3D Computer Vision Final Project
Deep Depth Estimation with Sparse Depth Map

黃宏鈺B09902128, 鮑鈺文B09902016, and 康家豪R12922035

National Taiwan University

Abstract. Our study focuses on enhancing depth estimation capabilities in self-driving cars within unknown environments, leveraging a novel integration of traditional stereo vision and deep learning methods. Our approach utilizes the depth and disparity information derived from stereo imagery, initially training our model on the Virtual KITTI 2 dataset [1]. This virtual training is a strategic move to overcome the limitations inherent in hand-crafted computer vision methods. We then apply calibration techniques on real-world data from the KITTI dataset [2], aiming to adapt and fine-tune our model for practical application. This hybrid method marks a significant shift from the reliance on extensive pre-collected datasets, improving the adaptability of autonomous vehicles to new environments. Our report elaborates on the methodologies, experiments, and findings, underscoring the effectiveness of combining stereo vision with deep learning for robust depth estimation in autonomous vehicles.

Keywords: Depth estimation · Deep Learning · 3D Computer Vision.

1 Introduction

In this research, we are dedicated to enhancing the depth estimation capabilities of self-driving cars in unknown environments. Our approach centers on the integration of traditional stereo vision with advanced deep learning methods, a synergy aimed at bridging the gap between virtual dataset training and real-world application. By leveraging the depth and disparity information extracted from stereo imagery, our model, initially trained on the Virtual KITTI 2 dataset, is designed to be adaptable and precise in real-world scenarios, as represented by the KITTI dataset.

Historically, the quest for accurate depth estimation in autonomous vehicles has seen a shift from hand-crafted computer vision techniques to sophisticated machine learning approaches. Before the deep learning revolution around 2014, focus was primarily on algorithms like SIFT, SURF, and PHOG, which, despite their lower computational requirements and ease in identifying trends, faced limitations like sparse depth map and lack of real-time processing. The advent of deep learning marked a significant transformation, offering the ability to handle large volumes of data and demonstrating unparalleled accuracy across diverse scenarios.

Our work embraces this evolution, proposing a model that combines the robustness of deep learning with the precision of stereo vision techniques. By training in virtual environments and calibrating with real-world data, our approach aims to overcome the limitations of purely hand-crafted methods. This research expands on the methodologies, experiments, and findings, showcasing the effectiveness of our hybrid approach in achieving accurate, adaptable depth estimation crucial for the deployment of autonomous vehicles in a variety of environments.

2 Related works

In this section, we will divide our discussion into two parts, one focusing on depth estimation using traditional 3D computer vision methods and the other on depth estimation using deep learning methods.

2.1 Traditional Stereo Vision in Depth Estimation

SGM-based method (Semi-Global Matching) [3] for depth estimation is a technique used in computer vision, particularly in stereo vision systems. It involves calculating the disparity between corresponding pixels in stereo image pairs to estimate depth. SGM balances between local pixel-wise matching and global consistency, using a semi-global optimization process that considers several paths through the image. This approach helps in achieving accurate depth estimation, especially in structured environments. SGM is favored for its effectiveness in producing high-quality depth maps and its relatively efficient computational performance compared to global methods.

PatchMatch-based method [4] in depth estimation is a unique algorithm that excels in finding correspondences between patches of two stereo images. Unlike traditional methods that rely on exhaustive searches, PatchMatch uses a randomized approach to quickly approximate the best match for each patch. It then refines these matches over successive iterations, leading to a more accurate depth map. This method is particularly beneficial in handling images with repetitive textures or occlusions, where conventional algorithms might struggle. PatchMatch's efficiency in generating depth maps from stereo images makes it an advantageous tool in applications like depth estiamtion.

2.2 Deep Learning in Depth Estimation

In the realm of deep learning for depth prediction, there are several notable methodologies. Here, we will delve into correlation-based approaches exemplified by DispNet, explore Cost Volume-based techniques such as GWCNet, and discuss the Transformer-based method, STTR, highlighting their unique contributions to depth estimation.

DispNet [5] is a deep learning architecture specifically designed for disparity estimation in stereo vision. It is a convolutional neural network (CNN) that directly learns to compute the disparity map from a pair of stereo images. This end-to-end approach allows DispNet to automatically learn feature representations relevant for disparity estimation, bypassing the need for hand-crafted features or complex preprocessing steps. DispNet's performance in generating accurate disparity maps has made it a significant advancement in the field of depth estimation, particularly useful in applications requiring precise 3D reconstructions or depth-aware processing.

GWCNet, or Group-Wise Correlation Network [6], is a method for stereo matching that focuses on estimating disparities between rectified image pairs, crucial for depth sensing and autonomous driving. It constructs a cost volume using group-wise correlation, dividing the left and right image features into groups along the channel dimension. Each group's correlation maps generate multiple matching cost proposals, packed into a cost volume. This approach provides efficient representations for feature similarities, preserves information better than full correlation, and maintains performance with fewer parameters. The paper demonstrates that GWCNet outperforms previous methods on various datasets, including Scene Flow and KITTI.

STTR (Stereo TRansformer) [7] is an approach in stereo depth estimation that innovates by applying a sequence-to-sequence correspondence model. It replaces traditional cost volume construction with dense pixel matching using positional information and attention mechanisms. STTR offers several benefits: it removes the constraints of a fixed disparity range, identifies occluded regions while providing confidence estimates, and enforces uniqueness in the matching process. This method has shown promising results on both synthetic and real-world datasets and demonstrates strong generalization capabilities across different domains without the need for fine-tuning.

3 Methodology

3.1 Overview

The pipeline of our method can be broken into two parts. The first part is using the traditional method, like stereo vision, LiDAR ...etc, to create a sparse depth estimation. The second part is a deep-learning model, which takes the RGB image as input and predicts the dense depth map. Then we will use the sparse depth estimation from the first part to adjust the depth map from the second part. The overall pipeline is below.

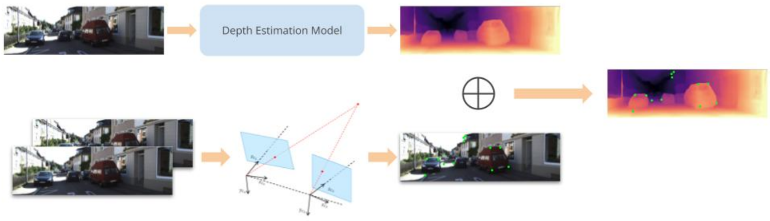

Fig. 1. The overview of our method

3.2 CNN-based Monocular Depth Estimation Model

Model Structure As mentioned in the previous section, there is a monocular depth estimation model to predict the dense depth map. Similar to Fang, Z. et al. [8], we take a CNN-based U-net structure as our backbone for the prediction of dense depth maps. We use the first three blocks and the first layer of the fourth block of VGG19 [9], which will be frozen, as our encoder. The outputs of the first three blocks will be concatenated with the output of the corresponding part of the decoder. The model structure is below.

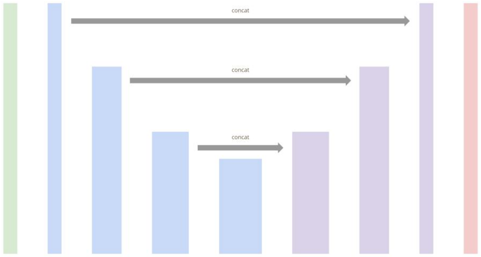

Fig. 2. The structure of our depth estimation model

Loss function In this project, we take monocular depth estimation as the simple regression problem to predict the depth for each pixel. Hence, instead of introduce some loss functions by domain knowledge, we use the mean-square-error as our loss function. What's more, the ground truth of the KITTI dataset is very sparse, so we use a mask to only calculate the loss of these pixel. Because the maximum depth in KITTI is 80 m, we scale ground truth of the prediction between $[0, 1]$ by dividing them with 80 for the better optimization of the deep learning.

Hyper-parameters For our experiment setting, we train four models to better understand the power of our proposed method. They are $KITTI_{Large}$, $KITTI_{Small}$, $VKITTI2$ and $KITTI_{Finetune}$. The first three models are trained from scratch respectively using the larger set of KITTI, the smaller set of KITTI,and Virtual KITTI2. The $KITTI_{Finetune}$ is fine-tuned from the $VKITTI2$ use the smaller set of KITTI.

The optimizer we used is $Adam$, the learning rate is $1e-4$ for trained from scratch, and $2e-5$ for fine-tuning. The training step is between 33320 and 48900 for trained from scratch, and 2140 for fine-tuning.

Training results The following is our result on testing set. We use the MSE as our metric and the ground truth is scaled to $[0, 1]$ as mentioned before.

MSE	$KITTI_{Large}$	$KITTI_{Small}$	$VKITTI2$	$KTIIT_{Finetune}$
KITTI Test	0.01880	0.02010	0.04707	0.02088
Virtual KITTI 2 Test	-	-	0.0418	0.07506

Table 1. Training results

Visual results The following is the visual result of the prediction of the model, $VKITTI2$, on Virutal KITTI2. We can see that our model is able to predict the depth, though it is blur at some place.

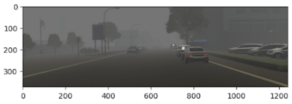

Fig. 3. The RGB input

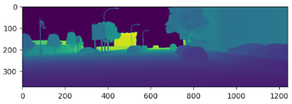

Fig. 4. The ground truth

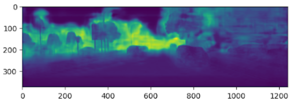

Fig. 5. The prediction

The following is the visual result of the prediction of the each model on KITTI. Since the ground truth of KITTI is more sparse than Virtual KITTI 2, we can see that the prediction of the $KITTI_{Large}$ and $KITTI_{Small}$ both focus on the prediction of the part that is nearer to the camera. On the other hand, $VKITTI2$ is able to predict with more texture. However, the performance is poor the due to the domain gap. And $KITTI_{Finetune}$ has both advantage from these two dataset, one is better performance on KITTI testing set, the other one is the prediction with more detail texture.

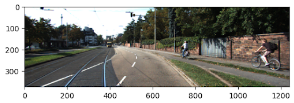

Fig. 6. The RGB input

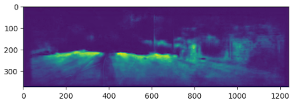

Fig. 7. The prediction of $KITTI_{Large}$

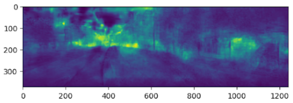

Fig. 8. The prediction of $VKITTI2$

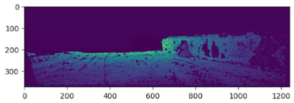

Fig. 9. The ground truth

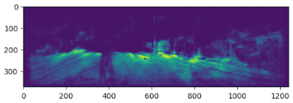

Fig. 10. The prediction of $KITTI_{Small}$

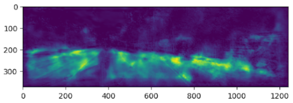

Fig. 11. The prediction of $KITTI_{finetune}$

3.3 Depth Adjustment

Scale Factor For the ground truth depth map d and our prediction d' such that $d, d' \in \mathbb{R}^{H \times W}$ where (H, W) is the size of an image, we assume that predictions can reconstruct the relative depth of scene appropriately *i.e.* $d = \alpha d'$. We try to find the scale function $S(\cdot) : \mathbb{R}^{H \times W} \times \mathbb{R}$:

$$\begin{cases} s = S(d) \\ s' = S(d') \end{cases} \quad \frac{s}{s'} = \alpha$$

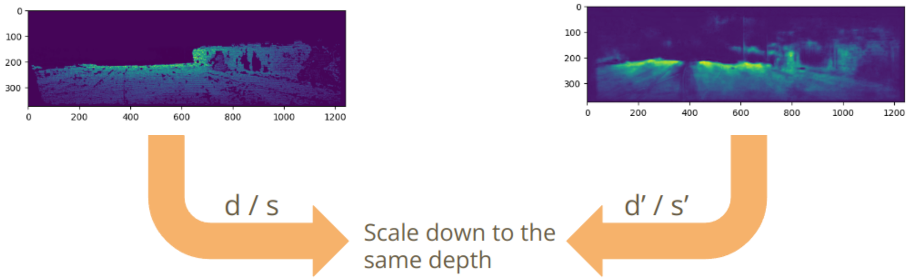

Fig. 12. Scale factor pipeline

Point cloud and ICP The scale-only approach relies on the strong assumption that our prediction can reconstruct the scene with high quality. However, inevitably there will be some disturbances; thus, we modify the ideal equation $d = \alpha d'$. First, we construct the point cloud from the depth map by camera intrinsic matrix. Subsequently, we hope that after taking the slight rotation and translation can calibrate the disturbances *i.e.* $p = \alpha[R \mid t]p'$ where p and p' are point clouds of s and s'. Finally, remap the point cloud back to the image space. We can modify the equation $p = \alpha[R \mid t]p'$ as following:

$$\frac{p}{s} = [R \mid t]\frac{p'}{s'}$$

We obtain the $[R \mid t]$ by solving the 3D geometric alignment with ICP algorithm [10].

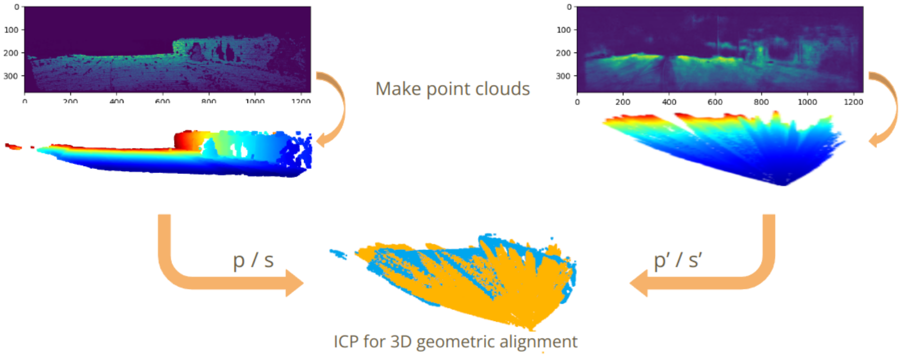

Fig. 13. point cloud with ICP pipeline

4 Experiments

4.1 Setting

We use the KITTI Test dataset to test our depth adjustment methods. For each image, 80% of valid pixels are used to determine the adjustment parameters like s, s' and $[R \mid t]$, and the rest 20% of valid pixels are used to evaluate the methods by calculating the mean square error between them and predict.

4.2 Scale function

Remind the definition of scale function, we are trying to find the appropriate scale factor representing the depth map:

$$s = S(d) \text{ where } d \in \mathbb{R}^{H \times W} \ \ i.e. \ d = \{d_{11}, ..., d_{ij}, ..., d_{HW}\}$$

following are our approaches.

Average The average is a simple and widely understood statistical measure. It's easy to calculate and interpret. Moreover, it provides a measure of central tendency, which represents a central or typical value in a set of numbers. We calculate the average as follows:

$$s = S(d) = \frac{1}{HW} \sum_{j=1}^{W} \sum_{i=1}^{H} d_{ij}$$

Average Between Mid-quartile Range (MQR) The mid-quartile range focuses on the middle 50% of the data, reducing the impact of outliers and extreme values that can skew the mean, which makes it more robust in the presence of non-normal data distributions, as it is less influenced by the shape of the distribution. We calculate the average between mid-quartile range as follows:

$$d_{sort} = \{d_1, ..., d_{H \times W} \mid d_i <= d_j \ \forall \ i < j\}$$

$$s = S(d) = S(d_{sort}) = \sum_{i=L_{q_1}}^{L_{q3}} d_i \quad \text{where} \begin{cases} L_{q_1} = \lfloor H \times W \times 0.25 \rfloor \\ L_{q_3} = \lfloor H \times W \times 0.75 \rfloor \end{cases}$$

The Third Quartile (Q3) As we can observe from Fig. 14. and Fig. 15., there is a large difference between the two distributions of point clouds. Since the deep learning model generates smooth depth maps all the time, the point cloud forms a conical distribution, with a higher density of points near the apex, while ground truth doesn't. Thus the average or the average between mid-quartile range might give a misleading impression of the data's central tendency. To bridge the difference between the two distributions, we also use the third quartile as the scale factor in our experiments. We calculate the third quartile as follows:

$$d_{sort} = \{d_1, ..., d_{H \times W} \mid d_i <= d_j \ \forall \ i < j\}$$

$$s = S(d) = S(d_{sort}) = d_{L_{q3}} \quad \text{where } L_{q3} = \lfloor H \times W \times 0.75 \rfloor$$

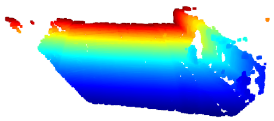

Fig. 14. Point cloud of ground truth.

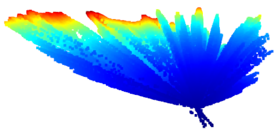

Fig. 15. Point cloud of prediction.

Result In Table 2. we show the results of the above three different strategies. We can observe the huge success of the scale method on the $VKITTI_2$ model. It does solve the domain shift problem better than finetuning, even better than $KITTI$ models. We speculate that the reason why the model trained with VKITTI2 outperforms the model trained with VKITTI after adjustment is that the former has a better understanding of the concept of relative depth in space. This can be attributed to the denser and more continuous data distribution in the VKITTI2 dataset. Additionally, we found that the model trained only with VKITTI2 showed a much greater degree of improvement after adjustment compared to the model exposed to the KITTI dataset. We speculate that this may be related to our incomplete training. Under limited time and computational resources, training with the KITTI dataset is more challenging, making it harder for the model to understand the relationships of relative depth in space. This also highlights the importance of synthetic data.

	$KITTI_{large}$	$KITTI_{small}$	$VKITTI2$	$VKITTI_{finetune}$
W/o Adjustment	0.112	0.115	0.093	0.119
The Average	0.083	0.117	0.027	0.099
MQR	0.083	0.118	0.027	0.100
Q3	0.077	0.099	0.030	0.087

Table 2. Result from scale only method.

4.3 ICP algorithm

In Table 3. we show the results of 'with' v.s 'without' ICP algorithm over two different scale strategies. Except for the $VKITTI_2$ model, all the models exposed to the KITTI dataset acquire improvement, especially for the Q3 scale strategy. It proves our assumption about slight translation and rotation is correct and our approach works. In the Q3 w/ ICP setting, perhaps the differences are not significant for test points, but we can still observe visible differences in Fig. 16.. For the model trained only on VKITTI2, the overall depth appears continuous, especially for distant roads and the sky, which are types of points that do not appear in the KITTI dataset annotations. This once again demonstrates the importance of virtual data.

	$KITTI_{large}$	$KITTI_{small}$	$VKITTI2$	$VKITTI_{finetune}$
MQR w/o ICP	0.083	0.118	0.027	0.100
MQR w/ ICP	0.058	0.087	0.028	0.086
Q3 w/o ICP	0.077	0.099	0.030	0.087
Q3 w/ ICP	0.034	0.033	0.034	0.040

Table 3. Result from scale with ICP method.

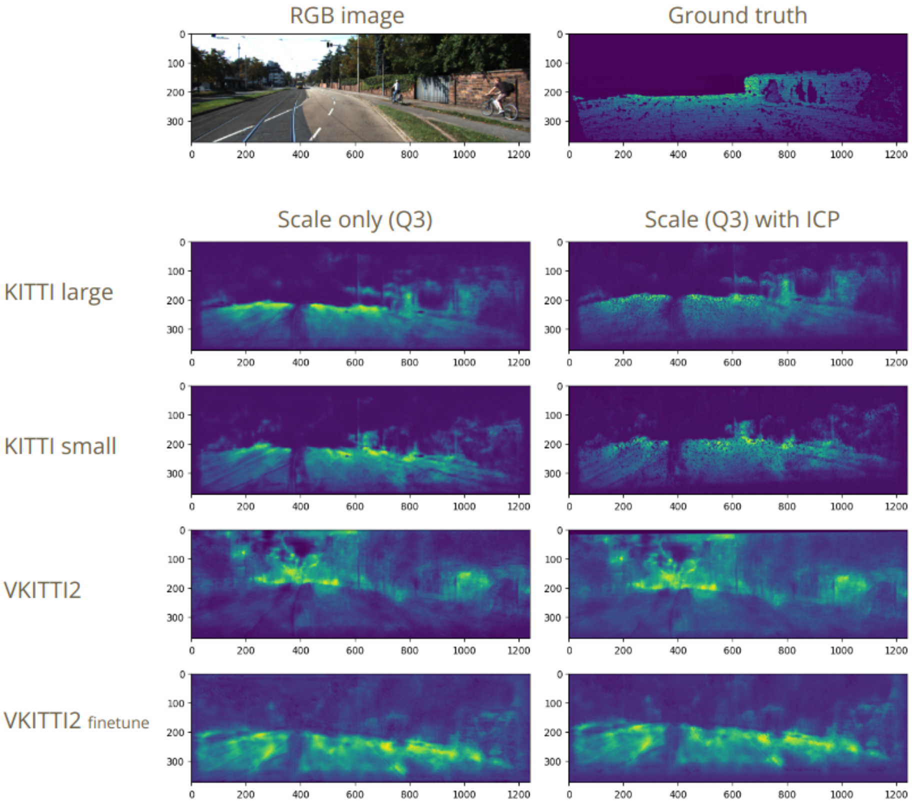

Fig. 16. Results comparison over models and ICP algorithm

5 Discussion

In this final project, we compared the differences between synthetic and real datasets. By applying a simple scaling method, we adjusted the predictions of the model trained only on VKITTI2, successfully surpassing the models that had been exposed to real datasets. Besides scaling, we also used the ICP algorithm to enhance the performance of each model. A point worth discussing is that the architecture of our adopted model, as well as the training resources and time, were far less than those of state-of-the-art (SOTA) specifications. This means that our experimental results are still at a very preliminary stage. However, we believe that merely validating the feasibility of the experimental direction is a good achievement within limited resources. As for the potential issue of incomplete training with real datasets, it is indeed unavoidable, but this also indirectly highlights one of the advantages of synthetic datasets: they are easier to train with.

References

1. Geiger, Andreas, et al. ”Vision meets robotics: The kitti dataset.” The International Journal of Robotics Research 32.11 (2013): 1231-1237.
2. Cabon, Yohann, Naila Murray, and Martin Humenberger. ”Virtual kitti 2.” arXiv preprint arXiv:2001.10773 (2020).
3. Hirschmuller, Heiko. ”Stereo processing by semiglobal matching and mutual information”. IEEE Transactions on Pattern Analysis and Machine Intelligence (2007).
4. Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. ”PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing.” ACM Transactions on Graphics, August 2009.
5. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., and Brox, T. ”A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation.” Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
6. Guo, X., Yang, K., Yang, W., Wang, X., and Li, H. (2019). ”Group-wise Correlation Stereo Network.” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3273–3282).
7. Li, Z., Liu, X., Drenkow, N., Ding, A., Creighton, F. X., Taylor, R. H., and Unberath, M. (2021). ”Revisiting Stereo Depth Estimation From a Sequence-to-Sequence Perspective With Transformers.” Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 6197-6206). October 2021.
8. Fang, Z., Chen, X., Chen, Y., & Gool, L. V. (2020). Towards good practice for CNN-based monocular depth estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 1091-1100).
9. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
10. S. Rusinkiewicz and M. Levoy, ”Efficient variants of the ICP algorithm,” Proceedings Third International Conference on 3-D Digital Imaging and Modeling, Quebec City, QC, Canada, 2001, pp. 145-152, doi: 10.1109/IM.2001.924423.

6 Appendix

6.1 Division of work table

name	work
康家豪	Data processing, report
鮑鈺文	Deep learning model, report
黃宏鈺	Depth adjustment, report

Table 4. Division of work table