Federal state autonomous educational institution
of higher education
“National Research University
“Higher School of Economics”

Faculty of Computer science
Major educational program
Applied mathematics and informatics

GRADUATION QUALIFICATION THESIS
titled
Disparity Estimation via Neural Networks

Performed by the student of the group
BAMI155, 4th year:

Pavel Igorevich Bogomolov

Research supervisor:

Associate Professor, A.S. Konushin

Moscow 2019

Contents

1 Abstract **2**

2 Аннотация **2**

3 Introduction **3**

4 Definitions of key terms **4**

4.1 Disparity 4

4.2 Fully convolutional neural networks 4

4.3 Encoder-decoder architectures 4

5 Literature review **6**

5.1 Depth estimation 6

5.2 Disparity estimation 6

6 Methods **7**

6.1 Baseline 7

6.2 Architectural changes 9

6.3 Implementation 9

7 Experiments **10**

7.1 Dataset description 10

7.2 Metrics description 10

7.3 Results 14

7.4 Performance 14

8 Conclusion **14**

References **16**

1 Abstract

Recent works in supervised depth estimation rely on datasets with the ground truth depth measured via sensors alongside the images. However, the quality of such measurements is usually low due to various hardware limitations; they are also expensive to obtain. We can mitigate these shortcomings by switching to a semi-supervised setting. Several methods have been proposed which utilize stereo cameras or video sequences as supervision. Our approach uses the stereo setup and is based heavily on the Monodepth framework by Godard et al. with a substantial change: we replace the network architecture with Double Refinement Network. This allows us to achieve much better results in depth estimation.

2 Аннотация

Современные методы оценки глубины требуют наличия датасета с картинками и точной глубиной, замеренной специальными сенсорами. Однако качество таких измерений, как правило, недостаточно высокое из-за ограничений оборудования, кроме того, это оборудование зачастую дорогостоящее. В данной работе предлагается обойти эти проблемы с помощью перехода к процессу обучения, не требующему данных с глубиной. Существует несколько методов, которые используют стереокамеры или видео для обучения. Рассматриваемый подход опирается на фреймворк Monodepth от Godard et al., с заменой использованной ими архитектуры нейронной сети на Double Refinement Network. Такое существенное изменение позволяет получить значительное улучшение качества на задаче предсказания глубины.

Keywords: Deep Learning, Computer Vision, Artificial Neural Networks Architecture, 3D Reconstruction, Visual Scene Understanding, Depth Estimation

3 Introduction

Depth estimation is the problem of determining the distance from the camera to the points in the world represented by the pixels on an image. It often appears as a part of larger computer vision problems. For instance, in mobile photography, to take portrait shots with blurred background, we need to know which parts of an image belong to the background, i.e. how far each pixel is from the camera. Another example is automatic 2D-to-3D film conversion [12]. Depth is also required in any systems with the need for scene understanding, e.g. self-driving cars or robot vacuum cleaners, to avoid collisions with the environment. The topic of monocular depth estimation, i.e. predicting depth from a single monocular image, is particularly interesting. By making a system require only an RGB camera and no additional hardware, we can significantly reduce the cost of producing it. Unfortunately, depth can be ambiguous in the monocular case, especially without prior knowledge about the camera's focal length. These limitations make monocular depth estimation a very challenging problem; however, human brains learn to solve it, which enables us to believe that a neural network could be capable of producing acceptable results as well.

The problem with neural networks is the need for large amounts of data with accurate ground truth. The appropriate equipment is expensive, while the quality of measurements may vary depending on the scene. For this reason, several unsupervised methods have been proposed, of which the most relevant to our research is training from synchronized stereo pairs, i.e. images, simultaneously taken from two points of view. We can use them to train a network to predict disparity maps and then compute depth from the predicted disparity. One of the most well-known approaches is [5]. We use it as our baseline. The goal of our work is to improve the quality of disparity predictions by changing the model architecture used in [5] to Double Refinement Network (DRN) [2], the vanilla version of which produces high-quality depth predictions at a high framerate. We modify the last layers of DRN and show that the resulting setup produces better results in disparity estimation, compared to the baseline.

Firstly, we review the existing work on monocular depth and disparity estimation. Secondly, we describe the existing framework and our proposed changes to it,

followed by the experiments and their results. Finally, we discuss the importance of the model architecture and the possible training in both supervised and unsupervised setups simultaneously.

4 Definitions of key terms

4.1 Disparity

Disparity is a vector representing the shift of a pixel between synchronized stereo cameras, i.e. the difference in the positions of the pixel on the left and right frames. Knowing disparity d, we can compute depth D using the following formula:

$$D = \frac{bf}{d}, \tag{1}$$

where b is the baseline, i.e. the distance between the cameras, and f is the focal length, which should be the same on both cameras in order for the method to work. We assume that b and f are fixed and are given to us as a part of a dataset.

4.2 Fully convolutional neural networks

This type of architecture is especially popular in computer vision. Fully convolutional networks are used to solve problems where input and output are images, typically of the same size (e.g. segmentation, depth estimation, image-to-image translation). Their defining feature is not having fully connected layers, thus learning only local filters, even for decision making.

4.3 Encoder-decoder architectures

In the context of fully convolutional networks, encoder-decoder usually means several downsampling convolutional blocks (the resolution lowers with each block, while the number of channels increases) followed by upsampling convolutional blocks (with the opposite effects). In some architectures, in addition to the output of the preceding part of the network, a decoder block has the encoder features of the same size as input. This technique was popularized by U-Net [8] (see Fig. 4.1).

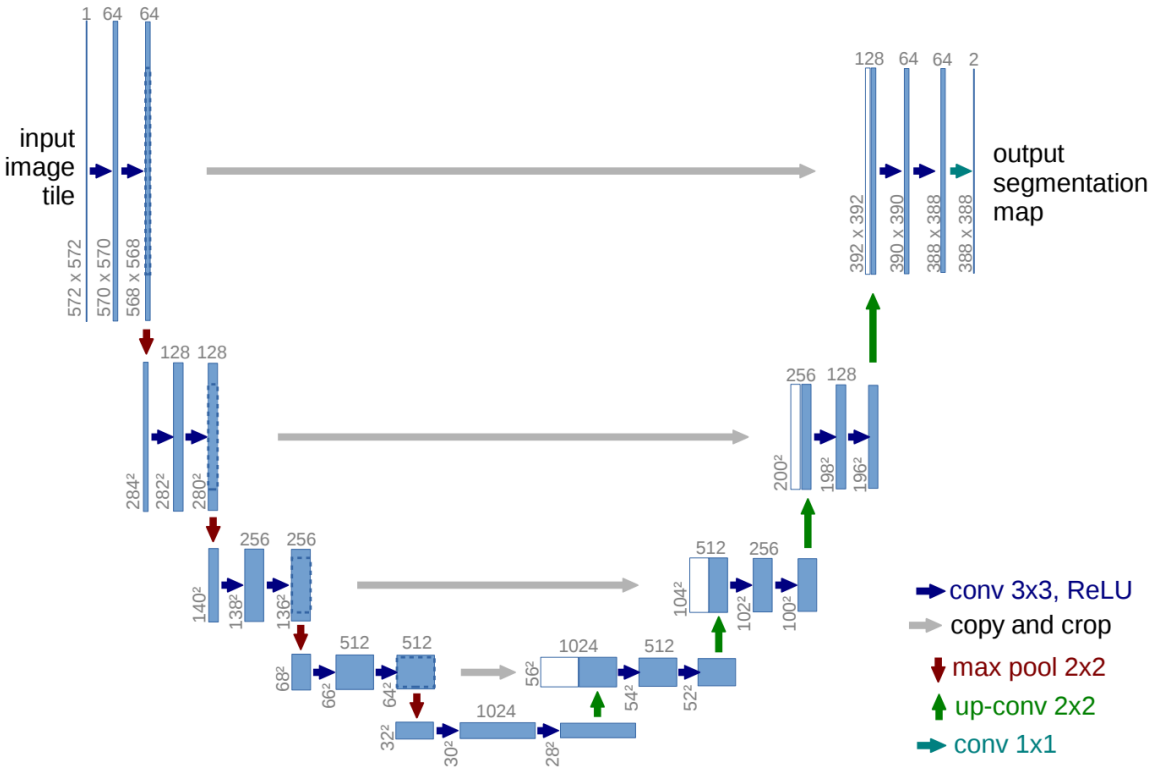

Figure 4.1: An example of a fully convolutional encoder-decoder network: the U-Net architecture (picture from [8]). The boxes represent tensors, and the arrows represent layers.

5 Literature review

5.1 Depth estimation

Recent deep learning methods for supervised monocular depth estimation usually employ some form of an encoder-decoder fully convolutional architecture. [3; 10] Some of the best-performing architectures are RSIDE [7] and DRN [2]. RSIDE uses a ResNet [1] encoder to produce feature maps, which are then upsampled to have the same size and fed through a decoder to obtain depth. DRN reduces the amount of bilinear interpolation by introducing an iterative refinement process with two upsampling branches. This allows the model to produce results of comparable quality with significantly faster runtime.

5.2 Disparity estimation

The previously mentioned depth estimation methods rely on having large datasets with accurate ground-truth depth. This kind of data can be expensive or even impossible to collect in some cases. Godard et al. [5] proposed a method, called Monodepth, capable of producing high-quality results without depth supervision, instead using stereo pairs. They represent disparity estimation as an image reconstruction problem (predict the right frame from the left image and vice versa). To solve it, Godard et al. train a network to predict left-to-right and right-to-left disparity maps, and use a bilinear sampler [9] to reconstruct the images. Having disparity maps for both viewpoints allows them to 1) post-process disparities during test time to achieve better results at image borders and 2) enforce left-to-right consistency to enhance the training process. While there may exist infinite wrong disparities leading to the correct reconstruction, Monodepth mitigates this problem by imposing a smoothness constraint on the produced disparity maps.

Poggi et al. further improve this framework, suggesting to utilize three images instead of two [6]. The idea is to train a network to predict the left and right images by the central frame. Since no trinocular datasets were available at the time of their research, Poggi et al. also propose interleaved training on stereo datasets, taking one frame as the central and the other as either left or right, and applying the loss function to one of the outputs. This enhanced framework shows a considerable increase in the

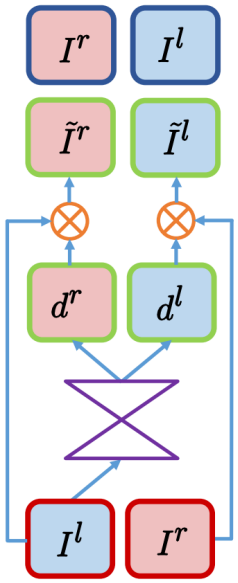

Figure 6.1: The Monodepth framework (picture from [5]). The CNN (represented by two triangles) is applied to the left image I^l to obtain two disparity maps (for the left and right frames). The bilinear sampler [9] (represented by a cross inside a circle) then applies the disparities to the original frames to reconstruct their respective opposite frames.

metrics.

6 Methods

6.1 Baseline

The framework of [5] (Fig. 6.1) consists of several parts. First, the disparity network (VGG or ResNet-50) takes the left frame as input and predicts the left-to-right and right-to-left disparity maps, d^r and d^l, respectively. After that, the bilinear sampler [9] reconstructs each image from the opposite frame and its respective disparity map using the following formulas:

$$\tilde{I}^l_{ij} = \sum_m^W I^r_{im} max(0, 1 - |j + d^l_{ij} - m|), \tag{2}$$

$$\tilde{I}^r_{ij} = \sum_m^W I^l_{im} max(0, 1 - |j + d^r_{ij} - m|), \tag{3}$$

where W is the width of the image, $\tilde{I}^l$ and $\tilde{I}^r$ are the reconstructed left and right

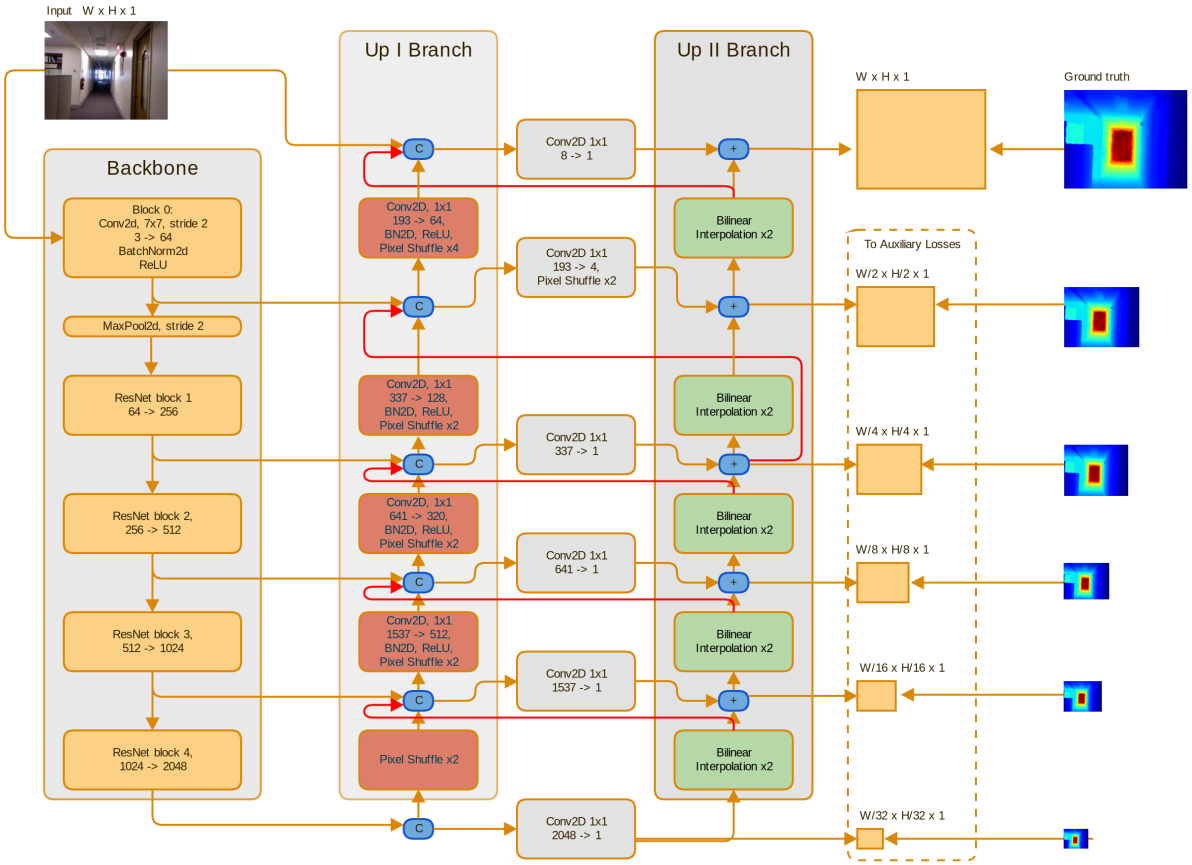

Figure 6.2: The Double Refinement Network architecture (picture from [2]). The encoder part is shown as ”Backbone”, while ”Up I Branch” and ”Up II Branch” represent the decoder. The yellow boxes are the outputs of the network; this is the only part that needs changing so as to predict disparity instead of depth.

frames respectively.

Finally, the following loss functions are applied:

- *reconstruction loss*: penalize the difference between a reconstructed frame and its real counterpart;
- *disparity alignment loss*: make the disparities of a stereo pair align with each other;
- *disparity smoothness loss*: make the disparities smooth where the original image is also smooth.

6.2 Architectural changes

Since the structure is completely modular, we can change each part without breaking the others. In this case, we replace the original disparity network (VGG or ResNet) with the Double Refinement Network (DRN) (Fig. 6.2), independently of the other parts. DRN achieves high metrics in depth benchmarks and can be easily repurposed to predict disparity instead of depth with slight adjustments.

First of all, we need a new activation function, since disparity and depth are very different in scale. While depth is measured in meters, disparity is usually represented in relative shift, i.e. the number of pixels of positional difference divided by the width of the image. We take the activation from [5] and apply it to the outputs of DRN:

$$f(x) = \frac{0.3}{1 + e^{-x}}, \tag{4}$$

which means that the possible values of disparity lie in the interval (0; 0.3).

Unlike vanilla DRN, we need to predict two disparity maps instead of one depth map. We can achieve this by changing the number of channels in the output from 1 to 2.

With these two changes, our network is fully capable of producing disparities for the rest of the framework. As we show in the experiments section, it significantly outperforms the baseline.

6.3 Implementation

The whole baseline framework was re-implemented using the PyTorch deep learning library for Python 3. Compared to TensorFlow, used in [5], PyTorch provides much more flexibility in terms of both development and modification and is becoming the industry standard in deep learning. After re-implementing, the process of changing the underlying neural network architecture became much easier. The final version of the code supports both ResNet-50 from the original paper (and is able to reproduce the results) and DRN.

Figure 7.1: Example of a stereo pair from the KITTI dataset.

7 Experiments

7.1 Dataset description

We perform all of our experiments on the KITTI dataset [11], which contains stereo pairs of images with high-quality ground truth disparities. We use the split of Eigen et al. (22600 train images, 888 validation and 697 test images) to train, evaluate and compare with other models. The images were recorded from a moving platform and depict the streets in Karlsruhe, Germany (see Fig. 7.1). The dataset also provides depth measurements, captured via a Velodyne 3D laser scanner.

7.2 Metrics description

Let d and g be the predicted and the ground truth depth respectively. We compare with the baseline and other methods using the following metrics:

- RMSE:
$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(d_i - g_i)^2}$$
- RMSE log:
$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log d_i - \log g_i)^2}$$
- Abs Rel:
$$\frac{1}{N}\sum_{i=1}^{N}\frac{|d_i - g_i|}{g_i}$$

Table 7.1: Comparison with the previous methods on the KITTI dataset. All numbers are obtained using the split of Eigen et al. [3] with the crop of Garg et al. [4]. Depth predictions are capped at 80 meters. The arrows represent whether higher values are better or worse. The best results are highlighted in bold. "+ pp" means additional post-processing, where the result is combined from the disparity for the original image and the flipped disparity for the flipped image.

	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
	↓	↓	↓	↓	↑	↑	↑
Monodepth [5] (VGG)	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Monodepth [5] (ResNet)	0.133	1.142	5.533	0.230	0.830	0.936	0.970
Monodepth [5] (ResNet) + pp	0.128	1.038	5.355	0.223	0.833	0.939	0.972
3Net [6] (VGG)	0.142	1.207	5.702	0.240	0.809	0.928	0.967
3Net [6] (ResNet)	0.129	0.996	5.281	0.223	0.831	0.939	0.974
3Net [6] (ResNet) + pp	0.126	0.961	5.205	0.220	0.835	0.941	0.974
Ours	0.112	0.898	4.973	0.207	0.865	0.949	0.975
Ours + pp	**0.108**	**0.794**	**4.738**	**0.199**	**0.871**	**0.954**	**0.978**

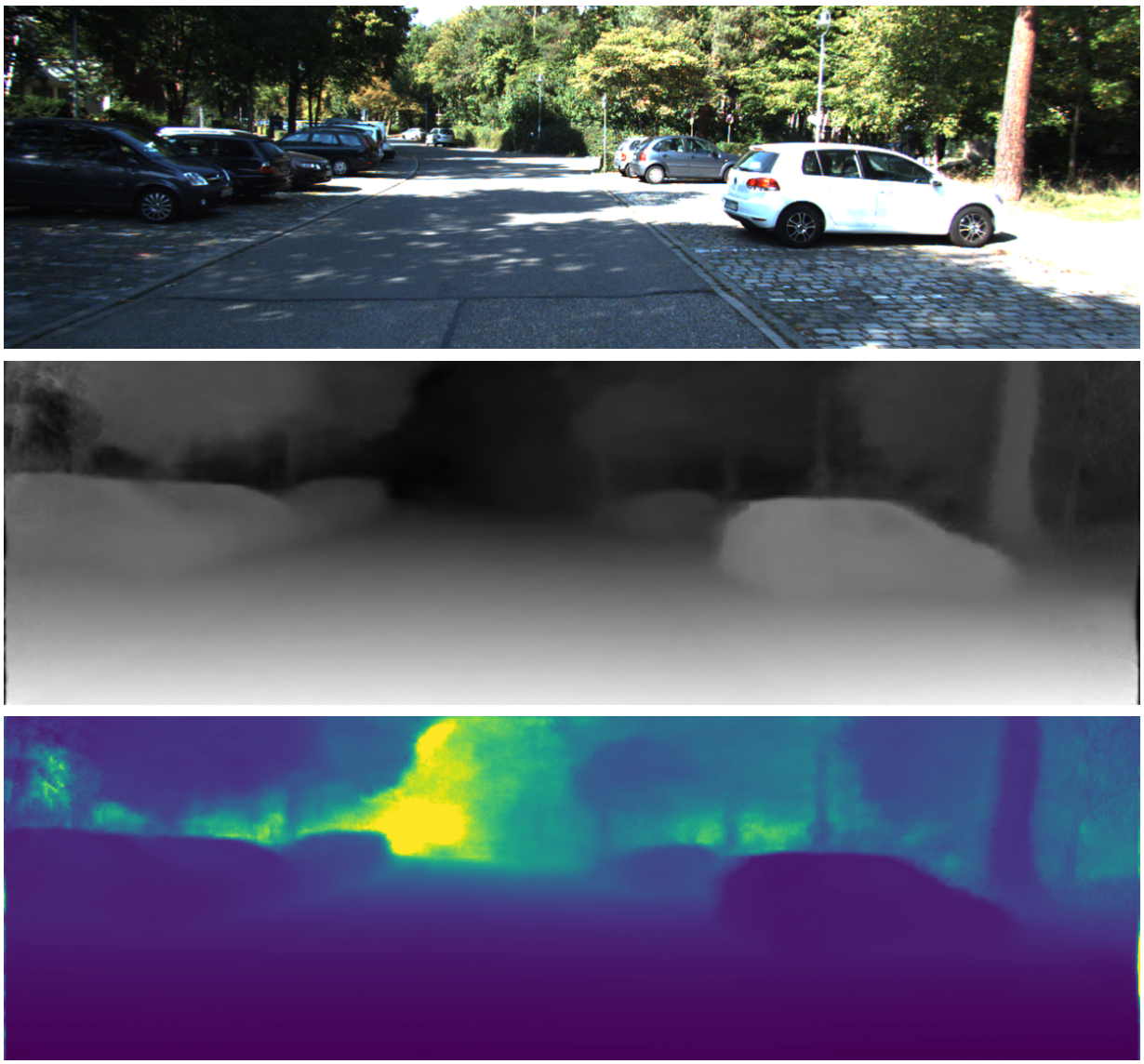

Figure 7.2: An image from the KITTI test set, the disparity predicted by our network and the calculated depth. Disparity values, in pixels: (1.41; 78.44382). Depth values, in meters: (4.9; 80)

- Sq Rel:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{(d_i - g_i)^2}{g_i}$$

- Correct prediction ratios ($\delta < t$):

$$\frac{1}{N}\sum_{i=1}^{N}\left[\max\left(\frac{d_i}{g_i}, \frac{g_i}{d_i}\right) < t\right],$$

where $t \in \{1.25, 1.25^2, 1.25^3\}$

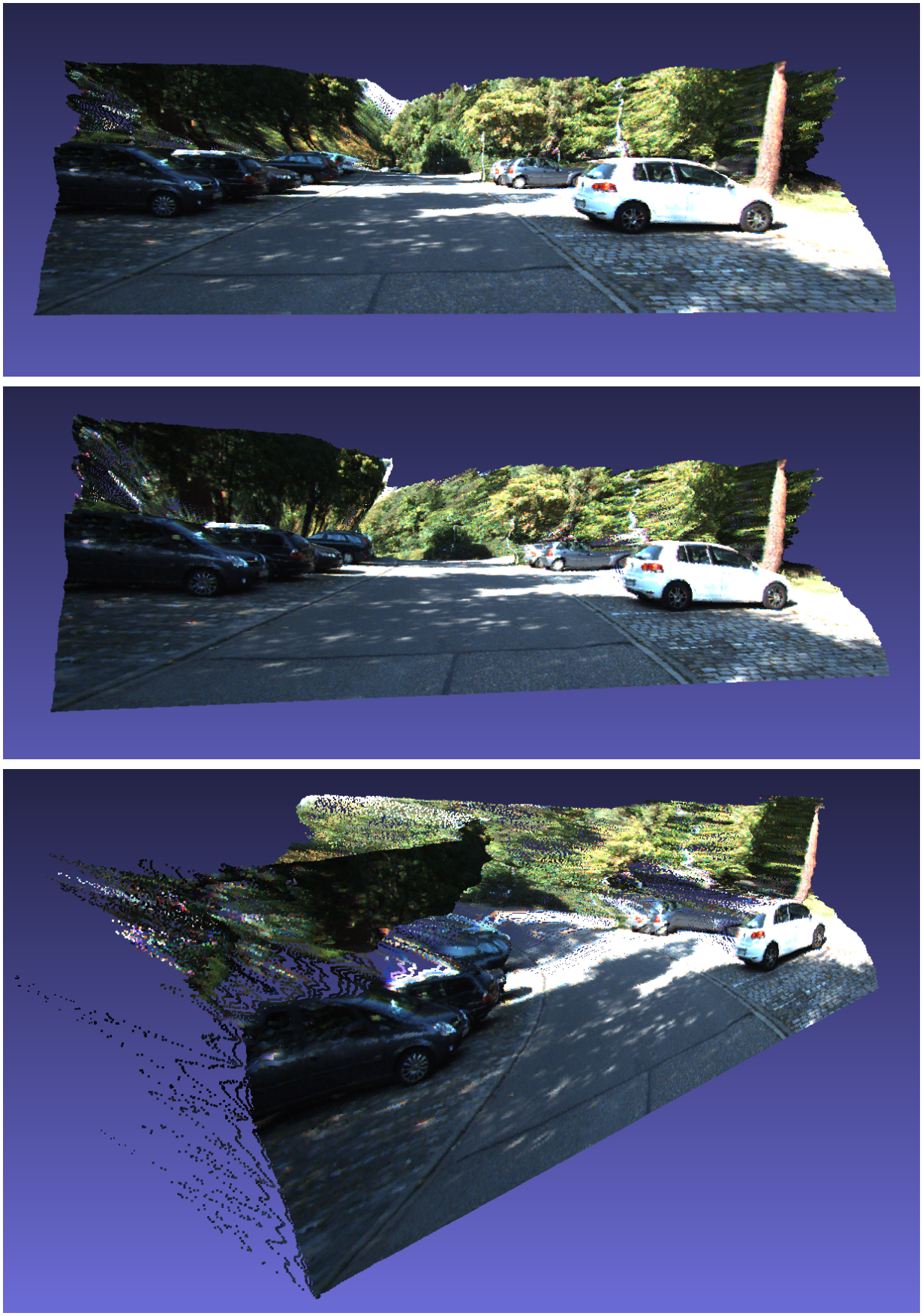

Figure 7.3: The 3D scene reconstruction based on the predicted depth.

7.3 Results

To ensure fairness in the benchmark, we include the results obtained on the KITTI dataset (for training and testing) without any supervision except for stereo. To compute metrics, we use the evaluation tool provided in [5], which converts predictions and ground truth from depth to disparity and calculates the metrics. The results are presented in Table 7.1. An example of our results is shown in Fig. 7.2 (the disparity prediction and the calculated depth) and Fig. 7.3 (the 3D scene reconstruction).

As we can see, our method outperforms even 3Net, despite not utilizing the improved interleaved training process. This shows us that the network architecture is just as important as the training framework, if not more so.

7.4 Performance

Additionally, we measure the FPS rate during inference for both the baseline (ResNet-50) and our method, with the post-processing. We set the real-time constraints, i.e. batch size equals 1.The results are 40 FPS in the case of ResNet-50, and 25 FPS for our method. While the speed reduces, it is still real-time performance.

8 Conclusion

In this work, we developed a disparity estimation method based on the Monodepth framework. We exploited the baseline's modular structure and integrated Double Refinement Network into it, improving the prediction quality on the KITTI dataset by 3.8% (absolute improvement in $\delta < 1.25$). The resulting setup was implemented in PyTorch.

Such an improvement could be noticeable in any computer vision systems where accurate depth estimation is crucial. Using further framework improvements (e.g. the interleaved training described in [6]) could lead to even better results.

Another possible way to improve this method is simultaneous learning of depth and disparity by adding a new loss, which compares the predicted disparity with the expected disparity, calculated from the ground truth depth. While this method does not benefit from requiring only stereo images (thus increasing the cost of obtaining

data), it could potentially produce more accurate predictions on the datasets where depth is available at training time.

References

1. Deep residual learning for image recognition / K. He [et al.] // Proceedings of the IEEE conference on computer vision and pattern recognition. — 2016. — P. 770–778.
2. Double Refinement Network for Efficient Indoor Monocular Depth Estimation / N. Durasov [et al.] // arXiv preprint arXiv:1811.08466. — 2018.
3. *Eigen D.*, *Fergus R.* Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture // Proceedings of the IEEE International Conference on Computer Vision. — 2015. — P. 2650–2658.
4. *Garg R.*, *G V. K. B.*, *Reid I. D.* Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue // CoRR. — 2016. — Vol. abs/1603.04992. — arXiv: 1603.04992. — URL: http://arxiv.org/abs/1603.04992.
5. *Godard C.*, *Mac Aodha O.*, *Brostow G. J.* Unsupervised Monocular Depth Estimation with Left-Right Consistency // CVPR. — 2017.
6. *Poggi M.*, *Tosi F.*, *Mattoccia S.* Learning monocular depth estimation with unsupervised trinocular assumptions // 6th International Conference on 3D Vision (3DV). — 2018.
7. Revisiting Single Image Depth Estimation: Toward Higher Resolution Maps with Accurate Object Boundaries / J. Hu [et al.] //. — 2018.
8. *Ronneberger O.*, *Fischer P.*, *Brox T.* U-net: Convolutional networks for biomedical image segmentation // International Conference on Medical image computing and computer-assisted intervention. — Springer. 2015. — P. 234–241.
9. Spatial Transformer Networks / M. Jaderberg [et al.] // CoRR. — 2015. — Vol. abs/1506.02025. — arXiv: 1506.02025. — URL: http://arxiv.org/abs/1506.02025.

10. *Spek A.*, *Dharmasiri T.*, *Drummond T.* CReaM: Condensed Real-time Models for Depth Prediction using Convolutional Neural Networks // arXiv preprint arXiv:1807.08931. — 2018.

11. Vision meets robotics: The KITTI dataset / A. Geiger [et al.] // The International Journal of Robotics Research. — 2013. — Vol. 32, no. 11. — P. 1231–1237. — DOI: 10.1177/0278364913491297. — eprint: https://doi.org/10.1177/0278364913491297. — URL: https://doi.org/10.1177/0278364913491297.

12. *Xie J.*, *Girshick R. B.*, *Farhadi A.* Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Convolutional Neural Networks // CoRR. — 2016. — Vol. abs/1604.03650. — arXiv: 1604.03650. — URL: http://arxiv.org/abs/1604.03650.