VERIFICATION OF STEREO CAMERA CALIBRATION USING LASER STRIPERS FOR A LUNAR MICRO-ROVER

Virtual Conference 19-23 October 2020

Srinivasan Vijayarangan and David Wettergreen

The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
E-mail: srinivasan@cmu.edu

ABSTRACT

Despite advancements in LiDAR technology, stereo cameras are still preferred for space applications [1], due to heritage and advantages in weight, power, and complexity of moving parts. The accuracy of stereo vision systems is affected by their calibration and even a small change in the position or orientation, for example from vibration or thermal expansion, can cause significant errors in feature matching or distance estimation. These errors can lead to the failure to identify obstacles and potentially the mission. Elaborate effort is taken to calibrate these systems under various settings before launch [3]. It is essential to ensure that the calibration parameters still hold after deployment on a planetary surface. However this is not an easy task due to the procedures and fixtures needed in the process. In this work, we develop a method to verify the stereo calibration *in situ* using associated stereo pixels produced by a line striping laser. We also show the sensitivity of the pixel associations to the estimated measurements in simulation.

1 INTRODUCTION

Bell,et.al., [3] detail the meticulous effort taken to calibrate the sensors both pre-flight and in-flight. These include characterizing the sensors for different environmental changes, as the calibration done for one environment may not hold for another. Various factors like vibration during launch, structural stress during landing, thermal expansion during operation can contribute to this issue. Therefore it becomes essential to develop procedures to enable calibration in the field. To calibrate stereo extrinsic parameters an object with known precise dimensions can be used. This could be the lander itself or any other payload on the lander. This becomes challenging if the stereo cameras are not articulated, as is the case for micro-rovers, and not facing the lander, as is the case for navigation cameras. For these new breed of micro-rovers designed to operate for brief mission, a few earth days time is another concern. A real-time calibration solution is preferred. We propose a calibration procedure which estimates the stereo extrinsic parameters in real-time *in situ*. We propose using laser line stripers to solve this problem. A laser line striper is a laser diode with a Powell lens[4] which projects a light plane. The light plane intersects with the ground surface and produces lines of points which are observed by cameras. Laser line stripers were used by Sojourner rover [3] for obstacle avoidance and have space heritage.

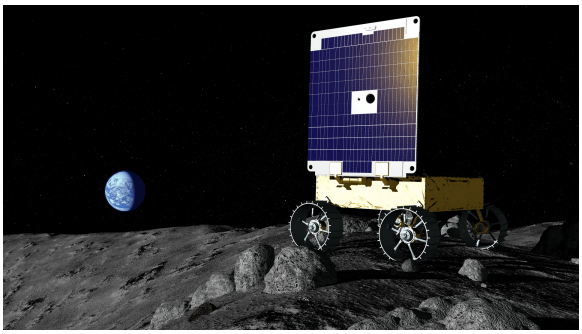

Figure 1: Artistic rendering of MoonRanger

We use a light plane to verify calibration which is similar to the traditional checkerboard method. When a checkerboard is used for calibration, the planar nature of the checkerboard is exploited to solve for the camera intrinsic parameters[2]. The same idea can be extended for the light plane. The major difference is: in the checkerboard method the position of the corners is known precisely. However using the light plane the corners are not known as no assumptions about the terrain can be made.

In this paper we look into aspects involved in using the line striping laser to verify the calibration of stereo cameras. It includes sensitivity analysis to quantify our results. We perform these experiments in simulation as

- we can obtain absolute ground truth to quantify our results
- simulation is unaffected by the accuracy of the pixel extraction or association algorithms which operate on image domain
- experiments are repeatable

We do recognize the importance and difficulty in evaluating real data. The efficiency of extraction and as-

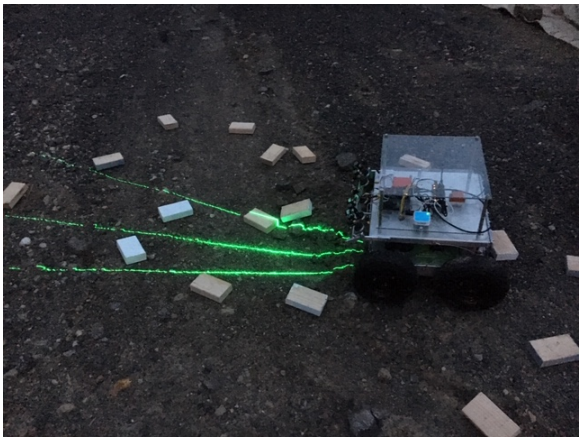

Figure 2: Prototype rover with laser strips and stereo cameras

sociation of the pixels in real data is a hard problem which needs separate investigation. However that should not impede understanding the efficacy of using line striping for verifying calibration.

There are two main contributions in this paper.

- Formulated a procedure to verify stereo calibration using pixel associations from a line striping laser
- Conducted sensitivity analysis experiments to study how changes in pixel error contributes to the estimated poses.

This technology is enabling for autonomous micro-rover exploration. It will be integral to the MoonRanger rover [Figure 1 and 2] which has been selected as a Lunar Surface Instrument and Technology Payload (LSITP). It will fly aboard the Masten XL-1 lander on a Commercial Lunar Payload Services (CLPS) mission to the pole of the moon in December 2022.

2 PROBLEM FORMULATION

Given: A stereo camera setup and a line striping laser. The cameras are pre-calibrated and the intrinsic and the extrinsic parameters are known. The intrinsics involve the camera matrix K and the distortion parameters D . The extrinsics involve the position of the cameras in some known world frame. The line striping laser is also calibrated and the laser plane is known in same world frame. The line strip is be observed on both the camera images. We have a reliable algorithm to extract the line striping points on both the images and associate the corresponding pixels.

Problem: Verify if the camera pose has changed with respect to prior calibration?

Formulation: Let the two cameras be c_1 and c_2. The intrinsic matrices of the cameras K_{c_1} and K_{c_2} are given by

$$K_{c_i} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f_x and f_y are the focal lengths and c_x and c_y are the camera centers in pixel coordinates. If R_{c_1}, R_{c_2} and T_{c_1}, T_{c_2} denote the rotation and translation components of the two cameras in an fixed world frame, then the projection of a 3D point in the world frame is given by

$$\begin{bmatrix} u_{c_1} \\ v_{c_1} \\ 1 \end{bmatrix} = K_{c_1} \left[R_{c_1} | T_{c_1}\right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} u_{c_2} \\ v_{c_2} \\ 1 \end{bmatrix} = K_{c_2} \left[R_{c_2} | T_{c_2}\right] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

where $\left[u_{c_1}, v_{c_1}\right]^T$ are the pixel locations of 3D point $\begin{bmatrix} X & Y & Z \end{bmatrix}^T$ for camera c_1.

Let the laser plane be represented using the parameters a, b, c, d . If the point $\begin{bmatrix} X & Y & Z \end{bmatrix}^T$ passes through the plane then it satisfies the equation,

$$aX + bY + cZ + d = 0$$

So formally, given pixel correspondences $\left[u^i_{c_1}, v^i_{c_1}\right]^T \Leftrightarrow \left[u^i_{c_2}, v^i_{c_2}\right]^T$ between images from both the cameras, can we verify that $[R_{c_2} | T_{c_2}]$ has not changed.

3 METHOD

The first step is to estimate the 3D position of the pixels on both the images. This can be done using a ray tracing algorithm[5]. The ray tracing algorithm works by projecting a ray emanating from the camera center going through the pixel coordinate to the world. The point of intersection of this ray to the laser plane gives the 3D position of the pixel. Using the estimated 3D points from one camera and the pixel locations from the other camera, we can use the Perspective-n-Point algorithm[6], to estimate the position of the other camera with respect to the first camera. Alternatively we propose a different method where we estimate the 3D position of the pixels from the other camera and then formulate an optimization process which estimates the camera pose using the differences in the 3D positions.

In the experiments section we will show the performance differences between these two methods.

3.1 Ray Tracing

Ray Tracing is a process of projecting a ray from the camera center through the pixel to the world. Let us consider the pixel coordinate $\begin{bmatrix} u^i_{c_1} v^i_{c_1} \end{bmatrix}$. The super script ^{i} refers to the ith pixel in the correspondence pairs. To create the ray, we need two points. The starting point is the camera center, which in the camera frame, is all zeros.

$$S^{c_1} = \begin{bmatrix} s^{c_1}_x & s^{c_1}_y & s^{c_1}_z \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$$

The ending point is obtained from the pixel coordinate which is first normalized and then transformed to the world coordinate.

$$E^{c_1} = \begin{bmatrix} e^{c_1}_x & e^{c_1}_y & e^{c_1}_z \end{bmatrix}^T = \begin{bmatrix} \frac{u^i_{c_1} - c_x}{f_x} & \frac{v^i_{c_1} - c_y}{f_y} & 1 \end{bmatrix}^T$$

The transform which transforms the points in the camera frame to the world frame, in the homogeneous form, is given by,

$$H^w_{c_1} = \begin{bmatrix} R_{c_1} & T_{c_1} \\ 0 & 1 \end{bmatrix}$$

The start and the end points of the ray in the world frame is given by,

$$S^w = H^w_{c_1} S^{c_1} = \begin{bmatrix} s^w_x & s^w_y & s^w_z \end{bmatrix}^T$$

$$E^w = H^w_{c_1} E^{c_1} = \begin{bmatrix} e^w_x & e^w_y & e^w_z \end{bmatrix}^T$$

Using the start and the end points of the ray we form a vector,

$$V^w = \begin{bmatrix} v^w_x & v^w_y & v^w_z \end{bmatrix}^T = E^w - S^w$$

The 3D position of the pixels is given by the intersection of the vector and the starting point to the laser plane described using the parameters $[a,b,c,d]$.

$$I^w_{3\times 1} = \begin{bmatrix} t s^w_x v^w_x & t s^w_y v^w_y & t s^w_z v^w_z \end{bmatrix}^T$$

where,

$$t = -\frac{(a s^w_x + b s^w_y + c s^w_z + d)}{a v^w_x + b v^w_y + c v^w_z}$$

3.2 Perspective-n-Point (PnP)

Given a set of 3D points in the world frame and their projection (pixel coordinates) on the camera frame, PnP algorithm estimates the position and rotation (6 degrees of freedom pose) of the camera with respect to the 3D points.

We start with the camera projection equation (2) and use one of the properties of cross product to solve. The cross product defined between two 3D vectors given by,

$$\vec{a} \times \vec{b} = [a]_\times \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b2 \\ b3 \end{bmatrix}$$

The cross product of a vector with itself is zero,

$$\vec{a} \times \vec{a} = 0$$

Start from the projection equation 2:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \alpha P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where $P = K[R|t]$ is the projection matrix of dimension 3×4. Note, we introduce α which accounts for the fact the the projection matrix is up-to scale. The subscript $_{c_2}$ is dropped for brevity. $P \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ is a 3×1 vector, called M

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \alpha M$$

Take a cross product of M on both sides,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times M = \alpha M \times M = 0$$

Using the property of cross product, $M \times M = 0$

Note, P is not a square matrix. So it does not have an inverse.

Replacing M,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times M = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \times P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

Expanding P,

$$\begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & -1 & v \\ 1 & 0 & -u \\ -v & u & 0 \end{bmatrix} \begin{bmatrix} P_{11}X + P_{12}Y + P_{13}Z + P_{14} \\ P_{21}X + P_{22}Y + P_{23}Z + P_{24} \\ P_{31}X + P_{32}Y + P_{33}Z + P_{34} \end{bmatrix} = 0$$

Rearranging and pulling out the P_{ij} entries to a vector, we get 12 unknowns. Each pixel point gives two values u and v. So we need at least 6 points to solve for P. However in practice we will use a lot more points to account for the inaccuracies in the pixel locations and solve using the least squares approach.

This is of the form $Ax = 0$ and can be solved by taking the singular value decomposition (SVD) of A. $svd(A) = UDV^T$ and $x = last\ column\ of\ V$

We know $P = K[R|t]$, since we already know K and we just calculated P from SVD, we can get $[R|t]$

$$[R|t] = K^{-1}P = P^n_{3x4}$$

P^n is 3×4. The first three columns of P_n corresponds to R and the last column corresponds to t but we cannot assign it directly as R is a rotation matrix and the columns need to be orthogonal. Hence we use SVD again to extract the orthogonal components.

$$svd(P_{c1:c3}) = UDV^T \implies R = UV^T$$

The diagonal matrix D has the eigen values of the form $D = diag(\alpha_1, \alpha_2, \alpha_3)$. The translation component t is given by dividing the last column of P^n by the first eigen value:

$$t = \frac{P^n_{c4}}{\alpha_1}$$

3.3 Depth difference algorithm

As an alternative to PnP, we can use the differences in depth to calculate the position of the cameras. To do this, we use the ray tracing algorithm described in section 3.1 to find the 3D positions of the points from both the cameras. Let $P^w_{c_1} = \begin{bmatrix} p^w_x p^w_y p^w_z \end{bmatrix}^T$ and $Q^w_{c_2} = \begin{bmatrix} q^w_x q^w_y q^w_z \end{bmatrix}^T$ be set of 3D points from the pixel associations from both the camera images c_1 and c_2 respectively.

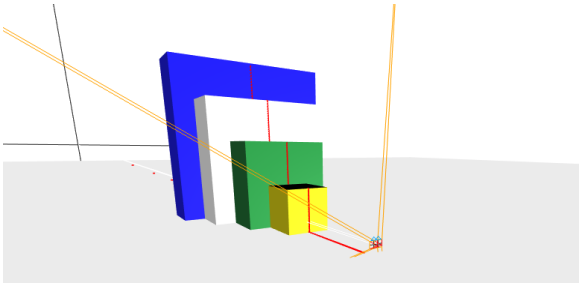

Figure 3: Simulation with stereo cameras and a line striper with varying size blocks

We setup an optimization that minimises the pose of c_2

$$\arg\min_{Rt_{c_2}} \sum \sqrt{(p^w_x + q^w_x)^2 + (p^w_y + q^w_y)^2 + (p^w_z + q^w_z)^2}$$

Here Rt_{c_2} is a 6 vector with 3 translation components and three rotation components in compressed axis-angle representation[7]. This is a non-linear optimization problem as it involves rotation components. We tested the Levenberg-Marquardt[8] optimization algorithm but switched to the Trust Region Reflective[9] algorithm which exhibited better converge.

4 EXPERIMENTS

As mentioned, all experiments are conducted in simulation. Detecting the line strip from the image depends on factors including the terrain's reflectivity, intensity of the laser and robustness of the detection algorithm. We did not want these parameters to influence our analysis. Also, simulation provides accurate ground truth for rigorous quantitative analysis.

4.1 Setup

The setup involves stereo cameras with 90° horizontal field of view and an aspect ratio of 1.3. We changed baselines between experiments. A laser line striper simulating a minimum of 200 rays is added is then projected to the floor to form the line stripes. We also added blocks of varying sizes and positions depending on what the experiment demanded. We were able to precisely move and control the position of the each of these components during our experiments. We developed [1] a tool which helps experimenting with the perception system for this purpose. Figure 3 shows a view of the simulation setup. After we setup the cameras, lasers and blocks, we extracted the point of intersection of the laser line to the ground plane and the

[1] www.merrysprout.com/tools/perception_design/

blocks directly from the simulator. We then re-project these points to the camera frame using the projection equations 1 and 2. This gives a accurate value for the pixel locations.

4.2 Pixel Sensitivity

We evaluated the sensitivity of the estimation algorithm by adding Gaussian noise to the pixel locations. We started from a low standard deviation of 0.01 and to increased it up to 5. In the real world example, it is nominal to expect a deviation of 0.1 to 2 pixels. But it is unlikely that the error model will be Gaussian. We chose Gaussian error for simplicity. Due to the stochastic nature of the noise, we ran multiple iterations of the same experiment and generated box plots to show the tread. Figure 4 shows the results of this experiment. The pose error is calculated by finding the norm of the difference between the estimate pose and the ground truth pose. For this analysis we did not consider the error in the estimated angle.

4.3 PnP vs Depth-difference

The second experiment compares the performance of the PnP method with the Depth-difference method. In this experiment we started with the true pixel locations and increased the standard deviation of the Gaussian noise for different runs. The experiment were run multiple times for each noise setting and the results were aggregated. We also experimented the different orientations of the laser line striper. We calculated the pose error using the same method described in section 4.2.

5 RESULTS

5.1 Pixel Sensitivity

Figure 4 shows the error in the estimated position of the PnP algorithm for different amount of Gaussian noise added to the pixel location. The error in the estimated pose is well below half a centimeter for pixels with noise of one standard deviation. The error in the estimated pose grows exponentially as the pixel noise increases. So, as long as we can keep the pixel noises to below one standard deviation, we should be able to predict the pose of the other camera to within half a centimeter accuracy.

5.2 PnP vs Depth-difference

Figure 5 and figure 6 show the error of the estimated pose using the depth-difference algorithm and the PnP algorithm. As we can see that the standard deviation of the Gaussian noise added to the pixel locations increase from 0 to 0.3. The error of the estimated pose is shown along the Y axis.

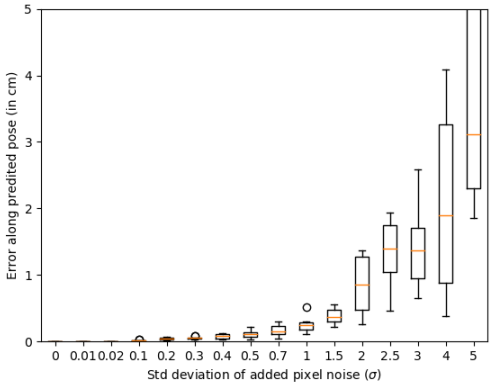

Figure 4: Box plot showing the error in the estimated pose of the PnP algorithm for varying level added Gaussian pixel noise

Figure 5 shows the comparison between the depth-difference method and the PnP method for a line striping laser which shines a vertical plane perpendicular to the ground plane. This casts a vertical line in both the images. Figure 3 shows a similar scenario. The two plots show the error of the estimated pose using these methods. Both these methods start with low error in the estimated pose but the error quickly increases as the as more noise is added to the pixel locations. Notice the scale of error in both the plots. The PnP method seems to be much more resilient to pixel noise and scales linearly compared to the depth-difference method which relies seems to scale exponentially. One possible explanation is, we used the PnP algorithm in OpenCV [10] which includes the RANSAC[11] algorithm for outlier removal. This makes the PnP method robust. The depth-difference method could be robustified using Bisquare weight [12] before feeding the residuals to the optimization algorithm.

Figure 6 shows the comparison for a horizontal laser. Note the change in scale of the error between the two plots. The error for the depth-difference method starts in the order of millimeters for low pixel noise but quickly climbs up to over a 1cm for a relatively low noise of 0.3 standard deviation. However, the error reported by the PnP algorithm stays very high. It is likely that the PnP algorithm runs into a singularity case at this angle. We also ran other experiments setting the laser at different angles[2]. We found that the PnP algorithm is robust for most of the angle differences and starts to accumulate errors when the laser plane is close to horizontal.

[2]we did not include those graphs for brevity

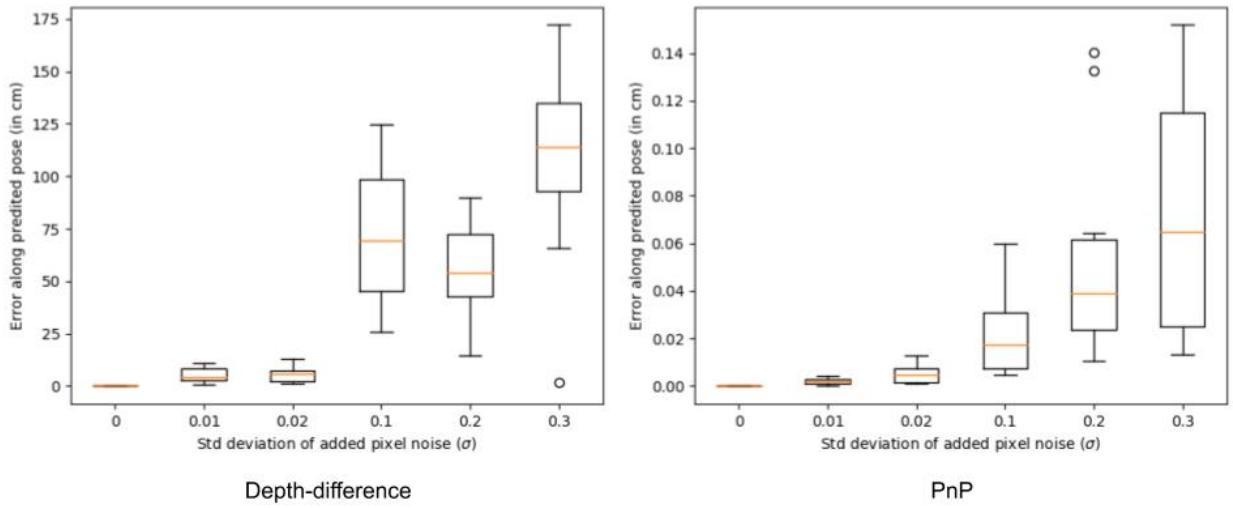

*Figure 5: For a **vertical** line striping laser, comparing the error of the estimated poses using the depth-difference and the PnP method. The PnP method performs with significantly lower error.*

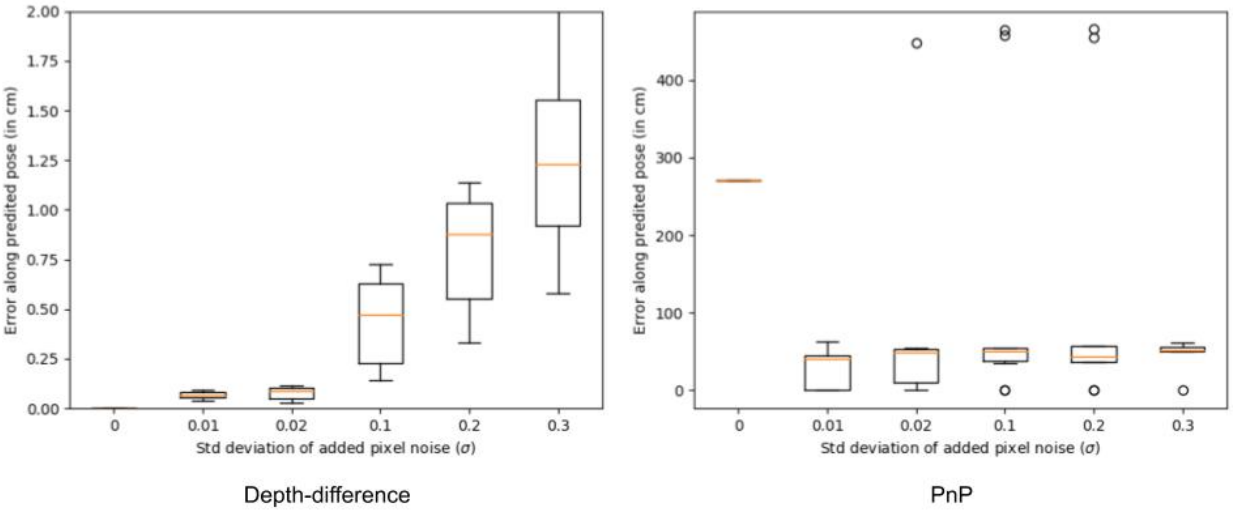

*Figure 6: For a **horizontal** line striping laser, comparing the error of the estimated poses using the depth-difference and the PnP method. The depth-difference method performs better.*

6 CONCLUSION

We proposed a method to verify stereo extrinsic using point correspondences from a line striping laser. We compared the widely used PnP algorithm with our depth-difference algorithm. We also showed how pixel noise affects the estimated camera poses. Future work should focus on considering the error in the estimated angles along with the positions as they will have much greater impact. We modelled the pixel noises are Gaussian distributions in our simulation for simplicity. However in real world this is not true. These should be modelled using more realistic error distributions. We did not address the association problem. That is, how a pixel in one image is associated with the corresponding pixel in the next image usually by feature matching techniques like SIFT, SURF or ORB. Recently, deep learning techniques are also used to get better results. Data association for laser striping will be investigated. Finally, we need to test and validate these findings on real data.

Acknowledgment

The technology, rover development and flight are supported by NASA LSITP contract 80MSFC20C0008 MoonRanger.

References

[1] J Maki, et.al., (May 2012) ”The Mars Science Laboratory Engineering Cameras,”, 1-17.

[2] Z. Zhang (2000) ”A flexible new technique for camera calibration.” IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 22(11), pp. 1330-1334

[3] J. F. Bell III. et.al., (2017) “The Mars Science Laboratory Curiosity rover Mastcam instruments:”

[4] Laserlineoptics ”Powell lens” Accessed September 18, 2020 www.laserlineoptics.com/powell_primer.html

[5] Roth, Scott D. (February 1982), ”Ray Casting for Modeling Solids”, Computer Graphics and Image Processing, 18 (2): 109–144

[6] Wikipedia ”Perspective-n-Point” Accessed September 18, 2020 en.wikipedia.org/wiki/Perspective-n-Point

[7] wikipedia ”Axis-angle representation” Accessed September 18, 2020 en.wikipedia.org/wiki/Axis

[8] Ananth Ranganathan 2004 ”The Levenberg-Marquardt Algorithm” Tutorial. Accessed September 18, 2020 ananth.in/docs/lmtut.pdf

[9] Sorensen, D. C. (1982). ”Newton’s Method with a Model Trust Region Modification”. SIAM J. Numer. Anal. 19 (2): 409–426. doi:10.1137/0719026

[10] ”Opencv PnP RANSAC” Accessed September 18, 2020 docs.opencv.org/3.4/d9/d0c/group__calib3d.html

[11] Martin A. Fischler, Robert C. Bolles (June 1981). ”Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography” (PDF). Comm. ACM. 24 (6): 381–395

[12] ”Engineering Statistics Handbook” Accessed September 18, 2020 www.itl.nist.gov/div898/handbook/mpc/section6/mpc6522.htm