Generative adversarial networks for depth map estimation from RGB video

Kin Gwn Lore, Kishore Reddy, Michael Giering, Edgar A. Bernal
United Technologies Research Center
411 Silver Lane, East Hartford CT 06018
(lorek, reddykk, gierinmj, bernalea)@utrc.utc.com

Abstract

Depth cues are essential to achieving high-level scene understanding, and in particular to determining geometric relations between objects. The ability to reason about depth information in scene analysis tasks can often result in improved decision-making capabilities. Unfortunately, depth-capable sensors are not as ubiquitous as traditional RGB cameras, which limits the availability of depth-related cues. In this work, we investigate data-driven approaches for depth estimation from images or videos captured with monocular cameras. We propose three different approaches and demonstrate their efficacy through extensive experimental validation. The proposed methods rely on processing of (i) a single 3-channel RGB image frame, (ii) a sequence of RGB frames, and (iii) a single RGB frame plus the optical flow field computed between the frame and a neighboring frame in the video stream, and map the respective inputs to an estimated depth map representation. In contrast to existing literature, the input-output mapping is not directly regressed; rather, it is learned through adversarial techniques that leverage conditional generative adversarial networks (cGANs).

1. Introduction

By some estimates [1], depth sensing is poised to penetrate the automotive sensor market as the lead technology behind safety and autonomy. Depth cues are essential for high-level scene understanding, as well as to determine the geometric relations between objects in the scene. The ability to reason about depth information in scene analysis tasks can often result in improved decision-making capabilities. Unfortunately, depth-capable sensors are far less ubiquitous as traditional RGB cameras, which in turn limits the availability of depth-related cues.

Literature introducing techniques that leverage stereoscopic images to estimate the depth map of a scene is plentiful. In contrast, we investigate approaches for depth map estimation from images or videos captured with monocular cameras which has significant value given the popularity and low-cost of traditional RGB cameras. Indeed, while stereoscopic images are relatively scarce due to the need for specialized equipment, monocular images and video frames are widely available in the internet, particularly on social media and video-sharing platforms. In addition, the ability to accurately estimate depth from monocular visual data may be useful to improve understanding of historical data available for existing scientific and industrial applications. In some instances, depth maps can be obtained via the use of LiDAR sensors, which measure the distance to a target by determining the time it takes pulses of emitted light to reflect off the target and return to the sensor. LiDAR technologies, however, may suffer from low data acquisition rates; also, LiDAR sensors have a certain degree of sophistication which has prevented them from becoming widespread commodities, unlike consumer-grade cameras. Being able to close the depth-sensing performance gap between the two technologies while leveraging the ubiquity of traditional imaging systems would prove to be impactful to the wide range of applications that benefit from knowledge of depth information.

In this paper, we investigate data-driven approaches for depth estimation from images or videos captured from monocular cameras. We propose three different approaches and demonstrate their efficacy through extensive experimental validation. The proposed methods rely on processing of (i) a single 3-channel RGB image frame, (ii) a sequence of RGB frames, and (iii) a single RGB frame plus the optical flow field computed between the frame and a neighboring frame in the video stream, and map these inputs to an estimated depth map representation. In contrast to existing literature, the input-output mapping is not directly regressed; rather, it is learned through adversarial techniques that leverage conditional generative adversarial networks (cGANs), which have the ability to generalize from smaller training sets.

The paper is organized as follows: in Sec. 2, related work

on technologies mapping from RGB images to depth maps is discussed. Specific details on the methodologies introduced in this paper, including the algorithm descriptions, problem formulation, datasets, and data preparation are presented in Sec. 3. Popular evaluation metrics are outlined in Sec. 4, along with discussions on the main results obtained from our experiments, before concluding the paper in Sec. 5.

2. Related work

Techniques for depth estimation from a single image via supervised learning are plentiful. For instance, [15] implemented a framework using Markov Random Fields (MRF) that incorporates multiscale local- and global-image features and models the relation between depth and visual values at different points. Later, [16] extended the same framework to capture both 3D location and 3D orientation of the scene. However, the approach required some effort in crafting convolutional filters for computation of texture energies and gradients. In [4], the authors construct 3D models from planar RGB images by computing superpixels to harvest the statistics of the geometric classes defined by the orientations of objects in the scene. Lastly, [9] incorporated class labels into the analysis to improve model performance.

A number of machine learning techniques have been explored for depth estimation from stereo imagery [8, 12, 21, 17]. Much of the existing work in this area relies on the use of hand-crafted features including texton features, GIST, SIFT, PHOG, and object banks. The explosion of deep learning techniques has resulted in the enhanced ability to automatically extract features without the need for feature handcrafting. The authors of [11, 20] explored the joint use of deep convolutional neural networks (CNN) and continuous conditional random fields (CRF) to learn depth estimation without relying on geometric priors. The technique in [2] performs depth estimation from a single image using two deep network stacks, with one specializing in coarse global predictions and the other in charge of performing local predictions, while [6] also explored modular CNN architectures for joint depth prediction and semantic segmentation. More recently, [10] proposed a fully convolutional architecture with residual learning to model the ambiguous mapping between monocular images and depth maps.

3. Proposed Approach

This section outlines the proposed data-driven approach for depth map estimation from monocular imagery; training methods and data preprocessing steps are described in detail.

3.1. Conditional GANs

Tne introduction of Generative Adversarial Networks (GANs) [3] represented a significant breakthrough in the field of unsupervised learning. A GAN consists of two modules, a generator and a discriminator, which are usually implemented in the form of neural networks. The generative module captures the distribution of the data, while the discriminative module estimates the probability that a sample to which it is exposed comes from the training data or is synthetically generated. Specifically, generator $G(z;\theta_g)$ builds a mapping function from a noise distribution $p_z(z)$ to the data space, while discriminator $D(x;\theta_d)$ produces a single scalar output representing the probability that observed sample x comes from the actual training data distribution rather than from the learned p_g.

G and D are trained simultaneously. The parameters for G are learned through gradient descent and error backpropagation to minimize loss function $\log(1-D(G(z)))$. At the same time, the parameters for D are learned by minimizing $\log D(X)$. Optimizing both networks can be posed as a two-player minimax game with value function $V(G,D)$:

$$\begin{aligned}&\min_G \max_D V(D,G)\\&= E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_x(z)}\left[1-D(G(z))\right]\end{aligned} \tag{1}$$

GANs can be naturally extended to learn conditional distributions by having the generator and discriminator conditioned on additional information y, rather than on the noise vector z; the networks that result by following this formulation are known as *conditional* generative adversarial networks (cGANs). y can be any auxiliary information such as class labels, actual images, or any data from other modalities. In cGANs, the prior input noise $p_z(z)$ is combined with y to form joint hidden layer representations, which results in a generative model that is capable of transforming samples from one domain into another domain. Applications of such domain transformations include image-to-image translation [5].

The objective function of the conditional two-player minimax game is:

$$\begin{aligned}&\min_G \max_D V(D,G)\\&= E_{x\sim p_{data}(x|y)}\left[\log D(x)\right] + E_{z\sim p_x(z)}\left[1-D(G(z|y))\right]\end{aligned} \tag{2}$$

3.2. Problem formulation

In this paper, we formulate the task of estimating a depth map from monocular inputs as an image translation task. Since the input and the output lie in different domains, we

can perform a pixel-to-pixel mapping between the input and the output, assuming that both have the same spatial dimensions. The three different formulations being proposed are described below and illustrated in Fig. 1.

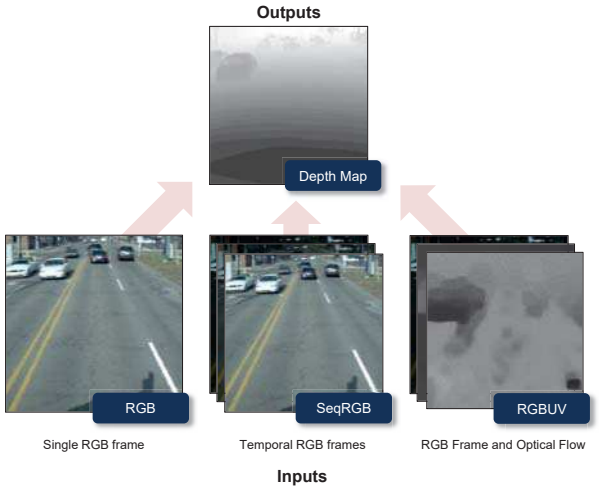

Figure 1. Three formulations of the task at hand. We estimate the depth map of a scene using individual frames (left), sequential frames (center), and optical flow (right) from the video data and attempt to estimate the depth map of the scene as the output. Note that each image frame contains three color channels.

Single RGB Frame to depth mapping. In this formulation, a single RGB frame is directly mapped to a grayscale pixel intensity representation of its corresponding LiDAR depth map. To this end, the architecture outlined in *pix2pix* [5] is modified to accept an 8-bit 3-channel frame (i.e., red, green, and blue channels) of width w and height h to a single intensity map of the same image dimension with 8-bit values.

Sequential RGB Frames to depth mapping. This approach attempts to leverage video data to infer the depth map. According to this formulation, a sequence of 3 RGB frames at time t, $t-1$, and $t-2$ are fed as the input to get a single-channel output representing the depth map at time t. In other words, the input is a concatenation of the 3 RGB frames along the color dimension, producing a total of 9 channels as the input and a single channel as the output. We hypothesize that frames in succession can capture the temporal motion of the objects and aid in inferring depth from motion as well as the nature of the objects. For instance, objects traversing the field of view of a front-facing camera at high speed might most likely be passing traffic, trees while making a turn at an intersection, or crossing pedestrians.

RGB Frame and optical flow to to depth mapping. Similar to the sequential RGB approach, the *optical flow* field between two successive frames at time t and $t-1$ is used to infer the depth map at time t. The RGB data for time t is retained and concatenated with the optical flow components U and V represented as two distinct image channels. As a result, there are 5 input channels (RGB+UV) and a single output channel.

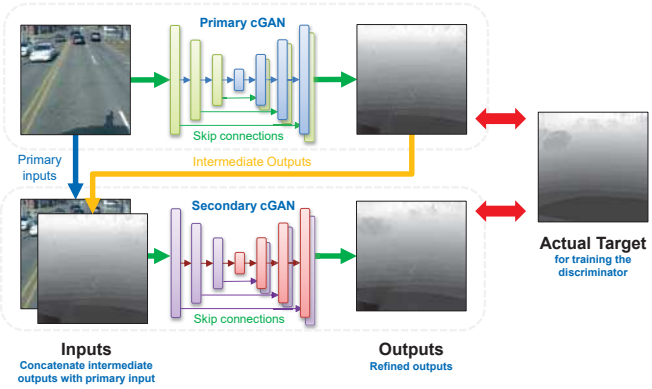

Figure 2. An illustration of cascaded refinement by training multiple GANs in stages. The first stage produces a coarse estimate of the depth map, while the second stage refines the first estimate by processing a fused version of it with the original inputs.

Cascaded model refinement. Additional GANs can be further utilized to refine the outputs in a staged manner. Using the single RGB frame formulation as an example, a GAN is trained to map an RGB frame to a depth map. Next, we introduce a secondary GAN that maps the concatenation of the RGB frame and depth map estimate to a more refined depth map. In other words, the secondary GAN is trained on the concatenation of the inputs and the outputs from the primary GAN.

3.3. Data description

Experiments were performed on both the Ford Campus Vision and LiDAR Data Set [14]. The imagery in the dataset was collected by an autonomous ground vehicle testbed on a Ford pickup truck equipped with multiple sensors including inertial measuring units (IMU), LiDAR scanners, and omnidirectional camera systems. Captured data has timestamps which allows establishing temporal correspondences between the acquired images and LiDAR depth maps. Before feeding the training data into the framework, several additional preprocessing steps on the data had to be performed which will be outlined below.

We compare the performance of our frameworks with other methods on the NYU Depth v2 dataset [13]. This dataset consists of indoor scenes captured using the Microsoft Kinect camera.

3.4. Data preparation

3.4.1 Ford Campus Vision and Lidar dataset

In this section, we discuss the details on how the dataset is prepared for training and testing. The number of frames used for training is 1480.

(i) Vehicle trajectory 1 for training (ii) Vehicle trajectory 2 for testing

Figure 3. The vehicle trajectory is used to determine the members of the training and testing datasets. Doing so guarantees a clean separation of data and minimizes data leakage.

Train-test partition. The dataset contains two sets of data collected from two different routes taken by the vehicle: one around downtown Dearborn and the other around the Ford Research Complex in Dearborn (Fig. 3). In order to minimize the effects of possible data leakage, the data partition is done based on the vehicle trajectory.

Camera selection. There are five cameras available in the dataset, each pointing towards a different orientation (e.g., front of vehicle, side of vehicle, back of vehicle). In all cases, only the first forward-facing camera is selected for training. Later in the section, we will discuss the problem formulation utilizing optical flow and sequential frames where the flow of optical features has an impact on efficient learning of features.

Gamma correction on depth map representation. In the LiDAR data, pixels with low (resp. higher) values represent nearer (resp. further) objects. In a typical depth map, the lower two-thirds of the image tend to be very dark due to most regions (e.g., road and objects on the road) falling within the camera's field of view. On the other hand, the upper third of the image contains more variation due to the presence of different objects within vicinity of the vehicle. Hence, the lower two-third tends to have low pixel values that are tightly distributed, whereas the values of the pixels in the upper third of the image are more widely spread. As a form of equalization, gamma correction is performed on the image to magnify (resp. compress) variations in the near (resp. far) regions.

Prior to gamma correction, the depth values d are first normalized using the following expression:

$$\tilde{d} = \frac{d - d_{lb}}{d_{ub} - d_{lb}}$$

where d_{ub} is the upper bound and d_{lb} is the lower bound of the depth values in the data. Then, the depth map is processed pixel-wise according to the following equation for gamma correction $\tilde{d}_{out} = A\tilde{d}_{in}^{\gamma}$, where $\tilde{d}_{out} \in [0, 1]$ is the normalized intensity of the output pixels, $\tilde{d}_{in} \in [0, 1]$ is the intensity of the input pixels, A is typically a constant with value 1, and γ is the parameter of interest. When $\gamma > 1$, the image is darkened. Therefore, we select values where $\gamma < 1$ (specifically, $\gamma = 1/3$ in our experiments) to make dark regions lighter. An example of the processed data is visualized in Fig. 4.

(i) RGB Frame (i) Non-gamma corrected map (ii) Gamma corrected map

Figure 4. Gamma correction on the LiDAR depth map amplifies darker regions to bring out more details.

Temporal registration between acquired video frames and LiDAR data. The on-board video is captured at 30 fps and LiDAR data is captured in 10 Hz. During the training process, only video frames for which LiDAR data is available were used.

Stationary data pruning. A significant portion of the data (about 40%) was acquired while the vehicle was stationary; consequently, many frames were found to have similar content. This may cause issues with the learning because the dataset can become imbalanced. We removed redundant frames by computing a similarity metric between the current video frame at time t and the previous video frame at time $t-1$, and discarding frames for which the similarity metric does not exceed a certain threshold. Specifically, consider an image frame that is represented by a vector $y \in R^{whc}$, where w, h, and c are the width, height, and number of channels of the image, respectively. Formally:

$$\Delta = \frac{\left(\frac{1}{whc}\sum_{i=0}^{whc}(y_{t,i} - y_{t-1,i})^2\right)^{1/2}}{\frac{1}{whc}\sum_{i=0}^{whc} y_{t,i}}$$

The pruning algorithm preserves frames for which Δ exceeds 0.05. Note that when the vehicle is stationary in front of the road intersection, passing vehicles may enter the upper regions of the field of view. Therefore, the pruning criteria is only done based on the computation on the lower-half of the image.

Training data augmentation. Affine transformations such as flipping are performed. Since we wish to tailor the framework to the estimation of depth maps of roads,

we only perform horizontal flipping (without vertical flipping) to preserve contextual information. As with other data-driven approaches, computing an accurate estimate of the prior distribution of the data is vital. Through vertical flips, objects such as cars and trees will appear upside down and negatively affect the estimate of the distribution.

3.4.2 NYU Depth v2 dataset

Methods in common. Preprocessing methods similar to those described above were applied to this dataset. In particular, we performed gamma correction on the target depth map. During test time, the outputs of the network were reverted back into actual depth values, in meters. During training, the same data augmentation techniques were applied.

Train-test separation. Among the 1449 images from the subset of NYU Depth v2 dataset, we used the first 1000 images for training, with the majority of the scenes having been acquired in kitchens, offices, and classrooms. For testing, we evaluated the performance on the remaining 449 images, where the majority of the scenes were acquired in living rooms and bedrooms. This clear separation between training and testing minimizes data leakage and forces to model to generalize to unseen settings.

3.5. Training procedure

The input and output data were processed based on the methods outlined above. We base the fundamental cGAN building block of our framework on that described in [5]. As the original architecture is designed for image-to-image translation, each image is expected to have 3 channels (RGB). Therefore, we made modifications to the architecture to support ingestion of images with an arbitrary number of input channels and output channels. In all experiments, the number of filters are kept the same, with input sizes of $(256 \times 256 \times C)$, where C denotes the number of channels (which depend on the specific formulation being implemented, as outlined in Sec. 3.2). A separate model is trained for each experiment, with 3 variants of problem formulation, 2 variants to study the effects of data pruning, and 2 variants to study the effects of gamma correction. Thus, the results correspond to a total of 12 model evaluations. Regardless of the varying number of channels, we limit the maximum number of training epochs to 50. Training was performed on an NVIDIA GeForce GTX Titan Black and took 5 hours for each experiment.

Additionally, we chose the top-performing model among the 12 experiments performed on the Ford Campus Vision dataset and extended it into the cascaded refinement formulation described in Sec. 3.2. For the NYU Depth v2 dataset, we only tested the RGB to depth map formulation without considering temporal correlations due to the nature of the dataset.

4. Results and Discussion

In this section, report the performance of the different frameworks.

4.1. Evaluation metrics

We employed multiple evaluation metrics to evaluate the quality of the depth map reconstruction. All evaluation metrics are computed by treating the 8-bit integers as a floating point value without normalizing into a range from 0 to 1.

Error-based metrics. The L2 norm has been a popular metric to measure errors between estimated data and ground truth data. In this context, the *root mean-squared error* (RMSE) between the reconstructed single-channel lidar depth map $\hat{y}$ and the ground truth y is computed via:

$$\text{RMSE}(y, \hat{y}) = \left(\frac{1}{wh} \sum_{i=0}^{wh} (\hat{y}_i - y_i)^2 \right)^{1/2}$$

Normalizing the RMSE facilitates the comparison between datasets or models with different scales. Normalization yields the *normalized root mean-squared error* (NRMSE), which is computed via:

$$\text{NRMSE}(y, \hat{y}) = \frac{\text{RMSE}}{\max_i(\hat{y} \oplus y)_i - \min_i(\hat{y} \oplus y)_i}$$

where $\oplus$ is the concatenation operator. In other words, the normalized RMSE is computed by dividing the RMSE by the difference between the global maximum and the global minimum of the image pair. Usually, NRMSE is reported as a percentage where lower values indicate less residual variance. In many cases, especially for smaller samples, the sample range is likely to be affected by the size of sample which would hamper comparisons.

Relative errors. We used a scale-invariant error (SIE) to measure the relationships between points in the scene, irrespective of the absolute global scale as in [2]. The scale-invariant mean squared error (in log scale) is expressed as:

$$\text{SIE}(y, \hat{y}) = \frac{1}{2wh} \sum_{i=1}^{wh} (\log \hat{y}_i - \log y_i + \alpha(\hat{y}, y))^2$$

where $\alpha(\hat{y}, y) = \frac{1}{wh} \sum_i (\log \hat{y}_i - \log y_i)$ is the value of α that minimizes the error for a given $(\hat{y}, y)$. For a prediction $\hat{y}$, e^{α} is the scale that best aligns it to the ground truth. All scalar multiples of $\hat{y}$ yield the same error, hence the scale invariance.

We also employed metrics that are widely used in the literature, such as the average $\log_{10}$ error:

$$\log_{10}\text{error}(y, \hat{y}) = \frac{1}{wh}\sum_{i=1}^{wh} |\log_{10} y_i - \log_{10} \hat{y}_i|$$

and the average relative error:

$$\text{rel}(y, \hat{y}) = \frac{1}{wh}\sum_{i=1}^{wh} \frac{|y_i - \hat{y}_i|}{y_i}$$

Structural similarity index (SSIM). While metrics such as MSE estimate absolute errors, SSIM is a perception-based metric that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. Structural information is the idea that the pixels have strong inter-dependencies especially when they are spatially close. These dependencies carry important information about the structure of the objects in a visual rendering of a scene. Note that by using this metric, we retain the original 2D structure of the image (as opposed to using a vector notation) since SSIM is computed on windows of images. For more information, we redirect the readers to [19].

4.2. Performance evaluation

The quality of the depth map reconstruction as effected by the proposed frameworks is presented next. Table 1 shows the reconstruction metrics for the different experimental setups. The SSIM for all approaches are very similar indicating that both the reconstruction and the actual depth maps are very similar in terms of the presence of local structures. However, the ability to reconstruct object structures does not necessarily imply high accuracy in terms of depth map reconstruction. It is equally important for the estimated values to be as close to the target values as possible. This capability is better captured by the rest of metrics. While each one of these metrics is designed to address a shortcoming exhibited by another metric, we found their difference estimates to be highly correlated. Consequently, only the box plots of the SSIM metric (see Fig. 6) and of the $\log_{10}$ error (see Fig. 7) are visualized for the sake of brevity.

4.2.1 Ford Campus Vision and Lidar dataset

Temporal information is beneficial. In all cases, it is observed that using optical flow information generally results in the best per-model performance in terms of reconstruction quality measured in SSIM and all error-based metrics, regardless of whether the data has been pruned or gamma corrected. The next best performance is achieved by using a sequence of frames, while using a single frame resulted in the least accurate reconstructions. This is in line with our expectations, where leveraging the temporal correlation between frames boosts performance. In practical applications, storing a sequence of recent RGB frames may be less computationally expensive than computing the optical flow between frames at every step.

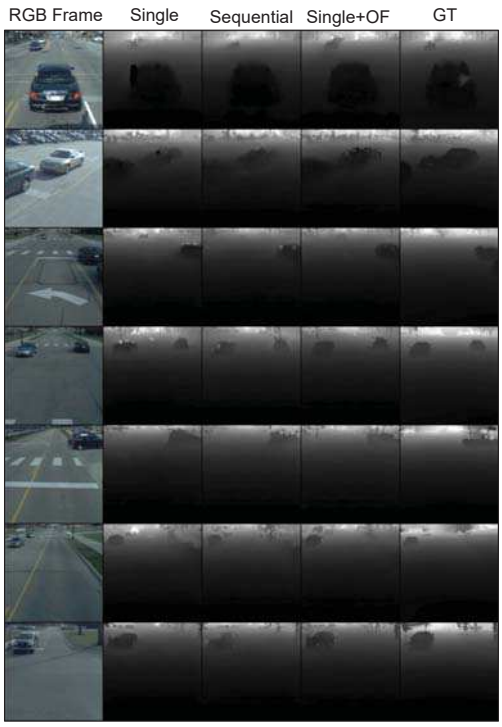

Figure 5. Output samples resulting from the implementation of different formulations. (i) Single: Using a single RGB frame. (ii) Sequential: Using 3 consecutive RGB frames. (iii) Single+OF: Single RGB frame with optical flow. (iv) GT: Ground truth. Outputs have been gamma-corrected for visualization purposes. The first two rows are sample outputs from the training set, whereas the rest are outputs from the testing set. While artifacts around the objects are visible, we note the learning task itself is challenging due to the noise present in the ground truth. An interesting result (see third row from the top) shows an accurately reconstructed approaching car that is not captured by the LiDAR measurements.

Data pruning improves performance. Generally, pruning stationary data for training results in better performance. As stated, the full dataset contains many frames with similar appearance due them having been captured while the vehicle was stationary. Consequently, the estimate of the prior distribution is skewed and affects the generalization capabilities of the model.

Gamma correction aids learning. It is unclear how gamma correction on the target affects the structural similarity measure between the target depth map and the reconstruction. The difference is, however, apparent if we look

Table 1. Reconstruction performance on the Ford Campus Vision and LiDAR test sets in terms of SSIM (higher is better) and other metrics (for which lower is better). The non-perceptual error metrics are highly correlated. Computed errors are based on the distance between the *pixel intensities* of the target image map and those of the input image map.

No Gamma Adjustment	Dataset	SSIM	RMSE	NRMSE	SIE	rel	$\log_{10}$
Single Frame	Full	0.8768	23.2945	0.0916	0.0754	0.2331	0.0984
	Pruned	0.8799	23.6085	0.0928	0.0820	0.2445	0.0973
Sequential Frames	Full	0.8758	23.1657	0.0911	0.0880	0.2490	0.0979
	Pruned	0.8816	22.8471	0.0898	0.0830	0.2307	0.0933
Single Frame + Optical Flow	Full	0.8854	22.3090	0.0877	0.0792	0.2255	0.0876
	Pruned	0.8818	22.2951	0.0876	0.0796	0.2378	0.0907
With Gamma Adjustment	**Dataset**	**SSIM**	**RMSE**	**NRMSE**	**SIE**	**rel**	$\log_{10}$
Single Frame	Full	0.8729	10.9335	0.0485	0.0274	0.0940	0.0320
	Pruned	0.8670	11.6861	0.0514	0.0270	0.0904	0.0312
Sequential Frames	Full	0.8654	11.6509	0.0470	0.0370	0.1014	0.0367
	Pruned	0.8834	10.2161	0.0447	0.0258	0.0829	0.0277
Single Frame + Optical Flow	Full	0.8748	11.5128	0.0515	0.0266	0.0910	0.0317
	Pruned	**0.8858**	**9.6548**	**0.0425**	**0.0252**	**0.0817**	**0.0275**
With Gamma Adjustment	**Dataset**	**SSIM**	**RMSE**	**NRMSE**	**SIE**	**rel**	$\log_{10}$
Best from above + Cascaded Refinement	Pruned	0.8739	12.1008	0.0535	0.0371	0.2003	0.0404

Table 2. Reconstruction performance on NYU Depth V2. For consistent comparison with other literatures, computed errors are based on the *depth values* recovered from inverse gamma correction where the 255×255-pixel output is resized back into its original size using nearest neighbor interpolation. RMSE errors are computed based on depth values in meters. The list is sorted from highest to lowest RMSE with an asterisk (*) denoting our method.

Evaluation on NYU Depth v2	Method	Modalities	RMSE	NRMSE	SIE	rel	$\log_{10}$
Karsch et al. [7]	Non-parametric sampling	Depth only	1.200				
Eigen et al. [2]	Multi-scale deep network	Depth only	0.907		0.219	0.215	
*Ours - Single Frame cGAN	Conditional GAN	Depth only	0.875	0.179	0.063	0.255	0.102
*Ours - Cascaded Refinement cGAN	Conditional GAN	Depth only	0.862	0.173	0.064	0.235	0.100
Liu et al. [11]	Deep conditional neural field	Depth only	0.824			0.230	0.095
Wang et al. [18]	Hierarchical conditional random field	Depth + Semantic	0.745			0.220	0.094
Jafari et al. [6]	Joint refinement network	Depth + Semantic	0.673			0.157	0.068

at the $\log_{10}$ error. While there are larger variances of pixel intensity in the target data, the model is able to produce reconstructions over a high dynamic range while maintaining small errors. This is largely related to the training of neural networks where it is more desirable where each update step during backpropagation can result in an impactful decrease in losses. As the training samples is considered limited, scaling the values also helped the model to converge faster.

4.2.2 NYU Depth v2 dataset and comparison to existing methods

Results are shown in Table 2. While our method is shown to outperform traditional non-parametric sampling methods [7] and deep networks [2], the current cGAN model which performs pixel-to-pixel image translation is still behind advanced techniques that use superpixels for context [11] or that incorporate semantic segmentation for depth estimation [18, 6]. (See **Limitations** in this section.)

On cascaded refinement: For the Ford Campus Vision and Lidar dataset, we observed that using temporal information, data pruning, and gamma correction resulted in the best performance among all experiments. Hence, we used the same training scheme and trained another GAN as the second stage to refine the outputs from the first stage. From our observations in Table 1, we find no improvement in the application of cascaded refinement. However, the opposite is true when cascaded refinement is applied on the NYU Dataset. This is perhaps due to the NYU Dataset having better (i.e., cleaner) ground truth maps compared to the jittery depth values present in the Ford dataset.

Limitations. Using only 1000 images for training, the model can sufficiently generalize to the unseen test set with comparable performance. Using cGANs has its shortcomings, however; as a generative model, cGANs are prone to the issue where they learn mappings to generate *realistic-looking* images instead of generating *accurate* images. This artifact is most likely seen in the last row, last column example given in Fig. 8. While the ground truth contains a desk in the foreground, the trained cGAN *hallucinated* the flat surface of the desk as the floor. Interestingly, due to contextual cues, this 'floor' leads up to a hallucinated doorway when in fact the actual object is a window. We believe that the model can be improved with more data, in addition to allowing some mixture in environmental settings that are common to both the training and testing set instead of performing train-test split over different environments. This hallucinating behavior contributed to larger RMSE which penalized the performance of the GAN-based approach against regression

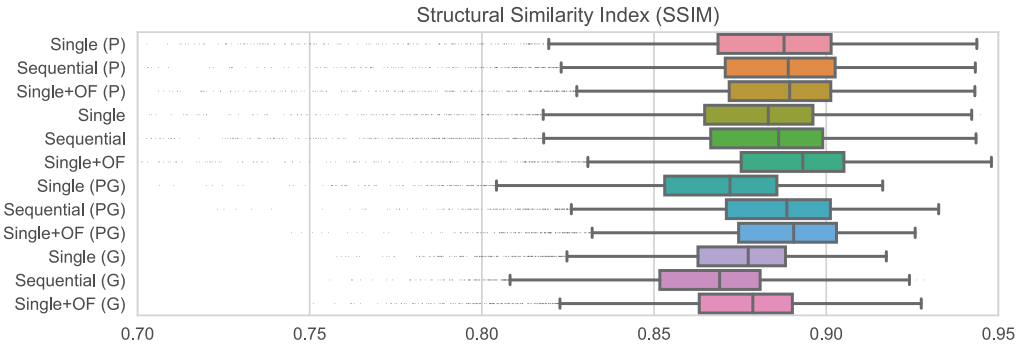

Figure 6. Box plot of the SSIM visualizing the distribution of reconstruction error for each frame in the FORD test set. P indicates that the dataset has been *pruned* to remove stationary data as opposed to using the full dataset. G indicates that the depth map has been gamma corrected. PG means that both processing techniques have been applied. The abbreviation OF means optical flow. Higher scores are better. An SSIM of 1 is the highest possible score indicating perfect reconstruction.

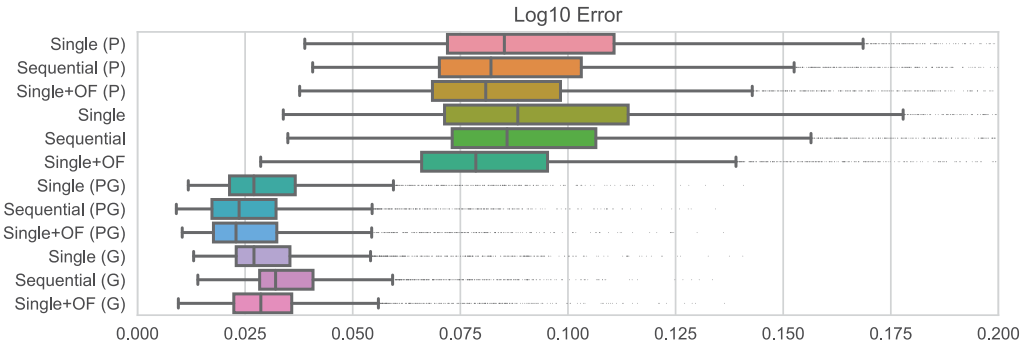

Figure 7. Box plot of the $\log_{10}$ error visualizing the distribution of reconstruction error for each frame in the FORD test set. The meaning of the abbreviations is the same as in Table 7. Lower scores are better.

methods.

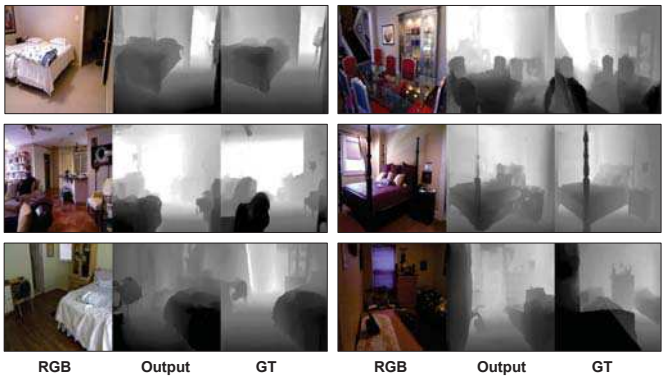

Figure 8. Outputs from the cascade refinement framework on the NYU Depth v2 dataset. Left: RGB input. Middle: Outputs from the proposed cascade refinement framework. Right: ground truth map.

5. Conclusion

The task of estimating a depth map from a single image is an area of active research. By using generative adversarial networks to learn the input-output mapping between two domains, we were able to construct a generative model that generalizes well to unseen test data. In this work, we used cGANs to map RGB images, as well as sequences of frames and optical flow information to the depth map of the scene, where the ground truth is measured using LiDAR. We found that using temporal information improves the model performance, while comparison with state-of-the-art yields comparable performance with many potential areas of improvements. Taking inspiration from works that performed better, our future research directions include:

- Using GPS data to infer vehicle velocity to intelligently sample frames for optical flow computations;
- Extending the framework to ingest auxiliary sensor information; and
- Incorporating semantic segmantation and object recognition in the cGAN model for depth estimation.

References

[1] Lidar: Driving the future of autonomous navigation. Technical report, Frost and Sullivan, the Growth Partnership Company, 2016. 1

[2] 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. 2, 5, 7

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[4] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM transactions on graphics (TOG)*, 24(3):577–584, 2005. 2

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016. 2, 3, 5

[6] O. H. Jafari, O. Groth, A. Kirillov, M. Y. Yang, and C. Rother. Analyzing modular cnn architectures for joint depth prediction and semantic segmentation. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4620–4627. IEEE, 2017. 2, 7

[7] K. Karsch, C. Liu, and S. Kang. Depth extraction from video using non-parametric sampling. *Computer Vision–ECCV 2012*, pages 775–788, 2012. 7

[8] K. Konda and R. Memisevic. Unsupervised learning of depth and motion. *arXiv preprint arXiv:1312.3429*, 2013. 2

[9] L. Ladicky, J. Shi, and M. Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–96, 2014. 2

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

[11] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015. 2, 7

[12] R. Memisevic and C. Conrad. Stereopsis via deep learning. In *NIPS Workshop on Deep Learning*, volume 1, page 2, 2011. 2

[13] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 3

[14] G. Pandey, J. R. McBride, and R. M. Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011. 3

[15] 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. 2

[16] A. Saxena, M. Sun, and A. Y. Ng. Learning 3-d scene structure from a single still image. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 2

[17] F. H. Sinz, J. Q. Candela, G. H. Bakır, C. E. Rasmussen, and M. O. Franz. Learning depth from stereo. In *Joint Pattern Recognition Symposium*, pages 245–252. Springer, 2004. 2

[18] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2800–2809, 2015. 7

[19] 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. 6

[20] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of CVPR*, 2017. 2

[21] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous markov random fields for robust stereo estimation. *Computer Vision–ECCV 2012*, pages 45–58, 2012. 2