

Localization of a Smart Infrastructure Fisheye Camera in a Prior Map for Autonomous Vehicles

Subodh Mishra*, Armin Parchami, Enrique Corona, Punarjay Chakravarty,
Ankit Vora, Devarth Parikh, and Gaurav Pandey

Abstract—This work presents a technique for localization of a smart infrastructure node, consisting of a fisheye camera, in a prior map. These cameras can detect objects that are outside the line of sight of the autonomous vehicles (AV) and send that information to AVs using V2X technology. However, in order for this information to be of any use to the AV, the detected objects should be provided in the reference frame of the prior map that the AV uses for its own navigation. Therefore, it is important to know the accurate pose of the infrastructure camera with respect to the prior map. Here we propose to solve this localization problem in two steps, (i) we perform feature matching between perspective projection of fisheye image and bird's eye view (BEV) satellite imagery from the prior map to estimate an initial camera pose, (ii) we refine the initialization by maximizing the Mutual Information (MI) between intensity of pixel values of fisheye image and reflectivity of 3D LiDAR points in the map data. We validate our method on simulated data and also present results with real world data.

Index Terms—Fisheye Camera, Camera Localization, Mutual Information

I. INTRODUCTION

Environment perception in Autonomous Vehicles (AV) is a challenging problem. With the current approach of using only on-board sensors to solve the perception problem, it is impossible to sense occluded areas and mitigate the effects of sensor outage. Complex traffic intersections with buildings close to the curb may minimize the field of view of an AV's sensors. Integration of smart infrastructure nodes (sensing and compute) on roads where AVs operate can help overcome these challenges. The elevated and static view-point of the smart sensors enables them to observe the environment, detect more objects in the scene, and communicate that information to AVs. AVs can fuse that information with their own sensor measurements and augment their situational awareness. Fisheye cameras are well known for their low cost and wide Field of View (FoV), making them suitable for such smart infrastructure based sensing applications. However, the fisheye camera needs to provide this information in the same coordinate frame as the vehicle. For this reason, they need to be localized or registered within the same map that is used by the AV for navigation. In this work, we propose a method to localize a downward looking static smart infrastructure fisheye camera in a prior map consisting of a metric satellite image, and a co-registered LiDAR map of ground points with their LiDAR reflectivity values. An overview of the approach is shown in Figure 1.

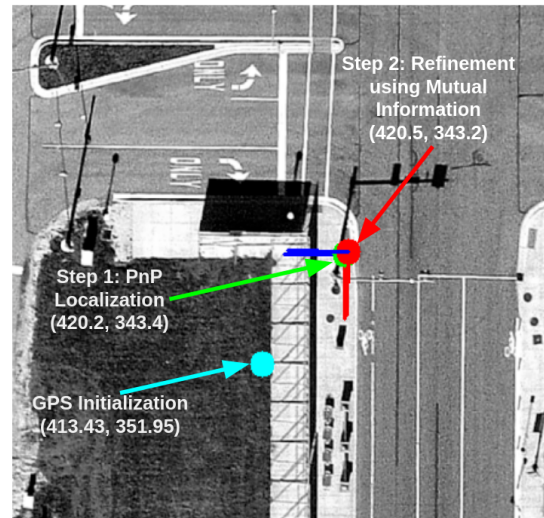


Fig. 1: Overview of our two step approach to camera localization. The coordinates shown are in meter, wrt the origin of the map. We start with a noisy GPS initialization (cyan), then perform feature matching between satellite image and rectified fisheye image to obtain an initial camera pose - PnP Localization (Green), and finally maximize the Mutual Information between LiDAR map and the fisheye image to obtain a refined camera localization estimate.

II. RELATED WORK

There are several contributions on localization of camera images in prior maps (satellite imagery or LiDAR generated 3D maps). Satellite maps can be procured easily from third party sources [1], [2], and with the widespread use of LiDARs in autonomous driving, we now have several high definition map providers which provide dense 3D map of the environment [3]. This has made localization of cheaper sensors like cameras on prior maps an active area of research with an ultimate aim of using cheaper sensors on-board an AV. [4] presents monocular perspective camera localization in pre-built 3D LiDAR maps using 2D-3D line correspondences. This method shows promise only in structured environment where lines can be easily detected in both camera and the LiDAR map. [5] presents LiDAR map based monocular camera localization in urban environment. Unlike [4] which depends on detection of geometric primitives like lines in both the sensing modalities, [5] uses dense appearance based approach which work in unstructured environments. In [5], given an initial belief of the camera pose, they generate several synthetic views of the environment by projecting the LiDAR map points using a perspective camera model, and compare these synthetic views against the live camera feed. The synthetic view which maximizes the Normalized Mutual

*Graduate student in the Department of Mechanical Engineering, Texas A&M University subodh514@tamu.edu, work done as intern
All other authors with the Ford Autonomous Vehicles LLC, USA

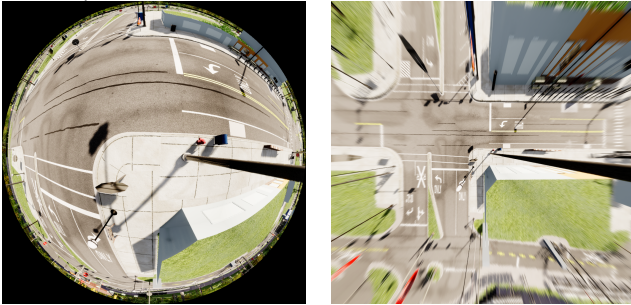
Information (NMI) between the real image gray scale values and the projected points' LiDAR reflectivity values, is the solution to the localization problem. [5] draws inspiration from work done on 3D-LiDAR Camera extrinsic calibration described in [6], which uses maximization of Mutual Information (MI) for calibrating a 3D-LiDAR Camera pair. [7] presents a camera localization technique which matches ground imagery obtained by cameras onboard an AV to the available satellite imagery. The camera images are warped to obtain a bird's eye view (BEV) of the ground. Next, the BEV image is matched with the given satellite imagery using SIFT [8] features.

While the above mentioned approaches provide solutions for perspective cameras, our focus is to localize (estimate $[\mathbf{C}\mathbf{R}_W, \mathbf{C}\mathbf{t}_W]$ in Equation 1) a downward looking static fisheye camera (Figure 2a) in a prior map (Figure 3) which consists of 2D satellite imagery with metric information (Figure 3a) and 3D LiDAR map of ground points (Figures 3b, 3c). We initialize the camera localization using feature matching as done in [7], and refine the camera localization using maximization of a MI based cost function as done in [5] and [6].

III. OVERVIEW

In this section we provide an overview of the various components of our implementation.

A. Fisheye Camera



(a) Fisheye image captured at a traffic intersection in our simulator (b) Rectification of fisheye image to perspective image

Fig. 2: Fisheye Image (Figure 2a) and its perspective rectification (Figure 2b)

We use the fisheye camera projection model proposed in [9] to project 3D point $\mathbf{P}_W = [X_W, Y_W, Z_W]$ defined in the prior map coordinate frame \mathbf{W} to a 2D point \mathbf{p} on the fisheye camera image plane using Equation 1.

$$\mathbf{p} = \Pi(K, D, \xi, [\mathbf{C}\mathbf{R}_W, \mathbf{C}\mathbf{t}_W], \mathbf{P}_W) \quad (1)$$

Here $\Pi()$ is the projection function, $K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \end{bmatrix}$, $D = [k_1, k_2, p_1, p_2]$ and ξ are the camera intrinsics, and $[\mathbf{C}\mathbf{R}_W, \mathbf{C}\mathbf{t}_W]$ is the camera extrinsic. $\mathbf{C}\mathbf{R}_W \in SO(3)$ is an orthonormal rotation matrix and $\mathbf{C}\mathbf{t}_W \in R^3$ is a 3D vector. The goal of this work is to estimate the unknown camera extrinsic $[\mathbf{C}\mathbf{R}_W, \mathbf{C}\mathbf{t}_W]$ in the prior map frame \mathbf{W} .

1) *Intrinsic Calibration*: We estimate the intrinsic parameters K , D and ξ by collecting several images of a large checkerboard at different poses, and feeding those images to the omnidirectional camera calibrator in OpenCV [10], which provides an implementation¹ of the intrinsic calibration technique presented in [11].

2) *Rectification*: The intrinsic camera calibration parameters are used to rectify the fisheye image (Figure 2a) into its corresponding perspective image (Figure 2b) utilizing OpenCV's rectification routines. Although perspective rectification results in loss of field of view, it makes the application of computer vision algorithms developed for perspective images possible for fisheye images.

B. Prior Map

The prior map (Figure 3) consists of two important components which are registered and expressed in the frame of reference \mathbf{W} . They are:

1) *Satellite Map*: The satellite map is a metric Bird's Eye View (BEV) satellite image (Figure 3a). In our case, a pixel on the image corresponds to 0.1 m on the ground.

2) *LiDAR Map*: The prior LiDAR map is built using an offline mapping process described in [5]. Broadly, a survey vehicle equipped with several 3D LiDAR scanners and a high end inertial navigation system is manually driven and sensor data is collected in the environment we want to map. Next, an offline pose-graph optimization SLAM (Simultaneous Localization and Mapping) problem is solved to obtain the accurate global pose of the vehicle. Finally, a dense ground point mesh is constructed from the optimized pose graph using region growing techniques which gives a dense 3D point cloud map. The ground points from this dense 3D cloud are used to generate the LiDAR ground reflectivity image (Figure 3b) and ground height image (Figure 3c). The LiDAR Map (Figures 3b and 3c) is aligned with the satellite imagery (Figure 3a) using the GPS measurements from the inertial navigation system.

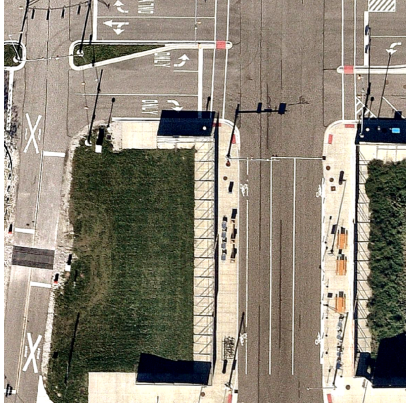
IV. PROBLEM FORMULATION

The goal of this work is to estimate the unknown camera pose $[\mathbf{C}\mathbf{R}_W, \mathbf{C}\mathbf{t}_W]$ in the prior map frame \mathbf{W} . We assume that we have a noisy estimate of the fisheye camera's (GPS) position (no orientation) in the map which helps us reduce the search space in the prior map. We follow a two step approach to register the camera in the prior map, the details of which are presented in Sections IV-A and IV-B, and a broad overview is provided in Figure 4.

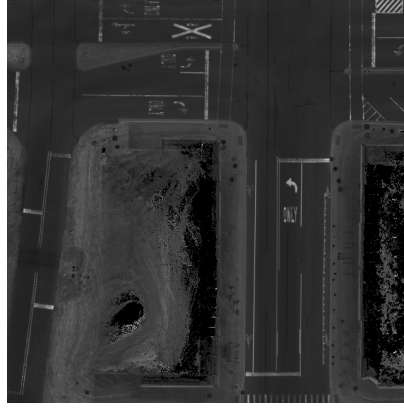
A. Initialization using sparse feature matching

Traditionally available feature detection, description and matching techniques are usually suitable for perspective images only. Therefore, we rectify the fisheye image into the corresponding perspective image as explained in Section III-A.2, and use SuperGlue [12], a pre-trained deep learning based feature matching algorithm, for matching features

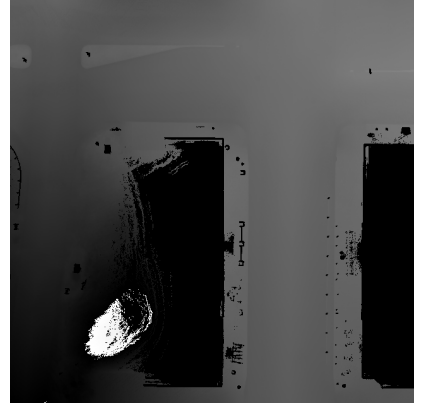
¹https://docs.opencv.org/4.5.2/dd/d12/tutorial_omnidir_calib_main.html



(a) Satellite Map



(b) LiDAR Map: Reflectivity of Ground Points



(c) LiDAR Map: Height of Ground Points

Fig. 3: **Prior Map:** Figures 3a, 3b & 3c show the components of our prior map. The full map is not shown in the interest of space.

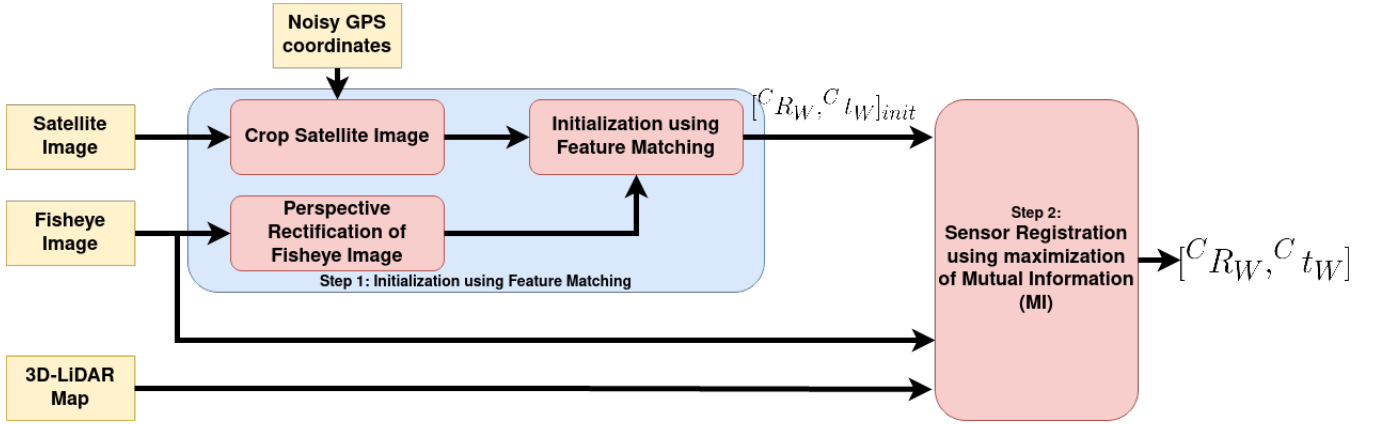


Fig. 4: **Overview of the method:** The block diagram shows the two steps involved in our approach. In Step 1, we match features between the perspective projection of fisheye image and a cropped satellite map to initialize camera pose, and in Step 2 we refine this initialization by maximization of Mutual Information between fisheye image and prior 3D-LiDAR map.

(Figure 5) between the rectified fisheye image and the cropped satellite image (cropped using GPS initialization, refer Figure 4). The matched features are used to solve a Perspective-n-Point (PnP) problem [13], [14] to estimate the initial pose of camera (also called the PnP estimate) in the prior map reference frame \mathbf{W} . As we know the metric scale of the satellite image (1 pixel = 0.1 m), we obtain the camera pose in metric units.

refine the initial camera pose estimate from Section IV-A by maximizing the Mutual Information (MI) between the LiDAR reflectivity of ground points and the fisheye grayscale values at the pixel locations onto which the LiDAR points are projected using the camera pose $[{}^C\mathbf{R}_W, {}^C\mathbf{t}_W]$.

1) *Theory:* MI (Equation 2) provides a way to statistically measure mutual dependence between two random variables X and Y .

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \quad (2)$$

Where $H(X)$ and $H(Y)$ are the Shannon entropy over random variables X and Y respectively, and $H(X, Y)$ is the joint Shannon entropy over the two random variables:

$$H(X) = - \sum_{x \in X} p_X(x) \log p_X(x) \quad (3)$$

$$H(Y) = - \sum_{y \in Y} p_Y(y) \log p_Y(y) \quad (4)$$

$$H(X, Y) = - \sum_{x \in X} \sum_{y \in Y} p_{X,Y}(x, y) \log p_{X,Y}(x, y) \quad (5)$$

The entropy $H(X)$ of a random variable X denotes the amount of uncertainty in X , whereas $H(X, Y)$ is the amount of uncertainty when the random variables X and

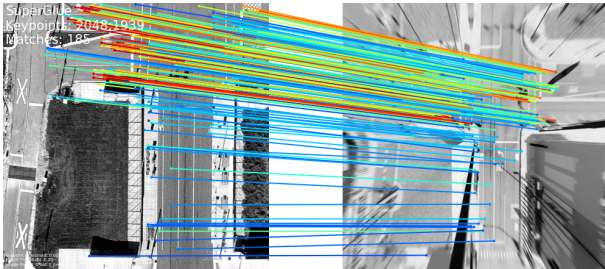


Fig. 5: SuperGlue [12] Matching between Satellite Image (Left) & perspective projection of fisheye image (Right)

B. Refinement of camera localization using Maximization of Mutual Information

Mutual Information (MI) has been used in several fields for registering data from multi-modal sensors [15], [16]. We

Y are co-observed. The formulation of MI in Equation 2 shows that maximization of $MI(X, Y)$ is achieved by minimization of the joint entropy $H(X, Y)$, which coincides with minimization of dispersion of two random variable's joint histogram.

2) *Mathematical Formulation:* Let $\{\mathbf{P}_{\mathbf{W}_i}; i = 1, 2, \dots, n\}$ be the set of 3D points whose coordinates are known in the prior map reference frame \mathbf{W} and let $\{X_i; i = 1, 2, \dots, n\}$ be the corresponding reflectivity values for these points ($X_i \in [0, 255]$). Equation 1 presents the relationship between $\mathbf{P}_{\mathbf{W}_i}$ and its image projection \mathbf{p}_i as a function of $[\mathbf{C}\mathbf{R}_{\mathbf{W}}, \mathbf{C}\mathbf{t}_{\mathbf{W}}]$. Let $\{Y_i; i = 1, 2, \dots, n\}$ be the grayscale intensity of the pixels \mathbf{p}_i where $\mathbf{P}_{\mathbf{W}_i}$ project onto, such that:

$$Y_i = I(\mathbf{p}_i) \quad (6)$$

where $Y_i \in [0, 255]$ and I is the grayscale fisheye image. Therefore, X_i is an observation of the random variable X , and for a given $[\mathbf{C}\mathbf{R}_{\mathbf{W}}, \mathbf{C}\mathbf{t}_{\mathbf{W}}]$, Y_i is an observation of random variable Y . The marginal ($p_X(x)$, $p_Y(y)$) and joint ($p_{X,Y}(x, y)$) probabilities of the random variables X and Y , required for calculating MI (Equation 2), can be estimated using a normalized histogram (Equation 7):

$$\hat{p}(X = k) = \frac{x_k}{n}, k \in [0, 255] \quad (7)$$

where x_k is the observed counts of the intensity value k .

3) *Global Optimization:* $\mathbf{C}\mathbf{R}_{\mathbf{W}} \in SO(3)$ is an orthonormal rotation matrix which can be parameterized as Euler angles $[\phi, \theta, \psi]^\top$ and $\mathbf{C}\mathbf{t}_{\mathbf{W}} = [x, y, z]^\top$ is an Euclidean 3-vector. ψ is the rotation of the camera along its principal axis. In our context, the fisheye camera is facing vertically downward so we do not refine the ϕ & θ and leave it at what the feature matching based technique (Section IV-A) determines it to be, which is very close to 0. Therefore, as far as rotation variables are concerned, we refine only ψ . We represent all the variables to be optimized together as $\Theta = [x, y, z, \psi]^\top$. The optimization is posed as a maximization problem:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} MI(X, Y; \Theta) \quad (8)$$

V. EXPERIMENTS AND RESULTS

This section describes the experiments performed to evaluate the proposed technique using data obtained from both our simulator and real world sensor.

A. Simulation Studies

We first validate our approach on a simulator which is built using data from real sensors. The Mathworks' tool RoadRunner [17] is used to generate the 2D features like lane geometry and lane markings with the satellite map used as a reference. The 3D structures are created using the Unreal Engine Editor [18] with the help of real satellite and 3D LiDAR maps. Since the simulated environment is created using the prior map components, it can safely be assumed that the simulator aligns with the real world to a high degree of accuracy. In order to generate the fisheye images, we model a

fish-eye camera in Unreal Engine using the equidistant model with a field of view of 180 degrees. We demonstrate our approach in simulation for the fisheye image shown in Figure 2a. The fisheye image is first rectified (Section III-A.2) to generate Figure 2b, which is used for estimating the initial camera pose using the approach in Section IV-A. Next, the initialization is refined using maximization of MI (Section IV-B).

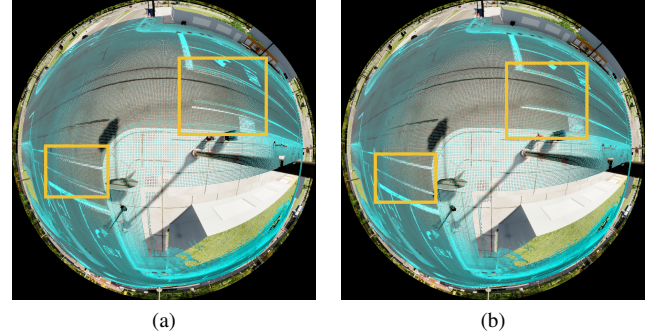


Fig. 6: Two step approach for camera localization in prior map : Figure 6a shows the projection of 3D-LiDAR ground points (cyan) using the initial estimate obtained using feature matching (PnP Estimate from Section IV-A), and Figure 6b shows the projection of 3D-LiDAR ground points using the refined camera pose estimate from maximization of MI (from Section IV-B). The misalignment visible in Figure 6a, is absent in Figure 6b (best viewed digitally).

We qualitatively validate the camera localization (Figure 6) estimate ($[\mathbf{C}\mathbf{R}_{\mathbf{W}}, \mathbf{C}\mathbf{t}_{\mathbf{W}}]$) by projecting points from the 3D-LiDAR map (Figures 3b and 3c) onto the fisheye image (Figure 2a). As shown in Figure 6a, the projection of LiDAR map points on the fisheye image obtained using the initial camera pose are not well aligned. When we plot (Figure 7) the MI around the initial camera pose, we observe that it is not at its maximum at the initial estimate (also called PnP Estimate), thus holding the promise for further improvement. Similarly, Figure 8 presents the surface plot of MI, which shows the presence of a global maximum in each sub-plot. Therefore, on solving the optimization problem posed in Equation 8 we obtain camera pose estimate which maximizes the MI between the two modalities and results in negligible misalignment of the projected LiDAR map points in Figure 6b.

We run 100 independent trials to evaluate the robustness of the MI based refinement method (Section IV-B) to change in initialization (Figure 9). The translation parameters show greater variance when compared to yaw ψ . The higher variance of the translation variables can be attributed to the fact that the MI based cost function is less sensitive to changes in translation variables, especially in the outdoor scenario where most of the points lie in the far field. In the limiting case, far away points are considered to be points at infinity, represented as $[x, y, z, 0]^T$, which, under camera projection (Equation 1), render the translation variable ($\mathbf{C}\mathbf{t}_{\mathbf{W}}$) in the optimization problem (Equation 8) un-observable. This result is also presented in [6], specifically when discussing sensor registration in an outdoor environment using only a single image - LiDAR scan pair, which is similar to our situation.

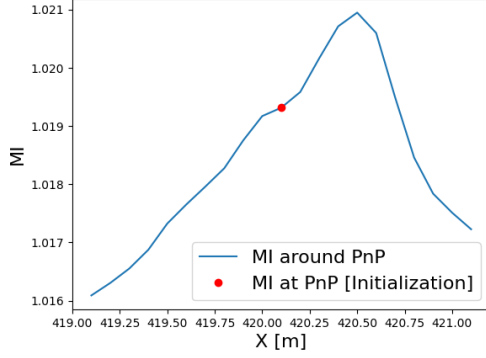


Fig. 7: Plot of MI around the PnP estimate (from Section IV-A). Plot shows that MI is not at maximum at the PnP estimate, therefore the maximization of MI may reduce the misalignment in projection of 3D-LiDAR points visible in Figure 6a

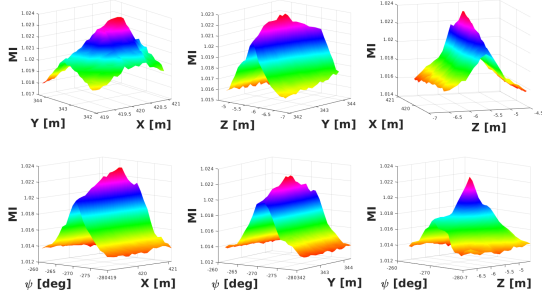


Fig. 8: Plot of MI by perturbing two degrees of freedom around the PnP Estimate (i.e. the initial estimate from Section IV-A). The cost function from single Image LiDAR Scan pair is not differentiable at several points. Hence, we use an exhaustive grid search around the initialization point to arrive at solution where MI is maximized.

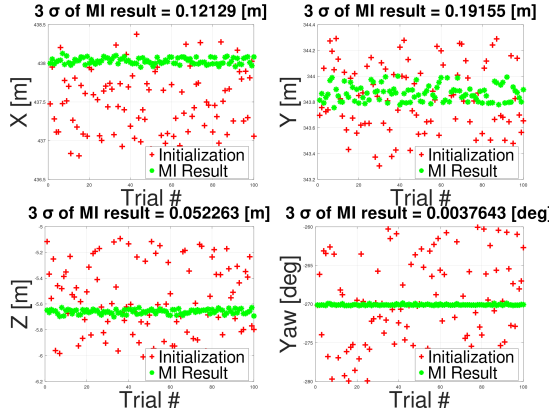


Fig. 9: Performance of MI based fisheye camera localization refinement for different initial conditions. Here we perform 100 independent trials. + is the initialization, * is the final result.

B. Real World Experiments

In order to demonstrate the validity of our algorithm in realistic situations, we conducted experiments with data collected from a real fisheye camera (Figure 10).

1) *System Description:* We use a fisheye lens Fujinon FE185C057HA 2/3 inch sensor, which provides 185° of vertical and horizontal FoV. Our camera is a 5MP Sony IMX264. We mount our sensor from a tripod (Figure 10), looking vertically down, and capture images of the en-

vironment. Our ultimate goal is to mount these cameras at challenging intersections for navigation of autonomous vehicles, and use the proposed method to register them in a prior map. We use an iPhone to provide us an approximate GPS location (without the orientation) of the fisheye camera, which is used to limit the search space in the prior map. We use a high accuracy RTK-GPS (uBlox ZED F9P GNSS + uBlox antenna ANN-MB-00) unit to measure GPS coordinates of distinctive corners on road markings that can be used for quantifying the accuracy of camera localization (Figure 12).

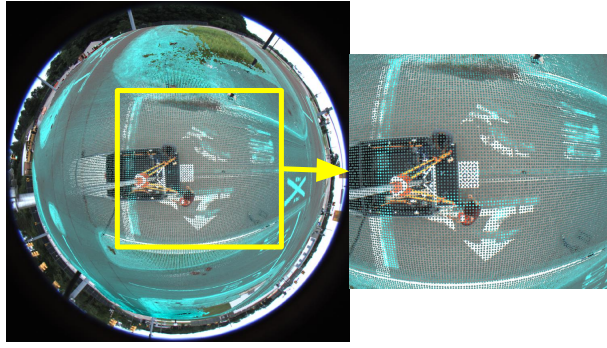


Fig. 10: Collecting data for real experiments using a tripod mounted downward looking fisheye camera. Our ultimate goal is to mount these cameras, along with our smart infrastructure nodes, at challenging intersections for navigation of autonomous vehicles.

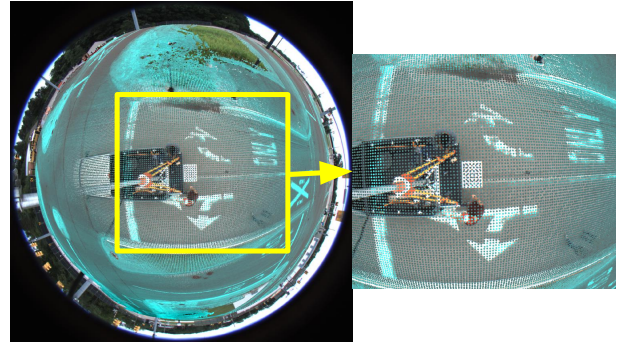
2) *Results:* We present results with real world data from two different locations in Figure 11 which qualitatively demonstrate the incremental improvement in camera localization using our two step approach. While the projection of LiDAR points onto the fisheye image using the initial camera localization appears misaligned in Figures 11a and 11c, the misalignment is reduced when the LiDAR points are projected using refined camera localization in Figures 11b and 11d. We quantify the veracity of camera localization by measuring the average reprojection error for points on the fisheye image whose GPS locations we have measured using high accuracy RTK-GPS. We manually mark these points on the fisheye image, and measure the difference between them and the reprojection of the corresponding 3D point in the prior map onto the fisheye image, using the estimated camera localization. The results presented in Figure 12 show that the reprojection error on the fisheye image plane reduces when we refine the initial camera localization by maximizing the mutual information.

VI. DISCUSSION & CONCLUSION

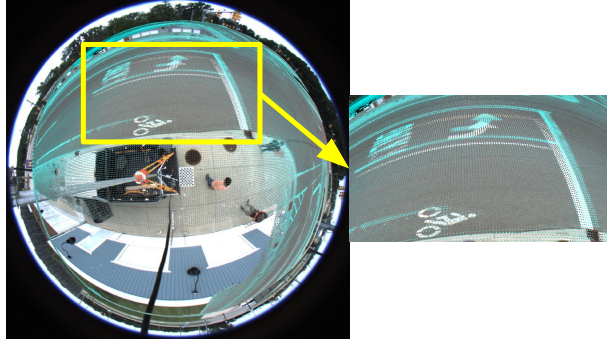
We present an approach to localize a smart infrastructure node equipped with a fisheye camera. The downward facing



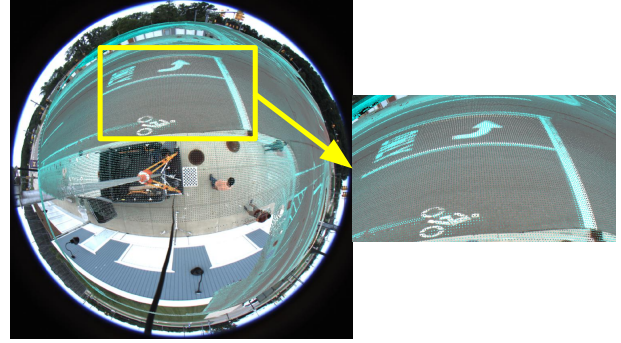
(a) Location 1 - Projection of LiDAR points (cyan) using PnP Estimate



(b) Location 1 - Projection of LiDAR points (cyan) using MI Estimate

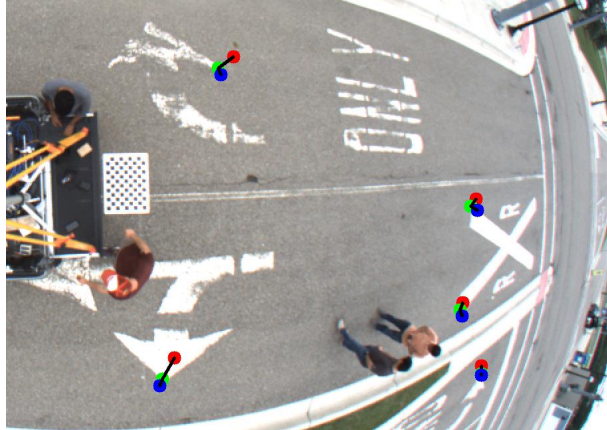


(c) Location 2 - Projection of LiDAR points (cyan) using PnP Estimate



(d) Location 2 - Projection of LiDAR points (cyan) using MI Estimate

Fig. 11: **Real World Experiments:** Projection of 3D-LiDAR ground points (cyan) onto fisheye camera using the initial camera pose (PnP Estimate) (Figure 11a, 11c) and MI based refinement of initial camera pose (Figure 11b, 11d). The misalignment of LiDAR points visible in highlighted areas in Figures 11a, 11c, are minimized in Figures 11b, 11d



(a) Location1 - Average Reprojection Error with PnP = 20.52 pixel, and with Maximization of MI = 9.12 pixel



(b) Location2 - Average Reprojection Error with PnP = 26.40 pixel, and with Maximization of MI = 13.11 pixel

Fig. 12: **Green Circle** - Hand annotated corner point whose GPS location was measured using a high accuracy RTK-GPS unit, **Red Circle** - Projection of corner point's position onto Fisheye Image using PnP estimate (Section IV-A), **Blue Circle** - Projection of corner point's position onto Fisheye Image using MI estimate (Section IV-B).

fisheye image is registered to a prior map, comprising of a co-registered satellite image and a ground reflectivity/height map from LiDAR-SLAM. Our two-step approach uses feature matching between the rectified fisheye image and the satellite imagery to get an initial camera pose, followed by maximization of MI between the fisheye image and 3D LiDAR map to refine the initial camera localization. Since we have only a single camera image to register against (the smart infrastructure node is static), the cost surface may not always be smooth [6] and therefore not differentiable - leading to

the failure of gradient descent methods. Hence, we use an exhaustive grid search method to find the optimal camera pose. Such a search may be time-consuming (depending on the number of 3D LiDAR map points used for calculating MI (Equation 2), the interval of exhaustive grid search and the available compute power), and not suitable for real-time operation. This is acceptable for our application, because we need to localize the smart infrastructure node once at install and this can be an offline process. Moreover, this method can be accelerated by use of GPUs.

REFERENCES

- [1] A. Vora, S. Agarwal, G. Pandey, and J. McBride, "Aerial imagery based lidar localization for autonomous vehicles," 2020.
- [2] M. Technologies. (2020). [Online]. Available: <https://www.maxar.com/>
- [3] A. G. Vora, S. Agarwal, J. N. Hoellerbauer, and F. Shaik, "High definition 3d mapping," Jun. 6 2019, uS Patent App. 15/831,295.
- [4] H. Yu, W. Zhen, W. Yang, J. Zhang, and S. Scherer, "Monocular camera localization in prior lidar maps with 2d-3d line correspondences," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4588–4594, 2020.
- [5] R. W. Wolcott and R. Eustice, "Visual localization within lidar maps for automated urban driving," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 176–183, 2014.
- [6] G. Pandey, J. McBride, S. Savarese, and R. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," *Twenty-Sixth AAAI Conference on Artificial Intelligence*, vol. 26, 01 2012.
- [7] A. Viswanathan, B. R. Pires, and D. Huber, "Vision based robot localization by ground to satellite matching in gps-denied situations," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 192–198.
- [8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>
- [9] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3945–3950.
- [10] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [11] B. Li, L. Heng, K. Koser, and M. Pollefeys, "A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1301–1307.
- [12] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *CVPR*, 2020. [Online]. Available: <https://arxiv.org/abs/1911.11763>
- [13] R. Haralick, D. Lee, K. Ottenburg, and M. Nolle, "Analysis and solutions of the three point perspective pose estimation problem," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 592–598.
- [14] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, Jun. 1981. [Online]. Available: <https://doi.org/10.1145/358669.358692>
- [15] P. Viola and W. Wells, "Alignment by maximization of mutual information," vol. 24, 01 1995, pp. 16–23.
- [16] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *IEEE Transactions on Medical Imaging*, vol. 16, no. 2, pp. 187–198, 1997.
- [17] Mathworks, "Roadrunner," 2019. [Online]. Available: <https://www.mathworks.com/products/roadrunner.html>
- [18] U. Engine, "Unreal Editor," 2016. [Online]. Available: <https://docs.unrealengine.com/4.26/en-US/Basics/UI/>