

# Adaptive Illumination based Depth Sensing using Deep Superpixel and Soft Sampling Approximation

Qiqin Dai, Fengqiang Li, Oliver Cossairt, and Aggelos K. Katsaggelos, *Fellow, IEEE*

**Abstract**—Dense depth map capture is challenging in existing active sparse illumination based depth acquisition techniques, such as LiDAR. Various techniques have been proposed to estimate a dense depth map based on fusion of the sparse depth map measurement with the RGB image. Recent advances in hardware enable adaptive depth measurements resulting in further improvement of the dense depth map estimation. In this paper, we study the topic of estimating dense depth from depth sampling. The adaptive sparse depth sampling network is jointly trained with a fusion network of an RGB image and sparse depth, to generate optimal adaptive sampling masks. Deep learning based superpixel sampling and soft sampling approximation are applied. We show that such adaptive sampling masks can generalize well to many RGB and sparse depth fusion algorithms under a variety of sampling rates (as low as 0.0625%). The proposed adaptive sampling method is fully differentiable and flexible to be trained end-to-end with upstream perception algorithms.

**Index Terms**—Depth estimation, adaptive sampling, deep learning, sensor fusion.

## I. INTRODUCTION

Depth sensing and estimation is important for many applications, such as autonomous driving [1], augmented reality (AR) [2], and indoor perception [3].

Based on the principle of operation, we can roughly divide current depth sensors into two categories: (1) Triangulation-based depth sensors (eg., stereo [4]) and (2) Time-of-flight (ToF) based depth sensors (including direct ToF LiDAR sensor and indirect ToF cameras [5]). Among these depth sensors, LiDAR has a much longer imaging range (e.g., tens of meters) with a high depth precision (e.g., mm) which enables it to be a competitive depth sensors for numerous emerging commercial applications. LiDAR has been widely used for machine vision applications, such as, navigation in self-driving cars and mobile devices (e.g., LiDAR sensor on Apple iPhone 12).

Most LiDAR sensors illuminate a single point of the object and measure the depth/distance information for that point at a time. LiDAR sensors then rely on raster scanning to generate a full 3D image of the object, which limits the acquisition speed. In order to produce a full 3D image of an object with a reasonable frame rate using mechanical scanning, LiDAR can only provide a sparse scanning with significant inter-sample spacing. This leads to very limited spatial resolution with

LiDAR sensors. To increase LiDAR’s spatial resolution and acquire more structural information of the scene, other high-spatial-resolution imaging modalities, such as RGB, have been used to be fused with LiDAR’s depth images [6], [7].

Traditionally, LiDAR sensors perform raster scanning following a regular grid to produce a uniformly spaced depth map. Recently, researchers explore adaptive illumination/scanning patterns for LiDAR sensors based on the scene and co-optimize the scanning pattern with the multimodal sensor fusion pipeline to further increase performance [8]. Pittaluga *et al.* [9] implement optimized LiDAR scanning on a real hardware device using a MEMS mirror. They co-optimize the scanning hardware and the fusion pipeline with an RGB sensor to increase LiDAR’s performance and achieve higher resolution depth map. Tasneem *et al.* [10] utilize adaptive fovea LiDAR scanning to achieve highest angular resolution over regions of interest (ROIs) which can help improve the machine perception accuracy. Yamamoto *et al.* [11] also propose adaptive LiDAR scanning for efficient and accurate detection on pedestrian with dense scans. With adaptive illumination/sampling, we might also achieve a reasonable depth map with smaller number of samples for post machine perception tasks [8], which reduces the sensor bandwidth and may potentially enable higher LiDAR sensor frame rate for those without mechanical scanning [9] where the acquisition time is linearly dependent on the number of samples.

In this paper, we study the topic of adaptive depth sampling and depth map reconstruction. The importance of performing adaptive depth sampling is shown in Figure 1. First, we formulate the pipeline of joint adaptive depth sampling and depth map estimation. Then, we propose a deep learning (DL) based algorithm for adaptive depth sampling. We show that the proposed adaptive depth sampling algorithm can generalize well to many depth estimation algorithms. Finally, we demonstrate a state-of-the-art depth estimation accuracy compared to other existing algorithms.

Our contribution is summarized as follows:

- We propose an adaptive depth sensing framework which benefits from active sparse illumination depth sensors.
- We propose a superpixel segmentation based adaptive sampling mask prediction network and a differentiable sampling layer, which translates the estimated sampling locations ( $x, y$  coordinates) into a binary sampling mask. We also experimentally show better sampling performance is achieved compared to existing sampling methods.

Q. Dai is with Geomagical labs, Mountain View, CA, 94041, USA. F. Li is with Apple Inc, Cupertino, CA, 95014. O. Cossairt and A. K. Katsaggelos are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, 60208 USA.

Manuscript received Feb 3, 2022.

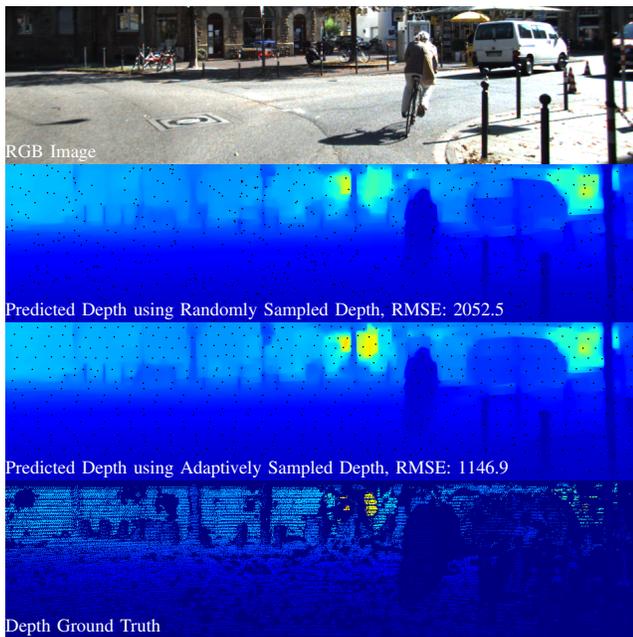


Fig. 1. LiDAR systems is able to capture accurate sparse depth map (bottom). By reducing the number of samples, we are able to reduce the LiDAR sensor bandwidth which potentially increase the capture frame rate. RGB image (top) can be fused with the captured sparse depth data and estimate a dense depth map. We demonstrate that choosing the sampling location is important to the accuracy of the estimated depth map. Under 0.25% sampling rate (with respect to the RGB image), using the same depth estimation method [6], the depth map estimated from the adaptively sampled sparse depth (third row) is more accurate than the depth map estimated from random samples (second row).

- We demonstrate that the proposed adaptive sampling method can generalize well to many depth estimation algorithms without fine tuning, thus establishing the effectiveness of the proposed sampling method. We also show that the trained adaptive depth sampling networks can generalize across different datasets. According to our knowledge, this is the first study in the literature that performs such generalization tests.
- We study the effect of capture time delay between the RGB image and the sampled depth map. We illustrate that the advantage of the proposed adaptive sampling method still holds if we take the temporal registration issue into account.

## II. RELATED WORK

In this section, we review work on algorithm-based depth estimation and sampling mask optimization and clarify the relationship of our proposed method to previous work.

### A. Depth Estimation

Given RGB images, early depth prediction methods relied on hand-crafted features and probabilistic graphics models. Karsch *et al.* [12], [13] estimate the depth based on querying an RGBD image database. A Markov random field model is applied in [14] to regress depth from a set of image features. Recently deep learning (DL) and convolutional neural

networks (CNNs) have been applied to learn the mapping from single RGB images to dense depth maps [15], [16], [17], [18], [19], [20], [21], [22]. These DL-based approaches achieve state-of-the-art performance because better features are extracted and better mappings are learned from large-scale datasets [23], [24], [25].

Given sparse depth measurements, traditional image filtering and interpolation techniques [26] can be applied to reconstruct the dense depth map. Hawe [27] and Liu [28] study the sparse depth map completion problem from the compressive sensing perspective. DL techniques have also been applied to the sparse depth completion problem. A sparse depth map can either be fed into conventional CNNs [29] or sparsity invariant CNNs [30]. When the sampling rate is low, the sparse depth map completion task is challenging.

If both RGB images and sparse depth measurements are provided, traditional guided filter approaches [31], [32] can be applied to refine the depth map. Optimization algorithms that promote depth map priors while maintaining fidelity to the observation are proposed in [33], [34], [35]. Various DL-based methods have been developed [29], [36], [6], [37], [38], [39], [40], [41]. During training and testing, most DL approaches are trained and tested using random or regular grid sampling masks. Because depth completion is an active research area, we do not want to limit our adaptive sampling method to a specific depth estimation method.

### B. Sampling Mask Optimization

Irregular sampling [42], [43], [44] is well studied in the computer graphics, image processing and computational imaging literature to achieve good representation of images, and we have witnessed its application in compressive sensing [45], ghost imaging [46], wireless imaging [47], quantitative phase imaging [48], Fourier ptychography [49], and etc. Making the sampling distribution adaptive to the signal can further improve representation performance. Eldar *et al.* [50] proposed a farthest point strategy which performs adaptive and progressive sampling of an image. Inspired by the lifting scheme of wavelet generation, several progressive image sampling techniques were proposed [51], [52]. Ramponi *et al.* [53] applied a measure of the local sample skewness. Lin *et al.* [54] utilized the generalized Ricci curvature to sample grey scale images as manifolds with density. A kernel construction technique is proposed in [55]. Taimori *et al.* [56] investigated space-frequency-gradient information of image patches for adaptive sampling.

Specific reconstruction algorithms are needed for each of these irregular or adaptive sampling methods [42], [50], [52], [51], [53], [54], [55], [56] to reconstruct the fully sampled signal. Furthermore, handcrafted features are applied to these sampling methods. Finally, these sampling techniques are all applied to the same modality (RGB or grey scale image). Recently, Dai *et al.* [57] applied DL technique to the adaptive sampling problem. The adaptive sampling network is jointly optimized with the image inpainting network. The sampling probability is optimized during training, and binarized during testing. Good performance is demonstrated for X-ray fluorescence (XRF) imaging at a sampling rate as low as 5%.

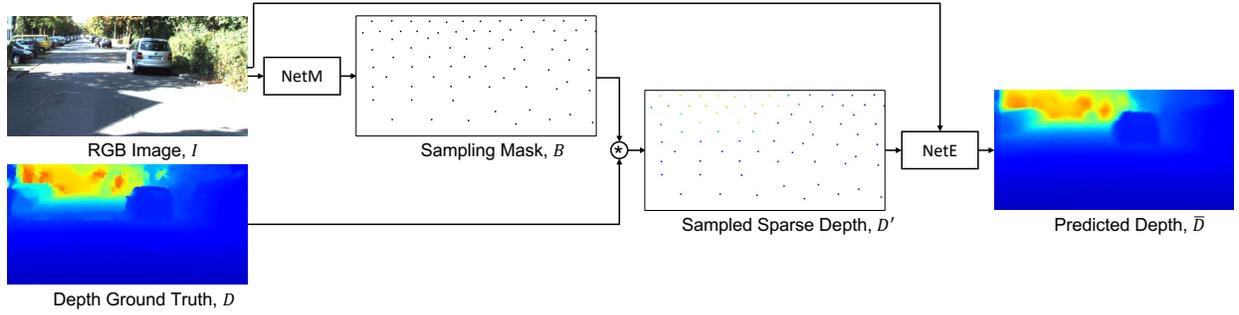


Fig. 2. The proposed pipeline contains two submodules, adaptive depth sampling Mask computation ( $NetM$ ) and depth Estimation ( $NetE$ ). The binary adaptive sampling mask is generated by  $NetM$  based on the RGB image. Then, the LiDAR system samples the scene based on this binary sampling mask and generates the sampled sparse depth map. Finally, both the RGB image and the sampled sparse depth map are input to  $NetE$  to estimate a dense depth map.

Kuznetsov *et al.* [58] predicted adaptive sampling maps jointly with reconstruction of Monte Carlo (MC) rendered images using DL. A differentiable render simulator with respect to the sampling map was proposed. Huijben *et al.* [59], [60] proposed a task adaptive compressive sensing pipeline. The sampling mask is trained with respect to a specific task and is fixed during imaging. Gumbel-max trick [61], [62] is applied to make the sampling layer differentiable.

All of the above DL-based sampling methods predict a per pixel sampling probability [57], [59], [60] or a sampling number [58]. Good sampling performance has not been demonstrated under extreme low sampling rates ( $< 1\%$ ). Directly enforcing priors on sampling locations is effective when the sampling rate is low. This requires the adaptive sampling network to predict sampling locations ( $(x, y)$  coordinates) directly and the sampling process to be differentiable. For the RGB and sparse depth adaptive sampling task, Wolff *et al.* [63] use the SLIC superpixel technique [64] to segment the RGB image and sample the depth map at the center of mass of each superpixel. A bilateral filtering based reconstruction algorithm is proposed to reconstruct the depth map. A spatial distribution prior is implicitly enforced by superpixel segmentation, resulting in good sampling performance under low sampling rates. The sampling and reconstruction methods are not optimized jointly, leaving room for improvement. In this paper, we show that jointly training recent DL-based superpixel sampling networks [65], [63] and depth estimation networks [42], [50], [52], [51], [53], [54], [55], [56] can be adapted to the problem of dense depth map estimation from sparse LiDAR data with improved accuracy. Bergman *et al.* [8] warp a uniform sampling grid to generate the adaptive sampling mask. The warping vectors are computed utilizing DL-based optical flow estimated from the RGB image. A spatial distribution prior is enforced by the initial uniform sampling grid. End-to-end optimization of the sampling and depth estimation networks is performed and good depth reconstruction is obtained under low sampling rates. In the pipeline of [8], there are 4 sub-networks, 2 for sampling and the other 2 for depth estimation. They are jointly trained but only the final depth estimation results are demonstrated. The whole pipeline is bulky and expensive. More importantly, it is hard to assess if

the improvement on depth estimation comes from the sampling part or the depth estimation part of the pipeline. In this paper, we decouple these two parts and study each individual module to better understand their contribution towards the final depth estimate. Finally, a bilinear sampling kernel is applied in [8] to make the optimization of the sampling locations differentiable. In contrast, we propose a novel differentiable relaxation of the sampling procedure and show its advantages over the bilinear sampling kernel.

### III. METHOD

#### A. Problem Formulation

As shown in Figure 2, the input RGB image is denoted by  $I$ . The mask generation network  $NetM$  produces a binary sampling mask  $B = NetM(I, c)$ , where  $c \in [0, 1]$  is the predefined sampling rate. Elements in  $B$  equal to 1 correspond to sampling locations and 0 to non-sampling location. The LiDAR system samples depth according to  $B$  and produces the measured sparse depth map  $D'$ . In synthetic experiments, if the ground truth depth map  $D$  is given, the measured sparse depth map  $D'$  is obtained according to

$$D' = D \odot B = D \odot NetM(I, c), \quad (1)$$

where  $\odot$  is the element-wise product operation. The reconstructed depth map  $\bar{D}$  is obtained by the depth estimation network  $NetE$ , that is,

$$\bar{D} = NetE(I, D') = NetE(I, D \odot NetM(I, c)). \quad (2)$$

The overall adaptive depth sensing and depth estimation pipeline is shown in Figure 2. End-to-end training can be applied on  $NetM$  and  $NetE$  jointly. The adaptive depth sampling strategy is learned by  $NetM$ , while  $NetE$  estimates the final dense depth map. An informative sampling mask is beneficial to depth estimation algorithms in general, not just to  $NetE$ . Given a limited depth sampling budget and an RGB image, we want to sample depth value on the ambiguous regions in a balanced way. During testing, we can replace the inpainting network  $NetE$  with other depth estimation algorithms. Network architectures and training details of  $NetE$  and  $NetM$  are discussed in the following subsections.

## B. Depth Estimation Network *NetE*

We use the network architecture in [29] for the depth estimation network. The network is an encoder-decoder pipeline. The encoder takes a concatenated  $I$  and  $D'$  as input (4 channels) and encodes them into latent features. The decoder takes the low spatial resolution feature representation and outputs the restored depth map  $\hat{D} = \text{NetE}(I, D')$ . Readers can refer to [29] for the detailed architecture of *NetE*.

Because method [29] is differentiable with respect to  $D'$  (unlike [37]) and its network architecture is standard without customized fusion modules [38], [6], [37], we choose it as *NetE* and jointly train *NetM* with it according to Figure 2. We found out that the trained *NetM* can generalize well to other depth estimation methods during testing.

## C. Sampling Mask Generation Network *NetM*

Existing irregular sampling techniques [50], [44] and adaptive depth sampling methods [8], [63] explicitly or implicitly make sampling points evenly distributed spatially. Such prior is important when the sampling rate is low. Inspired by the SLIC superpixel [64] based adaptive sampling method [63], we propose to utilize recent DL-based superpixel networks [66], [65] as *NetM*. As demonstrated in Figure 2, *NetM* adapts to the task of depth sampling after being jointly trained with *NetE*.

Superpixel with fully convolutional networks (FCN) [66] is one of the DL-based superpixel techniques. It predicts the pixel association map  $Q$  given an RGB image  $I$ . Its encoder-decoder network architecture is shown in Figure 3. Similar to the SLIC superpixel method [64], a combined loss that enforces similarity property of pixels inside one superpixel and spatial compactness is applied. Readers can refer to [66] for more details.

Given an RGB image  $I$  with spatial dimensions  $(H, W)$ , under the desired depth sampling rate  $c$ , we have  $N_p = H \cdot W$  pixels and  $N_s = c \cdot H \cdot W$  superpixels. The sampled depth location is the weighted mass center of each superpixel. We denote the subset of pixels as  $\mathcal{P} = \{\mathcal{P}_0, \dots, \mathcal{P}_{N_s-1}\}$ , where  $\mathcal{P}_i$  is a set of pixels associated with superpixel  $i$ . Pixel  $p$ 's CIELAB color property and  $(x, y)$  coordinates are denoted by  $\mathbf{f}(p) \in \mathbb{R}^3$  and  $\mathbf{c}(p) \in \mathbb{R}^2$ , respectively. CIELAB color space is used here as we follow the FCN [66] and SLIC superpixel [64] setup. The loss function is given by

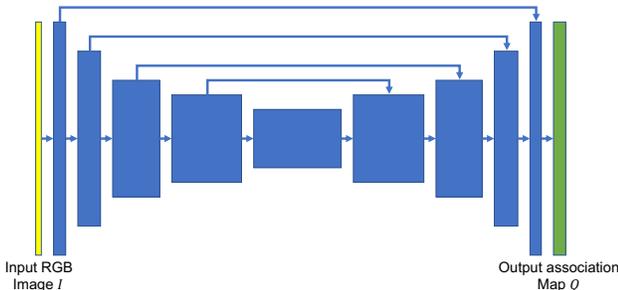


Fig. 3. Superpixel FCN [66]'s encoder-decoder network architecture.

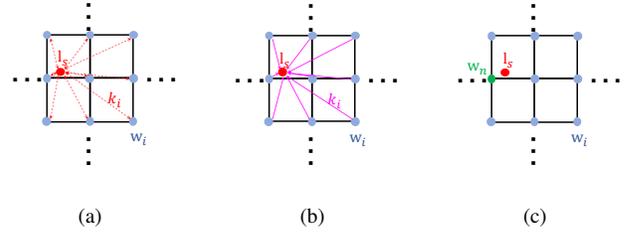


Fig. 4. Illustration of the sampling approximation. (a) We find a local window  $W$  of  $\mathbf{l}_s$  and compute distance  $\rho_i$ . (b) We represent  $\mathbf{l}_s$ 's depth value  $d_s$  as a linear combination of local window  $W$ 's depth values. (c) During testing, we sample the depth value at the nearest neighbour  $\mathbf{w}_n$  of  $\mathbf{l}_s$ .

$$\mathcal{L}_{SLIC}(\mathbf{f}, Q) = \sum_{p \in \mathcal{P}} \|\mathbf{f}(p) - \mathbf{f}'(p)\|_2 + m \|\mathbf{c}(p) - \mathbf{c}'(p)\|_2. \quad (3)$$

Here we have

$$\mathbf{u}_s = \frac{\sum_{p \in \mathcal{P}_s} \mathbf{f}(p) q_s(p)}{\sum_{p \in \mathcal{P}_s} q_s(p)}, \quad \mathbf{l}_s = \frac{\sum_{p \in \mathcal{P}_s} \mathbf{c}(p) q_s(p)}{\sum_{p \in \mathcal{P}_s} q_s(p)}, \quad (4a)$$

$$\mathbf{f}'(p) = \sum_{s \in \mathcal{N}_p} \mathbf{u}_s q_s(p), \quad \mathbf{c}'(p) = \sum_{s \in \mathcal{N}_p} \mathbf{l}_s q_s(p), \quad (4b)$$

where  $m$  is a weight balancing term between the CIELAB color similarity and spatial compactness,  $\mathcal{N}_p$  is the set of superpixels surrounding  $\mathbf{p}$ ,  $q_s(p)$  is the probability of a pixel  $p$  being associated with superpixel  $s$  and is derived from the associate map  $Q$ ,  $\mathbf{u}_s \in \mathbb{R}^3$  and  $\mathbf{l}_s \in \mathbb{R}^2$  are the color property and locations of superpixel  $s$ ,  $\mathbf{f}'(p) \in \mathbb{R}^3$  and  $\mathbf{c}'(p) \in \mathbb{R}^2$  are respectively the reconstructed color property and location of pixel  $p$ .

## D. Soft Sampling Approximation

Defined in Equation 4(a), we denote the collection of  $\mathbf{l}_s$ ,  $s = 0, \dots, N_s - 1$ , as  $S$ . Depth values at locations  $S$  would be measured during the depth sampling. In order to train *NetM* and *NetE* jointly, the sampling operation  $g$ , which computes the sampled sparse depth map  $D'$  from depth ground truth  $D$  and sampling location  $S$ ,  $D' = g(D, S)$ , needs to be differentiable with respect to  $S$ . Unfortunately, such sampling operation  $g$  is not differentiable in practice. Bergman *et al.* [8] apply a bilinear sampling kernel to differentially correlate  $S$  and  $D'$ . The computed gradients rely on the  $2 \times 2$  local structure of the ground truth depth map  $D$ . The computed gradients are not stable when the sampling location is sparse. Thus limited sampling performance is obtained. We propose a soft sampling approximation (SSA) strategy during training. SSA utilizes a larger window size compared to the bilinear kernel and achieves better sampling performance.

As shown in Figure 4, during training, given a sampling location  $\mathbf{l}_s \in S$ , we find a local  $h \times w$  window  $W$  around  $\mathbf{l}_s$ . The depth value  $d_s$  at  $\mathbf{l}_s$  is a weighted average of the depth values in  $W$ ,

$$d_s = \sum_{i \in \mathcal{N}_W} k_i d_i, \quad (5)$$

where  $\mathcal{N}_W$  includes the indices of all pixels in  $W$ ,  $\mathbf{w}_i$  is the  $i^{\text{th}}$  pixel's location in  $W$ ,  $d_i$  is the depth value of  $\mathbf{w}_i$ , the weights  $k_i$  are computed according to the Euclidean distance  $\rho_i$  between  $\mathbf{l}_s$  and  $\mathbf{w}_i$ , scaled by a temperature parameter  $t$ ,

$$k_i = \frac{e^{-\rho_i^2/t^2}}{\sum_{j \in \mathcal{N}_W} e^{-\rho_j^2/t^2}}. \quad (6)$$

When the temperature parameter  $t \rightarrow 0$ , the sampled depth value  $d_s$  is equal to the depth value  $d_n$  of the nearest pixel  $\mathbf{w}_n$ . When  $t$  is large, the soft sampled depth value  $d_s$  is different from  $d_n$ . We gradually reduce  $t$  during the training process. During testing, we find the nearest neighbor pixel  $\mathbf{w}_n$  of  $\mathbf{l}_s$  and sample the depth value  $d_n$  at  $\mathbf{w}_n$ .

### E. Training Procedures

Given the training dataset consisting of the aligned RGB image  $I$  and the ground truth depth map  $D$ , we first train *NetE* by minimizing the depth loss,

$$\mathcal{L}_{depth} = \|D - \text{NetE}(I, D')\|_2, \quad (7)$$

where  $D'$  is obtained by applying a random sampling mask on  $D$  with sampling rate  $c$ .

Then we initialize the superpixel network *NetM* using the RGB image  $I$ .  $\mathcal{L}_{SLIC}$  is minimized according to Equation 3. The initialized *NetM* approximates the SLIC superpixel segmentation on RGB image. If we sample the depth value on  $\mathbf{l}_s$  of each superpixel, the sampling pattern would be similar to [63].

Finally, we freeze *NetE* and train *NetM* in Figure 2 by minimizing

$$\mathcal{L} = \mathcal{L}_{depth} + q \cdot \mathcal{L}_{SLIC}, \quad (8)$$

where  $q$  is the weighting terms of  $\mathcal{L}_{SLIC}$ . The SSA trick shown in Figure 4 is applied and the temperature parameter  $t$  gradually decreases during training.

We fix *NetE* when training *NetM*. Optimizing *NetE* and *NetM* simultaneously would obtain *better* depth reconstruction accuracy [8]. However, similarly to [57], we would utilize other depth estimation methods than *NetE* during testing. We want to make the adaptive depth sampling mask be general and applicable to many depth estimation algorithms.

## IV. EXPERIMENTAL RESULTS

### A. Implementation Details

We use both the KITTI depth completion dataset [30] and the NYU-Depth-V2 dataset [23] for our experiments. The KITTI depth completion dataset consists of aligned ground truth depth maps (from LiDAR sensor) and RGB images. The original KITTI training and validation set split is applied. There are 42949 and 3426 frames in the training and testing sets, respectively. We only use the bottom center crop  $240 \times 960$  of the images because the LiDAR sensor has no measurements at the upper part of the images. The NYU-Depth-V2 dataset consists of RGB and depth images captured by a Microsoft Kinect. 48004 synchronized RGB-depth image pairs are used for training. 654 RGB-Depth image pairs from

the small labeled test dataset are used for testing. Following [29], the original frames of resolution  $480 \times 640$  are down sampled to half resolution, producing a final resolution of  $240 \times 320$ .

For the KITTI depth completion dataset, the ground truth depth maps are not dense because they are measured by a velodyne LiDAR device. In order to perform adaptive depth sampling, we need dense depth maps to sample from. Similarly to [8], a traditional image inpainting algorithm [31] is applied to densify the depth ground truth. During evaluation, we compare the estimated dense depth maps to the original sparse ground truth depth maps. For the NYU-Depth-V2 dataset, the raw depth values are projected onto the synchronized RGB images and inpainted using a bilateral filter method available in the official toolbox.

During the training of *NetE*, we follow Ma *et al.*'s setup [29]. The batch size is set equal to 16. The ResNet encoder is initialized with pretrained weights using the ImageNet dataset [67]. Stochastic gradient descent (SGD) optimizer with momentum 0.9 is used. We train 100 epochs in total. The learning rate is set to be equal to 0.01 at first and reduced by 80% at every 25 epochs. *NetE* is trained individually under different sampling rates  $c = 1\%, 0.25\%$  and  $0.0625\%$  using random sampling masks. We also train FusionNet [6] and SSNet [36] under different sampling rates using random sampling masks. The same training procedure in their original papers are used. They serve as alternative depth estimation methods.

We test the proposed sampling algorithm under 3 sampling rates,  $c = 1\%, 0.25\%$  and  $0.0625\%$ . For the KITTI depth completion dataset, they correspond to  $N_s = 2304, 576$  and 144 depth samples (superpixels) in the  $240 \times 940$  image. For the NYU-Depth-V2 dataset, they correspond to  $N_s = 768, 192$  and 48 depth samples (superpixels) in the  $240 \times 320$  image. *NetM* is configured to output the desired number of samples. During the training of *NetM*, we pretrain it using the SLIC loss.  $m$  in Equation 3 is set equal to be 1. ADAM optimizer [68] is applied. Learning rate is set to be  $5 \times 10^{-5}$ . We train 100 epochs in total.

After *NetM* is initialized, we finally jointly train *NetM* and *NetE* according to Figure 2. Loss defined in Equation 8 is optimized with  $q$  equal to  $10^{-6}$ , resulting in  $\mathcal{L}_{depth}$  being equal to about 10 times of  $q \cdot \mathcal{L}_{SLIC}$  in value. The window size of the soft depth sampling module is equal to 5. Temperature  $t$  defined in Equation 6 decreases from 1.0 to 0.1 linearly during training. We experimentally find that *NetM*'s performance is not sensitive to the SSA related settings, such as the window size, initial temperature and temperature decay policy. Batch size is set equal to 8 and this is the largest batch size we can use for both *NetM* and *NetE* in an NVIDIA 2080Ti GPU (11GB memory). As discussed in Section III-E, *NetE* is fixed during the training to make *NetM* generalize well to other depth estimation methods. Learning rate of *NetM* is assigned to be equal to  $10^{-4}$  and is reduced by 50% every 10 epochs. SGD optimizer with momentum 0.9 is used. We found that 50 epochs in total are adequate for converge.

Our proposed adaptive depth sampling framework is implemented in PyTorch and our implementation is available at:

	MAE (mm)											
	c=1%				c=0.25%				c=0.0625%			
	FusionNet	SSNet	<i>NetE</i>	Colorization	FusionNet	SSNet	<i>NetE</i>	Colorization	FusionNet	SSNet	<i>NetE</i>	Colorization
Random	324.8	466.6	425.7	764.6	488.6	654.9	557.1	1390.7	798.5	1021.1	779.4	2517.5
Uniform Grid	301.4	450.2	398.2	694.6	439.0	598.0	516.4	1257.3	692.5	843.2	715.3	2247.3
Poisson [44]	324.1	451.8	409.6	711.5	455.8	621.2	537.5	1314.1	736.2	901.0	743.1	2428.4
SPS [63]	297.2	439.9	388.1	<i>654.3</i>	436.9	594.2	507.8	<i>1197.2</i>	713.8	865.2	724.5	<b>2175.9</b>
DAL [8]	295.8	447.4	390.1	683.4	432.5	599.1	504.8	1239.7	694.4	838.4	710.2	2230.7
FCN [66]	298.2	440.8	390.0	672.5	426.8	<i>587.4</i>	<i>498.5</i>	1202.7	683.8	833.6	698.2	2227.6
<i>NetM – NYU</i>	<i>291.3</i>	<i>435.9</i>	<i>382.0</i>	<i>662.7</i>	<i>425.9</i>	590.1	499.2	1212.4	<i>657.4</i>	799.7	<i>678.9</i>	2221.0
<i>NetM</i>	<b>285.0</b>	<b>423.1</b>	<b>380.1</b>	<b>656.2</b>	<b>404.3</b>	<b>562.2</b>	<b>477.5</b>	<b>1189.2</b>	<b>634.9</b>	<b>778.0</b>	<b>652.2</b>	2265.8

	RMSE (mm)											
	c=1%				c=0.25%				c=0.0625%			
	FusionNet	SSNet	<i>NetE</i>	Colorization	FusionNet	SSNet	<i>NetE</i>	Colorization	FusionNet	SSNet	<i>NetE</i>	Colorization
Random	1060.0	1221.6	1294.8	1984.4	1476.0	1709.9	1704.3	3087.3	2135.6	2505.6	2262.61	4749.8
Uniform Grid	988.4	1139.2	1207.8	1840.9	1359.1	1570.2	1566.0	2854.2	1946.4	2132.5	2101.7	4315.9
Poisson [44]	1010.1	1140.2	1193.3	1844.8	1375.6	1589.1	1596.8	2897.3	2013.1	2256.0	2151.4	4508.6
SPS [63]	1039.1	1124.3	1160.5	<i>1742.9</i>	1360.1	1559.7	1553.1	<i>2718.1</i>	1993.8	2215.0	2141.5	4161.1
DAL [8]	969.9	1115.1	1177.5	1784.3	1336.1	1548.1	1532.1	2772.6	1937.6	2128.3	2085.7	4242.3
FCN [66]	982.8	1123.6	1165.1	1778.7	1324.1	<i>1530.4</i>	1517.1	2728.8	1893.9	2103.3	2046.2	4188.5
<i>NetM – NYU</i>	<i>957.2</i>	<i>1095.1</i>	<i>1139.2</i>	1744.3	<i>1322.0</i>	1532.7	<i>1514.7</i>	2743.1	<i>1849.6</i>	<i>2022.1</i>	<i>2010.6</i>	<i>4031.8</i>
<i>NetM</i>	<b>939.4</b>	<b>1074.9</b>	<b>1131.3</b>	<b>1725.9</b>	<b>1239.7</b>	<b>1436.8</b>	<b>1422.4</b>	<b>2584.5</b>	<b>1732.4</b>	<b>1930.5</b>	<b>1896.7</b>	<b>3972.9</b>

TABLE I

DEPTH SAMPLING AND ESTIMATION RESULTS ON KITTI DEPTH COMPLETION DATASET. RANDOM, POISSON [44], UNIFORM GRID, SPS [63], DAL [8], FCN [66] AND PROPOSED *NetM* SAMPLING STRATEGIES ARE COMPARED UTILIZING *NetE* [29], FUSIONNET [6], SSNET [36], AND COLORIZATION [31] DEPTH ESTIMATION ALGORITHMS. MAE AND RMSE METRICS ARE REPORTED. BEST RESULTS ARE SHOWN IN BOLD. SECOND BEST RESULTS ARE SHOWN IN ITALIC. THE RESULTS SHOWN ARE AVERAGED OVER A SET OF 3426 TEST FRAMES.

	MAE (mm)											
	c=1%				c=0.25%				c=0.0625%			
	FusionNet	SSNet	<i>NetE</i>	Colorization	FusionNet	SSNet	<i>NetE</i>	Colorization	FusionNet	SSNet	<i>NetE</i>	Colorization
Random	39.27	59.74	94.62	74.14	73.41	104.82	99.63	146.08	146.51	186.23	151.33	283.85
Uniform Grid	36.39	56.52	89.09	65.45	65.35	98.79	91.95	128.90	123.90	151.03	134.33	244.17
Poisson [44]	35.84	54.09	89.55	66.17	65.54	96.91	94.99	135.73	140.44	177.40	151.41	298.46
SPS [63]	<i>34.25</i>	<i>52.79</i>	<i>86.50</i>	<b>59.47</b>	<i>61.43</i>	<b>92.68</b>	<i>88.95</i>	<b>119.76</b>	126.06	<i>153.82</i>	<i>134.28</i>	<b>238.34</b>
DAL [8]	36.40	55.64	86.69	65.44	65.56	98.41	92.17	129.13	127.15	154.91	135.54	244.15
FCN [66]	35.20	53.66	87.43	63.27	62.62	94.52	90.03	124.20	123.60	<b>150.87</b>	134.52	<i>239.31</i>
<i>NetM – KITTI</i>	34.78	53.63	87.25	64.22	62.96	94.36	90.02	128.73	<i>123.44</i>	154.63	134.59	252.81
<i>NetM</i>	<b>34.08</b>	<b>52.64</b>	<b>86.38</b>	<i>62.54</i>	<b>61.31</b>	<i>93.18</i>	<b>88.85</b>	<i>124.05</i>	<b>123.10</b>	154.41	<b>133.64</b>	252.03

	RMSE (mm)											
	c=1%				c=0.25%				c=0.0625%			
	FusionNet	SSNet	<i>NetE</i>	Colorization	FusionNet	SSNet	<i>NetE</i>	Colorization	FusionNet	SSNet	<i>NetE</i>	Colorization
Random	98.05	116.79	167.91	150.76	156.80	182.50	192.17	249.83	255.80	303.42	269.22	425.56
Uniform Grid	94.00	111.01	152.33	138.36	143.39	169.00	178.29	224.29	228.21	249.13	245.00	370.02
Poisson [44]	91.07	106.34	152.11	137.53	142.22	165.17	180.39	230.84	245.22	278.14	259.86	436.13
SPS [63]	<i>88.19</i>	<b>102.89</b>	<i>144.55</i>	<b>126.23</b>	<i>136.01</i>	<b>158.49</b>	<i>169.80</i>	<b>209.53</b>	228.45	252.40	244.14	<b>357.96</b>
DAL [8]	93.87	110.24	149.41	138.13	143.56	168.79	177.96	225.04	231.91	254.32	246.99	368.31
FCN [66]	90.71	106.21	147.56	132.95	137.73	161.93	173.05	217.14	226.15	<i>247.40</i>	242.05	<i>361.41</i>
<i>NetM – KITTI</i>	89.52	105.57	146.37	134.58	138.00	161.05	172.81	221.45	223.65	248.00	239.59	374.80
<i>NetM</i>	<b>87.61</b>	<i>103.37</i>	<b>143.94</b>	<i>130.60</i>	<b>134.60</b>	<i>158.69</i>	<b>169.41</b>	<i>214.63</i>	<b>222.59</b>	<b>247.05</b>	<b>238.97</b>	375.03

TABLE II

DEPTH SAMPLING AND ESTIMATION RESULTS ON NYU-DEPTH-V2 DATASET. RANDOM, POISSON [44], UNIFORM GRID, SPS [63], DAL [8], FCN [66] AND PROPOSED *NetM* SAMPLING STRATEGIES ARE COMPARED UTILIZING *NetE* [29], FUSIONNET [6], SSNET [36], AND COLORIZATION [31] DEPTH ESTIMATION ALGORITHMS. MAE AND RMSE METRICS ARE REPORTED. BEST RESULTS ARE SHOWN IN BOLD. SECOND BEST RESULTS ARE SHOWN IN ITALIC. THE RESULTS SHOWN ARE AVERAGED OVER A SET OF 654 TEST FRAMES.

<https://github.com/usstdq/adaptive-depth-sensing>.

### B. Performance on Adaptive Depth Sensing and Estimation

For the adaptive depth sampling and estimation task, we demonstrate the advantages of our proposed adaptive sampling mask *NetM*, over the use of random, uniform grid and Poisson [44] sampling masks, as well as other state-of-the-art adaptive depth sampling methods, such as SuperPixel Sampler (SPS) [63] and Deep Adaptive Lidar (DAL) [8].

*NetM* is initialized by RGB images according to FCN [66]. To show the effectiveness of the proposed *NetE* and *NetM* joint training method, we also compare with the sampling mask computed by the initialized *NetM*. The sampling method is denoted as FCN.

To illustrate *NetM* can generalize across datasets, we train *NetM* on the NYU-Depth-V2 dataset and test it on the KITTI dataset. The depth sampling method is noted as *NetM – NYU* when testing on the KITTI dataset. Similarly, we train *NetM* on the KITTI dataset and test it on the NYU-Depth-V2 dataset. The sampling method is noted as *NetM – KITTI* when testing on the NYU-Depth-V2 dataset. Noted that *NetM* is fully convolutional, so it is straightforward to test on different image resolution.

Random, Uniform Grid, Poisson, SPS [63], DAL [8], FCN [66] and proposed *NetM* (including *NetM – KITTI* and *NetM – NYU*) sampling methods are applied to the test images. Sampling rates  $c = 1\%$ ,  $0.25\%$  and  $0.0625\%$  are tested. For the depth estimation methods, DL-based methods

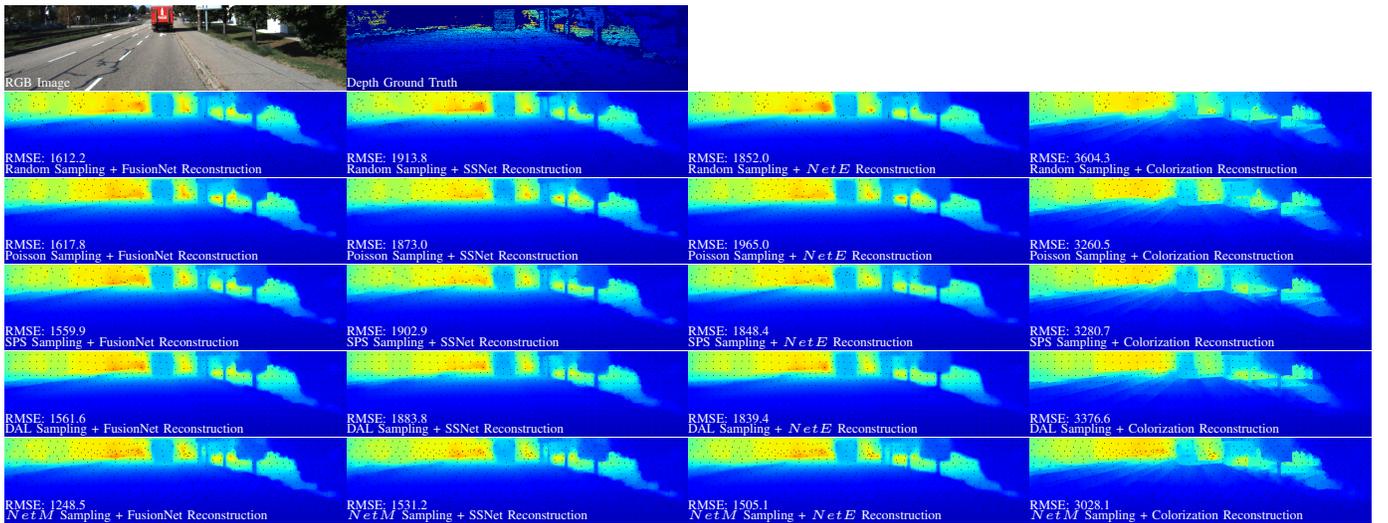


Fig. 5. Visual comparison of the estimated depth maps. Random, Poisson, SPS, DAL, and *NetM* sampling masks at sampling rate  $c = 0.25\%$  are applied and shown in the  $2^{nd} - 6^{th}$  rows, respectively. The first row includes the RGB image and the ground truth depth map. Sampling locations are indicated using black dots. FusionNet, SSNet, *NetE* and Colorization depth estimation methods are used to perform depth estimation and generate the depth maps of  $1^{st} - 4^{th}$  columns, respectively. RMSE is computed for each depth map with respect to the ground truth depth map.

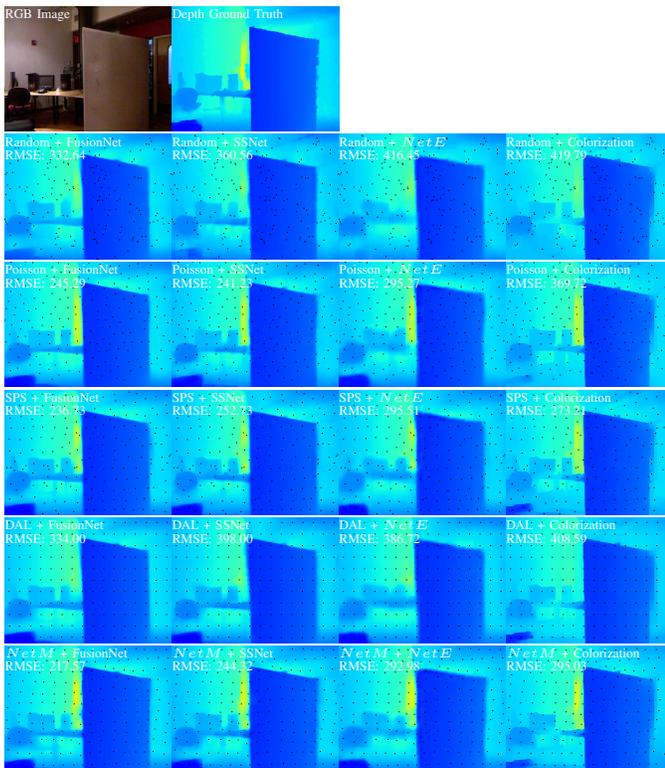


Fig. 6. Visual comparison of the estimated depth maps. Random, Poisson, SPS, DAL, and *NetM* sampling methods at sampling rate  $c = 0.25\%$  are applied and shown in the  $2^{nd} - 6^{th}$  rows, respectively. The first row includes the RGB image and the ground truth depth map. Sampling locations are indicated using black dots. FusionNet, SSNet, *NetE* and Colorization depth estimation methods are used to perform depth estimation and generate the depth maps of  $1^{st} - 4^{th}$  columns, respectively. RMSE is computed for each depth map with respect to the ground truth depth map.

*NetE* [29], FusionNet [6], SSNet [36] and traditional method Colorization [31] are used to estimate the fully sampled depth

map from the sampled depth map and RGB image. It's noted that all the DL-based depth estimation methods are trained using random sampling masks and the same training set of either KITTI depth completion or NYU-Depth-V2 dataset.

For the KITTI depth completion dataset, the average Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) over all 3426 test frames are shown in Table I. First, under all three sampling rates, the proposed *NetM* mask outperforms the random, Poisson, Uniform Grid, SPS, DAL and FCN masks consistently over all depth estimation methods in terms of RMSE and MAE. This demonstrates the effectiveness of our proposed adaptive depth sampling network. Furthermore, *NetM* is jointly trained with *NetE* and it still performs well with other depth estimation methods, demonstrating that it can generalize well to other depth estimation methods. The performance advantage of *NetM* is not tied to any specific depth estimation method. Finally, it can be concluded that the smaller the sampling rate, the larger the advantage of *NetM* compared to other sampling algorithms. This implies that *NetM* is able to handle challenging depth sampling tasks (extremely low sampling rates).

The depth sampling and reconstruction performance on the NYU-Depth-V2 dataset are shown in Table II. Similar conclusions as the ones from the KITTI depth completion dataset can be drawn. It is noticed that SPS method is comparable to *NetM*. NYU-Depth-V2 is an indoor dataset with a maximum 10m depth range. The variance of the depth map is small compared to the KITTI depth completion dataset, so the even spatial distribution prior in SPS works well here. Moreover, NYU-Depth-V2 dataset is captured by Microsoft Kinect. The ground truth depth maps have low spatial resolution ( $240 \times 320$ ) and are relative noisy. Thus some improvement on fine details in the estimated depth map is not reflected. Nevertheless, *NetM* outperforms SPS when sampling rate is low and more advanced DL-based depth

completion algorithms are applied.

According to Table I and Table II, FusionNet [6] has the best depth estimation accuracy under various of sampling masks. FusionNet extracts both global and local information and is more complex than *NetE*. Depth estimation from RGB image and sparse depth input is an active research topic. *NetM* is able to generalize with other depth estimation methods than *NetE*, so it is able to benefit from the latest depth estimation algorithms.

The visual quality comparison of various sampling strategies and depth estimation methods is shown in Figure 5 and Figure 6. For sampling rate equal to  $c = 0.25\%$ , we can observe the advantages of the proposed *NetM* mask over all other sampling masks by comparing the resulting depth maps by the same depth estimation algorithm. In Figure 5, *NetM* samples densely around the end of the road, trees and billboard, resulting in accurate depth estimation in such areas. Compared to other adaptive sampling algorithms, such as SPS and DAL, *NetM* samples more densely on distance objects, making the estimated depth more accurate. SPS uses SLIC [64] to segment the RGB image and such segmentation can not obtain distance information from the RGB images. DAL estimates a smooth motion field to warp a regular sampling grid. When the scene is complicated, it is not flexible enough to warp a regular sampling grid to optimal location. In Figure 6, *NetM* samples the distant vertical structure, as well as the table and chair on the left side. More detailed depth map reconstruction in those areas is obtained. Compared to SPS, *NetM* tends to sample uniformly on the wall regions, while still capture the shape of the foreground objects.

According to Table I, *NetM - NYU* is the second best performer in the KITTI dataset. From Table II, in the NYU-Depth-V2 dataset, *NetM - KITTI* outperforms all the other sampling methods except SPS and *NetM*. This demonstrates that the proposed *NetM* is able to generalize across different datasets. We visualize the sampling location difference between *NetM - NYU* and *NetM* in the KITTI dataset in Figure 7. It can be found that *NetM* samples densely on the upper part of the image compared to *NetM - NYU*. *NetM - NYU* does not learn the prior knowledge that upper part of the image has larger depth values and should be sampled more densely from the indoor NYU-Depth-V2 dataset. However, *NetM - NYU* is still able to sample on such objects as the bus, cyclist and poles. The sampling location difference between *NetM - KITTI* and *NetM* on the NYU-Depth-V2 dataset is shown in Figure 8. *NetM - KITTI* learns the prior knowledge that upper part of the image is more important from the KITTI dataset and samples densely on the upper part of the image, resulting sub-optimal reconstruction accuracy compared to *NetM*.

*NetM* is initialized with RGB images trained FCN [66] superpixel network using the SLIC loss (Equation 3). SPS [63] uses the SLIC superpixel technique to segment the RGB images. Sampling locations are determined by the weighted mass center of superpixels. Different superpixel segmentations result in different sampling quality. In Figure 9, we visualize the superpixel segmentation results and the derived sampling locations for SLIC, FCN and *NetM* when  $c = 0.0625\%$ .

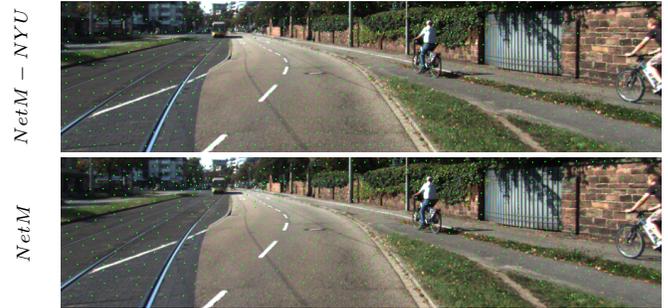


Fig. 7. Visual comparison of *NetM - NYU* and *NetM*'s sampling location on the KITTI dataset. Sampling locations are plotted in green.



Fig. 8. Visual comparison of *NetM - KITTI* and *NetM*'s sampling location on the NYU-Depth-V2 dataset. Sampling locations are plotted in green.

$c$	Kernel	FusionNet		SSNet		<i>NetE</i>	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
1%	Bilinear	290.5	948.4	436.3	1086.7	383.0	1138.8
	SSA	285.0	939.4	423.1	1074.9	380.1	1131.3
0.25%	Bilinear	431.1	1285.3	590.1	1487.5	494.1	1466.0
	SSA	404.3	1239.7	562.2	1436.8	477.5	1422.4
0.0625%	Bilinear	809.1	2161.4	989.2	2457.4	781.6	2253.0
	SSA	634.9	1732.4	778.0	1930.5	652.2	1896.7

TABLE III  
USING SSA AND BILINEAR KERNEL DURING TRAINING RESULTS  
DIFFERENT SAMPLING QUALITY OF *NetM*.

SLIC and FCN segment the input RGB image based on the color similarity and preserve spatial compactness. The segmentation density is spatially homogeneous. *NetM* is jointly trained with *NetE*, thus it has knowledge of distance given the RGB input image. Distance objects in the image are sampled denser. It also segments sparsely the pavement and grass areas. Such near objects as cars are segmented denser compared to the pavement and grass areas. We also observe that *NetM* segmentation does not preserve color pixel boundaries as well as FCN, which is expected as *NetM* also minimizes the depth estimation loss besides the SLIC loss in Equation 8. According to FCN and *NetM*'s reconstruction accuracy in Table I and Table II, it can be found that *NetM* always outperforms FCN. This demonstrates the effectiveness of the proposed *NetM* training mechanism (Equation 8).

### C. Effectiveness of Soft Sampling Approximation

In Section III-D, we propose the use of SSA to make the sampling process differentiable during training. Such differentiable sampling approximation is necessary to jointly train *NetM* with *NetE*. Compared to the  $2 \times 2$  bilinear



Fig. 9. Visual comparison of different superpixel segmentation and sampling location. Segmentation boundaries are plotted in blue and sampling locations are plotted in green.

kernel based differentiable sampling in [8], the proposed SSA provides better sampling performance. In order to show the advantages of SSA, we replace the SSA sampling of *NetM* by the bilinear kernel based sampling and perform the exact same training procedures. As demonstrated in Table III, the lower the sampling rate, the bigger the advantage of SSA over the bilinear kernel sampling. When sampling points are sparse, the gradients derived from a  $2 \times 2$  local window are too small to train *NetM* effectively. We empirically found that the  $5 \times 5$  window size for SSA provides reasonable sampling performance under all sampling rates.

#### D. End To End Depth Estimation Performance

In Table I, FusionNet [6] achieves the best depth estimation performance under various of sampling masks. The proposed global and location information fusion is effective and the network size is considerably larger than *NetE* [29]. Best depth sampling and estimation results are obtained using *NetM* sampling and FusionNet depth estimation under all sampling rates. It is noted that *NetM* is trained jointly with *NetE* and FusionNet is trained using random masks. Similar to DAL, *NetM* and FusionNet can also be optimized simultaneously. Starting from the *NetE* trained *NetM* and random mask trained FusionNet, we alternatively train *NetM* and FusionNet

$c$	Sampling	Reconstruction	MAE	RMSE
1%	SPS	SPS	406.3	1264.2
	DAL	DAL	550.3	1566.7
	<i>NetM</i>	FusionNet	285.0	939.4
	<i>NetM*</i>	FusionNet*	<b>284.6</b>	<b>932.6</b>
0.25%	SPS	SPS	812.7	2192.3
	DAL	DAL	597.7	1667.8
	<i>NetM</i>	FusionNet	404.3	1239.7
	<i>NetM*</i>	FusionNet*	<b>402.9</b>	<b>1229.4</b>
0.0625%	SPS	SPS	1668.6	3891.9
	DAL	DAL	789.1	2104.0
	<i>NetM</i>	FusionNet	634.9	1732.3
	<i>NetM*</i>	FusionNet*	<b>631.5</b>	<b>1721.1</b>

TABLE IV  
END TO END DEPTH ESTIMATION RESULTS COMPARISON.

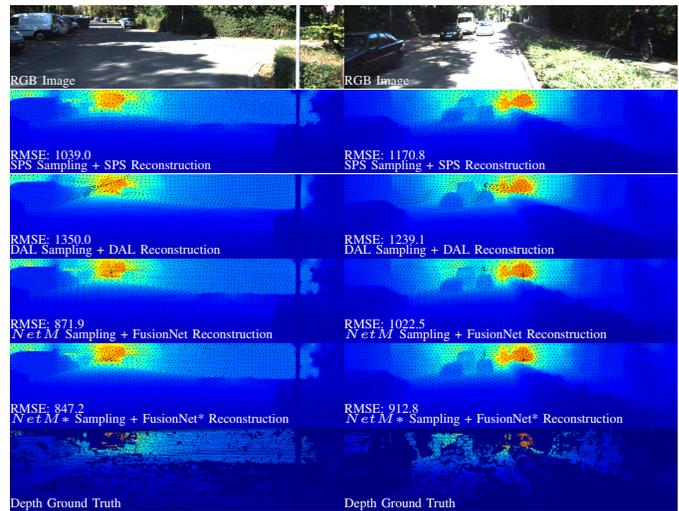


Fig. 10. Visual comparison of different depth sampling and estimation methods.

and denote the trained networks by *NetM\** and FusionNet\*, respectively. The joint depth sampling and reconstruction results are shown in Table IV. We also compare with the sampling and reconstruction methods proposed in SPS and DAL. *NetM\** with FusionNet\* slightly outperforms *NetM* with FusionNet and achieves the best accuracy. Utilizing random sampling masks during the training of depth estimation methods (FusionNet, SSNet, *NetE*) makes the methods robust to other sampling patterns in testing. We also found that *NetM* trained using different depth estimation methods has similar sampling patterns. So simultaneously training the sampling and reconstruction networks improves the results slightly.

In Figure 10, we visually compare the end to end depth sampling and reconstruction results. In the 2 test scenes, *NetM\** with FusionNet\* properly sample and reconstruct distant and thin objects, resulting in the best accuracy compared to other methods. With the developing depth estimation algorithms, we can integrate better depth estimation methods into our system. We show in Section IV-B that the performance advantages of *NetM* can generalize well to other than *NetE* depth estimation methods.

#### E. Running Speed

Apart from the superior adaptive depth sampling quality to other state-of-the-art methods, the proposed sampling method also has advantages in fast computation efficiency. In this section we evaluate the computation efficiency of different methods. The test images from both the KITTI depth completion dataset ( $240 \times 960$ ) and the NYU-Depth-V2 dataset ( $240 \times 360$ ) are used. Non DL-based methods, Poisson and SPS, are tested using one Intel i9-9820X CPU with 64GB memory. DL-based methods, DAL and *NetM*, are tested using the same CPU as well as one NVIDIA 2080Ti GPU with 11GB memory. We also measure the floating point operations (FLOPs) and number of parameters of both models. All methods are tested on the same test image set under 3

		KITTI Depth Completion								
		c=1%			c=0.25%			c=0.0625%		
	device	time (ms)	FLOPs	#params	time (ms)	FLOPs	#params	time (ms)	FLOPs	#params
Poisson	CPU	68.6	-	-	18.0	-	-	5.1	-	-
SPS	CPU	399.8	-	-	301.0	-	-	244.5	-	-
DAL	CPU	460.9	149.8G	42.4M	439.3	149.8G	42.4M	464.8	149.8G	42.4M
DAL	GPU	97.9	149.8G	42.4M	44.5	149.8G	42.4M	34.2	149.8G	42.4M
<i>NetM</i>	CPU	143.8	32.0G	2.3M	143.7	32.0G	2.3M	147.5	32.0G	2.3M
<i>NetM</i>	GPU	20.1	32.0G	2.3M	21.5	32.0G	2.3M	23.4	32.0G	2.3M

		NYU-Depth-V2								
		c=1%			c=0.25%			c=0.0625%		
	device	time (ms)	FLOPs	#params	time (ms)	FLOPs	#params	time (ms)	FLOPs	#params
Poisson	CPU	22.3	-	-	6.4	-	-	2.0	-	-
SPS	CPU	126.1	-	-	88.7	-	-	74.6	-	-
DAL	CPU	159.8	50.0G	42.4M	158.0	50.0G	42.4M	164.7	50.0G	42.4M
DAL	GPU	42.8	50.0G	42.4M	24.9	50.0G	42.4M	21.3	50.0G	42.4M
<i>NetM</i>	CPU	46.0	5.7G	2.3M	48.5	5.7G	2.3M	45.6	5.7G	2.3M
<i>NetM</i>	GPU	9.9	5.7G	2.3M	10.6	5.7G	2.3M	10.1	5.7G	2.3M

TABLE V

COMPUTATION TIME, FLOPS AND MODEL SIZE COMPARISON BETWEEN THE POISSON, SPS, DAL AND *NetM* SAMPLING METHODS. NOTICE THAT POISSON AND SPS ARE TESTED ON CPU, WHILE DL-BASED METHODS DAL AND *NetM* ARE TESTED ON BOTH CPU AND GPU.

different sampling rates for 100 times and the average run time in milliseconds (ms) are reported in Table V. It's noticed that comparing to DAL, *NetM* has smaller model size and faster run time. Also *NetM*'s run time is almost constant under different sampling rates, different from the other three methods. Such fast computation efficiency property makes *NetM* practical for a real time adaptive depth sensing system. *NetM* is also faster than Poisson, SPS and DAL when running on the same CPU.

### F. Temporal Registration Issue

All the experiments above assume the RGB images and the sampled depth maps are captured at the exact same time instant. Due to the system response time, *NetM* computation time, etc, there are temporal registration issue between the sampled depth maps and the RGB images in practice. To make the adaptive depth sampling method practical, it is important to understand how *NetM*'s sampling performance degrades with respect to the capture time delay  $\Delta t$  (between the sampled depth map and RGB image). Noted that the Random, Uniform Grid and Poisson sampling masks are free from such temporal registration issue, because the sampling masks are independent from the RGB images. In this section, we performed 2 sets of experiments to simulate the temporal registration issues.

For the first set of experiments, with 0.25% sampling rate on the KITTI dataset, during the depth sampling process at frame  $t$ , the sampling mask is computed using RGB image at frame  $t - \Delta t$ , to simulate the capture time difference directly. The KITTI dataset captures synchronized RGB and Depth data every 100ms. We simulate such time delay  $\Delta t$  from 0 to 500ms and use *NetE* to estimate the dense depth maps. The MAE and RMSE error is shown in Figure 11. It can be found that *NetM*'s sampling performance is fairly stable when the capture time difference increases, up to 500ms. One reason is that the far objects' motion is small given the temporal perturbation. Another reason is that the structure of the scene is relatively static in consecutive frames.

For the second set of experiments, also with 0.25% sampling rate on the KITTI dataset, additional perturbation is added on *NetM* predicted sampling location. The perturbation is done

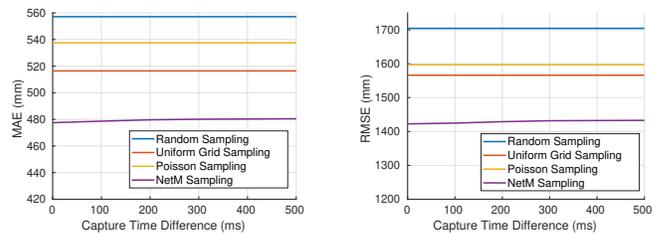


Fig. 11. MAE/RMSE with respect to the increasing capture time difference. Random, Uniform Grid and Poisson sampling masks do not need RGB image as input, thus they are free from the temporal registration issue.

by adding uniform distribution noise under different ranges to the sampling location. As shown in Figure 12, it can be found that under the MAE and RMSE metrics, even with  $+/- 15$  pixels perturbation ( $240 \times 960$  full image resolution) on the sampling location, *NetM* still outperforms Random, Uniform Grid and Poisson sampling masks (no perturbation added), under the same depth estimation method *NetE*. Such large pixel perturbation serves as a challenging test case because far objects' motion can not reach this level in practice.

According to the above 2 sets of experiments, the sampling performance advantage still holds if we take the capture time difference into consideration. The difficulty in sampling fast moving objects is one of the limitations of the proposed adaptive sampling approach. Such techniques as motion prediction can be applied to compensate the temporal registration issue. They are beyond the scope of this paper.

## V. CONCLUSION

In this paper, we presented a novel adaptive depth sampling algorithm based on DL. The mask generation network *NetM* is trained along with the depth completion network *NetE* to predict the optimal sampling locations based on an input RGB image. Experiments demonstrate the effectiveness of the proposed *NetM*. Higher depth estimation accuracy is achieved by *NetM* under various depth completion algorithms. We also show that best end to end performance is achieved by *NetM* with a state-of-the-art depth completion algorithm.

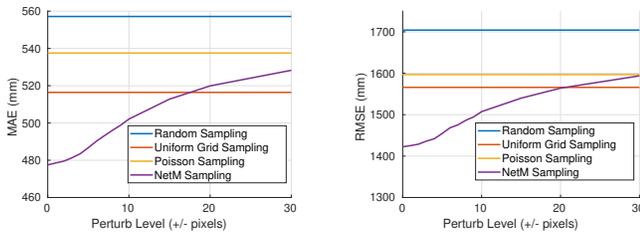


Fig. 12. MAE/RMSE with respect to the increasing sampling location perturbation. Random, Uniform Grid and Poisson sampling masks do not need RGB image as input, thus they are free from the location perturbation.

Such adaptive depth sampling strategy enables more efficient depth sensing and overcomes the trade-off between frame-rate, resolution, and range in an active depth sensing system (such as LiDAR and sparse dot pattern structured light sensor).

## REFERENCES

- [1] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 163–168.
- [2] HoloLens, "(microsoft) 2020." Retrieved from <https://www.microsoft.com/en-us/hololens>.
- [3] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE transactions on cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.
- [4] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, "A stereo machine for video-rate dense depth mapping and its new applications," in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1996, pp. 196–202.
- [5] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (tof) cameras: a survey," *IEEE Sens. J.*, vol. 11, no. 9, 2011.
- [6] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 2019, pp. 1–6.
- [7] D. B. Lindell, M. O'Toole, and G. Wetzstein, "Single-photon 3d imaging with deep sensor fusion," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–12, 2018.
- [8] A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Deep adaptive lidar: End-to-end optimization of sampling and depth completion at low sampling rates," in *2020 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2020, pp. 1–11.
- [9] F. Pittaluga, Z. Tasneem, J. Folden, B. Tilmon, A. Chakrabarti, and S. J. Koppal, "Towards a mems-based adaptive lidar," in *2020 International Conference on 3D Vision (3DV)*. IEEE, 2020, pp. 1216–1226.
- [10] Z. Tasneem, C. Adhivarahan, D. Wang, H. Xie, K. Dantu, and S. J. Koppal, "Adaptive fovea for scanning depth sensors," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 837–855, 2020.
- [11] T. Yamamoto, Y. Kawanishi, I. Ide, H. Murase, F. Shimura, and D. Deguchi, "Efficient pedestrian scanning by active scan lidar," in *2018 International Workshop on Advanced Image Technology (IWAIT)*. IEEE, 2018, pp. 1–4.
- [12] K. Karsch, C. Liu, and S. B. Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 11, pp. 2144–2158, 2014.
- [13] —, "Depth transfer: Depth extraction from videos using nonparametric sampling," in *Dense Image Correspondences for Computer Vision*. Springer, 2016, pp. 173–205.
- [14] A. Saxena, S. Chung, and A. Ng, "Learning depth from single monocular images," *Advances in neural information processing systems*, vol. 18, pp. 1161–1168, 2005.
- [15] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.
- [16] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2650–2658.
- [17] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth international conference on 3D vision (3DV)*. IEEE, 2016, pp. 239–248.
- [18] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2002–2011.
- [19] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "Geonet: Geometric neural network for joint depth and surface normal estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 283–291.
- [20] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *arXiv preprint arXiv:1812.11941*, 2018.
- [21] J. H. Lee, M.-K. Han, D. W. Ko, and I. H. Suh, "From big to small: Multi-scale local planar guidance for monocular depth estimation," *arXiv preprint arXiv:1907.10326*, 2019.
- [22] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6101–6108.
- [23] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgb-d images," in *European conference on computer vision*. Springer, 2012, pp. 746–760.
- [24] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [25] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3234–3243.
- [26] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the cpu," in *2018 15th Conference on Computer and Robot Vision (CRV)*. IEEE, 2018, pp. 16–22.
- [27] S. Hawe, M. Kleinsteuber, and K. Diepold, "Dense disparity maps from sparse disparity measurements," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 2126–2133.
- [28] L.-K. Liu, S. H. Chan, and T. Q. Nguyen, "Depth reconstruction from sparse samples: Representation, algorithm, and sampling," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1983–1996, 2015.
- [29] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.
- [30] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *2017 international conference on 3D Vision (3DV)*. IEEE, 2017, pp. 11–20.
- [31] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 689–694.
- [32] J. T. Barron and B. Poole, "The fast bilateral solver," in *European Conference on Computer Vision*. Springer, 2016, pp. 617–632.
- [33] J. Lu and D. Forsyth, "Sparse depth super resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2245–2253.
- [34] G. Drozdov, Y. Shapiro, and G. Gilboa, "Robust recovery of heavily degraded depth measurements," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 56–65.
- [35] F. Ma, L. Carlone, U. Ayaz, and S. Karaman, "Sparse depth sensing for resource-constrained robots," *The International Journal of Robotics Research*, vol. 38, no. 8, pp. 935–980, 2019.
- [36] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3288–3295.
- [37] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich, "Estimating depth from rgb and sparse sensing," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 167–182.
- [38] M. Jaritz, R. De Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with cnns: Depth completion and semantic segmentation," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 52–60.
- [39] A. Wong, X. Fei, S. Tsuei, and S. Soatto, "Unsupervised depth completion from visual inertial odometry," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1899–1906, 2020.

- [40] Y. Yang, A. Wong, and S. Soatto, "Dense depth posterior (ddp) from single image and sparse range," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3353–3362.
- [41] S. S. Shivakumar, T. Nguyen, I. D. Miller, S. W. Chen, V. Kumar, and C. J. Taylor, "Dfusenet: Deep fusion of rgb and sparse depth information for image guided dense depth completion," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 13–20.
- [42] R. L. Cook, "Stochastic sampling in computer graphics," *ACM Transactions on Graphics (TOG)*, vol. 5, no. 1, pp. 51–72, 1986.
- [43] R. Piroddi and M. Petro, "Analysis of irregularly sampled data: A review," *Advances in Imaging and Electron Physics*, vol. 132, pp. 109–167, 2004.
- [44] R. Bridson, "Fast poisson disk sampling in arbitrary dimensions." *SIGGRAPH sketches*, vol. 10, p. 1, 2007.
- [45] S. Wu, A. Dimakis, S. Sanghavi, F. Yu, D. Holtmann-Rice, D. Storchus, A. Rostamizadeh, and S. Kumar, "Learning a compressed sensing measurement matrix via gradient unrolling," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6828–6839.
- [46] F. Li, M. Zhao, Z. Tian, F. Willomitzer, and O. Cossairt, "Compressive ghost imaging through scattering media with deep learning," *Optics Express*, vol. 28, no. 12, pp. 17 395–17 408, 2020.
- [47] J. Wang, Q. Gao, X. Ma, Y. Zhao, and Y. Fang, "Learning to sense: Deep learning for wireless sensing with less training efforts," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 156–162, 2020.
- [48] M. R. Kellman, E. Bostan, N. A. Repina, and L. Waller, "Physics-based learned design: optimized coded-illumination for quantitative phase imaging," *IEEE Transactions on Computational Imaging*, vol. 5, no. 3, pp. 344–353, 2019.
- [49] M. Kellman, E. Bostan, M. Chen, and L. Waller, "Data-driven design for fourier ptychographic microscopy," in *2019 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2019, pp. 1–8.
- [50] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.
- [51] L. Demaret, N. Dyn, and A. Iske, "Image compression by linear splines over adaptive triangulations," *Signal Processing*, vol. 86, no. 7, pp. 1604–1616, 2006.
- [52] S. Rajesh, K. Sandeep, and R. Mittal, "A fast progressive image sampling using lifting scheme and non-uniform b-splines," in *2007 IEEE International Symposium on Industrial Electronics*. IEEE, 2007, pp. 1645–1650.
- [53] G. Ramponi and S. Carrato, "An adaptive irregular sampling algorithm and its application to image coding," *Image and Vision Computing*, vol. 19, no. 7, pp. 451–460, 2001.
- [54] A. S. Lin, B. Z. Luo, C. J. Zhang, and D. E. Saucan, "Generalized ricci curvature based sampling and reconstruction of images," in *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 604–608.
- [55] J. Liu, C. Bouganis, and P. Y. Cheung, "Kernel-based adaptive image sampling," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 1. IEEE, 2014, pp. 25–32.
- [56] A. Taimori and F. Marvasti, "Adaptive sparse image sampling and recovery," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 311–325, 2018.
- [57] Q. Dai, H. Chopp, E. Pouyet, O. Cossairt, M. Walton, and A. Katsaggelos, "Adaptive image sampling using deep learning and its application on x-ray fluorescence image reconstruction," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2564–2578, 2020.
- [58] A. Kuznetsov, N. K. Kalantari, and R. Ramamoorthi, "Deep adaptive sampling for low sample count rendering," in *Computer Graphics Forum*, vol. 37, no. 4. Wiley Online Library, 2018, pp. 35–44.
- [59] I. A. Huijben, B. S. Veeling, and R. J. van Sloun, "Deep probabilistic subsampling for task-adaptive compressed sensing," in *International Conference on Learning Representations*, 2019.
- [60] —, "Learning sampling and model-based signal recovery for compressed sensing mri," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8906–8910.
- [61] E. J. Gumbel, "Statistical theory of extreme values and some practical applications," *NBS Applied Mathematics Series*, vol. 33, 1954.
- [62] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.
- [63] A. Wolff, S. Praisler, I. Tcenov, and G. Gilboa, "Super-pixel sampler: a data-driven approach for depth sampling and reconstruction," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2588–2594.
- [64] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [65] V. Jampani, D. Sun, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Superpixel sampling networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 352–368.
- [66] F. Yang, Q. Sun, H. Jin, and Z. Zhou, "Superpixel segmentation with fully convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 964–13 973.
- [67] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.



**Qiqin Dai** received the B.S. degree in automation from Zhejiang University, China, in 2012. He completed his M.S. and Ph.D. degree with the Image and Video Processing Laboratory, Northwestern University, Evanston, IL, USA, in 2017. Now he is with Geomagic Labs, Mountain View, CA, USA, where he is currently working on indoor perception using machine learning. His research interests include machine learning techniques for digital image processing, computer vision and computational photography.



**Fengqiang Li** Fengqiang Li is currently a machine learning and computer vision algorithm engineer with Apple Inc. Previously, he received his Ph.D. degree in computer science from Northwestern University. He was with Prof. Oliver Cossairt in Computational Photography Lab at Northwestern University, where he worked on computational photography and computer vision. Before that, he obtained his BS degree in optoelectronic information engineering from Huazhong University of Science and Technology and his MS degree in electrical engineering from

Lehigh University.



**Oliver Cossairt** Oliver Cossairt is Associate Professor in the Computer Science (CS) and Electrical and Computer Engineering (ECE) departments at Northwestern University. Prof. Cossairt is director of the Computational Photography Laboratory (CPL) at Northwestern University ([compphoto.northwestern.edu](http://compphoto.northwestern.edu)), whose research consists of a diverse portfolio, ranging in topics from optics/photonics, computer graphics, computer vision, machine learning and image processing. The general goal of CPL is to develop imaging hardware and algorithms that can be applied across a broad range of physical scales, from nanometer to astronomical. This includes active projects on 3D nanotomography (10-9 m), computational microscopy (10-6 m), cultural heritage imaging analysis of paintings (10-3 m), structured light and ToF 3D-scanning of macroscopic scenes (1 m), de-scattering through fog for remote sensing (103 m), and coded aperture imaging for astronomy (106 m). Prof. Cossairt has garnered funding from numerous corporate sponsorships (Google, Ram-bus, Samsung, Omron, Oculus/Facebook, ZoloZ/Alibaba) and federal funding agencies (ONR, NIH, DOE, DARPA, IARPA, NSF CAREER Award).



**Aggelos K. Katsaggelos** received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees in Electrical Engineering from the Georgia Institute of Technology, in 1981 and 1985, respectively. In 1985, he joined the Department of Electrical Engineering and Computer Science at Northwestern University, where he is currently a Professor holder of the Joseph Cummings chair. He was previously the holder of the Ameritech Chair of Information Tech-

nology and the AT&T chair. He is also a member of the Academic Staff, NorthShore University Health System, an affiliated faculty at the Department of Linguistics and he has an appointment with the Argonne National Laboratory. He has published extensively in the areas of multimedia signal processing and communications, computational imaging, and machine learning (over 250 journal papers, 600 conference papers and 40 book chapters) and he is the holder of 30 international patents. He is the co-author of *Rate-Distortion Based Video Compression* (Kluwer, 1997), *Super-Resolution for Images and Video* (Claypool, 2007), *Joint Source-Channel Video Transmission* (Claypool, 2007), and *Machine Learning Refined* (Cambridge University Press, 2016). He has supervised 57 Ph.D. theses so far. Among his many professional activities Prof. Katsaggelos was Editor-in-Chief of the *IEEE Signal Processing Magazine* (1997–2002), a BOG Member of the *IEEE Signal Processing Society* (1999–2001), a member of the Publication Board of the *IEEE Proceedings* (2003–2007), and a Member of the Award Board of the *IEEE Signal Processing Society*. He is a Fellow of the *IEEE* (1998), *SPIE* (2009), *EURASIP* (2017), and *OSA* (2018). He is the recipient of the *IEEE Third Millennium Medal* (2000), the *IEEE Signal Processing Society Meritorious Service Award* (2001), the *IEEE Signal Processing Society Technical Achievement Award* (2010), an *IEEE Signal Processing Society Best Paper Award* (2001), an *IEEE ICME Paper Award* (2006), an *IEEE ICIP Paper Award* (2007), an *ISPA Paper Award* (2009), and a *EUSIPCO paper award* (2013). He was a Distinguished Lecturer of the *IEEE Signal Processing Society* (2007–2008).