Current SOTA Stereo Depth Estimation Techniques

Abhigyan Roy - *AE21B002* and Reva Dhillon - *AE21B108,*

Abstract

Stereo depth estimation plays a crucial role in autonomous driving, robotics, and 3D perception systems. This project compares classical stereo correspondence methods (Block Matching and Semi-Global Block Matching) with two state-of-the-art deep learning–based approaches (Fast-ACVNet and Dusk Till Dawn). We implement and benchmark shallow and deep methods on the KITTI and Oxford RobotCar datasets. We have also implemented improvements for the shallow methods as well as tested out a novel deep learning method using FNOs. This report describes the methods, datasets, implementation pipeline and experiments, with results and quantitative evaluation.

I. Introduction

Stereo depth estimation aims to recover scene depth using two rectified camera images. Classical methods rely on handcrafted correspondence functions and smoothness priors, while modern approaches employ deep neural networks for robust and accurate disparity estimation. This project investigates both categories to understand their performance differences under varying lighting conditions and scene structures.

A. Problem Definition

Given a left–right rectified stereo image pair, estimate the disparity map and convert it to a depth map using the triangulation relation:

$$Z = \frac{fB}{d}$$

where f is focal length, B is baseline, and d is pixel disparity.

B. Motivation

Depth plays a key role in localization, SLAM, obstacle avoidance and scene reconstruction. However, classical methods struggle in low-light or textureless regions. Deep learning improves accuracy but requires significant computation and training data. This study compares the accuracy between both families of methods.

II. ALGORITHMIC DESCRIPTION

A. Classical Stereo Correspondence

Classical stereo vision solves correspondence using deterministic image-based matching, assuming brightness constancy and local smoothness of disparity. These methods operate without learning and depend entirely on handcrafted cost functions and geometric constraints.

- **Block Matching (BM):** Local window-based matching using SAD.
- **Semi-Global Matching (SGBM):** Aggregation of matching cost along multiple scanlines for smoothness.

These methods are computationally efficient but sensitive to lighting variations and low-texture areas.

Improvements:

- **Edge-overlayed**
- **Intensity Gradient matching using Sobel and Canny operators**
- **Census Transformed** [5]
- **Coarse-to-Fine**

B. Deep Learning Methods

Deep stereo matching formulates disparity estimation as an end-to-end learning problem, where neural networks infer correspondences through learned feature encoders and cost-volume regularization. These methods significantly outperform classical approaches in accuracy and generalization.

- **Fast-ACVNet (2022):** Builds an attention–concatenation cost volume for efficient and accurate disparity estimation. [2]
- **Dusk Till Dawn (2024):** Self-supervised nighttime stereo depth network for illumination-invariant feature extraction. [1]
- **FNOs :** Projects the data to the fourier space in the architecture. [4]

C. Methodology

- **Block Matching (BM)**

  BM computes local correspondences by comparing fixed-size windows using the Sum of Absolute Differences (SAD). Disparities are selected using a winner-takes-all strategy, where each block independently chooses the disparity with the lowest matching cost, making BM highly sensitive to noise and ambiguous regions.

- **Semi-Global Block Matching (SGBM)**

  SGBM augments local matching with semi-global cost aggregation along multiple scanlines to enforce smoothness constraints. This yields more stable disparity estimates, especially near depth discontinuities, while remaining computationally efficient.

- **Fast-ACVNet**

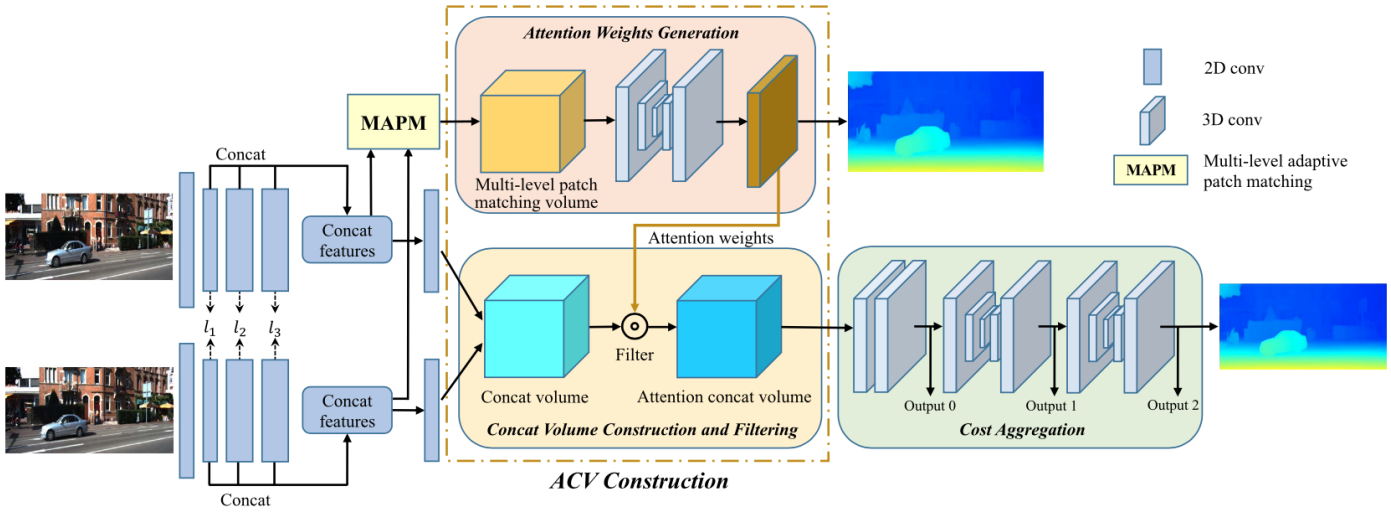

Fig. 1: The construction process of ACV consists of three steps: Attention Weights Generation, Initial Concatenation Volume Construction and Attention Filtering. First obtain feature maps at three different levels l1, l2 and l3 from the feature extraction module, and the number of channels for l1, l2 and l3 is 64, 128 and 128 respectively. l1, l2 and l3 are concatenated to form 320-channel concat features for the generation of attention weights. Then two convolutions are applied to compress the 320-channel concat features to 32-channel features for construction of the initial concatenation volume.[2]

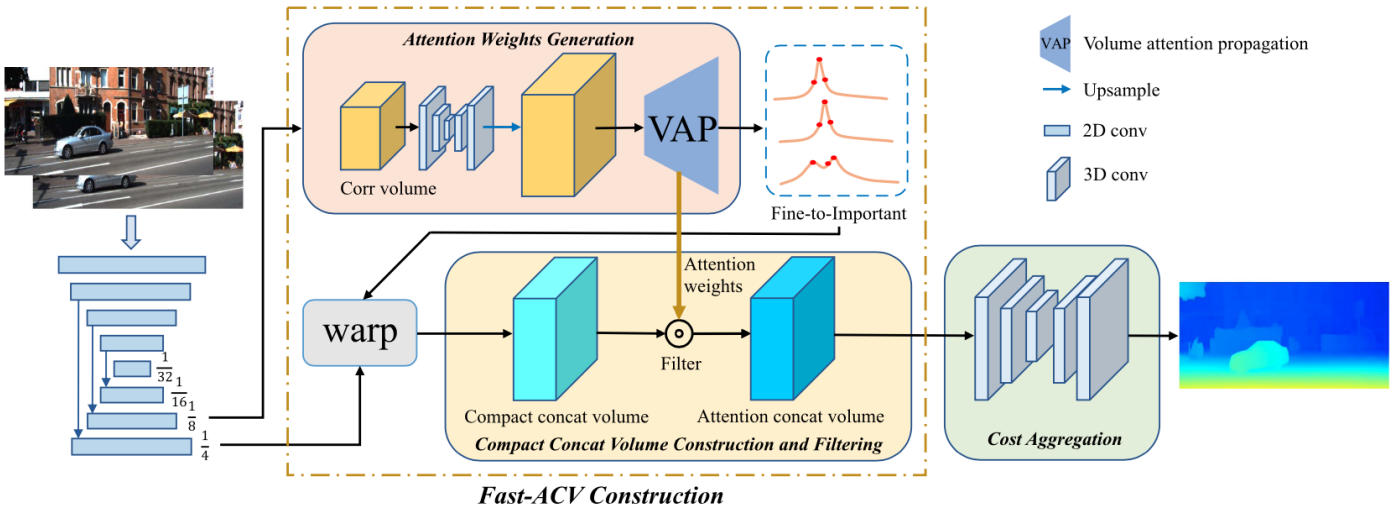

Fig. 2: First exploit a correlation volume to generate disparity hypotheses with high likelihood and the corresponding attention weights. Then we use the attention weights to filter the compact concatenation volume constructed based on disparity hypotheses, deriving our Fast-ACV.[2]

- SOTA algorithm for real-time depth estimation from stereo images.
- A novel cost volume construction method (attention concatenation volume - ACV), which generates attention weights from correlation clues to suppress redundant information and enhance matching-related information in the concatenation volume.
- Propose a Volume Attention Propagation module and a Fine-to-Important sampling strategy, which are key success factors of Fast-ACV.

- **Dusk Till Dawn (DTD)**

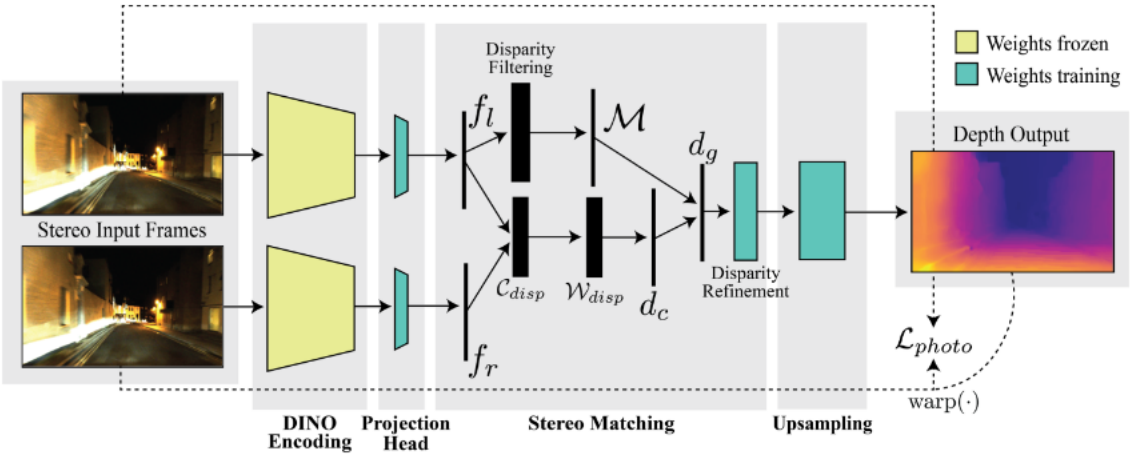

Fig. 2: Our approach consists of four main elements. Features are encoded independently for each input using DINO [13], a learnable projection head adapts these features and reduces their dimension, giving f_l and f_r. Stereo matching of the features then takes place, with disparity filtering yielding the mask $\mathcal{M}$, and the combination of f_l and f_r providing the correspondence volume $\mathcal{C}_{disp}$. $\mathcal{W}_{disp}$ is found by using softmax on $\mathcal{C}_{disp}$, which is used to find coarse disparity d_c. Coarse disparity and the mask combine to give global disparity d_g, which is refined and upsampled to give final depth.

Fig. 3: Model Architecture from DTD[1]

- Uses Visual Foundation Models (VFMs) for extracting lighting-invariant features
- Self-supervised photometric loss is used for training with no ground-truth depth
- Combined with smoothness and left–right consistency losses performs better than baseline methods and also presents novel evaluation metrics

- **Proposed Method (FNO-based Depth Estimation)**

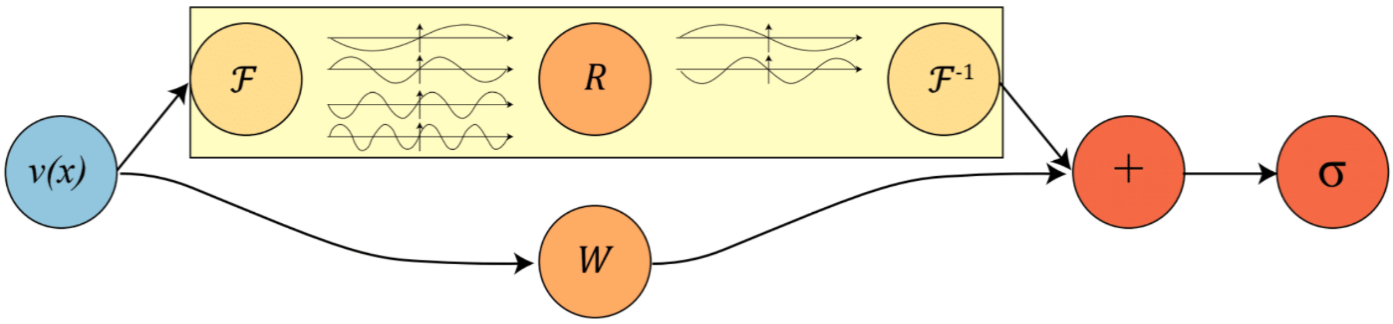

Fig. 4: FNO Architecture[**?**]

- The FNO layers project the data to Fourier space (Convolutions to Multiplications).
- Image data is composed of wavelengths.
- Accurate results expected because of the high nonlinearity in the model and because it is natural to represent image data in fourier space.

D. Datasets

1) KITTI Stereo Dataset: Outdoor driving dataset with ground-truth LiDAR disparity - used for evaluating ShallowResults1 and deep stereo matching methods. [6]

2) Oxford RobotCar Dataset: Contains day, dusk, and nighttime driving sequences - DTD model is evaluated on nighttime subsets [7]. Stereo camera intrinsics and calibration files are used from the RobotCarDataset SDK.

3) Custom + Demo Dataset: A small manually collected dataset is used to test generalization properties.

E. Implementation Details

Software and Tools:

- Python (OpenCV, NumPy, PyTorch)
- DTD official repository
- ACVNet official repository
- RobotCarDataset-SDK

Pipeline:

1) Run BM and SGBM for classical disparity on rectified stereo images and the improvements
2) Run ACVNet and DTD models for respective datasets
3) Evaluate using error metrics (MAE, RMSE, SD, SAD, SSD, Bad Pixels)
4) Generate depth maps and tabulate errors for both

III. OUTPUT

A. *Shallow Methods*

1) **Edge-Overlayed Method**

We use Canny's edge detection to extract the edges from the raw image and overlay it as black/white edges based on the lighting of the original image. BM and SGBM are performed on this modified images.

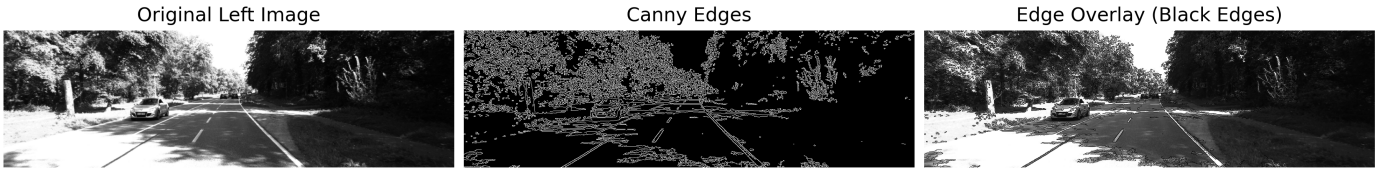

Fig. 5: Original Image, Canny Edge Map, and Canny Overlayed Image

TABLE I: Error Metrics for Edge-Overlayed Stereo Matching on KITTI $Image000157_10$

Method	MAE	SD	RMSE	SAD	SSD	Bad3	Bad5
StereoSGBM							
Original	0.902	1.576	2.011	52776.5	192394.7	3.42%	1.91%
Canny Edge Overlay	0.976	1.848	2.090	50901.7	227838.3	4.17%	2.35%
Block Matching (BM)							
Original	0.825	1.383	1.864	23543.0	76992.6	2.23%	1.00%
Canny Edge Overlay	0.842	1.502	1.722	28580.3	100641.4	2.58%	1.24%

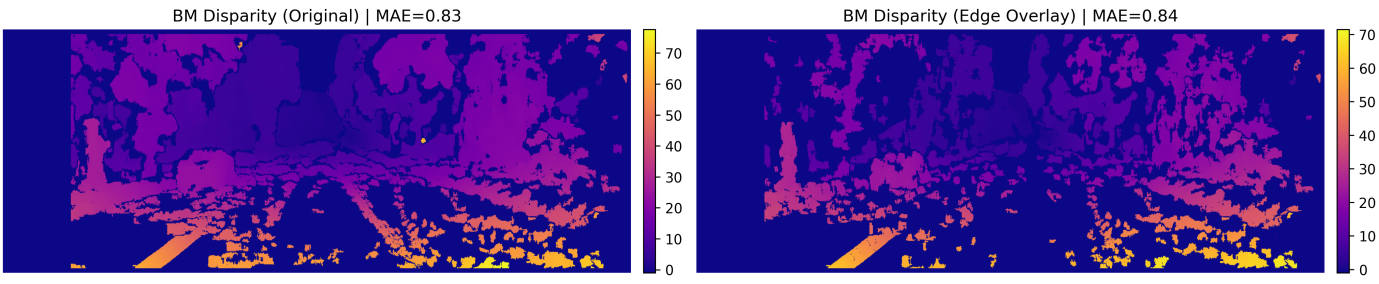

Fig. 6: BM Disparity Maps: Original vs Canny Edge Overlay

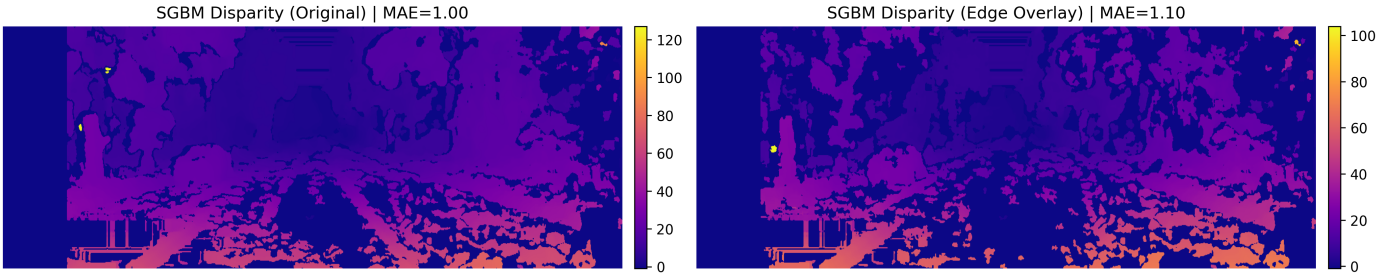

Fig. 7: SGBM Disparity Maps: Original vs Canny Edge Overlay

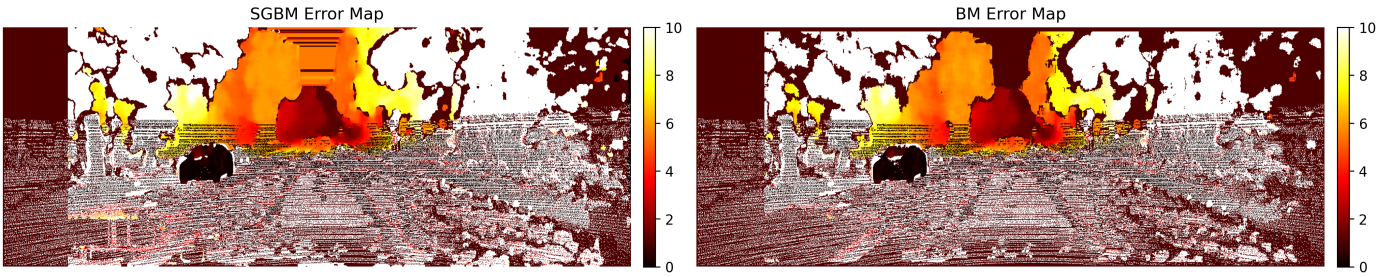

Fig. 8: Error Maps for SGBM and BM

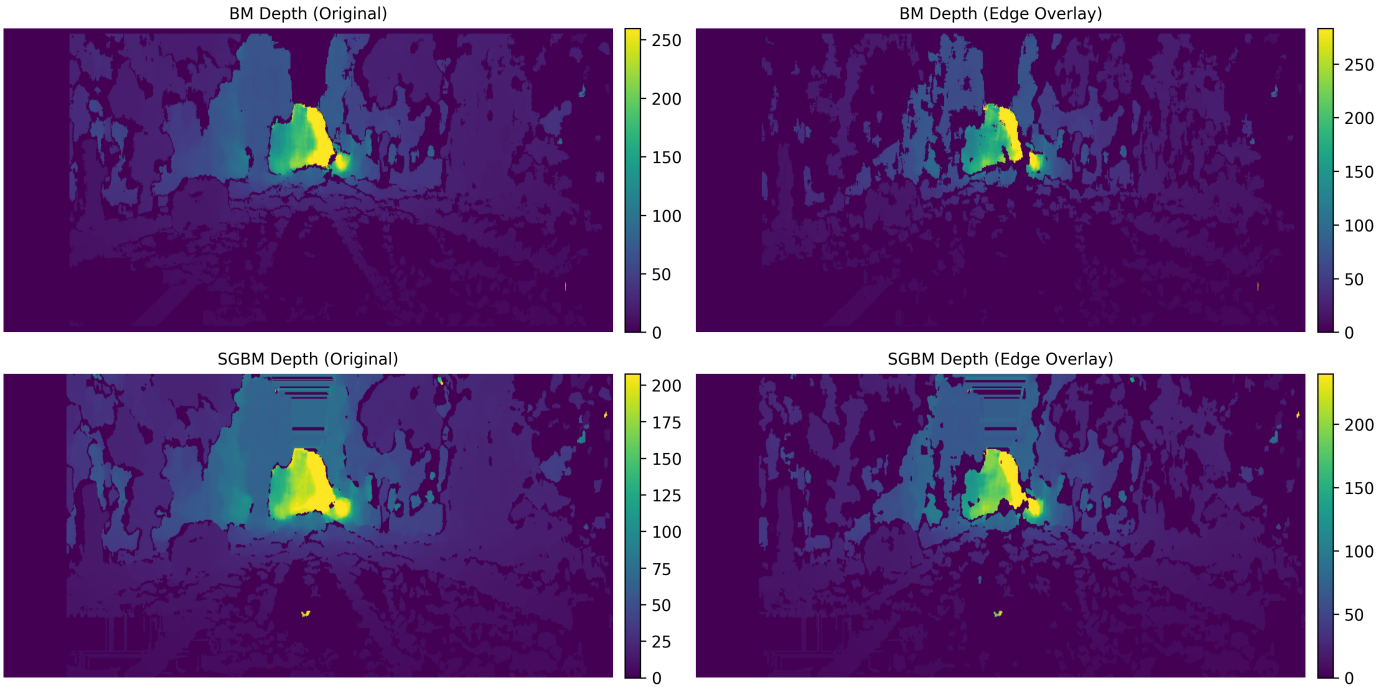

Fig. 9: Depth Maps for BM and SGBM (Raw vs Edge-Overlayed)

2) **Intensity-Gradient Matching Method**

We use Sobel and Canny operator to obtain the intensity gradient of the image pixel intensities and perform BM and SGBM on them.

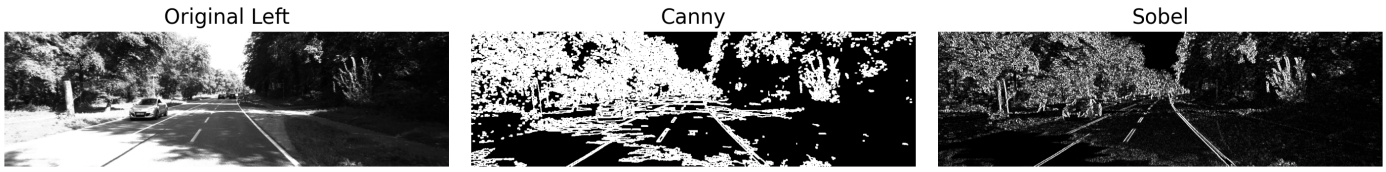

Fig. 10: Original Image, Canny Edges, and Edge Input

TABLE II: Error Metrics for Stereo Matching on KITTI Image 000157_10

Method	MAE	SD	RMSE	SAD	SSD	Bad3	Bad5
				StereoSGBM			
Raw	0.998	2.169	2.388	63233.3	361058.0	4.40%	1.96%
Canny	2.802	6.521	7.098	126848.1	2280563.0	21.01%	9.17%
Sobel	0.980	1.807	2.055	35311.6	152215.1	5.17%	2.39%
				Block Matching (BM)			
Raw	0.825	1.672	1.864	40026.7	168573.4	2.23%	1.00%
Canny	3.125	9.900	10.381	64241.7	2215459.5	14.61%	7.86%
Sobel	0.717	1.251	1.442	16672.9	48348.8	2.48%	0.98%

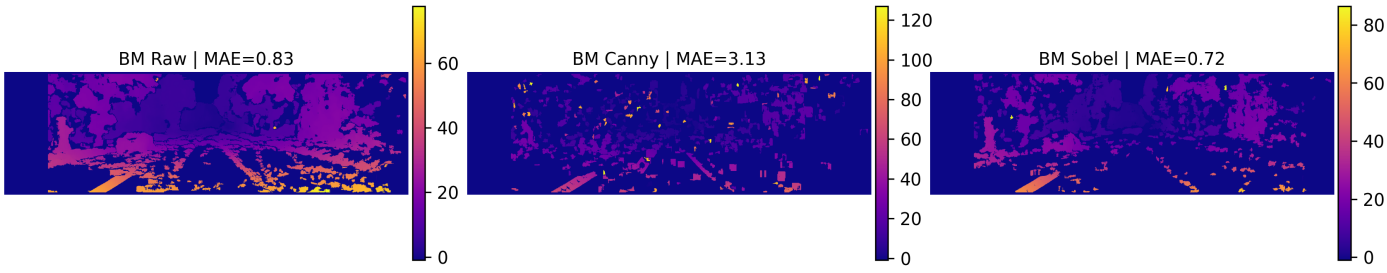

Fig. 11: BM Disparity Comparison (Raw vs Canny vs Sobel)

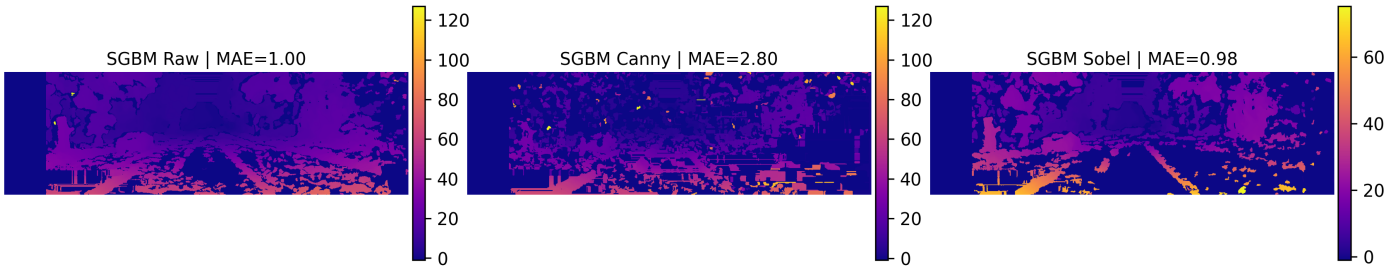

Fig. 12: SGBM Disparity Comparison (Raw vs Canny vs Sobel)

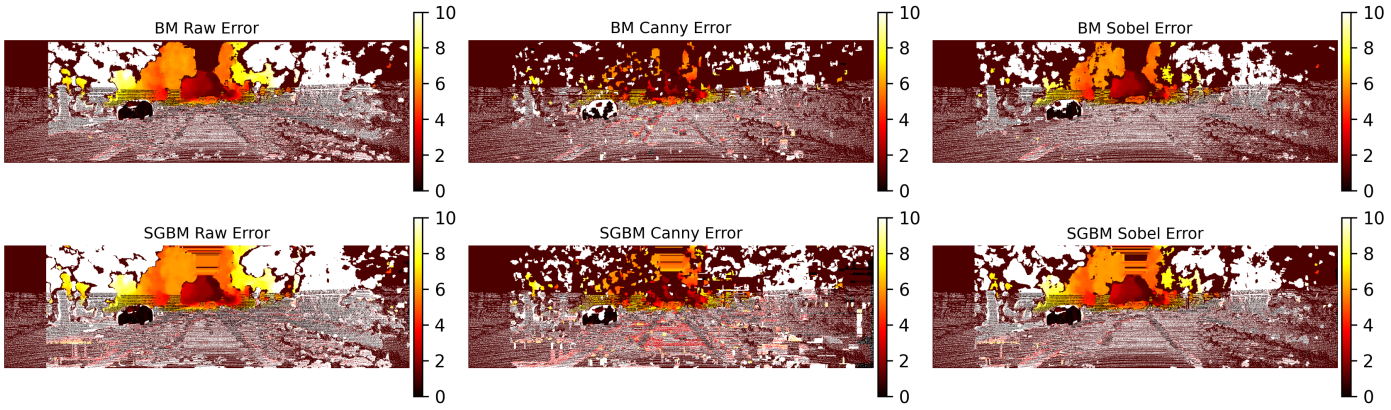

Fig. 13: Error Maps (SGBM and BM)

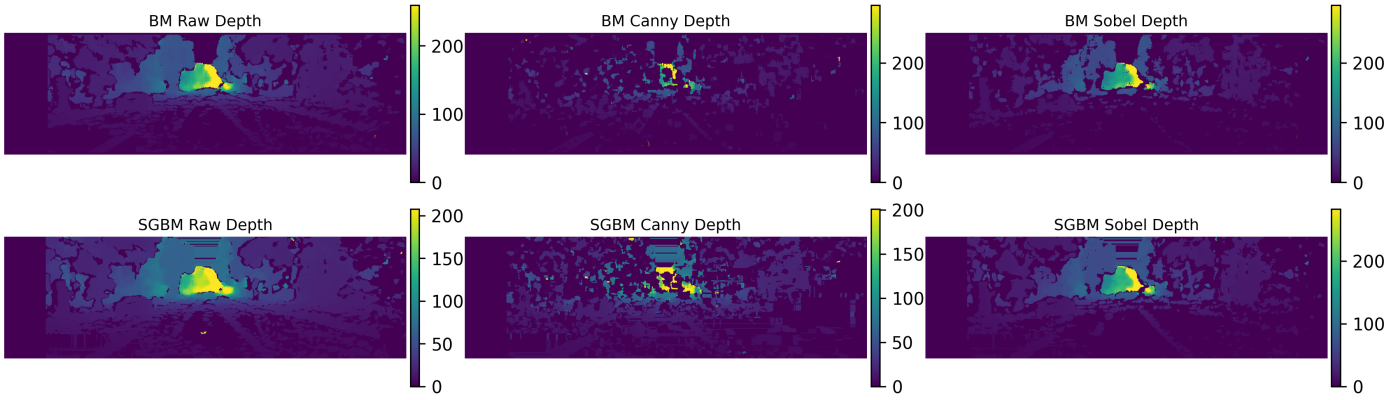

Fig. 14: Depth Maps for BM and SGBM (Raw vs Canny vs Sobel)

3) Census Transform Based Matching

We apply the Census Transform [5] to convert local neighborhoods (a fixed 5 x 5 window) into binary descriptors, enabling BM and SGBM to match pixels more reliably under lighting variations as each pixel is converted into a binary string that

encodes how its intensity compares with its neighbors.

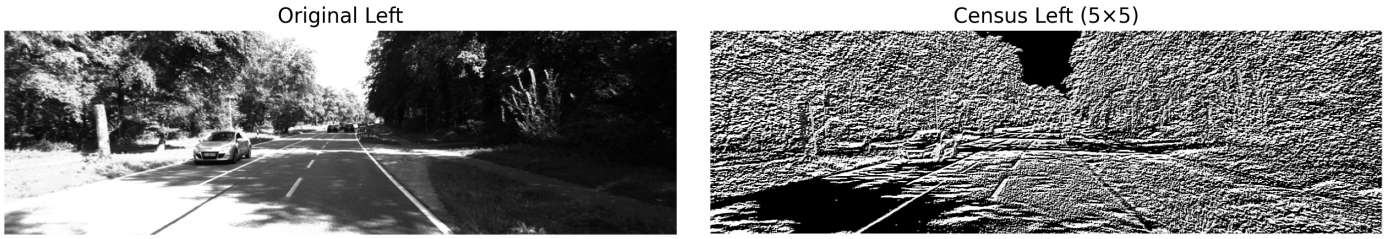

Fig. 15: Original Left and Census Transformed Left Images

TABLE III: Error Metrics for RAW vs Census Stereo Matching on KITTI Image 000157_10

Method	MAE	SD	RMSE	SAD	SSD	Bad3	Bad5
				StereoSGBM			
Raw	0.998	2.169	2.388	63233.3	361058.0	4.40%	1.96%
Census	0.953	1.714	1.961	52187.4	210600.7	4.29%	1.70%
				Block Matching (BM)			
Raw	0.825	1.672	1.864	40026.7	168573.4	2.23%	1.00%
Census	0.763	1.383	1.580	23543.0	76992.6	2.67%	0.97%

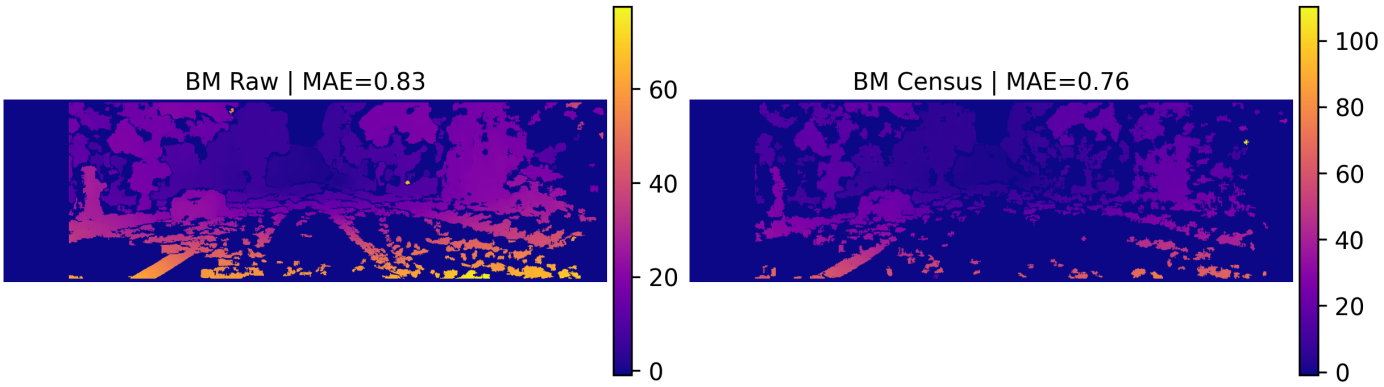

Fig. 16: BM Disparity Comparison (Raw vs Census)

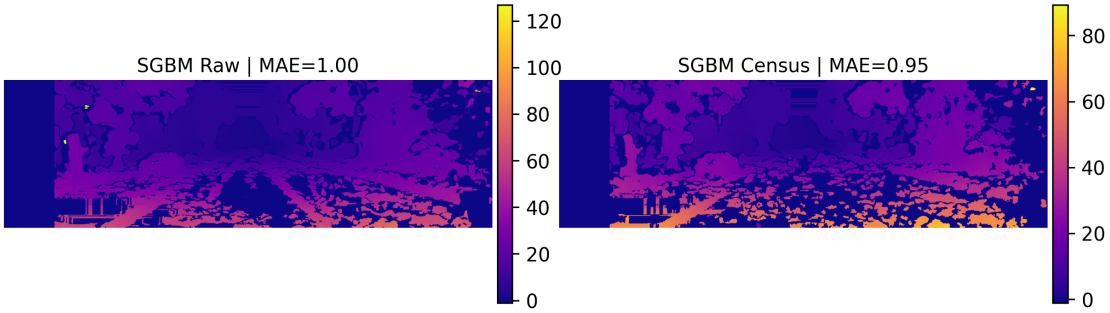

Fig. 17: SGBM Disparity Comparison (Raw vs Census)

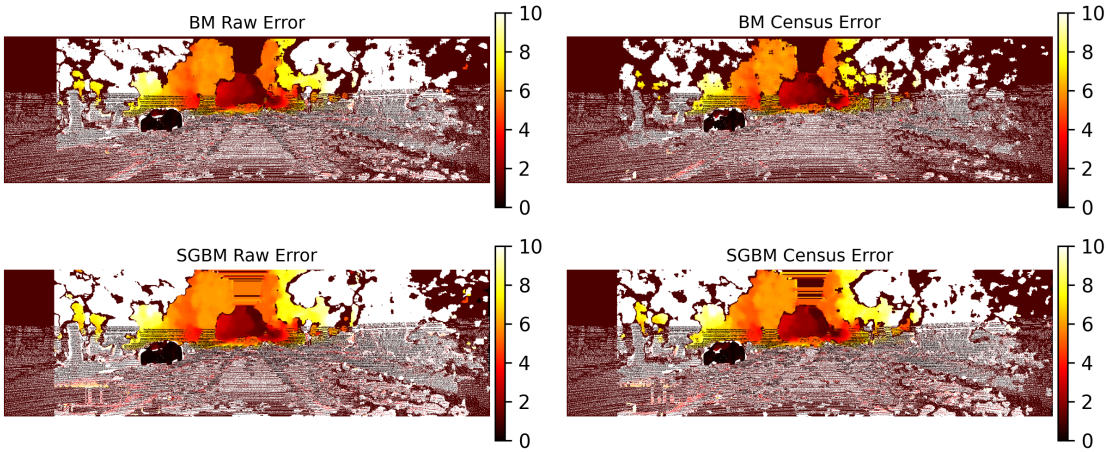

Fig. 18: Error Maps for BM and SGBM (Raw vs Census)

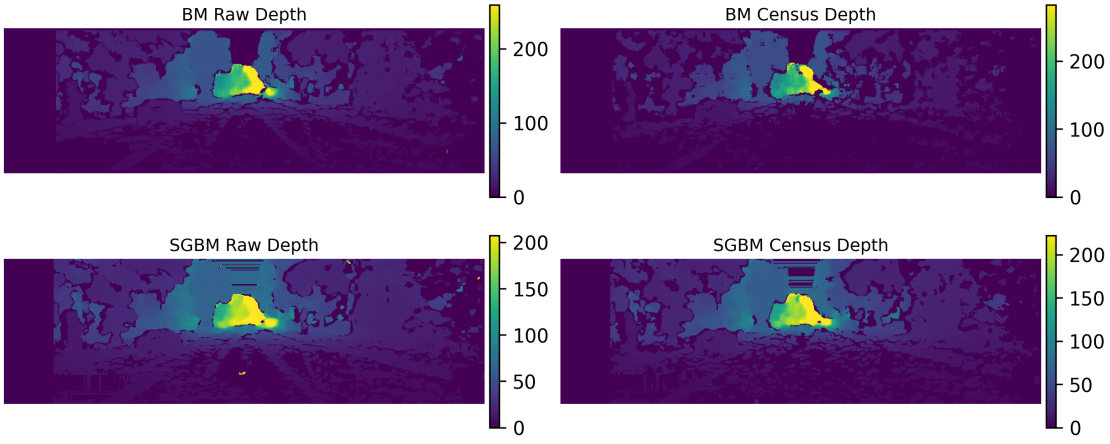

Fig. 19: Depth Maps for BM and SGBM (Raw vs Census)

4) **Coarse-to-fine Strategy**

Here, we use BM or SGBM to generate an initial disparity map (Coarse). Then for each block, we check a fine window of pixels ($d_0 \pm \Delta$) to refine the disparity estimate based on a hybrid cost of MEA of image intensities, intensity gradients and census transform values.

Fig. 20: Original Left Image

TABLE IV: Error Metrics for RAW vs Coarse-to-Fine Matching on KITTI Image 000157_10

Method	MAE	SD	RMSE	SAD	SSD	Bad3	Bad5
			StereoSGBM				
Raw	1.709975	5.296	5.5647	160428.14	2905234.5	5.95%	4.761%
Coarse-to-fine	1.713346	5.273	5.5443	160744.36	2883983.5	6.08%	4.797%
			Block Matching (BM)				
Raw	1.056236	4.123403	4.256534	53237.445	913205.8	1.453%	0.803%
Coarse-to-fine	1.056174	4.1234007	4.2565165	53234.344	913198.2	1.450%	0.804%

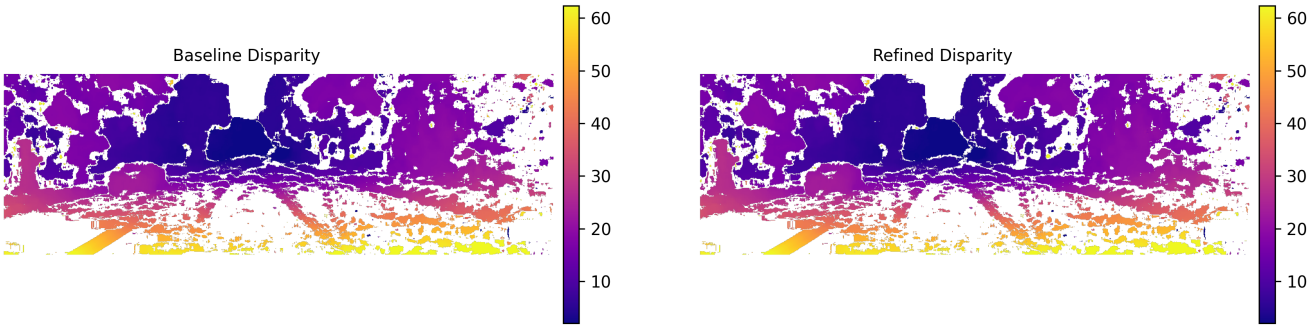

Fig. 21: BM Disparity Comparison

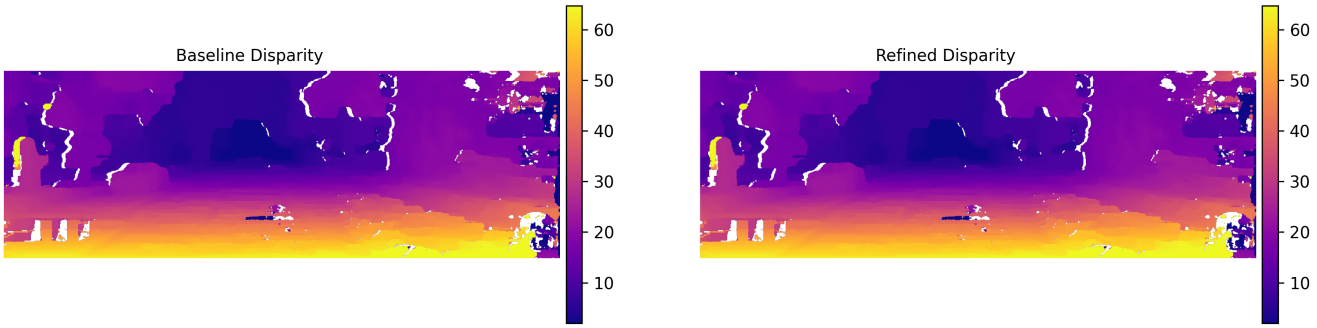

Fig. 22: SGBM Disparity Comparison

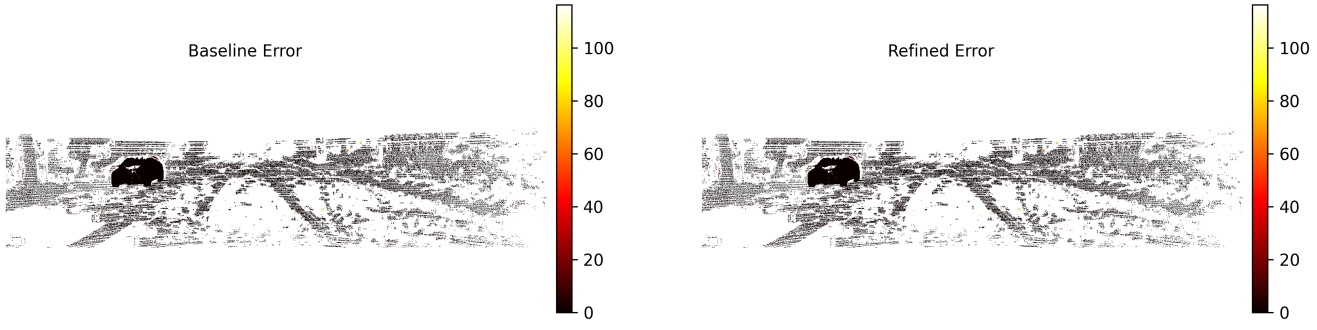

Fig. 23: Error Maps for BM

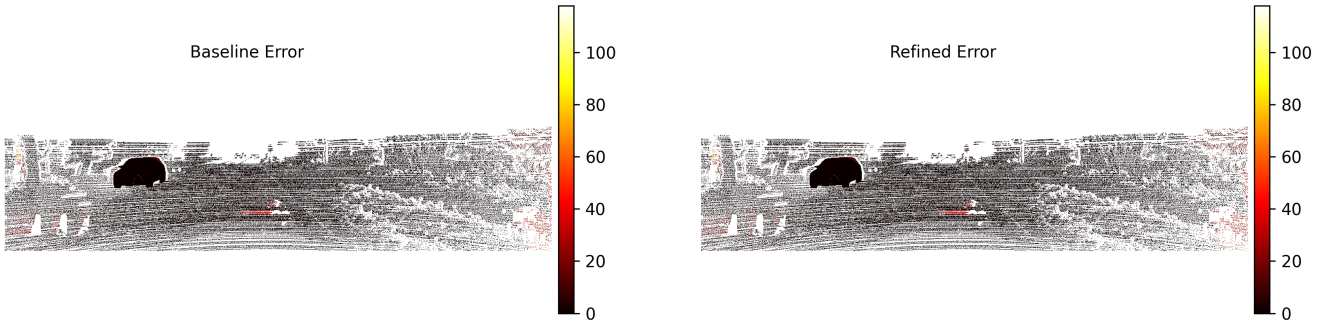

Fig. 24: Error Maps for SGBM

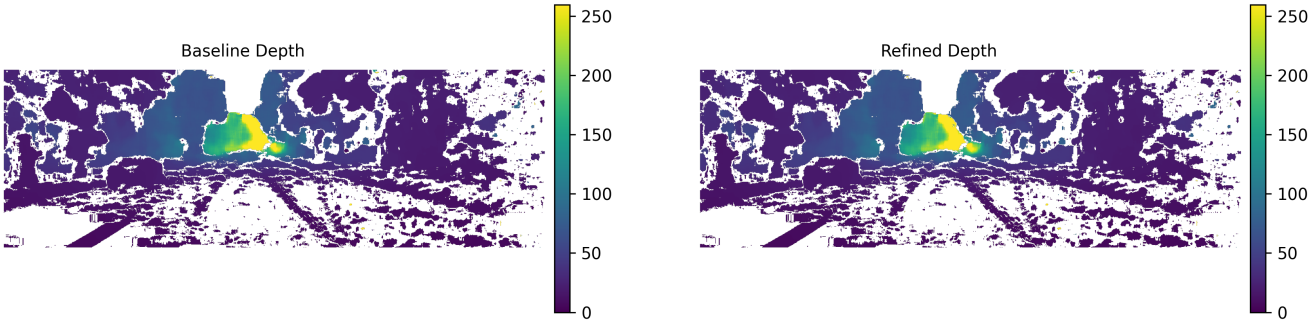

Fig. 25: Depth Maps for BM

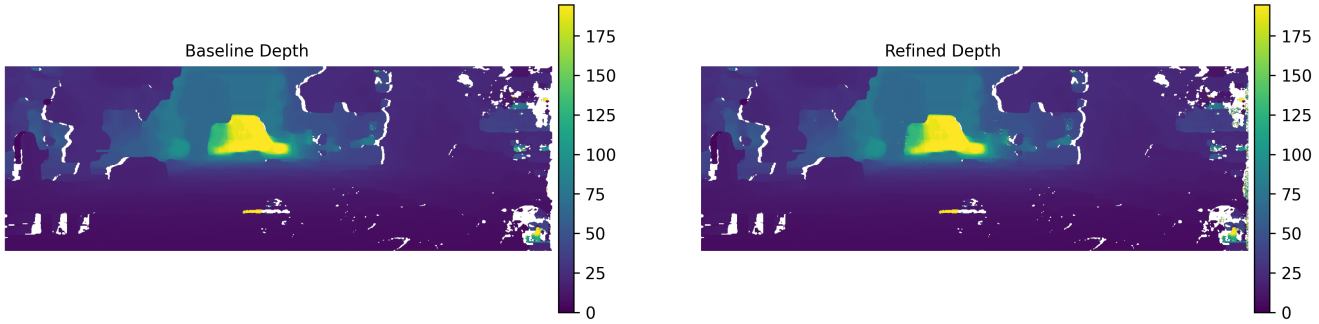

Fig. 26: Depth Maps for SGBM

1) Observation:

- For the shallow methods, variation of windows size and the number of pixels it slid over, majorly affected the disparity results. Other parameters available through OpenCV python package like ignoring speckles of certain size and range did not produce as much of a major effect.

- Edge-overlayed results showed improvement only in SAD for SGBM method and in SD, RMSE, SAD, SAD and SSD for BM method, while it showed slightly poorer performance on all other metrics.
- For intensity-gradient matchind results, Sobel transformed images outperformed the raw images while Canny was worse for all error metrics.
- Census transformed images also outperformed raw images for both BM and SGBM methods on all error metrics.
- Coarse to fine strategy shows better results for BM while there is scope for improvement using SGBM.

B. Deep Methods

- DTD

Fig. 27: Input Stereo Images

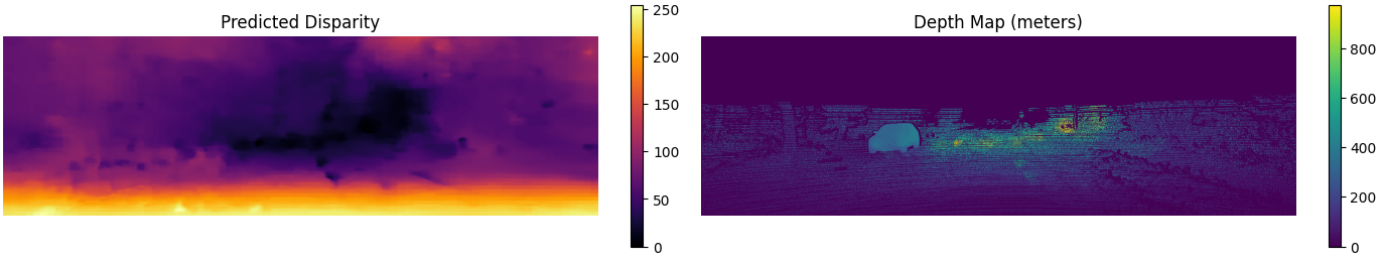

Fig. 28: Disparity and Depth Maps

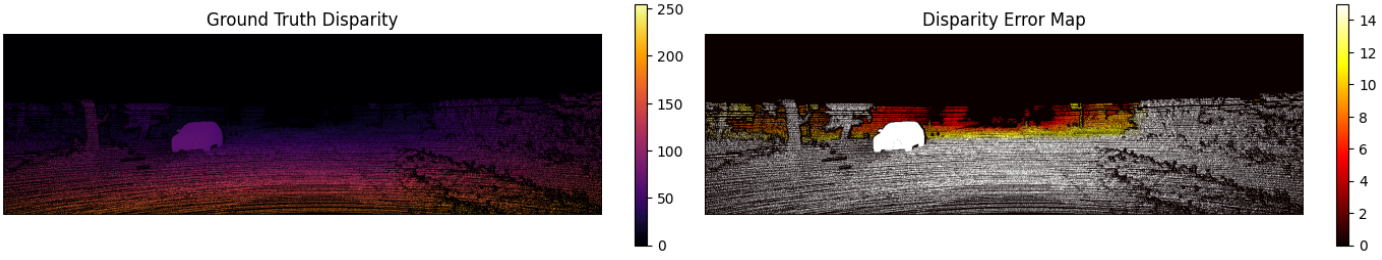

Fig. 29: Disparity Error with Ground Truth

- FNO

Fig. 30: Input Stereo Images

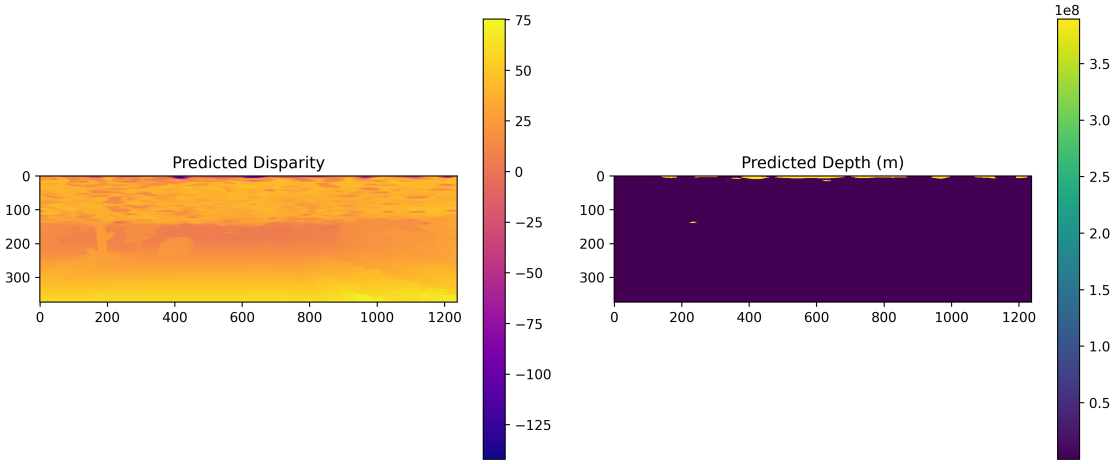

Fig. 31: Disparity and Depth Maps

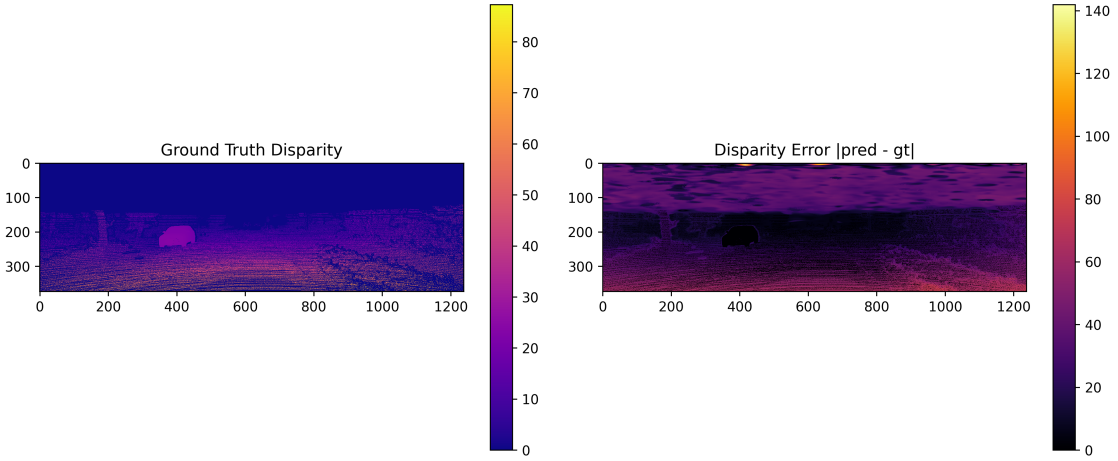

Fig. 32: Disparity Error with Ground Truth

C. Test/Demo Data

Custom Data

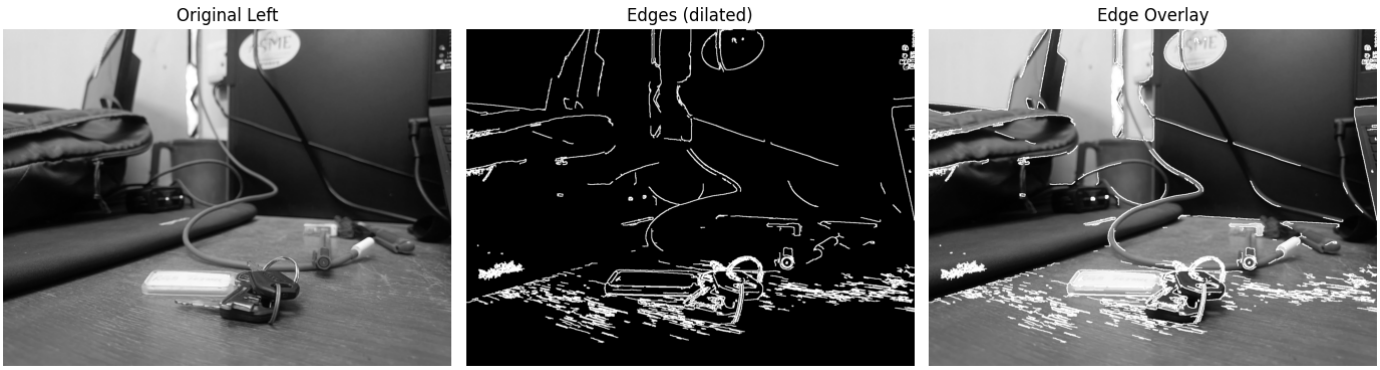

Fig. 33: Input Image and Edge Overlay

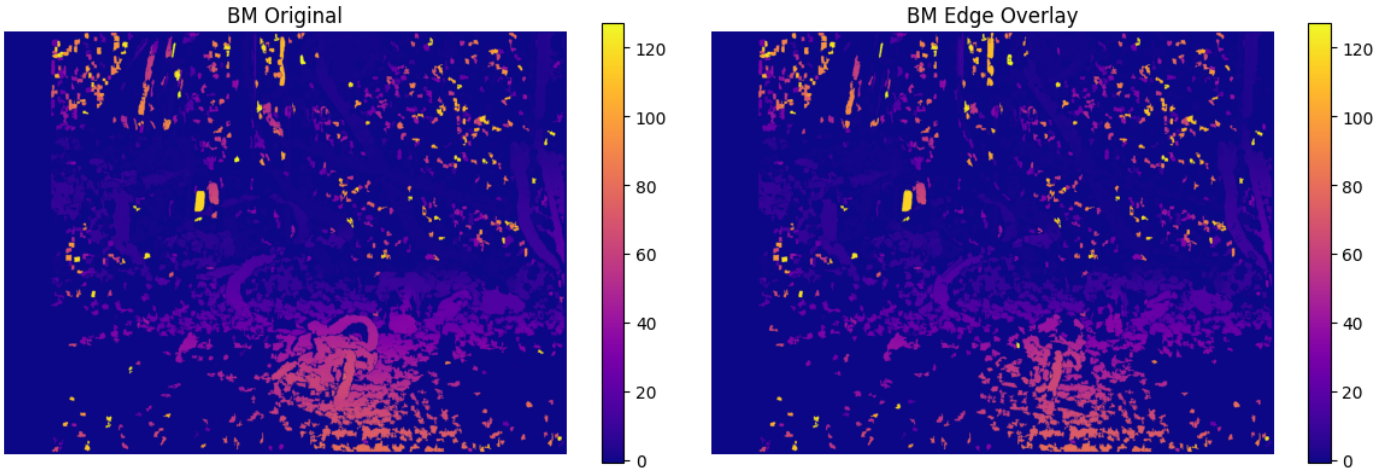

Fig. 34: BM Disparity with Edge Overlay

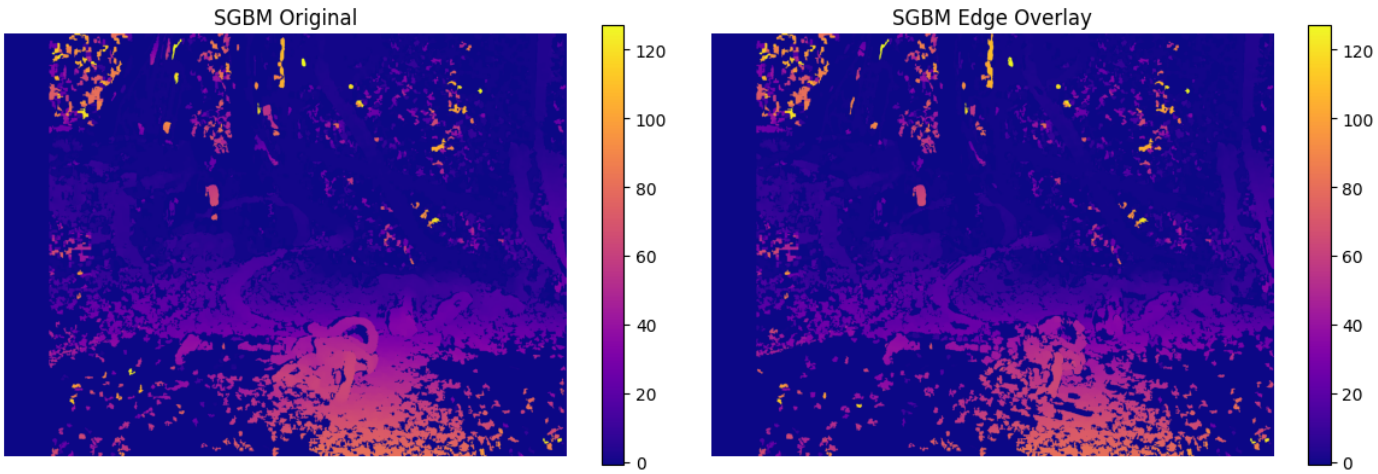

Fig. 35: SGBM Disparity with Edge Overlay

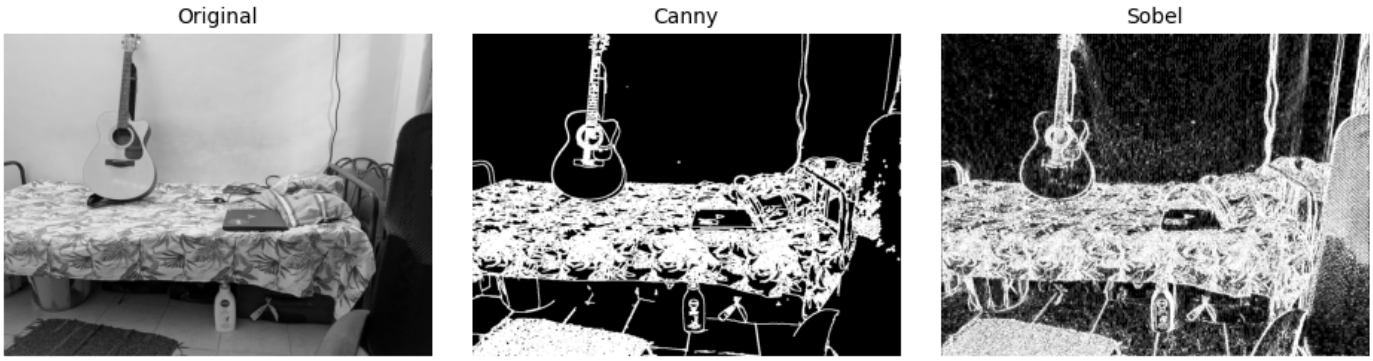

Fig. 36: Input Image with Sobel and Canny Operators

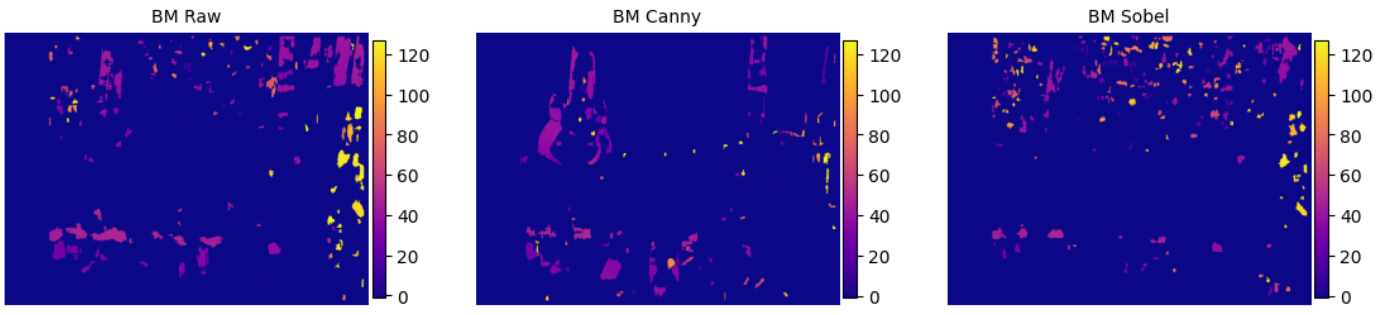

Fig. 37: BM Disparity with Sobel and Canny Operators

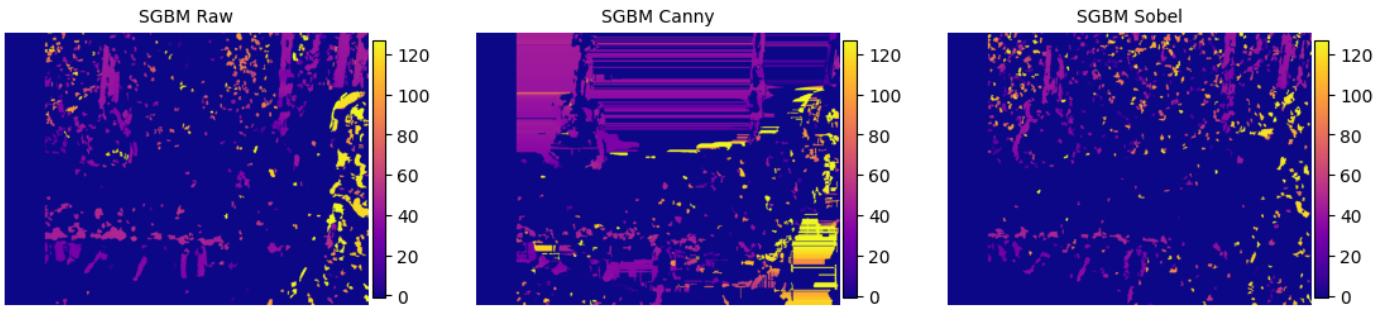

Fig. 38: SGBM Disparity with Sobel and Canny Operators

Demo data using Census Transform

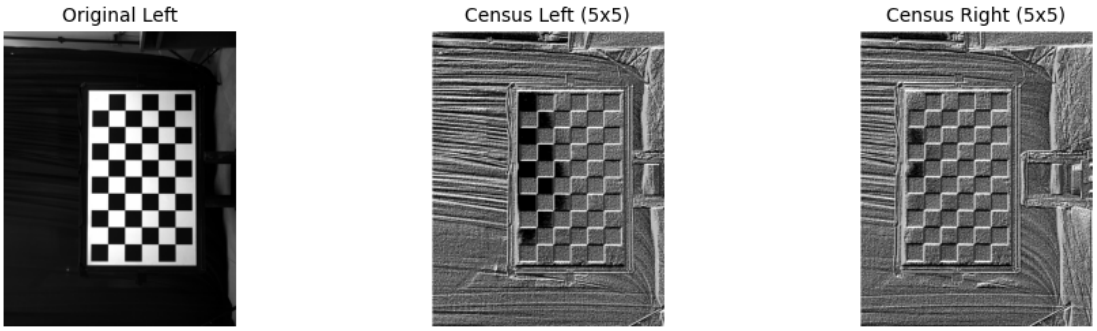

Fig. 39: Input Image — Checkerboard

Fig. 40: BM Disparity — Checkerboard

Fig. 41: SGBM Disparity — Checkerboard

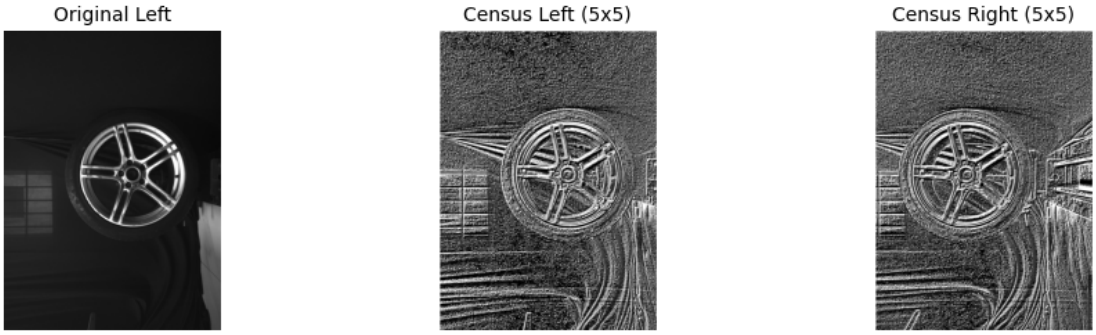

Fig. 42: Input Image — Tyre

Fig. 43: BM Disparity — Tyre

Fig. 44: SGBM Disparity — Tyre

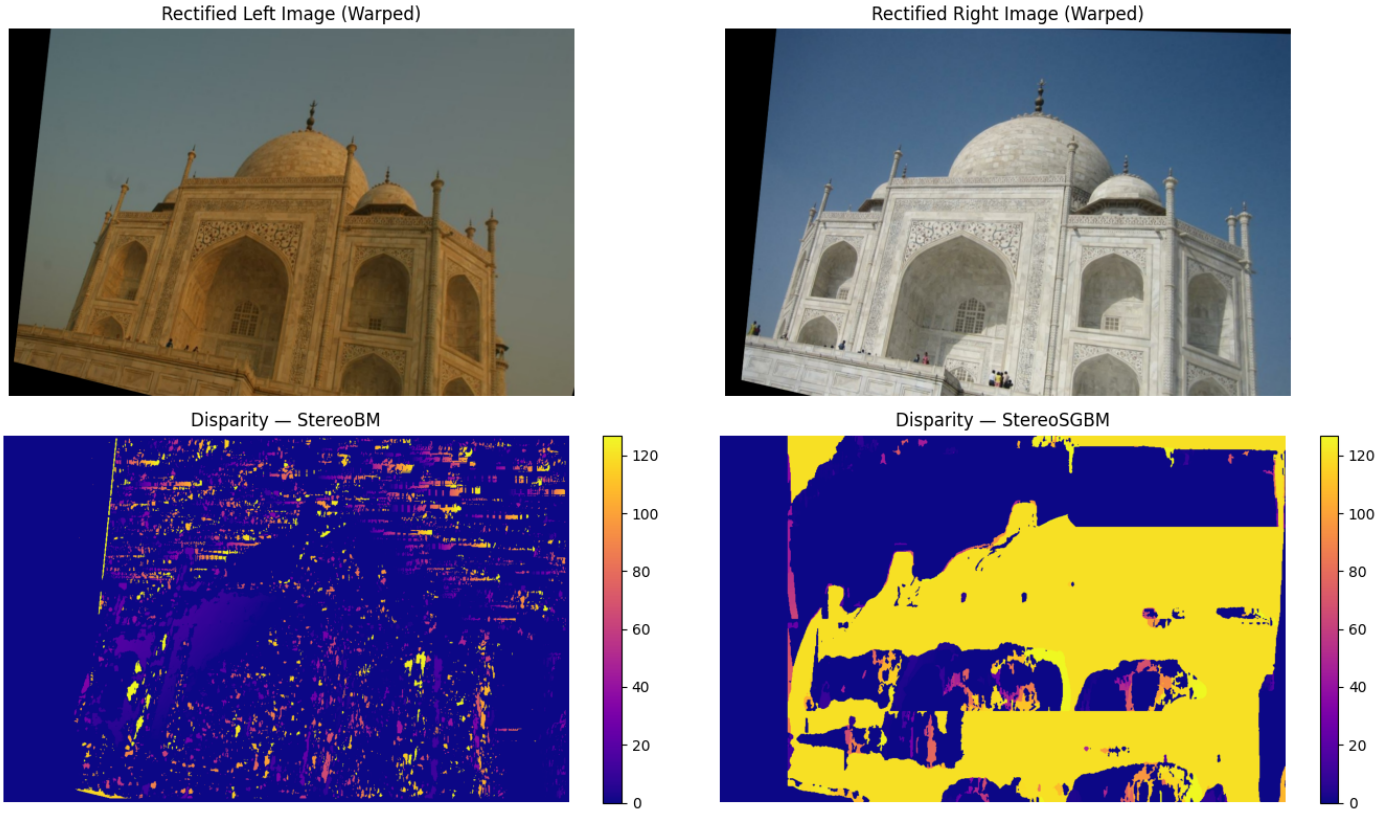

Fig. 45: Taj Mahal Different Views

IV. Conclusion

The experimental evaluation demonstrates that classical stereo correspondence is highly sensitive to preprocessing and matching strategies, with different transformations influencing disparity quality in distinct ways. Window size and search range had the strongest impact across all shallow methods, while OpenCV's secondary parameters contributed minimally. Among the tested enhancements, Census Transform consistently delivered the most robust improvement for both BM and SGBM across nearly all error metrics, confirming its strong invariance to illumination and local intensity variations. Sobel-based gradient matching also improved performance relative to raw images, whereas Canny-edge inputs degraded results due to excessive sparsity and loss of texture information. Edge-overlaying techniques provided only marginal or inconsistent benefits. Finally, using a coarse-to-fine strategy for matching also showed promising results. Overall, the results confirm that carefully chosen preprocessing—particularly Census and gradient-based transformations can strengthen classical stereo performance, even without deep learning.

The deep learning methods generally produced smoother and more stable disparity maps compared to the classical approaches, particularly in regions where traditional matching struggled. Fast-ACVNet benefited from its attention-based cost volume, while Dusk Till Dawn showed improved robustness under low-light conditions due to its foundation model features. The preliminary FNO-based results indicate that operator-learning may offer an alternative way to model correspondence, though further evaluation is needed. Overall, the deep models showed potential advantages, but their performance depends strongly on training data, parameter tuning, and computational resources.

References

[1] M. Vankadari *et al.*, "Dusk Till Dawn: Self-supervised Nighttime Stereo Depth Estimation using Visual Foundation Models," 2024. Source Code @ Github. Available: https://github.com/madhubabuv/dtd

[2] X. Guo, Y. Li, and S. Yi, “Accurate and Efficient Stereo Matching via Attention Concatenation Volume,” 2022. Source Code @ Github. Available: https://github.com/gangweix/Fast-ACVNet

[3] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations, 2021. URL https://arxiv.org/abs/2010.08895.

[4] Z. Li, P. Raissi, and G. E. Karniadakis, “Geometry-Informed Neural Operator for Large-Scale 3D PDEs,” 2023.

[5] M. A. Ibarra-Manzano, D. L. Almanza-Ojeda, M. Devy, J. L. Boizard, and J. Y. Fourniols, “Stereo Vision Algorithm Implementation in FPGA Using Census Transform for Effective Resource Optimization,” *Proc. Euromicro Conf. Digital System Design*, 2009.

[6] M. Menze and A. Geiger, “Object Scene Flow for Autonomous Vehicles,” in *Proc. CVPR*, 2015.

[7] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “1 Year, 1000 km: The Oxford RobotCar Dataset,” *Intl. J. Robotics Research (IJRR)*, 2017.