From Monocular Images to Depth Information: Generating Depth Maps, Camera Calibration and Point Clouds

Faizah Binte Naquib
University of Alberta
Alberta
Email: faizahbi@ualberta.ca

***Abstract*—Combining different sensor information to obtain depth perception for autonomous driving is a common concept that is in use. Despite the abundance of sensors and camera integration, people are looking to reduce the number of devices being used, but maximize the information obtained. This is where the concept of getting both depths maps and point clouds from monocular images comes in. Attaining all information from a single image will substantially reduce the number of sensors required. In this paper, we dig into the possibility of obtaining point cloud information by exploiting preexisting resources, such as generating depth maps from single RGB images and generating point clouds from 2D depth images. In the end, we reach a conclusion on if this is a feasible approach or not.**

Key Words: Depth Map, Point Cloud, Depth Estimation, Autonomous Driving

1. Introduction

Depth Estimation (DE) is a common yet unsolved problem in the field of Computer Vision. It is the process of calculating the distance between a camera and an object, and for the case of point clouds, it is the process of accurately recording the external surface of an object or scene. This ability to grasp 3D information has great potential to be used in robot locomotion, robot surgery, auto-driving, etc.

Currently, many sensor elements are being attached to autonomous driving cars to achieve the functionality of collecting and building environmental information in an attempt to replicate the characteristics of the human eye. 3D information such as depth maps and point clouds obtained from these sensors can be combined to attain high-precision point cloud information [8], which helps to distinguish objects around vehicles more precisely. This ability to distinguish enables better identification and avoidance of obstacles while driving. But with Tesla's new announcement denying integration of multiple sensors to their autonomous cars, the research direction is slowly diverting from sensor fusion to obtaining maximum information from monocular images.

Generating point clouds from monocular images is a task that has been explored in terms of 3D reconstruction. The reconstructed data are mostly 3D model objects such as chairs, airplanes, benches, cabinets, etc., and rarely environmental scenarios like traffic or street images [7] [3]. Most recent papers have concentrated on improving the quality of depth maps obtained from self-supervised models when estimating depth information from monocular street images. The rationale behind using self-supervised models was that they show great potential in terms of perceiving depth and they have low annotation cost [9]. So, work relating to 3D reconstruction with point clouds from traffic/street monocular images is still unexplored.

This paper contains one of the approaches attempted to achieve point cloud information from monocular images by exploiting the two well-known concepts mentioned above. The idea was to construct a framework that would scale the practice of 3D reconstruction on a single image of smaller objects to larger environments. To achieve this, the pre-trained model EPCDepth [9] was modified to successfully extract both Depth Maps and camera calibration data from a single image. Following the general concept of attaining point clouds from depth maps, the obtained 2D coordinate depth image can be converted to a 3D coordinate system, resulting in point cloud information for that particular image. The contribution of this work is the following:

- Modifying a model pre-trained on depth maps to render depth and camera matrix.
- Fusing the depth map and its respective camera calibration data to generate point clouds for monocular images in the KITTI [11] dataset.
- Determining evaluation approaches, as there is limited preexisting work.

2. Literature Review

Depth estimation from a monocular image refers to estimating pixel-wise depth from a single RGB image. The depth estimation can be represented in the form of depth maps or point clouds. This section explores existing work for both of these concepts.

2.1. Camera Matrix

The camera matrix or intrinsic matrix is a matrix that transforms 3D camera coordinates to 2D homogenous image coordinates. According to Hartley et al [10] the intrinsic matrix is parameterized as:

$$\begin{pmatrix} fx & s & x \\ 0 & fy & y \\ 0 & 0 & 1 \end{pmatrix}$$

Here,
fx and fy are the focal lengths, and x and y indicates the principal point offsets.

2.2. Monocular image to Point Clouds

There are many end-to-end networks that generate a dense point cloud from a single image; among them, Lu et al [7] suggested a two-stage training network that would convert RGB images to sparse cloud points at first and in the second stage, generate a dense point cloud from the sparse cloud. Another method suggested by Zeng et al [3] proposes first determining an RGB image's depth map before producing the same image's point cloud. Their proposed approach is also a two-step process, the first step of the process determines the depth map of a 2D image using an encoder-decoder architecture, and the second step generates a point cloud from the depth map with a 3D CNN. Both of these methods have opted towards a multi-stage architecture, as predicting point coordinates is a large-scale regression task that complicates the task of training networks. Keeping the disadvantages of training point clouds in an end-to-end model in mind a different approach was attempted in this model.

2.3. Monocular image to Depth Maps

With the introduction of Deep Learning, there have been more advancements in the field of Depth map generation or Depth Estimation (DE). A survey by [2] discusses existing methods for Monocular Depth Estimation (MDE) and a comparative evaluation between MDE and stereo-based DE is made. The study by [2] suggests that stereo-based DE are prone to more errors as a mapping relation between left and right images must be built by stereo matching while monocular images require no such thing. Therefore, looking into the MDE section of the review, the paper has generalized that all MDE models have two main sections: **depth network** and **pose network**. The depth network generates the depth map , whereas the pose network calculates the egomotion that is, the movement (translation, rotation, etc.) of the camera being used. The input shapes of the data that are used in these networks can be divided into three categories: Mono Sequence (single image input for Unsupervised models), Stereo Sequence (mapping relationship between the left and right images as input for both Supervised and Unsupervised models), Sequence to Sequence (sequence of images are taken as input for RNN). It was stated in the review that among the three categories, Mono sequence is a more novel concept as only one image is being used as an input. Similarly, in terms of training data, all existing processes can be categorized into three groups, **Unsupervised Learning (UL)**, **Supervised Learning (SL)**, and **Semi-supervised Learning**. The reviewers draw a conclusion near the end that while supervised learning provides more accuracy, the dataset for it is less accessible, which makes unsupervised learning the preferred method. Another observation was made that DL models that use graph convolutional, 3D convolution, and 3D geometry constraint outperforms other DL models. Taking the derived conclusions from the review into consideration, the reviewed papers from this point forward contain a mono-sequence training set, trained in an unsupervised manner. Zhou et al [1] tried to introduce a method that mimics how humans perceive depth by feeding a series of images to the model to get camera motion and scene structure, following the same intuition of a human's depth understanding. The method introduced simultaneously calculated depth map for one input image using DispNet architecture and camera pose for three consecutive images using Pose-Deep neural network. The proposed approach showed remarkable accuracy for depth estimation using unlabeled videos as input. But they also claim that they are nowhere near solving the general problem of unsupervised learning for 3D scene structure. Gorard et al [4] suggest that even though the monocular video is a good alternative to stereo-based supervision, it still requires the estimation of egomotion between temporal image pairs during training. And for stereo images, if the camera pose estimation is used as a one-time calibration, it can cause occlusion and texture copy errors. To resolve these issues, and to introduce a system with improved MDE performance when trained on monocular video, stereo data, or both, three changes were suggested- a matching loss that addresses the issues caused by pixels that are visible in the target image but not visible in source images, second a simple auto masking method is introduced that filters out pixels that do not change appearance from frame to frame- i.e. ignoring objects in camera velocity that may cause holes in the depth, and lastly it suggests upsampling lower-resolution depth map to compute error at higher resolution and reconstructing high-resolution input target image as accurately as possible, Shu et al [5] introduce Feature Net which calculates feature matrix loss where the feature representation is additionally learned by two extra regularizers to ensure the convergence of depth and pose. The feature matrix loss was used in place of the commonly used photometric loss. The cause behind replacing photometric loss is that from an optimization perspective, if there are no textures in the images, then the gradient calculated with respect to depth and egomotion will tend to zero, making it a zero gradient. Masoumin et al [6] proposed that a Graph Convolutional Network (GCN) can handle non-Euclidean data in irregular images for self-supervised depth estimation. The model they introduced consists of two parallel encoder-decoder, DepthNet and PoseNet, the first extracts features of input images using ResNet-50, and the second calculates egomotion between two consecutive frames using ResNet-18. The proposed method reduced 40% of the number of parameters used in the network with respect to other state-of-the-art solutions.

3. Method

3.1. Data

The data to train the model was taken from KITTI [11] dataset. The Calibration and raw image files of the date ”2011_09_26” were picked.

3.2. Model

The model that has been used in this proposed method is EPCDepth [9]. EPCDepth is a self-supervised model that has been trained on rectified stereo images. The model claims to outperform other state-of-the-art depth estimation methods due to an added self-distillation layer to their encoder-decoder framework. No changes were made to the depth map generation segment of the architecture. A new Neural Network layer is introduced to the model by connecting it to the encoder and is trained on a subsample of the same set of the rectified stereo image data. Since the EPCDepth is a pre-trained model, instead of training the entire model from scratch, the encoder and decoder segment of the model is frozen while training for the camera calibration. Therefore the neural network model is trained on the image features from the common encoder.

The introduced Neural Network, which will be termed as **”Intrinsic Decoder”**, is a simple neural network model comprising one convolutional layer, one batch normalization, and one max pool layer. The features obtained from the max pool layer are flattened and filtered into the three linear layers. Finally, a nine-digit matrix is obtained, which is the predicted intrinsic matrix.

A visual representation of the entire architecture is given in 1. The changes that have been brought to the original model are as follows:

- The intrinsic matrices are also being input alongside their images.
- The encoder is now acting as a common encoder that feeds features into two different networks: the Depth Decoder and the Intrinsic Decoder.
- Lastly, the depth decoder outputs the disparity of the images, while the intrinsic decoder outputs its camera calibration data.

3.3. Point Cloud Calculation

A point cloud 3D image will be generated by using the depth map and its respective camera matrix both obtained from the same model. For every pixel in the image, the following formula will be applied.

$z = depth(i, j)$
$x = ((j - cx)) * z/fx$
$y = ((i - cy)) * z/fy$
Here,
i, j = old pixel values
x and y and z = new pixel values
fx, fy = focal lengths
cx and cy = optical centers
This formula transforms the depth pixel in the disparity from 2D coordinate system to 3D.

A sample input and output obtained from the proposed pipeline, is shown in Fig 2.

4. Evaluation

The Evaluation method has been broken down into two main sections. The first part evaluates the intrinsic matrices generated from the model by calculating its Mean Squared Loss (MSE) against its given ground truth. This paper will not evaluate the depth maps since they were generated using EPCDepth, and their evaluation detail is already given [9]. The second portion of the evaluation was the tricky part: the evaluation of the constructed point clouds. The problem came down to evaluating the cloud points with the existing ground truth in the KITTI dataset due to its format of LiDar data, which did not coincide with the cloud points being calculated from the depth images. So instead, we opted to determine which evaluation metrics among the two main matrices Chamfer's Distance (CD) and Earth Mover's Distance(EMD), is the best choice for this problem scenario. Considering both of these evaluation metrics were used in previous work [7] [3] but for smaller objects. Moving forward, it would be best to determine which metric best suits street data.

5. Results

All experiments in this section were performed on ten randomly picked images from the KITTI dataset, depth completion section. The dataset contains, a camera image of the scenario, a LiDar raw image, and its intrinsic matrix.

5.1. Evaluation of the obtained intrinsic matrix with MSE Loss:

This section of the experiment calculates the MSE loss between the intrinsic matrices obtained from the proposed method and the ground truth to estimate how far the predicted values are from the truth. The results are portrayed in Table 1

TABLE 1. EVALUATION RESULTS FOR INTRINSIC MATRICES

Image	MSE	Image	MSE
Image 0	9.2021	Image 5	17.87
Image 1	19.800	Image 6	**7.72**
Image 2	14.87	Image 7	10.96
Image 3	15.58	Image 8	13.36
Image 4	8.11	Image 9	10.17

5.2. Evaluation of the point clouds with EMD and CD:

Following the point cloud generation algorithm, the depth map and camera matrix from the model are considered

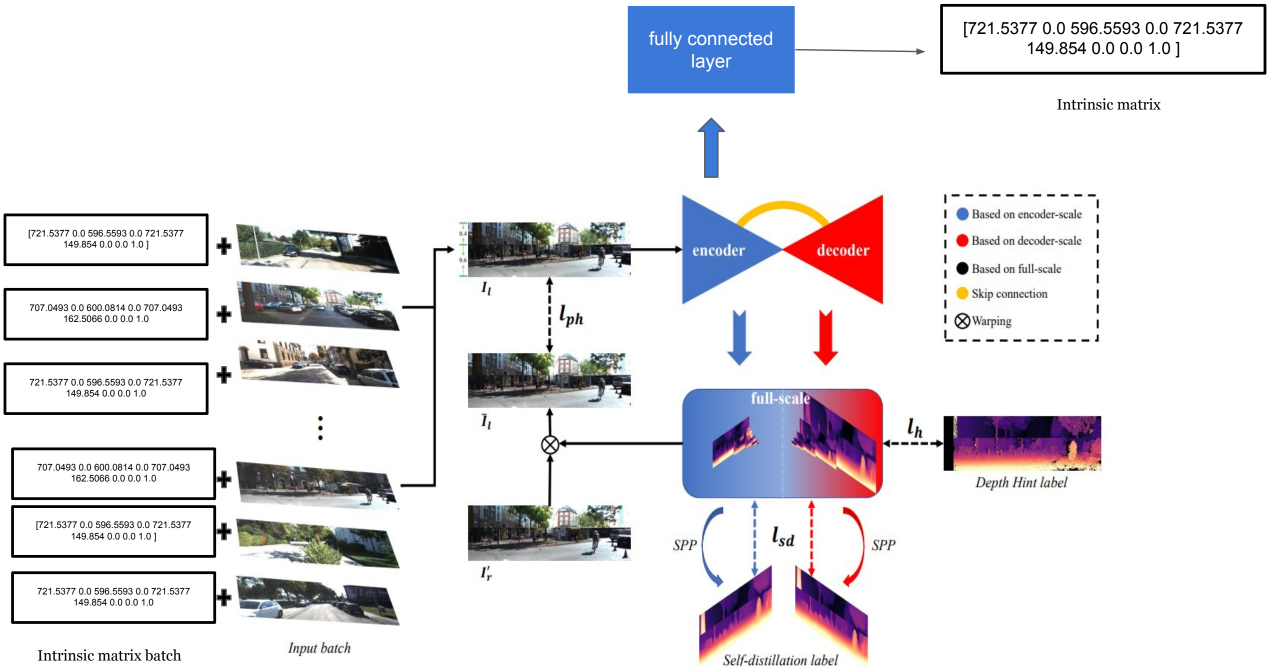

Figure 1. Architecture of the model proposed by [9] along with its proposed modifications

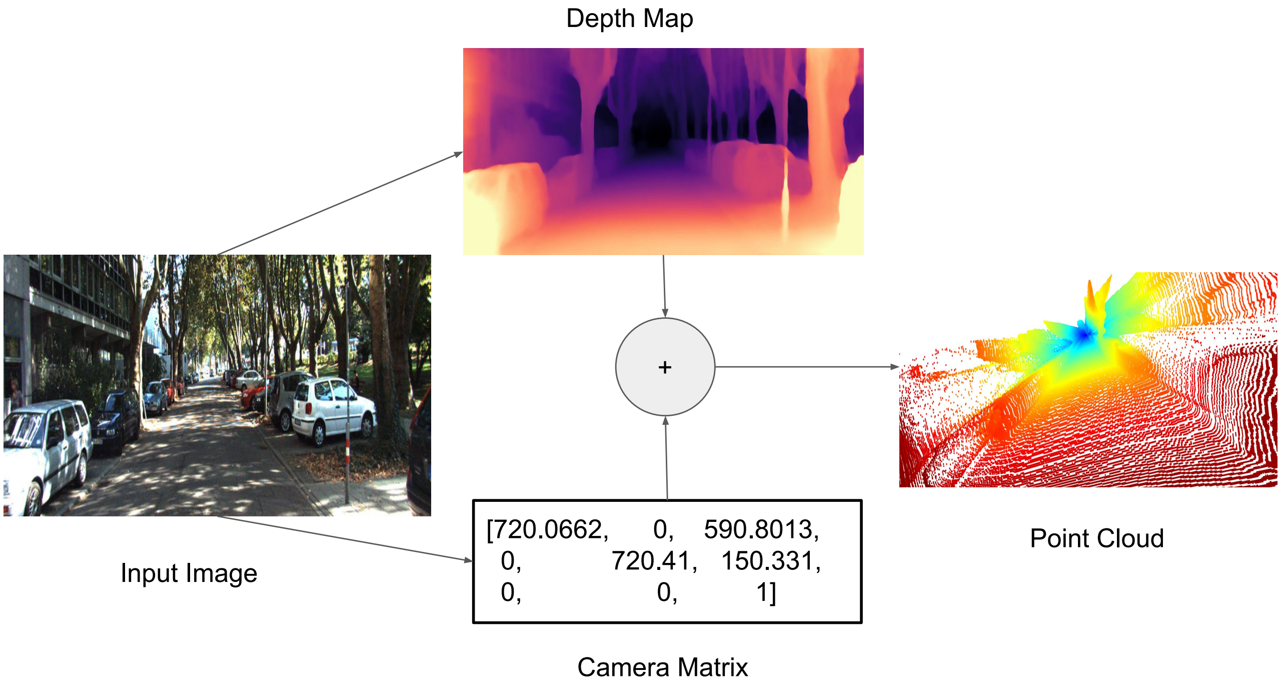

Figure 2. Flowchart representing how the point clouds of an image are obtained.

as the source, while the point clouds generated from the raw LiDar image and its intrinsic matrix from the dataset are considered the target. The CD and EMD of the points inside the images are calculated. The results are shown in Table 2

TABLE 2. EVALUATION RESULTS FOR POINT CLOUDS

Image	EMD	CD	Image	EMD	CD
Image 0	0.3893	113235.78	Image 5	0.4182	89871.71
Image 1	1.79	2149330.41	Image 6	0.25	2149330.41
Image 2	2.56	3711498.46	Image 7	3.2	5417185.55
Image 3	9.48	23853347.54	Image 8	1.54	3718498.46
Image 4	**0.25**	**17205.67**	Image 9	0.4187	123455.76

6. Discussion

The entire work was an experimental approach to observe if the resources that are currently available can be combined to construct point clouds from a single image. But from the experimentations so far, it is can be said that, even though training an end-to-end model for generating points in an image is computationally costly, it might be the better approach. Some of the findings from the experiments are explained below to justify the aforementioned claim further.

- While theoretically, it is possible to generate point clouds from 2D depth data, the points obtained in this way are very different from the actual LiDar obtained data. Which raises the problem of comparison and evaluation.
- Another assumption was this pipeline could be used on the ShapeNet dataset as well. The EPCDepth model was trained to determine depth for street images, therefore performs poorly when it comes to detailed depth representation of smaller objects.
- The intrinsic matrices obtained through the model is quite similar to the original values. Training with more data and fine-tuning the model further can help to improve performance.
- Lastly, from Table 2, we can observe that the values of CD obtained are very high even though, in comparison, the EMD values are practical. This can be because CD concentrates on finer details while EMD emphasizes global matching. Therefore, in our case, EMD is the better metric for evaluation.

7. Conclusion

In conclusion, connecting the separate entities, i.e., depth maps to point clouds, might theoretically make sense, but the camera-obtained point clouds are so drastically different from LiDar point clouds that there is no basis for comparison. Therefore, in future approaches, the concept of depth maps can be integrated inside an end-to-end model, specifically trained on getting LiDar point cloud output from monocular images.

References

[1] T. Zhou, M. Brown, N. Snavely and D. G. Lowe, ”Unsupervised Learning of Depth and Ego-Motion from Video,” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6612-6619, doi: 10.1109/CVPR.2017.700.

[2] Masoumian, Armin, Hatem A. Rashwan, Julián Cristiano, M. Salman Asif, and Domenec Puig. ”Monocular Depth Estimation Using Deep Learning: A Review.” Sensors 22, no. 14 (2022): 5353.

[3] Zeng, Wei, Sezer Karaoglu, and Theo Gevers. ”Inferring point clouds from single monocular images by depth intermediation.” arXiv preprint arXiv:1812.01402 (2018).

[4] Godard, C., Mac Aodha, O., Firman, M. and Brostow, G.J., 2019. Digging into self-supervised monocular depth estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 3828-3838).

[5] Shu, C., Yu, K., Duan, Z. and Yang, K., 2020, August. Feature-metric loss for self-supervised learning of depth and egomotion. In European Conference on Computer Vision (pp. 572-588). Springer, Cham.

[6] Masoumian, A., Rashwan, H.A., Abdulwahab, S., Cristiano, J. and Puig, D., 2021. Gcndepth: Self-supervised monocular depth estimation based on graph convolutional network. arXiv preprint arXiv:2112.06782.

[7] Q. Lu, M. Xiao, Y. Lu, X. Yuan and Y. Yu, ”Attention-Based Dense Point Cloud Reconstruction From a Single Image,” in IEEE Access, vol. 7, pp. 137420-137431, 2019, doi: 10.1109/ACCESS.2019.2943235.

[8] G. -H. Lin, Y. -S. Xiao, H. -A. Hsieh, K. -Y. Liao, Y. -C. Liu and Y. -C. Fan, ”3D Point Cloud Matching Technology Based on Depth Image Based Rendering,” 2021 IEEE International Conference on Consumer Electronics (ICCE), 2021, pp. 1-2, doi: 10.1109/ICCE50685.2021.9427683.

[9] Peng, Rui, Ronggang Wang, Yawen Lai, Luyang Tang, and Yangang Cai. ”Excavating the potential capacity of self-supervised monocular depth estimation.” In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15560-15569. 2021.

[10] Hartley, Richard, and Andrew Zisserman. Multiple View Geometry in Computer Vision. 2nd ed. Cambridge: Cambridge University Press, 2004. doi:10.1017/CBO9780511811685.

[11] Moritz Menze and Andreas Geiger. Object Scene Flow for Autonomous Vehicles. Conference on Computer Vision and Pattern Recognition (CVPR),2015.