Monocular Depth Estimation Via Multi-Layer Perceptron and Camera Calibration

Rohan Wadhwa
Email: rohanw12334@gmail.com

***Abstract*—Abstract - Monocular Depth Estimation aims to determine the depth and dimensions of an object solely based on a single image, cutting hardware costs and complexity while enhancing the ease of deployment(Zhang). This project aims to address monocular depth estimation by proposing two different methodologies to tackle object dimension estimation. The first methodology proposed the aim of using a custom calibration function to determine deterministic variables, fixing the non-linearity at close distances and verifying the accuracy of the pdp value, regardless of camera altitude(Enhanced Single-Camera). A Machine Learning Linear Regression model was utilized to compute the variables. The second methodology involved a custom built hybrid pipeline, adapting the calibration function with a MLP model, with a similar aim of tacking non-linearity but using Machine Learning for predicting the object's length. Similar to the first methodology, the calculations to determine the function's variables was done using a Machine Learning Linear Regression model.**

I. Introduction

As the dynamic landscape of medical robotics research continues to advance, with the emergence of more robots in medical assessments, the need for software capable of aiding such assessments becomes a necessity. In colonoscopy, the ability for small cameras to accurately capture the depth and dimension of an object continues to be impeded due to the lack of accounting for various factors. These factors are the non-linear relationship between the camera and the object at close distances, the camera lens distortion, and the presence of reflective surfaces. To account for these factors, we propose the use of a calibration function model and a Multi-Layer Perceptron(MLP) model to tackle these factors and limitation. The calibration function methodology was compared against the use of a Multi-Layer Perceptron(MLP) model to evaluate the performance of each to determine which was the most effective in tackling this issue.

II. Background and Related Works

The main foundation from which Monocular Depth Estimation is derived is the pinhole camera projection model. The pinhole camera projection model is a model that describes how three-dimensional points, based on a world coordinate system that accounts for all three axes, are projected onto a two-dimensional image(Nayer, 2025). The projection of those three-dimensional points on an image by every camera is different due to the camera's distinct intrinsic parameters, the projection of points in a 3D scene to a 2D image, and extrinsic parameters, the point and orientation of the camera(Nayar, 2025).

Monocular Depth Estimation aims to determine the depth of an object solely based on a single image, cutting hardware costs and complexity while enhancing the ease of deployment(Zhang). Monocular Depth Estimation has evolved over time from geometric based methods to Machine Learning based methods due to the emergence of Artificial Intelligence(AI). Previous non-AI traditional methodologies had to rely on various hardware systems which are parallax imaging, binocular camera systems, and stereo visions to obtain the depth information(Zhang). These methods were categorized into one of two categories: active methods and passive methods(Khan et al). Active methods involved the interaction of the object and environment with the aim of computing depth(Khan et al). For example, sensors that captured light were a viable method to determine depth(Zhang). Microsoft's Kinect v1 utilized structured light to map a predefined pattern onto a 3D scene(Zhang). The visible deformations in the light-based map allowed computation of the object's depth(Zhang). Passive methods were methods that involved computational image processing to compute depth(Khan et al). For instance, stereo vision methods involved the use of dual camera systems to capture the images from various angles, thus the matched corresponding pixels in all images helped compute depth(Zhang).

III. Application of Camera Calibration to Acquire Intrinsic Parameters

In order to comprehend the relationship between the three dimensional coordinates and their corresponding two dimensional coordinates, the camera's intrinsic and extrinsic parameters must be extracted via camera calibration(Nayer, 2025). In this project the process of applying camera calibration involved the usage of a paper containing a two dimensional image of a checkerboard and the use of inbuilt OpenCV functions. The checkerboard served as the image plane of the pinhole camera projection model for which its corresponding two and three dimensional points were extracted using various inbuilt functions that identified the corners of the chessboard and the distance between them in pixels to approximate pixel coordinate pairs. The primary aim of using camera calibration for this project was to specifically obtain one important intrinsic parameter that would be utilized for later mathematical use, the camera's horizontal focal length. The horizontal and vertical focal lengths of a camera is the distance of the camera's sensor from its lens.

IV. Methodologies Overview

The first methodology proposed the aim of using a custom calibration function to determine deterministic variables, fixing the non-linearity at close distances and verifying the accuracy of the pdp value, regardless of camera altitude(Enhanced Single-Camera). A Machine Learning Linear Regression model was utilized to compute the variables. In general, the process to utilize this function is by first determining a baseline pixel distance ratio parameter. This baseline can serve as a reference point that reinforces the computation of the unknown variables, a_1 and a_2 based on a known z height and $P_{DP}(z)$ value. The specific details on the function and computations would be elaborated further in later sections. The second methodology involved a custom built hybrid pipeline, adapting the calibration function with a MLP model, with a similar aim of tacking non-linearity but using Machine Learning for predicting the object's length. The MLP model will be used specifically to predict the object's length. Similar to the first methodology, the calculations to determine the function's variables was done using a Machine Learning Linear Regression model.

V. PDP(Pixel Distance Sampling Ratio Parameter) Concept and Application

Previous applications have proposed the utilization of pixel ratio parameters(PDP) for conversion of pixel values to real world measurement values. The main issue with previous applications is that they did not account for actions for instance, camera lens distortion, reflective surfaces and the non-linearity between the camera and object at close distances. In this project, we started with obtaining a baseline pixel distance ratio parameter P_{DP0}. We tackle the camera len's distortion, an extraneous variable, by removing it before obtaining the P_{DP0} value. This removal process was done using a python function and resulted in it not being accounted for in P_{DP0} value we obtained. In order to obtain the P_{DP0} value we used the equation we had to obtain the R value(Enhanced Single-Camera) and rewrote it to obtain the P_{DP0} value. This equation utilized the following variables which we had already obtained during the camera calibration process with the checkerboard: the camera's focal length, the digital pixel distance from camera to object, the K constant value, and the height of the camera.

This is the formula used to obtain PDP0:

$$R = K \times d_t \times d_{pix} \times P_{DP0} \tag{1}$$

$$P_{DP0} = \frac{R}{K \times d_t \times d_{pix}} \tag{2}$$

Where:

- R is the length of the object
- K is a constant
- d_t is the distance from the object to the camera
- P_{DP0} is the PDP baseline value at a specific baseline altitude
- d_{pix} is the distance between each pixel coordinate, obtained using the checkerboard in camera calibration

The camera X was used to take a photo of the sample OOI used in the project, a cube, at a height of 180 mm.

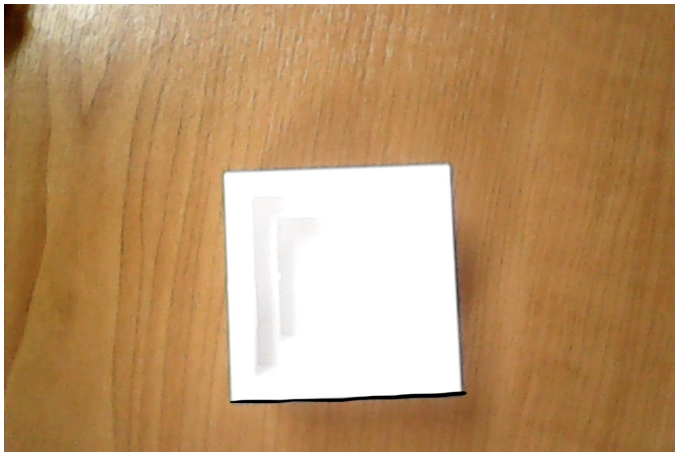

Fig. 1. *image of a cube, the OOI in this project, with a dark line drawn for ML detection of the object's length*

The R value of the cube was measured using a ruler, it was 50 mm. Substituting R for that value and 180 mm for dt, the PDP0 value was able to be obtained which was 0.262253 mm.

VI. Calibration Function Concept and Application

The primary reason for the use of the calibration function is to address the non-linearity between the camera and the object at close distances, specifically between 2mm and 30 mm(Enhanced Single Camera). The model we propose is:

$$P_{DP}(z) = P_{DP0} + \frac{a_1}{z} + \frac{a_2}{z^2} \tag{3}$$

Where:

- $P_{DP}(Z)$ is the P_{DP} value at altitude z
- P_{DP0} is the the P_{DP} baseline value at a specific baseline altitude
- a_1 is a coefficient that captures the non-linearity between the camera and an object
- a_2 is a coefficient that captures the non-linearity between the camera and an object

In order to determine the a_1 and a_2 values, the first methodology simply uses the calibration function by plugging in the known values: the $P_{DP}(Z)$ at known altitude z, the baseline P_{DP0} which was obtained earlier, and altitude z.

VII. Multi-Layer Perceptron (MLP) Model Concept

Multi-Layer Perceptron models also called feed forward models are Machine Learning models that are used to represent non-linear functional mappings between a set of input variables and output variables(Bishop). In order to represent these non-linear functional mappings, non linear functions of multiple variables are composed into a single non-linear

function called an activation function (Bishop). For MLPs with differentiable activation functions, a technique called back propagation is used to iteratively change the weights and biases in order to improve the model's prediction accuracy(Bishop).

VIII. Multi-Layer Perceptron (MLP) Model Application

In this project the Multi-Layer Perceptron(MLP) model was used to predict the length of the object which replaced the R value which allowed a new set of PDP values to be acquired based on specific altitudes. A MLP can be used because as mentioned in the previous section, MLPs are great at representing non-linear functional mappings between inputs and outputs and thus making a final prediction of the length of an object, the main application of using a MLP for this project.

IX. The Use of a Linear Regression Machine Learning Model

A Linear Regression ML model was used to obtain the a_1 and a_2 values for a variety of altitudes. This model was utilized in both methodologies due to complexity of manual calculations which would be inefficient due to factors such as too much time used and high likelihood of human error during calculations. The reason why linear regression works is because the calibration function is a quadratic function where $\frac{1}{z}$, $\frac{1}{z^2}$, and P_{DP0} is used to represent x, x^2, and c of the standard quadratic function $y = ax^2 + bx + c$.

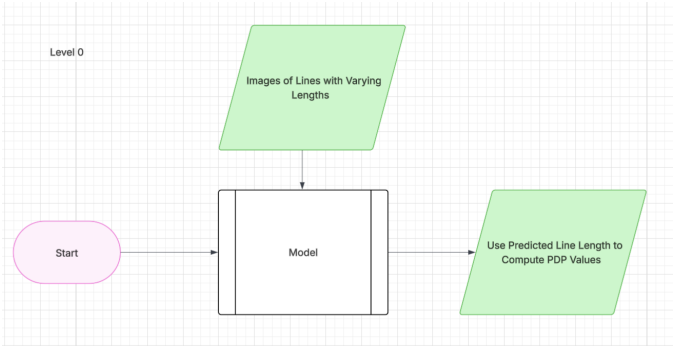

Fig. 2. *This shows a surface level representation of the Machine Learning Model's implementation*

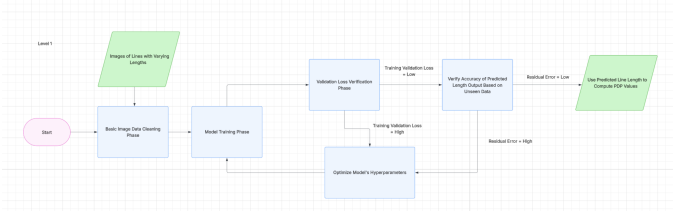

Fig. 3. *This shows a more complex representation of the Machine Learning Model's implementation*

X. Specular Reflections Removal

One factor that must be handled, before presenting the final object's HSV image, which would showcase the intensities, is specular reflections. When the images of the object were taken, the light that bounced back to the camera's lens caused various bright areas to appear. When converting the image from RGB to HSV and indicating the various regions of high/low intensities, based on color red and blue, the bright areas interfere with this process. Thus one way to eliminate this issue is by removing the bright areas which is done by using various OpenCV morphological functions, specifically image dilation and erosion.

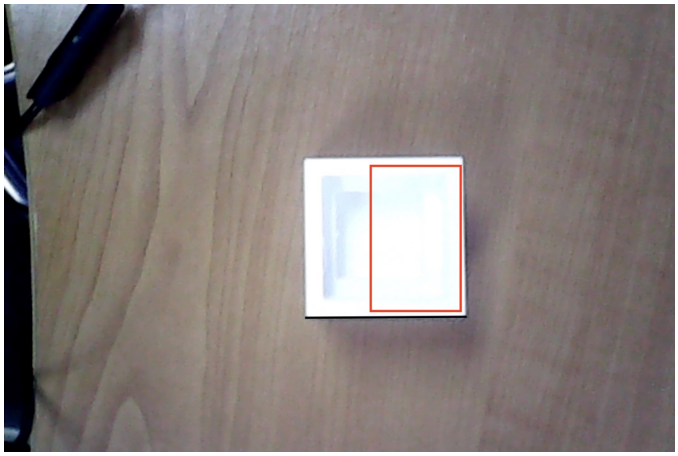

Fig. 4. *The red box indicates the region of the object that is very bright due to specular reflections covering the object's smaller inner lines*

As the image shows, this is the area where the specular reflections affect the visibility of the object's inner lines. Without reducing the brightness of this area, determining the inference of the object's regions of intensity with depth becomes challenging.

XI. Experimental Environment

A. Hardware Specs

The hardware utilized in this project was minimal, to ensure easy reproducibility of the setup and results. The peripherals used were

B. Software Specs

The software utilized in this project was Visual Studio Code, the IDE in which the modules were written in and executed, and OpenCV, the Computer Vision framework to implement the camera calibration along with other Computer Vision related processes. Visual Studio Code was used due to its speed in executing modules at a rate that ensures convenience with results being easily reproducible. OpenCV is a framework that was used due to its power and simplicity. The user is able to tackle camera calibration without having to write overly complex code to execute the same process. Similarly to Visual Studio Code, less time is utilized and thus this framework ensures that results are easily reproducible.

XII. Experimental Setup

A. ML Data Collection

For the Machine Learning methodology process, the data collected were images of lines, with differing lengths, on pieces of paper. A ruler was used to draw linear horizontal lines with a dark blue pen. It was important for the images to be taken with varying light visibility so that the model would easily be able to better generalize the prediction of the lines' lengths. In addition, the images were taken from varying heights and angles to also reinforce the model generalization process. One primary constraint that had to be accounted for was on the maximum possible length that the model was capable of determining. In order to account for this, a decision was made for the maximum possible line length, which the model was trained to predict, to be 200 mm. It's accuracy would significantly drop if a line's length exceeded 200 mm.

B. Training and Validation Method

Training and validation were divided, with 80 percent of the data allocated for training and the remaining 20 percent for validation. This split ensured that most of the data were being prioritized for training the model to learn the overall pattern of lines on pieces of paper with varying lengths that were provided from various angles and light intensities.

The batch size was adjusted to prevent overfitting and underfitting. The batch size could not be too small or else it would cause the model to overfit and it could not be too large or else underfitting would occur. A mechanism that was implemented to prevent overfitting was a stopping early mechanism. This mechanism stopped the training if the loss function dropped below 1 percent. This ensured the model was not too complex, which would cause it to have low bias in training but high variance in validation due to the absorption of noise in the data.

C. Performance Metrics

One metric that was used to determine the performance of the Machine Learning training process was the loss function. The loss function indicates the margin of error between the true value and the predicted value. A low loss function represents a minimization in the error margin between the true value and the predicted value. As the MLP model trained the loss function started to decrement significantly. The decrement of this function signaled that the weights of each node were iteratively updated as the gradient descent algorithm minimized the loss function by approaching the minimum. If the loss function was too low it would indicate negative performance of the MLP model due to the minimum being overshot. The overshooting of the minimum would represent the MLP model having over-corrected weights thus causing it to likely predict the value to be higher than it actually is.

XIII. Results and Evaluation of Methodologies

Fig. 5. *image of a cube, the OOI in this project, with a dark line drawn for ML detection of the object's length*

These were the results using the Machine Learning model:

Predicted R Value: 48.12 mm

a_1 Value: 45.427631

a_2 Value: 0.00000

The results of using just the calibration function where the R value was the known length of the cube:

R Value: 50.00 mm

a_1 Value: 47.205616

a_2 Value: 0.000067

The results showed that the Machine Learning Methodology performed significantly worse compared to the calibration function methodology. The Machine Learning model had predicted the R value, length of the cube, to be approximately 77 mm which indicates that there was a 27 percentage error margin since the actual length was 50 mm.

XIV. Discussion

A. Clinical Applications and Implications

As mentioned before, the main clinical application that this can be used for is in a colonoscopy where a camera would be inserted into the colon[4]. When using the camera, the depth of the colon must be ascertained, thus having a system that can determine the width of the colon and the depth of the camera is a critical(Enhanced Single-Camera). The use of a single camera is also convenient as it enhances the efficiency of the

depth estimation process due to the convenience of using just one camera. The efficiency of this process is also highlighted by the use of Machine Learning to determine the length of an object. In colonoscopies the camera must be able to determine the length of the colon and this ML Computer Vision pipeline provides this capability.

B. Contributions and Innovations

The main technical contribution is the development of the end to end Computer Vision Machine Learning pipeline that allows for the input of object images, annotated so the length is visible, and the output of a predicted length of the object. This contribution provides another methodology in the process of depth estimation, since the length information can be directly used in the calibration function to verify accurate camera calibration regardless of the camera's distance.

An innovation that has contributed significantly to this project is the P_{DP0} equation(Enhanced Single-Camera). This was very useful in not only obtaining the P_{DP0} value which was used in the calibration function but it also allowed various factors, specifically lens distortion and the non-linearity between the object and camera, to be accounted for(Enhanced Single-Camera). Using a P_{DP0} value in the calibration function was necessary because it ensured that there was a baseline PDP value based on a specific height, 180 mm. Knowing the P_{DP0} value when solving for the a1 and a2 values in the calibration not only makes it possible to solve those constants but it also serves as a reference value that makes it possible to determine whether the other pdp values are similar or whether they significantly deviate from the norm(Enhanced Single-Camera). From the baseline pdp0 value we know what the a1 and a2 values are supposed to be regardless of the pdp(z) value(Enhanced Single-Camera).

C. Challenges and Limitations

During the execution stage of this project, there were many challenges and limitations. One challenge we faced was the lack of existing training data to train the MLP model. In order to tackle this challenge, data augmentation was utilized along with duplicating existing images in order to have a diverse and sufficiently large enough dataset for accurate AI predictions on an object's length. A limitation that the AI has is its functional incapability in detecting the length of objects that are curved or spherical in dimensions. This limitation makes this AI model suitable for only predicting the lengths of objects that have linear-shaped edges.

XV. Conclusion and Future Works

In conclusion, this paper provided a comparison between two methodologies both tackling monocular depth estimation, specifically using their own processes. The methodology primarily involving the calibration function proved to be much more accurate in providing reliable a_1 and a_2 values in comparison to the Machine learning methodology that had a significant error margin in predicting the object's length. Due to the Machine Learning model's error being significant future works would aim to reduce the residual error by altering several parts of the methodology. For instance, instead of using a MLP model, a Convolutional Neural Network(CNN) can be used due to its ability to capture complex patterns in any image. This would make the Machine Learning model more reliable in complex medical scenarios such as using it in a colonoscopy, where the environment is much more complex due to low visibility and complex surfaces. This makes it harder for the camera to detect the dimensions of the colon. Also the current MLP model that was implemented only worked well when using objects with predictable dimensions like a cube. An object that is spherical or lacks proper edges would cause the model to fail in accurately predicting the length. The CNN model would be able to identify objects that lack edges due to the model's use of convolutional layers.

XVI. References

Zhang, Jiuling. "Survey on Monocular Metric Depth Estimation." Survey on Monocular Metric Depth Estimation, 27 Mar. 2025, arxiv.org/html/2501.11841v2.

Khan, Faisal, et al. "Deep Learning-Based Monocular Depth Estimation Methods-a State-of-the-Art Review." MDPI, Multidisciplinary Digital Publishing Institute, 16 Apr. 2020, www.mdpi.com/1424-8220/20/8/2272.

Enhanced Single-Camera Depth Estimation for Medical Endoscopy: Calibration Method Accounting for Specular Reflections

Nayar, Shree K. "Image Formation." Image Formation - Columbia CAVE, 15 Feb. 2022, cave.cs.columbia.edu/Statics/monographs/Image

Bishop, Christopher M. Neural Networks for Pattern Recognition, 1995, people.sabanciuniv.edu/berrin/cs512/lectures/Book-Bishop-Neural