Low-Light Depth Enhancement for UAV Applications

Shane Allan
University of Alberta
sjallan@ualberta.ca

Sophia Wagner
University of Alberta
sjwagner@ualberta.ca

Almas Sahar
University of Alberta
asahar1@ualberta.ca

Abstract

Depth estimation is critical for applications such as autonomous driving, augmented reality, 3D reconstruction, and robot vision which require a 3D assessment of the environment. Depth information can be obtained accurately by active sensors such as light detection and ranging (LiDAR) in real-time applications, however these can be costly. To overcome the cost and accuracy trade-off, stereo cameras can be used to estimate depth more precisely with robust and well-trained algorithms. In this work, an enhanced depth estimation methodology is proposed using a low-cost stereo-vision camera that can deliver quality depth images in low-light environments suitable for real-time implementation. To address this problem, a convolutional neural network (CNN) is recommended to enhance the raw stereo red, green, and blue (RGB) and depth images for use on a prototype unmanned aerial vehicle (UAV). To deliver the real-time performance of the trained CNN model for the intended application, a data set of varying quality and light conditions is generated. The experimental network architecture produces an average depth accuracy improvement of 6.38% at depths greater than 5.5m over the input. However, the input outperforms our model by an average of 4.01% at depth ranges closer than 5.5m. The code is available at https://github.com/sophiajwagner/depth-image-enhancement-for-UAV-exploration

1. Introduction

Depth estimation (DE) is one of the most important subtasks in 3D scene reconstruction [11]. Considerable advancements in robotics engineering and autonomous vehicles have increased the requirement for precise depth measurements [9]. Autonomous vehicles navigate and control themselves using perception systems that create a three-dimensional map of their environment [16]. DE is a typical computer vision task that estimates depth from one or more two-dimensional (2D) images. The DE task takes the red, green, and blue (RGB) image input and returns a depth image output. The depth image contains data of the distance from the object to the cameras viewpoint [9]. Utilizing this methodology to estimate reliable depth information of a scene are active 3D scanners such as red, green, blue and depth (RGB-D) sensors or alternatively, 3D light detection and ranging (LiDAR) scanners[12].

LiDAR generates a three-dimensional map of the environment by emitting laser pulses. Depth is inferred by measuring the time difference between the emitted light from the source and the reflected light returning from the scene [14]. However, LiDAR is not feasible for every application. Stereo matching, another technique for depth estimation, has benefited greatly from the advances in deep learning. Stereo matching recognizes point correspondences across two cameras and then uses the relative position of two cameras to reconstruct the depth of each point in the image [16]. A stereo matching algorithm first estimates the disparity map between the left and right images before applying geometric triangulation to reconstruct depth. Stereoscopic depth estimation relies on estimating the relative displacement of an object between two views. This displacement is termed disparity and is inversely proportional to the object's distance. For this purpose, a dense correspondence between pairs of images is performed, followed by a disparity refinement phase [3].

For the unmanned aerial vehicle (UAV) used in this work, low light depth enhancement is achieved through a convolutional neural network (CNN). The Zed 2 camera is used to capture low/high light and low/high quality images, which are post-processed for depth enhancement. Suffering from large amounts of noise and occlusions, the stereo camera under-performs in low-lighting conditions. The aim of our proposed network is to enhance the quality of the raw images. The CNN model is trained and tested on the database of the gathered low/high light and low/high-quality images. For depth enhancement, we performed three different methods. The first approach attempted to improve the quality of low-light stereo image pairs through the use of a CNN model followed by a second CNN enhancement of a derived depth image. The second approach used only depth images which were enhanced using a CNN model. The third approach used a combination of stereo

and depth images as inputs to the CNN to improve depth accuracy.

2. Literature Review

Stereo matching remains one of the most widely used techniques for depth estimation in the literature due to its strong resemblance with human binocular vision [5].
Alagoz [2] states that depth estimation from stereo images is composed of two steps, the first being the estimation of the disparity map from the stereo image pair by means of a stereo-matching algorithm and the second being the transformation of the disparity map into the depth map. The disparity map contains pixel disparities between the projection of the same object in the left and right images while the depth map is composed of the depth estimates of each pixel. Therefore, this combination of techniques can render useful visual information for depth perception [2]. The algorithm for depth estimation from stereoscopic images produces the disparity map between the left and right images before applying a geometric triangulation which results in imperfect depth estimates. The depth error is quadratically proportional to the depth of the observed object. Stereo reconstruction is challenging within texture-less, poorly illuminated, and occluded regions [3].
Bracha [3] proposed a refinement network for depth estimation based on the algorithm which used depth instead of disparity. They suggested a two-step methodology in which first the stereo range estimation models are trained on disparity, and second, a depth refinement model optimized using the computed disparity map and the left and right images. As a result, the refined depth depicted better images. This two-step method split the overall pipeline into a disparity estimation stage followed by a dedicated depth refinement model which resulted in significantly improved final depth estimation. The proposed algorithm was evaluated on benchmark data sets and demonstrated a 15% to 39% improvement in depth accuracy over the baseline. Additionally, the refined output exhibits a weaker quadratic relation between distance and depth error [3].

To accomplish high-level computer vision tasks like image classification, object recognition, and semantic segmentation, CNN's have delivered substantial results. CNN's are being used to deliver low-level computer vision tasks including optical flow prediction and depth estimation from stereo images [4]. Zbontar [17] used various CNN-based architectures for computing image patches. It was found that concatenating the left and right image patches as different channels produced the most accurate results, but suffered from computational inefficiency. Luo [6] substituted the concatenation layer and subsequent processing layers with a single product layer to compute the score. They trained the network using cross-entropy over all possible disparities and to obtain calibrated scores. This resulted in better matching performance, however, their algorithm suffered from texture-less regions and regions with repetitive patterns. Nonetheless, the algorithm delivered competitive results when tested on benchmark data sets [6]. Tankovich [15] presented HITNet (Hierarchical Iterative Tile Refinement Network) for real-time stereo matching that used a fast multi-resolution initialization step to determine high-resolution matches. This method uses learned features, followed by tile initialization fused using propagation and fusion steps. Additional accuracy is provided by the use of slanted support windows with learned descriptors. However, a data set with ground truth depth is required for training purposes. The proposed approach was evaluated on benchmark data sets and showed competitive results at minimal computational time.

3. System Components

Hardware systems create additional complexities to all software deployment, especially when attempting to replicate the simulated performance. Firmware issues, software dependencies, and computational power are all barriers to efficient real-time performance.
The model developed is deployed on a prototype quadcopter to improve its future path planning and obstacle avoidance capabilities in different lighting saturation conditions. Robot Operating System (ROS), Python, and PyTorch are the primary softwares used in this project. The Zed 2 stereo camera, a DJI F450 flight platform, and a Jetson Xavier NX onboard computing module with a 384-core NVIDIA Volta GPU comprise the main hardware components of the system. A full summary of the onboard software and hardware systems being used can be found in table 1.

Table 1. Prototype quadcoptor hardware and software components

Hardware	Software
DJI F450 Flamewheel Frame	PyTorch 1.10.0
Garmin ToF Laser Rangefinder	Python 2.7, 3.6
Pixhawk 5x Flight Controller	C++
Sereolabs Zed 2 Camera	ROS Melodic
Jetson Xavier NX	Ubuntu 18.04

Real time flight tests have proven that the Jetson Xavier NX is capable of processing the required software systems at an excess of 30 Hz. The complete design of the quadcoptor is shown in figure 1.

The systems in table 1, when integrated, are able to produce a UAV capable of steady and controllable flight patterns. The system architecture and data exchange is presented in figure 2.

4. Database Generation

In part, the fundamental success of any machine learning architecture is derived from the data it is trained on. Size,

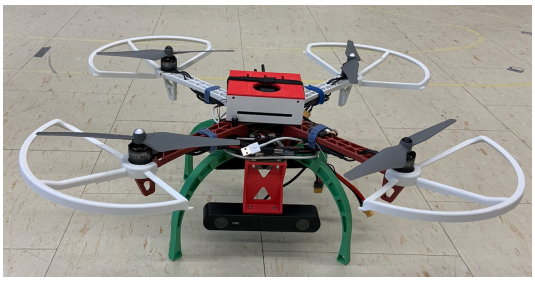

Figure 1. Prototype quadcoptor design

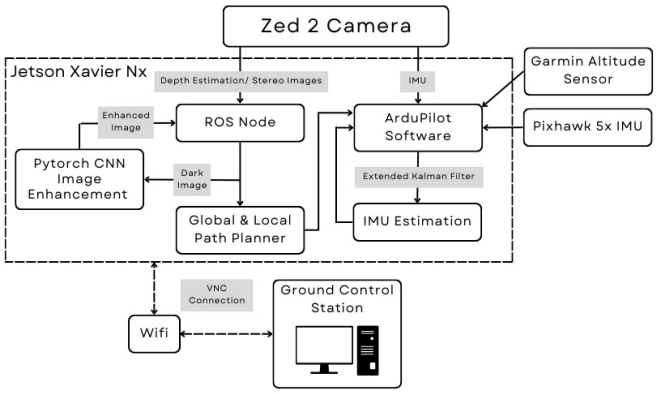

Figure 2. System architecture and data flow

diversity, and quality of the database directly impacts the accuracy of results [18]. There are many well established data sets such as the Oxford RobotCar data set [8, 7], KITTI Stereo 2015 data set [10], and SYNTHIA data set[13]. Each of these image banks provide a diverse range of image data. From stereo image pairs onboard cars to synthetic scenes all of these provide useful training options for image enhancement. When searching for a rich stereo-depth data set including both well lit and low-light environments we were unsuccessful. The majority of prepared data sets were for LiDAR based applications, therefore we elected to generate our own. The Zed 2 camera offers a variety of options including two modes for depth images. The first mode contains occlusions, suffers from temporal instability during motion, and provides less detail; but processes much faster. The second mode eliminates the performance issues of the first, but is unavailable on the chosen hardware. These modes and the images produced are referred to as high and low quality depth images in this report. We gathered 100 depth images in addition to 100 RGB stereo image pairs produced from the Zed 2 camera at low/high light and low/high quality. The low light and low quality RGB and depth images were trained on the ground truth data consisting of high light and high quality RGB and depth images. By generating our own data set using the available camera we are able to maintain the data types within the image pixels and image resolutions to ensure the performance of a trained CNN model could be retained during real time implementation.

5. Method

The goal of our proposed network is to enhance the quality of the raw images. Suffering from large amount of noise and occlusions, the stereo camera under performs in low lighting conditions.

We propose the use of a CNN to enhance the raw stereo RGB and depth images for use on a prototype UAV. Each image class is scaled differently. RGB images contain pixel values from 0-255 for each color channel. The depth images have a scale from 0.5m to 20m, containing a depth estimate for each pixel within these bounds. Each of these image classes were analyzed through their pixel arrays after they were normalized between 0 and 1.

For depth enhancement we attempted three different models, two of which were successful. The first approach aimed to use two CNN models for image enhancement. The first CNN was used to improve the quality of the stereo image pairs under low-light conditions while the second enhanced the depth map derived from the stereo pairs. However, it was found that after the stereo image enhancement the spatial relation between the two images was lost. As a result the methodology of extracting a depth map from stereo images was unable to render an accurate depth image. Therefore, this initial approach was abandoned.

The second approach is to directly process a low-light low-quality depth image using a CNN to improve it's quality. The ground truth is the corresponding high-light high-quality depth image. The architecture of this model is shown in figure 3. It consists of an encoder and a decoder. The encoder consists of two convolutional layers each followed by a batch norm layer and ReLU activation function. While not changing the spatial dimension, they increase the number of channels to 8, then 16. The decoder again consists of two convolutional layers followed by a batch norm layer and ReLU activation for the first and Sigmoid activation by the last layer. The Sigmoid activation function is used in order to bound the output between 0 and 1. The number of channels is decreased back to 8 and then 1 for the final output.

The third approach is to take the low-light depth and stereo images as inputs to improve depth. The architecture is shown in figure 4. It comprises two encoders and one decoder. The first encoder takes the low-light depth image as input. It consists of two convolution layers with 8 channels and each followed by a batch norm layer and ReLU activation function. The second encoder follows the same architecture, but takes a gray-scaled low-light stereo image as input. It has shared weights for both the left and right image. The three encoder outputs are then concatenated into one 3D array and fed into a decoder. The decoder's archi-

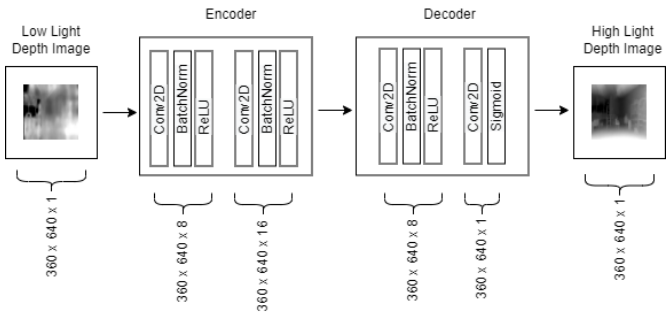

Figure 3. Architecture of the second model

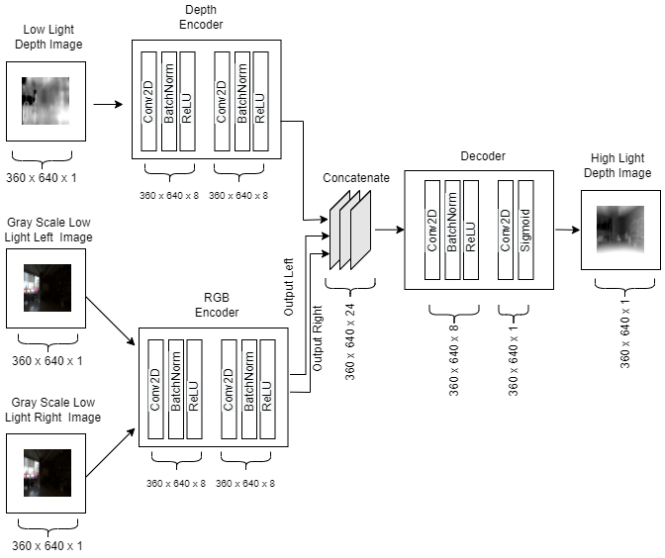

Figure 4. Architecture of the third model

tecture is the same as for the second model.

In order to train the models, mean squared error (MSE) loss is minimized by using Adam optimization. The data is split into two sections, 80% training and 20% validation.

6. Results

As described above, for the second approach we attempt to directly enhance the low light depth images while using the corresponding high light depth images as a ground truth. The predictions can be seen in figure 5. The output looks smoother with fewer occlusions and artifacts than the input images. Nevertheless, the model is not able to generate all the details as in the ground truth images. Figure 6 shows heat maps, where each pixel corresponds to the absolute error in meters between the two considered images. It can be observed that the error between the ground truth and the input or predicted image is very high with up to 7.7 meters, especially for parts of the room that are far away from the camera. Therefore, the reliability of the ground truth is further investigated in proceeding analyses.

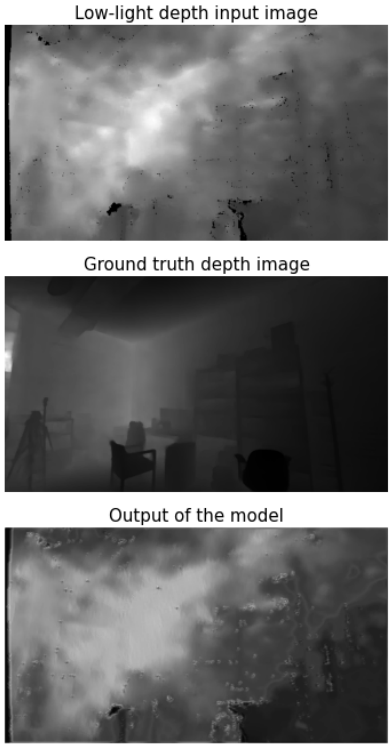

Figure 5. Input, ground truth and output of the second approach

The third approach takes the low-light depth images as well as the left and right low-light images as input and outputs the corresponding depth image. The results and heat maps can be seen in figures 7 and 8. Compared to the input depth image, the output of the model is much more detailed. However, by including the left and right images the depth is not preserved. For example, the predicted depth of the white board in the middle of the images in figure 7 are inaccurate up to 6.6 meters compared to the ground truth. Therefore, this approach is more useful for visualization purposes opposed to depth prediction.

Quantitatively, we compared our models depth accuracy to the input depth image and against the baseline of the ground truth. The depth image seen by the camera has a resolution of 640 x 360 pixels. This totals 230,400 pixels each with depth estimates. To calculate the depth of an object at specific distance we positioned a screen according to the estimates of the ground truth at distances ranging from 1m to 7m in increments of 0.5m. Knowing that this screen will not encapsulate the entire image seen by the camera we sampled a small 5x5 pixel window which we knew was positioned over the screen in the image. The average depth estimate of the 25 pixels over a total of 150 camera frames was recorded for comparison. This procedure can be seen in figure 9.

Performance of the models was assessed though their similarity to that of the ground truth estimates. From fig-

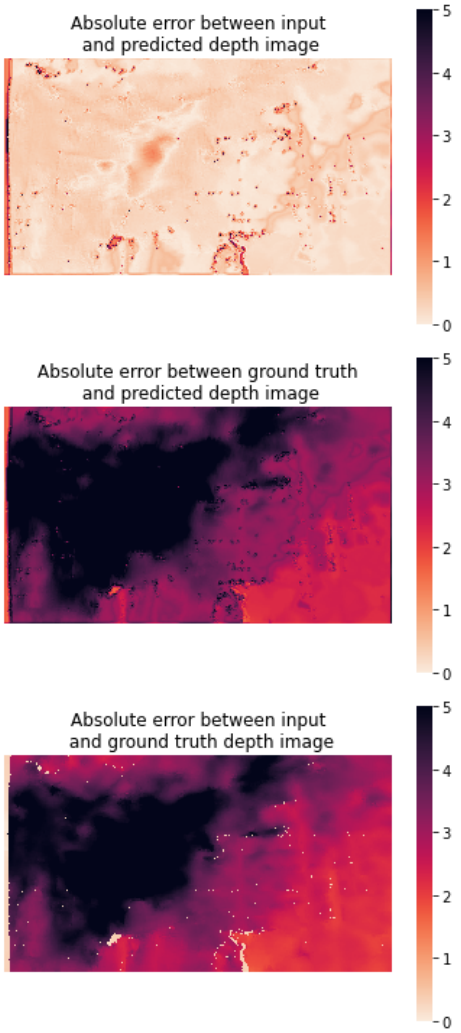

Figure 6. Absolute error between the input, ground truth and output depth images of the second approach

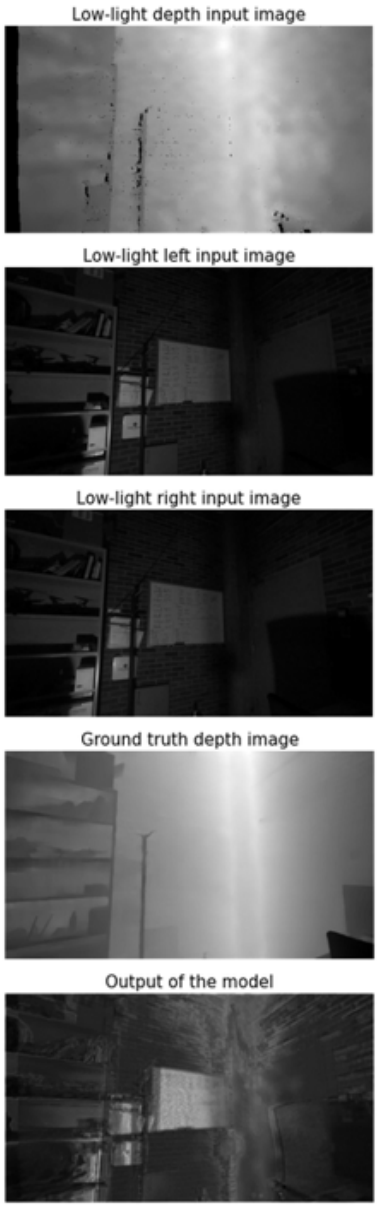

Figure 7. Inputs, ground truth and output of the third approach

ure 10 we see that the input on average outperforms our enhanced model by 4.01% at depth ranges from 1m to 5.5m. However, at distances greater than 5.5m we see our enhanced model outperform that of the input by an average of 6.38%. Due to the restrictions of the room size we were unable to capture images at distances greater than 7m to see if this trend continued at higher depth ranges.

As our model under-performed at lower distances we investigated the accuracy of the ground truth used in the model training. The results of this test are found in figure 11.

In this investigation the distance from the screen to the camera was physically measured from 1m to 7m in increments of 1m. Ideally, the measurements would be identical or within error ranges, however, it is observed that they are dissimilar and become more inaccurate as the distance increases. The average error in the distance from 1m to 7m is 19.8%. This result disproves the claim made by Stereolabs in which the manufacturer states the error in the depth estimate is to be 1% at low distances and will increase quadratically up to 9% at high depth ranges[1]. From figure 11 it is clear the depth estimates obtained from the ground truth were not within the stated error ranges. Therefore, the inaccuracy in the ground truth is the result of our models underperformance. This error was included in figure 10 as error bars to demonstrate the variability of acceptable depth ranges due to the inaccuracies of the ground truth.

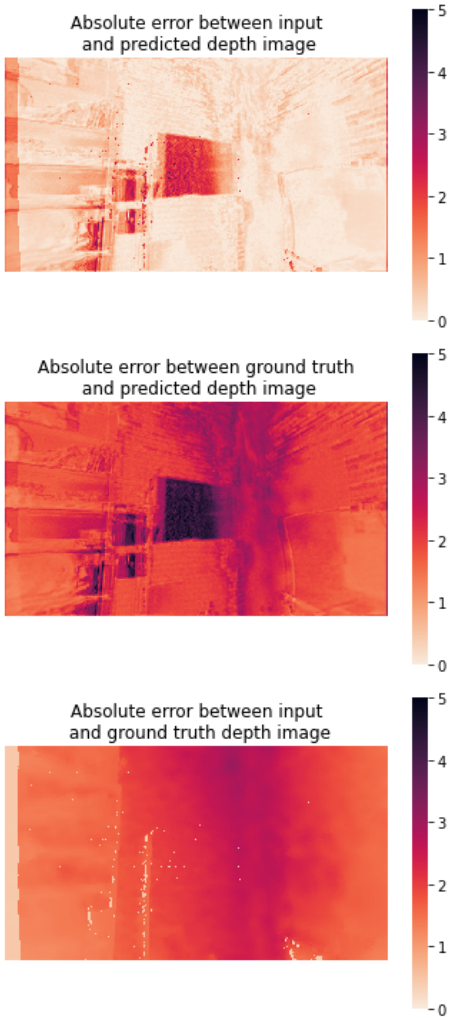

Figure 8. Absolute error between the input, ground truth and output depth images of the third approach

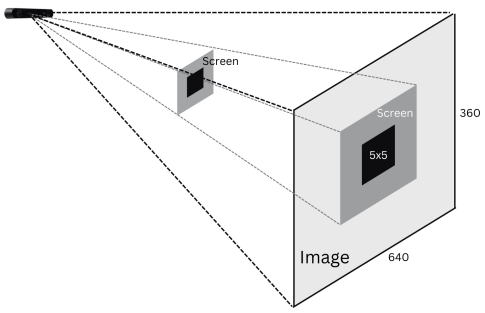

Figure 9. Setup to obtain the average depth estimate

7. Conclusion

Accurate depth perception of machines has become a necessary requirement in advanced robotics and automation. To achieve this there are a variety of sensors available.

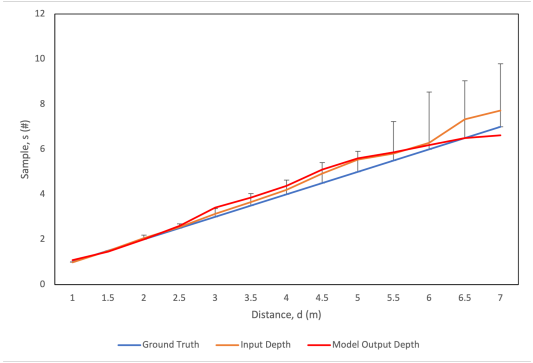

Figure 10. Average depth estimation of models

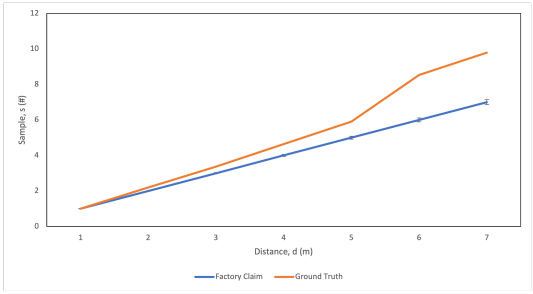

Figure 11. Ground truth depth estimation accuracy

However, the trade-off between price and quality proves to be a challenging reality. In this study we proposed the artificial enhancement of stereo depth estimation for use onboard a prototype UAV. This enhancement was achieved independently using two separate CNN models.

The first model was used to increase the depth estimation accuracy and the second to improve the depth image for visualization. The first model was able to effectively improve depth estimations by an average of 6.38% at distances greater than 5.5m. However, the accuracy was not improved at more conservative distances where the input outperformed our model by an average of 4.01%. Due to this inconsistent result we further investigated the ground truth used in the training of the CNN. It was found that although the ground truth image was free from occlusions, had more refined details, and improved temporal stability during movement, the depth estimates contained within the image suffered from an average inaccuracy of 19.8%.

The second model was able to effectively enhance the quality of the low light images. This refined image had more texture and details of the scene which were not obvious in the original images. However, this model did not preserve the depth estimates within each pixel.

Future work in this area should investigate an improved ground truth for the training of the CNN models. Additionally, increasing the size of the data set should be considered to improve the model's robustness in unknown environments. Including images in diverse settings at depth

ranges in and at the boundaries of the cameras operating range (0.5m to 20m) will ensure optimal performance in real-time implementation.

8. Contributions

Shane Allan: To accomplish the final results of this project I contributed in all areas. I helped to define the initial project scope and CNN model pipelines including relevant preliminary research, I wrote the python scripts to interface with the Zed 2 camera and extract the various images required for training. I researched and manipulated data types for data collection to match those which the UAV will receive. I implemented a time of flight laser rangefinder onboard the drone to improve altitude stabilization, I wrote the python script to communicate with the produced CNN, I wrote the python script for depth estimation in various pixel ranges for the quantitative results, I implemented all of the software onboard the prototype drone, I operated test flights and resulting data collection, I facilitated four separate dates for data collection, I contributed substantially to writing the mid-term project report and extensively to the final report.

Sophia Wagner: In order to realize this project, I was responsible for the development of the CNN models. I developed the different approaches, did research about possible architectures and wrote the python code for the CNN models. I tested different architectures and hyperparameters in order to find the best model. I wrote the python scripts for visualizing the results including error heatmaps. I contributed in writing the mid-term project report and the final report, mainly in terms of method description and results. Besides from that, I helped to define the initial scope of the project and did some preliminary research.

Almas Sahar: This Project provided me an opportunity to learn the practical implementation of theoretical concepts taught in class specifically machine learning models, stereo vision, depth estimation, and depth enhancement. I contributed to the literature review for the project, image collection, and report writing.

References

[1] Depth Settings — Stereolabs. 5

[2] Baris Baykant Alagoz. A Note on Depth Estimation from Stereo Imaging Systems. *Computer Science*, 1(1):8–13, 2016. 2

[3] Amit Bracha, Noam Rotstein, David Bensaïd, Ron Slossberg, and Ron Kimmel. Depth Refinement for Improved Stereo Reconstruction. 12 2021. 1, 2

[4] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning Optical Flow with Convolutional Networks. 4 2015. 2

[5] Hamid Laga, Laurent Valentin Jospin, Farid Boussaid, and Mohammed Bennamoun. A Survey on Deep Learning Techniques for Stereo-Based Depth Estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):1738–1764, 4 2022. 2

[6] Wenjie Luo, Alexander G. Schwing, and Raquel Urtasun. Efficient Deep Learning for Stereo Matching. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5695–5703. IEEE, 6 2016. 2

[7] Will Maddern, Geoffrey Pascoe, Matthew Gadd, Dan Barnes, Brian Yeomans, and Paul Newman. Real-time Kinematic Ground Truth for the Oxford RobotCar Dataset. 3

[8] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *International Journal of Robotics Research*, 36(1):3–15, 1 2017. 3

[9] Armin Masoumian, Hatem A. Rashwan, Julián Cristiano, M. Salman Asif, and Domenec Puig. Monocular Depth Estimation Using Deep Learning: A Review, 7 2022. 1

[10] Moritz Menze and Andreas Geiger. Object Scene Flow for Autonomous Vehicles. 3

[11] Alican Mertan, Damien Jade Duff, and Gozde Unal. Single image depth estimation: An overview, 4 2022. 1

[12] Kihong Park, Seungryong Kim, and Kwanghoon Sohn. High-precision depth estimation with the 3D LiDAR and stereo fusion. In *Proceedings - IEEE International Conference on Robotics and Automation*, pages 2156–2163. Institute of Electrical and Electronics Engineers Inc., 9 2018. 1

[13] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. 2016. 3

[14] Santiago Royo and Maria Ballesta-Garcia. An overview of lidar imaging systems for autonomous vehicles. *Applied Sciences (Switzerland)*, 9(19), 10 2019. 1

[15] Vladimir Tankovich, Christian Hane, Yinda Zhang, Adarsh Kowdle, Sean Fanello, and Sofien Bouaziz. HITNet: Hierarchical Iterative Tile Refinement Network for Real-time Stereo Matching. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14357–14367. IEEE, 6 2021. 2

[16] Paden Tomasello, Sammy Sidhu, Anting Shen, Matthew W. Moskewicz, Nobie Redmon, Gayatri Joshi, Romi Phadte, Paras Jain, and Forrest Iandola. DSCnet: Replicating Lidar Point Clouds with Deep Sensor Cloning. 11 2018. 1

[17] Jure Žbontar and Yann LeCun. Computing the Stereo Matching Cost with a Convolutional Neural Network. 9 2014. 2

[18] Wei Zheng, Jialiang Gao, Xiaoxue Wu, Fengyu Liu, Yuxing Xun, Guoliang Liu, and Xiang Chen. The impact factors on the performance of machine learning-based vulnerability detection: A comparative study. *Journal of Systems and Software*, 168:110659, 10 2020. 3