
Predicting High-precision Depth on Low-Precision Devices Using 2D Hilbert Curves

Mykhailo Uss^{1,2} Ruslan Yermolenko^{1,3} Oleksii Shashko^{1,4} Olena Kolodiazna⁵ Ivan Safonov¹
Volodymyr Savin^{1,5} Yoonjae Yeo⁶ Seowon Ji⁷ Jaeyun Jeong⁶

Abstract

Dense depth prediction deep neural networks (DNN) have achieved impressive results for both monocular and binocular data, but still they are limited by high computational complexity, restricting their use on low-end devices. For better on-device efficiency and hardware utilization, weights and activations of the DNN should be converted to low-bit precision. However, this precision is not sufficient to represent high dynamic range depth. In this paper, we aim to overcome this limitation and restore high-precision depth from low-bit precision predictions. To achieve this, we propose to represent high dynamic range depth as two low dynamic range components of a Hilbert curve, and to train the full-precision DNN to directly predict the latter. For on-device deployment, we use standard quantization methods and add a post-processing step that reconstructs depth from the Hilbert curve components predicted in low-bit precision. Extensive experiments demonstrate that our method increases the bit precision of predicted depth by up to three bits with little computational overhead. We also observed a positive side effect of quantization error reduction by up to 4.6 times. Our method enables effective and accurate depth prediction with DNN weights and activations quantized to eight-bit precision.

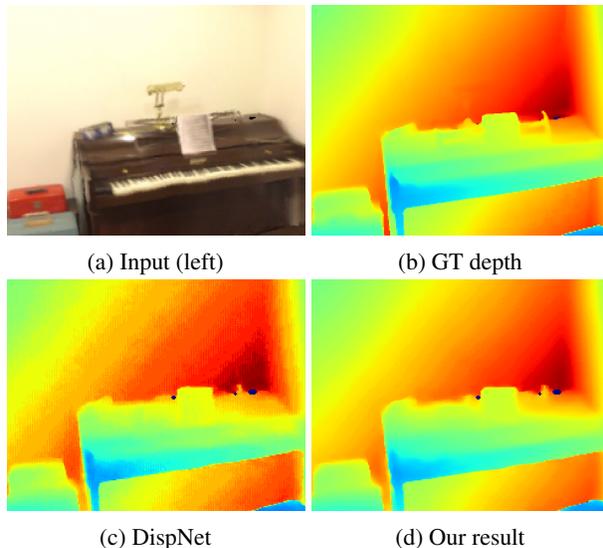


Figure 1. Illustration of DispNet (Mayer et al., 2016) quantization to INT8 precision (W8A8). Running inference of the quantized model on Qualcomm Hexagon DSP results in depth precision loss and quantization artifacts (c). Our method increases depth bit-width and reduces quantization error (d).

1. Introduction

Dense depth prediction has received considerable attention in recent years due to its wide-ranging applicability in such areas as scene understanding (Dai et al., 2017; Yeshwanth et al., 2023), autonomous driving (Wang et al., 2019b; You et al., 2020), robotics (Wofk et al., 2019), augmented reality / virtual reality (Rasla & Beyeler, 2022), Internet of Things (Ignatov et al., 2022), drones (Kolbeinsson & Mikolajczyk, 2024; Zhang et al., 2023). Usually, applications in these areas have strong limitations on hardware capabilities and energy consumption. Accurate depth estimation plays a vital role in enabling robust perception and decision-making processes in these contexts. A majority of recent developments in monocular (Ranftl et al., 2021; Bao et al., 2022; Oquab et al., 2024; Ke et al., 2023; Yang et al., 2024a;b; Shao et al., 2024) and binocular (Lipson et al., 2021; Xianqi et al., 2024) depth estimation employ large-scale frameworks leveraging complex convolutional neural networks

¹Samsung R&D Institute Ukraine, Kyiv 01032, Ukraine
²Department of Information-Communication Technologies, National Aerospace University, Kharkiv 61070, Ukraine
³Faculty of Physics, Taras Shevchenko National University of Kyiv, Kyiv 01032, Ukraine
⁴Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, Kharkiv 61070, Ukraine
⁵Institute of Physics and Technology, NTUU "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv 01032, Ukraine
⁶Samsung Research, Seoul 06765, Republic of Korea
⁷Department of Computer Science and Engineering, Konkuk University, Seoul, Republic of Korea.
Correspondence to: Mykhailo Uss <m.uss@samsung.com>.

(including Convolutional Gated Recurrent Units) (Nicolas et al., 2015; Lipson et al., 2021; Xianqi et al., 2024), transformer (Ranftl et al., 2021; Bao et al., 2022; Oquab et al., 2024) and diffusion (Ke et al., 2023; Yang et al., 2024a;b; Shao et al., 2024) architectures. Furthermore, this trend stems from attempts to develop foundational models to solve the task. However, the impressive progress made so far provokes a significant increase in the complexity of the proposed solutions to achieve better accuracy, which typically demands considerable computational and memory resources. Hardware limitations cause additional difficulties with on-device deployment of these complex models.

The efficiency of DNN inference on low-end devices can be achieved by applying several strategies. One is to optimize a model architecture to reduce number of parameters or latency, avoid using computationally intensive layers, use network pruning (Wofk et al., 2019; Oh et al., 2020; Liu & Zhou, 2023; Song et al., 2023). Another strategy consists in usage of low-precision computations (Li et al., 2021; Jacob et al., 2018). Low-bit fixed-point representation of DNN weights and activations, such as INT8, increases the number of operations run in parallel, reduces the amount of data transferred to and from memory, and decreases the energy consumption of Multiply-Accumulate (MAC) operation (Nagel et al., 2021). When moving to lower bit-width precision, the memory amount reduces linearly, and the computational cost of matrix multiplications reduces quadratically (Nagel et al., 2021). Full-precision models are usually trained in FP32 precision and quantized to low-bit precision for on-device deployment. Quantization-aware training (QAT) emulates the quantization process during full-precision model training (Yang et al., 2019; Sakr et al., 2022; Bhalgat et al., 2020; Li et al., 2022); post-training quantization (PTQ) works on trained models and seeks to convert weights into integer values with minimum quality degradation (Fang et al., 2020; Yuan et al., 2022).

Running inference of quantized models comes at the cost of additional low-precision computation errors and data precision loss (Nagel et al., 2021). Considerable research has been devoted towards reducing quantization error by DNN’s architecture modification (Sheng et al., 2018; Wang et al., 2020), quantization techniques improvement (Sakr et al., 2022; Bhalgat et al., 2020; Li et al., 2022; Nagel et al., 2021; Fang et al., 2020; Yuan et al., 2022), adaptation to specific hardware (Shlezinger et al., 2019; Kiyama et al., 2019; Wang et al., 2019a). The problem of data precision loss is largely unexplored, presumably because it does not pose a problem for models predicting RGB images, which are naturally presented as three eight-bit channels. Nevertheless, precision loss becomes an issue for High Dynamic Range images. This prevents the use of efficient homogeneous eight-bit quantization and forces to resort to less efficient mixed quantization schemes (Borrego-Carazo et al., 2021).

Depth maps require a high bit-width for accurate representation. For example, representing depth in the range $0 \dots 10$ m with 1 cm accuracy requires ten bits. Depth represented in eight bits suffers from the false edges on planar surfaces (Fig. 1c), loss of spatial details and distortion of objects with low depth contrast. Consequently, low-end devices with low-precision computations necessarily lead to on-device depth quality loss. To address this issue, in (Jiang et al., 2022) authors suggested keeping the final layer of the depth completion model at full floating-point precision while quantizing the weights and activations of other layers to either four-bit or eight-bit precision. This solution is not efficient because the last layer works in the highest spatial resolution and is computationally intensive. Low-end devices not always support mixed-precision quantization or different bit-width for different model layers. For example, SNPE library (Qualcomm Technologies, Inc., 2025) for converting DNNs for inference on Qualcomm Hexagon Digital Signal Processor (DSP) supports the same bit-width for all layers (can be different for weights and for activations). The Coral Edge Tensor Processing Unit (TPU) supports only INT8 or UINT8 data. All operations with float-precision data are executed on CPU (Google, 2020). It means that the depth prediction model on devices with low-precision arithmetic will suffer from depth precision loss independently of the quantization quality. This problem relates to the hardware limitations and cannot be solved by improving the quality of DNN quantization with PTQ/QAT.

In this work, we propose to overcome this limitation by using depth representation as points on a 2D Hilbert curve. This transform codes high dynamic range depth as two Hilbert curve components that can be represented in low-bit precision with minimum quality loss (Fig. 1). Our method adds an on-device post-processing step that converts low-bit precision Hilbert components to higher precision depth. The post-processing does not involve addition or multiplication operations and can be implemented as a simple lookup table (LUT). The main difference of our method is that for improving quality of DNN inference on device we consider not only DNN architecture modification, training process (like QAT), efficient weights and activations quantization (like PTQ) but also structure of the signal predicted by the DNN. Our method does not modify quantization process by itself and is built on top of existing quantization methods (either PTQ or QAT); progress of these methods in specific domains or for specific architectures will bring improvement in the performance of our method as well.

Our main contributions can be summarized as follows:

- (a) We propose a novel method for high-quality depth prediction on devices with low-precision arithmetic. The method consists of representing depth as 2D Hilbert curve components, training a full-precision model to predict 2D Hilbert

curve components, and reconstructing high-precision depth from low-precision Hilbert curve components.

(b) We evaluate the proposed method and demonstrate that the modified model quantized to W8A8 format (8-bit weights and activations) can predict depth with similar or better quality than the original model in W8A16 format (8-bit weights and 16-bit activations). At the same time, the inference time and power consumption of the modified W8A8 model with post-processing are 1.5 times less than that of the original W8A16 model.

(c) We observe that the application of the proposed method has additional positive side effect on the quantization quality, specifically quantization error reduction by up to 4.6 times.

2. Method

In this section, we introduce our method that allows high-precision depth prediction on devices with low-precision computations. The method rests on depth representation as two components of a 2D Hilbert curve. On-device, these components are predicted in low-bit precision on TPU or NPU and then converted to high-bit precision depth using a simple post-processing algorithm run on CPU. In the following subsections, we describe the method in general (Sec. 2.1), discuss properties of the Hilbert curve and its utility for depth coding (Sec. 2.2), explain implementation details (Sec. 2.3), discuss expected change of quantization error (Sec. 2.4). Lastly, we present the loss function for training the modified model (Sec. 2.5).

2.1. High-precision depth on a low-precision device

Let us consider a DNN that predicts some quantity q bounded in the range $[0, 1]$. A quantized model runs on a low-end device that has an efficient DNN inferencing module (in our case, it is DSP) with b -bit output data representation and a general-purpose CPU with full-precision arithmetic. The DSP outputs value $q_{\text{quant},b}$ in b -bit precision (e.g. INT8). The error between q and $q_{\text{quant},b}$ is called the quantization error. We denote the standard deviation (SD) of this error as σ_{quant} .

We seek to increase the bit-width of the predicted value of q with minimum computational overhead. For the depth prediction case, q corresponds to normalized depth or normalized disparity, depending on the DNN design.

Due to DSP limitations, the bit-width of the model output cannot be increased, so we can operate only with the number and structure of the output channels. These channels should be transferred from DSP to CPU and used to reconstruct q in higher bit precision. To make this scheme effective, the following key problems need to be solved: (1) minimize the amount of data transferred from DSP to CPU, (2) minimize

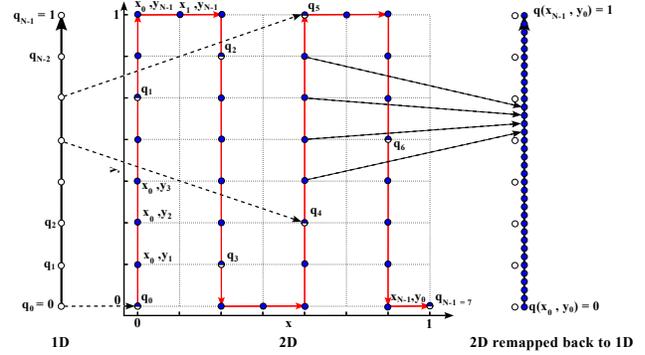


Figure 2. Idea illustration. 1D range is quantized to $N = 8$ values $q_0 = 0 \dots q_{N-1} = 1$ marked by white circles. The 1D range is mapped to a 2D curve shown in red color. Both x and y axes are also quantized into $N = 8$ values yielding 64 2D points. Among them, 36 points lie on the curve (shown in blue color). Mapping the 2D curve back to the 1D range results in 36 different quantization values. Quantization error has effectively been reduced by the factor equal to the curve length $L = 35/7 = 5$.

the complexity of post-processing run on CPU. In the solution proposed in (Jiang et al., 2022), the last layer of the depth completion model is in float precision. This version is ineffective because it requires passing to the CPU a large amount of data from the penultimate layer and processing the high-resolution data associated with the last layer.

Our idea is to represent q as a point on a 2D parametric curve $(x(q), y(q))$ with length $L > 1$, where both $x(q)$ and $y(q)$ are bounded in the $[0, 1]$ range. The full-precision DNN is trained to directly predict $x(q)$ and $y(q)$. When running inference on DSP, the value of q is calculated from x and y values predicted in b -bit precision. The parametric curve of length L will pass through approximately $L \cdot 2^b$ discrete points $(x(q), y(q))$ effectively increasing precision of reconstructed value of q by $\log_2 L$ bits from b to $b + \log_2 L$ (Fig. 2). In this implementation, the amount of data to transfer from DSP to CPU increases only twofold, post-processing on CPU is simple and implemented using LUT. We will later refer to unmodified DNN as original model and to model predicting Hilbert components (and post-processing step when it is clear from the context) as modified model.

On-device deployment of a depth prediction DNN consists of the following steps (Fig. 3): (a) training the full-precision model to directly predict the Hilbert curve components of depth representation, (b) applying standard quantization methods (either PTQ or QAT), (c) running inference of the modified quantized model on-device and obtaining Hilbert curve components in low-bit precision, (d) applying post-processing to Hilbert curve components and reconstruct depth in higher-bit precision.

Selection of function $(x(q), y(q))$ is crucial for successful implementation of the proposed idea.

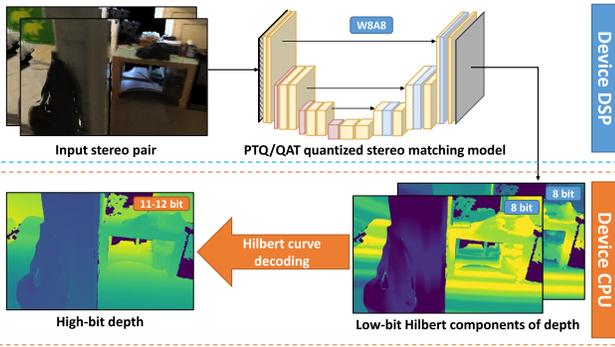


Figure 3. Scheme of the proposed method’s inference pipeline on device.

2.2. Selecting a suitable parametric curve

Let us consider the simplest function: $x(q) = \lfloor q * 255 \rfloor / 255$, $y(q) = q - x(q)$, where $\lfloor \cdot \rfloor$ denotes the floor function. In this case, $x(q)$ is a coarse representation of q and $y(q)$ adds fine details (see example in Fig. 4e and 4f). The length of this curve is 256, and it allows reconstructing q in INT16 precision from $x(q)$ and $y(q)$ in INT8 precision. However, structure of $y(q)$ is unfriendly for the full-precision model training: it has multiple sharp transitions from zero to one and from one to zero. During our experiments we found that this curve degrades full-precision model training and is very susceptible to quantization errors.

Let us discuss the desired properties of the curve $(x(q), y(q))$:

1. Continuity: small changes in q should result in small changes in both $x(q)$ and $y(q)$. For depth coding, this property ensures that $x(q)$ and $y(q)$ preserve spatial smoothness and do not introduce new depth discontinuities.
2. Non-self-intersection: the curve should preserve one-to-one correspondence between q and $(x(q), y(q))$.
3. Boundedness and self-avoidance: the curve should cover the unit square uniformly and avoid close points $(x(q_1), y(q_1))$ and $(x(q_2), y(q_2))$ for distant q_1 and q_2 .

Curves with the desired properties are known as space-filling curves or Peano curves (Sagan, 2012). For the goals of this work, we found one particular curve, namely the Hilbert curve (Sagan, 2012; Bader, 2012), to be the simplest and the most flexible option. We describe this choice in more detail in Appendix A.

The Hilbert curve is a continuous fractal space-filling curve that is constructed as a limit of piece-wise linear curves (Bader, 2012). The Hilbert curve starts from a single point in the middle of the unit square. Each subsequent curve order is produced by replicating and linking

points of the curve of the previous order. We will later refer to the curve order as p . The approximating polygons for curves with orders 1 – 4 are shown in Fig. 5. In order to avoid boundary effects, we scale each curve so that it fits into the square $\{(x, y) | b \leq x, y \leq 1 - b\}$, where $b = 0.1$. In this case, the length of the p -th order curve is $L_p = (2^p + 1)(1 - 2b)$. The length of an edge of p -th order approximation polygon equals $h_p = (1 - 2b)/(2^p - 1)$. This value also defines the minimum distance between points of different parallel edges of the Hilbert curve approximation polygon (Fig. 5).

2.3. Forward and backward transformations

For a full-precision and quantized models, predicted values (x, y) will not have an exact match with any $(x(q), y(q))$. To convert arbitrary point (x, y) back to 1D value, we link (x, y) to the closest point on the Hilbert curve:

$$q_{xy} = \operatorname{argmin}_{q \in [0,1]} \|(x - x(q), y - y(q))\|. \quad (1)$$

Building forward (1D \rightarrow 2D) and backward (2D \rightarrow 1D) mappings for Hilbert curves of arbitrary order is based on iterative algorithms (Bader, 2012). Because we work with one specific curve order, a faster transformation can be implemented with LUTs. We build two LUTs for corresponding mappings. The first LUT is built using bilinear interpolation of the nodes of the low-order Hilbert curve. This LUT allows us to get $(x(q), y(q))$ for the given q . An example of this transformation applied to the ground truth (GT) disparity map is shown in Fig. 4a-4b. To map 2D values (x, y) (Fig. 4c-4d) back to the 1D representation, we use a second LUT that is built based on Eq. (1). In our experiments, this LUT has a size of 256 by 256 elements.

2.4. Quantization error transformation

For the original model, the quantization error is measured at the model output level. For our method, this error is measured after post-processing of Hilbert components and accounts for Hilbert components quantization error and its transformation during post-processing. We denote SD of this error as $\bar{\sigma}_{\text{quant}}$ and SD of Hilbert components quantization error as $\sigma_{\text{xy.quant}}$.

Locally, Eq. (1) reduces to the projection of (x, y) to the closest Hilbert curve edge, discarding x or y component (depending on the edge orientation), compressing the remaining component L times. We discussed above that this transform adds $\log_2 L$ bit to q precision. In addition, Hilbert components quantization error is compressed L times yielding $\bar{\sigma}_{\text{quant}} = \sigma_{\text{xy.quant}}/L$.

Quantization error compression takes place if (a) Hilbert components quantization error is independent between channels and identically distributed, and (b) if $\sigma_{\text{xy.quant}}$ is small

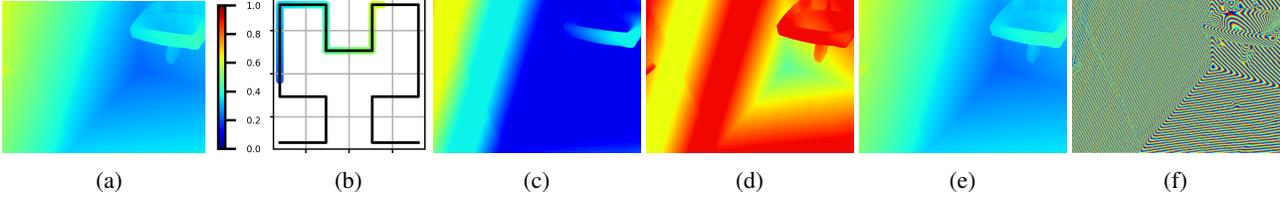


Figure 4. Illustration of disparity transforms: (a) disparity map; (b) mapping to 2D with second order Hilbert curve; (c, d) x and y components of the Hilbert curve; (e, f) coarse and fine details of disparity map. Fine details in (f) are the least significant byte of disparity (a) represented in 16-bit format. High-frequency oscillations make it appear different from the original disparity and difficult to predict by a DNN model.

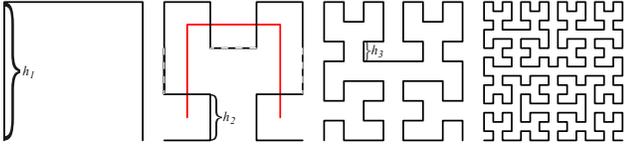


Figure 5. Hilbert curves for orders $p = 1, 2, 3, 4$ (from left to right). Every order is formed by the replacement of every node by an elementary 3-segment sequence.

enough for Hilbert components quantization error to be smaller than h_p . Both original and modified models have similar architecture, solve the same task, are trained on the same data, quantized by the same procedure. We can additionally assume (c) that $\sigma_{\text{quant}} = \sigma_{xy.\text{quant}}$. If this additional condition is valid then our method will reduce quantization error by L times compared to the original model: $\bar{\sigma}_{\text{quant}} = \sigma_{\text{quant}}/L$.

We satisfy condition (b) by proper selection of the Hilbert curve order. For the stereo matching, we experimentally found that $p = 2, 3$ are suitable choices, providing curve length 4–7.2 and bit-width increase by 2–2.85. Assumptions (a) and (c) can only be verified experimentally for a specific DNN architecture. We discuss their validity in detail in Section 3.5. Notice, that bit-width increase will take place even if conditions (a) and (c) are not fulfilled. These additional conditions are only needed for potential quantization error reduction.

2.5. Model and Loss Function Modification

To implement the proposed approach, a DNN with one head predicting quantity q should be modified to have two heads predicting Hilbert curve components x and y . This modification for the DispNet model (Mayer et al., 2016) is illustrated in Fig. 6.

The loss function for the proposed method is composed of two terms: depth loss $\Lambda(q_{\text{GT}}, q_{xy})$ (Hu et al., 2019) and additional term $\Lambda_H(x_{\text{GT}}, y_{\text{GT}}, x, y)$ that assures model con-

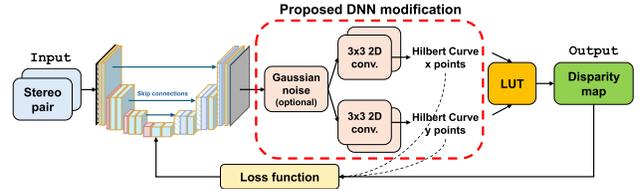


Figure 6. DispNet (Mayer et al., 2016) model modification required by the proposed approach. The input RGB stereo pair is processed by the encoder-decoder network from the original model. Features from the encoder-decoder are fed to the optional Gaussian noise layer, followed by two identical heads for Hilbert curve components. At the post-processing stage, Hilbert components are converted to the final disparity map.

vergence to the Hilbert curve-based representation:

$$\Lambda_{full} = \Lambda(q_{\text{GT}}, q_{xy}) + \alpha \cdot \Lambda_H(x_{\text{GT}}, y_{\text{GT}}, x, y), \quad (2)$$

where x_{GT} and y_{GT} are GT values for Hilbert curve components calculated from GT value q_{GT} , and α is hyperparameter. The additional component Λ_H is calculated as follows:

$$\Lambda_H(x_{\text{GT}}, y_{\text{GT}}, x, y) = (x_{\text{GT}} - x)^2 + (y_{\text{GT}} - y)^2 + \beta \cdot r_{xy}^2, \quad (3)$$

where $r_{xy} = \|(x - x(q_{xy}), y - y(q_{xy}))\|$ and β is additional hyperparameter. The Hilbert curve loss component serves two goals: it penalizes the distance between GT and predicted points in the 2D representation; it penalizes deviation across the Hilbert curve and forces the model to predict only points that belong to the curve. We set $\alpha = 1$ and $\beta = 25$ in our experiments.

3. Experiments

Two models are selected for the stereo matching experiment: DispNet with the original architecture proposed in (Mayer et al., 2016) and Dense Prediction Transformer (DPT) (Ranftl et al., 2021) with MobileViTv3-S (Wadekar & Chaurasia, 2022) as an encoder. In all experiments, the models' input size is 384×512 pixels and the output size is 192×256 pixels. All modifications of the DPT model are described in detail in Appendix B. The head predicting Hilbert curve

components for the DPT model is the same as for the DispNet but includes an additional convolutional and up-sample layer in each branch to adapt to feature shapes of the DPT decoder. We found that injecting small amount of Gaussian noise at the beginning of the Hilbert components head improves quantization of modified models with the SNPE library and has no effect on unmodified models. Experimental results for modified models are presented with a Gaussian noise layer with SD equals 0.02.

3.1. Implementation Details

For stereo matching models training we adapted ScanNet v2 (Dai et al., 2017) dataset in the following way. Training, validation, and test data are rendered from meshes provided for each ScanNet v2 scene using PyRender v.0.1.45 library. The camera poses for the left camera are fixed to the same values as specified in ScanNet v2. The right camera is shifted by 60 mm along the axis x to form the horizontal baseline. Intrinsic parameters for the left and right cameras correspond to ScanNet data: pinhole camera with $f_x = f_y = 577.87$, $c_x = 320$, $c_y = 240$. The split of the dataset into training and test parts corresponds to the official ScanNet v2 split.

All models were quantized using SNPE SDK v.2.24 and tested on Samsung S24+ device with Qualcomm Snapdragon 8 Gen 3 processor and Hexagon DSP. We compare models in FP16, W8A16 and W8A8 formats running on Hexagon DSP. Power consumption is measured with Monsoon Solutions FTA22D Power Monitor in power save mode.

3.2. Evaluation Metrics

We characterize the quality of predicted depth using standard metrics (Eigen et al., 2014; Geiger et al., 2012): mean absolute relative error (Abs Rel), root mean square error (RMSE), end-point-error (EPE), and D1.

Pixel-level errors themselves are not sufficient to characterize quantization artifacts in the predicted depth maps. For example, INT8 depth representation has minimal impact on the Abs Rel metric. To address this issue, we experimented with the SSIM metric (Wang et al., 2005; Rouse & Hemami, 2008). However, we found it barely affected by quantization artifacts. Therefore, we propose using cosine similarity (Lahitani et al., 2016) between discrete cosine transform (DCT) (Ahmed et al., 1974) coefficients of GT and predicted depth maps. For this, $n \times n$ DCT is applied in a scanning window manner to both GT and predicted depth maps. The DCT coefficients are flattened to vector representations and zero coefficients are discarded. Cosine similarity between flattened vectors is calculated at each

scanning window position and then averaged:

$$S_C = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M \frac{\mathbf{c}_{ij} \cdot \hat{\mathbf{c}}_{ij}}{\|\mathbf{c}_{ij}\| \cdot \|\hat{\mathbf{c}}_{ij}\|}, \quad (4)$$

where N is the number of frames, M is the number of scanning windows, \mathbf{c}_{ij} and $\hat{\mathbf{c}}_{ij}$ are vectors of DCT coefficients of GT and predicted depth maps for frame i and window j . Experiments show that S_C calculated in 4×4 window is sensitive to depth blurring and INT8 quantization artifacts. The value of S_C close to unity (the maximum possible value) indicates high-quality depth maps with sharp edges and absence of artifacts in homogeneous areas.

To characterize quantization error, we use SD $\hat{\sigma}$ between raw outputs of full-precision and quantized models measured in a robust way as Scaled Median Absolute Deviation (Rousseeuw & Croux, 1993).

3.3. Analysis of W8A8 Models

We trained DispNet and DPT models using $p = 1, 2, 3, 4$. These models will be later referred to as hpDispNet and hpDPT, respectively. We observed that float-precision models hpDispNet and hpDPT learn to predict points close to the Hilbert curve. Depth prediction accuracy of modified models is similar to the original models w.r.t. all metrics.

Next, we compare the original and modified models in W8A8 format running on CPU and DSP. The difference is that W8A8 model runs on CPU in dequantization mode using full precision arithmetic, while on DSP inference is done in lower-precision arithmetic. Quantitative results are presented in Table 1. For the original DispNet model, quantization leads to noticeable quality degradation on both CPU and DSP. On CPU degradation is seen for S_c metric that drops from 0.86 to 0.68 reflecting loss of spatial details.

Modified models for all p perform better than the original ones; the best result achieved for $p = 3$. As shown in Fig. 7, the quantized h3DispNet model retains the ability to predict points on the Hilbert curve for both CPU and DSP inference. The h3DispNet model on CPU shows almost the same quality as the FP32 model and outperforms the original DispNet w.r.t. all metrics. On DSP, the quality drop of the original DispNet is more significant: Abs Rel increases from 1.01 to 2.03, and S_c decreases from 0.86 to 0.58. The h3DispNet model compensates this drop almost completely reaching Abs Rel 0.93 and S_c 0.81.

For the DPT model, the situation is similar, but the quality drop for the quantized model is more significant on both CPU and DSP. On CPU, the h3DPT model performs better than the original FP32 DPT model. On DSP the best result shows h2DPT model with Abs Rel improving from 4.18% to 1.12% and S_c increasing from 0.52 to 0.62 as compared to the original model. The curve of the third order provides

Model	Abs Rel, % ↓			RMSE, px ↓			S_C ↑			EPE, px ↓			D1, % ↓		
	FP32	W8A8	W8A8	FP32	W8A8	W8A8	FP32	W8A8	W8A8	FP32	W8A8	W8A8	FP32	W8A8	W8A8
	CPU	DSP	CPU	CPU	DSP	CPU	CPU	DSP	CPU	CPU	DSP	CPU	CPU	DSP	CPU
DispNet	1.01	2.03	1.15	1.12	2.24	1.10	0.86	0.58	0.68	0.29	0.69	0.31	1.81	5.35	1.73
h1DispNet	1.06	1.50	1.12	0.97	1.09	0.97	0.86	0.67	0.79	0.27	0.35	0.29	1.27	2.25	1.27
h2DispNet	0.85	0.98	0.88	0.90	0.94	0.91	0.87	0.75	0.83	0.22	0.25	0.23	1.01	1.26	1.02
h3DispNet	0.88	0.93	0.87	1.00	1.03	1.00	0.87	0.81	0.86	0.24	0.24	0.24	1.25	1.26	1.25
h4DispNet	0.90	0.94	0.92	1.02	1.02	1.02	0.85	0.83	0.85	0.24	0.25	0.24	1.24	1.25	1.24
DPT	0.75	4.18	1.48	0.87	2.09	1.03	0.89	0.52	0.87	0.21	1.03	0.39	0.95	5.78	1.39
h1DPT	0.70	1.48	0.78	0.88	1.41	0.89	0.88	0.54	0.88	0.20	0.41	0.22	1.01	2.77	1.03
h2DPT	0.71	1.12	0.72	0.91	1.02	0.91	0.88	0.62	0.88	0.20	0.29	0.20	1.07	1.27	1.08
h3DPT	0.55	1.35	0.63	0.80	1.33	0.80	0.90	0.70	0.90	0.15	0.32	0.17	0.78	1.28	0.78
h4DPT	0.74	1.27	0.76	0.94	1.38	0.94	0.87	0.73	0.86	0.21	0.32	0.21	1.14	1.62	1.14

Table 1. Metrics of DispNet, hpDispNet, DPT and hpDPT models. All metrics are presented for FP32 model (CPU inference), W8A8 model running on DSP and W8A8 model running on CPU (dequantization mode). The best results on DSP are in bold font. Results for the original DispNet and DPT models are marked in grey color.

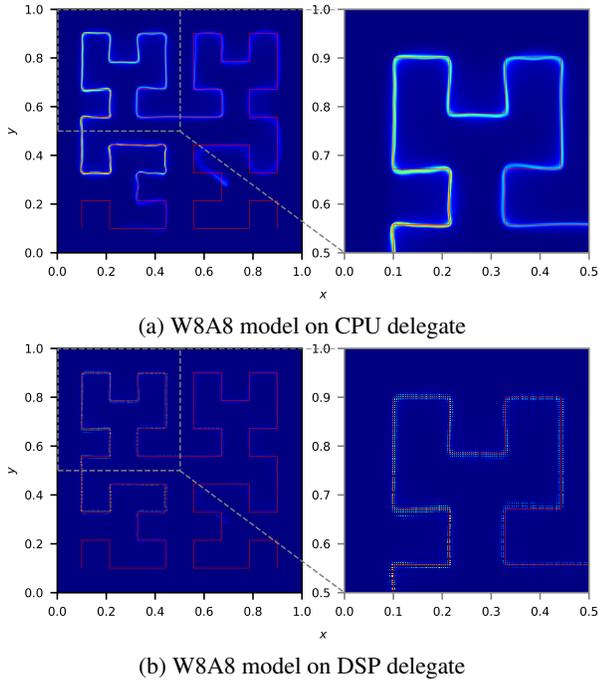


Figure 7. 2D histogram of h3DispNet W8A8 model output for CPU and DSP delegates.

the best compromise between Abs Rel and S_C improvement for both DispNet and DPT models. This conclusion holds when RMSE, EPE or D1 are taken into account.

Qualitative results for the h3DPT model are illustrated in Fig. 8. Reduction of quantization error between the original model (Fig. 8b) and the modified model (Fig. 8c) is very significant as can be seen on error maps in Fig. 8e and Fig. 8f. We also observe from Fig. 8c and Fig. 8b that h3DPT demonstrates better spatial details than the original model. This effect is caused by increase of effective number of bits for depth map coding by $\log_2 L$ bits (approximately by 2.85 bits for h3DPT resulting in INT10-INT11 precision). Indirectly, this effect is characterized by increase of S_C value with p (see Table 1). Visually, bit-precision increase is well

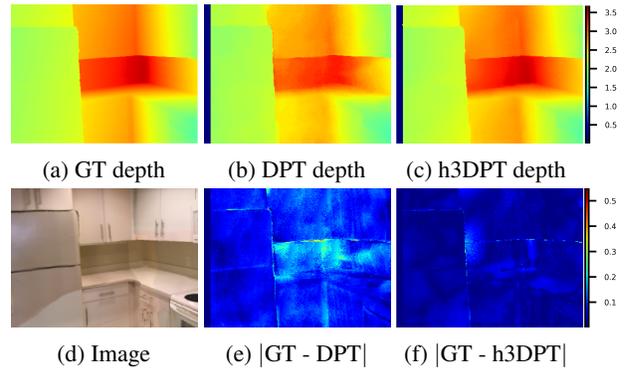


Figure 8. Quantization errors of DPT and h3DPT, W8A8, DSP.

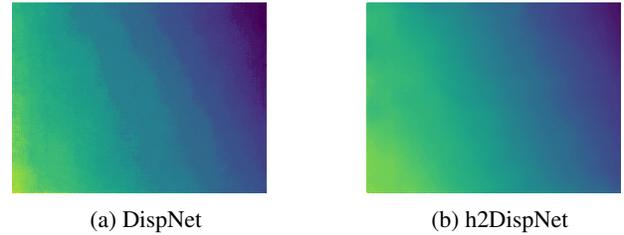


Figure 9. Effect of increased bit-width on the quality of depth prediction for homogeneous areas on DSP.

seen for homogeneous planar areas as illustrated in Fig. 9.

As one can see from these results, the proposed method not only solves the main problem of bit-width increase but also significantly reduces the level of quantization error of depth prediction. The latter is a positive side effect and we will study it more in detail in Section 3.5. Effect of the improved quality of depth prediction on scene mesh fusion is discussed in Appendix D and additional examples of predicted depth maps are given in Appendix E. Additional experiment on KITTI 2012 dataset is described in Appendix C.

Predicting High-precision Depth on Low-Precision Devices Using 2D Hilbert Curves

Precision	Abs Rel, %	EPE, px	D1, %	S_C	T , ms	P , mW·s/infr.
DispNet						
FP32	1.01	0.29	1.81	0.858	-	-
FP16	1.50	0.37	1.80	0.855	19.54	19.52
W8A16	1.78	0.63	5.22	0.798	18.7	12.3
W8A8	2.02	0.69	5.34	0.585	10.5	7.1
Ours, W8A8	0.93	0.24	1.26	0.807	12.0	8.7
DPT						
FP32	0.75	0.21	0.95	0.889	-	-
FP16	1.14	0.27	0.97	0.884	54.1	110.5
W8A16	4.03	0.97	5.58	0.825	46.2	64.5
W8A8	4.16	1.03	5.76	0.520	26.7	28.3
Ours, W8A8	1.35	0.32	1.28	0.697	30.4	29.7

Table 2. On-device performance of the original and modified models including runtime (T) and power consumption (P). Measurements for our method include data transfer from DSP to CPU and Hilbert components post-processing on CPU. The overhead of our method is in runtime and power consumption increase between modified and original W8A8 models (compare lines marked by orange and green colors). Models in FP32 format are run on CPU; models in FP16, W8A16, W8A8 formats are run on DSP.

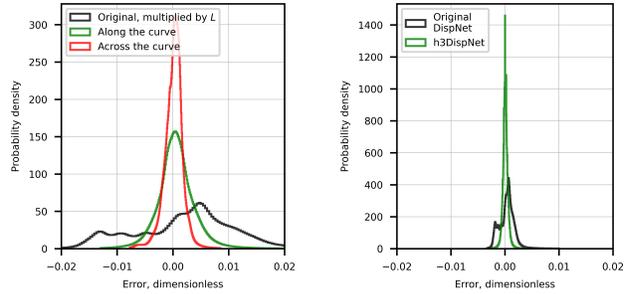
3.4. Comparison of FP16, W8A16 and W8A8 Models

Compared to the original model, the overhead of our method consists of three parts: (1) increased model complexity, (2) additional data transfer from DSP to CPU, (3) post-processing of Hilbert components. We found that this overhead ($\approx 14\%$) is significantly smaller than the performance gain from using W8A8 models instead of W8A16 or FP16 models. Detailed profiling results are presented in Table 2 for models with the curve order 3.

The modified DispNet model in W8A8 format shows better quality than the original model in W8A16 format while simultaneously reducing energy consumption by 35% and latency by 30%. Compared to the original model in W8A16 format, the modified DPT model in W8A8 format shows significantly better Abs Rel metric and only slightly worse S_C . It also lowers energy demands by 34% and processing time by 54%.

3.5. Analysis of Quantization Errors Reduction

For the detailed analysis of quantization error reduction, we selected the h3DispNet model. To exclude precision loss influence, we perform analysis for W8A8 model run on CPU. The difference between the original and modified model lies in the number of channels utilized: the relationship between the quantization errors among two Hilbert components requires further examination. Our initial analysis in Section 2.4 was made under the assumption of independent quantization errors for the Hilbert components. However, assessment using real data demonstrated that this hypothesis does not hold universally, as correlation has been discovered along the Hilbert curve. Quantitatively, the distribution



(a) Distributions of errors across and along the Hilbert curve. (b) Disparity errors distribution for h3DispNet and DispNet models.

Figure 10. Quantization errors distributions for the h3DispNet model. Disparities d are calculated from Hilbert components x and y and normalized to $[0, 1]$ range.

of quantization errors along-the-curve is wider than that across-the-curve (Fig. 10a).

Across-the-curve errors are mainly nullified in the post-processing, along-the-curve errors are compressed L times and define the level of errors of the target quantity, that is disparity. To compare errors in the same scale, we added in Fig. 10a disparity error of the original model multiplied by the curve length L . Quantization error in Hilbert curve-based output representation is significantly smaller than expected if the original model errors are scaled L times. As a result, in disparity space, the quantization error of the modified model is significantly smaller than that of the original one (see Fig. 10b): $\hat{\sigma}$ is $5.65 \cdot 10^{-4}$ and $17.66 \cdot 10^{-4}$, respectively. Thus, we obtained quantization error reduction by ≈ 3.1 times on CPU. On DSP this gain is ≈ 4.6 times. Similar effect is observed for the DPT.

In order to bring further understanding, we applied Hilbert curve-based output representation to the Human Pose Estimation (HPE) task (see Appendix F). For the ResNet-RS (Bello et al., 2021) encoder and keypoints representation as heatmaps, the effect is similar to stereo matching but with a stronger along-the-curve correlation. Quantization error reduction on DSP is 2.69 times for $p = 3$ curve. For direct keypoints regression along-the-curve correlation is so strong that quantization error reduction is not observed. For both stereo matching and HPE, we found that the magnitude of the across-the-curve quantization error is similar to the quantization error of the original model.

4. Discussion and Limitations

The proposed method is designed with the main goal of increasing the bit-precision of the predicted depth on devices with low-bit precision output. Apart from this, we observed a positive side effect consisting of a significant reduction of the quantization error. The effect reveals itself

in different degrees for the depth prediction and human pose estimation with heatmaps; it is not observed for the human pose estimation with direct keypoints regression.

Our experiments show that the quantization error of Hilbert components can be correlated and have a large SD along the Hilbert curve. This effect manifests itself strongly for the human pose estimation with direct regression and only slightly for the stereo matching models. Such a correlation indicates that instead of learning independent paths for predicting Hilbert curve components, the modified model learns an internal representation of the target quantity (for example, disparity, depth or keypoints) and converts it into Hilbert components at the final model layers. In this case, the modified model predicts the target quantity with the same quantization error as the original model, this error undergoes forward transform to Hilbert representation (inside the model) and backward transform to the target quantity (at the post-processing step). As a result, the quantization error level remains the same, and the effect of our method application is only in bit-width precision increase. This is the case for the HPE with direct regression.

If both x and y are predicted by independent paths inside the model, the quantization errors of the Hilbert components are not correlated and have similar level as for the original model. This error is compressed L times at the Hilbert components post-processing stage. As a result, our method allows both reducing quantization error and increasing bit-width precision as has been manifested by our stereo matching experiment. If the source of correlation in the along-the-curve direction can be found and eliminated, the proposed method could be used to reduce quantization error in cases when bit-precision is not an issue.

While our method improves the quality of predicted depth, it requires retraining of the full-precision model. In addition, our method requires that errors of the quantized depth prediction model are bounded, since large values of the quantization error violate one-to-one correspondence between the depth values and the points on the Hilbert curve. The next step involves joint usage of QAT and our method and moving to higher dimensions where a multidimensional Hilbert curve of the same length fills the space less densely.

We present results for stereo-matching models that predict disparity in one step. The recent models showing the best quality for stereo matching (Xu et al., 2023; Xianqi et al., 2024) and monocular depth prediction (Shao et al., 2023) incorporate iterative disparity refinement using GRU or similar recurrent units. In these architectures, the predicted disparity is refined over multiple iterations by adding a small correction signal to the initially predicted disparity map. Integration of the proposed idea of disparity representation as two Hilbert curve components into iterative models differs significantly from one-stage models like DispNet and DPT.

While we experimented with modification of the output disparity representation, modification of iterative models requires integration of Hilbert components inside the model at multiple places. We leave this interesting problem for future work.

5. Conclusions

In this paper, we proposed a novel method for high dynamic range depth prediction on devices with low-precision arithmetic that exploits depth representation as points on a 2D Hilbert curve. This representation essentially codes the high dynamic range depth as two low dynamic range Hilbert curve components. The depth prediction model is trained to directly predict two Hilbert curve components that are calculated on-device in low-bit precision and used to reconstruct depth in high-bit precision. Apart from increasing bit-precision, our method reduces quantization error by a factor of up to 4.6. Experiments demonstrate that for the stereo matching task our method reconstructs depth in INT10-INT11 bits for a model quantized in W8A8 format and with quality similar to or even better than the original model quantized in W8A16 format. In this manner, depth can be predicted on-device 1.4-2 times faster and with 65% of power consumption without sacrificing its quality. The proposed approach is beneficial for on-device application of different dense depth prediction methods including monocular and stereo depth prediction, Multi-View-Stereo, depth completion, depth quality enhancement, and depth inpainting. Future efforts need to be made to understand and fully explore the effect of quantization error reduction.

Impact Statement

This paper proposed an innovative approach for increasing bit-precision of quantized depth prediction models on devices with low-precision arithmetic. This paper contributes to the advancement of quantization techniques for dense prediction tasks for devices with limited resources. While there are possible social implications that arise from our work, none of them are particularly relevant in this context.

References

- Ahmed, N., Natarajan, T., and Rao, K. R. Discrete cosine transform. *IEEE Trans. on Computers*, 23(1):90–93, 1974. doi: 10.1109/T-C.1974.223784.
- Bader, M. *Space-filling curves: an introduction with applications in scientific computing*, volume 9. Springer Science & Business Media, 2012. doi: 10.1007/978-3-642-31046-1.
- Bao, H., Dong, L., and Wei, F. BEiT: BERT pre-training

- of image transformers. In *Int. Conf. Learn. Represent.*, 2022. doi: 10.48550/arXiv.2106.08254.
- Bello, I., Fedus, W., Du, X., Cubuk, E. D., Srinivas, A., Lin, T.-Y., Shlens, J., and Zoph, B. Revisiting ResNets: Improved training and scaling strategies. In *Int. Conf. on Neural Information Processing Systems*, pp. 22614–2262, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393. doi: 10.5555/3540261.3541993.
- Bhalgat, Y., Lee, J., Nagel, M., Blankevoort, T., and Kwak, N. LSQ+: Improving low-bit quantization through learnable offsets and better initialization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 2978–2985, 2020. doi: 10.1109/CVPRW50498.2020.00356.
- Borrego-Carazo, J., Ozay, M., Laboyrie, F., and Wisbey, P. A mixed quantization network for computationally efficient mobile inverse tone mapping. *The British Machine Vision Conf.*, 2021. doi: 10.48550/arXiv.2203.06504.
- Cabon, Y., Murray, N., and Humenberger, M. Virtual KITTI 2. *arXiv*, 2020. doi: 10.48550/arXiv.2001.10773.
- Curless, B. and Levoy, M. A volumetric method for building complex models from range images. In *Proc. of Annual Conf. on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pp. 303–312, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917464. doi: 10.1145/237170.237269.
- Dai, A., Chang, A. X., Savva, M., Halber, M., Funkhouser, T., and Nießner, M. ScanNet: richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. doi: 10.1109/CVPR.2017.261.
- Eigen, D., Puhrsch, C., and Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inform. Process. Syst.*, pp. 2366–2374, 2014. doi: 10.5555/2969033.2969091.
- Fang, J., Shafiee, A., Abdel-Aziz, H., Thorsley, D., Georgiadis, G., and Hassoun, J. H. Post-training piecewise linear quantization for deep neural networks. In *Eur. Conf. Comput. Vis.*, pp. 69–86. Springer, 2020. doi: 10.1007/978-3-030-58536-5_5.
- Geiger, A., Lenz, P., and Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.
- Google. Coral Edge TPU documentation, 2020. URL <https://coral.ai/docs/edgetpu/models-intro/#model-requirements>.
- Hu, J., Ozay, M., Zhang, Y., and Okatani, T. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *IEEE Winter Conf. on Applications of Comput. Vis. (WACV)*, pp. 1043–1051, 2019. doi: 10.1109/WACV.2019.00116.
- Ignatov, A., Malivenko, G., Timofte, R., Treszczotko, L., Chang, X., Ksiazek, P., Lopuszynski, M., Pioro, M., Rudnicki, R., Smyl, M., Ma, Y., Li, Z., Chen, Z., Xu, J., Liu, X., Jiang, J., Shi, X., Xu, D., Li, Y., Wang, X., Lei, L., Zhang, Z., Wang, Y., Huang, Z., Luo, G., Yu, G., Fu, B., Li, J., Wang, Y., Huang, Z., Cao, Z., Conde, M. V., Sapozhnikov, D., Lee, B. H., Park, D., Hong, S., Lee, J., Lee, S., and Chun, S. Y. Efficient single-image depth estimation on mobile devices, mobile ai & aim 2022 challenge: Report. In *Eur. Conf. Comput. Vis.*, pp. 71–91, 2022. doi: 10.1007/978-3-031-25066-8_4.
- Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 2704–2713, 2018. doi: 10.1109/CVPR.2018.00286.
- Jiang, X., Cambareri, V., Agresti, G., Ugwu, C. I., Simonetto, A., Cardinaux, F., and Zanuttigh, P. A low memory footprint quantized neural network for depth completion of very sparse time-of-flight depth maps. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pp. 2687–2696, June 2022. doi: 10.1109/CVPRW56347.2022.00302.
- Ke, B., Obukhov, A., Huang, S., Metzger, N., Daudt, R. C., and Schindler, K. Repurposing diffusion-based image generators for monocular depth estimation. *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 9492–9502, 2023. doi: 10.1109/CVPR52733.2024.00907.
- Kiyama, M., Nakahara, Y., Amagasaki, M., and Iida, M. A quantized neural network library for proper implementation of hardware emulation. In *Int. Symp. on Computing and Networking Workshops (CANDARW)*, pp. 136–140, 2019. doi: 10.1109/CANDARW.2019.00032.
- Kolbeinsson, B. and Mikolajczyk, K. DDOS: The drone depth and obstacle segmentation dataset. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pp. 7328–7337, 2024. doi: 10.1109/CVPRW63382.2024.00728.
- Lahitani, A. R., Permanasari, A. E., and Setiawan, N. A. Cosine similarity to determine similarity measure: study case in online essay assessment. In *Int. Conf. on Cyber and IT Service Management*, pp. 1–6, 2016. doi: 10.1109/CITSM.2016.7577578.
- Li, G., Xue, J., Liu, L., Wang, X., Ma, X., Dong, X., Li, J., and Feng, X. Unleashing the low-precision computation potential of tensor cores on gpus. In *IEEE/ACM Int. Symp. on Code Generation and Optimization (CGO)*, pp. 90–102, 2021. doi: 10.1109/CGO51591.2021.9370335.

- Li, Z., Yang, T., Wang, P., and Cheng, J. Q-ViT: Fully differentiable quantization for vision transformer. *arXiv*, 2022. doi: 10.48550/arXiv.2201.07703.
- Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, 2014. doi: 10.1007/978-3-319-10602-1_48.
- Lipson, L., Teed, Z., and Deng, J. RAFT-stereo: Multilevel recurrent field transforms for stereo matching. In *Int. Conf. on 3D Vision (3DV)*, pp. 218–227, 2021. doi: 10.1109/3DV53792.2021.00032.
- Liu, Q. and Zhou, S. LightDepthNet: Lightweight CNN architecture for monocular depth estimation on edge devices. *IEEE Trans. on Circuits and Systems II: Express Briefs*, 71(4):2389–2393, 2023. doi: 10.1109/TCSII.2023.3337369.
- Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2016. doi: 10.1109/cvpr.2016.438.
- Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., Baalen, M. V., and Blankevoort, T. A white paper on neural network quantization. *arXiv*, 2021. doi: 10.48550/arXiv.2106.08295.
- Nicolas, B., Li, Y., Chris, P., and Aaron, C. Delving deeper into convolutional networks for learning video representations. *arXiv*, 2015. doi: 10.48550/arXiv.1511.06432.
- Oh, S., Kim, H.-J. S., Lee, J., and Kim, J. RRNet: Repetition-reduction network for energy efficient depth estimation. *IEEE Access*, 8:106097–106108, 2020. doi: 10.1109/ACCESS.2020.3000773.
- Oquab, M., Darcet, T., Moutakanni, T., Vo, H. Q., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y. B., Li, S.-W., Misra, I., Rabat, M. G., Sharma, V., Synnaeve, G., Xu, H., Jégou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. DINOv2: Learning robust visual features without supervision. *Trans. on Machine Learning Research*, 2024. ISSN 2835-8856. doi: 10.48550/arXiv.2304.07193. URL <https://openreview.net/forum?id=a68SUt6zFt>.
- Qualcomm Technologies, Inc. Snapdragon neural processing engine SDK, 2025. URL <https://www.qualcomm.com/developer/software/neural-processing-sdk-for-ai>.
- Ranftl, R., Bochkovskiy, A., and Koltun, V. Vision transformers for dense prediction. In *Int. Conf. Comput. Vis.*, pp. 12179–12188, October 2021. doi: 10.1109/ICCV48922.2021.01196.
- Rasla, A. and Beyeler, M. The relative importance of depth cues and semantic edges for indoor mobility using simulated prosthetic vision in immersive virtual reality. In *Proc. of ACM Symp. on Virtual Reality Software and Tech.*, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450398893. doi: 10.1145/3562939.3565620.
- Rouse, D. M. and Hemami, S. S. Understanding and simplifying the structural similarity metric. In *IEEE Int. Conf. Image Process.*, pp. 1188–1191, 11 2008. doi: 10.1109/ICIP.2008.4711973.
- Rousseeuw, P. J. and Croux, C. Alternatives to the median absolute deviation. *J. of the American Statistical Association*, 88:1273–1283, 1993. doi: 10.1080/01621459.1993.10476408.
- Sagan, H. *Space-filling curves*. Springer Science & Business Media, 2012.
- Sakr, C., Dai, S., Venkatesan, R., Zimmer, B., Dally, W., and Khailany, B. Optimal clipping and magnitude-aware differentiation for improved quantization-aware training. In *Int. Conf. Mach. Learn.*, volume 162, pp. 19123–19138. PMLR, 2022. URL <https://proceedings.mlr.press/v162/sakr22a/sakr22a.pdf>.
- Shao, J., Yang, Y., Zhou, H., Zhang, Y., Shen, Y., Poggi, M., and Liao, Y. Learning temporally consistent video depth from video diffusion priors. *arXiv*, 2024. doi: 10.48550/arXiv.2406.01493.
- Shao, S., Pei, Z., Chen, W., Wu, X., and Li, Z. NDDepth: Normal-distance assisted monocular depth estimation. In *Int. Conf. Comput. Vis.*, pp. 7931–7940, 2023. doi: 10.1109/ICCV51070.2023.00729.
- Sheng, T., Feng, C., Zhuo, S., Zhang, X., Shen, L., and Aleksic, M. A quantization friendly separable convolution for MobileNets. In *1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2)*, pp. 14–18, 3 2018. doi: 10.1109/EMC2.2018.00011.
- Shkolnik, M., Chmiel, B., Banner, R., Shomron, G., Nahshan, Y., Bronstein, A., and Weiser, U. Robust quantization: one model to rule them all. In *Adv. Neural Inform. Process. Syst.*, volume 33, 2020. URL <https://proceedings.neurips.cc/paper/2020/file/3948ead63a9f2944218de038d8934305-Paper.pdf>.

- Shlezinger, N., Eldar, Y. C., and Rodrigues, M. R. D. Hardware-limited task-based quantization. In *IEEE Int. Workshop on Signal Process. Advances in Wireless Communications (SPAWC)*, pp. 1–5, 8 2019. doi: 10.1109/SPAWC.2019.8815422.
- Song, L., Shi, D., Xia, J., Ouyang, Q., Qiao, Z., Jin, S., and Yang, S. Spatial-aware dynamic lightweight self-supervised monocular depth estimation. *IEEE Robotics and Automation Lett.*, 9(1):883–890, 2023. doi: 10.1109/LRA.2023.3337991.
- Toshev, A. and Szegedy, C. DeepPose: Human pose estimation via deep neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 1653–1660, 2014. doi: 10.1109/CVPR.2014.214.
- Ventrella, J. *Brainfilling curves – a fractal bestiary*. Eye-brain Books, 2012.
- Wadekar, S. N. and Chaurasia, A. MobileViTv3: mobile-friendly vision transformer with simple and effective fusion of local, global and input features. *arXiv*, 2022. doi: 10.48550/arXiv.2209.15159.
- Wang, K., Liu, Z., Lin, Y., Lin, J., and Han, S. HAQ: Hardware-aware automated quantization with mixed precision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 8604–8612, 2019a. doi: 10.1109/CVPR.2019.00881.
- Wang, T., Wang, K., Cai, H., Lin, J., Liu, Z., and Han, S. APQ: Joint search for network architecture, pruning and quantization policy. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020. doi: 10.1109/CVPR42600.2020.00215.
- Wang, Y., Chao, W.-L., Garg, D., Hariharan, B., Campbell, M., and Weinberger, K. Q. Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for autonomous driving. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 8437–8445, 2019b. doi: 10.1109/CVPR.2019.00864.
- Wang, Z., Bovik, A. C., and Sheikh, H. R. Structural similarity based image quality assessment. *Digital Video Image Quality and Perceptual Coding, Ser. Series in Signal Process. and Communications*, 11 2005. doi: 10.1201/9781420027822.ch7.
- Wofk, D., Ma, F., Yang, T.-J., Karaman, S., and Sze, V. FastDepth: Fast monocular depth estimation on embedded systems. In *Int. Conf. on Robotics and Automation (ICRA)*, pp. 6101–6108, 2019. doi: 10.1109/ICRA.2019.8794182.
- Xianqi, W., Gangwei, X., Hao, J., and Xin, Y. Selective-stereo: Adaptive frequency information selection for stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 19701–19710, 2024. doi: 10.1109/CVPR52733.2024.01863.
- Xiao, B., Wu, H., and Wei, Y. Simple baselines for human pose estimation and tracking. In *Eur. Conf. Comput. Vis.*, pp. 472–487, 2018. doi: 10.1007/978-3-030-01231-1_29.
- Xu, G., Wang, X., Ding, X., and Yang, X. Iterative geometry encoding volume for stereo matching. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 21919–21928, 2023. doi: 10.1109/CVPR52729.2023.02099.
- Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J., and sheng Hua, X. Quantization networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 7308–7316, 2019. doi: 10.1109/CVPR.2019.00748.
- Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., and Zhao, H. Depth anything: Unleashing the power of large-scale unlabeled data. *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 10371–10381, 2024a. doi: 10.1109/CVPR52733.2024.00987.
- Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., and Zhao, H. Depth anything v2. *arXiv*, 2024b. doi: 10.48550/arXiv.2406.09414.
- Yeshwanth, C., Liu, Y.-C., Nießner, M., and Dai, A. ScanNet++: A high-fidelity dataset of 3D indoor scenes. In *Int. Conf. Comput. Vis.*, pp. 12–22, 2023. doi: 10.1109/ICCV51070.2023.00008.
- You, Y., Wang, Y., Chao, W.-L., Garg, D., Pleiss, G., Hariharan, B., Campbell, M. E., and Weinberger, K. Q. Pseudo-LiDAR++: Accurate depth for 3D object detection in autonomous driving. In *Int. Conf. Learn. Represent.*, 2020. URL <https://openreview.net/forum?id=BJedHRVtPB>.
- Yuan, Z., Xue, C., Chen, Y., Wu, Q., and Sun, G. PTQ4ViT: Post-training quantization for vision transformers with twin uniform quantization. In *Eur. Conf. Comput. Vis.*, pp. 191–207. Springer, 2022. doi: 10.1007/978-3-031-19775-8_12.
- Zhang, N., Nex, F., Vosselman, G., and Kerle, N. Channel-aware distillation transformer for depth estimation on nano drones. *arXiv*, 2023. doi: 10.48550/arXiv.2303.10386.
- Zhou, Q.-Y., Park, J., and Koltun, V. Open3D: A modern library for 3D data processing. *arXiv*, 2018. doi: 10.48550/arXiv.1801.09847.

A. Parametric curve selection

The Hilbert curve is one representative of the wide class of space-filling curves (Ventrella, 2012). Let us provide additional arguments in favor of Hilbert curve selection for high-precision depth prediction. In our analysis, we follow the terminology of (Ventrella, 2012) that classifies curves on square and triangular grids, and divides them into families \sqrt{N} . In family \sqrt{N} , the distance between starting and ending points of the curve generator equals to \sqrt{N} .

In order to fully utilize depth representation as two components, it is desirable that a space-filling curve uniformly covers the unit square. This requirement eliminates all curves on triangular grids.

Curves with non-orthogonal generators have the drawback of filling square not uniformly as for example Z-order curves. Among curves with orthogonal generation curves defined on square grid our choice is limited to the Hilbert curve ($\sqrt{4}$ family) (Fig. 11a), the Peano curve ($\sqrt{9}$ family)(Fig. 11b) and the Quadratic Gosper curve ($\sqrt{25}$ family)(Fig. 11c). For Hilbert and Peano curves different generators are possible (e.g., the Moore curve is a variant of the Hilbert curve) but they are different only in the order the space is filled.

The value of N defines how fast the curve length L_p increases and curve edge size h_p decreases with the curve order. Our experiments show that it is desirable to have the ability of fine-tuning the curve length depending on the DNN quantization error magnitude. From this point of view, the Hilbert curve is the most flexible as it has the lowest N value. For the Hilbert curve, the number of nodes grows as 1, 4, 16, 64, 256 with the order p . For Peano curve, the number of nodes grows as 1, 9, 81, 729, 6561, and for Quadratic Gosper curve as 1, 25, 625, 15625, 390625. If we limit the number of nodes to a reasonable value of 256, the Hilbert curve provides 4 usable low-order curves (that we experiment with in the paper), Peano – 2, Quadratic Gosper Curve – 1.

The Hilbert curve is the simplest and the most flexible curve that satisfies all requirements essential for coding depth values. To construct even more flexible list of curves, it is possible to use Hilbert, Peano and Quadratic Gosper curve of different orders to create a sequence of curves with the number of nodes 4, 9, 16, 25, 64, 81, 256.

Provided the main requirements for the parametric curve (self-avoidance, uniform filling of unit square, continuity) are satisfied, the detailed structure of the curve is not important. For example, we can use arbitrary non-self-similar curves that fill unit square with a given number of nodes, curves with smoothed corners, curves that stretch different parts of 1D value in a different degree (to emphasize the most probable range of target quantity variation).

B. DPT model modification

As it mentioned in the paper, one of the models chosen for the experiments is Dense Prediction Transformer (DPT) (Ranftl et al., 2021). All modifications of the model architecture are illustrated in Fig. 12. An additional 1×1 2D convolution layer was used for proper integration of the input RGB stereo pair into the MobileViTv3-S (Wadkar & Chaurasia, 2022) backbone.

Also, MobileNet blocks in the encoder are modified for better quantization as described by Sheng et al. (Sheng et al., 2018). For disparity prediction, the original DPT head was used. The Hilbert curve head architecture for this model includes an additional up-sample layer after the first 3×3 2D convolution layer.

The DPT model includes a MobileViTv3-S encoder with skip connections before each MobileViT block. Each skip

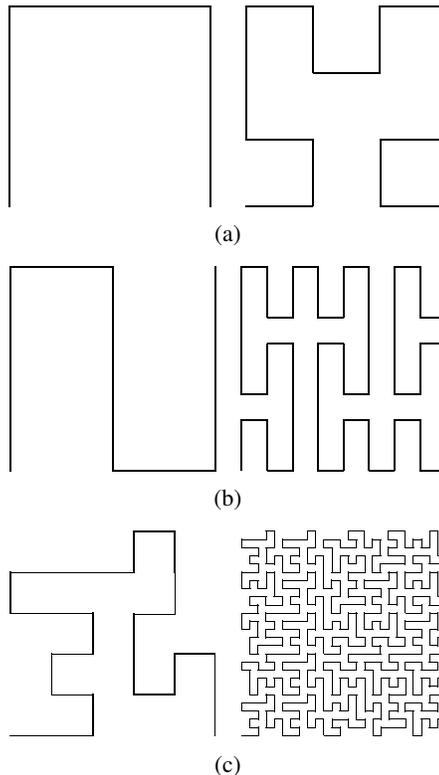


Figure 11. Space-filling curves filling unit square. The first (left column) and the second (right column) order p curves for: (a) Hilbert curve, (b) Peano curve, and (c) Quadratic Gosper curve.

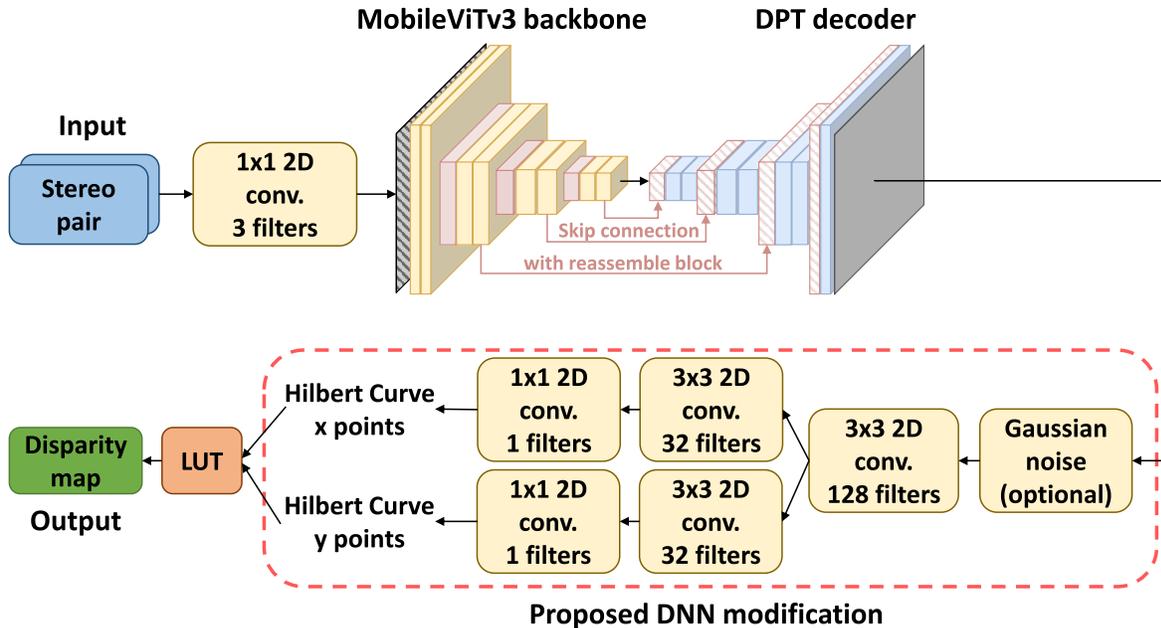


Figure 12. The hpDPT model architecture. The input RGB stereo pair is processed by an encoder which is MobileViTv3-S backbone and the decoder proposed for depth prediction in DPT. Features from the decoder are fed to an optional Gaussian noise layer and 3×3 2D convolution layer followed by two heads for Hilbert curve components. They consist of one 3×3 and one 1×1 2D convolution layers with a decreasing number of filters: 32 and 1 respectively. At the post-processing stage, Hilbert components are converted to the final disparity map.

connection integrates into the decoder part using a reassemble block proposed by Ranftl et al. (Ranftl et al., 2021).

During analysis of the network architecture, we found that layers in MobileNet blocks have large kurtosis values of their weights’ distributions. It is suggested in (Shkolnik et al., 2020) that large kurtosis values might lead to the model quantization quality degradation because of outliers clipping. We added kurtosis regularization proposed in (Shkolnik et al., 2020) to all 1×1 convolutions in MobileNet blocks in MobileViTv3-S to reduce quantization error in both experiments with depth and Hilbert components output.

C. Experiment on KITTI 2012 dataset

We evaluate our approach on KITTI 2012 (Geiger et al., 2012) dataset. KITTI 2012 is a real-world dataset for autonomous driving domain with sparse ground-truth disparities collected with a LiDAR system. Our evaluation is based on 194 images in the training part of KITTI 2012. Training dataset is composed of ScanNet and Virtual KITTY 2 (Cabon et al., 2020) datasets with 25/75% balancing. The DispNet and h2DispNet models were trained in 256 by 1152 px input resolution and 128 by 576 px output disparity resolution. Training settings are the same as for the ScanNet experiment.

As shown in Table 3, for the original DispNet model we reached EPE 1.38 px and D1 9.13%. Interestingly, h2DispNet shows better results with EPE 1.07 px and D1 5.22%. We exclude S_c from the analysis, because it is not applicable to sparse ground-truth disparities of KITTI 2012 dataset. The DispNet model quantized to W8A8 format and run on DSP shows degradation of both EPE (to 1.63 px) and D1 (to 9.95%). At the same time the h2DispNet in W8A8 format on DSP shows very minor quality degradation. Measured as EPE between disparity predicted by FP32 and W8A8 models, quantization error is 1.01 px for the DispNet and 0.38 px for the h2DispNet. It corresponds to gain in disparity prediction quality on device of 2.6 times. Overall, on DSP the h2DispNet improves D1 by approximately 4.6% and EPE by about 33% compared to the original

Precision	Abs Rel, % ↓	EPE, px ↓	D1, % ↓
FP32	3.88	1.38	9.13
W8A8	5.01	1.63	9.95
Ours, FP32	3.24	1.07	5.22
Ours, W8A8	3.30	1.09	5.36

Table 3. Results on KITTI 2012 for the DispNet and h2DispNet models. Models in FP32 format are run on CPU; models in W8A8 formats are run on DSP. Best metrics for W8A8 models are shown in bold.

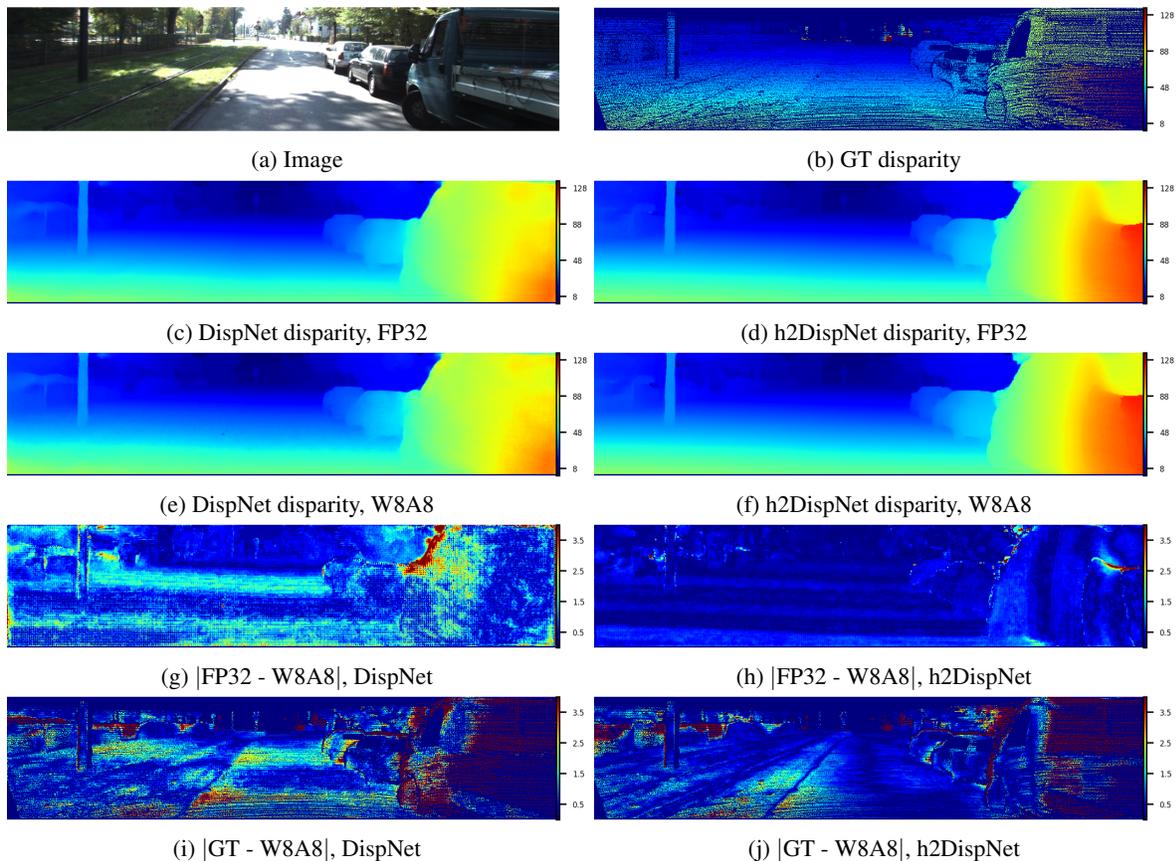


Figure 13. Comparison of DispNet and h2DispNet on KITTI 2012 dataset. Differences between disparity maps predicted by FP32 (CPU) (c, d) and W8A8 (DSP) (e, f) models are shown in (g) and (h); absolute disparity prediction errors for W8A8 (DSP) models are shown in (i) and (j). All values are presented in pixels.

DispNet model. Examples of predicted disparity are given in Fig. 13. This experiment demonstrates that the proposed approach can be successfully applied to real datasets.

D. Quantization quality influence on mesh fusion

We provide additional experiment to understand how quantization artifacts in depth maps affect a scene mesh reconstruction. For a mesh fusion we utilize truncated signed distance function (TSDF) (Curless & Levoy, 1996) approach as implemented in Python library Open3D (Zhou et al., 2018). We used scalable TSDF volume with parameters $voxel_length = 0.01 m$, $sdf_trunc = 0.15$. For experiments we chose ScanNet scene scene0050_02 comprising 4379 frames. Each 40th frame was used to reconstruct mesh. Quality of the mesh fused from GT depth maps is illustrated in Fig. 14.

In Fig. 15 (Fig. 16), we show qualitative results of h2DispNet (h2DPT) model compared to the corresponding baseline variant. 3D meshes fused from depth maps predicted by FP32 h2DispNet (Fig. 15b) and FP32 h2DPT (Fig. 16b) models have very similar structure and depth smoothness compared to the FP32 DispNet (Fig. 15a) and FP32 DPT (Fig. 16a). Both baseline and modified FP32 models' variants produce quality of reconstructed 3D mesh comparable to GT (Fig. 14) but with slightly smoother structure.

We observe two types of quantization artifacts present in fused meshes for models run on CPU delegate. The first is the noise on flat surfaces for original DispNet in W8A8 format (Fig. 15e); the second is presence of visible edges of different frames' depth maps (step-like structures) in the mesh for original DPT in W8A8 format (Fig. 16e). We attribute the second type of artifacts to the systematic errors in depth prediction leading to errors in depth scale. Models modified according to the proposed method lead to mesh reconstruction with much reduced noise level (Figs. 15f and 16f).

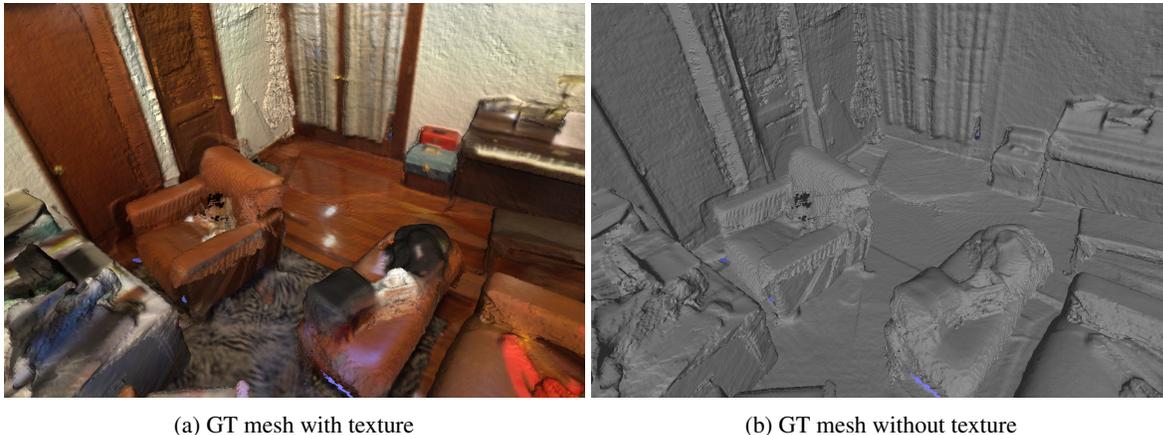


Figure 14. A view of 3D mesh fused with GT depth maps for ScanNet scene scene0050.02.

For the baseline models run on DSP delegate, we observe the same artifacts but more pronounced (Figs. 15c–16c). The h2DispNet model in W8A8 format almost eliminates quantization artifacts (Fig. 15d) and restores mesh spatial details. The h2DPT in W8A8 format model removes step-like artifacts and reduces noise for flat surfaces.

E. Additional details of depth maps quality

In Figs. 17–21 we show additional examples of depth maps predicted by original and modified DispNet and DPT models. These examples represent both simpler (Figs. 17–20) and more complex scenes (Fig. 21). In all examples, modified models have significantly smaller quantization error; remaining errors are concentrated on depth discontinuities. Errors in the vicinity of depth discontinuities are also present for FP32 models and not linked to the proposed approach.

F. Human pose estimation experiment

Two approaches were realized for the Human pose estimation (HPE) experiment: direct keypoints regression (Toshev & Szegedy, 2014) and via representing keypoints as heatmaps (Xiao et al., 2018). For training, the MS COCO (Lin et al., 2014) dataset with human pose keypoints labeling was used. Both models have ResNet-RS (Bello et al., 2021) as an encoder with alpha equals 0.5. The head for direct regression is composed of fully-connected layers as it was introduced by Toshev et al. The model for heatmaps prediction includes DispNet decoder and convolutional head proposed by Xiao et al.

For HPE experiment we trained two lightweight models: direct keypoints prediction with $AP_{50} = 43$, and heatmaps prediction with $AP_{50} = 83$. We found that the most stable way to train the modified HPE model is training the original model first, using it as a teacher model, and calculating GT Hilbert curve components x_{GT} and y_{GT} from the teacher model predictions in the loss term (3). In this manner, the modified model can be trained even when GT data for calculation of x_{GT} and y_{GT} are not available, for example in case of heatmaps learning.

For model with direct keypoints prediction, along-the-curve quantization error for model with Hilbert curve-based representation for $p = 1, 2$ increases approximately L times as compared to the keypoints quantization error of the original model (see Fig. 22a for $p = 1$ case); across-the-curve error is an order of magnitude smaller and is similar to the quantization error of the original model. As a result, quantization error of the modified model in keypoints space remains almost unchanged. This result suggests that for the direct regression task quantization error is strongly signal-dependent. The only effect from the proposed modification is in increase of output bit-precision.

The situation is different for the heatmaps prediction model. In this case, quantization error along the Hilbert curve of the second order increases less significantly as compared to the original model. From Fig. 22b it is seen that the along-the-curve error distribution is narrower than the original heatmaps quantization error multiplied by the curve length. As a result, quantization error of the modified model in heatmaps space is 2.69 times smaller than for the original model. In addition, the spatial structure of predicted heatmaps is improved as shown in Figs. 22c and 22d. The latter effect is caused by increase of the output bit-precision.

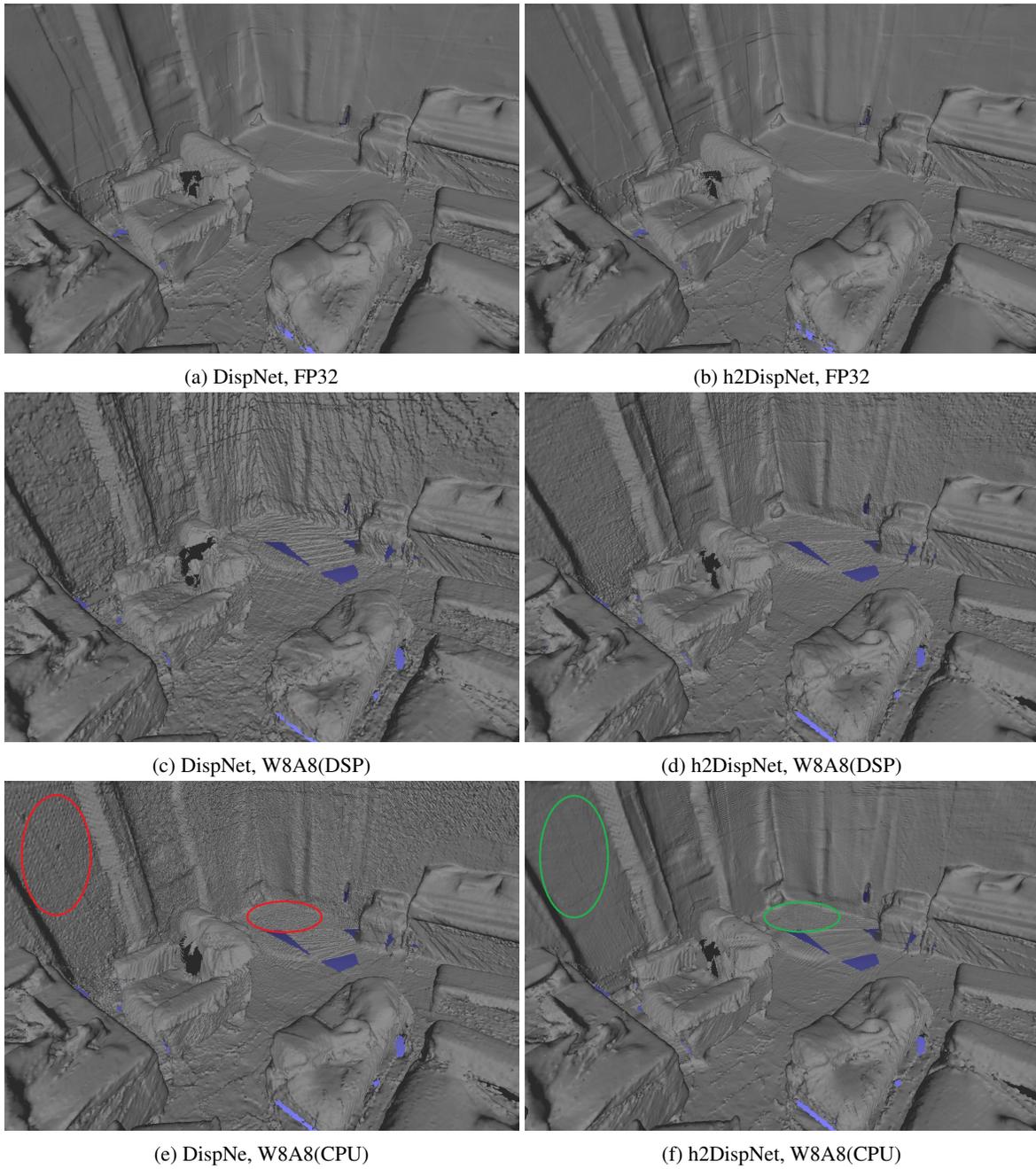


Figure 15. A view of 3D mesh fused with predicted depth maps by DispNet and h2DispNet for ScanNet scene scene0050_02. Some reconstruction errors are highlighted by red and improved structures are marked by green.

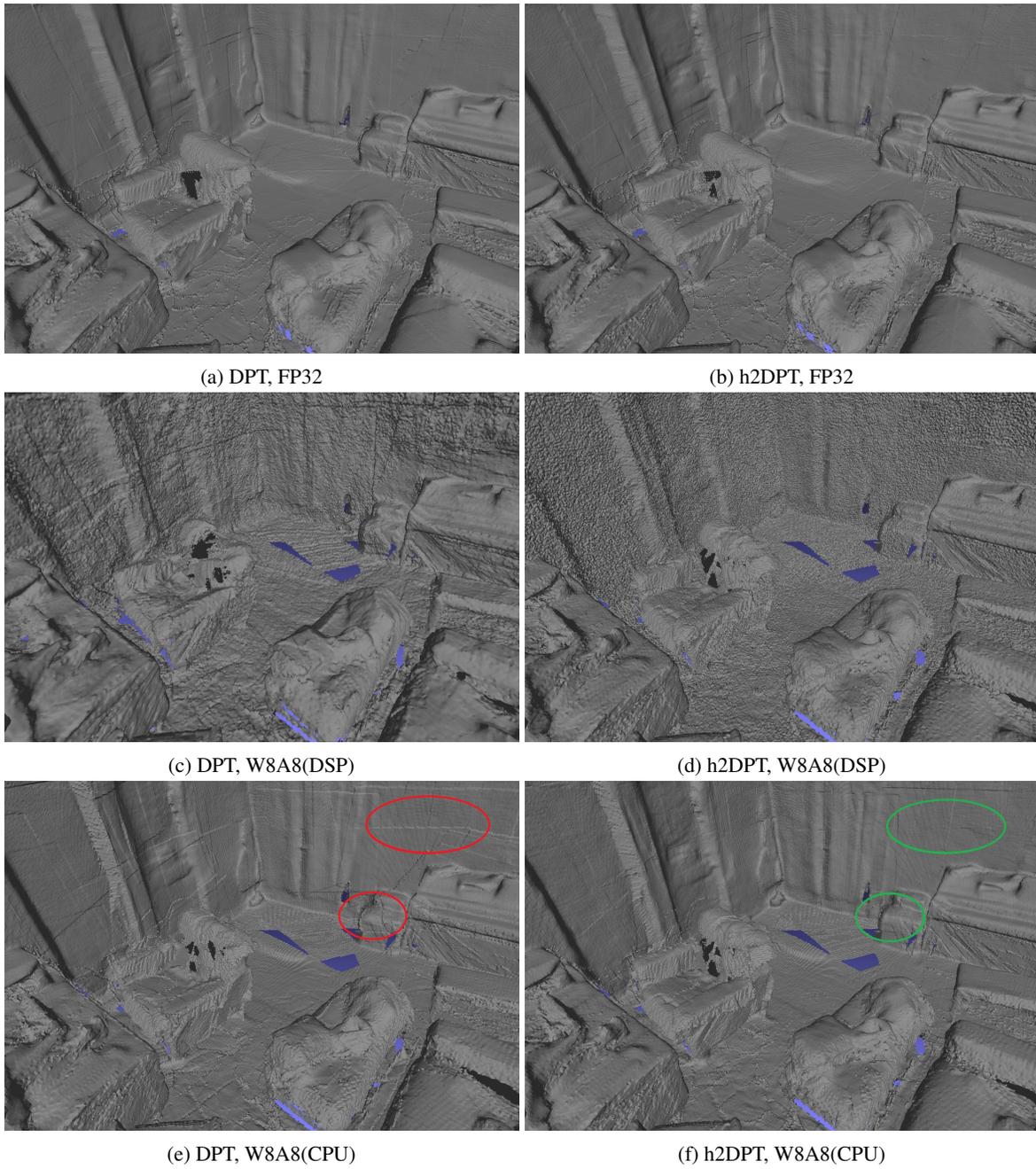


Figure 16. A view of 3D mesh fused with predicted depth maps by DPT and h2DPT for ScanNet scene scene0050_02. Some reconstruction errors are highlighted by red and improved structures are marked by green.

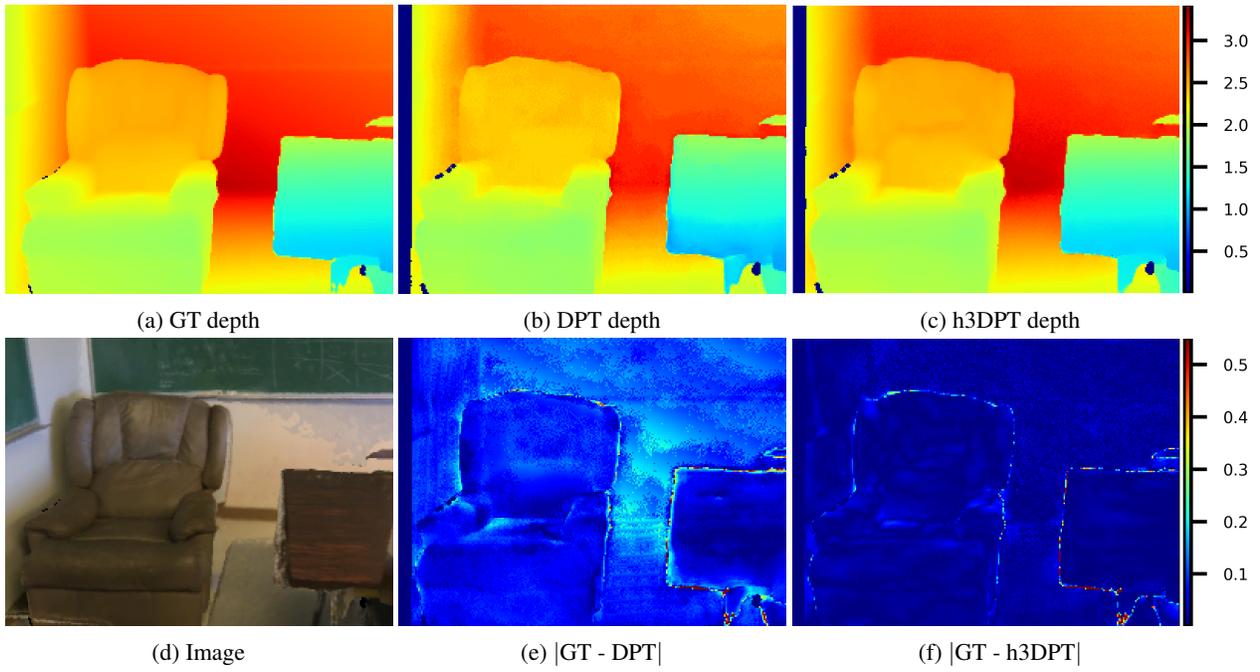


Figure 17. Depth errors of DPT and h3DPT models on DSP. ScanNet scene scene0030_02. All values are presented in meters.

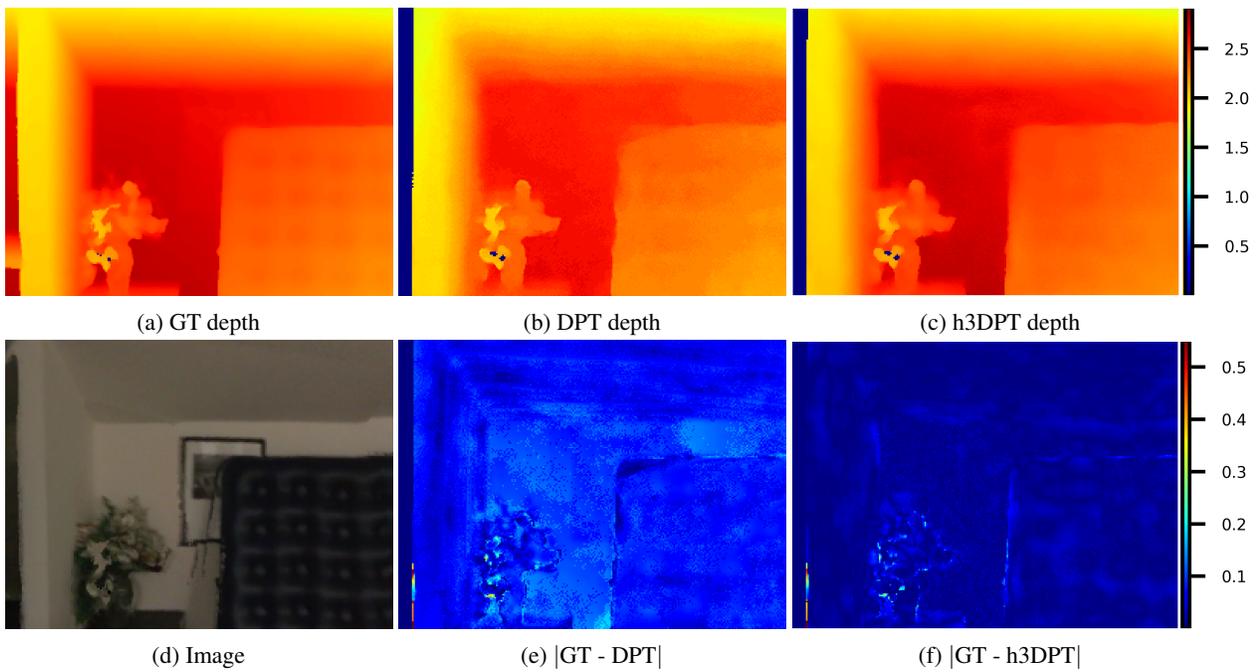


Figure 18. Depth errors of DPT and h3DPT models on DSP. ScanNet scene scene0629_00. All values are presented in meters.

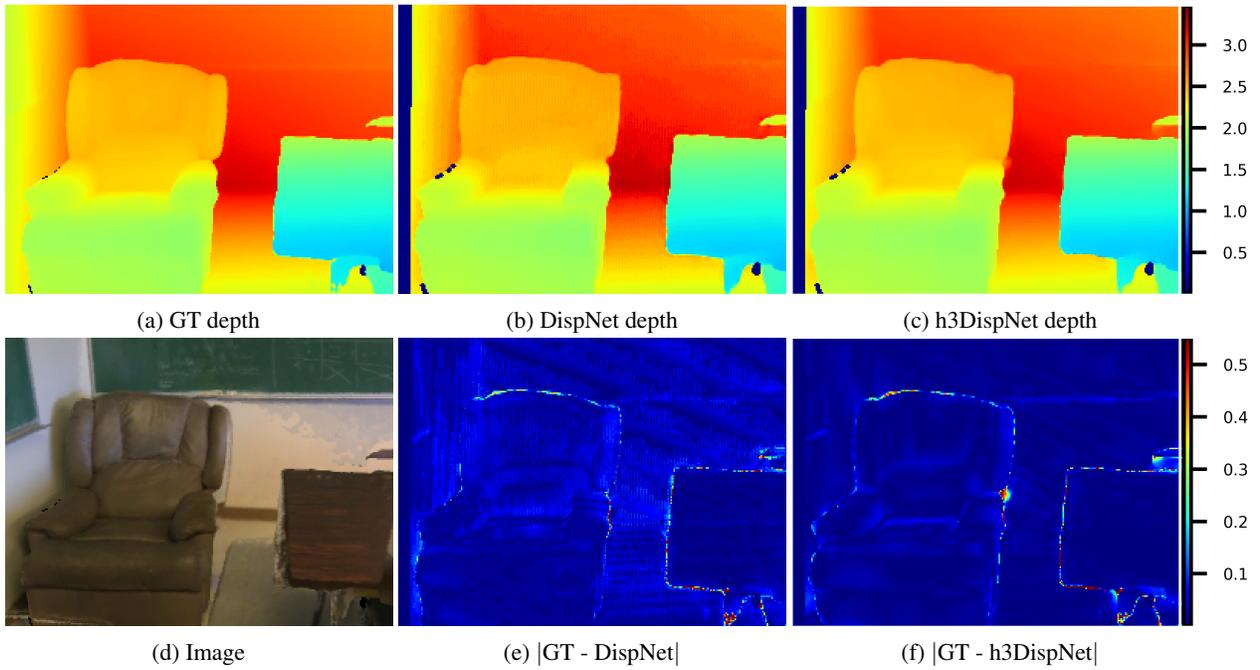


Figure 19. Depth errors of DispNet and h3DispNet models on DSP. ScanNet scene scene0030_02. All values are presented in meters.

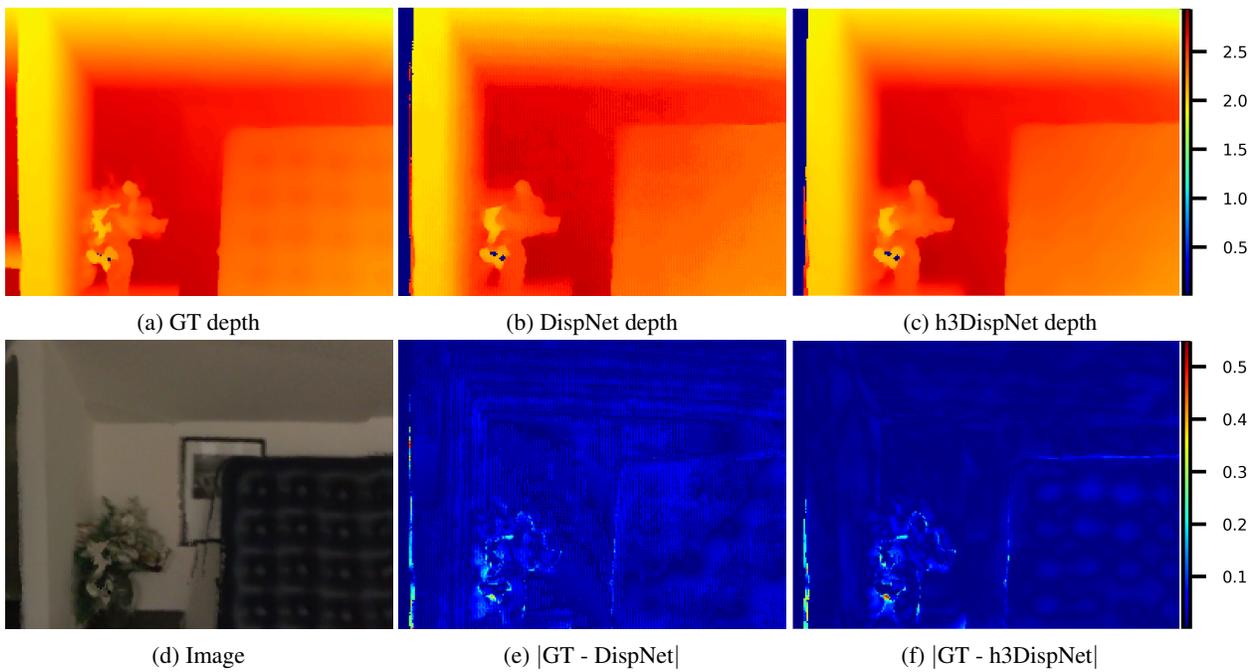


Figure 20. Depth errors of DispNet and h3DispNet models on DSP. ScanNet scene scene0629_00. All values are presented in meters.

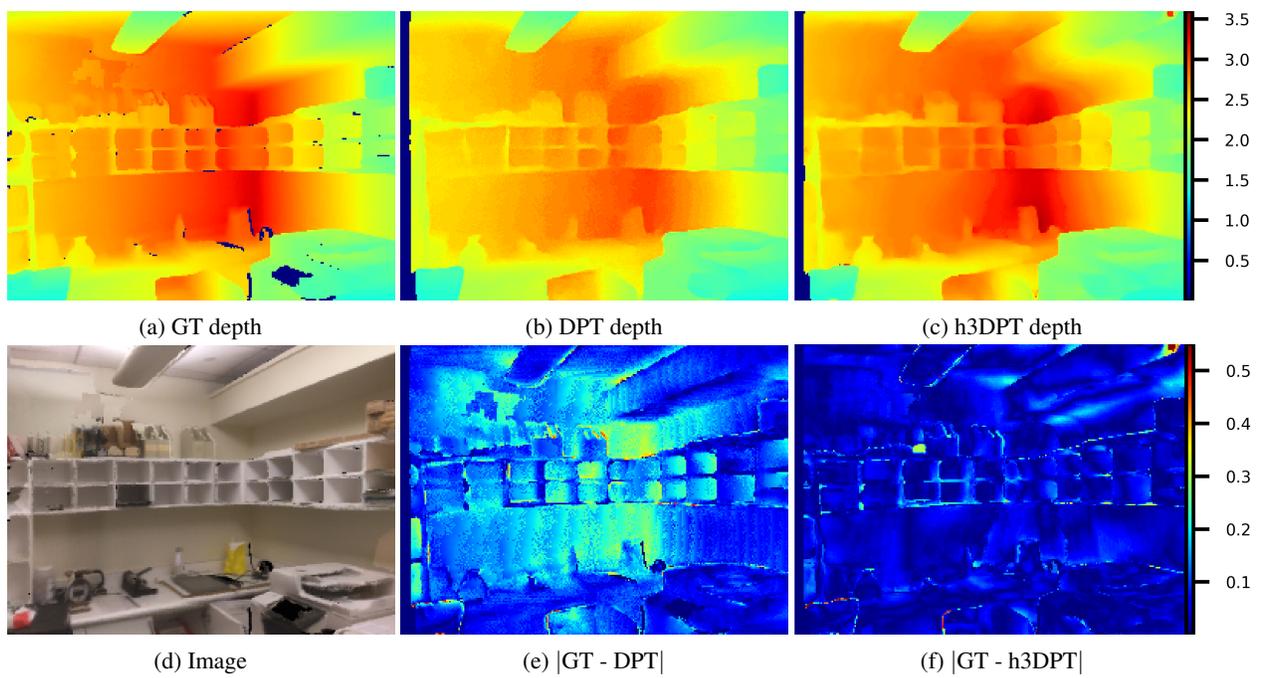
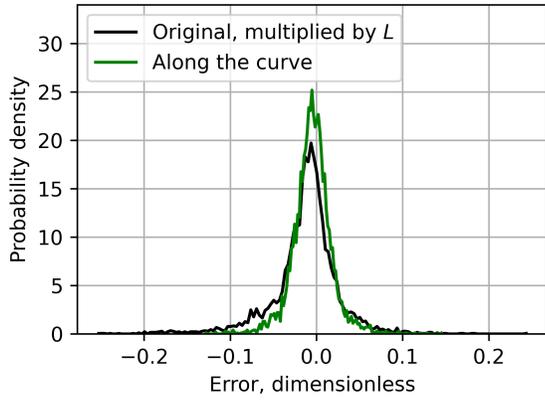
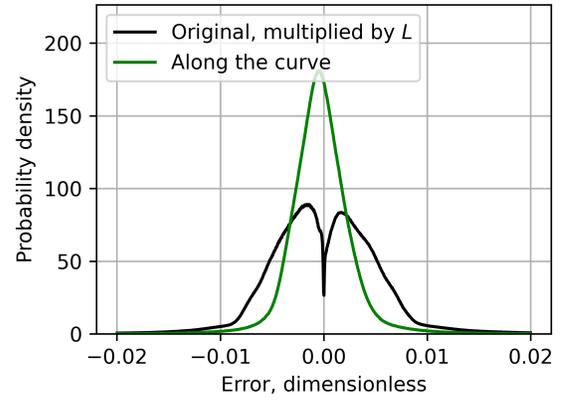


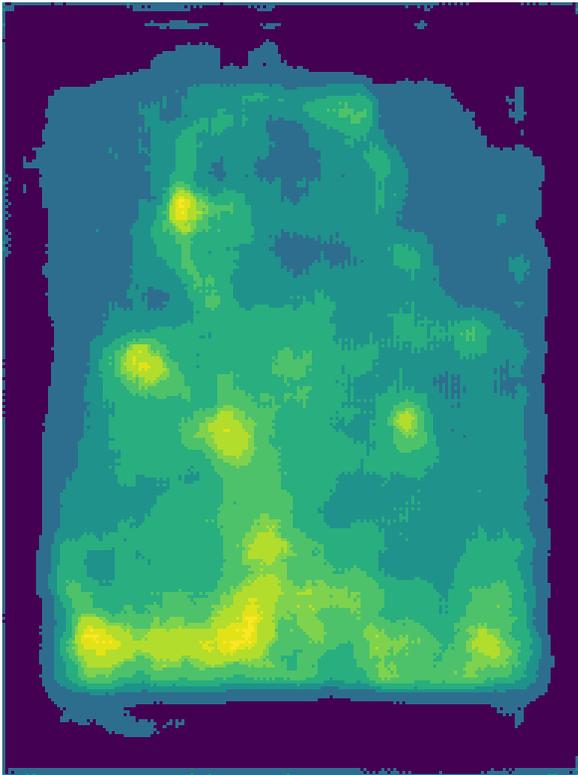
Figure 21. Depth errors of DPT and h3DPT models on DSP. ScanNet scene scene0804.00. All values are presented in meters.



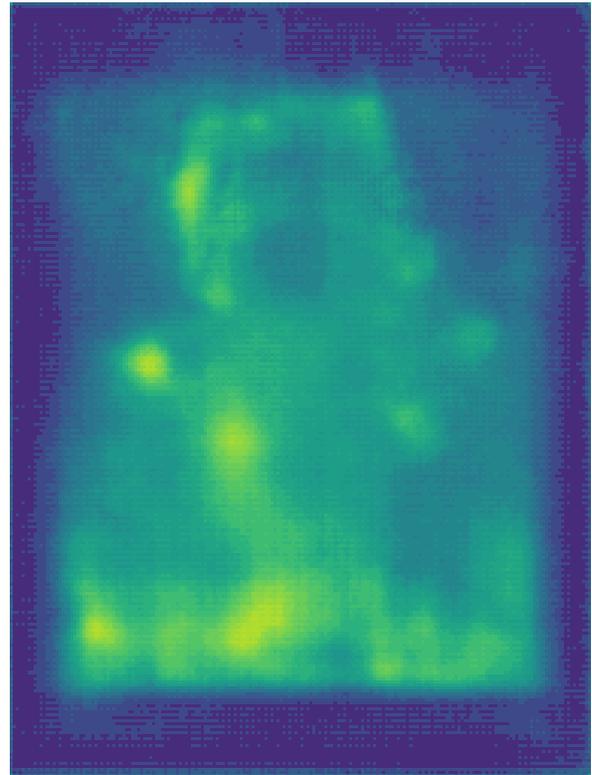
(a) Keypoints error



(b) Heatmaps error



(c) Heatmap by the original HPE model



(d) Heatmap by the modified HPE model

Figure 22. Quantization error for HPE using direct keypoints regression (a) and heatmaps (b). Comparison of heatmaps prediction on DSP using W8A8 models (c, d).