VEHICLE DEPTH ESTIMATION FOR AUTONOMOUS DRIVING

David-Traian IANCU[1], Mihai NAN[2], Alexandra Ștefania GHIȚĂ[3], Adina-Magda FLOREA[4]

In the latest years, it has been done a lot of research regarding the depth estimation of the objects in an image. In autonomous driving, one of the most important tasks is to estimate the distance to the surrounding cars. This paper analyzes the state of the art regarding the depth estimation from single and stereo sources and the datasets that were made for these tasks and proposes a new depth dataset, recorded with an Intel RealSense depth camera. This dataset is used together with another dataset made previously in our university in order to compare one of the most used neural networks for depth prediction from a single camera, for both full image depth estimation and vehicle-only depth estimation. The results are computed regarding the Root Mean Square Error (RMSE) and take in account the time of the day (day, dusk or night), the inference time and the dimension of the cars.

Keywords: depth estimation, autonomous driving, neural networks, RGB-D dataset

1. Introduction

Autonomous driving is one of the most challenging tasks in the latest years, for both practical and theoretical reasons. There are a lot of components regarding an autonomous car – scene understanding, motion control, path following, decision making, etc. In our research center at Politehnica University of Bucharest we aim to make an autonomous car. We begin with the scene understanding and in our previous studies [1,2] we discussed the object detection and the semantic segmentation. In each study, we analyzed the most important works available now and we tested some of the best networks against our datasets, discussing the results by considering the light (day, dusk or night). This study is the final one from our scene understanding series and analyzes the depth estimation task. We made a new depth dataset recorded with an Intel RealSense depth camera and we tested some of the best networks against it, again regarding

[1] As., Dept.of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: david_traian.iancu@upb.ro
[2] As., Dept.of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: mihai.nan@upb.ro
[3] As., Dept.of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: stefania.a.ghita@upb.ro
[4] Prof., Dept.of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: adina.florea@upb.ro

the time of the day. The biggest challenge is to estimate and compare the depth maps without having a ground truth. To tackle this task, we tested the networks on our dataset used for segmentation, varying the ground truth as being the results obtained by some of the networks tested, knowing from the previous experiments what are the best networks. For both datasets, we tested the depth for all the pixels in the image and the depth considering only the surrounding vehicles. In Section 2, we analyze the most important works, we discuss the datasets in Section 3, we present the experiments in Section 4 and the results in Section 5. The conclusion is presented in Section 6.

2. Related work

This section will overview the reviews that were made regarding this task, considering the analysis of the algorithms from the perspective of autonomous driving and how we compare with them. Also, we describe here the most important depth estimation architectures. We divided them into two big categories – depth estimation networks that work with images obtained from stereo cameras and depth estimation networks that work with images obtained from monocular cameras. Our experiments only use the second ones, considering that the task of monocular depth estimation is more complicated and better to resolve regarding autonomous driving, where you want to minimize the hardware involved in the processing of the images.

2.1. Depth estimation reviews

Unlike the previous tasks of object detection and segmentation, the depth estimation has not been reviewed in many papers, because the importance of object detection and segmentation is bigger. However, there are some articles that make a survey on the existing architectures. For example, in [3], there is a small survey on five existing architectures for monocular depth estimation, only briefly discussing the stereo depth estimation. They present the architectures of the selected networks, some of their results and some classical dataset, without further analyzing other factors such as the light or introducing a new dataset. They compare for supervised network and one unsupervised network. We can see a new, interesting article at [4]. They divide the depth methods into geometric based methods, such as Structure from Motion [5], sensor-based methods, such as the LIDAR, and deep learning methods. They present some of the most important datasets used in depth, some metrics (including the RMSE which we used in this paper), and some of the networks that we used. They further divide the monocular networks into unsupervised, semi-supervised in supervised methods and analyze some of the most important networks. They consider that the semi-supervised networks are those that learn from stereo images, but without knowing the ground truth of the depths, and the unsupervised networks are those that learn from

monocular sequences, without any information regarding the depth. They evaluate some of the unsupervised and weakly supervised networks on KITTY and slightly discuss the time and some application, but no other tests or discussions are made. In [6] we can see a survey regarding stereo models. They also make a difference between old methods, based on pixels matching, and deep learning algorithms. The study offers o comprehensive view regarding depth datasets, they further divide the stereo methods into 3 categories – depth estimation by stereo matching, by end-to-end training and multi-view stereo methods. They also test some inference times, but they don't offer any results. We can see another comprehensive stereo review of older architectures in [7]. Although they present a lot of networks and architectures, including implementations on dedicated hardware, they discuss the inference time and results briefly, only presenting some theoretical aspects regarding the architectures. Our review also offers a new depth dataset, perform a comparative analysis of some state-of-the-art methods by considering some of them as ground truth, discusses the datasets by taking the light into account and offers information regarding the inference time.

2.2. Depth estimation from stereo camera

In this subsection, we analyze the most important works that tackle the problem of depth estimation given stereo images for the training set with their corresponding ground truth. We must make the distinction that there are not just the stereo images, the ground truth is included, that approach will be considered unsupervised learning and analyzed in the following subsection. Following the work from [6], we will further present works that try to do stereo matching, end-to-end training and multi-view stereo estimation.

The older methods did not use deep learning. Although there are not used anymore, there are worth mentioning because some of the techniques are used in the deep networks. In [8] the depth is computed from stereo images and monocular cues also from the images, with a Markov Random Field algorithm. In [9] we can see a very old approach that tries to make similar values in a disparity map regarding the same object, which is predicted with a segmentation algorithm. In [10] they compute the stereo matching in a robot environment, with particle filtering. In [11] and [12] we can see some implementation of depth estimation in hardware, with FPGAs.

The most popular approaches in stereo depth estimation are to try to learn the matching between the two images and to create the disparity map, which is very similar with the human eye depth perception or to learn the matching in a pure neural end-to-end approach. In the latest years, the second approach has increased its popularity due to simplicity. In [13], the authors claim that monocular depth estimation is not good enough for autonomous driving and they propose a stereo architecture, which has different layers to compute the left-right

matching and the right-left matching and uses a semi-supervised loss function, taking in account both LIDAR ground truth and unsupervised components. Older approaches using CNNs for matching can be found at [14] and [15], which are very similar architectures, the first one uses pooling layers and the second one does not. A more recent network can be seen at [16], which proposes a complex architecture for stereo matching with multilevel skip connections, another deep neural network for predicting the confidence of a disparity map and a final refinement step. Another stereo matching algorithm is [17], which different approaches for the refinement step and the matching cost, which is computed with two independent networks with multi-layer pooling modules.

As we said earlier, in the latest years the usage of end-to-end networks increased. In [18] we can see stereo training using optical flow. They also made a synthetic dataset for the task. In [19] we can see a CNN for end-to-end training, with the loss function based on the warping errors. In [20] there is an end-to-end model based on a CNN architecture combined with conditional random fields (CRFs). [21] proposes a recurrent model which tries to minimize the left-right consistency and improve the disparity maps. [22] claims to make stereo end-to-end training with less memory usage and a wider range of image sizes, by modifying the hourglass network with a bottleneck matching module. [23] proposes a new attention mechanism which improves the stereo depth estimation task. In [24] we can see an anytime model, which produces an initial disparity map, then refines it, made for mobile devices. In [25] the training process is also refined with some sparse, cheap, LIDAR sensors. [26] tries a new approach, by learning a cost volume from the data, then regressing the disparities from it. One of the newest architectures for this task, MADNet [27], uses a mechanism which trains independently only portions of the network, combined with a self-adaptive unsupervised network, which can adapt to any environment.

The last category of stereo networks consists of multi-view stereo training, which has multiple views of the same objects, to better estimate the 3D model. We can see this approach in [28], using Conditional Random Fields, which is based on [29], which makes an initial depth estimation based on features from all the images, then refines it with the reference image. Also, [30] could be considered an old multi-view stereo training, because it tries to regress the depth model based on a small motion clip.

2.3. Depth estimation from monocular camera

In this subsection, we analyze the most recent works regarding the depth estimation from a monocular camera. We divide the architectures as in [4], given the training data, in three categories – unsupervised learning, semi-supervised learning and supervised learning. In our experiments, we used supervised

learning, having pretrained networks and the ground truth from the RGB-D camera.

The first models that tried to learn the disparity maps used some techniques that currently apply to the neural networks, but without using deep learning, for example, optical flow with local cues [31], stereo and defocus cues [32], albedo and shading [33].

One of the most used networks is Monodepth [34], which uses a convolutional network which tries to have left-right depth consistency in the training process. It also has an improvement which we used in our tests, Monodepth 2 [35]. Another model used for our tests, Megadepth [36], learns the depth from photos downloaded from the internet, combined with multi-view stereo methods, and proposes a new dataset, too. The loss function used is composed from a scale-invariant data loss, a gradient-matching loss and an ordinal depth loss:

$$L_{Megdepth} = L_{grad} + \alpha L_{grad} + \beta L_{ord} \quad (1)$$

SfMLearner [37] uses view synthesis for the network supervision, without needing stereo images even for the training. Dense Depth [38] uses an encoder-decoder architecture for monocular depth estimation. Their loss function is also composed from three different terms – a loss for the depth values, a loss for the gradient of the depth image and a structural similarity loss:

$$L_{DenseDepth} = \lambda L_{depth} + L_{grad} + L_{SSIM} \quad (2)$$

DORN [39] tackles the depth estimation task as a regression problem and tries to minimize the mean square error also uses a strategy to discretize the depth. They use an ordinal loss, which is the average of pixelwise ordinal loss over the image:

$$L_{DORN} = -\frac{1}{N}\sum_{w=0}^{W-1}\sum_{h=0}^{H-1} L_{ord}(w,h) \quad (3)$$

LKVOLearner [40] is based on another method, the direct visual odometry [41]. A newer method [42] combines depth estimation with ego-motion learning by taking some existing architectures, like [18], and adding a new loss function. AdaBins [43] uses an encoder-decoder model to estimate the depth ranges, which are divided into some bins. [44] is a recent network which offers another encoder-decoder CNN architecture with an improved decoder. [45] learns the depth values by using the dual pixel autofocus hardware which exists on current cameras. [46] is an encoder-decoder model used for monocular prediction on embedded systems. [47] estimates the depths using an image but also some sparse estimations from low depth sensors. The use of CRFs can also be found in some recent networks for estimating the depth using monocular cameras [48,49]. An older interesting approach can be found at [50], which uses two networks – one for an initial prediction and the other one for refinement.

Even if most of the networks are using CNNs, there are also models that use Recurrent Neural Networks (RNNs) [51], a Variational Autoencoder (VAE) combined with a CNN [52] or Generative Adversarial Networks (GANs) [53, 54], with the mention that [53] is trained unsupervised. Another interesting model can be found at [55], which solves three problems at once – semantic segmentation, instance segmentation and depth estimation, based on a modified version of a semantic segmentation architecture, ENet. [56]

The next networks mentioned are trained in a semi-supervised fashion. A semi-supervised model based on Monodepth, with LIDAR point clouds and stereo images, using a loss function that enforces left-right consistency, can be seen at [57]. A model for self-supervised learning that is based on stereo images and semantic segmentation can be found at [58]. Other semi-supervised models that use both LIDAR and stereo images can be found at [59,60]. In [61] the semi-supervised training is done with sparse 3D data taken from a laser sensor. There are also some unsupervised networks for this task. The approach proposed in [62] is an unsupervised framework based on multiple neural networks that collaborate between them to recreate depth, motion segmentation and optical flow. Another interesting model, based on a stack of GANs, can be found at [63]. [64] trains a CNN using stereo images for unsupervised depth estimation. [65] is based on SfMLearner [37] but trains the data in an unsupervised fashion, changing the loss function in order to incorporate the generated 3D scene. A similar approach, based on the geometry of the image, but with an additional refinement step which improves the prediction, is found at [66]. Finally, at [67] we can see another unsupervised model that learns the depth, egomotion and camera intrinsics based on stereo images. The architecture is based on Unet [68].

3. Datasets

One of the most challenging tasks regarding the depth estimation is the recording of a dataset, because acquiring depth maps require expensive hardware. This is the reason for the existence of a lot of unsupervised or semi/self-supervised networks. The depth data is expensive, there are not enough images and probably the biggest problem is that not all the depth maps are reliable, especially if taken with cheap sensors. In this section, we review the most used depth estimation datasets, and we also present our dataset for this task, which made possible the results studied in this article.

3.1. Depth datasets

Even if there are not so many datasets as for image recognition, object detection and semantic segmentation, we can find some data in order to train the depth networks. The most used dataset for depth is KITTY – their latest dataset [69] contains over 94 thousand RGB images annotated with depth. Another

popular dataset is NYUv2 [70], which contains around 1500 annotated images taken from 3 cities. Make3D [71] have 400 depth images for training and 134 for testing. Sun RGB-D [72] is another used dataset for training and testing, with over 10.000 images annotated with depth.

Beside these datasets, there are also some datasets less used but with large amounts of data, for example. Megadepth [36], which contains about 130 thousand images. Another new and big dataset is DrivingStereo [73], with over 180 thousand images. Sceneflow [18] offers almost 35k images, of which over 4000 are driving scenes. Middlebury [74] offers 33 dense annotated images with depth. There are also some datasets used for multi-view stereo. The most used are DTU, with 124 scenes [75], and Tanks and Temples [76], which contains thousands of images.

3.2. POLI Depth dataset

Our dataset was taken with an Intel RealSense RGB-D camera. Following the idea from our previous articles, we divided the data into 3 sets – images recorded during the day, images recorded during the dusk or dawn and images recorded during the night. We recorded some video sequences in the Politehnica University campus during different times of the day. The dataset contains 516 images for the day, 1039 for the dusk/ dawn and 637 for the night, totaling 2192 annotated images. All the images have corresponding depth maps attached. The dataset was obtained by driving a personal car around our university campus during different times of the day. There were some challenges regarding the Intel camera – the depth frames did not have the same size as the image frames (some of the depth frames were lost), which required multiple recordings and an adjustment between the image frames and the depth frames. Also, the camera had lower quality maps during the night.

The RealSense D435 camera recorded photos of high definition resolution (1280x720 pixels), however the masks were recorded at 848x480 pixels, which required a preprocessing step which resized the images to the same size. The resolution of the frames had also been resized regarding the model used (the smaller resolution had to be chosen). The camera was mounted on the windscreen of the car when the frames were recorded.

4. Implementation and experiments

We made multiple experiments in order to evaluate some of the most used monocular depth estimation networks. For the experiments, we used two datasets – our depth dataset described in the previous section and a dataset without a corresponding ground truth, taken from our semantic segmentation article. The dataset contains 735 images recorded in the day, 133 images recorded in dusk and

165 images recorded in the night. Even if there is no ground truth, one of the purposes of our experiments was to evaluate the networks related to each other, i.e. considering as ground truth one of the networks itself. We tested with the recorded ground truth in order to see how the networks are performing, then we checked the relative performance by considering the ground truth as being the result from a specific network. For ground truth, we considered the results of Megadepth, Monodepth and Dense Depth, beside the result of the Intel camera. For both datasets we computed the Round Mean Squared Error (RMSE) considering the pixel values of the black and white images:

$$RMSE = \sqrt{\left(\frac{\sum_{i=1}^{N}(predicted_i - real_i)^2}{N}\right)} \quad (4)$$

The RMSE was computed for the day, dusk, night and for the whole dataset. In order to see the network performances for autonomous driving, we made two different experiments. The first experiment considered all the pixels of the images and the second one was more driving-related because we considered only the depth estimation of the cars that were found in the images. All the cars were manually annotated in both datasets. In the first set, we have 1733 cars in the day, 3582 in the dusk and 1375 in the night. In the second set, we have 1491 cars in the day, 256 in the dusk and 342 in the night. We also computed the inference time regarding different image sizes, as it can be seen in the next section, and we divided the car sizes into 13 categories, in order to see if the depth estimation is influenced by the size of the images.

The testing stream works in the following way: first, the Intel RealSense camera captures the depth maps and the RGB frames. The RGB frames are further going to a preprocessing step, in order to fit the requirements from the depth architectures. The frames are processed by the depth estimation architectures, which outputs a depth estimation map, which is further compared to the recorded depth by a validation module. The testing stream can be seen in Fig. 1.

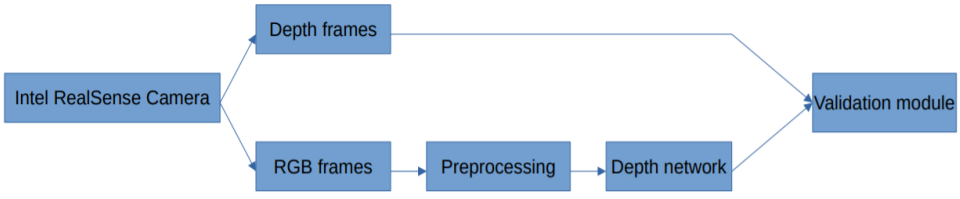

Fig. 1. Testing stream

5. Results

In this section, we discuss the results for our experiments. The first experiments were made using the whole images and computing the RMSE regarding the whole image. In Set 1, we tested the results of the networks against the ground truth taken with the RealSense camera, but we also tested the networks against each other. In Set 2 there is no ground truth data, but we tested the networks against each other in order to see if the qualitative results are conserved. The second experiments used the same networks and datasets, but were considered only the cars, so the results are better from obvious reasons – less pixels that can increase the RMSE.

In Table 1 we can see the results for Set 1. For the ground truth, we have the real ground truth, recorded with the RealSense camera, and we also considered the ground truth as being the results from Megadepth, Dense Depth and Monodepth. The reason for this choice is that we manually looked at the results and we have chosen the networks that behaved the best. Dense Depth and DORN have the best results regarding the ground truth, but because the RealSense camera is not very precise the results of Megadepth and Monodepth were not that good. We can also see that generally, the worst results are in the dusk, considering the ground truth the RealSense camera. For Dense Depth ground truth, we can see that the closest network is LKVOLearner, with Megadepth being the ground truth the closest ones are SfMLearner and LKVOLearner, and with Monodepth again SfMLearner and LKVOLearner were the closest ones. We can see that for Dense Depth as ground truth the best results were in the night, but for the rest of the networks, the best results were in the day and the worst in the night.

In Table 2 we can see the depth results for the second set, with no annotated ground truth, like an unsupervised network. For Dense Depth, the best network was LKVOLearner, the best results were in the night. For Megadepth, the best networks were LKVOLearner and SfMLearner, with Monodepth being a close call, and for Monodepth, the same applied – LKVOLearner and SfMLearner were the best, with Megadepth being a close call. Generally, the best results were in the day, the worst in the night, excepting the ground truth being considered Dense Depth, where the best results were generally achieved in the night, excepting form Megadepth, where the best results were in the day. The results vary a lot because the networks are different and the ground truth is not precise, even with the RealSense camera, which made us compare the networks against each other, in an unsupervised fashion, in order to see how the networks behave comparing to multiple depth estimations. Also, in Table 1 and Table 2 we can see the same experiments but taking in account only the cars when computing the RMSE. This benchmark is especially important when speaking about autonomous cars – in an autonomous car, the depth estimation is important because the

software should know about the estimated distance from the car to other cars (and people, too, but in a perfect environment people would cross the street only at crosswalks), in order to know when to brake. Regarding the depth camera from the first set, the results are close, with SfMLearner and LKVOLearner being, again, the best. Also, the best results are not clearly in the day or in the night, but the worst results are in the dusk. For Dense Depth, the best results were from Megadepth and LKVOLearner and again the worst were obtained in the dusk. With Megadepth as ground truth, the results are different – the best are obtained in the dusk. However, SfMLearner and LKVOLearner have again the best performances. With Monodepth as ground truth, the best results were obtained by DORN, with SfMLearner very close, and the worst results are obtained in the night, in general. In Table 2, we can also see the metrics for the second set, taking only the cars in the account. With Dense Depth as ground truth, we can see that LKVOLearner and Megadepth have de best results, and the worst results are in the dusk. With Megadepth as the reference network, the best results are obtained by SfMLearner and DORN, the best results are generally during the day and the worst during the dusk. Finally, with Monodepth as the main network, the results are very good - the best network is SfMLearner, seconded by DORN, and the best results are in the day, the worst in the dusk or in the night. The last two experiments were regarding the size of the objects and the speed of the networks.

Table 1

Depth results – set 1 (RMSE)

Model	**Day**	**Dusk**	**Night**	**Avg**	**Day (car only)**	**Dusk (car only)**	**Night (car only)**	**Avg**
Ground truth - depth camera								
Megadepth	128.12	140.99	139.38	137.59	47.51	71.74	68.75	65.66
DORN	72.51	98.46	55.39	82.00	58.31	84.97	34.84	70.68
LKVOLearner	98.36	109.63	97.29	103.56	47.68	69.96	48.25	60.73
SfMLearner	113.91	126.74	109.32	118.92	51.33	73.68	41.18	62.75
Monodepth	122.81	135.66	120.28	128.37	56.77	84.92	42.34	71.17
Dense Depth	82.96	83.85	87.65	84.77	62.80	59.18	71.31	62.79
Ground truth - Dense Depth								
Megadepth	105.37	109.40	90.15	103.19	52.76	60.23	29.84	53.35
DORN	90.96	93.09	85.26	90.38	74.62	75.19	65.95	73.24
LKVOLearner	80.78	79.69	64.99	75.98	54.25	56.34	40.38	52.89
SfMLearner	96.37	100.18	84.74	95.03	60.48	61.40	50.82	59.13
Monodepth	111.97	111.97	100.42	108.74	71.12	75.73	57.89	71.20
Ground truth – Megadepth								
DORN	125.66	130.95	139.30	132.23	45.89	30.01	61.68	42.54

LKVOLearner	47.89	54.03	57.44	53.69	24.39	17.72	31.53	22.95
SfMLearner	45.13	52.83	60.04	53.38	33.71	19.60	43.24	29.74
Monodepth	55.49	64.44	70.30	64.26	38.05	27.08	48.93	35.49
Dense Depth	105.37	109.40	90.15	103.19	52.76	60.23	29.84	53.35
Ground truth – Monodepth								
Megadepth	55.49	64.44	70.30	64.26	38.05	27.08	48.93	35.49
DORN	104.46	100.40	114.93	105.76	19.61	15.08	25.23	18.77
LKVOLearner	44.49	46.07	51.81	47.46	22.75	22.41	24.00	22.83
SfMLearner	42.98	42.74	47.87	44.35	20.27	18.65	21.29	19.64
Dense Depth	111.97	111.97	100.42	108.74	71.12	75.73	57.89	71.20

In Fig. 2 we can see the RMSE of the networks regarding the car size. The results shown are from the first set, the second set has similar properties. We can see that for very small car sizes the results vary a little, then the RMSE decreases as the car size increases – the depth estimation is better for bigger cars. The RMSE decreases for all the networks.

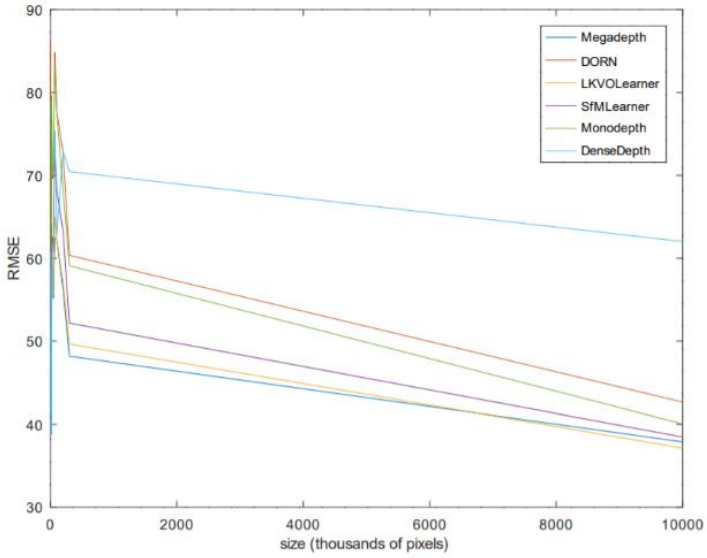

Fig. 2. RMSE regarding car size

In Fig. 3 we measured the time regarding the size of the images. We made several images with different dimensions, from 480x360 pixels to 4k. We can see from the plot that most of the networks have similar times regarding the image size, only a small increase for bigger sizes. However, Monodepth and Megadepth have a linear increase in time, with Megadepth performing the worst.

Table 2

Depth results – set 2 (RMSE)

Model	**Day**	**Dusk**	**Night**	**Avg**	**Day (car only)**	**Dusk (car only)**	**Night (car only)**	**Avg**
Ground truth - Dense Depth								
Megadepth	99.52	101.37	104.43	100.56	52.72	59.24	44.25	52.27
DORN	91.49	92.99	86.44	90.90	60.60	67.36	62.29	61.75
LKVOLearner	81.77	82.27	72.82	80.47	51.14	58.69	43.84	51.01
SfMLearner	101.46	101.56	88.81	99.56	59.87	67.36	46.28	58.84
Monodepth	114.31	115.04	105.41	113.03	64.97	74.54	56.72	64.95
Ground truth – Megadepth								
DORN	114.81	120.19	126.21	117.40	20.12	21.49	25.10	52.27
LKVOLearner	40.10	46.43	51.42	42.94	12.13	13.45	13.38	61.75
SfMLearner	38.57	44.07	41.18	39.74	15.30	17.41	16.05	51.01
Monodepth	45.79	52.42	59.19	49.04	19.95	22.34	20.21	58.84
Dense Depth	99.52	101.37	104.43	100.56	52.72	59.24	44.25	64.95
Ground truth – Monodepth								
Megadepth	45.79	52.42	59.19	49.04	19.95	22.34	20.21	20.30
DORN	104.57	104.59	100.99	104.01	13.15	16.75	11.72	13.42
LKVOLearner	39.32	40.10	45.96	40.55	15.10	16.81	16.35	15.53
SfMLearner	27.72	28.66	44.76	31.18	7.53	9.08	15.33	9.45
Dense Depth	114.31	115.04	105.41	113.03	64.97	74.54	56.72	64.95

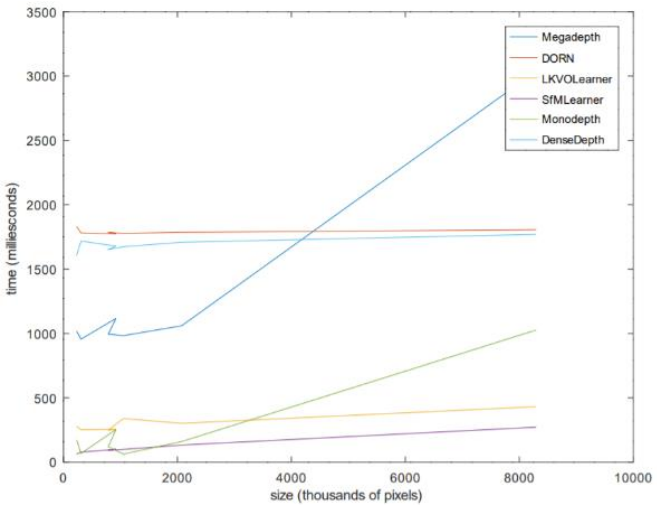

Fig. 3. Inference time regarding car size

All the experiments were made using a personal computer with an Nvidia GTX 2060 graphic card, in order to keep the proportion of the inference times. Some examples of the recorded frames in day, dusk and night, together with the corresponding depth mask recorded by the camera and an estimated depth (by Dense Depth) can be seen in Fig. 4.

Fig. 4. Depth estimation examples

6. Conclusion

In this article, we made a comprehensive study regarding the depth estimation, focusing on the benefits regarding the autonomous driving. We analyzed the most influential works regarding both stereo and monocular depth estimation, further dividing the stereo estimation into stereo matching, an end-to-

end learning and multi-view stereo estimation and the monocular estimation into supervised, semi-supervised and unsupervised monocular estimation. We tested some of the most used monocular depth estimation networks, because we believe that this category has the biggest advantages regarding autonomous driving, where there is not always a stereo camera system. Our contribution is that we made a new dataset for depth estimation and tested the networks against each other, in an unsupervised fashion, in order to see how they compare tested on multiple variants for the ground truth. Also, we tested the networks on a previous dataset made in our campus, without a ground truth, in order to estimate the results in an unsupervised fashion by varying the ground truth as the depth estimation of the best networks tested. We also considered the daylight in our experiments. We tested 6 networks – DORN, LKVOLearner, SfMLearner, Monodepth 2, Megadepth and Dense Depth. In our tests, LKVOLearner and SfMLearner have the best results, with Monodepth 2 being another close call. We found that the networks tend to have better results with bigger object sizes and that most of the networks behave well regarding bigger image sizes, though they can be used in real-life applications with full HD or even 4K cameras. Another conclusion is that the worst results are in the night and the best results are during the day. The tested networks are state of the art depth estimation architectures, having one of the best results regarding other depth estimation networks, as it can be seen from experiments in peer-reviewed published articles. Compared with traditional methods, which do not involve neural networks, the qualitative difference is huge regarding the more modern deep learning approaches. We believe that only deep architectures could be used in real life scenarios, regarding the inference time and the quality. In our future projects, we will use one of the best networks tested in order to estimate the depth.

Acknowledgement

We want to thank to the Politehnica University of Bucharest which made this study possible, for offering the recording hardware and the permissions to make the datasets and for offering the infrastructure for some of the tests, too. This research was funded by grants PN-III-P2-2.1-PED-2019-4995 and PN-III-P1-1.2-PCCDI-2017-0734.

R E F E R E N C E S

[1]. *D. Iancu, A. Sorici and A. M. Florea.* Object detection in autonomous driving - from large to small datasets, 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2019.

[2]. *D. Iancu, A. Sorici and A. M. Florea*, Neural road segmentation in driving scenarios, 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2020.

[3]. *Bhoi, A*, Monocular depth estimation: A survey, arXiv preprint arXiv:1901.09402, 2019.

[4]. *Chaoqiang Zhao, Qiyu Sun, Chongzhen Zhang, Yang Tang, and Feng Qian*, Monocular depth estimation based on deep learning: An overview, arXiv preprint arXiv:2003.06620, 2020.
[5]. *Ullman S*, The interpretation of structure from motion, Proc R Soc Lond B, 1979.
[6]. *Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun*, A survey on deep learning techniques for stereo-based depth estimation, arXiv preprint arXiv:2006.02535, 2020.
[7]. *Hamzah, R.A., Ibrahim, H*, Literature survey on stereo vision disparity map algorithms, Journal of Sensors, 2016
[8]. *A. Saxena, J. Schulte, and A. Y. Ng*, Depth estimation using monocular and stereo cues, Proc. IEEE Int. Joint Conf. Artificial Intelligence, 2007.
[9]. *S. Y. Park, S. H. Lee, and N. I. Cho*, Segmentation based disparity estimation using color and depth information, Proceedings of the International Conference on Image Processing (ICIP '04), vol. 5, pp. 3275–3278, Singapore, 2004.
[10]. *S. Ploumpis, A. Amanatiadis, and A. Gasteratos*, A stereo matching approach based on particle filters and scattered control landmarks, Image and Vision Computing, vol. 38, pp. 13–23, 2015.
[11]. *C. Colodro-Conde, F. J. Toledo-Moreo, R. Toledo-Moreo, J. J. Martínez-Álvarez, J. Garrigós Guerrero, and J. M. Ferrández-Vicente*, Evaluation of stereo correspondence algorithms and their implementation on FPGA, Journal of Systems Architecture, vol. 60, no. 1, pp. 22–31. 2014.
[12]. *W. Wang, J. Yan, N. Xu, Y. Wang, and F.-H. Hsu*, Real-time high-quality stereo vision system in FPGA, Proceedings of the 12th International Conference on Field-Programmable Technology (FPT '13), pp. 358–361, Kyoto, Japan, 2013.
[13]. *S. B. Nikolai Smolyanskiy, Alexey Kamenev*, On the Importance of Stereo for Accurate Depth Estimation: An Efficient Semi-Supervised Deep Neural Network Approach, arXiv Prepr. arXiv1803.09719v1, 2018.
[14]. *S. Zagoruyko and N. Komodakis*, Learning to compare image patches via convolutional neural networks, CoRR, abs/1504.03641, 2015.
[15]. *J. Zbontar and Y. LeCun*, Stereo matching by training a convolutional neural network to compare image patches, arXiv preprint arXiv:1510.05970, 2015.
[16]. *A. Shaked and L. Wolf*, Improved stereo matching with constant highway networks and reflective confidence learning, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[17]. *X. Ye, J. Li, H. Wang, H. Huang, and X. Zhang*, Efficient stereo matching leveraging deep local and context information, IEEE Access, 5:18745–18755, 2017.
[18]. *N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox*, A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, CVPR, 2016.
[19]. *Zhong, Y., Dai, Y., Li, H*, Self-supervised learning for stereo matching with self improving ability, arXiv preprint arXiv:1709.00930, 2017.
[20]. *Knobelreiter, P., Reinbacher, C., Shekhovtsov, A., Pock, T*, End-to-end training of hybrid cnn-crf models for stereo, CVPR, 2017.
[21]. *Zequn Jie, Pengfei Wang, Yonggen Ling, Bo Zhao, Yunchao Wei, Jiashi Feng, and Wei Liu*, Left-right comparative recurrent model for stereo matching, CVPR, 2018
[22]. *S. Tulyakov, A. Ivanov, and F. Fleuret*, Practical deep stereo (pds): Toward applications-friendly deep stereo matching, arXiv preprint arXiv:1806.01677, 2018
[23]. *Shen, Z., Zhang, M., Zhao, H., Yi, S., and Li, H*, Efficient attention: Attention with linear complexities, arXiv preprint arXiv:1812.01243, 2020.

[24]. *Y. Wang, Z. Lai, G. Huang, B. H. Wang, L. van der Maaten, M. Campbell, and K. Q. Weinberger*, Anytime stereo image depth estimation on mobile devices, arXiv preprint arXiv:1810.11408, 2018.
[25]. *Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger*, Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving, ICLR, 2020.
[26]. *A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry*, End-to-end learning of geometry and context for deep stereo regression, IEEE International Conference on Computer Vision (ICCV), 2017.
[27]. *Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano*, Real-time self-adaptive deep stereo, CVPR, 2019
[28]. *Youze Xue, Jiansheng Chen, Weitao Wan, Yiqing Huang, Cheng Yu, Tianpeng Li, and Jiayu Bao*, Mvscrf: Learning multi-view stereo with conditional random fields, Proceedings of the IEEE International Conference on Computer Vision, 2019.
[29]. *Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan*, Mvsnet: Depth inference for unstructured multi-view stereo, ECCV, 2018.
[30]. *Ha, H., Im, S., Park, J., Jeon, H.G., Kweon, I.S*, High-quality depth from uncalibrated small motion clip, Proc. of Computer Vision and Pattern Recognition, CVPR, 2016.
[31]. *K. Karsch, C. Liu, and S. B. Kang*, Depth extraction from video using non-parametric sampling, Proc. Eur. Conf. Comp. Vis., pp. 775–788, 2012.
[32]. *A. Rajagopalan, S. Chaudhuri, and U. Mudenagudi*, Depth estimation and image restoration using defocused stereo pairs, TPAMI, 2004.
[33]. *N. Kong and M. J. Black*, Intrinsic depth, Improving depth transfer with intrinsic images, ICCV, 2015.
[34]. *Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao*, Deep ordinal regression network for monocular depth estimation, CVPR, 2018.
[35]. *Godard, C., Mac Aodha, O., Brostow, G*, Digging into self-supervised monocular depth estimation, arXiv preprint arXiv:1806.01260, 2018.
[36]. *Zhengqi Li and Noah Snavely*, Megadepth: Learning single view depth prediction from internet photo, CVPR, 2018.
[37]. *T. Zhou, M. Brown, N. Snavely, and D. Lowe*, Unsupervised learning of depth and ego-motion from video, CVPR, 2017.
[38]. *I. Alhashim and P. Wonka*, High quality monocular depth estimation via transfer learning, arXiv e-prints, abs/1812.11941, 2018.
[39]. *Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao.* Deep ordinal regression network for monocular depth estimation, CVPR, 2018.
[40]. *Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey*, Learning depth from monocular videos using direct methods, CVPR, 2018.
[41]. *F. Steinbruecker, J. Sturm, and D. Cremers*, Real-time visual odometry from dense RGB-D images, Proc. Int'l Conf. Computer Vision Workshops, 2011
[42]. *J. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid*, Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video, NeurIPS, 2019.
[43]. *Bhat, Shariq Farooq, Ibraheem Alhashim, and Peter Wonka*, AdaBins: Depth Estimation using Adaptive Bins. 2020.
[44]. *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:1907.10326, 2019
[45]. *R. Garg, N. Wadhwa, S. Ansari, and J. T. Barron*, Learning single camera depth estimation using dual-pixels, ICCV, 2019
[46]. *Y. Zhang and T. Funkhouser*, Deep depth completion of a single rgb-d image, CVPR, 2018.

[47]. *F. Ma and S. Karaman*. Sparse-to-dense: Depth prediction from sparse depth samples and a single image, arXiv preprint arXiv:1709.07492, 2017
[48]. *Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe*, Monocular depth estimation using multi-scale continuous crfs as sequential deep networks, TPAMI, 2018.
[49]. *Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci*, Structured attention guided convolutional neural 9808 fields for monocular depth estimation, CVPR, 2018.
[50]. *D. Eigen, C. Puhrsch, and R. Fergus*, Depth map prediction from a single image using a multi-scale deep network, arXiv preprint arXiv:1406.2283, 2014.
[51]. *Wang, R., Pizer, S.M., Frahm, J.M*, Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth, CVPR, 2019.
[52]. *P. Chakravarty, P. Narayanan, and T. Roussel*, GEN-SLAM: Generative modeling for monocular simultaneous localization and mapping, arXiv:1902.02086v1, 2019.
[53]. *Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia*, Generative adversarial networks for unsupervised monocular depth prediction, ECCV, 2018.
[54]. *Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju*, Spatial correspondence with generative adversarial network: Learning depth from monocular videos, ICCV, 2019.
[55]. *D. Neven, B. D. Brabandere, S. Georgoulis, M. Proesmans, and L. V. Gool*, Fast scene understanding for autonomous driving, IV Symposium Workshop, 2017.
[56]. *Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello*, Enet: A deep neural network architecture for real-time semantic segmentation, arXiv preprint arXiv:1606.02147, 2016.
[57]. *Ali Jahani Amiri, Shing Yan Loo, and Hong Zhang*, Semi-supervised monocular depth estimation with left-right consistency using deep neural network, arXiv preprint arXiv:1905.07542, 2019.
[58]. *Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon*, Semantically-Guided Representation Learning for Self-Supervised Monocular Depth, ICLR, 2020.
[59]. *L. He, C. Chen, T. Zhang, H. Zhu, and S. Wan*, Wearable Depth Camera: Monocular Depth Estimation via Sparse Optimization Under Weak Supervision, IEEE Access, vol. 6, pp. 41337–41345, 2018.
[60]. *Y. Kuznietsov, J. Stuckler, and B. Leibe*, Semi-supervised deep learning for monocular depth map prediction, CVPR, 2017.
[61]. *Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin*, Single view stereo matching, CVPR, 2018.
[62]. *Anurag Ranjan, Varun Jampani, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black*, Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation, CVPR, 2019.
[63]. *Tuo Feng and Dongbing Gu*, SGANVO: Unsupervised deep visual odometry and depth estimation with stacked generative adversarial networks, IEEE Robotics and Automation Letters. 2019.
[64]. *R. Garg, V. Kumar BG, and I. Reid*, Unsupervised CNN for single view depth estimation: Geometry to the rescue, ECCV, 2016.
[65]. *Reza Mahjourian, Martin Wicke, and Anelia Angelova*, Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints, CVPR, 2018.
[66]. *Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova*, Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos, AAAI, 2019.
[67]. *A. Gordon, H. Li, R. Jonschkowski, and A. Angelova*, Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras, arXiv preprint arXiv:1904.04998, 2019.

[68]. *O. Ronneberger, P. Fischer, and T. Brox*, U-net: Convolutional networks for biomedical image segmentation, MICCAI, 2015
[69]. *Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T., Geiger, A*, Sparsity invariant CNNs, arXiv preprint arXiv:1708.06500, 2017
[70]. *N. Silberman, D. Hoiem, P. Kohli, and R. Fergus*, Indoor segmentation and support inference from rgbd images, ECCV, 2012.
[71]. *Ashutosh Saxena, Min Sun, Andrew Y. Ng*, Make3D: Learning 3D Scene Structure from a Single Still Image, IEEE Transactions of Pattern Analysis and Machine Intelligence (PAMI), 2009.
[72]. *S. Song, S. P. Lichtenberg, and J. Xiao*, Sun RGB-D: A RGB-D scene understanding benchmark suite, CVPR, 2015.
[73]. *Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou*, Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios, CVPR, 2019.
[74]. *D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling*, High-resolution stereo datasets with subpixel-accurate ground truth, German Conference on Pattern Recognition (GCPR), 2014.
[75]. *Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, Henrik Aanaes*, Large Scale Multi-view Stereopsis Evaluation, CVPR, 2014.
[76]. *Knapitsch, A., Park, J., Zhou, Q., Koltun, V*, Tanks and temples: benchmarking large-scale scene reconstruction, ACM Trans. Graph. 36(4), 78:1–78:13, 2017.