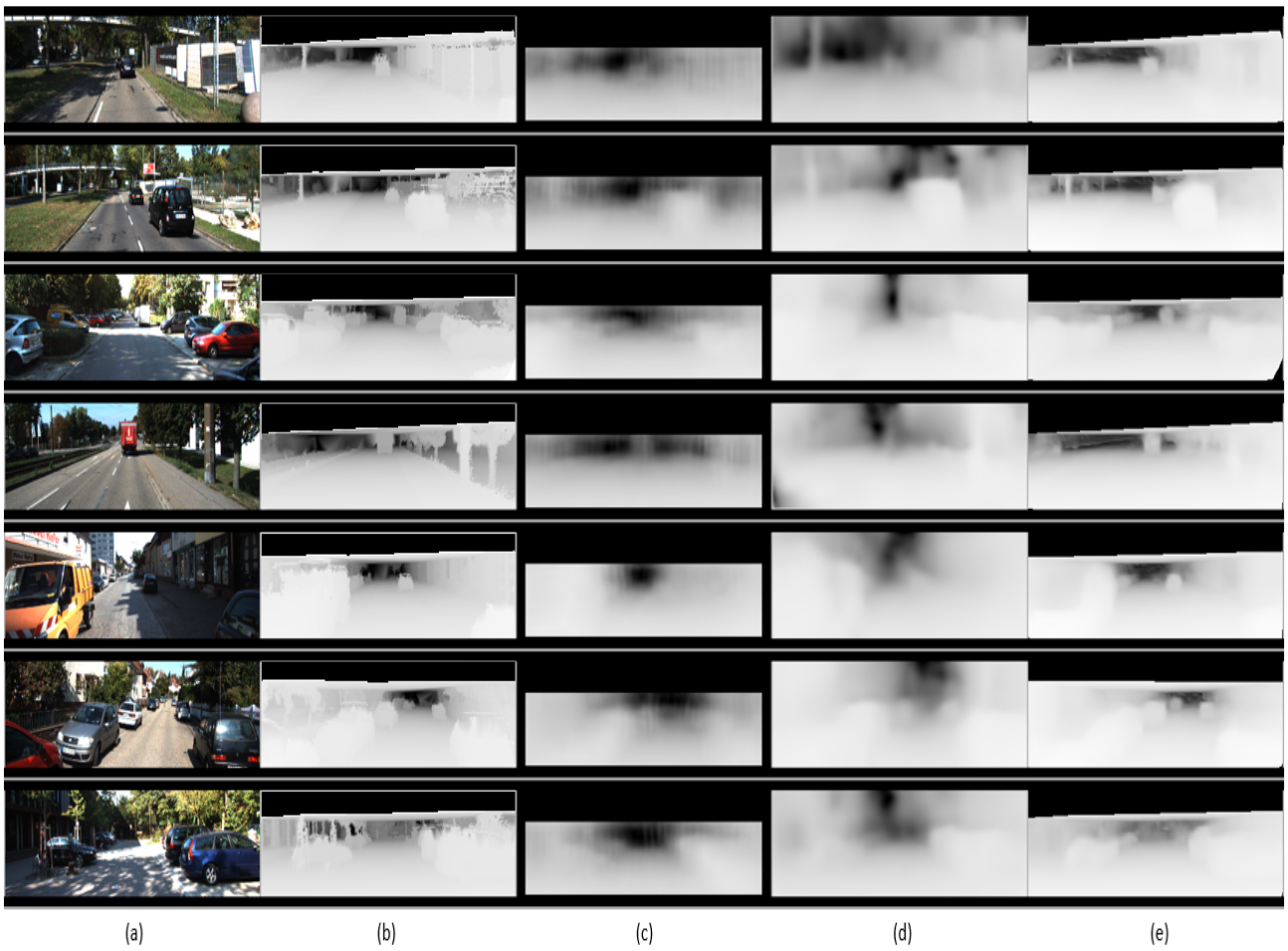

Figure 2: Qualitative results *(good). (a)* Ground truth Image *(b)* Corresponding ground truth depth map *(c)* Depth predictions of Eigen *et al* [9] (Depth supervised) *(d)* Depth predictions of [31] *(e)* Our depth predictions (*depth supervision + photoconsistency + adv. loss*).

unsupervised learning we set $\lambda_1 = 0.995$ and $\lambda_2 = 0.005$.

When training discriminator with adversarial examples, at times, due to the camera motion, regions in the preceding scene, especially corner regions of images, tend to be missing in the current scene causing occlusion. As discussed above, we learn and use the explainability mask to tackle this problem by masking out regions that are occluded. But while training the discriminator, the generated images does not appear realistic because of the mask on the boundaries of the image as shown in Figure 5 column (c). Penalizing such images by letting the discriminator classify these as fake, affects the overall pose learning as it restricts the generator by allowing only a negligible motion for camera, which slows down the overall learning rate indeed, or, at times, halts the learning at all. In order to tackle this problem, we apply occlusion masks computed for projected images, to the real images as well, within the minibatch, so that the discriminator learns to classify images as real or fake, irrespective of the presence of the occlusion/explainability masks. Examples of occlusion masks estimated using the explainability mask prediction is shown in Figure 5 (c)). In addition, to retain the stability of the network throughout the learning, we train the generator and the discriminator with different learning rates, while the learning rate of discriminator is set to be 20 times lower than the generator, to induce more stability in learning. Also, while training discriminator, we randomly shuffle the source/ground truth and generated images (of the mini-batch) before feeding them to the discriminator, so that the discriminator does not learn to associate the generated and ground truth images as pairs and learns to just differentiate between them, instead learns a more generic objective of classifying real *vs.* fake images.

Figure 3: (a) Source images *(left (I_{t-1}) or right (I_{t+1}))* (b) Target image (c) Generated (transformed source) image, using the estimated *pose* and *depth* parameters (d) photoconsistency error between generated *(c)* and *(b)*, (e) Adversarial example with induced photoconsistency loss. The discriminator is trained with images in *(a) and (b)* as real *vs.* images (e) as fake. Image (e) has been enhanced with higher values of ω (=0.6) to amplify the difference for visual demonstration, but during training, the induced photoconsistency loss is kept lower.

Figure 4: Qualitative results of depth prediction on KITTI dataset, where the network is trained in an unsupervised manner. *Top* row consists of target images and the *bottom* row consists of the corresponding predicted depth maps.

5. Experimental Evaluation

For the purpose of evaluation, we train and test our method on the KITTI dataset [14] along with testing on Cityscapes dataset [3], the model that is trained on KITTI dataset. Both KITTI [14] and Cityscapes [3] are similar datasets, comprised of sequences of stereo images, primarily of streets and highways, along with the groundtruth depth maps captured using velodyne laser sensors. With KITTI dataset, we use the train/test split provided by *Eigen et al.* [9], to train the network on 34 image sequences test it on 697 images, whereas for cityscapes we use the default test set provided by the [3]. Also, while training on KITTI, we use the left and right images of the stereo sequences as independent

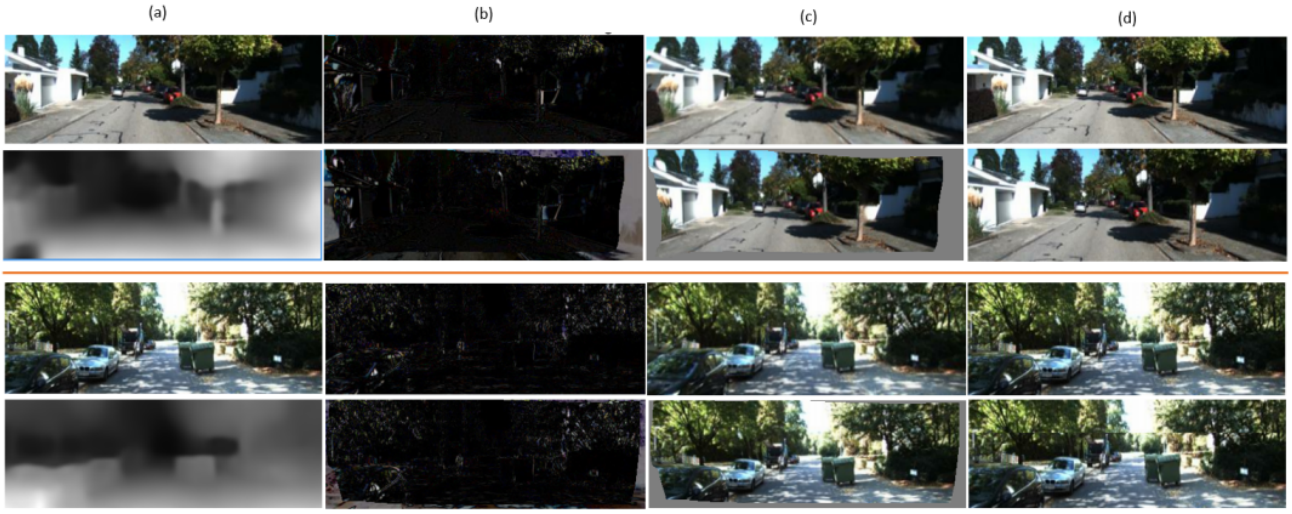

Figure 5: (a) Target image and its estimated depth map, (b) left and right photoconsistency error, (c) Projected left and right images (d) groundtruth left and right images.

θ	Supervision			Error Metric				Accuracy Metric		
	Depth	Pose	Unsupervised	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen *et al. [9]* (Coarse)	✓			0.214	1.605	6.563	0.292	0.673	0.884	0.957
Eigen *et al. [9]* (Fine)	✓			0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al. [11]*	✓			0.202	1.614	6.523	0.275	0.678	0.895	0.965
(**Ours** - *depth+photo.*)	✓			0.1431	0.9741	5.3693	0.2131	0.8001	0.9373	0.9790
(**Ours** - *depth+photo.+adv.*)	✓			**0.1355**	**0.8653**	**5.1736**	**0.2084**	**0.8183**	**0.9450**	**0.9802**
(**Ours** - *depth+photo.+adv.**)	✓			**0.1204**	**0.7466**	**4.7560**	**0.1869**	**0.8486**	**0.9553**	**0.9848**
(**Ours** - *photo.**)			✓	0.2190	1.9758	6.3398	0.2730	0.7081	0.8668	0.9339
(**Ours** - *photo. + adv.**)			✓	0.2114	1.9797	6.1540	0.2636	0.7319	0.8977	0.9593
Godard *et al.* [15]		✓		0.148	1.344	5.927	0.247	0.803	0.922	0.964
Garg *et al.* [13] (50m cap)		✓		0.169	1.080	5.104	0.273	0.740	0.904	0.962
Zhou *et al.* [31](w/o exp. mask)			✓	0.221	2.226	7.527	0.294	0.676	0.885	0.954
Zhou *et al.* [31]			✓	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Zhou *et al.* [31](50m cap)			✓	0.208	1.551	5.452	0.273	0.695	0.900	0.964
Kuznietsov et al. [29]	✓	✓(stereo)		**0.113**	**0.741**	**4.621**	**0.189**	**0.875**	**0.964**	**0.988**
Kuznietsov et al. [29]		✓(stereo)		0.308	9.367	8.700	0.367	0.752	0.904	0.952

Table 1: Comparison of Monocular depth prediction results on KITTI dataset [14]. (*-since our depth prediction is not up to scale, we normalized ground truth and estimated depth maps [31]).

image sequences.

The quantitative evaluation of our algorithm is shown in Table 1. We report the performance of our system using the standard metrics used in [9], for both supervised and unsupervised setting. With supervised learning, we obtain *state-of-the-art* results in comparison with other depth-supervised learning techniques. For the task of depth prediction, we obtain a Root Mean Squared Error *(RMSE)* of 4.75, outperforming all other depth-supervised techniques, while comparing equally with *Kuznietsov et al.* [29], who uses the *stereo* information in addition to using ground truth depth maps for supervision. Moreover, *Kuznietsov et al.* [29] uses a more sophisticated and much deeper ResNet [6] with pre-trained weights learned on ImageNet object classification challenge [23], whereas we are able to achieve comparable results by using a much less sophisticated autoencoders with architecture similar to that of [31], without transferring weights learned from other larger datasets. Figure 3 shows qualitatively, results of depth maps computed using our approach. It has to be noted that our unsupervised learning models (Table 1, *rows 7-8*) were trained only for 75K iterations, which is why the numbers are slightly inferior to Zhou *et al.,* [31]. Our network trained using just photo-consistency loss (Table 1, *row 7*) is coarsely our adaptation

θ	Supervision		Error Metric				Accuracy Metric		
	Depth	Unsupervised	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou et al.*et al. [31]* (50m cap)		✓	0.267	2.686	7.580	0.334	0.577	0.840	0.937
(Ours-*depth+photo.+adv.* (50m cap)*)	✓		0.3934	4.6831	10.5380	0.4123	0.3518	0.6889	0.9047

Table 2: Comparison of Monocular depth prediction results on Cityscapes dataset [3] (* - model trained on KITTI dataset and evaluated on cityscapes dataset, whereas results of [31] are from a model trained explicitly on [3].)

of [31]; the goal of this paper is to demonstrate that the use of the proposed adversarial scheme improves the overall depth prediction performance of the system in comparison with the baseline (Table 1, *row 8*).

Also, for both unsupervised and supervised methods, the estimated depth map is defined up to a scale, so, for the purpose of evaluation, like [31], we normalize them by scaling the predicted depth maps such that their medians match (Table 1). In order to demonstrate the generalizable nature of our architecture, we also test our method on the cityscapes [3] dataset, while the training done solely on the KITTI dataset. Despite being trained in a different dataset, our method performs reasonably well in contrast to other methods that are trained on cityscapes (or cityscapes + KITTI datasets). The results are shown in table 2. Figure 4 shows qualitative results of our method trained unsupervised; we use only the photoconsistency and adversarial losses. We show considerable improvement by training using adversarial loss in contrast to its baseline which is trained using photoconsistency loss alone. We reported results for unsupervised learning at the end of 40K iterations. It has to be noted that the accuracy improves for unsupervised learning with more iterations [31]. Also we observed that when using adversarial loss, it takes slightly longer for convergence than usual, as the adversarial loss penalizes the pose and depth networks, ceaselessly, throughout the training process.

6. Conclusion

We extended a state-of-the-art deep learned depth and pose prediction model and couple it with the adversarial learning to harness the generative power of the GANs, subsequently improving the depth and pose estimation accuracy. We further introduce a technique to generate context-aware adversarial examples, that allows our method to trick the discriminator to work against the generator and at the same time indirectly minimizing the same objective as that of the generator. This proposed method is shown to learn and perform well in both depth-supervised and unsupervised setting, obtaining new state-of-the-art results among depth supervised approaches, and comparing favorably against other pose-supervised and unsupervised techniques. Furthermore, the use of adversarial loss for learning have been successfully demonstrated, both qualitatively and quantitatively, to improve depth and pose prediction accuracy, on two of the important benchmarks.

References

[1] Xu B, Wang N, Chen T, and Li M. Empirical evaluation of rectified activations in convolutional network. In *arXiv preprint arXiv*, page 1505.00853, 2015. 4

[2] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. *Proc. Intl. Conf. on Robotics and Automation*, abs/1606.02378, 2017. 2

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, 2016. 2, 6, 8

[4] Kingma D and Ba J. Adam: A method for stochastic optimization. In *arXiv preprint arXiv*, page 1412.6980, 2014. 4

[5] Pathak D, Krahenbuhl P, Donahue J, and Darrell T. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, pages 2536–2544, 2016. 3, 4

[6] D.He, K Zhang, X Ren S, and Sun J. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 4, 7

[7] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, pages 2758–2766, 2015. 2

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *Proc. ICCV*, abs/1411.4734, 2015. 2

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NIPS*, abs/1406.2283, 2014. 2, 5, 6, 7

[10] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proc. ECCV*, September 2014. 2

[11] Liu F, Shen C, Lin G, and Reid I. Learning depth from single monocular images using deep convolutional neural fields. In *Proc. CVPR*, pages 2024–2039, 2016. 7

[12] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis. *IEEE PAMI*, 32(8):1362–1376, 2010. 1

[13] R. Garg, B. G. Vijay Kumar, and I. D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *Proc. ECCV*, abs/1603.04992, 2016. 7

[14] Andreas Geiger. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. CVPR*, pages 3354–3361, Washington, DC, USA, 2012. 2, 6, 7

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. CVPR*, 2017. 2, 3, 7

[16] Laina I, Rupprecht C, Belagiannis V, Tombari F, and Navab N. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV)*, pages 239–248, 2016. 2

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025. 2015. 2, 3

[18] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114. 2012. 2

[19] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proc. CVPR*, 2016. 2

[20] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. 31(5):1147–1163, Oct 2015. 1

[21] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. Dtam: Dense tracking and mapping in real-time. In *Proc. ICCV*, ICCV '11, pages 2320–2327, Washington, DC, USA, 2011. IEEE Computer Society. 2

[22] Bojan Pepik, Peter Gehler, Michael Stark, and Bernt Schiele. 3d2pm – 3d deformable part models. In *Proc. ECCV*, Lecture Notes in Computer Science, pages 356–370, Firenze, Oct 2012. 2

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. In *IJCV*, 2015. 7

[24] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proc. CVPR*, pages 1067–1073, 1997. 2

[25] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *Proc. ACM SIGGRAPH*, pages 835–846, 2006. 1

[26] James Thewlis, Shuai Zheng, Philip H. S. Torr, and Andrea Vedaldi. Fully-trainable deep matching. *Proc. BMVC.*, abs/1609.03532, 2016. 2

[27] Alexander Thomas, Vittorio Ferrari, Bastian Leibe, Tinne Tuytelaars, and Luc Van Gool. Shape-from-recognition: Recognition enables meta-data transfer. *CVIU*, (12):1222–1234, 2009. 2

[28] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *CoRR*, 2017. 2

[29] Kuznietsov Y, J Stuckler, and B Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proc. CVPR*, pages 6647–6655, 2017. 7

[30] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *J. Machine Learning Research*, 2016. 2

[31] Tinghui Zhou, Matthew Brown, Noah Snavely, and David Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017. 2, 3, 4, 5, 7, 8