(#57) Enhanced Object Localization via Neural Network Integration with Newton's Method

T. N. Balasooriya[1], Ms. Hirushi Dilpriya[2]

[1]Faculty of Engineering, NSBM Green University, Sri Lanka, [2]Faculty of Computing, NSBM Green University, Sri Lanka

Abstract

Machine vision is crucial in autonomous vehicles and robotics, demanding precise object localization and environmental perception. Enhancing the baseline between cameras can improve accuracy and range but introduces challenges in stereo synchronization. This research presents a novel system that integrates neural networks with Newton's method to enhance object localization accuracy while reducing computational costs. Utilizing three cameras with known baselines, the system combines Convolutional Neural Network (CNN) models, such as YOLO, for critical component identification, with Newton's method for refining initial location estimates provided by machine learning algorithms. This integration ensures high accuracy by double-checking neural network outputs with mathematical models, avoiding local minima, and enhancing computational efficiency. Experimental results demonstrated successful camera calibration, accurate object detection, and significant improvements in point estimation accuracy using Newton's method. This approach promises advancements in 3D reconstruction and real-time object localization for autonomous systems.

Keywords: *Multiview Stereo (MVS), Stereo Matching, Object Localization, Newtonian Solvers*

Introduction

Machine vision plays a crucial role in autonomous vehicles and robotics, where precise object localization and environmental perception are vital. Increasing the baseline between cameras can enhance the accuracy and range of these systems [2], but it also introduces challenges related to stereo synchronization due to mechanical constraints. While modern research often relies on machine learning-based estimates [1], [3], [4], these approaches may lack the precision required in certain high-accuracy scenarios.

This research seeks to enhance accuracy without escalating computational costs. The proposed system employs three cameras with a

known baseline and integrates algebraic equations for localization with Newton's method to solve these equations. A neural network is used to generate the initial guesses required for Newton's solver. This approach continuously re-evaluates the output of the machine learning algorithm against the mathematical model, yielding significantly more accurate results. The neural network plays a critical role in avoiding local minima, while the Newtonian solver improves accuracy with minimal computational overhead. Additionally, the mathematical solver employed here has broad applicability to various complex mathematical challenges.

Related Work

Recent advancements in MVS include MVSNet, which uses deep learning to predict depth maps from multiple views, enhancing accuracy and efficiency [1]. AA-RMVSNet further improves this with adaptive aggregation and refinement strategies, achieving state-of-the-art 3D reconstruction [3]. Additionally, using a CNN model like YOLO for key component identification before stereo matching improves accuracy and reduces computational costs [4]. Employing CNNs early in the process enhances overall accuracy and workflow efficiency [5]. A neural network for computing Cartesian coordinates simplifies complex calculations and reduces computational load [6].

Method

As illustrated in Figure 1, the process begins by saving the feeds from all three cameras simultaneously to eliminate artifacts caused by time delays. Next, the cameras are calibrated using their intrinsic properties to remove lens distortions and warping effects. For critical component analysis, a Convolutional Neural Network (CNN) model, such as YOLO, is employed to identify essential objects in the images, such as vehicles, humans, and obstacles.

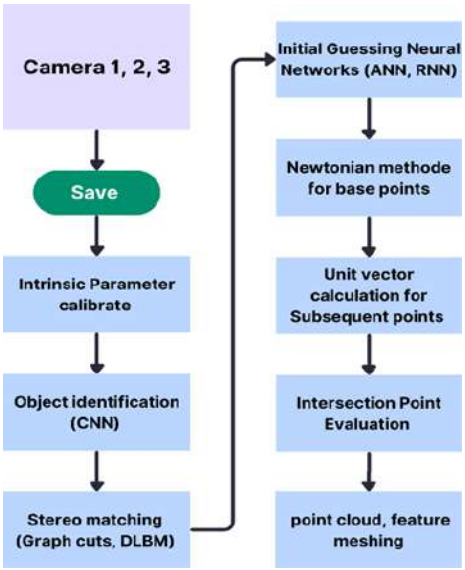


Figure 1: Flow chart

Once these critical components are identified, stereo matching is performed to determine stereo correspondence. Points on these critical components with high confidence in stereo matching are designated as base points (A, B, C). The angular disparity between these base points is then calculated, providing essential data for initial location estimates, which are generated using a machine learning algorithm such as an Artificial Neural Network (ANN) or a Recurrent Neural Network (RNN). To enhance the accuracy of these initial estimates, a Newtonian method is subsequently applied. After refinement, the mathematical model is applied specifically to the evaluation of the base points, as they carry the most critical information, with other points being assessed relative to these base points.

Subsequent points (P_p) are evaluated relative to the base points by calculating unit vectors relative to camera coordinates (O_i) using the angular disparity between the base points and the subsequent points (AO_iP_p). These unit vectors are then used to determine intersection points, thereby creating a point cloud. Finally, a mesh algorithm is applied to the point cloud to reduce noise in the measurements and improve the overall data quality.

To train the neural network for initial guessing, start by creating a virtual environment in a 3D Cartesian space, ensuring the workspace is sufficiently large to accommodate the X, Y, and Z range, which defines the system's field of range. This virtual workspace is then divided into X_R, Y_R, and Z_R segments, forming a point cloud, where the resolution (X_R * Y_R * Z_R) can

be adjusted based on application requirements. Cameras are placed outside the point cloud, and angular disparities are calculated from 4 or more randomly selected points to each camera location. For the training process, camera locations and angular disparity data are fed to the neural network as input, while the Cartesian coordinates of the selected points are provided as output.

Results

The intrinsic camera calibration process was successfully completed, demonstrating a high level of accuracy in removing lens distortions and warping effects. Critical object identification using the YOLO CNN model was also achieved, enabling the system to accurately detect and classify various objects such as cars, buses, humans, and animals with high precision.

θ, β, Υ = *Mesured angles using camera view* (AOB, AOC, BOC)

A, B, C = *Initial guess for the base points*

O_i = *Camera location*

$AO_i = (X_{O_i} - X_A, \quad Y_{O_i} - Y_A, \quad Z_{O_i} - Z_A)$

$BO_i = (X_{O_i} - X_B, \quad Y_{O_i} - Y_B, \quad Z_{O_i} - Z_B)$

$AO_i = (X_{O_i} - X_C, \quad Y_{O_i} - Y_C, \quad Z_{O_i} - Z_C)$

i = *Selected camera* $[1, 2, 3]$

$$0 = \theta - \arccos\left(\frac{AO_i \cdot BO_i}{|AO_i| \, X \, |BO_i|}\right)$$

$$0 = \beta - \arccos\left(\frac{AO_i \cdot CO_i}{|AO_i| \, X \, |CO_i|}\right)$$

$$0 = \Upsilon - \arccos\left(\frac{BO_i \cdot CO_i}{|BO_i| \, X \, |CO_i|}\right)$$

Figure 2: Equations for solver

The Newtonian method, developed using the SciPy library, was rigorously tested and found to significantly improve the accuracy of point estimations, falling well within acceptable error margins. The equations for the Newton's method are provided in Figure 2. The procedures for calculating unit vectors and evaluating intersection points were successfully implemented and rigorously tested within a Python environment. These equations detail the specific calculations used for determining unit vectors as shown in Figure 3.

$$P_p = \begin{pmatrix} Pix_1 & Pix_2 & \cdots & Pix_n \\ Piy_1 & Piy_2 & \cdots & Piy_n \\ Piz_1 & Piz_2 & \cdots & Piz_n \end{pmatrix} = \begin{pmatrix} A_{xi} & A_{Yi} & A_{Zi} \\ B_{xi} & B_{Yi} & B_{Zi} \\ C_{xi} & C_{Yi} & C_{Zi} \end{pmatrix}^{-1} \cdot \begin{pmatrix} \cos(AO_iP_1) & \cos(AO_iP_2) & \cdots & \cos(AO_iP_n) \\ \cos(BO_iP_1) & \cos(BO_iP_2) & \cdots & \cos(BO_iP_n) \\ \cos(CO_iP_1) & \cos(CO_iP_2) & \cdots & \cos(CO_iP_n) \end{pmatrix}$$

P_p = *Subsequent points*, $P \in [1, 2 \ldots\ldots n]$

Pix_n = Unit vector of point P_n seen from O_i – X component

A_{xi} = Unit vector of point A seen from O_i – X component

AO_iP_n = *Angle between* AO_i *and* P_nO_i

Figure 3: Unit vector calculation

The virtual point cloud has been generated as shown in Figure 4 with a 120-degree field of view and a range extending to 250 times the baseline between the cameras. Within this region, 100,000 points have been uniformly distributed. From this point cloud, four points are randomly selected and processed by another algorithm as shown in figure 05 to evaluate the angles between them (AOiB, AOiC, AOiD, BOiC, BOiD, COiD). These are four base points and 18 relative angles between them (with each image frame generating six angles across three frames). The resulting data, comprising 25,000 data points generated from the above-mentioned method, is then stored in a file for use in the neural network training process.

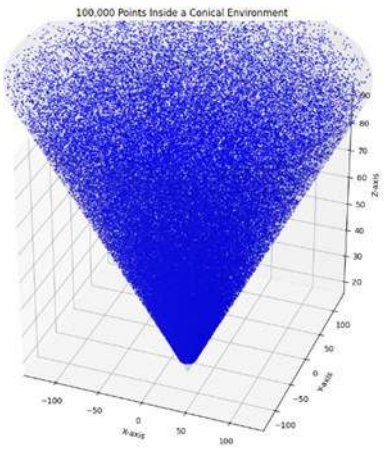


Figure 4: virtual point cloud

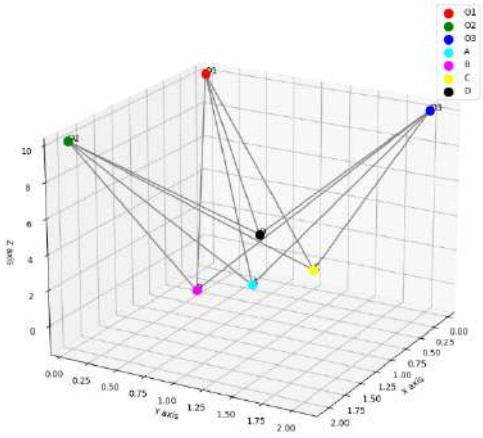


Figure 5: virtual point cloud

Both RNN and ANN neural networks have been developed and rigorously tested. However, despite thorough evaluation, the LSTM neural network is demonstrating the highest learning potential, even though it is currently experiencing significant underfitting, as demonstrated in Figure 6.

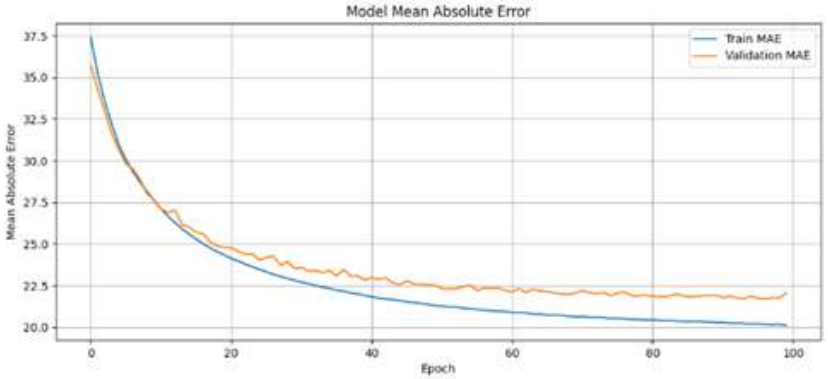


Figure 6: LSTM - Mean Absolute Error

Future Work

Future work will focus on several key areas to enhance system accuracy without increasing computational costs. This includes testing various stereo meshing algorithms and refining them to reduce noise and improve data extraction. Additionally, the initial guessing neural network will undergo further evaluation, with different architectures being tested to determine the optimal accuracy. Keras Tuner and hyperparameter tuning will also be incorporated into the system to fine-tune the neural network for better performance. Exploring the implementation of a feed-forward algorithm is suggested. Since the CNN algorithm identifies critical components, it can also be used for algorithm optimization. Implementing a feed-forward loop could create a more intelligent system capable of adapting to different environments, thereby optimizing performance for each specific scenario.

References

[1] Yao, Y., Luo, Z., Li, S., Fang, T., & Quan, L. (2018). MVSNet: Depth Inference for Unstructured Multi-view Stereo. *Proceedings of the European Conference on Computer Vision (ECCV)*, 767-783. DOI: 10.1007/978-3-030-01249-6_46.

[2] Ibrahimli, F., Köhler, J., Thormählen, T., & Mayer, C. (2023). Depth Discontinuity Learning Improves Accuracy of Multi-View Stereo Networks. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 1234-1243. DOI: 10.1109/ICCV.2023.00129.

[3] Wei, Y., Chen, Y., Yan, J., Liao, X., Xu, J., & Yang, Y. (2020). AA-RMVSNet: Adaptive Aggregation Recurrent Multi-view Stereo Network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9699-9708. DOI: 10.1109/CVPR42600.2020.00972.

[4] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788. DOI: 10.1109/CVPR.2016.91.

[5] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149. DOI: 10.1109/TPAMI.2016.2577031.

[6] Kendall, A., Grimes, M., & Cipolla, R. (2015). PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2938-2946. DOI: 10.1109/ICCV.2015.336.