High-Resolution Synthetic RGB-D Datasets for Monocular Depth Estimation

Aakash Rajpal[1,2], Noshaba Cheema[2,3,4], Klaus Illgner-Fehns[1],
Philipp Slusallek[2,4], and Sunil Jaiswal[1,*]

[1] K|Lens GmbH, Germany, [2] Saarland Informatics Campus, Germany,
[3] MPI Informatics, Germany, [4] German Research Center for Artificial Intelligence (DFKI), Germany

Abstract

Accurate depth maps are essential in various applications, such as autonomous driving, scene reconstruction, point-cloud creation, etc. However, monocular-depth estimation (MDE) algorithms often fail to provide enough texture & sharpness, and also are inconsistent for homogeneous scenes. These algorithms mostly use CNN or vision transformer-based architectures requiring large datasets for supervised training. But, MDE algorithms trained on available depth datasets do not generalize well and hence fail to perform accurately in diverse real-world scenes. Moreover, the ground-truth depth maps are either lower resolution or sparse leading to relatively inconsistent depth maps. In general, acquiring a high-resolution ground truth dataset with pixel-level precision for accurate depth prediction is an expensive, and time-consuming challenge.

In this paper, we generate a high-resolution synthetic depth dataset (HRSD) of dimension 1920×1080 *from Grand Theft Auto (GTA-V), which contains 100,000 color images and corresponding dense ground truth depth maps. The generated datasets are diverse and have scenes from indoors to outdoors, from homogeneous surfaces to textures. For experiments and analysis, we train the DPT algorithm, a state-of-the-art transformer-based MDE algorithm on the proposed synthetic dataset, which significantly increases the accuracy of depth maps on different scenes by 9%. Since the synthetic datasets are of higher resolution, we propose adding a feature extraction module in the transformer's encoder and incorporating an attention-based loss, further improving the accuracy by 15 %.*

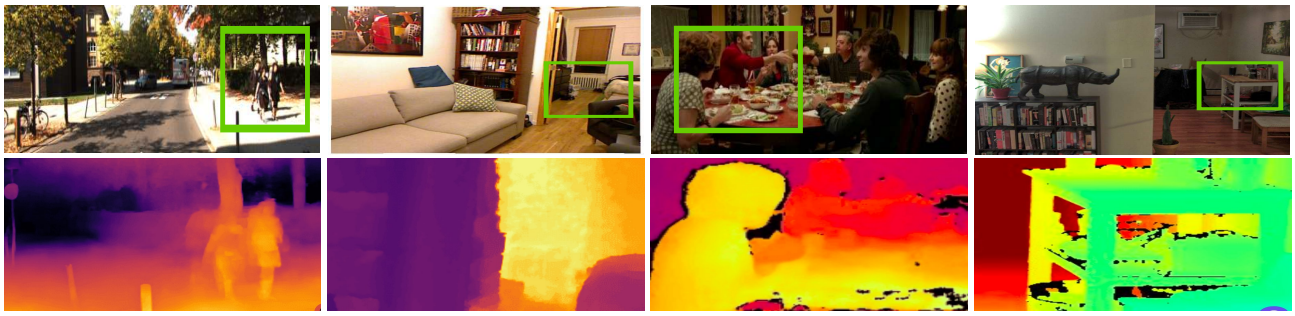
1. Introduction

Artificial intelligence's success in several computer vision applications has recently led to the low cost, small size, and wide applications of monocular cameras. Monocular depth estimation (MDE) algorithms are mainly based on neural networks and have shown great abilities in estimating depth from a single image [4, 18, 21, 30, 35, 52]. These algorithms require leveraging high-level scene priors [42], so training a deep neural network with supervised data becomes the defacto solution.

MDE algorithms using convolutional neural networks (CNN) deployed in an encoder-decoder structure learn a depth map with a similar spatial resolution to an RGB image. The encoder learns the feature representations from the input and provides a low-level output to the decoder. Using these features, the decoder first aggregates them and learns the final predictions. While CNN's have been the preferred architecture in computer vision, transformers have also recently gained traction motivated by their success in natural language processing [49]. Transformer-based encoder structures have significantly contributed to many vision-related problems, such as image segmentation [54], optical

Figure 1. Improvement over state-of-the-art DPT [39]. Ours DPT-B and DPT-B+R+Al are two variants of DPT [39] trained on the proposed HRSD datasts. The green rectangle represents area of the image where DPT fails to give precise and consistent depth compared to our algorithm.

This work was partially funded by the German Ministry for Education and Research (BMBF) under the grant PLIMASC.
*Corresponding author: sunil.jaiswal@k-lens.de



(a) KITTI RGB-D Dataset [22] (b) NYU V2 Dataset [44] (c) MIDAS RGB-D [40] (d) DioDE Dataset [48]

Figure 2. Example frames from publicly available datasets used for MDE. We zoom-in the problematic areas in the depth map.

flow estimation [28], and image restoration [32]. In contrast with CNN's that progressively down-sample the input image and lose feature resolution across the layers, vision transformer (ViT) [17] processes feature maps at a constant resolution with a global receptive field at every stage. Feature resolution and granularity are important for dense depth estimation, and an ideal architecture should resolve features at or close to the resolution of the input RGB image.

Generally, an MDE algorithm based on CNN or transformer requires large RGB-D datasets consisting of diverse scenes with precise ground truth depth maps. However, the publicly available depth datasets mostly consist of ground-truth depth maps, either lower resolution (e.g., NYU V2) [44] or sparse (e.g., KITTI RGB-D) [22], or have been estimated from multiview depth estimation algorithms MiDaS [40]. Figure 2 highlights some ground-truth depth maps from these publicly available datasets. This is one of the reasons why the existing MDE algorithms lack fine-grained details, as shown in Figure 1, and fail to perform accurately in diverse scenes. In this paper, we generate a high-resolution synthetic dataset using a commercial video game, Grand Theft Auto (GTA-V) [1]. The dataset contains around 100,000 pairs of color images with precise dense depth maps of resolution 1920×1080, and we refer to this dataset as a High-Resolution Synthetic Depth (HRSD) dataset for monocular depth estimation. To show the effectiveness of the proposed HRSD datasets, we re-train the state-of-the-art transformer-based MDE algorithm: DPT [39] with different experiments explained in section 4.1. More specifically, we fine-tune DPT on the proposed HRSD datasets for dense depth prediction on high-resolution images, demonstrate the performance on other public datasets, and compare them with the SOTA algorithms. Further, to exploit the HRSD dataset, we propose adding a feature-extraction module with an attention-loss that further helps improve the results.

In summary, we introduce the following concepts:

- We have generated a high-resolution synthetic dataset (HRSD) from the game GTA-V [1] with precise ground truth depth values. The HRSD dataset consists of diversified images enabling MDE networks to train on varied scenes, leading to a good generalization of real-world data.
- We propose to add a feature extraction module that processes the color image and converts them into feature maps to use them as patches for both the ViT [17] and DPT algorithms [39]. In addition, we optimize the training procedure by using an attention-based loss instead of shift-invariant loss [23, 39], improving performance in terms of efficiency and accuracy, resulting in smooth, consistent depth maps for high-resolution images.

We conduct experiments on standard public datasets with different input image sizes, such as NYU V2 [44] (640×480), KITTI [22] (1216×352), and our HRSD (1920×1080). We compare the performance with DPT [39], the state-of-the-art MDE algorithm, and Multi-res, [37] a depth estimation network built specifically for high-resolution images. We observe that the depth maps produced after training on the HRSD dataset outperformed existing algorithms on different public datasets.

2. Related Work

RGB-D Dataset Various datasets have been proposed that are suitable for monocular depth estimation, i.e., they consist of RGB images with corresponding depth annotation of some form. These datasets differ in captured environments and objects, type of depth annotation (sparse/dense, absolute/relative depth), accuracy (laser, stereo, synthetic data), image resolution, camera settings, and dataset size. Earlier RGB-D datasets have relied on either Kinect [15, 27, 44, 45] or LIDAR [6, 22] or stereo vision [43] for depth annotation. Existing Kinect-based datasets are limited to indoor scenes; existing LIDAR-based datasets are biased towards scenes of man-made structures and have a low spatial resolution. Every dataset comes with its characteristics and has its own biases and problems [46].

Ranftl et al. [40] introduced a vast data source from 3D films capturing high-quality frames from movies to get a diversity of dynamic environments for depth estimation. They captured 80,000 frames at 1920×1080 resolution while using stereo matching to extract relative depth. However, stereo matching has its downside, as the depth maps are not precise and fail to perform on homogeneous scenes [53]. DIODE [48], another high-resolution dataset generated using a laser sensor for dense depth annotation. It contained 25,000 indoor & outdoor RGB images at a 768×1024, but it also led to inconsistent depth maps with artifacts in background objects and textured scenes, as shown in Figure 2. Synthetic dataset generation from computer games [20, 25, 41] has been used extensively for training computer vision algorithms like semantic segmentation [9, 38], object detection [29], and depth estimation [23, 50]. Mohammad et al. [23] have also used the GTA-V game to generate a synthetic RGB-D dataset but with relative depth at the focus. The resolution of the dataset used for training is 256×256 which is much smaller than the publicly available datasets. Furthermore, they need a preprocessing phase, such as histogram equalization, to use the datasets before feeding them to the training. For training, they use a Resnet architecture and process the RGB image and GT depth with resolution 256×256.

Depth from Single Image Convolutional networks within an encoder-decoder paradigm [4] is the standard prototype architecture for dense depth prediction from a single image. The building blocks of such a network consist of convolutional and sub-sampling as their core elements. However, CNN as an encoder suffers from a local receptive field problem [2], leading to less global representation learning at higher resolutions. Several algorithms adapt different techniques to learn features at different resolutions to address this issue like dilated convolutions [14, 34] or parallel multi-scale feature aggression [26, 31].

Recently, transformer architectures such as vision transformer (ViT) [17] or data-efficient image transformers (DeiT) [47] have outperformed CNN architectures in image recognition [17, 47], object detection [10], and semantic segmentation [54]. Inspired by the success of transformers in various topics, René et al. [39] use a vision transformer for dense depth prediction and have outperformed all the existing MDE algorithms. Nonetheless, all transformer architectures are data-hungry networks [8, 17] and thus require a huge dataset.

3. Proposed Method

In this section, we introduce our high-resolution synthetic dataset (HRSD) and then discuss the proposed architecture changes to ViT [17] and DPT [39] algorithms to provide consistent and accurate dense depth maps on high-resolution images.

3.1. RGB-D Dataset

Acquiring an accurate ground truth depth dataset for high-resolution images is challenging and expensive. Most RGB-D datasets have low image quality and sparse or relatively inaccurate depth maps. Inspired by the success of synthetic data in different computer vision applications [9, 23, 29, 38, 50], we propose to generate a synthetic high-resolution RGB-D dataset for monocular depth estimation. The advantage of having a synthetic RGB-D dataset is to have precise ground truth depth maps for diverse color images. Also, one can control the lighting environments, objects, and background and the datasets can be as large depending on the applications.

We use the Game Grand-Theft-Auto-V (GTA-V) [1] to generate a high-resolution synthetic RGB-D dataset. On a high level, we use the GTA-V game's in-build model and mechanics to calculate the ground truth (GT) depth and store them along with the high-resolution RGB image. Note that a similar idea of GTA-V-based synthetic data generation is designed for semantic segmentation [9, 41] and depth estimation [23, 38, 50]. Grand Theft Auto (GTA) [1], one of the prominent interactive games, consists of many diversified environments, including people, vehicles, recreational areas, and architecture. The precision of graphics and 3D rendered models in this game is exceptional, making it a favorable alternative to acquiring demanding large-scale real-world datasets such as RGB-D.

Real-time rendering To explain the data collection process, we must first review deferred shading, an important aspect of modern video games real-time rendering pipeline. Deferred shading utilizes geometric resources to produce depth and normal image buffers by communicating to the GPU [3]. The intermediate outputs of the rendering pipeline are collected in buffers called G-buffers, which are then illuminated rather than the original scene. Rendering is accelerated significantly due to decoupled geometry processing, reflectance properties, and illumination. To utilize the rendering pipeline, we identify how the game communicates with the graphics hardware to extract the different types of resources, i.e., texture maps and 3D meshes. These resources are combined for the final scene composition. APIs such as OpenGL, Direct3D, or DirectX, provided via dynamically loaded libraries, are used for inter-communication. Video Games load these libraries into their application memory and use a wrapper to the specific library to initiate the communication and record all commands.

G2D and Scripthook V Using a DirectX driver, we communicate with GTA-V and redirect all rendering commands to the driver for extracting depth maps. To obtain the necessary image datasets under varying conditions from GTA-V, we use an image simulator software G2D: from

Figure 3. Examples from HRSD dataset consisting of indoor & outdoor scenes with diversified objects and environments.

GTA to Data [16]. G2D allows to manipulate the environment on the fly, by injecting a customized mod in the game that controls global lighting and weather conditions by timing sunrises and sunsets. Using an accelerated time frame, we can create multiple high-resolution images from a single scene during sunrise, midday, sunset, and midnight. At the same time, automatic screen capture is enforced to record each frame displayed within the game, and the depth information is extracted. This enables us to accumulate a massive dataset as shown in Figure 3, comprising diverse environmental conditions (time of day, weather, season, etc.) and resulting in training computer vision algorithms that are robust and reliable in the real world.

Using the above strategy, we can generate as many images, and in this paper, we synthetically generate $100,000$ images with resolution 1920×1080 along with depth maps. The minimum and maximum depths are set to 0.1 m and 10 m, respectively, for indoor scenes, and the max depth for outdoor scenes is 50 m. We choose a small depth range to enable efficient training of our modified transformer network using L_1 loss instead of the scale-invariant loss [19] used in most MDE networks [7, 23, 39, 40]. We then use this dataset to train ViT [17] & the DPT [39] algorithm for dense depth-map prediction.

3.2. Architecture

The state-of-the-art DPT [39] algorithm uses ViT [17] as a backbone encoder and then adds a decoder to get a depth map of the same resolution as a color image. Here, we discuss our modification to the DPT [39] architecture. More specifically, we add a feature extractor module (pretrained Resnet [24]) in the DPT [39] architecture and a loss function consisting of attention supervision and L_1 loss to efficiently train the algorithm. The important blocks of the architecture are described below, and an overview of the network is depicted in Figure 4.

Feature Extractor Module Previous works [13, 17, 54] split the input RGB Image $I \in \mathbb{R}^{H \times W \times 3}$ into equal size

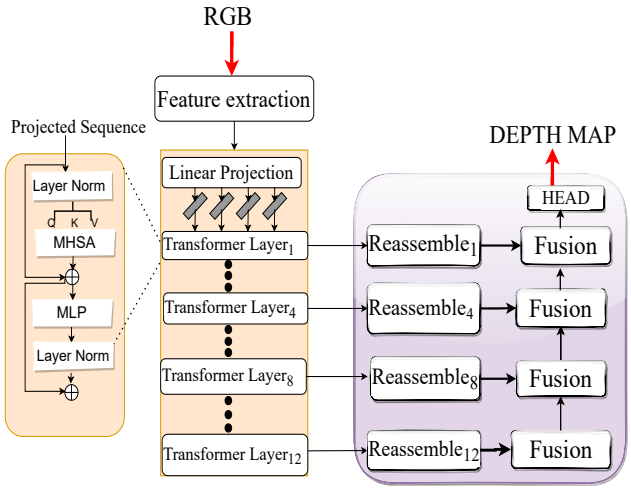


Figure 4. Overview of our proposed changes to DPT [39] architecture. We introduce a feature extraction module.

2D patches and used linear projection to convert them into tokens for ViT. These tokens have a one-to-one correspondence with image patches and thus grow as the size of the input image increases. For high-resolution images, this creates a bottleneck for the ViT, given as the number of tokens increases, ViT's computation performance increases leading to higher inference time. As shown in Figure 4, we replaced the color-image patches in ViT [17] (or DPT's encoder) with a feature extraction module similar to DPT-Hybrid [39]. In this paper, we use pretrained ResNet [24] for our feature extraction module and a detailed analysis is given in Table 4.

The input sequence to the ViT now comes from a ResNet backbone [24]. This feature module (ResNet-50 [24]) converts the input image into patches of feature maps. We use the final layer feature map, which gives a pre-defined dimensional representation of any image size. To match the input dimension of the transformer with the output of the

final ResNet-50 layer, we flatten the spatial dimension of these feature maps and project them to the transformer's dimension. So the vectorized patches from the output of Resnet are projected into a latent embedded sequence, the input to the first transformer layer, as shown in Figure 4. The input tokens are then processed by L transformer layers consisting of multi-headed self-attention (MHSA) and multi-layer perceptron (MLP) blocks.

Vision Transformer Encoder The spatial resolution of the initial embedding is maintained throughout the ViT encoder enabling a global receptive field at every transformer layer. This, along with MHSA being an inherently global operation, helps to achieve higher performance on high-resolution images. We have experimented with different values of L, with 12 transformer layers providing consistent dense depth maps similar to DPT-Base [39]. At each layer L, the input of MHSA is a triplet of Q (query), K (key), and V (Value) computed as follows. [17, 49]

$$Q = z_{\ell-1} \times W_Q, K = z_{\ell-1} \times W_K, V = z_{\ell-1} \times W_V \quad (1)$$

Here, z_{l-1} is the output from the previous transformer layer, with z_0 being the input to the first layer. Then, the attention head (AH) is calculated using the triplet Q, K, and V as given in [17, 49]. Later, all the AH's are combined with a weight matrix calculated during training for multi-head attention. The MHSA output is then fed to the MLP block, which calculates the final output.

Convolutional Decoder. Our decoder resembles the one used in DPT [39] as it assembles the set of tokens from different transformer layers at various resolutions. An image-like representation is recovered from the output of the encoder using a three-function calculation called the reassemble operation [39]. We use four reassemble blocks which extract tokens from the 1^{st}, 4^{th} , 8^{th} , 12^{th} (final) transformer layers. Transformer networks need more channels than their convolutional counterparts, so we double the number of channels in the three last reassemble modules. Each stage in the decoder layer, known as fusion blocks, is based on refinement [33]. They progressively combine the feature maps from consecutive stages into the final dense prediction. Unlike in the DPT decoder, batch normalization is helpful for dense depth prediction. We also reduce the number of channels in the fusion block to 96 from 256 in DPT [39] to enable faster computations. The final block is the head block which outputs relative depth for each pixel.

3.3. Attention-Based Loss

The depth range of our HRSD dataset allows us to utilize the standard loss function for depth regression problems known as L_1 loss or Mean Absolute Error loss (MAE). Unlike a scale-invariant loss [19], whose variants are used for many MDE networks [7, 23, 39, 40], L_1 loss is more efficient and performs better [11]. However, training with only

Dataset	Algorithms	Error & Accuracy		
		AbsRel ↓	RMSE ↓	$\delta < 1.25$ ↑
NYU V2 [44]	ViT-B	0.115	0.509	0.828
	ViT-B + R	0.108	0.416	0.875
	ViT-B + R + AL	**0.104**	**0.362**	**0.916**
	DPT-B	0.101	0.375	0.895
	DPT-B + R	0.103	0.364	0.903
	DPT-B + R + AL	**0.094**	**0.310**	**0.945**
KITTI [22]	ViT-B	0.106	4.699	0.861
	ViT-B + R	0.101	4.321	0.889
	ViT-B + R + AL	**0.078**	**2.933**	**0.915**
	DPT-B	0.098	3.821	0.894
	DPT-B + R	0.069	2.781	0.939
	DPT-B + R + AL	**0.056**	**2.453**	**0.962**
HRSD	ViT-B	0.125	0.471	0.828
	ViT-B + R	0.107	0.342	0.882
	ViT-B+ R + AL	**0.099**	**0.322**	**0.912**
	DPT-B	0.118	0.421	0.835
	DPT-B + R	0.101	0.330	0.894
	DPT-B + R + AL	**0.074**	**0.288**	**0.921**

Table 1. Quantitative comparison on three RGB-D datasets. The three variants of ViT [17] and three variants of DPT [39] are trained on the proposed HRSD datasets.

L_1 loss leads to discontinuities and noisy artifacts in depth maps [5]. Other loss functions considered to aid L_1 loss include employing the edge accuracy between real and predicted depth maps known as Structural Similarity (SSIM) [51] used for the MDE network [4]. Inspired by [12], in our method, we use an attention-based supervision loss to smooth the overall prediction and control the number of depth discontinuities and noisy artifacts in the final output.

To estimate the true values of the attention map at each pixel p, we calculate A_p from the ground-truth depth map as

$$A_p = \text{SOFTMAX}\left(-\lambda \left| y_p - \hat{y}_p \right|\right) \quad (2)$$

where y_p, $\hat{y}_p$ are the ground-truth and predicted depth map, respectively, and λ is the hyper-parameter. We use the ground truth attention map (A_p) and the predicted attention values ($\hat{A}_p$) to calculate the attention-based loss term [12].

$$\mathcal{L}_{\text{as}} = \frac{1}{n} \sum_{p=0}^{n} \left| A_p - \hat{A}_p \right| \quad (3)$$

For training our final network, we define the final loss L between y and $\hat{y}$ as the weighted sum of two loss terms:-

$$L(y, \hat{y}) = \lambda L_{\text{depth}(}(y, \hat{y}) + L_{as}(y, \hat{y}) \quad (4)$$

where L_{depth} is the point-wise $L1$ loss defined on depth

values and is given as.

$$L_{\text{depth}}(y, \hat{y}) = \frac{1}{n} \sum_{p=0}^{n} |y_p - \hat{y}_p| \tag{5}$$

Note that only one weight parameter λ is required for calculating the loss function, and empirically = 0.1 is set in equation (4) [12].

4. Experiments

In this section, we make an extensive study that demonstrates the advantage of the proposed HRSD dataset. We evaluate public datasets, such as KITTI [22] and NYU [44], and the HRSD dataset. We show quantitative and qualitative comparisons in our experiments for analysis and discussion.

4.1. Training

Architecture details DPT [39] uses an encoder-decoder architecture. It uses ViT [17] architecture for its encoder and designs a decoder to get a depth map of the same size as the color image. Thus, DPT [39] use pretrained ViT [17] weights as their initial encoder weights to train the entire encoder-decoder architecture on datasets, such as KITTI (outdoor) [22], NYU V2 (indoor) scenes [44], and the MIDAS 3D dataset [40] for dense depth prediction. We adopt the same transformer-based encoder-decoder architecture used in DPT for our experiments and improve it further, as mentioned above in section 3.2. We initialize the encoder weights with DPT's [39] and train the entire encoder-decoder architecture on the proposed HRSD datasets (the decoder is initialized from scratch). To show the effect of proposed architecture changes, such as feature module and attention loss, we do six training experiments on the proposed HRSD dataset. Three training is done using DPT encoder weights [39], and they are:

1) Training with DPT [39] encoder weights * on the proposed HRSD dataset with no changes in architecture and loss functions which refer to DPT-B in this paper. *2)* Training DPT [39] weights with feature module (DPT-B + R). *3)* Training DPT [39] with feature module and attention-loss (DPT-B + R + AL).

Later, we do a similar process using ViT weights [17], i.e., initialize the encoder weights with ViT's [17] and train the entire encoder-decoder architecture on the proposed HRSD datasets and do three separate trainings.

Training details Our proposed algorithm is implemented in PyTorch, and we use 4 NVIDIA RTX A6000 48GB GPU for training. The proposed HRSD dataset contains $100,000$ images of resolution 1920×1080 and we use $75,000$ for training, 15000 for validation, and 10000 for the test dataset. We crop the images to the nearest 32 multiples, and the network outputs the depth at the same resolution as of color image. We trained the model for 80 epochs with a batch size of 4 and use the ADAMW [36] optimizer with $\beta 1$ = 0.9 & $\beta 2$ = 0.999 and a learning rate of 1×10^{-4} for encoder & 1×10^{-5} for the decoder. The learning rate decayed after 15 epochs by a factor of 10. Finally, we test the algorithm's performance on different datasets, such as KITTI, NYU, and HRSD, for indoor and outdoor scenes to show the generalization and robustness of the proposed algorithm in real-world scenes.

*In this paper, for training and evaluation, DPT weights refers to DPT-Hybrid weights as provided in the original DPT [39] paper.

Dataset	Algorithms	Error & Accuracy		
		AbsRel ↓	RMSE ↓	$\delta < 1.25$ ↑
NYU [44]	DPT [39]	0.110	0.392	0.864
	MultiRes [37]	0.102	0.347	0.921
	ViT-B+ R + AL	0.104	0.362	0.916
	DPT-B + R + AL	**0.094**	**0.310**	**0.945**
KITTI [22]	DPT [39]	0.114	4.773	0.849
	MultiRes [37]	0.059	2.756	0.956
	ViT-B+ R + AL	0.078	2.933	0.915
	DPT-B + R + AL	**0.056**	**2.453**	**0.962**
HRSD	DPT [39]	0.127	0.494	0.822
	MultiRes [37]	0.096	0.339	0.920
	ViT-B + R + AL	0.099	0.322	0.912
	DPT-B + R + AL	**0.074**	**0.288**	**0.921**

Table 2. Quantitative comparison on three RGB-D datasets. Here DPT [39] and MultiRes [37] results are obtained using the author's weights. ViT-B + R + AL and DPT-B + R + AL are the variants of ViT [17] and DPT [39] trained on the proposed HRSD datasets.

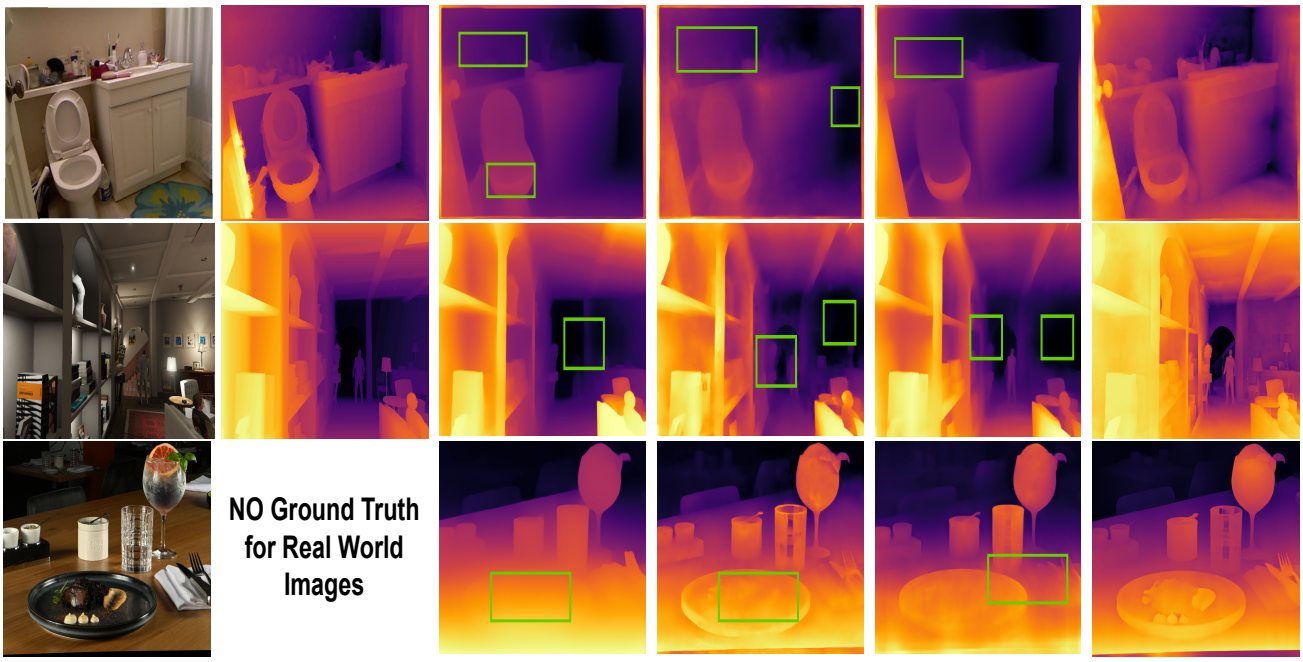
4.2. Evaluation

Like [19, 39, 40], we use three error metrics, such as RMSE, AbsRel, and percentage of correct pixels, to evaluate the performance of depth maps.

Datasets for evaluation To demonstrate the competitiveness of our approach, we evaluate the methods against KITTI [22] and NYU V2 [44] RGB-D datasets. These are the standard datasets for outdoor and indoor scenes, respectively. KITTI dataset contains 697 images (1216×352) with re-projected Lidar points as sparse depth maps and NYU V2 contains 694 images (640×480). In addition, we evaluate depth maps on 10,000 images (1920×1080) of our HRSD dataset as high-resolution datasets like [40, 48] are unavailable to the public.

4.3. Results

Quantitative Results To show the effect of the proposed HRSD datasets on different variants of training, we make a quantitative comparison of the evaluation of different datasets and present them in Table 1 and Table 2. In Ta-



(a) RGB (b) GT-Depth (c) DPT [39] (d) MultiRes [37] (e) ViT-B + R + AL (f) DPT-B + R + AL

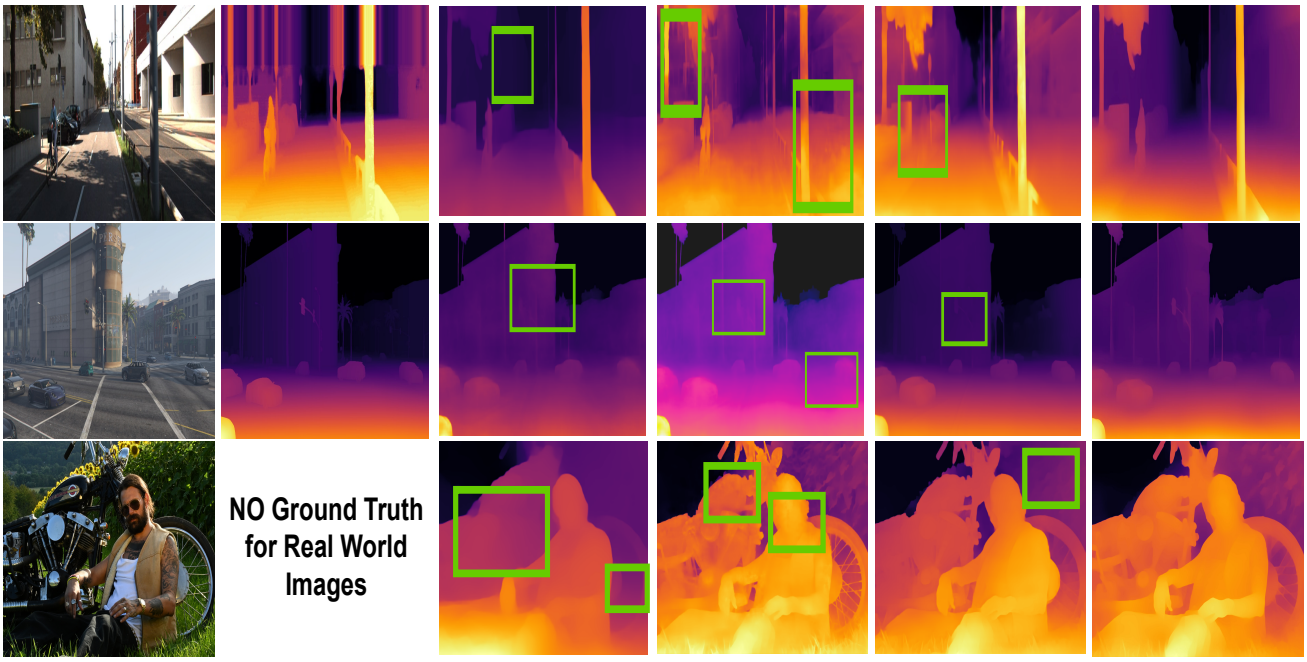
Figure 5. Indoor Scenes. 1st Row:- NYU [44]. 2nd Row:- HRSD indoor. 3rd Row:- RealWorld. Our DPT-B + R + AL gives a consistent depth map across all regions and displays sharp structure for overall objects i.e. items on the table in the real-world images. Original DPT [39] fails to identify objects in the background as shown by the green rectangles i.e. no structure of human in HRSD indoor. Multires [37] leads to an inconsistent depth map highlighted by green rectangles i.e. the toilet seat in NYU image.

ble 1, we compare six variants of training performed using DPT [39] encoder weights & ViT [17] weights as described earlier. The addition of the feature module and attention-loss performs better in all three datasets in all metrics. In Table 2, we compare the original DPT [39] and MultiRes [37] with the proposed variant of ViT (ViT-B+ R + AL) and DPT (DPT-B+ R + AL). From Table 2, we can conclude that the proposed variants are better in almost all the datasets. This indicates the effectiveness of the proposed HRSD datasets, which results in a lower absolute error and higher accuracy.

Qualitative Results The quantitative results are supported by visual comparisons in Figures 5 & 6. We compare indoor (Figure 5) and outdoor scenes (Figure 6) for analysis and discussion. We also include real-world scenes in both figures to test the performance of various algorithms. In Figures 5 & 6, DPT [39] fails to give precise depth edges and lacks details of the structure of background objects. Also, DPT [39] fails to get a clear object boundary in real-world scenes. MultiRes [37] can get a sharper depth map and details but provides inconsistent depth within an object, and these artifacts are highlighted using a rectangular box in Figures 5 & 6. Although the proposed variant ViT-B + R+ AL lacks details in the background, the DPT-B + R + AL gives a more structured and consistent depth with precise object shapes and clearer edges, even for objects further away in the scene. This is reasonable because DPT [39] is trained on RGB-D datasets, whereas ViT [17] is trained for image recognition tasks. Compared to both DPT [39] and MultiRes [37], our method results in a smoother depth map closest to ground truth maps.

Running time analysis We further compare the inference time from different algorithms and present them in Table 3. It shows the inference speed in milliseconds per frame, averaging over 400 images on the three different resolutions. Timings were conducted on an Intel I5 10th generation @ 2. 90GHz with eight physical cores and a single Nvidia RTX A6000 GPU. Multires [37] take the longest running time as it merges different resolutions to compute a high-quality depth map. DPT [39] and the proposed network take similar inference time on smaller resolutions. However, our proposed network is more efficient for high-resolution images due to the fewer patches produced by the feature-extraction module, which is later processed by the transformer layers.

Feature extraction module analysis Here, we compare the effect of the feature extraction module that provides image embedding to our transformer layers, as shown in Figure 4. ViT [17] uses a simple image flattening technique to transform the image into patches and comes in two variants with ViT-32 and ViT-16. Since we use the Resnet [24]



(a) RGB (b) GT-Depth (c) DPT [39] (d) MultiRes [37] (e) ViT-B + R + AL (f) DPT-B + R + AL

Figure 6. Outdoor Scenes. 1st Row:- KITTI [22]. 2ndRow:- HRSD outdoor. 3rd Row:- RealWorld. Similar to indoor scenes, our DPT-B + R + AL gives the best performance outputting a consistent depth map with precise overall structure i.e. the motorbike in the real-world image. Original DPT [39] again fails to identify objects in the background as shown by the green rectangle i.e. no structure of background buildings in the KITTI image. Multires [37] leads to the inconsistent depth map, highlighted by green rectangles i.e. depth around the biker's body is fluctuating in the real-world image.

Method	DPT [39]	MultiRes [37]	Proposed
640×480	17	50	**14**
1216×352	22	80	**20**
1920×1080	55	190	**38**

Table 3. Inference Speed (ms) per frame. 3 different resolutions are displayed for comparison. Here Proposed implies: DPT-B+R+AL.

Method	AbsRel ↓	RMSE ↓	$\delta > 1.25$↑
ViT-32 [17]	0.137	0.494	0.803
ViT-16 [17]	0.125	0.471	0.828
ViT-16+ R-101	0.112	0.387	0.852
ViT-16 + R-50	**0.107**	**0.342**	**0.882**

Table 4. Feature module analysis. Different variants of ViT [17] and Resnet [24] are experimented with to choose the best feature extraction module.

feature module to process color RGB images, we compare different pre-trained ResNet encoders, such as Resnet-50 and Resnet-101, with ViT [17]. In Table 4, we first make a comparison with ViT-16 and ViT-32. We use these weights as an initial encoder weight and train the entire encoder-decoder on the proposed HRSD datasets. We observe that ViT-16 achieves low error and higher accuracy than ViT-32. Later, we fixed ViT-16 as the backbone architecture and experimented with two resnet encoders, such as Resnet-50 and Resnet-101. We again make two different training, one with Resnet-50 with ViT-16 and the other with Resnet-101 as our initial encoder weights, and train the entire encoder-decoder on the proposed HRSD datasets. As shown in Table 4, introducing the Resnet encoders significantly improves the performance. However, Resnet-50 outperforms and thus becomes the final choice for our MDE network.

5. Conclusion

In this paper, we proposed to generate a high-quality synthetic RGB-D datasets from the game GTA-V [1] with precise dense depth maps. Since we can control all the aspects of the GTA game, we can capture scenes with varied lighting, different environments, and diverse objects. We trained the DPT [39] architecture with DPT [39] and ViT [17] weights on the proposed HRSD datasets. We observed a significant improvement in the performance of the depth maps, both objectively and subjectively. The performance is further improved by modifying the DPT [39] architecture and loss function.

References

[1] Grand theft auto v, [2014]. Includes all Grand theft auto online and Grand theft auto V free-to-access content updates. 2, 3, 8

[2] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, and Mannat Kaur Bingbing Liu. Bidirectional attention network for monocular depth estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11746–11752. IEEE, 2021. 3

[3] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-time rendering*. AK Peters/crc Press, 2019. 3

[4] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2019. 1, 3, 5

[5] Hritam Basak, Sagnik Ghosal, Mainak Sarkar, Mayukhmali Das, and Soham Chattopadhyay. Monocular depth estimation using encoder-decoder architecture and transfer learning from single rgb image. In *2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pages 1–6, 2020. 5

[6] Tali Basha, Shai Avidan, Alexander Hornung, and Wojciech Matusik. Structure and motion from scene registration. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1426–1433, 2012. 2

[7] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 4, 5

[8] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10231–10241, 2021. 3

[9] Adriano Cardace, Luca De Luigi, Pierluigi Zama Ramirez, Samuele Salti, and Luigi Di Stefano. Plugging self-supervised monocular depth into unsupervised domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1129–1139, 2022. 3

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3

[11] Marcela Carvalho, Bertrand Le Saux, Pauline Trouvé-Peloux, Andrés Almansa, and Frédéric Champagnat. On regression losses for deep depth estimation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2915–2919, 2018. 5

[12] Wenjie Chang, Yueyi Zhang, and Zhiwei Xiong. Transformer-based monocular depth estimation with attention supervision. 2021. 5, 6

[13] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021. 4

[14] 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 transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 3

[15] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv preprint arXiv:1602.02481*, 2016. 2

[16] Anh-Dzung Doan, Abdul Mohsi Jawaid, Thanh-Toan Do, and Tat-Jun Chin. G2d: from gta to data. *arXiv preprint arXiv:1806.07381*, 2018. 4

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3, 4, 5, 6, 7, 8

[18] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE international conference on computer vision*, pages 2650–2658, 2015. 1

[19] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 4, 5, 6

[20] Matteo Fabbri, Guillem Brasó, Gianluca Maugeri, Orcun Cetintas, Riccardo Gasparini, Aljoša Ošep, Simone Calderara, Laura Leal-Taixé, and Rita Cucchiara. Motsynth: How can synthetic data help pedestrian detection and tracking? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10849–10859, 2021. 3

[21] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016. 1

[22] Andreas Geiger, P Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: the kitti dataset. *The International Journal of Robotics Research*, 32:1231–1237, 09 2013. 2, 5, 6, 8

[23] Mohammad Mahdi Haji-Esmaeili and Gholamali Montazer. Playing for depth, 2018. 2, 3, 4, 5

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 7, 8

[25] Braden Hurl, Krzysztof Czarnecki, and Steven Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 2522–2529. IEEE, 2019. 3

[26] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *European Conference on Computer Vision*, pages 581–597. Springer, 2020. 3

[27] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work.

In *Consumer depth cameras for computer vision*, pages 141–165. Springer, 2013. 2

[28] Wei Jiang, Eduard Trulls, Jan Hosang, Andrea Tagliasacchi, and Kwang Moo Yi. Cotr: Correspondence transformer for matching across images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6207–6217, 2021. 2

[29] Benjamin Kiefer, David Ott, and Andreas Zell. Leveraging synthetic data in object detection on unmanned aerial vehicles. *arXiv preprint arXiv:2112.12252*, 2021. 3

[30] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016. 1

[31] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 3

[32] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1833–1844, 2021. 2

[33] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017. 5

[34] Chenxi Liu, Liang-Chieh Chen, Florian Schroff, Hartwig Adam, Wei Hua, Alan L Yuille, and Li Fei-Fei. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 82–92, 2019. 3

[35] Fayao Liu, Chunhua Shen, and Guosheng 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. 1

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[37] S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9680–9689, 2021. 2, 6, 7, 8

[38] Umberto Michieli, Matteo Biasetton, Gianluca Agresti, and Pietro Zanuttigh. Adversarial learning and self-teaching techniques for domain adaptation in semantic segmentation. *IEEE Transactions on Intelligent Vehicles*, 5(3):508–518, 2020. 3

[39] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[40] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2, 3, 4, 5, 6

[41] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 3

[42] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. *Advances in neural information processing systems*, 18, 2005. 1

[43] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings 36*, pages 31–42. Springer, 2014. 2

[44] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 2, 5, 6, 7

[45] Shuran Song, Samuel P. Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, 2015. 2

[46] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 2

[47] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3

[48] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 2, 3, 6

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 1, 5

[50] Kaixuan Wang and Shaojie Shen. Flow-motion and depth network for monocular stereo and beyond. *IEEE Robotics and Automation Letters*, 5(2):3307–3314, 2020. 3

[51] Zhou 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. 5

[52] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. Fastdepth: Fast monocular depth estimation on embedded systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6101–6108. IEEE, 2019. 1

[53] Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang, and Feng Qian. Monocular depth estimation based on deep learning: An overview. *Science China Technological Sciences*, 63(9):1612–1627, 2020. 3

[54] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 1, 3, 4