Veritatem Dies Aperit - Temporally Consistent Depth Prediction Enabled by a Multi-Task Geometric and Semantic Scene Understanding Approach

Amir Atapour-Abarghouei[1] Toby P. Breckon[1,2]
[1]Department of Computer Science – [2]Department of Engineering
Durham University, UK
{amir.atapour-abarghouei,toby.breckon}@durham.ac.uk

Abstract

Robust geometric and semantic scene understanding is ever more important in many real-world applications such as autonomous driving and robotic navigation. In this paper, we propose a multi-task learning-based approach capable of jointly performing geometric and semantic scene understanding, namely depth prediction (monocular depth estimation and depth completion) and semantic scene segmentation. Within a single temporally constrained recurrent network, our approach uniquely takes advantage of a complex series of skip connections, adversarial training and the temporal constraint of sequential frame recurrence to produce consistent depth and semantic class labels simultaneously. Extensive experimental evaluation demonstrates the efficacy of our approach compared to other contemporary state-of-the-art techniques.

1. Introduction

As scene understanding grows in popularity due to its applicability in many areas of interest for industry and academia, scene depth has become ever more important as an integral part of this task. Whilst in many current autonomous driving solutions, imperfect stereo camera set-ups or expensive LiDAR sensors are used to capture depth, research has recently focused on refining estimated depth with corrupted or missing regions in post-processing, rendering it more useful in any downstream applications [6, 78, 84]. Moreover, monocular depth estimation has received significant attention within the research community as a cheap and innovative alternative to other more expensive and performance-limited technologies [8, 24, 29, 87].

Pixel-level image understanding, namely semantic segmentation, also plays an important role in many vision-based systems. Significant success has been achieved using Convolutional Neural Networks (CNN) in this field [10, 17, 53, 66, 70] and many others such as image classification [54], object detection [88] and alike in recent years.

Veritatem Dies Aperit: Time discovers the truth.

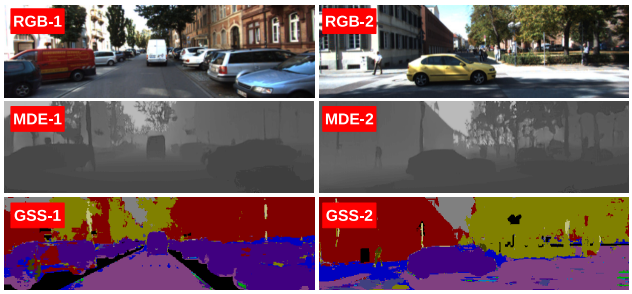

Figure 1: Exemplar results of the proposed approach. **RGB:** input colour image; **MDE:** Monocular Depth Estimation; **GSS:** Generated Semantic Segmentation.

In this work, we propose a model capable of semantically understanding a scene by jointly predicting depth and pixel-wise semantic classes (Figure 1). The network performs semantic segmentation (Section 3.3) along with monocular depth estimation (*i.e.*, predicting scene depth based on a single RGB image) or depth completion (*i.e.*, completing missing regions of existing depth sensed through other imperfect means, Section 3.2). Our approach performs these tasks within a single model (Figure 2 (A)) capable of two separate scene understanding objectives requiring low-level feature extraction and high-level inference, which leads to improved and deeper representation learning within the model [41]. This is empirically demonstrated via the notably improved results obtained for each individual task when performed simultaneously in this manner.

Within the current literature, many techniques focus on individual frames to spatially accomplish their objectives, ignoring temporal consistency in video sequences, one of the most valuable sources of information widely available within real-world applications. In this work, we propose a feedback network that at each time step takes the output generated at the previous time step as a recurrent input. Furthermore, using a pre-trained optical flow estimation model, we ensure the temporal information is explicitly considered by the overall model during training (Figure 2 (A)).

In recent years, skip connections have been proven to

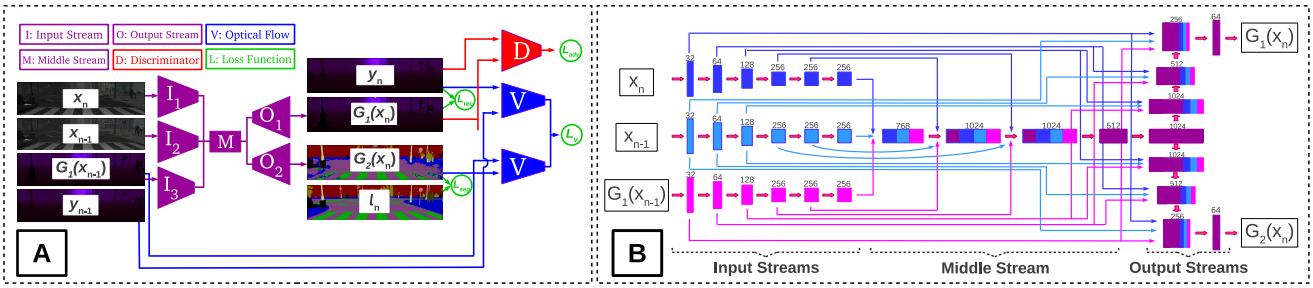

Figure 2: Overall training procedure of the model (A) and the detailed outline of the generator architecture (B).

be very effective when the input and output of a CNN share similar high-level spatial features [60, 66, 73, 79]. We make use of a complex network of skip connections throughout the architecture to guarantee that no high-level spatial features are lost during training as the features are downsampled. In short, our main contributions are as follows:

- *Depth Prediction* - via a supervised multi-task model adversarially trained using complex skip connections that can predict depth (monocular depth estimation and depth completion) having been trained on high-quality synthetic training data [67] (Section 3.2).
- *Semantic Segmentation* - via the same multi-task model, which is capable of performing the task of semantic scene segmentation as well as the aforementioned depth estimation/completion (Section 3.3).
- *Temporal Continuity* - temporal information is explicitly taken into account during training using both recurrent network feedback and gradients from a pre-trained frozen optical flow network.

This leads to a novel scene understanding approach capable of temporally consistent geometric depth prediction and semantic scene segmentation whilst outperforming prior work across the domains of monocular depth estimation [8, 25, 29, 49, 83, 87], completion [9, 36, 50, 82] and semantic segmentation [10, 17, 40, 52, 53, 59, 74, 75, 86].

2. Related Work

We consider relevant prior work over three distinct areas, semantic segmentation (Section 2.1), monocular depth estimation (Section 2.2), and depth completion (Section 2.3).

2.1. Semantic Segmentation

Within the literature, promising results have been achieved using fully-convolutional networks [53], saved pooling indices [10], skip connections [66], multi-path refinement [48], spatial pyramid pooling [85], attention modules focusing on scale or channel [18, 81] and others.

Temporal information in videos has also been used to improve segmentation accuracy or efficiency. [26] proposes a spatio-temporal LSTM based on frame features for higher accuracy. Labels are propagated in [58] using gated recurrent units. In [27], features from preceding frames are warped via flow vectors to reinforce the current frame features. On the other hand, [69] reuses previous frame features to reduce computation. In [89], an optical flow network [23] is used to propagate features from key frames to the current one. Similarly, [77] uses an adaptive key frame scheduling policy to improve both accuracy and efficiency. Additionally, [47] proposes an adaptive feature propagation module that employs spatially variant convolutions to fuse the frame features, thus further improving efficiency. Even though the main objective of this work is not semantic segmentation, it can be demonstrated that when the main objective (depth prediction) is performed alongside semantic segmentation, the results are superior to when the tasks are performed individually (Table 1).

2.2. Monocular Depth Estimation

Estimating depth from a single colour image is very desirable as unlike stereo correspondence [68], structure from motion [16] and alike [1, 71], it leads to a system with reduced size, weight, power and computational requirements. For instance, [11] employs sparse coding to estimate depth, while [24, 25] generates depth from a two-scale network trained on RGB and depth. Other supervised models such as [45, 46] have also achieved impressive results despite the scarcity of ground truth depth for supervision.

Recent work has led to the emergence of new techniques that calculate disparity by reconstructing corresponding views within a stereo correspondence framework without ground truth depth. The work by [76] learns to generate the right view from the left image used as the input while producing an intermediary disparity map. Likewise, [29] uses bilinear sampling [39] and left/right consistency incorporated into training for better results. In [87], depth and camera motion are estimated by training depth and pose prediction networks, indirectly supervised via view synthesis. The model in [44] is supervised by sparse ground truth depth and the model is then enforced within a stereo framework via an image alignment loss to output dense depth.

Additionally, contemporary supervised approaches such as [8] have taken to using synthetic depth data to produce sharp and crisp depth outputs. In this work, we also utilize synthetic data [67] in a directly supervised training framework to perform the task of monocular depth estimation.

Method	Depth Error (lower, better)				Depth Accuracy (higher, better)			Segmentation (higher, better)	
	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$	Accuracy	IoU
Two Models	0.245	1.513	6.323	0.274	0.803	0.856	0.882	0.604	0.672
One Model	0.208	1.402	6.026	0.269	0.836	0.901	0.926	0.748	0.764

Table 1: Comparison of depth prediction and segmentation tasks performed in one single network and two separate networks.

Figure 3: Comparing the results of the approach on synthetic test set when the model is trained with and without temporal consistency. **RGB:** input colour image; **GTD:** Ground Truth Depth; **GTS:** Ground Truth Segmentation; **TS:** Temporal Segmentation; **TD:** Temporal Depth; **NS:** Non-Temporal Segmentation; **ND:** Non-Temporal Depth.

2.3. Depth Completion

While colour image inpainting has been a long-standing and well-established field of study [3, 13, 21, 62, 72, 80], its use within the depth modality is considerably less effective [6]. There have been a variety of depth completion techniques in the literature including those utilizing smoothness priors [33], exemplar-based depth inpainting [7], low-rank matrix completion [78], object-aware interpolation [5], tensor voting [43], Fourier-based depth filling [9], background surface extrapolation [55, 57], learning-based approaches using deep networks [4, 84], and alike [12, 19, 51]. However, prior work does not include any work focusing on enforcing temporal continuity in a learning-based approach.

3. Proposed Approach

Our approach is designed to perform two tasks using a single joint model: depth estimation/completion (Section 3.2) and semantic segmentation (Section 3.3). This has been made possible using a synthetic dataset [67] in which both ground truth depth and pixel-wise segmentation labels are available for video sequences of urban driving scenarios.

3.1. Overall Architecture

Our single network takes three different inputs producing two separate outputs for two tasks - *depth prediction and semantic segmentation*. Moreover, temporal information is explicit in our formulation, as one of the inputs at every time step is an output from the previous time step via recurrence. The network comprises three different components: the input streams (Figure 2 (B) - left), in which the inputs are encoded, the middle stream (Figure 2 (B) - middle), which fuses the features and begins the decoding process, and finally the output streams (Figure 2 (B) - right), in which the results are generated.

As seen in Figure 2 (A), two of the inputs are RGB or RGB-D images (depending on whether monocular depth estimation to create depth, or depth completion to fill holes within an existing depth image, is the focus) from the current and previous time steps. The two input streams that decode these share their weights. The third input is the depth generated at the previous time step. The middle section of the network fuses and decodes the input features and finally the output streams produce the results (scene depth and segmentation). Every layer of the network contains two convolutions, batch normalization [37] and PReLU [31].

Following recent successes of approaches using skip connections [60, 66, 73, 79], we utilize a series of skip connections within our architecture (Figure 2 (B)). Our inputs and outputs, despite containing different types of information (RGB, depth and pixel-wise class labels), relate to consecutive frames from the same scene and therefore, share high-frequency information such as certain object boundaries, structures, geometry and alike, ensuring skip connections can be of significant value in improving the results. By combining two separate objectives (predicting depth and pixel-wise class labels) within our network, in which the input streams and middle streams are fully trained on both tasks, the results are better than when two separate networks are individually trained to perform the same tasks (Table 1).

Even though the entire network is trained as one entity, in our discussions, the parts of the network responsible for predicting depth will be referred to as G_1 and the portions involved in semantic segmentation G_2. These two modules are essentially the same except for their output streams.

3.2. Depth Estimation / Completion

We consider depth prediction as a supervised image-to-image translation problem, wherein an input RGB image (for depth estimation) or RGB-D image (with the depth channel containing holes for depth completion) is translated to a complete depth image. More formally, a generative model (G_1) approximates a mapping function that takes as its input an image x (RGB or RGB-D with holes) and outputs an image y (complete depth image) $G_1 : x \rightarrow y$.

The initial solution would be to minimize the Euclidean distance between the pixel values of the output ($G_1(x)$) and the ground truth depth (y). This simple reconstruction mechanism forces the model to generate images that

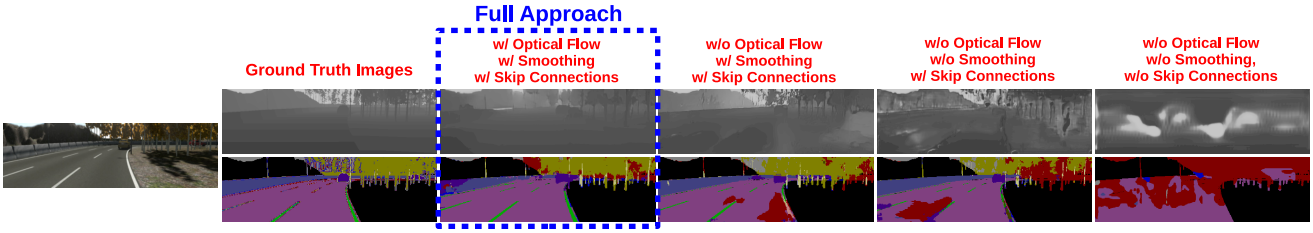

Figure 4: Comparing the performance of the approach with differing components of the loss function removed.

Method	Depth Error (lower, better)				Depth Accuracy (higher, better)			Segmentation (higher, better)	
	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$	Accuracy	IoU
T/R	0.991	1.964	7.393	0.402	0.598	0.684	0.698	0.156	0.335
T/R/A	0.851	1.798	6.826	0.368	0.692	0.750	0.778	0.341	0.435
T/R/A/SC	0.655	1.616	6.473	0.278	0.753	0.812	0.838	0.669	0.738
T/R/A/SC/S	0.412	1.573	6.256	0.258	0.793	0.875	0.887	0.693	0.741
N/R/A/SC/S	0.534	1.602	6.469	0.275	0.758	0.820	0.856	0.614	0.681
T/R/A/SC/S/OF	**0.208**	**1.402**	**6.026**	**0.269**	**0.836**	**0.901**	**0.926**	**0.748**	**0.764**

Table 2: Numerical results with different components of loss. **T:** Temporal training; **T:** Non-Temporal training; **R:** Reconstruction loss; **A:** Adversarial loss; **SC:** Skip Connections; **S:** Smoothing loss; **OF:** Optical Flow.

are structurally and contextually close to the ground truth. For monocular depth estimation, this reconstruction loss is:

$$\mathcal{L}_{rec} = ||G_1(x) - y||_1, \tag{1}$$

where x is the input image, $G_1(x)$ is the output and y the ground truth. For depth completion, however, the input x is a four-channel RGB-D image with the depth containing holes that would occur during depth sensing. Since we use synthetic data [67], we only have access to hole-free pixel-perfect ground truth depth. While one could naïvely cut out random sections of the depth image to simulate holes, as other approaches have done [62, 80], we opt for creating realistic and semantically meaningful holes with characteristics of those found in real-world images [6]. A separate model is thus created and tasked with predicting where holes would be by means of pixel-wise segmentation. A number of stereo images $(30,000)$ [28] are used to train the *hole prediction* model by calculating the disparity using Semi-Global Matching [34] and generating a hole mask (M) which indicates which image regions contain holes. The left RGB image is used as the input and the generated mask as the ground truth label, with cross-entropy as the loss function.

When our main model is being trained to perform depth completion, the hole mask generated by the *hole prediction* network is employed to create the depth channel of the input RGB-D image. Subsequently, the reconstruction loss is:

$$\mathcal{L}_{rec} = ||(1 - M) \odot G_1(x) - (1 - M) \odot y||_1, \tag{2}$$

where $\odot$ is the element-wise product operation and x the input RGB-D image in which the depth channel is $y \odot M$. Experiments with an $L2$ loss returned similar results.

However, the sole use of a reconstruction loss would lead to blurry outputs since monocular depth estimation and depth completion are multi-modal problems, *i.e.*, several plausible depth outputs can correctly correspond to a region of an RGB image. This multi-modality results in the generative model (G_1) averaging all possible modes rather than selecting one, leading to blurring effects in the output. To prevent this, adversarial training [30] has become prevalent within the literature [8, 22, 38, 62, 80] since it forces the model to select a mode from the distribution resulting in better quality outputs. In this vein, our depth generation model (G_1) takes x as its input and produces fake samples $G_1(x) = \tilde{y}$ while a discriminator (D) is adversarially trained to distinguish fake samples $\tilde{y}$ from ground truth samples y. The adversarial loss is thus as follows:

$$\begin{aligned}\mathcal{L}_{adv} = \min_{G_1} \max_{D} & \mathop{\mathbb{E}}_{x,y\sim\mathbb{P}_d(x,y)} [logD(x,y)] + \\ & \mathop{\mathbb{E}}_{x\sim\mathbb{P}_d(x)} [log(1 - D(x, G_1(x)))],\end{aligned} \tag{3}$$

where $\mathbb{P}_d$ is the data distribution defined by $\tilde{y} = G_1(x)$, with x being the generator input and y the ground truth.

Additionally, a smoothing term [29, 32] is utilized to encourage the model to generate more locally-smooth depth outputs. Output depth gradients $(\partial G_1(x))$ are penalized using $L1$ regularization, and an edge-aware weighting term based on input image gradients (∂x) is used since image gradients are stronger where depth discontinuities are most likely found. The smoothing loss is therefore as follows:

$$\mathcal{L}_s = |\partial G_1(x)| e^{||\partial x||}, \tag{4}$$

where x is the input and $G_1(x)$ the depth output. The gradients are summed over vertical and horizontal axes.

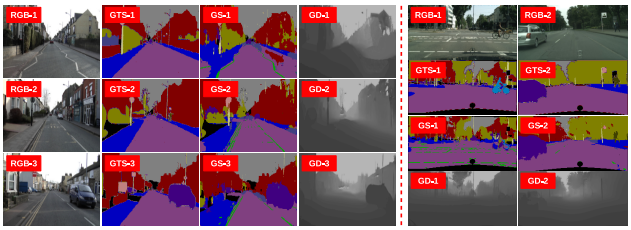

Figure 5: Results on CamVid [14] (left) and Cityscapes [20] (right). **RGB:** input colour image; **GTS:** Ground Truth Segmentation; **GS:** Generated Segmentation; **GD:** Generated Depth.

Method	IoU	Method	IoU
CRF-RNN [86]	62.5	DeepLab [17]	63.1
Pixel-level Encoding [74]	64.3	FCN-8s [53]	65.3
DPN [52]	66.8	Our Approach	**67.0**

Table 3: Segmentation on the Cityscapes [20] test set.

Another important consideration is ensuring the depth outputs are temporally consistent. While the model is capable of implicitly learning temporal continuity when the output at each time step is recurrently used as the input at the next time step, we incorporate a light-weight pre-trained optical flow network [65], which utilizes a coarse-to-fine spatial pyramid to learn residual flow at each scale, into our pipeline to explicitly enforce consistency in the presence of camera/scene motion. At each time step n, the flow between the ground truth depth frames n and $n-1$ is estimated using our pre-trained optical flow network [65] as well as the flow between generated outputs from the same frames. The gradients from the optical flow network (F) are used to train the generator (G_1) to capture motion information and temporal continuity by minimizing the End Point Error (EPE) between the produced flows. Hence, the last component of our loss function is:

$$\mathcal{L}_{V_n} = ||F(G_1(x_n), G_1(x_{n-1})) - F(y_n, y_{n-1})||_2, \quad (5)$$

where x and y are input and ground truth depth images respectively and n the time step. While we utilize ground truth depth as inputs to the optical flow network, colour images can also be equally viable inputs. However, since our training data contains noisy environmental elements (*e.g.*, lighting variations, rain, etc.), using the sharp and clean depth images leads to more desirable results.

Within the final decoder used exclusively for depth prediction, outputs are produced at four scales, following [29]. Each scale output is twice the spatial resolution of its previous scale. The overall depth loss is therefore the sum of losses calculated at every scale c:

$$\mathcal{L}_{depth} = \sum_{c=1}^{4} (\lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_s\mathcal{L}_s + \lambda_V\mathcal{L}_{V_n}). \quad (6)$$

The weighting coefficients (λ) are empirically selected (Section 3.4). These loss components, used to optimize depth fidelity, are used alongside the semantic segmentation loss, explained in Section 3.3.

Method	IoU	Method	IoU
SegNet-Basic [10]	46.4	DeconvNet [59]	48.9
SegNet [10]	50.2	Bayesian SegNet-Basic [40]	55.8
Reseg [75]	58.8	Our Approach	**59.1**

Table 4: Segmentation on the CamVid [14] test set.

3.3. Semantic Segmentation

As semantic segmentation is not the primary focus of our approach, but only used to enforce deeper and better representation learning within our model, we opt for a simple and efficient fully-supervised training procedure for our segmentation (G_2). The RGB or RGB-D image is used as the input and the network outputs class labels. Pixel-wise softmax with cross-entropy is used as the loss function, with the loss summed over all the pixels within a batch:

$$P_k(x) = \frac{e^{a_k(x)}}{\sum_{k'=1}^{K} e^{a_{k'}(x)}}, \quad (7)$$

$$\mathcal{L}_{seg} = -log(P_l(G_2(x))), \quad (8)$$

where $G_2(x)$ denotes the network output for the segmentation task, $a_k(x)$ is the feature activation for channel k, K is the number of classes, $P_k(x)$ is the approximated maximum function and l is the ground truth label for image pixels. The loss is summed for all pixels within the images.

Finally, since the entire network is trained as one unit, the joint loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{depth} + \lambda_{rec}\mathcal{L}_{seg}. \quad (9)$$

with coefficients selected empirically (Section 3.4).

3.4. Implementation Details

Synthetic data [67] consisting of RGB, depth and class labels are used for training. The discriminator follows the architecture of [64], and the optical flow network [65] is pre-trained on the KITTI dataset [56]. Experiments with the Sintel dataset [15] returned similar, albeit slightly inferior, results. The discriminator uses convolution-BatchNorm-leaky ReLU ($slope = 0.2$) modules. The dataset [67] contains numerous sequences some spanning thousands of frames. However, a feedback network taking in high-resolution images (512×128) back-propagating over thousands of time steps is intractable to train. Empirically, we found training over sequences of 10 frames offers a reasonable trade-off between accuracy and training efficiency. Mini-batches are loaded in as tensors containing two sequences of 10 frames each, resulting in roughly $10,000$ batches overall. All implementation is done in *PyTorch* [61], with Adam [42] providing the best optimization ($\beta_1 =$

Figure 6: Results of our approach applied to KITTI [2, 56]. **RGB:** input colour image; **GTD:** Ground Truth Depth; **MDE:** Monocular Depth Estimation; **GTS:** Ground Truth Segmentation; **GS:** Generated Segmentation.

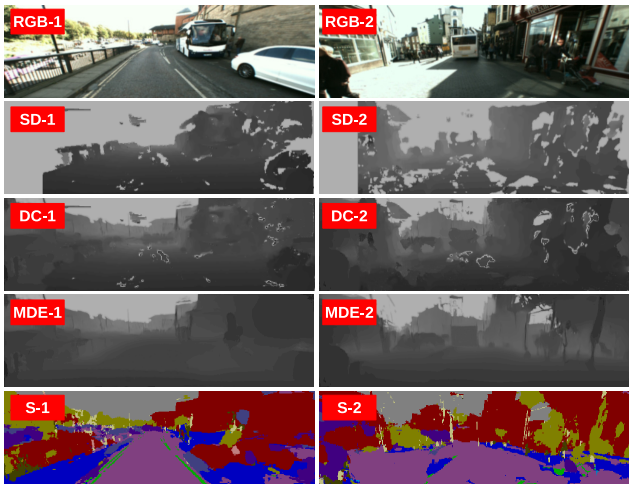

Figure 7: Our results on locally captured data. **SD:** Depth via Stereo Correspondence; **DC:** Depth Completion; **MDE:** Monocular Depth Estimation; **S:** Semantic Segmentation.

0.5, $\beta_2 = 0.999$, $\alpha = 0.0002$). The weighting coefficients in the loss function are empirically chosen to be $\lambda_{rec} = 1000, \lambda_{adv} = 100, \lambda_s = 10, \lambda_V = 1, \lambda_{seg} = 10$.

4. Experimental Results

We assess our approach using ablation studies and both qualitative and quantitative comparisons with state-of-the-art methods applied to publicly available datasets [2, 14, 20, 28, 56]. We also utilize our own synthetic test set and data captured locally to further evaluate the approach.

4.1. Ablation Studies

A crucial part of our work is demonstrating that every component of the approach is integral to the overall performance. We train our model to perform two tasks based on the assumption that the network is forced to learn more about the scene if different objectives are to be accomplished. We demonstrate this by training one model performing both tasks and two separate models focusing on each and conducting tests on randomly selected synthetic sequences [67]. As seen in Table 1, both tasks (monocular depth estimation and semantic segmentation) perform better when the model is trained on both. Moreover, since the segmentation pipeline does not receive any explicit temporal supervision (from the optical flow network) and its temporal continuity is only enforced by the input and middle streams trained by the depth pipeline, when the two pipelines are disentangled, the segmentation results become far worse than the depth results (Table 1).

Method	PSNR	SSIM	Method	PSNR	SSIM
Holes	33.73	0.372	GTS [36]	31.47	0.672
ICA [82]	31.01	0.488	GIF [50]	44.57	0.972
FDF [9]	46.13	0.986	Ours	**47.45**	**0.991**

Table 5: Structural integrity analysis post depth completion.

Figure 3 depicts the quality of the outputs when the model is a feedback network trained temporally compared to our model when the output depth from the previous time step is not used as the input during training. We can clearly see that both depth and segmentation results are of higher fidelity when temporal information is used during training.

Additionally, our depth prediction pipeline uses several loss functions. We employ the same test sequences to evaluate our model trained as different components are removed. Table 2 demonstrates the network temporally trained with all the loss components (T/R/A/SC/S/OF) outperforms models trained without specific ones. Qualitatively, we can see in Figure 4 that the results are far better when the network is fully trained with all the components. Specifically, the set of skip connections used in the network make a significant difference in the quality of the outputs.

4.2. Semantic Segmentation

Segmentation is not the focus of this work and is mainly used to boost the performance of depth prediction. However, we extensively evaluate our segmentation pipeline which outperforms several well-known comparators. We utilize Cityscapes [20] and CamVid [14] test sets for our performance evaluation despite the fact that our model is solely trained on synthetic data and *without any domain adaptation* should not be expected to perform well on naturally sensed real-world data. The effective performance of our segmentation points to the generalization capabilities of our model. When tested on CamVid [14], our approach produces better results compared to well-established techniques such as [10, 40, 59, 75] despite the lower quality

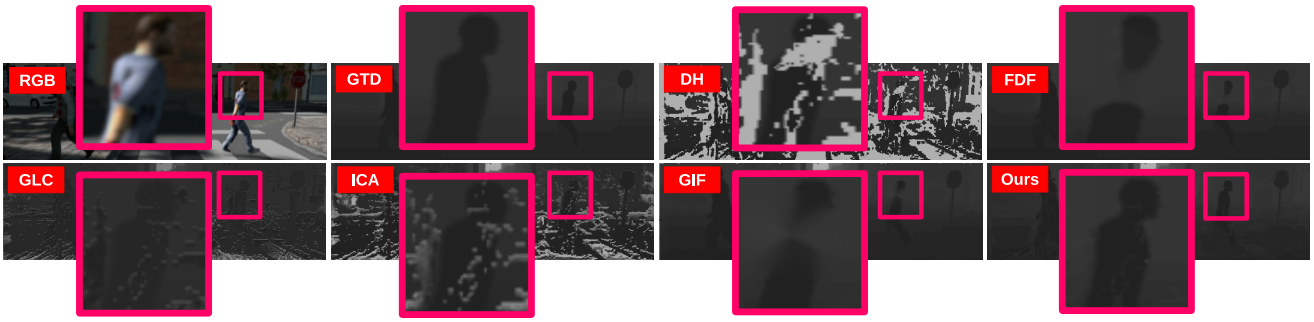

Figure 8: Comparison of various completion methods applied to the synthetic test set. **RGB:** input colour image; **GTD:** Ground Truth Depth; **DH:** Depth Holes; **FDF:** Fourier based Depth Filling [9]; **GLC:** Global and Local Completion [36]; **ICA:** Inpainting with Contextual Attention [82]; **GIF:** Guided Inpainting and Filtering [50].

Method		Error Metrics (lower, better)				Accuracy Metrics (higher, better)		
		Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$
Train Set Mean	[28]	0.403	0.530	8.709	0.403	0.593	0.776	0.878
Eigen *et al.*	[25]	0.203	1.548	6.307	0.282	0.702	0.890	0.958
Liu *et al.*	[49]	0.202	1.614	6.523	0.275	0.678	0.895	0.965
Zhou *et al.*	[87]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Godard *et al.*	[29]	0.148	**1.344**	5.927	0.247	0.803	0.922	0.964
Zhan *et al.*	[83]	**0.144**	1.391	**5.869**	0.241	0.803	0.928	**0.969**
Our Approach		0.193	1.438	5.887	**0.234**	**0.836**	**0.930**	0.958

Table 6: Numerical comparison of monocular depth estimation over the KITTI [28] data split in [25]. All comparators are trained and tested on the same dataset (KITTI [28]) while our approach is trained on [67] and tested using [28].

of the input images as seen in Table 4. As for Cityscapes [20], the test set does not contain video sequences, but our temporal model still outperforms approaches such as [17, 52, 53, 74, 86], as demonstrated in Table 3.

Examples of the segmentation results over both datasets are seen in Figure 5. Additionally, we also use the KITTI semantic segmentation data [2] in our tests and as shown in Figure 6, our approach produces high fidelity semantic class labels despite including *no domain adaptation*.

4.3. Depth Completion

Evaluation for depth completion ideally requires dense ground truth scene depth. However, no such dataset exists for urban driving scenarios, which is why we utilize randomly selected previously unseen synthetic data with available dense depth images to assess the results. Our model generates full scene depth and the predicted depth values for the missing regions of the depth image are subsequently blended in with the known regions of the image using [63]. Figure 8 shows a comparison of our results against other contemporary approaches [9, 36, 50, 82]. As seen from the enlarged sections, our approach produces minimal artefacts (blurring, streaking, etc.) compared to the other techniques. To evaluate the structural integrity of the results post completion, we also numerically assess the performance of our approach and the comparators. As seen in Table 5, our approach quantitatively outperforms the comparators as well.

While blending [63] might work well for colour images with a connected missing region, significant quantities of small and large holes in depth images can lead to undesirable artefacts such as stitch mark or burning effects post blending. Examples of artefacts can be seen in Figure 7, which demonstrates the results of the approach applied to locally captured data. This is further discussed in Section 5.

4.4. Monocular Depth Estimation

As the main focus of our model, our monocular depth estimation model is evaluated against contemporary state-of-the-art approaches [8, 25, 29, 49, 83, 87]. Following the conventions of the literature, we use the data split suggested in [25] as the test set. These images are selected from random sequences and do not follow a temporally sequential pattern, while our full approach requires video sequences as its input. As a result, we apply our approach to all the sequences from which the images are chosen but the evaluation itself is only performed on the 697 test images.

For numerical assessment, the generated depth is corrected for the differences in focal length between the training [67] and testing data [28]. As seen in Table 6, our approach outperforms [25, 49, 87] across all metrics and stays

Figure 9: Comparing the results of the approach against [87, 29, 44, 8]. Images have been adjusted for better visualization. **RGB:** input colour image; **GTD:** Ground Truth Depth; **DEV:** Depth and Ego-motion from Video [87]; **LRC:** Left-Right Consistency [29]; **SSE:** Semi-supervised Estimation [44]; **EST:** Estimation via Style Transfer [8]; **GS:** Generated Segmentation.

competitive with [29, 83]. It is important to note that all of these comparators are trained on the *same* dataset as the one used for testing [28] while our approach is trained on synthetic data [67] *without domain adaptation* and has not seen a single image from [28]. Additionally, none of the other comparators is capable of producing temporally consistent outputs as all of them operate on a frame level. As this cannot be readily illustrated via still images within Figures 8 and 9, we kindly invite the reader to view the supplementary **video** material accompanying the paper.

We also assess our model using the data split of KITTI [56] and qualitatively evaluate the results, since the ground truth images in [56] are of higher quality than the laser data and provide CAD models as replacements for the cars in the scene. As shown in Figure 6, our method produces sharp and crisp depth outputs with segmentation results in which object boundaries and thin structures are well preserved.

5. Limitations and Future Work

Even though our approach can generate temporally consistent depth and segmentation by utilizing a feedback network, this can lead to error propagation, *i.e.*, when an erroneous output is generated at one time step, the invalid values will continually propagate to future frames. This can be resolved by exploring the use of 3D convolutions or regularization terms aimed at penalizing propagated invalid outputs. Moreover, as mentioned in Section 4.3, blending the depth output into the known regions of the depth [63] produces undesirable artefacts in the results. This can be rectified by incorporating the blending operation into the training procedure. In other words, the blending itself will take place before the supervisory signal is back-propagated through the network during training, which would force the network to learn these artefacts, removing any need for post-processing. As for our segmentation component, no explicit temporal consistency enforcement or class balancing is performed, which has lead to frame-to-frame flickering and lower accuracy with unbalanced classes (*e.g.*, pedestrians, cyclists). By improving segmentation, the entire model can benefit from a performance boost. Most of all, the use of domain adaptation [8, 35] can significantly improve all results since despite its generalization capabilities, the model is only trained on synthetic data and should not be expected to perform just as well on naturally-sensed real-world images.

6. Conclusion

We propose a multi-task model capable of performing depth prediction and semantic segmentation in a temporally consistent manner using a feedback network that takes as its recurrent input the output generated at the previous time step. Using a series of dense skip connections, we ensure that no high-frequency spatial information is lost during feature down-sampling within the training process. We consider the task of depth prediction within the areas of depth completion and monocular depth estimation, and therefore train models based on both objectives within the depth prediction component. Using extensive experimentation, we demonstrate that our model achieves much better results when it performs depth prediction and segmentation at the same time compared to two separate networks performing the same tasks. The use of skip connections is also shown to be significantly effective in improving the results for both depth prediction and segmentation tasks. Although certain isolated issues remain, experimental evaluation demonstrates the efficacy of our approach compared to contemporary state-of-the-art methods tackling the same problem domains [17, 29, 36, 40, 53, 82, 83, 87].
*We kindly invite the readers to refer to the **video**: https://vimeo.com/325161805 for more information and larger improved-quality result images.*

References

[1] Austin Abrams, Christopher Hawley, and Robert Pless. Heliometric stereo: Shape from sun position. *Euro. Conf. Computer Vision*, pages 357–370, 2012.

[2] Hassan Alhaija, Siva Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *Int. J. Computer Vision*, 126(9):961–972, 2018.

[3] Pablo Arias, Gabriele Facciolo, Vicent Caselles, and Guillermo Sapiro. A variational framework for exemplar-based image inpainting. *Computer Vision*, 93(3):319–347, 2011.

[4] Amir Atapour-Abarghouei, Samet Akcay, Gregoire Payen de La Garanderie, and Toby Breckon. Generative adversarial framework for depth filling via wasserstein metric, cosine transform and domain transfer. *Pattern Recognition*, 91:232–244, 2019.

[5] Amir Atapour-Abarghouei and Toby Breckon. Depthcomp: Real-time depth image completion based on prior semantic scene segmentation. In *British Machine Vision Conference*, pages 1–13, 2017.

[6] Amir Atapour-Abarghouei and Toby Breckon. A comparative review of plausible hole filling strategies in the context of scene depth image completion. *Computers and Graphics*, 72:39–58, 2018.

[7] Amir Atapour-Abarghouei and Toby Breckon. Extended patch prioritization for depth filling within constrained exemplar-based RGB-D image completion. In *Int. Conf. Image Analysis and Recognition*, pages 306–314, 2018.

[8] Amir Atapour-Abarghouei and Toby Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–12, 2018.

[9] Amir Atapour-Abarghouei, Gregoire Payen de La Garanderie, and Toby Breckon. Back to butterworth - a Fourier basis for 3D surface relief hole filling within RGB-D imagery. In *Int. Conf. Pattern Recognition*, pages 2813–2818, 2016.

[10] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.

[11] Mohammad Haris Baig, Vignesh Jagadeesh, Robinson Piramuthu, Anurag Bhardwaj, Wei Di, and Neel Sundaresan. Im2Depth: Scalable exemplar based depth transfer. In *Winter Conf. Applications of Computer Vision*, pages 145–152, 2014.

[12] Marcelo Bertalmio, Andrea Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages I–I, 2001.

[13] Toby Breckon and Robert Fisher. A hierarchical extension to 3D non-parametric surface relief completion. *Pattern Recognition*, 45:172–185, 2012.

[14] Gabriel Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[15] Daniel Butler, Jonas Wulff, Garrett Stanley, and Michael Black. A naturalistic open source movie for optical flow evaluation. In *Euro. Conf. Computer Vision*, pages 611–625, 2012.

[16] P. Cavestany, A.L. Rodriguez, H. Martinez-Barbera, and T.P. Breckon. Improved 3D sparse maps for high-performance structure from motion with low-cost omnidirectional robots. In *Int. Conf. Image Processing*, pages 4927–4931, 2015.

[17] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.

[18] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Computer Vision and Pattern Recognition*, pages 3640–3649, 2016.

[19] Weihai Chen, Haosong Yue, Jianhua Wang, and Xingming Wu. An improved edge detection algorithm for depth map inpainting. *Optics and Lasers in Engineering*, 55:69–77, 2014.

[20] 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 *IEEE Conf. Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[21] Ding Ding, Sundaresh Ram, and Jeffrey Rodriguez. Perceptually aware image inpainting. *Pattern Recognition*, 83:174–184, 2018.

[22] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[23] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Int. Conf. Computer Vision*, pages 2758–2766, 2015.

[24] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Int. Conf. Computer Vision*, pages 2650–2658, 2015.

[25] David Eigen, Christian Puhrsch, and Rob 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.

[26] Mohsen Fayyaz, Mohammad Hajizadeh Saffar, Mohammad Sabokrou, Mahmood Fathy, Reinhard Klette, and Fay Huang. STFCN: Spatio-temporal FCN for semantic video segmentation. In *Asian Conf. Computer Vision Workshop*, pages 493–509, 2016.

[27] Raghudeep Gadde, Varun Jampani, and Peter V Gehler. Semantic video CNNs through representation warping. In *Int. Conf. Computer Vision*, pages 4463–4472, 2017.

[28] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *Robotics Research*, pages 1231–1237, 2013.

[29] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 6602 – 6611, 2017.

[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Int. Conf. Computer Vision*, pages 1026–1034, 2015.

[32] Philipp Heise, Sebastian Klose, Brian Jensen, and Alois Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Int. Conf. Computer Vision*, pages 2360–2367, 2013.

[33] Daniel Herrera, Juho Kannala, Janne Heikkilä, et al. Depth map inpainting under a second-order smoothness prior. In *Scandinavian Conf. Image Analysis*, pages 555–566, 2013.

[34] Heiko Hirschmuller. Stereo processing by semi-global matching and mutual information. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30:328–341, 2008.

[35] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In *Int. Conf. Machine Learning*, pages 1–13, 2018.

[36] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graphics*, 36(4):107, 2017.

[37] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Int. Conf. Machine Learning*, pages 1–9, 2015.

[38] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 5967–5976, 2017.

[39] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[40] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *British Machine Vision Conference*, pages 1–12, 2017.

[41] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–10, 2018.

[42] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learning Representations*, pages 1–15, 2014.

[43] Mandar Kulkarni and Ambasamudram Rajagopalan. Depth inpainting by tensor voting. *J. Optical Society of America A*, 30(6):1155–1165, 2013.

[44] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 6647–6655, 2017.

[45] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *Int. Conf. 3D Vision*, pages 239–248, 2016.

[46] Bo Li, Chunhua Shen, Yuchao Dai, Anton van den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical CRFs. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1119–1127, 2015.

[47] Yule Li, Jianping Shi, and Dahua Lin. Low-latency video semantic segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 5997–6005, 2018.

[48] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 5168–5177, 2017.

[49] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 38(10):2024–2039, 2016.

[50] Junyi Liu, Xiaojin Gong, and Jilin Liu. Guided inpainting and filtering for kinect depth maps. In *Int. Conf. Pattern Recognition*, pages 2055–2058, 2012.

[51] Miaomiao Liu, Xuming He, and Mathieu Salzmann. Building scene models by completing and hallucinating depth and semantics. In *Euro. Conf. Computer Vision*, pages 258–274, 2016.

[52] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Int. Conf. Computer Vision*, pages 1377–1385, 2015.

[53] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[54] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Trans. Geoscience and Remote Sensing*, 55(2):645–657, 2017.

[55] Kiyoshi Matsuo and Yoshimitsu Aoki. Depth image enhancement using local tangent plane approximations. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 3574–3583, 2015.

[56] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *Photogrammetry and Remote Sensing*, pages 60–76, 2018.

[57] Suryanarayana M Muddala, Marten Sjostrom, and Roger Olsson. Depth-based inpainting for disocclusion filling. In *3DTV Conf.: The True Vision-Capture, Transmission and Display of 3D Video*, pages 1–4, 2014.

[58] David Nilsson and Cristian Sminchisescu. Semantic video segmentation by gated recurrent flow propagation. In *IEEE*

Conf. Computer Vision and Pattern Recognition, pages 1–11, 2018.

[59] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Int. Conf. Computer Vision*, pages 1520–1528, 2015.

[60] Emin Orhan and Xaq Pitkow. Skip connections eliminate singularities. In *Int. Conf. Learning Representations*, pages 1–11, 2018.

[61] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Advances in Neural Information Processing Systems*, pages 1–4, 2017.

[62] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[63] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. In *Graphics*, volume 22, pages 313–318, 2003.

[64] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[65] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 2720–2729, 2017.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Int. Conf. Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.

[67] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 3234–3243, 2016.

[68] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 47:7–42, 2002.

[69] Evan Shelhamer, Kate Rakelly, Judy Hoffman, and Trevor Darrell. Clockwork ConvNets for video semantic segmentation. In *Euro. Conf. Computer Vision*, pages 852–868, 2016.

[70] Marijn F Stollenga, Wonmin Byeon, Marcus Liwicki, and Juergen Schmidhuber. Parallel multi-dimensional LSTM, with application to fast biomedical volumetric image segmentation. In *Advances in Neural Information Processing Systems*, pages 2998–3006, 2015.

[71] Michael W Tao, Pratul P Srinivasan, Jitendra Malik, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1940–1948, 2015.

[72] Alexandru Telea. An image inpainting technique based on the fast marching method. *Graphics Tools*, 9(1):23–34, 2004.

[73] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Int. Conf. Computer Vision*, pages 4809–4817, 2017.

[74] Jonas Uhrig, Marius Cordts, Uwe Franke, and Thomas Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *German Conf. Pattern Recognition*, pages 14–25, 2016.

[75] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops*, pages 41–48, 2016.

[76] Junyuan Xie, Ross Girshick, and Ali Farhadi. Deep3D: Fully automatic 2D-to-3D video conversion with deep convolutional neural networks. In *Euro. Conf. Computer Vision*, pages 842–857, 2016.

[77] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 6556–6565, 2018.

[78] Hongyang Xue, Shengming Zhang, and Deng Cai. Depth image inpainting: Improving low rank matrix completion with low gradient regularization. *IEEE Trans. Image Processing*, 26(9):4311–4320, 2017.

[79] Jin Yamanaka, Shigesumi Kuwashima, and Takio Kurita. Fast and accurate image super resolution by deep CNN with skip connection and network in network. In *Neural Information Processing*, pages 217–225, 2017.

[80] Raymond Yeh*, Chen Chen*, Teck Yian Lim, Schwing Alexander, Mark Hasegawa-Johnson, and Minh Do. Semantic image inpainting with deep generative models. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 6882–6890, 2017.

[81] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–10, 2018.

[82] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas Huang. Generative image inpainting with contextual attention. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–15, 2018.

[83] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 340–349, 2018.

[84] Yinda Zhang and Thomas Funkhouser. Deep depth completion of a single RGB-D image. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 175–185, 2018.

[85] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.

[86] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr. Conditional random fields as recurrent neural networks. In *Int. Conf. Computer Vision*, pages 1529–1537, 2015.

[87] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from

video. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 6612–6619, 2017.

[88] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Int. Conf. Computer Vision*, pages 408–417, 2017.

[89] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 4141–4150, 2017.

7. Appendix

In this section, we provide additional information that could not be placed within the main paper due to space restrictions. We kindly invite the readers to watch the **video** submitted as part of the supplementary material along with this document.

7.1. Hole Prediction Network

As mentioned in the main manuscript, when the model is being trained to perform depth completion, the input must be a four-channel RGB-D image, in which the depth channel contains holes that would naturally occur when sensed through imperfect capture technologies. However, the dataset used for training our model [67] consists of pixel-prefect depth images without any holes.

This synthetic dataset [67] does contain stereo image pairs, so a simple solution would be to calculate the disparity and subsequently the depth using a well-established stereo matching approach such as Semi-Global Matching [34] and use the resulting depth image (which will contain holes) as the input.

However, each image in a stereo pair in [67] (left and right) comes with its own corresponding (left and right) depth image, and half of the dataset (aligned RGB and depth images) will be rendered useless if stereo matching is used to calculate depth images with hole.

As a result, we opt for training an entirely separate model that would be responsible for creating holes in the depth images. Even though the details regarding the training or use of this network have no bearing on the approach proposed in the main manuscript, we will attempt to cover the inner workings and experimental evaluation of our *hole prediction* model here.

This *hole prediction* model is a fully convolutional encoder-decoder network inspired by [66] with skip connections between all corresponding layers in the encoder and the decoder. The last decoder layer is connected to a soft-max classifier. Each convolutional layer is followed by batch normalization [37] and a ReLU. The network architecture can be seen in Figure 10.

The training data for this *hole prediction* network is made up of $30,000$ pairs of stereo images from [28]. Disparity is calculated using Semi-Global Matching (SGM) [34] and a hole mask (M) is subsequently generated which indicates which pixels are holes. Although SGM is used here, this is interchangeable with any other passive or active depth capture approach. The left RGB images are thus used as inputs with the generated masks as ground truth labels. Binary cross-entropy is used as the loss function since the segmentation task involves only two classes: hole and non-hole.

Qualitative analyses reveal that holes are predicted where expected. From Figure 11, we see that in regions where camera overlap is absent or featureless surfaces, sparse shrubbery, unclear object boundaries, and very distant objects are present, such pixels are correctly classified as holes.

Method	Error				Accuracy
	Abs. Rel.	Sq. Rel.	RMSE	RMSE log	$\sigma < 1.25^3$
[87]	0.401	1.601	6.598	0.363	0.788
[29]	0.334	1.556	6.304	0.302	0.852
Ours (full)	**0.208**	**1.402**	**6.026**	**0.269**	**0.926**

Table 7: Comparisons using synthetic data [67].

7.2. Additional Experiments

Following the conventions of the expansive literature on monocular depth estimation, we measure the performance of our approach against the KITTI dataset [28]. However, we have re-trained and tested all the comparators using the synthetic dataset of [67] but for brevity and due to our superior performance on the *unseen* KITTI dataset, against comparators *actually* trained on KITTI, we have not included these extra results in the main manuscript. Table 7 presents the comparison of our approach against [29, 87] trained on the synthetic dataset of [67] under the exact same conditions as outlined in Section 3.4 of the main manuscript. Our approach outperforms the comparators by a large margin (Table 7).

7.3. Figures

Due to the space restrictions, some of the figures within the main paper may be too small for appropriate viewing. While some of the results are better seen in the accompanying video, we also provide enlarged versions of some of the figures here. Please see Figures 12, 13, 14, 15, 16 and 17 in the appendix.
Video URL: https://vimeo.com/325161805

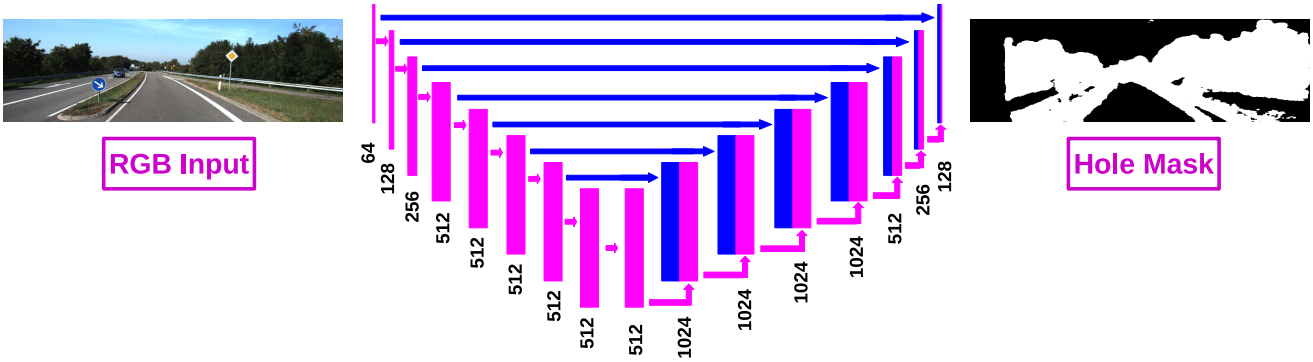

Figure 10: Overview of the architecture of the *hole prediction* network.

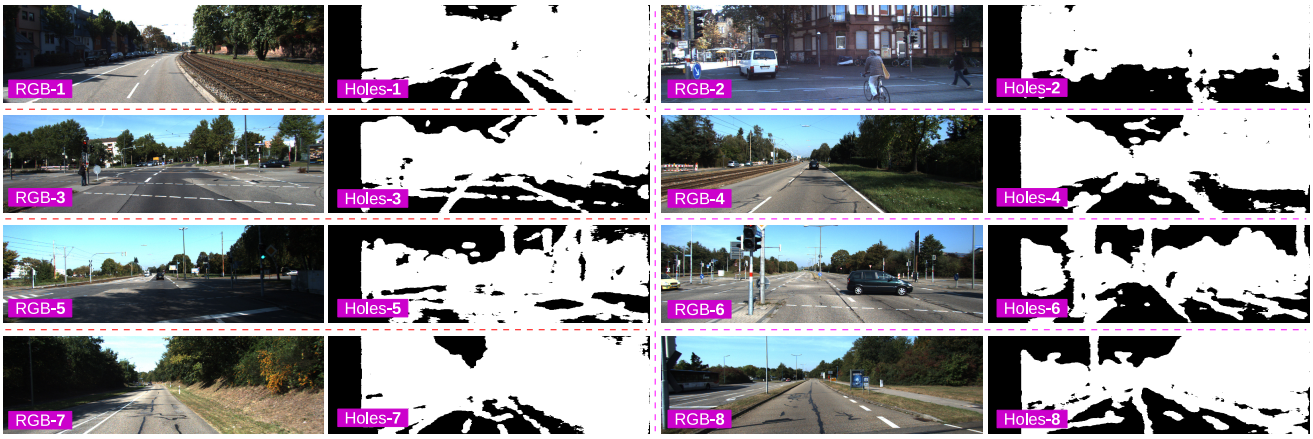

Figure 11: Examples of results of the *hole prediction* model applied to unseen images from [28].

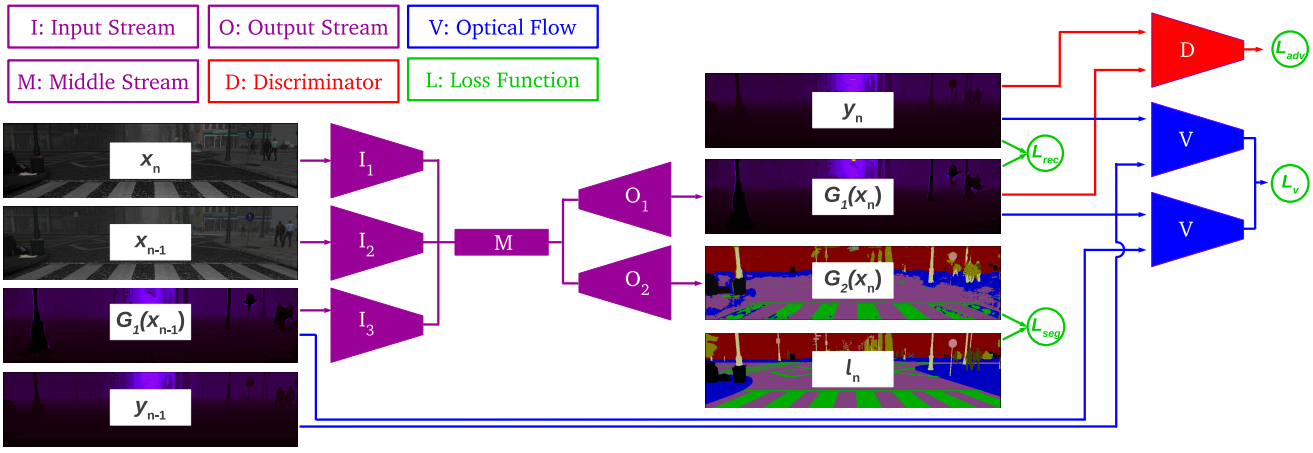

Figure 12: An outline of the training procedure of the main proposed approach.

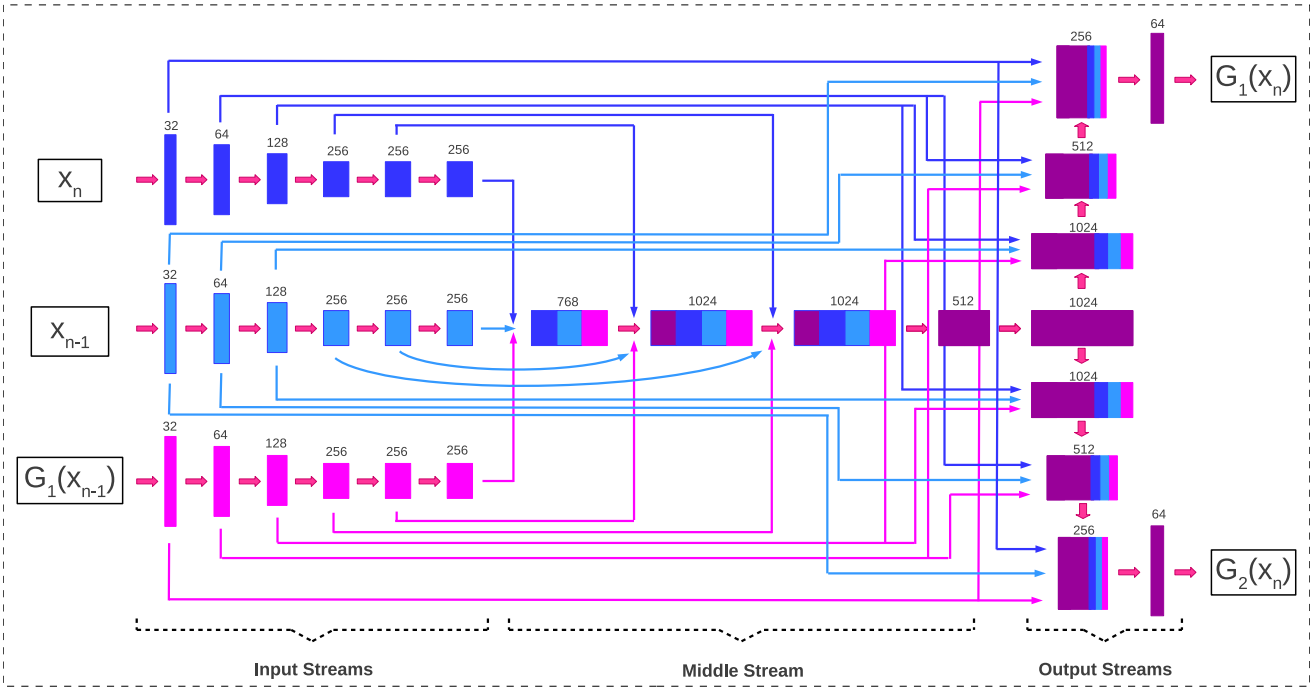

Figure 13: An overview of the generator architecture.

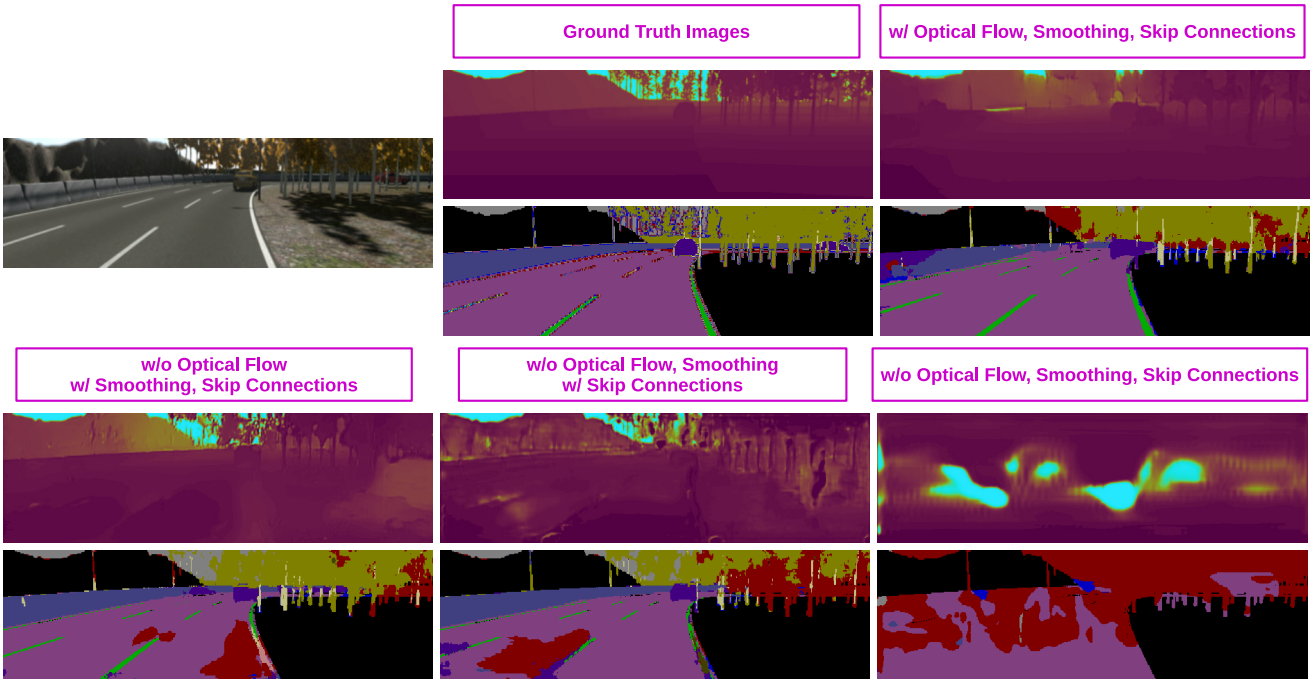

Figure 14: Comparing the results of our model (with monocular depth estimation) when different components of the approach are removed.

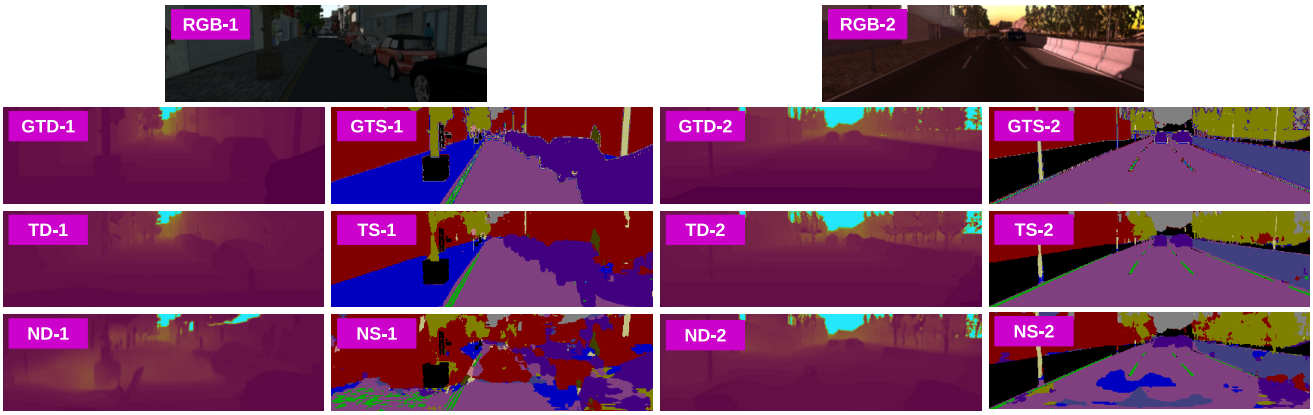

Figure 15: Comparing the results of the approach on the synthetic test set when the model is trained with and without temporal consistency. **RGB:** input colour image; **GTD:** Ground Truth Depth; **GTS:** Ground Truth Segmentation; **TS:** Temporal Segmentation; **TD:** Temporal Depth; **NS:** Non-Temporal Segmentation; **ND:** Non-Temporal Depth.

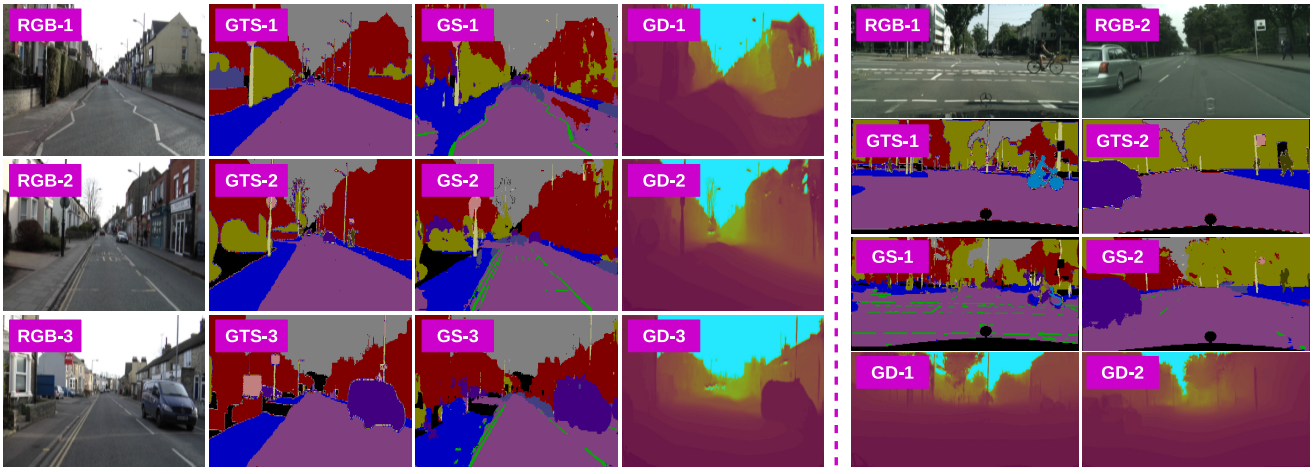

Figure 16: Results of our approach on CamVid [14] (left) and Cityscapes [20] (right) datasets. **RGB:** input colour image; **GTS:** Ground Truth Segmentation; **GS:** Generated Segmentation; **GD:** Generated Depth.

Figure 17: Comparison of depth completion methods applied to synthetic test set. **RGB:** input colour image; **GTD:** Ground Truth Depth; **DH:** Depth Holes; **FDF:** Fourier based Depth Filling [9]; **GTS:** Global and Local Completion [36]; **ICA:** Inpainting with Contextual Attention [82]; **GIF:** Guided Inpainting and Filtering [50].