Computers & Graphics (2023)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Vision UFormer: Long-Range Monocular Absolute Depth Estimation

Tomas Polasek[a], Martin Čadík[a,*], Yosi Keller[b], Bedrich Benes[c]

[a] *Faculty of Information Technology, Brno University of Technology, Bozetechova 2/1, Brno, 612 00, Czech Republic*
[b] *Faculty of Engineering, Bar-Ilan University, Ramat Gan, 1102-1105, Israel*
[c] *305 N University St., Purdue University, West Lafayette, IN 47907-2107, United States of America*

ARTICLE INFO

Article history:
Received February 20, 2023

2000 MSC: 94A08, 68U10

Keywords: Absolute Depth Prediction, Monocular, Long-range, Transformer

ABSTRACT

We introduce Vision UFormer (ViUT), a novel deep neural long-range monocular depth estimator. The input is an RGB image, and the output is an image that stores the absolute distance of the object in the scene as its per-pixel values. ViUT consists of a Transformer encoder and a ResNet decoder combined with the UNet style of skip connections. It is trained on 1M images across ten datasets in a staged regime that starts with easier-to-predict data such as indoor photographs and continues to more complex long-range outdoor scenes. We show that ViUT provides comparable results for normalized relative distances and short-range classical datasets such as NYUv2 and KITTI. We further show that it successfully estimates absolute long-range depth in meters. We validate ViUT on a wide variety of long-range scenes showing its high estimation capabilities with a relative improvement of up to 23%. Absolute depth estimation finds application in many areas, and we show its usability in image composition, range annotation, defocus, and scene reconstruction. Our models are available at **cphoto.fit.vutbr.cz/viut**.

1. Introduction

The vast majority of contemporary imaging sensors provide 2D information. However, depth is one of the most critical sources of information needed for many tasks, including relighting [1], scene understanding [2], and perception [3]. We introduce the `Vision UFormer` (`ViUT`), a network that provides absolute depth estimation from RGB images.

Recent approaches in machine learning allow depth prediction from 2D images that have the potential to replace, or at least complement, the expensive and often imprecise depth sensors. Moreover, 2D depth estimation is a passive process, whereas most physical depth sensors are active as they send and receive signals [4]. However, depth estimation from images is an ill-posed problem requiring large training datasets and is prone to errors. Thus, recent approaches to depth prediction are limited to the normalized relative depth that lacks the crucial scaling factor or close-range metric depth, which is not usable in open outdoor environments. A common problem is also the high dynamic range of estimated depths. If the scene includes a wide variety of objects close to the camera and another far interval, there will likely be a large quantization error leading to inaccurate prediction. This problem is further exacerbated in natural images, where common urban hints are absent.

Our key inspiration lies in the human ability to interpret a complex scene. Motivated by the idea that depth is only one modality used in such a task, we cast the model training as a multi-task scenario. However, unlike previous work, we not only predict multiple output modalities but also provide our network with more input modalities as well. `ViUT` analyzes the input modalities at multiple levels. `ViUT` consists of a Transformer [5, 6] encoder and a ResNet [7] decoder with UNet [8] style of skip connections. Finally, we use a staged training regime, starting with easier-to-predict synthetic data, leading into indoor photos and, finally long-range outdoor scenes.

*Corresponding author: Tel.: +420 54114 1272
e-mail: `ipolasek@fit.vut.cz` (Tomas Polasek), `cadik@fit.vut.cz` (Martin Čadík), `yosi.keller@gmail.com` (Yosi Keller), `bbenes@purdue.edu` (Bedrich Benes)

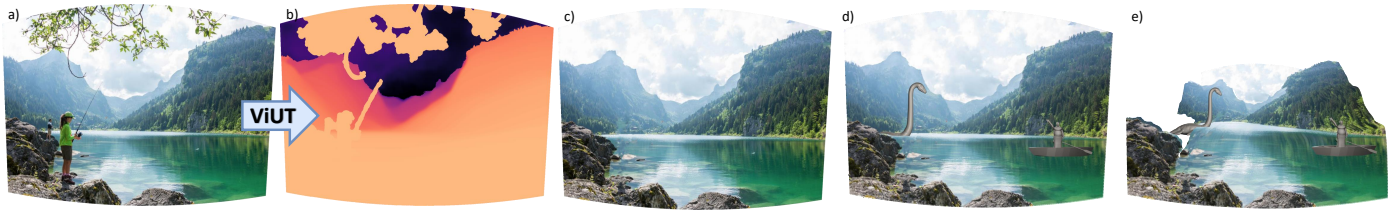

Fig. 1: **Depth Estimation:** `Vision UFormer` (`ViUT`) model uses the input RGB image **(a)** to estimate its absolute depth **(b)**. The resulting map can then be used for additional applications, such as object removal **(c)**, 3D scene manipulation **(d)**, or scene reconstruction **(e)**.

We used `ViUT`to estimate absolute depth in various scenes, and our results and validation show that the novel model architecture and training regime contribute to successfully tackling the under-constrained task of absolute depth prediction in natural images. We show that the multiscale approach allows us to predict depth in scenes with widely varying depth values, ranging from meters to tens of kilometers. We show applications of our method in image composition, range annotation, defocus, and scene reconstruction – see Fig. 1 and Sec. 4.5.
We claim the following contributions:

1. A novel dense depth prediction model `Vision UFormer` that combines the global context-aware Vision Transformer with a UNet spatial reduction.
2. A staged training regime that facilitates the training of the prediction model and allows the estimation of the metric depth for images with highly varied depths.

We evaluate our approach and compare it to other state-of-the-art techniques, showing a considerable improvement in the case of high dynamic long-range depths. We observe a relative improvement of up to 23.82% in accuracy under the threshold $\delta > 1.25$ and 22.99% in RMS. Our models are available at **cphoto.fit.vutbr.cz/viut**.

2. Related Work

2.1. Dense Regression Models

ML models in computer vision (CV) use the concept of localized spatial structure in natural images [9]. The initial attempts used Multilayer Perceptron (MLP), while the later models used convolutional layers [10]. Benchmark used to gauge model efficiency include classification [11, 12], segmentation [7, 8, 13], object detection [14, 15], and dense regression [16, 17].

Typically, CV models for dense regression – for example, depth prediction – utilize a sequence of convolutions with pooling operations, progressively increasing their receptive field until they reach global information spread [11]. Deep convolutional models such as the AlexNet [11] or VGG [12] use them to extract information about localized image patches. However, reaching wider receptive fields requires deeper networks, resulting in difficult training, higher data requirements, and prohibitive memory use [7]. Moreover, loss of resolution due to pooling operations leads to reduced fidelity and artifacts [18].

The resolution of feature maps can be kept higher by utilizing techniques such as dilated convolutions [16] or multiscale features [15]. However, network depth still presents an issue for training and data requirements. UNet [8] introduces U-shaped architecture with skip connections, which allow aggregation of multiple scales of feature maps. He *et al.* present the ResNet [7] architecture utilizing residual skip connections, which help propagate gradients, leading to very deep networks.

Although many vision tasks do not require global information diffusion, it has been shown to increase model performance, especially in dense prediction tasks [7, 19]. While convolutions allow reaching a global receptive field, their spatially-local nature requires multiple layers to achieve this, not allowing random access to the image.

Attention, introduced by Vaswani *et al.* [5], is used in the Transformer model to provide random access to an input sequence of tokens. In contrast to MLP, which has static pretrained weights, the attention mechanism builds a dynamic routing matrix based on the keys and queries extracted from the inputs. Although this approach is highly versatile, it carries a prohibitive memory cost of $O(n^2 D)$, where n is the sequence length and D is the token dimension. The Vision Transformer [6] (ViT) adapts the Transformer to image inputs by splitting the image into 16×16 patches, reducing memory consumption. Each patch is transformed through a learning embedding process and augmented with positional embedding. Then, it is passed through a sequence of multi-head self-attention (MHSA) followed by an output head producing the class estimates.

The ViT facilitates an equivalent of a global receptive field, making it useful for predicting dense depth [19]. However, the architecture proposed by Dosovitskiy *et al.* is designed for classification tasks and requires a large amount of data to train successfully. We take inspiration from the UNet [8] architecture to adapt it for depth prediction. We use a Vision Transformer as an encoder, extracting features that are then passed through deembedding and rescaling to transform them back into 2D maps. Finally, we use a residual decoder that progressively combines and upscales the feature maps into the output prediction.

2.2. Monocular Depth Prediction

Depth estimation is a special case of the dense regression task from CV. Specifically, monocular depth prediction transforms a single input RGB image into a depth map – a difficult task because of its loosely-constrained nature. Based on the target application, the predicted depth can be relative [23, 24, 25, 26, 27, 28] - preserving the ordinal nature but missing the scaling factor - or absolute/metric [29, 30, 31, 32, 33, 34, 35, 36]. Although early methods mainly focused on using Markov Random Fields [37] or non-parametric approaches [38], recent works are almost exclusively based on machine learning techniques of computer vision [27].

Due to the increased availability of data, relative depth prediction techniques are usually more robust and easier to realize. Eigen *et al.* [23] used a fully-convolutional model with

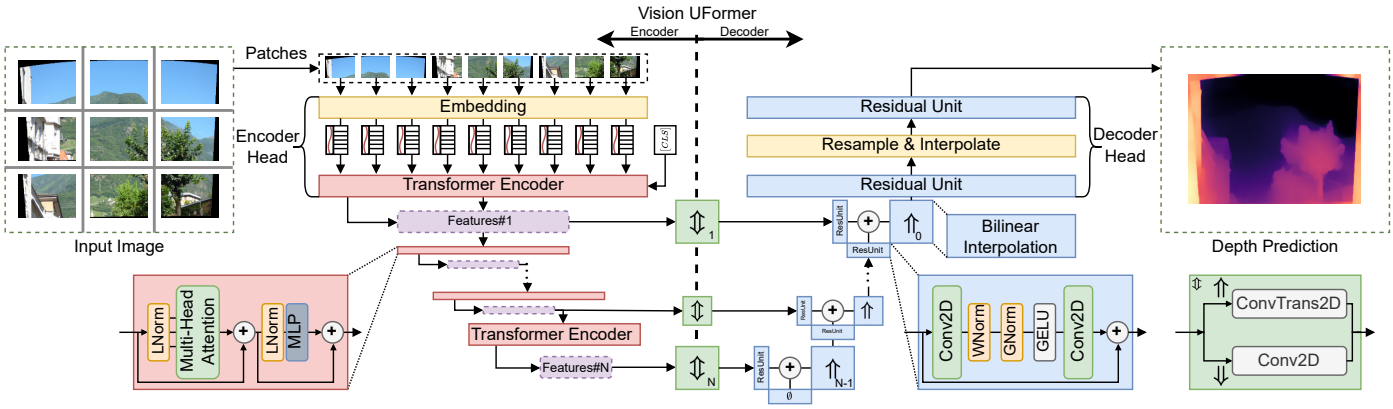

Fig. 2: **Model Overview:** `Vision UFormer` (`ViUT`) consists of a Vision Transformer [5, 6] encoder and ResNet [7] decoder in a UNet [8] configuration. The input image is split into embedded patches and passed through a sequence of multi-head self-attention layers. Multi-scale feature vectors are extracted from individual tiers of the encoder and processed into 2D feature maps by the *up-down-rescale* operation (⇳). The decoder aggregates these maps, upscaling them (⇑) with bilinear interpolation into the final depth prediction. We use Group Normalization [20] (GNorm), Weight Normalization [21] (WNorm), and Gaussian Error Linear Units [22] (GELU) within the residual units to stabilize the training.

two branches – one calculated a coarse global depth map, while the second refined the results with fine local details. In [24], they generalized this approach to multiple scales while also using a multitask objective [39]. GC-Net [25] represents a semi-supervised approach using a pair of stereo images to predict their disparity. To obtain higher resolution depth maps, Fu *et al.* [26] introduced a Deep Ordinal Regression Network, adopting a multi-scale structure to avoid spatial pooling. Xian *et al.* [27] use web stereo images to automatically generate annotations, predicting depth in outdoor scenarios. Sequential data is processed in [28] by a convolutional LSTM encoder, extracting spatiotemporal features.

Metric depth prediction techniques estimate the relative depth maps and provide the correct scaling factor. This task is complicated by the limited available data, especially for long-range outdoor scenarios [4]. To simplify the initial task, Zoran *et al.* [29] start by estimating the order between pairs of points and follow with training on the full metric data. Chen *et al.* [30] build upon this approach by using a large dataset of point-wise ordinal relations and a fully-convolutional network with skip connections. Liu *et al.* [31] train on relations between neighboring superpixels and use a Conditional Random Field to constrain the output, producing sharper details. Instead, the model presented by [32] instead trains directly on depth derivatives of different orders, orientations, and scales. Li *et al.* [33] use a two-branch network, cooperatively processing both depths and gradients to produce superior details. The MiDaS model [34] uses a multi-scale ResNet [27] architecture trained directly on the depths. It acknowledges the missing ground truth data using a mix of datasets with varying properties, using multi-objective learning [39] and auxiliary tasks. Dense Prediction Transformer (DPT) [36] builds on the mixed data set, moving from a fully convolutional network to the Transformer architecture instead. AdaBins [35] utilizes a transformer-based architecture that divides the depth into bins, which are estimated for each image. DepthFormer [40] uses multi-frame prediction combined with geometric constraints for improved depth estimate. Finally, Self-Distilled Feature Aggregation [41] employs three branches for offset feature maps to better aggregate multi-scale features.

We focus on the challenging scenario of metric depth prediction for highly variable long-range natural scenes with distances ranging from meters to tens of kilometers. Most convolution-based techniques fail at this task due to global context requirements that require large amounts of data to train. We amend the former problem by using the Transformer model like DPT [36] and AdaBins [35]. However, in contrast to previous work, we use a staged training regime, progressively training from easier to more difficult data and multiple data modalities.

2.3. Applications

Although many devices have multiple cameras or even active scanning sensors that allow the recovery of depth information, their quality and range are still limited. Relative depths are sufficient for some effects [42], and the correctly scaled metric depths allow for true scene manipulation. One of the important applications is dehazing [1, 43, 44], compensating for atmospheric scattering to sharpen the images. Deblurring [45] uses depth to bring parts of the image into focus, and relighting [1] brings the image content to different illumination using depth. Another application is depth completion [46], which attempts to create dense depth maps from sparse depth measurements. Other applications include novel view synthesis [47], object manipulation [1, 42, 48], depth of field [49, 50, 51, 52], inpainting [53, 54], shadow placement to images [55], and reconstruction [38, 56].

3. Proposed Method

We focus on the task of monocular depth prediction, which we formulate as a regression task. Given an input RGB image $I \in \langle 0, 1 \rangle^{w \times h \times 3}$, we search for a parameterized function $f_\theta(I) = d$ producing the depth map $d \in \mathbb{R}_+^{w \times h \times 1}$ containing the estimated distance in meters for each pixel. This is generally an ill-posed task requiring a deeper knowledge of the physical world.

We propose a novel network architecture named `Vision UFormer` (`ViUT`) (Fig. 2). It is based on the UNet [8] encoder-decoder design, facilitating complex multi-scale analysis of the input map. We use a Vision Transformer [5, 6] as the encoder and a ResNet [7] styled network for the decoder. This allows the model to utilize the global receptive field provided by the attention mechanism, allowing training on the limited available data. Moreover, we combine this with a staged training regime, training the model from easier to more difficult datasets.

3.1. Vision UFormer

This section presents the architectural details of the ViUT model, seen in Fig. 2, which is used to approximate the function $f_\theta(I)$. We estimate the function's parameters θ by training the prediction model end-to-end.

Model Architecture: is based on the UNet [8] encoder-decoder architecture, which we modify to make it more viable for the prediction of the high dynamic range of depth values present in our target data. Typically, CV models for dense regression utilize a sequence of convolutions and pooling operations, progressively increasing their receptive field [7]. However, this poses problems with training and graphical artifacts and results in limited information diffusion [18]. We side-step these issues by replacing the convolutional encoder of the UNet model [8] with a stack of multi-head self-attention (MHSA) [5] modules, keeping the decoder as a fully-convolutional ResNet [7].

As a backbone for the encoder (Fig. 2 left), we use the small, base, and large variants of the ViT defined in [6]. Based on the Vision Transformer (ViT) [6] model, we split the input RGB image with a resolution of 384×384 into 16×16 patches and embed them into feature vectors by utilizing a 2DConv layer. We use learnable 1D positional embedding, as recommended by Dosovitskiy *et al.* [6]. Each tier of the MHSA modules takes the resulting stream of tokens $T_i \in \mathbb{R}^n$ and transforms them into a same-length sequence of output tokens $T_o \in \mathbb{R}^n$ along with a special *[cls]* token representing aggregate information.

We tap the encoder at selected tiers (Fig. 2 middle, detail in bottom-right) to recover tokens which we use as features for the depth prediction. However, since these tokens are in their embedded state, we first transform them back into 2D feature maps through the *up-down-rescale* $\Updownarrow$ operation. This includes resampling of the features into a regular grid. We use strided 2D convolution for down-scaling and transposed 2D convolution for up-scaling. This approach allows matching of dimensions in the encoder to those of the decoder.

Finally, the ResNet [7] decoder (Fig. 2 right) consists of a sequence of residual tiers ending with the output prediction head. Each tier contains two residual units – one for the running signal other for the feature map recovered from the encoder. Since the lowest tier does not have any running signal, we use $\vec{0}$ instead. To stabilize training, we use Group Normalizationt [20], Weight Normalization [21], and GELU activations [22]. After each signal is processed by their residual unit, we combine them through addition. The $\Uparrow$ operation guarantees that both maps share the same dimension. By utilizing bilinear interpolation, we avoid the graphical artifacts associated with pooling.

3.2. Training Data

Training of neural networks requires a large amount of data to prevent overfitting and assure a high level of generalization. This is especially true for the Vision Transformer model because of its large amount of weights ($\sim 300M$) and weaker inductive bias when compared to convolutions [6]. However, getting accurate ground-truth depth data is difficult, especially in the long-range metric monocular depth scenario. Most existing datasets are collected through depth sensors or inferred using multi-view stereo [57]. However, devices such as the Microsoft Kinect [58] are limited to indoor scenes, LIDARs [59] provide limited-range sparse depth maps, and stereoscopic reconstruction requires wide camera separation.

#	Dataset	In	Out	Dns	Abs	Mod	Images	δ
1	EDEN [60]	✗	✓	✓	✓	RLINFT	368.7K	7 m
2	SINTEL [61]	✓	✓	✓	✓	RLFT	1.06K	288 m
3	DIW [62]	✓	✓	✗	✗	PD	478.9K	N/A
4	NYU [58]	✓	✗	✓	✓	PIL	1.45K	4 m
5	TUM [63]	✓	✗	✓	✓	P	140.2K	7 m
6	MEGA [64]	✗	✓	✓	✗	PD	128.2K	N/A
7	ETH3D [65]	✓	✓	✗	✓	P	5.25K	16 m
8	KITTI [59]	✗	✓	✗	✓	P	93.7K	19 km
9	GP3K [66]	✗	✓	✓	✓	PRLN	3.1K	39 km
10	LSAR [67]	✗	✓	✓	✓	PRLNS	8.2K	35 km

Table 1: **Training Data:** (left to right) training order, name, indoor, outdoor, dense, absolute, modalities, images, and average depth range. The modalities include photos (P), synthetic renders (R), labels (L), instances (I), normals (N), flow (F), intrinsics (T), point/ordinal depth (D), and silhouettes (S).

To reach a sufficient amount of training data, we use a combination of the ten datasets presented in Tab. 1. EDEN [60] dataset contains synthetic images and depths from garden scenes under various lighting conditions with a multitude of other modalities. SINTEL [61] uses synthetic 3D scenes rendered with various effects to provide semi-realistic alternatives to photographs. DIW [62] dataset covers a wide range of scenes but provides only a single manually annotated point-wise ordinal relationship for each sample image. NYUv2 [58] utilizes a Microsoft Kinect RGB-D camera to directly capture metric depth data, offering a clean labeled dataset. TUM [63] provides sequences of RGB-D images from indoor environments along with reconstructed relative depths. MegaDepth [64] dataset uses multi-view photos from a wide range of internet collections to reconstruct their relative depth maps and ordinal depth masks. ETH3D [65] is a high-resolution dataset covering both indoor and outdoor scenes, for which sparse metric depths are provided. KITTI [59] dataset contains sequences of images covering cars driving through urban outdoor areas along with sparse LIDAR depth maps. Geopose3K [66] focuses on mountainous long-range environments with aligned photographs along with synthetic rendered depth maps. The LandscapeAR [67] augments photos from long-range outdoor environments with synthetic renders and model-based depth maps. The rendering of synthetic depth maps for both GP3K [66] and LSAR [67] was performed by their original authors, and they produce precise ground-truth depth maps by aligning the virtual camera with a terrain model. The distortion present in the final images is due to the cylindrical projection and image alignment made by the authors of these datasets. Finally, we also derive additional data for use in training, converting to dense depth maps, calculating masks, and normalizing the RGB images. For details, please see the supplementary materials.

3.3. Model Training

Although the amount of data provided by the above datasets (Tab. 1) is substantial, we find that training on any one of them was insufficient to successfully train the ViUT model for long-range metric depth estimation. For this reason, we propose the staged training regime.

Inspired by Curriculum Learning [68], we train the network in stages, from easier tasks to more difficult ones. We order the datasets from Sec. 3.2 by their difficulty, resulting in the order presented in Tab. 1. To define the order, we primarily consider the quality of the available depth ground truth, the available scenes, and the range of depth values present. For example, we start with the synthetic EDEN [60] dataset, which provides rendered images with a simple shading scheme and pixel-perfect depth maps. Specifically, the simple shading excludes realistic effects – such as shadows, reflections, and atmospheric or weather effects. Conversely, we choose DIW [62] as the first source of non-synthetic since its task is limited to choosing ordinal relation between two points within the image. Finally, the most complex datasets, the GeoPose3K [66] and LandscapeAR [67], are used last. They present the most significant challenge with long-range open scenes. At each stage, we initialize a fresh set of input and output heads. For each input and output modality, there is a separate head meaning that the render input head for EDEN and SINTEL are both freshly initialized. This approach led to the best training performance.

We combine this approach with multi-task learning inspired by the Multi-Objective Optimization [39]. Data within each dataset has its unique specifics. This concerns not only the output modalities (Tab. 1) but the form of the input images as well. We consider two input modalities: photos and renders and nine output modalities: ordinal/relative/metric depth, segmentations, instances, normals, optical flow, diffuse color, and shading.

For our loss function, we combine the gradient matching term from MegaDepth [64] with the scale and shift-invariant losses used by the MiDaS model [34]. During the staged training, we use the scale-invariant loss function $\mathcal{L}_{si}$:

$$\mathcal{L}_{si}(d_i, d'_i, m_i) = \mathcal{L}_{ssi}(\hat{d}_i, \hat{d'}_i, m_i) + \mathcal{L}_g(\hat{d}_i, \hat{d'}_i, m_i), \quad (1)$$

where m_i is the mask and d_i, d'_i are the ground-truth and predicted depth, respectively. As per [34], $\hat{d} = (d - \mu(d))/\sigma(d)$ represents the depth value transformed to have zero translation and unit scale, where $\mu(d) = median(d)$ and $\sigma(d) = (\sum m_i)^{-1} \sum m_i |d_i - \mu(d)|$ are calculated over each training batch. The trimmed scale and shift-invariant loss are then [34]:

$$\mathcal{L}_{ssi}(\hat{d}, \hat{d'}, m) = \frac{1}{M} \sum_{i}^{M_m} \left(\hat{d'}_i - \hat{d}_i\right)^2, \quad (2)$$

where $\left|\hat{d'}_i - \hat{d}_i\right| \leq \left|\hat{d'}_{i+1} - \hat{d}_{i+1}\right|$, $M = \sum m_i$, and $M_m = \gamma M$. This results in considering only the $\gamma\%$ of the lowest mean squared errors for each sample. The gradient matching loss is then [64]:

$$\mathcal{L}_g(\hat{d}, \hat{d'}, m) = \frac{1}{M} \sum_{k}^{K} \sum_{x,y}^{W_k, H_k} \left(\left|\nabla_x R^k_{xy}\right| + \left|\nabla_y R^k_{xy}\right|\right), \quad (3)$$

where R^k_{xy} is the difference at position (x, y) and scale k.

We use an additional scale-preserving loss for the final fine-tuning, which no longer needs to be invariant to shift and scale. We define it as:

$$\mathcal{L}_{sp}(d_i, d'_i, m_i) = \mathcal{L}_d(d_i, d'_i, m_i) + \mathcal{L}_g(d_i, d'_i, m_i), \quad (4)$$

where the $\mathcal{L}_d(d, d', m) = 1/M \sum (d' - d)^2$ represents a simple mean squared error loss.

The model training procedure starts by initializing the encoder using weights pre-trained on the ImageNet [11]. We then extract the encoder input head (Fig. 2) consisting of patch embedding, dropout, and the first MHSA block. Similarly, we initialize the decoder's weights, preparing its output head. We then proceed with the staged training regime, training on datasets in order, as presented in Tab. 1. All of the ten datasets used in training were first divided into training and testing sets. We use the training sets for the Staged Training and the testing is always performed on the sets which are never used for training. Where possible, we keep to the original training and testing split as specified by the authors, falling back to an 80 : 20 split.

We consider the available input and output modalities during each training stage as follows. We initialize a fresh head for each input and output modality at each training stage, copying the rest of the network from the last stage. Then, we train on batches of alternate modalities.

We optimize the model using the AMSGrad variant of ADAM method [69, 70] with the $\mathcal{L}_{si}$ loss, running it until convergence with a mini-batch size of four. The learning rate is initialized to the value of $\alpha = 0.0005$, reducing by a factor of ten each time the loss function does not improve by at least 0.01% in the last five epochs, *i.e.,* using the reduce-on-plateau technique. Finally, we fine-tune the model on the same training test split with fresh heads, utilizing the scale-preserving loss $\mathcal{L}_{sp}$.

4. Implementation, Experiments, and Results

We conducted comparative experiments on several datasets described in Sec. 3.2 to show the efficiency of the `ViUT` depth estimator. We organized the experiments into three categories: 1) ablation study analyzing individual components of our method, 2) comparison to other SotA depth prediction techniques, and 3) applications showing its uses.

4.1. Implementation

We implemented the `ViUT` model in Python using the PyTorch framework accelerated with the CUDA backend. The training and inference measurements were performed on a desktop computer with AMD Ryzen 5 3600 processor, 48GB RAM, and NVIDIA GeForce RTX 3080 10GB GPU. The `ViUT` model architecture takes a week of training from start to finish, with individual stages ranging from two hours for SINTEL [61] to 24 hours for MegaDepth [64].

4.2. Evaluation Protocol

The model evaluation was performed on the test data of each respective dataset in Sec. 3.2. We evaluate each fully trained model by first performing a limited fine-tuning to the target dataset, training them with a lower learning rate $\alpha = 0.00005$ using the AMSGrad Adam [69, 70] with the $\mathcal{L}_{sp}$ loss (Sec. 3.3). We then follow by calculating the predictions on the test set of the target dataset. We base our quantitative analysis of the performance on the following commonly accepted evaluation metrics [23, 27, 29]: Root Mean Square Error (RMS), Relative Error (REL), Logarithmic Error (Log10), Threshold ($\delta > thr$), and Weighted Human Disagreement Rate (WHDR). For their definition, please see the supplementary materials.

4.3. Ablation Study

Encoder Backbone To study how the choice of the encoder affects the prediction performance, we train the `ViUT` model's encoder with several base architectures of the Vision Transformer [6]. We use models pre-trained on the ImageNet [11] dataset. The results can be seen in Tab. 2 and include the following variants: ViT Tiny 224 (Tiny_224), ViT Base 224 (Base_224), ViT Base 384 (Base_384), and ViT Large (Large_384). We take each model through the complete staged training program, perform fine-tuning, and evaluate on the LandscapeAR [67]. As expected, we observe that larger encoder architectures improve results with diminishing returns. From the 13 MHSA modules and 7.66M parameters for Tiny_224, 13/95.97M for Base_224 and Base_384, and 24/319.60M for Large_384, we see relative improvement of around 6.50%, 4.44%, and 3.94%. This hints at the difficulty of the LandscapeAR [67] dataset. We also compare our results against a Convolution-based encoder in Tab. 3.

Encoder	RMS	REL	Log10	$\delta >$	1.25^1	1.25^2	1.25^3
Tiny_224	595.812	0.465	0.311		29.72	16.70	8.41
Base_224	341.068	0.294	0.227		23.22	8.83	3.57
Base_384	296.282	0.264	0.198		18.78	5.31	1.68
Large_384	**142.079**	**0.113**	**0.078**		**14.84**	**2.62**	**0.52**

Table 2: **Encoder Backbone:** Ablation experiments concerning encoder backbone architectures. The resulting metrics were calculated on the LandscapeAR [67] dataset.

	Variant	RMS	REL	Log10	$\delta >$	1.25^1	1.25^2	1.25^3
Enc	**Transformer**	**142.079**	**0.113**	**0.078**		**14.84**	**2.62**	**0.52**
	Convolution	155.176	0.123	0.087		17.67	4.10	1.12
Head	**Fresh**	**142.079**	**0.113**	**0.078**		**14.84**	**2.62**	**0.52**
	Old	146.121	0.116	0.080		15.40	3.48	0.78
Mod	**Included**	**142.079**	**0.113**	**0.078**		**14.84**	**2.62**	**0.52**
	Excluded	150.930	0.121	0.084		16.96	3.96	0.94

Table 3: **Training Variants:** Ablation experiments illustrating the improvements brought by different model and training choices. Encoder (**Enc**) includes Transformer and Convolution-based variants. Re-initializing heads (**Head**) with fresh weights, compared to using configuration from the last training step. Finally, use of additional modalities (**Mod**) results in additional performance gains. The resulting metrics were calculated on the LandscapeAR [67] dataset.

Decoder	RMS	REL	Log10	$\delta >$	1.25^1	1.25^2	1.25^3
(0)	1011.614	0.831	0.362		43.23	31.74	25.33
(1)	914.174	0.789	0.352		42.46	31.18	24.84
(24)	846.978	0.709	0.344		42.12	30.95	24.61
(0,4,6,8)	182.906	0.160	0.114		18.86	5.12	1.65
(0,4,8,16,24)	**142.079**	**0.113**	**0.078**		**14.84**	**2.62**	**0.52**

Table 4: **Decoder Tiers:** Ablation experiments of the number and location of the decoder tiers. Each value represents the target MHSA block within the encoder the tier connects to, with the 0th block being the encoder head. Results were calculated on the LandscapeAR [67] dataset.

Decoder Scaling We experiment with the sizing of the ResNet [7] decoder by varying the number of skip connections and their location within the encoder. In these experiments, we fix the encoder to use the Large_384 variant and follow the proposed staged training regime (Sec. 3.3). The encoder contains 25 MHSA modules – one for the input and 24 for the Vision Transformer [6] itself. We present the results in Tab 4. We show that Tapping the encoder only at one location – *i.e.,* removing the hierarchical decoder – leads to unsatisfactory results. We tap the encoder after the input head *(0)*, after the first MHSA module *(1)*, and after the last MHSA module *(24)*. However, regardless of location, the resulting model reaches a very low performance of 43.23%, 42.46%, and 42.12%, respectively. Finally, utilizing multiple locations *(0, 4, 6, 8)* improves the results by 23.26% and *(0, 4, 8, 16, 24)* by further 4.02%. Adding further tiers lead to instability and overfitting, possibly due to the insufficient size of the dataset. In Tab. 3, we show that re-initializing of head weights results in improved performance.

Training	RMS	REL	Log10	$\delta >$	1.25^1	1.25^2	1.25^3
No Pretrain	3272.564	2.663	0.618		62.89	45.34	36.61
ImageNet	1763.486	1.436	0.463		46.28	33.57	27.22
Staged 1-3	1008.459	0.731	0.378		37.75	27.68	21.01
Staged 1-6	735.287	0.487	0.317		35.55	25.77	18.35
Staged 1-8	386.382	0.311	0.303		30.23	17.39	8.93
Staged 1-10	**142.079**	**0.113**	**0.078**		**14.84**	**2.62**	**0.52**

Table 5: **Training Regime:** Results of the experiments with the initialization of the weight and staged training. Scenarios include no pre-training, ImageNet [11] baseline, and staged training using datasets #1 through #10 (Tab. 1). The metrics were calculated using the LandscapeAR [67] dataset. The *No Pre-train* did not converge on large datasets.

Staged Training The curriculum learning is critical for the prediction model to successfully learn prediction on the challenging GeoPose3K [66] and LandscapeAR [67] datasets. To corroborate, we perform a series of `ViUT` training experiments seen in Tab. 5. We fix the model architecture to the Large_384 variant along with taps positioned at *(0, 4, 8, 16, 24)* and fine-tune on the LandscapeAR [67] dataset, changing only the pre-training routine. We start with a *No Pre-Train* baseline, where we perform only random initialization of both the encoder and the decoder. The resulting model fails at the prediction task with a threshold error of 62.89%. Starting with encoder weights pre-trained on the ImageNet [11] provides a large boost of around 16.61%. Next, we bootstrap the model by using the staged training on a selection of the datasets, as seen in Tab. 1. Although pre-training on the synthetic datasets (#1 [60], #2 [61]) and the ordinal data from DIW (#3 [62]) with the *Staged 1-3* model improves the performance by 8.53%, the overall performance is still only 37.75%, indicating that using only the synthetic data is insufficient. Next, for the *Staged 1-6* model, we expand the stages to include datasets focusing on real photos (#4 [58], #5 [63], #6 [64]). Interestingly, we see only a 2.20% improvement, possibly hinting at the efficiency of the DIW [62] to provide the model with a strong enough corpus to generalize from the first two synthetic datasets to photographs. With *Staged 1-8*, we added the medium-range sparse metric datasets (#7 [65], #8 [59]), which led to an improvement of

Model	Test → Train ↓	DIW [62] WHDR	ETH3D [65] REL	Sintel [61] REL	KITTI [59] $\delta > 1.25^1$	NYU [58] $\delta > 1.25^1$	TUM [63] $\delta > 1.25^1$	GP3K [66] $\delta > 1.25^1$	LSAR [67] $\delta > 1.25^1$
ViUT (**ours**)	Staged Training	12.18%	0.093	**0.270**	**8.34**	8.98	10.14	**13.91**	**14.84**
DPT [36]	MIX 6 [36]	12.24%	**0.091**	0.276	8.44	**8.84**	**9.98**	17.87	19.48
AdaBins [35]	KITTI [59]	**12.02**%	0.122	0.294	8.40	10.4	10.27	18.47	19.94
MiDaS [34]	MIX 5 [36]	12.76%	0.131	0.324	23.29	9.65	15.02	24.82	26.48
MegaDepth [64]	MegaDepth [64]	24.26%	0.180	0.378	36.31	27.31	19.37	37.47	39.74
Pix2Pix [42]	Mannequin [42]	28.86%	0.179	0.415	47.64	18.34	17.96	43.27	43.24
WSVD [71]	WSVD [71]	21.59%	0.215	0.394	30.52	29.64	20.24	32.86	34.74

Table 6: **Depth Estimation:** Evaluation of prediction methods on datasets. Models were trained using the specified dataset or technique, fine-tuned to the test dataset, and evaluated (Sec. 4.2). Each model was trained using the specified **Train** dataset or technique, fine-tuned to the training split of the **Test** dataset, and evaluated on the testing split of the **Test** dataset. Further details are provided in Sec. 4.2.

Model	Training	RMS	REL	Log10	$\delta >$	1.25^1	1.25^2	1.25^3
ViUT (**ours**)	Staged	142.079	0.113	0.078		14.84	2.62	0.52
DPT [36]	MIX6 [36]	184.484	0.140	0.094		19.48	5.45	1.82
DPT [36]	Staged	169.812	0.129	0.086		17.81	4.02	1.10
AdaBins [35]	KITTI [59]	188.189	0.149	0.102		19.94	5.83	1.99
AdaBins [35]	Staged	174.132	0.138	0.099		18.02	4.35	1.34

Table 7: **Training Comparison:** Ablation experiments showing improved performance when using the Staged Training approach for other models. The resulting metrics were calculated on the LandscapeAR [67] dataset.

5.32%. Finally, for *Staged 1-10*, we add the challenging long-range datasets (#9 [66], #10 [67]) into the training mix. This results in an overall improvement of 15.39%, leading to the final performance figure of 14.84%. Notably, the training fails to converge when we attempt to train directly on long-range datasets. We also gain additional performance by using additional modalities provided by the datasets, as seen in Tab. 3. Furthermore, Staged Training can be used with other models, resulting in improved performance, as seen in Tab. 7.

4.4. Model Comparison

We compare our prediction model against other state-of-the-art approaches on selected datasets (Tab. 6). The `ViUT` model represents our final architecture using the Large_384 encoder, *(0, 4, 8, 16, 24)* decoder, and full a staged training regime. For other techniques, we use implementations provided by the authors along with best-performing pre-trained models when available. Each model is fine-tuned for the testing dataset using the data as specified in Sec. 3.2.

The results show that our method is on par with the previous state-of-the-art prediction models on most of the compared datasets while excelling on the challenging Geopose3K [66] and LandscapeAR [67] datasets. We find that `ViUT` has similar performance to other Transformer-based approaches – DPT [36], AdaBins [35] – showing the potential of its global receptive field for dense depth prediction.

The **Geopose3k** [66] data set presents several challenges for depth prediction models, as noted by the higher prediction errors of all models presented. It focuses on open long-range scenes from mountainous environments with highly varying depth values. The depths within the dataset range from 0 – 295km with an average min-max difference of 38.7km. Moreover, in these environments, the usually accessible depth hints are either absent or drastically reduced in quantity. The dataset contains 3,114 images, which is a relatively modest amount to train some of the larger architectures. Finally, the problem is also compounded by the quality of the ground-truth data, which is provided in the form of synthetically rendered depth maps that are missing crucial environmental details while being automatically matched to the source photographs.

The results for **Geopose3K** in Tab. 6 show the superior performance of the `ViUT` model. We reach $\delta > 1.25^1$ of 13.91%, which is 3.95% lower, an improvement of almost 22.12% when compared to the next-best method represented by the DPT [36]. Interestingly, we see a tendency for the Transformer-based models – `ViUT`, DPT [36], and AdaBins [35] – to achieve overall better results when compared to the other methods. This is possibly due to the global receptive field of the MHSA modules, which seem critical for predictions in open environments.

The **LandscapeAR** [67] dataset focuses on open and long-range mountainous environments, providing 9,242 RGB photos with synthetic depths and additional modalities. Although it has limitations similar to the Geopose3k [66] dataset, it is much more difficult to work with. The distances range from 0m to 292km, with an average range of 34.8km. Moreover, the alignment of the photos to the synthetic depth maps is not pixel-perfect, leading to difficult training. The results on LSAR as provided in Tab. 6 show the increased difficulty of the predictions for this dataset. The `ViUT` achieves $\delta > 1.25^1$ of 14.84%, which is 4.64% lower compared to the DPT [36], a relative improvement of 23.82%. Again, we see the superiority of the Transformer-based architectures with the original CNN-based MiDaS [34] model reaching only $\delta > 1.25^1$ of 26.48%

Our findings indicate that illumination and transparency affect the prediction accuracy as they would affect a human observer. The predictions are less precise in darker scenes with melding shadows, as compared to the well-lit scenes. Transparency can also result in the switching of predicted depth between a closer transparent object and its background or second transparent object – especially when the objects are mostly see-through.

4.5. Applications

In this section, we show some potential applications and uses for the `Vision UFormer` prediction model within the framework of long-range absolute depth prediction. Examples of application outputs can be found in Fig. 1, 3, and 4 with additional samples available on the project website. For visual samples of the predicted depths, see Fig. 5.

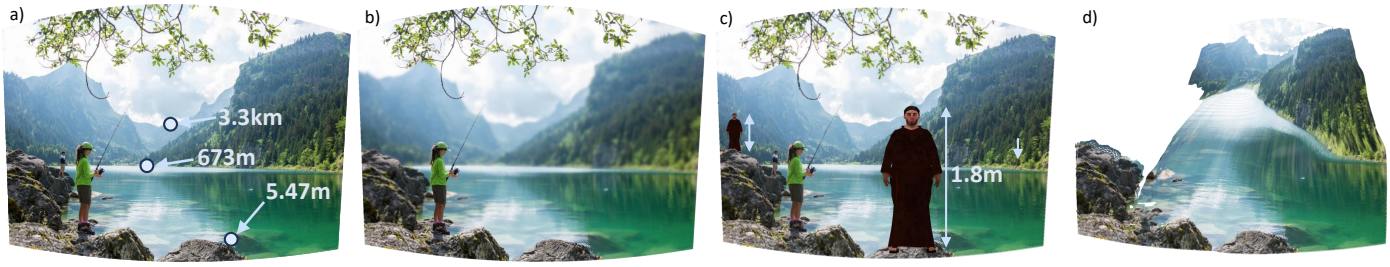

Fig. 3: **Depth Applications:** We show several applications of the depths produced by the ViUT model. **(a)** shows direct use of depth in annotation or range-finding. We apply a depth-dependant Gaussian filter in **(b)** for depth of field synthesis. In **(c)**, we show metric object insertion by placing three identical copies of a human character of approximately 1.8m in height into the scene. Thanks to the absolute depth, we observe the size of the character diminishing the further it is placed. Lastly, **(d)** shows a full 3D scene reconstruction with correct absolute scaling.

Range Finding One way to directly utilize the metric depth predictions is finding a distance to a given target object. We show this application in Fig. 3 a) and in Fig. 4 a). We directly read the predicted depth map and annotate the images to gain a rough estimate of the distance from the camera. This could be especially useful in open long-range scenes, where humans' abilities to gauge distance – even imprecisely – are quite poor.

Depth of Field A classical use for both absolute and relative depth is synthetic defocus and depth of field. We show this application in Fig. 3 b) and Fig. 4 c). The predicted depth map is used as a guide to control a variably-sized Gaussian blur filter. We first normalize the depth into a $0-1$ interval and then apply it as a multiplicative factor to the kernel size, rounding down with a minimum size of 1. This results in gradual increase of the filter size with distance from the point of focus while using the depth as a mask to keep the contours sharp.

Object Selection The predicted depth maps can also be used for object selection. In Fig. 4 d) and e), we use depth as a guide for automatic mask generation for a given object. The user selects the desired object and sets a range of depth values to be included. Then, a mask is generated automatically, which we visualize with red color. The resulting mask can then be used in further downstream tasks.

Object Removal After selecting an object, the mask can then be used to facilitate object removal. In Fig. 1 c), we use the depth map b) to select an object within a larger scene. Next, we apply the Resynthesizer plugin [72] to automatically synthesize the image texture within the masked region, using the depth as a texture guide. This approach can be used to remove unwanted objects from a photograph, as seen in Fig. 4 e) and f).

Scene Reconstruction The dense depth maps produced by ViUT can be used for a single-image scene reconstruction, shown in Fig. 3 d) and Fig. 4 h). We first pre-process the depth map by filtering and smoothing it with a Gaussian filter. Then, we use Open3D [73] to combine the input RGB image with the depth map, converting the result into a point cloud structure. Each point is colored with an RGB value of a given image pixel, while its distance from the camera is given by the depth recovered from the depth map. To ensure correct mapping, we use the properties of the camera used to take the given photo. Where unavailable, we instead fall back to default camera properties used by the library. Then, we visualize individual points as camera-oriented triangles with flat colors, placing them at a corresponding distance.

Object Placement Finally, the absolute metric depth predicted by ViUT can be used for realistic object placement. For example, see Fig. 3 c) and Fig. 4 h). By reconstructing the scene, we gain an interactive model and use the depth to recover corresponding scaling factors to gain its true metric scale. Then we placed a model of a human ~1.8 m in height into the scene at various locations. Critically, the absolute scale allows us to preserve the perspective. Thus the character model appears smaller the farther from the virtual camera it is placed.

5. Conclusion

We introduced Vision UFormer, a depth estimator that generates a per-pixel absolute depth map for input RGB monocular images. ViUT uses three main novel ideas: a Transformer encoder that provides global context, ResNet decoder combined with UNet skip connections that assemble the output in a hierarchical structure, and a staged training regime, allowing us to train this sizeable model. We showed that our ViUT generates results comparable to existing estimators on previously available relative and metric depth datasets. Its main strength comes from estimating long-range absolute depth values. Our ablation study shows that ViUT benefits from larger network architectures, where the number of skip connections helps to improve the prediction accuracy. We also show that the staged training regime is critical for its success. The long-range absolute dept estimation was then shown in computational photography applications such as image composition, synthetic defocus, and scene reconstruction. Future work includes extending our approach to different scenes by expanding the available data through the use of additional modalities. Finally, experiments with advanced decoder architectures represent promising way to further improve the prediction results.

Acknowledgments

This work was supported by project *LTAIZ19004 Deep-Learning Approach to Topographical Image Analysis*; by the Ministry of Education, Youth and Sports of the Czech Republic within the activity INTER-EXCELENCE (LT), subactivity INTER-ACTION (LTA), ID: SMSM2019LTAIZ. Computational resources were partly supplied by the project *"e-Infrastruktura CZ" (e-INFRA CZ ID:90140)* supported by the Ministry of Education, Youth and Sports of the Czech Republic.

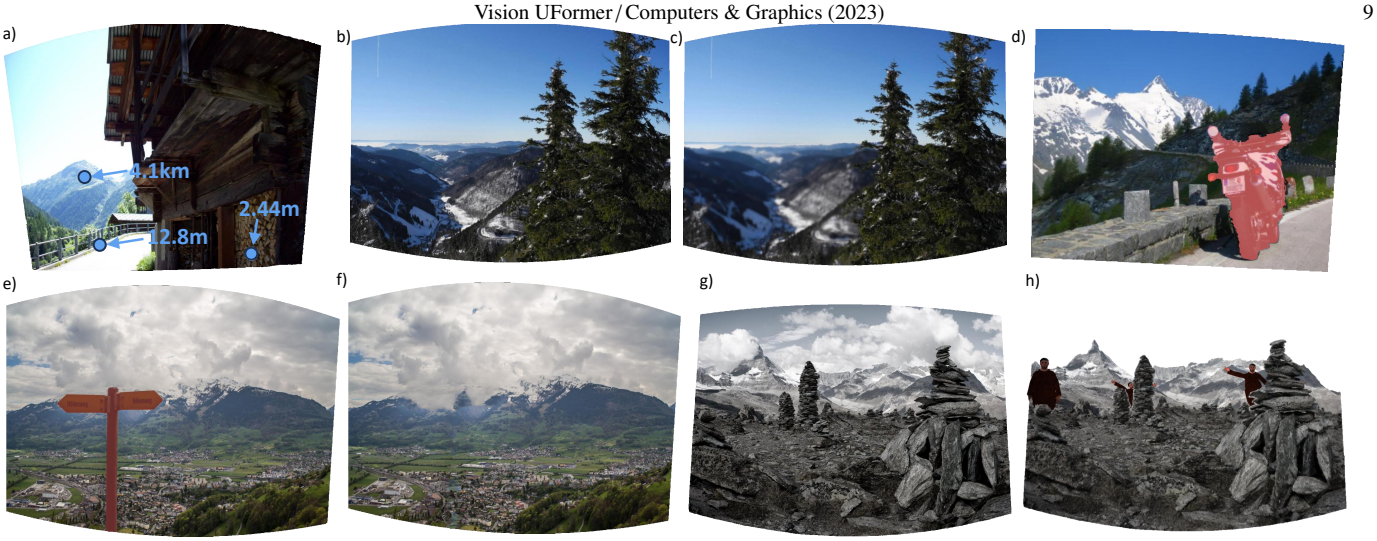

Fig. 4: **Applications Examples:** Additional application examples utilizing the predicted depths. **(a)** shows a direct interpretation of the depth maps for distance annotations. The source image **(b)** is enhanced with an artificial depth of field, introducing a degree of depth in **(c)**. We use the detected depth to automatically generate a mask for a selected object, visualized with red color ((**d**) and (**e**)). The mask is then used for object removal in **(f)**. Finally, we perform a scene reconstruction using the image **(g)** and place characters into the 3D scene (**h**).

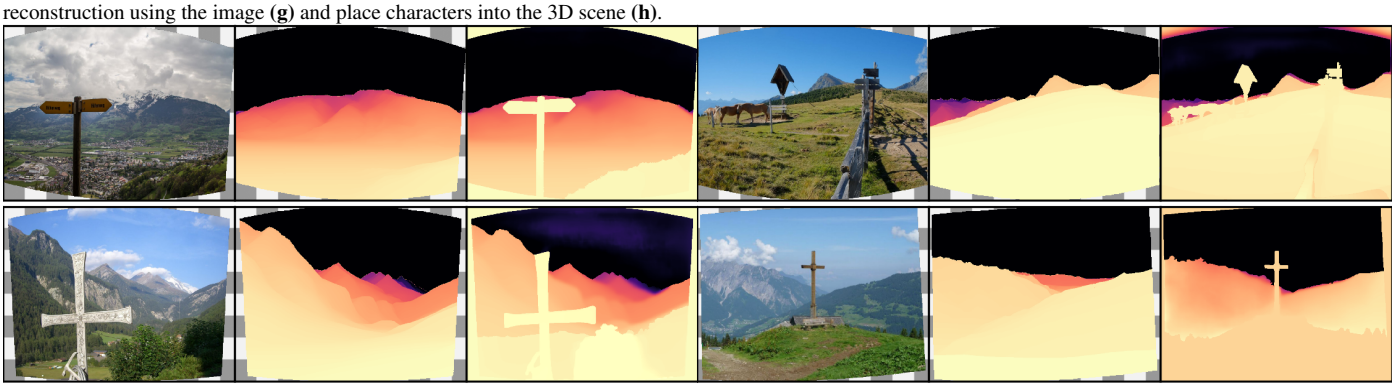

Fig. 5: **Depth Predictions:** Examples of depth maps predicted by the `ViUT` model. The images contain the input RGB image, visualized ground-truth depth, and the predicted depth. Notably, the ground-truth in the Geopose3K [66] dataset is completely missing the details which are correctly predicted by the `ViUT` model.

References

[1] Kopf, J, Neubert, B, Chen, B, Cohen, M, Cohen-Or, D, Deussen, O, et al. Deep photo: Model-based photograph enhancement and viewing. ACM Trans Graph 2008;27(5):1–10.

[2] Chen, PY, Liu, AH, Liu, YC, Wang, YCF. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2019,.

[3] Palmer, S. Vision Science: Photons to Phenomenology. A Bradford book; Bradford Bokk; 1999. ISBN 9780262161831.

[4] Ming, Y, Meng, X, Fan, C, Yu, H. Deep learning for monocular depth estimation: A review. Neurocomputing 2021;438:14–33.

[5] Vaswani, A, Shazeer, N, Parmar, N, Uszkoreit, J, Jones, L, Gomez, AN, et al. Attention is all you need. In: Adv. in Neural Inf. Proc. Systems (NIPS); vol. 30. Curran Associates, Inc.; 2017,.

[6] Dosovitskiy, A, Beyer, L, Kolesnikov, A, Weissenborn, D, Zhai, X, Unterthiner, T, et al. An image is worth 16x16 words: Transformers for image recognition at scale. 2021. `arXiv:2010.11929`.

[7] He, K, Zhang, X, Ren, S, Sun, J. Deep residual learning for image recognition. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2016,.

[8] Ronneberger, O, Fischer, P, Brox, T. U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer; 2015, p. 234–241.

[9] Hyvarinen, A, Hurri, J, Hoyer, P. Natural Image Statistics: A Probabilistic Approach to Early Computational Vision. Computational Imaging and Vision; Springer London; 2009. ISBN 9781848824911.

[10] LeCun, Y, Boser, B, Denker, JS, Henderson, D, Howard, RE, Hubbard, W, et al. Backpropagation applied to handwritten zip code recognition. Neural Computation 1989;1(4):541–551.

[11] Krizhevsky, A, Sutskever, I, Hinton, GE. Imagenet classification with deep convolutional neural networks. In: Pereira, F, Burges, CJC, Bottou, L, Weinberger, KQ, editors. Adv. in Neural Inf. Proc. Systems (NIPS); vol. 25. 2012,.

[12] Simonyan, K, Zisserman, A. Very deep convolutional networks for large-scale image recognition. 2015. `arXiv:1409.1556`.

[13] Lin, G, Milan, A, Shen, C, Reid, I. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2017,.

[14] Hariharan, B, Arbelaez, P, Girshick, R, Malik, J. Hypercolumns for object segmentation and fine-grained localization. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2015,.

[15] Lin, TY, Dollar, P, Girshick, R, He, K, Hariharan, B, Belongie, S. Feature pyramid networks for object detection. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2017,.

[16] Yu, F, Koltun, V. Multi-scale context aggregation by dilated convolutions. 2015.

[17] Zhao, H, Shi, J, Qi, X, Wang, X, Jia, J. Pyramid scene parsing network. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2017,.

[18] Wang, P, Chen, P, Yuan, Y, Liu, D, Huang, Z, Hou, X, et al. Understanding convolution for semantic segmentation. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). 2018, p. 1451–1460.

[19] Luo, W, Li, Y, Urtasun, R, Zemel, R. Understanding the effective receptive field in deep convolutional neural networks. In: Adv. in Neural Inf. Proc. Systems (NIPS); vol. 29. Curran Associates, Inc.; 2016,.

[20] Wu, Y, He, K. Group normalization. In: Proceedings of the European conference on computer vision (ECCV). 2018, p. 3–19.

[21] Salimans, T, Kingma, DP. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. Advances in neural information processing systems 2016;29.

[22] Hendrycks, D, Gimpel, K. Gaussian error linear units (gelus). arXiv preprint arXiv:160608415 2016;.

[23] Eigen, D, Puhrsch, C, Fergus, R. Depth map prediction from a single image using a multi-scale deep network. In: Adv. in Neural Inf. Proc. Systems (NIPS); vol. 27. Curran Associates, Inc.; 2014,.

[24] Eigen, D, Fergus, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: IEEE International Conference on Computer Vision (ICCV). 2015,.

[25] Kendall, A, Martirosyan, H, Dasgupta, S, Henry, P, Kennedy, R, Bachrach, A, et al. End-to-end learning of geometry and context for deep stereo regression. In: IEEE International Conference on Computer Vision (ICCV). 2017,.

[26] Fu, H, Gong, M, Wang, C, Batmanghelich, K, Tao, D. Deep ordinal regression network for monocular depth estimation. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2018,.

[27] Xian, K, Shen, C, Cao, Z, Lu, H, Xiao, Y, Li, R, et al. Monocular relative depth perception with web stereo data supervision. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2018,.

[28] Liu, Y. Multi-scale spatio-temporal feature extraction and depth estimation from sequences by ordinal classification. Sensors 2020;20(7).

[29] Zoran, D, Isola, P, Krishnan, D, Freeman, WT. Learning ordinal relationships for mid-level vision. In: IEEE International Conference on Computer Vision (ICCV). 2015,.

[30] Chen, W, Fu, Z, Yang, D, Deng, J. Single-image depth perception in the wild. In: Adv. in Neural Inf. Proc. Systems (NIPS); vol. 29. Curran Associates, Inc.; 2016,.

[31] Liu, F, Shen, C, Lin, G, Reid, I. Learning depth from single monocular images using deep convolutional neural fields. IEEE Trans Pattern Anal Mach Intell 2016;38(10):2024–39.

[32] Chakrabarti, A, Shao, J, Shakhnarovich, G. Depth from a single image by harmonizing overcomplete local network predictions. In: Adv. in Neural Inf. Proc. Systems (NIPS); vol. 29. Curran Associates, Inc.; 2016,.

[33] Li, J, Klein, R, Yao, A. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In: IEEE International Conference on Computer Vision (ICCV). 2017,.

[34] Ranftl, R, Lasinger, K, Hafner, D, Schindler, K, Koltun, V. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Trans Pattern Anal Mach Intell 2022;44(3):1623–1637.

[35] Bhat, SF, Alhashim, I, Wonka, P. Adabins: Depth estimation using adaptive bins. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2021, p. 4009–4018.

[36] Ranftl, R, Bochkovskiy, A, Koltun, V. Vision transformers for dense prediction. In: IEEE International Conference on Computer Vision (ICCV). 2021, p. 12179–12188.

[37] Saxena, A, Sun, M, Ng, AY. Make3d: Learning 3d scene structure from a single still image. IEEE Trans Pattern Anal Mach Intell 2009;31(5):824–840.

[38] Karsch, K, Liu, C, Kang, SB. Depth transfer: Depth extraction from video using non-parametric sampling. IEEE Trans Pattern Anal Mach Intell 2014;36(11):2144–2158.

[39] Sener, O, Koltun, V. Multi-task learning as multi-objective optimization. In: Adv. in Neural Inf. Proc. Systems (NIPS); vol. 31. Curran Associates, Inc.; 2018,.

[40] Guizilini, V, Ambruș, R, Chen, D, Zakharov, S, Gaidon, A. Multi-frame self-supervised depth with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, p. 160–170.

[41] Zhou, Z, Dong, Q. Self-distilled feature aggregation for self-supervised monocular depth estimation. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I. Springer; 2022, p. 709–726.

[42] Li, Z, Dekel, T, Cole, F, Tucker, R, Snavely, N, Liu, C, et al. Learning the depths of moving people by watching frozen people. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2019,.

[43] Fattal, R. Single image dehazing. ACM Trans Graph 2008;27(3):1–9.

[44] Tal, I, Bekerman, Y, Mor, A, Knafo, L, Alon, J, Avidan, S. Nld-net++: A physics based single image dehazing network. In: Intl. Conf. on Computational Photography (ICCP). 2020, p. 1–10.

[45] Xu, L, Jia, J. Depth-aware motion deblurring. In: Intl. Conf. on Computational Photography (ICCP). 2012, p. 1–8.

[46] Bergman, AW, Lindell, DB, Wetzstein, G. Deep adaptive lidar: End-to-end optimization of sampling and depth completion at low sampling rates. In: Intl. Conf. on Computational Photography (ICCP). 2020, p. 1–11.

[47] Daribo, I, Pesquet-Popescu, B. Depth-aided image inpainting for novel view synthesis. In: 2010 IEEE International Workshop on Multimedia Signal Processing. 2010, p. 167–170.

[48] Hu, X, Fu, CW, Zhu, L, Heng, PA. Depth-attentional features for single-image rain removal. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2019,.

[49] Shi, J, Tao, X, Xu, L, Jia, J. Break ames room illusion: Depth from general single images. ACM Trans Graph 2015;34(6).

[50] Yang, Y, Lin, H, Yu, Z, Paris, S, Yu, J. Virtual dslr: High quality dynamic depth-of-field synthesis on mobile platforms. In: Digital Photography and Mobile Imaging. 2016,.

[51] Wadhwa, N, Garg, R, Jacobs, DE, Feldman, BE, Kanazawa, N, Carroll, R, et al. Synthetic depth-of-field with a single-camera mobile phone. ACM Trans Graph 2018;37(4).

[52] Wang, L, Shen, X, Zhang, J, Wang, O, Lin, Z, Hsieh, CY, et al. Deeplens: Shallow depth of field from a single image. 2018.

[53] Liao, M, Lu, F, Zhou, D, Zhang, S, Li, W, Yang, R. Dvi: Depth guided video inpainting for autonomous driving. In: European Conference on Computer Vision (ECCV). Cham: Springer International Publishing. ISBN 978-3-030-58589-1; 2020, p. 1–17.

[54] Shih, ML, Su, SY, Kopf, J, Huang, JB. 3d photography using context-aware layered depth inpainting. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2020,.

[55] Sheng, Y, Zhang, J, Benes, B. Ssn: Soft shadow network for image compositing. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2021, p. 4380–4390.

[56] Han, X, Zhang, Z, Du, D, Yang, M, Yu, J, Pan, P, et al. Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2019,.

[57] Zhao, C, Sun, Q, Zhang, C, Tang, Y, Qian, F. Monocular depth estimation based on deep learning: An overview. Science China Technological Sciences 2020;63(9):1612–1627.

[58] Silberman, N, Hoiem, D, Kohli, P, Fergus, R. Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision (ECCV). Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-33715-4; 2012, p. 746–760.

[59] Geiger, A, Lenz, P, Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2012, p. 3354–3361.

[60] Le, HA, Mensink, T, Das, P, Karaoglu, S, Gevers, T. Eden: Multimodal synthetic dataset of enclosed garden scenes. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). 2021, p. 1579–1589.

[61] Butler, DJ, Wulff, J, Stanley, GB, Black, MJ. A naturalistic open source movie for optical flow evaluation. In: European Conference on Computer Vision (ECCV). Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-33783-3; 2012, p. 611–625.

[62] Chen, W, Fu, Z, Yang, D, Deng, J. Single-image depth perception in the wild. In: Lee, D, Sugiyama, M, Luxburg, U, Guyon, I, Garnett, R, editors. Advances in Neural Information Processing Systems; vol. 29. Curran Associates, Inc.; 2016,.

[63] Sturm, J, Engelhard, N, Endres, F, Burgard, W, Cremers, D. A benchmark for the evaluation of rgb-d slam systems. In: Proc. of the International Conference on Intelligent Robot Systems (IROS). 2012,.

[64] Li, Z, Snavely, N. Megadepth: Learning single-view depth prediction from internet photos. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2018,.

[65] Schops, T, Schonberger, JL, Galliani, S, Sattler, T, Schindler, K, Pollefeys, M, et al. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: IEEE Computer Vision and Pattern Recognition (CVPR). 2017,.

[66] Brejcha, J, Čadík, M. Geopose3k: Mountain landscape dataset for camera pose estimation in outdoor environments. Image and Vision Computing 2017;66:1–14.

[67] Brejcha, J, Lukac, M, Hold-Geoffroy, Y, Wang, O, Cadik, M. Landscapear: Large scale outdoor augmented reality by matching photographs with terrain models using learned descriptors. In: European Conference on Computer Vision (ECCV). Springer International Publishing. ISBN 978-3-030-58526-6; 2020, p. 295–312.

[68] Bengio, Y, Louradour, J, Collobert, R, Weston, J. Curriculum learning. In: Proc. of the 26th Annual Intl. Conf. on Machine Learning. ICML '09; New York, NY, USA: Association for Computing Machinery. ISBN 9781605585161; 2009, p. 41–48.

[69] Kingma, DP, Ba, J. Adam: A method for stochastic optimization. 2017. arXiv:1412.6980.

[70] Reddi, SJ, Kale, S, Kumar, S. On the convergence of adam and beyond. arXiv preprint arXiv:190409237 2019;.

[71] Wang, C, Lucey, S, Perazzi, F, Wang, O. Web stereo video supervision for depth prediction from dynamic scenes. In: International Conference on 3D Vision (3DV). 2019, p. 348–357.

[72] Konneker, L. Resynthesizer plugin: Suite of gimp plugins for texture synthesis. Resynthesizer Plugin; 2022. URL: https://github.com/bootchk/resynthesizer.

[73] Zhou, QY, Park, J, Koltun, V. Open3D: A modern library for 3D data processing. 2018.