PhoneDepth: A Dataset for Monocular Depth Estimation on Mobile Devices

Fausto Tapia Benavides
ETH Zürich
fausto.tapiabenavides@gmail.com

Andrey Ignatov
ETH Zürich
andrey@vision.ee.ethz.ch

Radu Timofte
JMU Würzburg, ETH Zürich
radu.timofte@uni-wuerzburg.de

Abstract

Monocular depth estimation has been studied as a classic and learning based computer vision problem for decades. However, little attention received the efficiency and the deployment of methods on mobile hardware. All publicly available datasets have severe limitations related to their applicability to camera data captured with real mobile devices. For instance, the main issues with current datasets include (but not exhaustively) low quality of images due the cameras or collection methods, domain specifically generated datasets as is the case for autonomous driving, small number of samples, sparse depthmaps, etc. In response, we introduce PhoneDepth, a novel dataset that aims to take advantage of modern phones hardware and professional stereo cameras. Depthmaps are acquired from three sources: Time of Flight sensor, Dual Pixel sensor and stereo camera; while the images correspond to mobile phone photos. We prove its high value by training neural networks with multiple depth supervision, fine-tuning on other datasets and for depth refinement. Along with the dataset we present benchmark models and a toolbox to facilitate the dataset usage.

1. Introduction

The understanding of an environment in the 3D world has been studied for a number of years in the computer vision community due to its wide range of applications including robotics, mapping, measuring, *etc*. For instance, early methods focused merely in the optical characteristics of cameras, including techniques such as SLAM, structure from motion (SfM), visual odometry [12, 23, 26]. These generally acquire sparse data, are computationally demanding and require special equipment to compute real scale data. Furthermore, they depend on many parameters that need to be set manually according to the application or scenario in mind.

Furthermore, there exist some hardware based approaches to solve this problem such as RGB-D cameras that provide dense depthmaps but have low resolution and small range of measurement as stated by [19, 20, 33]. LIDAR is another hardware technology able to acquire highly accurate depthmaps although quite sparse and with a high sensor cost [7, 18].

It is natural then that camera based approaches are desirable to improve performance, resolution and costs of depth estimation solutions. In this context, multiple deep-learning based methods have been developed for monocular depth estimation where only a single image or stream for a single camera is used to render corresponding depthmaps. The accuracy of deep learning algorithms tends to outperform more classical methods, however they also tend to need heavy computing hardware for example in the cases of [8, 13, 31, 34].

Data is the center of deep-learning algorithms, and as such a crucial part for advances in visual depth estimation. However, it is significantly difficult to acquire due to special equipment needs, calibration, non-trivial post-processing, etc. Hence, available datasets tend to be specialized into some niche which increases biases to this geometric problems. For instance, KITTI [29], Cityscapes [6] and Synthia [24] show predominantly self-driving datasets, Megadepth [17] includes mostly touristic places, NYU [21] focuses on domestic indoor scenarios, being MAI [14] and Megadepth [17] the broader in context.

Additionally, these datasets are acquired with out-dated technologies, neglecting newer cameras, resolutions and hardware now available in mobile devices such as factory-calibrated cameras, in-built time of flight sensors (ToF) and modern camera sensors. Therefore, we find the need to acquire a new dataset that leverages these technologies to improve depth acquisition in mobile devices.

In this paper, we propose a novel dataset with multiple depth sources meant to especially aid deep learning problems for light-weight networks to be used with mobile and hardware restricted devices.

2. Related work

Even though there are multiple depth-from-image learning based techniques, the datasets used have been relatively stagnant. Therefore, there has been a somewhat tacitly ac-

cepted over-fitting to these contexts. We here describe a selection of the most important datasets in the domain, their drawbacks and advantages. Table 1 summarizes their main characteristics.

MAI. The Mobile AI depth estimation challenge dataset [14] provides with image to depthmap correspondences. It comprises of 8K samples that were acquired in outdoors environments using a ZED-camera for both color image and depthmap collection. They all have VGA resolution (640x480), which is quite low compared to cameras in phones. RGB images are collected as regular 8-bit photos while depthmaps are represented as 16bit images, giving enough bit-depth to represent the ZED-camera sensor accuracy of 0.2m at distances below 8m.

Megadepth. The Megadepth dataset [17] was introduced as one of the largest datasets for the monocular depth estimation problem with a total of 130K images among which 100K are depthmaps and 30K ordinal. Ordinal images simply indicate relative depth order between two contextual objects in the same image. They collect images from an internet dataset of photos taken from different locations and follow 3D estimation with classical methods such as Multi-view-stereo (MVS) and Structure from Motion (SfM). Outliers are filtered in multiple stages based on applying more conservative constraints to the MVS problem and using semantic filtering. They use a categorization of foreground, background and sky objects to automatically eliminate semantically inconsistent pixels. The depthmaps are partially dense in the sense that they are filtered in a way that large continuously valid pixels are present in the image. However, most images have large portions of invalid pixels that are masked out. Furthermore, the quality of the color images is dubious since they are all photos collected from the internet. Hence, motion blur, noise, and lack of detail in the photos are common issues in this dataset that would propagate inaccuracies through the 3D reconstruction process.

NYU. As the first dataset that collected dense-depth samples for images, the NYU dataset [21] has been a popular option. Their main aim was to acquire data to understand surfaces and their interactions in 3D environments. However, part of the dataset corresponds to images with corresponding depthmaps. It was collected from video recordings of Microsoft's Kinect devices including an RGB source and a structured-light based depth source. As the videos are not static, the number of images is severely pruned to a total number of 1449 image-depth correspondences (synchronized). The resolution of these images is VGA (640×480) which alike MAI, is quite low if we compare them to current phones' camera resolutions. Furthermore, due to the low number of samples this dataset provides, deep learning approaches often over-fit as shown in [1]. Additionally, this dataset is only representative for indoor environments, which are generally easier to learn than outdoors due to a lower variance in the image distribution.

Make3D. This dataset presented in [2, 25] was the pioneer for single-image depth estimation. It contains 534 images separated as 400 training as 134 testing. Images have a constant size of 2272×1704 which is representative of modern phone cameras. However, the depth maps resolution is very low for detailed structures (305×55) as they were collected with custom-made 3D laser scanner. Furthermore, images are collected in a variety of similar outdoors scenes, given some degree of variance in the dataset distribution. Unfortunately, the size of this dataset is quite reduced and even though it was used for training in early methods, it was later adopted mostly for generalization testing in works such as [10, 17].

Cityscapes. With another learning task, the Cityscapes dataset proposed in [6] mainly for semantic segmentation contains a range of images collected in 50 different cities. Each instance of this dataset comprises a semantically labelled stereo pair. Therefore, for depth prediction purposes, the semantic labels are usually disregarded and the stereo pair is used to retrieve depth information. Each image has a resolution of 1024×2048 adding up to a total number of 20K frames. Mostly unsupervised and self-supervised methods use this dataset including works in [3, 5, 9, 10, 31]. From these, all except from the second show that pretraining on Cityscapes results in better results when finetuning especially on KITTI. However, it has been shown that in semi-supervised methods (where some ground-truth is available for training in the form of sparse depth usually) accuracy is improved dramatically due to the inevitable uncertainties of stereo-matching especially in occluded regions. This is evidenced for example in studies such as [7, 16, 32] where they experience significant performance improvements.

KITTI. As the most well known self-driving dataset, KITTI [29] presents with images and their corresponding dense depthmap. The depthmaps are acquired by projecting points scanned via a LIDAR sensor into calibrated camera frames. Even though the LIDAR sensor used is of high resolution, the depthmaps acquired are still sparse and missing some details that are usually dealt with by leveraging interpolation or sparse supervision. The dataset provides a total of 93k samples corresponding to 56 scenes and with a resolution of 1224×368. It is also used to benchmark different geometric tasks for autonomous driving including visual odometry and stereo learning as it also has some sequences with ground truth pose.

DDAD. Another self-driving related dataset is the so called Dense Depth for Autonomous Driving (DDAD) presented in [11]. This dataset was collected with a similar set-up to KITTI but on a fleet of cars over multiple cities in the United States and Japan. The samples con-

tain monocular images covering the 360 degrees around the car as well as high density LIDAR scan covering the same range. The depthmaps projected to each image are not dense, in fact they are sparser than KITTI. However, the images are of higher quality and resolution with image sizes of 1936×1216. Additionally, the different directions of the cameras provide different street perspectives that KITTI lacks. In total, it provides with 16600 samples each containing 6 images and a 360 degree synchronized LIDAR scan that can be projected to each image frame.

nuScenes. The nuScenes dataset [4] compiles a range of streams also related to autonomous driving. It is the public dataset that collects the most number of data-streams in its domain. It includes 360 degree coverage around the vehicle with cameras, a LIDAR sensor and radars. It additionally includes 3D location of 23 semantic classes present around each scene. The dataset collects 93k fully annotated images from which they also form 13 sample sequences. The images have a resolution of 1600×900 and the LIDAR scans can be projected to the images although the depthmaps resulting are quite sparse. since the laser scanner only has 32 beams meaning very limited vertical resolution.

In general, all datasets that serve single-image depth estimation have caveats of different characteristics given the collection difficulty and sometimes hardware restrictions. Common issues include too small datasets, low image resolution, and highly sparse depthmaps. Furthermore, no other dataset collects depth from multiple sensor sources neither with the aim of aiding depth estimation in mobile phones and real-time.

3. PhoneDepth Dataset

In this section, we present a novel PhoneDepth dataset and describe its contents, data collection and processing setups.

3.1. Dataset Description

The dataset is composed of 6035 image sets containing 1202 outdoor and 4833 indoors scenes. The data was collected using two modern smartphones and a professional ZED stereo camera [22] that demonstrates an average depth estimation error of less than 0.2m for objects located closer than 8 meters. All the dataset was acquired in the city of Zürich. Indoors samples were captured inside university and residential buildings adding up to a total of 17 buildings. Each building includes on average 20 sites. Outdoor images correspond to sidewalk views, houses, facades, parks, gardens, etc. of 8 neighborhoods from which 3 were highly urban and 5 suburban. The phones used were a Huawei P40 Pro with a dedicated Time-of-Flight (ToF) depth sensor, and a Google Pixel 2 device estimating depth maps using the dual pixel technology. In the next sections, these phones are named as Phone 1 and Phone 2, respectively. All devices were mounted on a rig to fix their relative position and get an accurate projection of the captured stereo depth maps to phone frames.

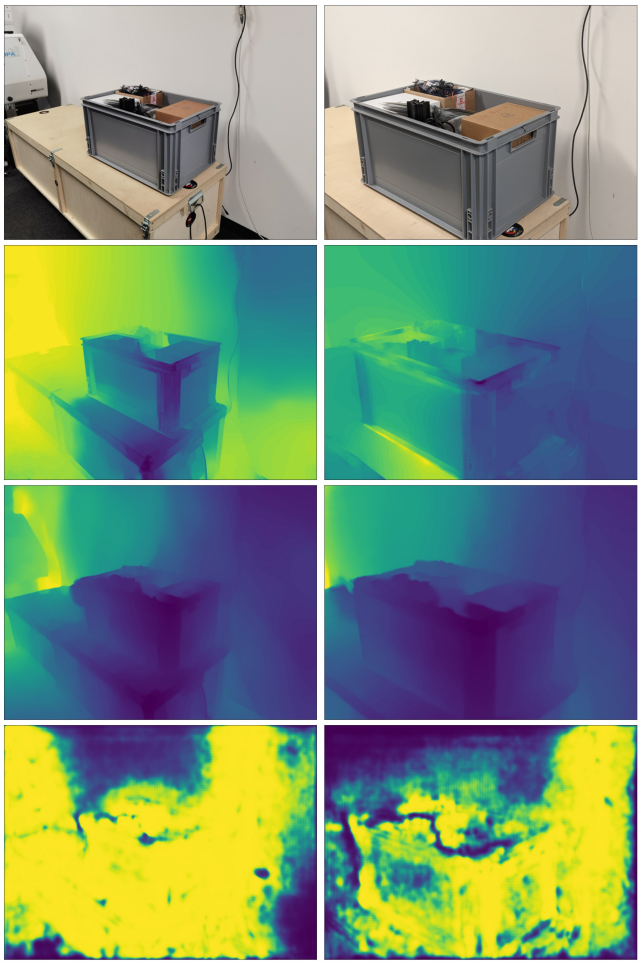
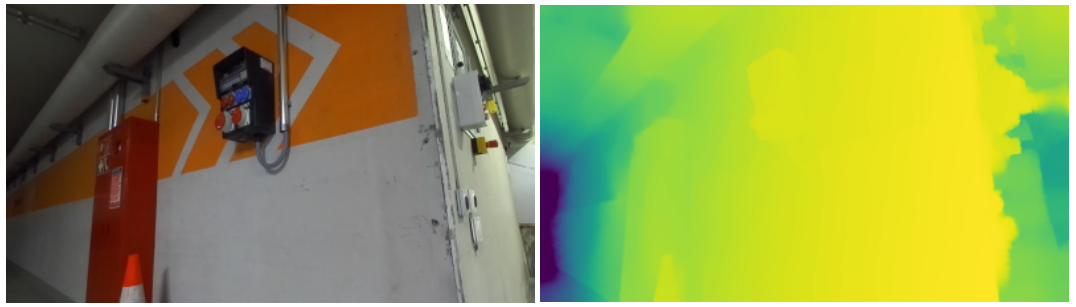
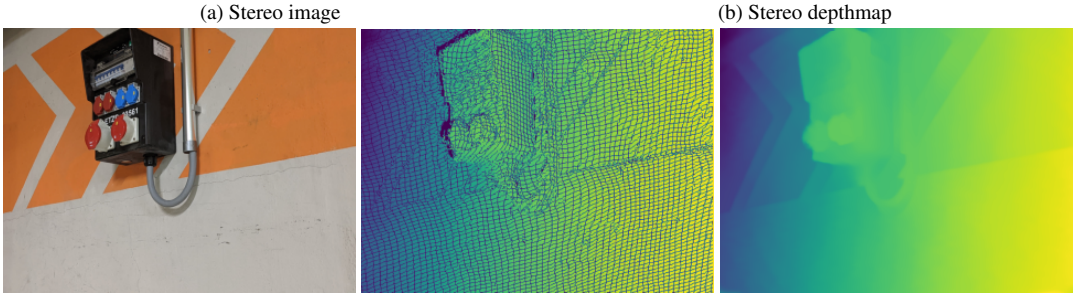


Figure 1. Sample data from the PhoneDepth dataset (the left column corresponds to Phone 1, right - to Phone 2). From top to bottom: the original RGB phone image, phone depth map, projected depth from the stereo camera, and corresponding confidence map.

Figure 2. The rig setup used for the dataset collection. Phone 2 is on the left, Phone 1 is on the right, and the Zed camera is in the middle. A portable laptop is also shown on this photo.



(a) Stereo image (b) Stereo depthmap



(c) Phone image (d) Calibration based projection (e) Dense matching

Figure 3. Depiction of depth transfer from stereo frames to phone frames

Each collected image set from the PhoneDepth dataset is composed of the following data:

1. Two original full-resolution RGB photos captured using the standard mobile cameras installed in the considered smartphones;

2. Two coarse depth maps acquired with mobile sensors (ToF and dual-pixel) installed in the Huawei P40 Pro and the Google Pixel 2 phones;

3. RGB image, an accurate HD-resolution 16-bit depth map and the associated confidence map collected with the ZED stereo camera.

Figure 1 demonstrates the collected data taken from a sample set. Note that the depth maps obtained with phones hardware tend to be inaccurate due to the small space they need to be fit into. Additionally, phone sensors have a limited bit-depth of 8-bits, thus quantization artifacts are usually an issue (see Phone 2 depth), though they might show a better edge sharpness than the stereo depth maps.

3.2. Data collection protocol

To ensure a high quality of the dataset, we performed the data collection following several important steps. First, a special dataset collection software was developed to acquire phone and stereo camera data synchronously while requiring no manual manipulations with the devices. Next, we explored numerous indoor and outdoor scenes during 21 days, within which the tripod height was varied to get perspective variations. Finally, during the dataset collection period, the rig was standing still and only scenes with no motion were captured to avoid invalid depth predictions and projections due to objects movement. Hence, strict sensor synchronization was not needed.

3.3. Image/Depth sources

This subsection provides the resolutions and some other characteristics of the captured data. Phone 1 collects RGB photos of resolution of 4096×3072px, while its ToF sensor captures depthmaps of size 1027×768 with a field of view (FOV) of 78 degrees. Phone 2 retrieves photos and depthmaps of resolution 2686×2016, and its dual pixel sensor has a FOV of 55 degrees. Lastly, the ZED camera and its custom software allows to acquire images and depthmaps of resolution 1280×720px with a 90 degree FOV. We analyzed the features from its SDK used to obtain dense depthmaps and disabled the texture confidence filtering option while allowed for 90% stereo matching confidence. Furthermore, we set the measuring depth to lie between 0.2 and 10m to avoid excessive depth uncertainty.

3.4. Depth transfer

The last step to get our dataset was to project the stereo based depthmaps to the corresponding phone frames. To do so, we used the state-of-the-art deep learning based dense matching technique called the PDCNet [28]. This method estimates a dense flow field relating two images and allows to transfer depthmaps between images having a different geometry / field of view by using a homography based model

Dataset	Samples	Image resolution	Depth resolution	Image source	Depth source	Confidence map	True depth	depthmap density	Context
MegaDepth	121K (19K ordinal)	1400x1000 (avg)	1400x1000 (avg)	Internet images (mostly tourist style)	MVS	No	No	Medium	Landmark tourist attractions
MAI	7385	640x480	640x480	ZED Camera	ZED Camera (Stereo)	No	Yes	High	City Outdoors (Zurich)
NYU	1449	640x480	640x480	Microsoft Kinect	Microsoft Kinect	No	True	High	Indoors rooms
Make3D	534	2272x1704	305x55	Custom 3D scanner	Custom 3D scanner	No	Yes	High	Outdoors (campus, street)
Cityscapes	20K	2048x1024	Inexistent (stereo pairs)	HDR stereo videos	Inexistent (from stereo)	No	No	Not provided	Cities roads
KITTI	93K	1224x368	1224x368	Stereo Pair	LIDAR	No	Yes	Medium	Cities road (self-driving)
DDAD	16.6K	1936x1216	30k points per frame	6 cameras 360 degree	LIDAR (Luminar H2)	No	Yes	Medium	Cities roads (self-driving)
nuScenes	93K	1600x900	32 beam LIDAR	6 cameras 360 degree	LIDAR	No	Yes	Low	Cities roads (self-driving)
PhoneDepth (ours)	6035	4096x3072 and 2686x2016	1027x768, 2686x2016 and 960x720	Huawei P40 Pro, Pixel 2	ToF, Dual pixel, Professional stereo	Yes	Yes	High	Outdoors + indoors

Table 1. Comparison of depth estimation datasets.

after the dense matching is performed. We use PDCNet to compute flow maps from the stereo images to each of the phone images. Then, depthmaps from the stereo pair are projected to the phone frames using these flow maps, allowing us to generate fine-grained depthmaps for each phone image. Furthermore, PDCNet also provides confidence maps for each correspondence, which we also transfer to the phone frames thus providing depth confidence to our projected images. It is also worth mentioning that this matching was performed on 960×720 images, which is the final resolution of our depthmaps.

We show the benefits of using this depth transfer methodology by comparing the above results to the depthmaps generated by performing geometrically-based cameras calibration and then projecting the depthmaps represented as point-clouds to the image frames. Figure 3 shows how the dense matching method is able to transfer smooth and accurate depthmaps against the calibration based projection.

3.5. Remarks

Finally, we put the PhoneDepth dataset in comparison with the current datasets used for the depth estimation problem and highlight their characteristics in Table 1. We would like to emphasize that the PhoneDepth is the only dataset containing data from mobile depth sensors, and the only one with more than one source of depth / confidence maps. The latter could be used to improve convergence via giving less importance to low confidence (inaccurate) regions.

4. Experiments

In this section we describe the models, training setup and experiments and discuss the achieved results validating the proposed PhoneDepth dataset.

4.1. Models and training setup

Models. Four different models were trained in different ways to show the value of our dataset. The first one is FastDepth presented in [30]. The second model is Park model, winner of the MAI 2021 challenge [14]. The third model uses a U-net topology with EfficientNetB4 [27] as encoder and the decoder part of Park (EffnetB4-park). Lastly, to take advantage of the depthmaps that these phones have available on run-time, we present the task of depth refinement. As so, the fourth model is a combination of EffnetB4-park with a light-weight multi-level encoder that extracts features of the depthmap to then be concatenated with the EfficientNet outputs and fed into the decoder. The depth encoder is composed of $DepthSepCov \rightarrow Upsize \rightarrow BN \rightarrow Concat \rightarrow DepthSepConv \rightarrow BN$ blocks, where up-sizing is performed bi-linearly (further details on this model in the supplementary materials). These four models run with latencies of 10, 5, 99 and 253 ms respectively as measured by the AI Benchmark app [15] on a Huawei P40 Pro phone GPU after default Tensorflow-lite optimization.

Loss. The loss utilized for training is the one presented by MegaDepth and is composed of three parts: data loss, gradient loss and ordinal loss. Losses and error metrics are computed on the original resolution of the dataset images by performing bi-linear resizing to the models' output. Ordinal loss is ignored for datasets without ordinal samples. For cases where two separate depthmaps are used for supervision, data loss is applied to the stereo depthmap and gradient loss is applied to the phone depthmap.

Data. We use samples from Phone 1 in our dataset to perform our experiments. We expect similar results when using Phone 2.

Hyper-parameters. We use Adam optimizer with 0.00005 learning rate. We train up to 65 epochs on Megadepth and 120 on the others. Then, we choose the epoch with the best validation scale invariant RMSE (si-RMSE).

4.2. Double depth supervision

Our first set of experiments shows how using both streams of depth from our PhoneDepth dataset as supervision (phone and stereo-camera projection), improves performance. Hence, we train networks with regular Megadepth loss (excluding ordinal) as baseline and compare them with models trained with data loss on the projected depth (as more precise) and gradient loss on the phone depth. Table 2 depicts that training with this modality improves performance of models. Note that there is special benefit for very small models (Fastdepth and Park).

Model	si-RMSE	RMSE	Avg rel
Fastdepth	0.2433	0.07755	0.2362
〃 〃 double depth	**0.2371**	**0.07545**	**0.2283**
Park	0.2468	0.08095	0.2363
〃 〃 double depth	**0.2283**	**0.07317**	**0.2196**
EffnetB4-park	0.2071	0.06890	**0.1909**
〃 〃 double depth	**0.2005**	**0.06512**	0.1927

Table 2. Depth estimation results on PhoneDepth.

4.3. Pretraining and transfer

To validate to which extent the performance on other benchmarks can benefit from our dataset, we use the double depth trained versions of the models in the previous section and fine-tune them to the Megadepth (MD) and MAI datasets and compare with the models trained from scratch on MD and MAI, respectively. Table 3 shows the results and we can overwhelmingly see that pretraining on our dataset improves the results on MD and MAI datasets for most cases. An exception is for Park model which is small and does not benefit from the pretraining on MD, a dataset with significant differences in content distribution and depthmaps when compared with MAI and PhoneDepth.

4.4. Depth refinement

Our last experiment corresponds to defining a different stream of training with this dataset, namely depth-refinement. This has not been analyzed at all in the realm of mobile devices, so our dataset provides great value in this sense. The aim is to make use of the raw depth acquired by phone sensors as an input as well as the raw color image. Hence, our dataset provides with two ways of training for depth refinement: Using stereo-projected depth only (ID2P) for supervision or using both stereo and phone depth for supervision (ID2DP). Table 4 shows the results after training (with loss as described in section 4.1) compared to the benchmark (I2P) trained only with stereo depth for supervision. Note that there are performance improvements of more than 11% compared to the benchmark. Consequently, it is a potentially important task to investigate in further research.

Model	Dataset	si-RMSE	RMSE	Avg rel
Fastdepth	MAI	0.2562	3.062	0.1910
〃 〃 fine-tuned	MAI	**0.2552**	**3.061**	**0.1884**
Park	MAI	0.2802	3.244	0.2131
〃 〃 fine-tuned	MAI	**0.2570**	**3.046**	**0.1909**
EffnetB4-park	MAI	0.2046	2.926	0.1635
〃 〃 fine-tuned	MAI	**0.2005**	**2.554**	**0.1459**
Fastdepth	MD	0.0780	2.425	0.05458
〃 〃 fine-tuned	MD	**0.0757**	**2.312**	**0.05256**
Park	MD	**0.0773**	**2.419**	**0.05374**
〃 〃 fine-tuned	MD	0.0793	2.574	0.05550
EffnetB4-park	MD	0.0525	1.749	0.03477
〃 〃 fine-tuned	MD	**0.0501**	**1.676**	**0.03277**

Table 3. Test depth estimation results on MAI and MD for the models trained from scratch or with pretraining on PhoneDepth and finetuning on MAI and MD train datasets, resp.

Model	Train method	si-RMSE	RMSE	Avg rel
EffnetB4-park	I2P	0.2071	0.06890	0.1910
EffnetB4-DepthEnc-park	ID2P	0.1836	0.06083	0.1707
EffnetB4-DepthEnc-park	ID2DP	0.1837	0.06034	0.1735

Table 4. Depth refinement results on PhoneDepth.

5. Conclusion

In this paper, we present the first depth estimation and refinement dataset specific for mobile devices. It collects depthmaps from two different phones and a professional stereo camera (hardware and software) considered as truth source. Compared to other publicly available datasets in the domain, ours is the first one to also include confidence maps. It also has superb depthmap density and multiple depth sources while collecting indoors and outdoors samples, characteristics missing in other datasets. Lastly, we introduce the depth-refinement task, which facilitated by our dataset has not been explored in mobile devices, especially within the machine learning domain. We expect that our dataset aids further developments in geometric and high accuracy 3D measurement tasks on mobile devices.

Acknowledgments

This work was supported by ETH Zürich General Fund and by Humboldt Foundation.

References

[1] Filippo Aleotti, Giulio Zaccaroni, Luca Bartolomei, Matteo Poggi, Fabio Tosi, and Stefano Mattoccia. Real-time single image depth perception in the wild with handheld devices. *CoRR*, abs/2006.0, 2020. 2

[2] Saxena Ashutosh, Sun Min, and Ng Andrew Y. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI)*, 30(5):824–840, 2009. 2

[3] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 3

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 2

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2

[7] Nícolas dos Santos Rosa, Vitor Campanholo Guizilini, and Valdir Grassi. Sparse-to-Continuous: Enhancing Monocular Depth Estimation using Occupancy Maps. *2019 19th International Conference on Advanced Robotics (ICAR)*, pages 793–800, 2019. 1, 2

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. *CoRR*, abs/1406.2, 2014. 1

[9] Ravi Garg, Vijay Kumar B. G, and Ian D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *CoRR*, abs/1603.04992, 2016. 2

[10] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CoRR*, abs/1609.03677, 2016. 2

[11] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[12] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, USA, 2 edition, 2003. 1

[13] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding Monocular Depth Estimation Using Depth-Attention Volume. *CoRR*, abs/2004.0, 2020. 1

[14] Andrey Ignatov, Grigory Malivenko, David Plowman, Samarth Shukla, Radu Timofte, Ziyu Zhang, Yicheng Wang, Zilong Huang, Guozhong Luo, Gang Yu, Bin Fu, Yiran Wang, Xingyi Li, Min Shi, Ke Xian, Zhiguo Cao, Jin-Hua Du, Pei-Lin Wu, Chao Ge, Jiaoyang Yao, Fangwen Tu, Bo Li, Jung Eun Yoo, Kwanggyoon Seo, Jialei Xu, Zhenyu Li, Xianming Liu, Junjun Jiang, Wei-Chi Chen, Shayan Joya, Huanhuan Fan, Zhaobing Kang, Ang Li, Tianpeng Feng, Yang Liu, Chuannan Sheng, Jian Yin, and Fausto T Benavide. Fast and Accurate Single-Image Depth Estimation on Mobile Devices, Mobile AI 2021 Challenge: Report, 2021. 1, 2, 5

[15] Andrey Ignatov, Radu Timofte, Andrei Kulik, Seungsoo Yang, Ke Wang, Felix Baum, Max Wu, Lirong Xu, and Luc Van Gool. AI benchmark: All about deep learning on smartphones in 2019. *CoRR*, abs/1910.06663, 2019. 5

[16] Yevhen Kuznietsov, Jörg Stückler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. *CoRR*, abs/1702.02706, 2017. 2

[17] Zhengqi Li and Noah Snavely. MegaDepth: Learning Single-View Depth Prediction from Internet Photos. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[18] Fangchang Ma and Sertac Karaman. Sparse-to-Dense: Depth Prediction from Sparse Depth Samples and a Single Image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4796–4803, 2018. 1

[19] Alican Mertan, Damien Jade Duff, and Gozde Unal. Single Image Depth Estimation: An Overview. *CoRR*, abs/2104.0, 2021. 1

[20] Yue Ming, Xuyang Meng, Chunxiao Fan, and Hui Yu. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, 2021. 1

[21] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 1, 2

[22] Luis Enrique Ortiz, Elizabeth V Cabrera, and Luiz M Gonçalves. Depth data error modeling of the zed 3d vision sensor from stereolabs. *ELCVIA: electronic letters on computer vision and image analysis*, 17(1):0001–15, 2018. 3

[23] Onur Özyeşil, Vladislav Voroninski, Ronen Basri, and Amit Singer. A survey of structure from motion. *Acta Numerica*, 26:305–364, 2017. 1

[24] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[25] Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Learning 3-d scene structure from a single still image. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007. 2

[26] Johannes L Schonberger and Jan-Michael Frahm. Structure-From-Motion Revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, jun 2016. 1

[27] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *CoRR*, abs/1905.1, 2019. 5

[28] Prune Truong, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning Accurate Dense Correspondences and When to Trust Them. *CoRR*, abs/2101.0, 2021. 4

[29] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. 1, 2

[30] Diana Wofk, Fangchang Ma, Tien Ju Yang, Sertac Karaman, and Vivienne Sze. FastDepth: Fast monocular depth estimation on embedded systems. *Proceedings - IEEE International Conference on Robotics and Automation*, 2019-May:6101–6108, mar 2019. 5

[31] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. *CoRR*, abs/1803.0, 2018. 1, 2

[32] Mehmet Kerim Yucel, Valia Dimaridou, Anastasios Drosou, and Albert Saà-Garriga. Real-time monocular depth estimation with sparse supervision on mobile. *CoRR*, abs/2105.12053, 2021. 2

[33] Chao Qiang Zhao, Qi Yu Sun, Chong Zhen Zhang, Yang Tang, and Feng Qian. Monocular depth estimation based on deep learning: An overview. *Science China Technological Sciences*, 63(9):1612–1627, 2020. 1

[34] Xiao-Yun Zhou, Jian-Qing Zheng, and Guang-Zhong Yang. Atrous Convolutional Neural Network (ACNN) for Biomedical Semantic Segmentation with Dimensionally Lossless Feature Maps. *CoRR*, abs/1901.0, 2019. 1