Predictive Display for Teleoperation Based on 3D Reconstruction Using Lidar-Camera Fusion*

Gaurav Sharma, Raunak Manekar and Rajesh Rajamani

Abstract— **Teleoperation could be used to replace a backup safety driver on autonomous vehicles and could play a valuable role in scaling up deployment of autonomous vehicles. The surrounding environment of the remote vehicle needs to be recreated for the teleoperator using video images received over wireless channels. To handle the significant time delays in receiving remote video data, this paper develops a predictive display system that uses deep learning to estimate the current video display for the teleoperator from old (delayed) camera images. First, old camera and Lidar data are fused to create a 3D reconstruction of the remote vehicle environment using conventional and deep-learning-based algorithms. Then the ego-vehicle's real-time position and orientation variables are estimated using an extended Kalman filter. Predictive modification of the reconstructed old 3D scene is performed using the ego-vehicle's estimated new trajectory variables. Deep-learning based image in-painting is used to improve image quality. Furthermore, this paper also introduces a new image comparison metric for evaluating the accuracy of the object detection and localization performance in the predictive display image. Real-world experimental data from the nuScenes and Kitti datasets are used for evaluation of the proposed system. The predictive display images are compared with ground truth images using various image comparison metrics and shown to provide significantly superior performance compared to the actual delayed images received over wireless channels.**

I. INTRODUCTION

There is significant ongoing research related to autonomous vehicles (AVs) in various parts of the world. However, even according to the most optimistic estimates, a fully self-driving vehicle will not be sold to the public till at least 2028 at the earliest [1]. Current AV technology is primarily at Level 3 and Level 4 of SAE capability levels, with significant ongoing testing of Level 4 capabilities. While AVs can operate autonomously a majority of the time, occasional intervention from a human driver is needed. Such scenarios requiring human intervention may include unexpected snow on the road, construction zones, blocked roads or component failures. In many AV companies' backup safety drivers are essential for vehicle testing. Enabling the teleoperation of AVs will essentially allow remote human intervention in unexpected situations and will also eliminate the need for a backup safety driver.

Some startup companies like Vay [2] and Halo currently use human teleoperators to drive rental cars to and from customers remotely. A Swedish company, Einride [3], utilizes remote operators to assist their autonomous trucks in complex driving scenarios. Future applications of occasional AV teleoperation may include taxi fleets, valet parking, ride-sharing, and autonomous buses for public transport. Hence, AV teleoperation is likely to be a valuable technology in the near future.

In teleoperation, the surrounding environment of the remote vehicle needs to be recreated for the teleoperator using video images received over wireless channels. The camera and Lidar data, owing to their high data size, can experience significant delays in their wireless communication. In the context of AVs, even a 0.2 s delay can degrade the vehicle control performance of the teleoperator [4]. Furthermore, if the delay increases to 0.5 s, it can become virtually impossible for the teleoperator to effectively control the vehicle [5].

One solution to this latency problem lies in using Predictive Display (PD) to provide realistic intermediate display updates to the teleoperator to compensate for the delayed image feed. An essential feature for using PD is to know the ego vehicle's current real-time position and orientation, which is obtained in this paper through state estimation using GPS and IMU signals. Since these sensor signals are small, they can be transmitted more frequently and consistently than camera images. Thus, this estimation-based PD system can obtain real-time estimates of the ego vehicles' trajectory and synthesize new images, which compensate for the delayed camera data. This paper proposes to develop an estimation-based predictive display (PD) system that performs novel sensor fusion of delayed camera and Lidar data to recreate the 3D environment around the ego vehicle, estimate trajectories of the ego vehicle, and use these trajectory variables to synthesize predictive images for accurate teleoperation. Furthermore, deep learning-based image-inpainting will be used to improve the quality of the synthesized images.

Prior research work in PD has been simplistic and includes displaying the predicted position of the ego-vehicle on the delayed image feed using a pointing line [6], a semitransparent vehicle [7], or a rectangular frame and tracks [8]. However, in these papers, the authors assumed that the ego position was known and did not do state estimation.

* Research supported in part by NSF Grant CNS 2321531 and by the University of Minnesota InterS&Ections Program. Gaurav Sharma and Raunak Manaker contributed equally to this paper.

R. Rajamani and Gaurav Sharma are with the University of Minnesota, Twin Cities, Minneapolis, MN 55455 USA (Email: rajamani@umn.edu;TEL:612-626-7961; sharm936@umn.edu)

R. Manekar is with BITS Pilani, K.K. Birla Goa Campus, Goa, India, 403726 (Email: raunakm@goa.bits-pilani.ac.in)

Predictors [9] and clothoids [10,11] have also been used to predict the future position of the ego vehicle.

Some researchers have also used image transformation techniques based on real-time throttle/brake/steering inputs from the teleoperator [12,13]. Recently, a more rigorous estimation-based PD system was developed in a MATLAB-Unreal Engine co-simulation environment to evaluate the teleoperation performance of human subjects [5]. However, none of the authors have so far been able to implement their PD system on real-world camera images.

One way to create PD system for AV teleoperation is to by formulating it as a video prediction task by directly trying to predict the future frames of the video stream from the past frames. Generative models such as GANs, VAEs and most recent diffusion models have shown remarkable capabilities for image and video generation tasks. However, these models often struggle with videos featuring complex motion such as those found in urban environments encountered in autonomous driving datasets. Furthermore, the large inference times for diffusion models have prevented their widespread use in real-time applications such as autonomous driving [14]. In such scenarios, methods based on deep learning based optical flow [15,16,17] are more popular. Such methods require the input of a sequence of delayed camera images and can then predict the future frames. However, such generative models suffer from generalization issues required for real life driving scenarios [18].

In contrast, the approach presented in this paper, where image synthesis is performed using a combination of state estimation and generative models, is much more generalizable. Furthermore, deep learning-based methods are unable to effectively handle scenarios involving ego vehicle turning and tend to generate random images during such maneuvers. On the other hand, the inpainting pipeline introduced in this work can effectively handle such scenarios.

The estimation-based PD system relies on accurate state estimation of the ego vehicle for which EKF has been used which relies on an accurate vehicle dynamic model along with IMU and GPS measurements. This method is especially suitable for teleoperation as it relies on sending low size IMU and GPS data with very little delay. Advanced optical flow and scene estimation methods [19] can also be used to estimate the states of the ego vehicle, but such methods rely on a sequence of camera images. Since the camera images themselves have huge delays (of the order of 500 milliseconds), the state estimation using these methods will in turn be further delayed resulting in large errors. Thus, such methods although suitable for estimating the camera pose of ego vehicle are not suitable in the context of teleoperation. Furthermore, computing optical flow or scene flow using deep learning methods can't be used in a generalized framework for unknown roads. The proposed EKF based state estimation on the other hand can provide accurate state estimates across varied environments. Nonlinear observers, in lieu of EKFs, have also been used for vehicle related nonlinear estimation applications [20].

The primary contributions of this paper are as follows:

1) The paper develops an estimation-based predictive display (PD) system that reconstructs the 3D environment around the ego-vehicle using a deep learning-based meshing algorithm based on the sensor fusion of delayed camera and Lidar data.
2) This paper further uses a dynamic vehicle model proposed in this paper together with an extended Kalman Filter to perform predictive modification of the reconstructed old 3D scene using faster updates from GPS and IMU sensors. Furthermore, the system can also handle GPS denied environments by utilizing measurements from Lidar based SLAM methods.
3) This paper also presents a novel pipeline to improve the image quality of the PD images using deep learning-based image inpainting. Furthermore, to tackle ego-vehicle turning the images from the front and side cameras are combined using the developed deep learning-based meshing algorithm.
4) The paper further develops a deep learning-based image comparison metric for AV teleoperation for evaluating the accuracy of object detection and localization.
5) This paper presents for the first time the application of PD on experimental real-world AV data, clearly pointing out the benefits of using such algorithms to compensate for delayed camera data. In this paper, the open-source KITTI and nuScenes datasets are used for analysis.

The outline of the rest of the paper is as follows. In section II, the estimation-based PD algorithm is described which includes state estimation, 3D reconstruction and image-inpainting. Section III describes a new metric for image comparison based on object detection and localization. Section IV discusses the results using various image metric comparison for the developed algorithms and proves the efficacy of estimation-based PD system on real-world data. Section V contains the conclusions.

II. Estimation Based Predictive Display

Latency due to wireless transmission can affect AV teleoperation significantly, and even a 0.5 s delay can highly deteriorate the lateral and longitudinal control performance of the remote teleoperator [5]. Hence, it is vital to reduce latency and provide accurate intermediate visual updates to the teleoperator. This section describes the PD system aimed at enhancing AV teleoperation by recreating the 3D environment around the vehicle using delayed camera and Lidar data and updating this image based on state estimates of the Ego vehicle. The given PD system relies on GPS and IMU sensors for state estimation and the front camera and Lidar for 3D reconstruction. Furthermore, image inpainting has also been used to improve the quality of the generated images.

A. State Estimation of Ego Vehicle

The accurate position and yaw angle of the ego vehicle are critical variables for accurately transforming the delayed camera images. Hence, ego-vehicle state estimation is an integral part of the PD system.

In this work the inertial position of the ego vehicle is estimated which requires the use of Inertial and Ego frames.

The Inertial Frame $\{I\}$ is a global fixed frame which is stationary and hence has no linear or angular rates. The GPS readings as well as the state vector are defined with respect to this frame. The origin of the Inertial Frame is located at O_I and the basis vectors are given by x_I, y_I and z_I as shown in Fig. 1. The Ego Frame $\{E\}$ is located on the center of mass (CoM) of the ego vehicle and moves with the ego vehicle. The IMU readings are defined with respect to this frame. The origin of the Ego Frame is located at O_E and the basis vectors are given by x_E, y_E and z_E. The position of the CoM of the ego vehicle in the inertial frame is given by x and y and the yaw angle is given by ψ. Let, the state vector X be,

$$X = [x \quad y \quad \dot{x} \quad \dot{y} \quad \psi]^T \tag{1}$$

where $\dot{x}$ and $\dot{y}$ are the velocity of the CoM of the ego vehicle in the inertial frame. The IMU provides the acceleration and the yaw rate of the CoM of the ego vehicles and its measurements are given as follows,

$$u = \begin{bmatrix} a_x & a_y & \dot{\psi} \end{bmatrix}^T \tag{2}$$

where, the accelerometer reading about the x_E and y_E axis is given by a_x and a_y respectively, and the yaw rate provided by the gyroscope is given by $\dot{\psi}$. Due to the presence of biases the IMU reading differs from that of the true accelerations and rotational rates and the noisy and biased measurements are given as follows,

$$\begin{bmatrix} a_x \\ a_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ \dot{\psi} \end{bmatrix}_t + \begin{bmatrix} a_x \\ a_y \\ \dot{\psi} \end{bmatrix}_{bt} + \begin{bmatrix} a_x \\ a_y \\ \dot{\psi} \end{bmatrix}_n \tag{3}$$

where, $[\,]_t$ are the true readings, $[\,]_{bt}$ are the constant biases and $[\,]_n$ are zero mean noise signals with constant standard deviation. The various frames of reference as well as IMU inputs are shown in Fig. 1.

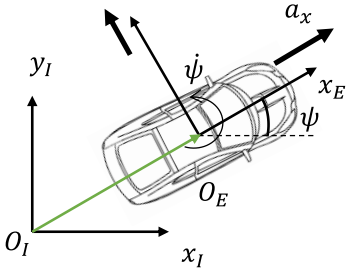


Fig. 1. Inertial and Ego Frames for state estimation

The state dynamics for the ego vehicle are as follows,

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ a_x \cos(\psi) - a_y \sin(\psi) \\ a_x \sin(\psi) + a_y \cos(\psi) \\ \dot{\psi} \end{bmatrix} = f(X, u) \tag{4}$$

The state dynamics is nonlinear in nature and due to the presence of biases in the measured IMU signals, direct use of the dynamic model will result in a drift error over time. Hence, there is a need to use additional GNSS measurements for accurate ego vehicle localization. In Lidar or camera based odometry methods there is a large bandwidth requirement which adds further delays to state estimation these are thus infeasible to use in teleoperation. On the other hand, GNSS has very low data size and is thus a feasible option for teleoperation. In this work, real-world data which uses GNSS with a specified accuracy of 10 cm has been used. The measurement equation used for estimating the state vectors is given as follows,

$$\boldsymbol{y} = [x \quad y \quad \psi]^T + v \tag{5}$$

where, v is zero mean gaussian white noise. The measurement of the yaw angle can be obtained from a dual antenna GPS device or for small slip angles can be approximated to be equal to the heading angle of the vehicle. Accurate measurement of GPS and yaw angle for both KITTI and nuScenes dataset were available and were used for accurate state estimation. Given the nonlinear state dynamics of (4) and the measurements of (5), the Extended Kalman Filter (EKF) was used for state estimation of the ego vehicle [21]. The EKF performs prediction and correction to obtain accurate state estimates. In the prediction step the apriori state estimates $\bar{x}_k^-$ are obtained using the system's nonlinear dynamics along with the apriori state covariance matrix P_k^- as follows,

$$\bar{x}_k^- = f(\bar{x}_{k-1}^+, u_{k-1}) \tag{6}$$

$$P_k^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \tag{7}$$

where, $F_{k-1} = I_5 + TA_{k-1}$,

$A_{k-1} = \frac{\partial f}{\partial x}(\bar{x}_{k-1}^+, u_{k-1}) =$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -a_x \sin(x_5) - a_y \cos(x_5) \\ 0 & 0 & 0 & 0 & a_x \cos(x_5) - a_y \sin(x_5) \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

where, I_5 is 5×5 identity matrix, T is the time step and Q_{k-1} is the process noise covariance matrix.

In the correction step the apriori estimates and state covariance are corrected using the measurements y_k and the Kalman gain K_k to obtain aposterior estimates $\bar{x}_k^+$ and state covariance P_k^+ as follows,

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{8}$$

$$\bar{x}_k^+ = \bar{x}_k^- + K_k(\boldsymbol{y_k} - H_k \bar{x}_k^-) \tag{9}$$

$$P_k^+ = (I_5 - K_k H_k) P_k^- \tag{10}$$

where, $H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$ and R_k is the measurement noise covariance matrix. It has been assumed that there is no latency in the transmission of GPS and IMU data due to their much lower data size compared to camera and Lidar data.

However, there are instances where GPS data may have poor accuracy or may be completely lost. For example, in urban areas tall buildings and other structures can block the line of sight to GPS satellites resulting in loss of GPS signals. In such cases it is important to use the EKF based filter with other localization techniques like visual SLAM or Lidar SLAM.

In the context of teleoperation, the SLAM based localization algorithm cannot run on the teleoperation station due to huge delay in image and Lidar data. Therefore, such methods must be able to run on the remote vehicle and the localization data (which includes the positions and the yaw angle) can then be transmitted to the teleoperation station. This localization data can essentially replace the measurement given by Eq. (5) for the EKF thus allowing for improved localization even in GPS denied regions. Compared to monocular camera-based SLAM, Lidar based SLAM are more suitable for outdoor environment [22] and hence, in this work two well-known methods are used. The first one is KISS-ICP [23] which is a simple, accurate and robust Lidar odometry and the second one is FAST-LIO [24] which is a robust and computationally efficient for Lidar-inertial odometry which integrates IMU data along with Lidar data. Both the methods has reduced computational load such that both can easily run at 10 Hz. Thus, the new measurement equation for the EKF becomes,

$$\boldsymbol{y} = \begin{cases} [x \quad y \quad \psi]^T|_{GPS}, & GPS\ available \\ [x \quad y \quad \psi]^T|_{SLAM}, & GPS\ outage \end{cases} \tag{11}$$

One of the benefits of this integrated localization filter is that the measurement model is independent of the SLAM algorithm and hence can accommodate any kind of future SLAM based algorithms which may provide more accurate and real-time measurements.

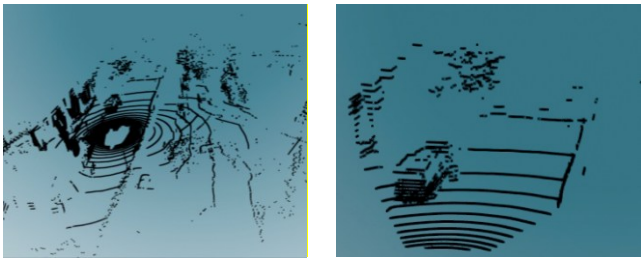
B. 3D Reconstruction

Updating the delayed camera images requires the current ego state estimates as well as an accurate 3D representation of the environment around the ego vehicle to generate new images based on the estimated position of the ego vehicle. The point cloud obtained from the delayed Lidar data is used to create a 3D mesh for the environment around the ego vehicle and the image synthesis is done based on this mesh.

In this paper, the images generated from a monocular camera are considered (i.e. from the front camera) along with the 360^0 point cloud data from Lidar. The image synthesis pipeline has four major key processes:

1. Point cloud filtering
2. Mesh creation
3. Raycasting
4. Camera motion using state estimation

In the first process the 360^0 point cloud is reduced to just the points in the field of view of the camera. This allows for decreasing the size of the point cloud data thus decreasing the delay as well as speeding up 3D reconstruction. The points in the Lidar frame were first transformed to the camera frame. The Lidar points which were behind the camera were removed and only those points which were in the front of the camera were projected onto the image plane. Among these points, those points whose pixels were outside the image were removed and the Lidar points corresponding to the pixels inside the image were obtained and the normal vectors for each of these points were computed using Open3D. These points, which are called filtered points were then used for generating the 3D mesh. The Lidar point cloud and the filtered points for a sample nuScenes dataset are shown in Fig. 2.



(a) Lidar point cloud (b) Filtered Lidar points

Fig. 2. Lidar point cloud and filtered points

In the second process two different methods were explored to create a 3D mesh for the filtered points. The first method used was Poisson Surface Reconstruction (PSR) which is a classical method relying on mathematical and algorithmic techniques to generate a surface from point cloud data [25]. It aims to fit a watertight, triangulated mesh for a given input of points and the normal of each point. In this algorithm an indicator function χ which has a value less than 1 outside the mesh and greater than 1 inside, is computed by solving the Poisson problem given as follows,

$$\nabla^2 \chi = \nabla \cdot \vec{V} \tag{12}$$

where, $\vec{V}$ is the vector field for the surface normal. PSR is aimed at generating closed surfaces, and in many test cases it created unnatural closure like an artificial "lid" over the scene, closing off the open environment in an unnatural way. Because of this closed mesh it was not possible to perform accurate image synthesis. This problem will be further explained in Section IV.

Neural Kernal Surface Reconstruction (NKSR) was then employed to counter this disadvantage of PSR. NKSR employs neural networks and kernel methods to capture complex surface details and model intricate geometries and fine details. Unlike PSR, which forces a closed surface, NKSR can handle open structures more naturally, which is particularly beneficial for outdoor scenes where open boundaries and partial structures are common.

NKSR is trained on the data from a CARLA simulator and the trained model has been successfully used in real outdoor environment also [26]. Although NKSR is able to handle open boundaries, often it produces black patches in the image due to non-meshed regions.

In the third process, raycasting using Open3D was employed to cast rays from the camera (of the delayed camera image) onto the mesh to obtain the corresponding colors of the triangles on the mesh. Raycasting requires the input of the camera intrinsic matrix K and the transformation matrix from the Lidar frame to the Camera frame. Given the pixel triangle correspondence, each triangle was assigned the color according to its pixel correspondence. Those triangles that spanned multiple pixels were assigned the same color, which was the mean for all the pixels to which it corresponded.

The transformation matrix of frame $\{B\}$ with respect to

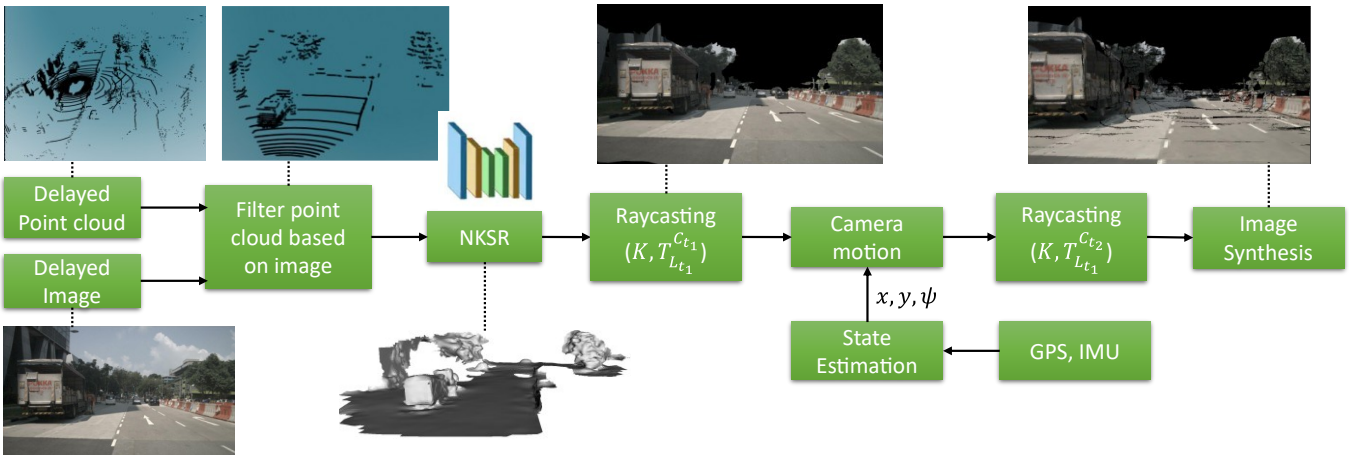


Fig. 3. Image synthesis pipeline for estimation-based PD using 3D reconstruction

frame $\{A\}$ is given by $T_B^A \in \mathbb{R}^{4\times4}$ and this convention is followed throughout this paper. Consider the Lidar and camera frame at time t to be $\{L_t\}$ and $\{C_t\}$. The delayed image and Lidar data is obtained for time t_1 and the image is synthesized for time $t_2 = t_1 + \Delta t$. To synthesize this image, it is required to know $T_{L_{t_1}}^{C_{t_2}}$ which can be computed as follows,

$$T_{L_{t_1}}^{C_{t_2}} = T_{E_{t_2}}^{C_{t_2}} T_{E_{t_1}}^{E_{t_2}} T_{L_{t_1}}^{E_{t_1}} \tag{13}$$

where, the matrices $T_{E_{t_i}}^{C_{t_i}}$ and $T_{E_{t_i}}^{L_{t_i}}$ are computed based on sensor locations and are fixed $\forall\, t_i$. The transformation matrix from the ego frame at time t_1 to the ego frame at time t_2, $T_{E_{t1}}^{E_{t2}}$ is given as follows,

$$T_{E_{t1}}^{E_{t2}} = \left(T_{E_{t2}}^{I}\right)^{-1} T_{E_{t1}}^{I} \tag{14}$$

The transformation matrix from the Ego frame at time t_1 to the Inertial Frame can be obtained from the state estimates of the ego vehicle and is given by,

$$T_{E_{t_1}}^{I} = \begin{bmatrix} cos(\psi(t_1)) & -sin(\psi(t_1)) & 0 & x(t_1) \\ sin(\psi(t_1)) & cos(\psi(t_1)) & 0 & y(t_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

Using (14), the expression $T_{E_{t1}}^{E_{t2}}$ can be obtained as follows,

$$T_{E_{t1}}^{E_{t2}} = \begin{bmatrix} cos(\Delta\psi_E) & -sin(\Delta\psi_E) & 0 & t_x \\ sin(\Delta\psi_E) & cos(\Delta\psi_E) & 0 & t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

where, $\Delta\psi_E = \psi(t_1) - \psi(t_2)$,

$t_x = \cos\psi(t_2)\left(x(t_1) - x(t_2)\right) + \sin\psi(t_2)\left(y(t_1) - y(t_2)\right)$

$t_y = \cos\psi(t_2)\left(y(t_1) - y(t_2)\right) + \sin\psi(t_2)\left(x(t_1) - x(t_2)\right)$

From (15), it can be observed that image synthesis requires the knowledge of both the delayed time t_1 and the time delay Δt (which gives $t_2 = t_1 + \Delta t$). Once these variables are known image synthesis can be performed for both constant delay as well as variable delay cases.

Furthermore, the PD system can also be used to handle cases when the camera and Lidar data are not synchronized. Consider the case when the Lidar data is received at time $t_1{}'$ and the camera data is received at time t_1, in such a case the point cloud obtained at time $t_1{}'$ can be transformed to time t_1 as follows,

$$p(t_1) = T_{E_{t_1}}^{L_{t_1}} T_{E_{t_1}{}'}^{E_{t_1}} T_{L_{t_1}{}'}^{E_{t_1}{}'} p(t_1') \tag{17}$$

where, $p(t_1)$ is the filtered point cloud at time t_1. This transformed point cloud is then used to create the mesh and then (12) can be used to synthesize images. Since the GPS and IMU have very small data size, hence it is assumed that they have negligible delay and hence the use of the estimation-based PD can counter the effects of high constant delay, variable delay and even asynchronized sensor use.

Given the new transformation of Lidar frame $\{L_{t_1}\}$ w.r.t new camera $\{C_{t_2}\}$ in (12), another raycasting was performed using camera intrinsic matrix K and $T_{L_{t_1}}^{C_{t_2}}$ to obtain the new triangles corresponding to the pixels at the new camera pose. Thus, an image based on the estimated position of the ego vehicle was generated. The entire image synthesis pipeline using 3D reconstruction is shown in Fig. 3. From the figure one can observe that the mesh created via NKSR has many open spaces (white regions in the mesh) which results in many black patches in the image due to the presence of unmeshed regions. Also, due to the presence of uncolored triangles in the mesh, moving the camera in the 3D environment further creates more black regions in the synthesized image.

It was observed that if the size of triangles in the mesh is large, then blurry images are generated due to the correspondence of many pixels with one triangle, thus creating an image having effects similar to Gaussian blurring. Hence, after constructing the 3D mesh, the triangles were subdivided into smaller triangles to synthesize sharp and more realistic images. However, as the size of the triangles decreases, there are many uncolored triangles, which results in even more black patches (apart from unmeshed regions) in the synthesized image. Image inpainting was done to fill up these black patches, which will be described in the following subsection.

C. *Image Inpainting*

The image obtained after the second raycasting in the

NKSR/PSR mesh contains three kinds of black patches:

1. Large black patches are present due to the presence of unmeshed regions generated due to NKSR meshing.
2. Small black patches which are present due to the presence of unmeshed regions generated due to NKSR meshing.
3. Small black patches which are present due to the presence of uncolored triangles in the mesh.

The three kinds of black patches are shown in Fig. 4 where the green region represents the first kind of black patch which is always obtained on the top part or side part of the synthesized image and is a result of lack of Lidar points in such areas. The blue regions represent the second kind of black patches which are a result of unmeshed regions in the environment generated due to sparse Lidar points in the data. The red regions represent the third kind of black patch which

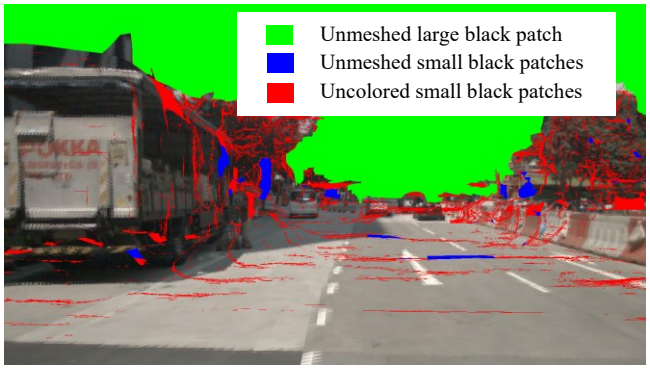


Fig. 4. Three kinds of black patches for image inpainting

are generated due to uncolored triangles in the mesh generated due to moving the camera in the 3D environment.

To improve the quality of the synthesized image it is very important to fill these black patches and hence for this purpose image inpainting is used. The first kind of black patches are bound to appear in regions when the top part of the camera (such as the sky) is not captured at all by the Lidar point cloud. Such black patches present an outpainting problem rather than an inpainting problem. When these regions are very far away and do not change much with ego motion, they can be replaced with the corresponding parts of the delayed image, and this is what is being done in this work. Otherwise, it is very important to utilize deep learning-based image outpainting to fill these regions. However, this is out of the scope for this paper and is a topic for further research.

The second and third kind of black patches can be filled by applying image inpainting. One of the simplest methods to inpaint is to use the color of the nearest colored triangle and this method in this paper is being referred to as nearest inpaint. With this method one can fill up smaller black patches but it becomes very difficult to fill up larger regions. Hence for this purpose the state-of-the-art large hole inpainting Mask-Aware Transformer (MAT) is being used [27]. This deep learning network utilizes the merits of transformers and convolutions to efficiently process high-resolution images. MAT is provided with the synthesized image and the mask for the black patches and it inpaints the black regions, thus improving the quality image quality which is then displayed to the teleoperator.

Furthermore, during situations involving turning of the vehicle the delayed image from the center front camera is not able to capture the entire scene. In such cases the delayed camera feed from side cameras can be utilized along with the NKSR method to fill the otherwise black regions encountered during the turning of the ego vehicle. When the ego vehicle is turning (determined using the yaw rate), the point cloud projected on the side camera along with the a part of the image from the side camera (one fourth of the camera image) can be sent to the teleoperator. Using these two as inputs, the NKSR pipeline shown in Fig. 3 is applied to synthesize side camera image at the new ego position which can be effectively used to inpaint the side large black patches during the turning of the ego vehicle. Hence the use of additional camera measurements effectively inpaints the unknown black regions and increases the system's effectiveness to handle turns where even the state-of-the-art deep learning methods fails. The entire image inpainting pipeline is shown in Fig. 5. Algorithm 1 describes the estimation-based PD system utilizing state estimation, NKSR and image inpainting.

Algorithm 1: Estimation-based PD with image inpainting

Require: Predictive Display ← f(Camera Matrix P, IMU, GPS, Lidar point cloud X_L, Camera image I_C, Current time t_2, Delayed time t_1, $T_{E_{t1}}^{C_{t1}}$ and $T_{E_{t1}}^{L_{t1}}$)

1. **while** The system runs **do**
2. ▷3D Reconstruction:
3. Filtered Point Cloud, $\bar{X}_L \in \{X_L | \, x = P(T_L^C X_C)_{front} \in I_C\}$
4. mesh, S = NKSR($\bar{X}_L$)
5. Colored triangles, S_c = RayCasting($S, P, T_{L_{t1}}^{C_{t1}}$)
6. $T_I^{E_{t1}}$ = StateEstimation($IMU(t_1), GPS(t_1)$)
7. $T_I^{E_{t2}}$ = StateEstimation($IMU(t_2), GPS(t_2)$)
8. $T_{L_{t1}}^{C_{t2}} = T_{E_{t2}}^{C_{t2}} T_I^{E_{t2}} T_{E_{t1}}^{I} T_{L_{t1}}^{E_{t1}}$
9. New triangles, S'_c = RayCasting($S, P, T_{L_{t1}}^{C_{t2}}$)
10. Predicted image, $I'_C = \{S'_c | \, S'_c \in S_c\}$
11. ▷Image Inpainting:
12. Mask for uncoloured small black patch, $M_1 = \{S'_c | \, S'_c \notin S_c\}$
13. Mask for other patches, $M = \{I'_c | I'_c - M_1 \in (0,0,0)\}$
14. Mask for unmeshed large black patch, $M_2 = \{M | M$ has largest area$\}$
15. Mask for unmeshed small black patch, $M_3 = \{I'_c | I'_c - M_1 - M_2 \in (0,0,0)\}$
16. Inpainted Image, I''_c=MAT(I'_C, M_2, M_3) + $\{I_C | I_C \in M_1\}$
17. **end while**
18. **Result:** PD using state estimation of ego vehicle.

III. New Image Comparison Metric For Teleoperation Studies

In the context of evaluating the generated image quality, various pixel comparison metrics like Peak Signal to Noise ratio (PSNR), Structural Similarity Index (SSIM), Multi-Scale Structural Similarity Index (MS-SSIM) and Feature Similarity Index (FSIM) can be used. However, such metrics suffer from the problem of lack of contextual understanding and cannot differentiate between important and unimportant parts of image regions for teleoperation. Furthermore, they do

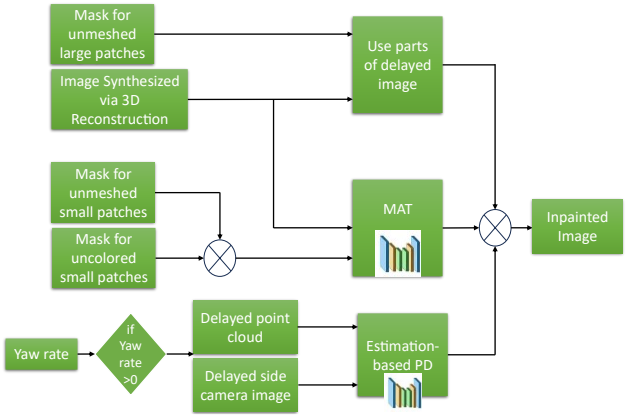


Fig. 5. Image inpainting pipeline

not consider high-level semantics, such as the presence of objects and their relationships to the ego vehicle, which are crucial for human perception and teleoperation. These existing metrics can provide some insights into image quality, but their limitations make them less than ideal for the specific needs of autonomous vehicles (for example, teleoperation).

Deep learning metrics like LPIPS are generally used to compare the perceptual similarity between images, often in generative models, image compression, or style transfer. However, for AV applications (especially teleoperation), the primary focus is on accurate nearby object detection and localization. High LPIPS scores indicate good perceptual quality but do not guarantee improved detection performance. The primary concern for AV systems is the accuracy and reliability of object detection, not just image quality. Also, the effectiveness of LPIPS depends on the specific pre-trained neural network used. The quality assessment might be less reliable if the pre-trained model does not generalize well to the diverse scenarios AV systems encounter. Hence, a new metric must be developed to evaluate the performance of PD algorithms.

One of the most critical requirements to teleoperate a vehicle successfully is to accurately know the position of various objects in the environment. Hence, accurate object detection is a key ingredient for successful teleoperation, and the synthesized images from a PD algorithm must be able to cater to this need. Deep learning-based object detection algorithms like YOLO can detect various objects like vehicles and pedestrians in undelayed images. However, when multiple such objects are present, finding them and comparing their overlap in a test image is important. This test image can be either the delayed image or the image synthesized using PD. The proposed new Teleoperation Object Location Metric (TOLM) is aimed at this task. This new metric relies on accurately detecting and comparing objects from undelayed and test images, matching the corresponding boxes and computing the Intersection Over Union (IoU) for the two boxes. The computation of the TOLM metric can be divided into three parts:

1. Object detection.
2. Feature Matching
3. IoU computation

For detecting various objects in the undelayed and test image, YOLO v8 [28] is being used. Ultralytics YOLO v8 is the latest version of a well-known real-time object detection and image segmentation model and offers good performance in terms of speed and accuracy.

Once the bounding boxes for the various objects in the undelayed and test image are obtained, each of the bounding boxes in the undelayed image is then compared to all the bounding boxes obtained from the test image using SIFT feature matching. This step aims to obtain the correspondence of each bounding box in the undelayed image with a box in the test image for accurate object localization. SIFT identifies invariant features across scale, rotation, and translations and can hence be used for robust feature detection and matching. For each bounding box in the undelayed image, a region of interest (RoI) is obtained (based on the dimension of the bounding box) for which key points are located using SIFT, and a descriptor for each key point is also obtained. The same is done for all the RoI in the test images, and the descriptors are compared using ratio test. The correspondence between the two boxes is established when the number of good matches is greater than four. Each corresponding bounding box in the undelayed and test image IoU is computed which is an indicator of object localization in the image. An IoU of 1 corresponds to a complete overlap of the two bounding boxes, and an IoU of 0 corresponds to no overlap. IoU for each of the bounding boxes in the undelayed image is computed and the TOLM is obtained as follows,

$$TOLM = \frac{\sum_{i=1}^{n} IoU_i}{n} \tag{18}$$

where, n is the number of bounding boxes present in the

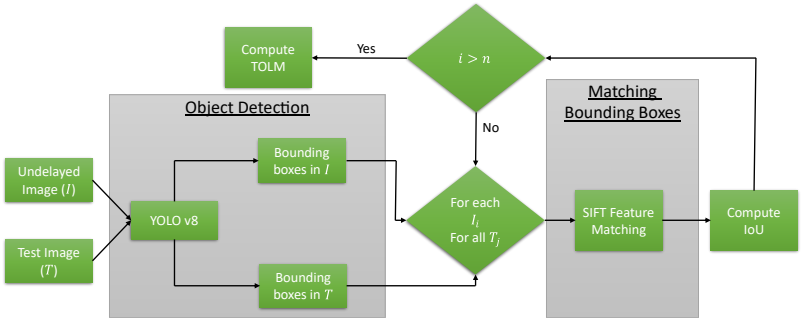


Fig. 6. Pipeline for TOLM

undelayed image. The complete pipeline for this new image comparison metric is given in Fig. 6.

IV. Results

To evaluate the performance of the estimation-based PD system on real world experimental data, two open-source datasets KITTI and nuScenes have been used. This section describes the various results obtained on these two datasets. The results obtained from estimation-based PD using 3D reconstruction are compared with those obtained from the delayed display feed using image comparison metrics like PSNR, SSIM, FSIM and MS-SSIM. Furthermore, the results are also compared using TOLM to evaluate the performance of PD for object detection and localization.

TABLE I. Sensor Specification of Ego Vehicle

Sensor Specification	KITTI	nuScenes
GPS rate (Hz)	5	50
IMU rate (Hz)	10	100
Camera display rate (FPS)	10	12
Image size (pixels)	1242×375	1600×900
Lidar rate (Hz)	10	12
Lidar vertical field of view (deg)	26.8	41.33
Lidar accuracy (cm)	2	2

A. Sensor Rates and State Estimation

The details of the various sensors for both KITTI and nuScenes datasets that were used for estimation-based PD is given in Table I. From the table it can be observed that the GPS and IMU sensor rates for the KITTI dataset are lower than the nuScenes dataset, which allows the evaluation of the state estimation algorithm on varied datasets. Furthermore, the camera and Lidar rates for both datasets are almost similar. However, the image size and Lidar vertical field of view are different for both datasets, which allows effective evaluation of PD for assorted datasets.

TABLE II. Ego-Vehicle State Estimation Results

Error	KITTI		nuScenes	
	RMSE	Max error	RMSE	Max error
$\tilde{x}$ (m)	0.03	0.11	0.029	0.145
$\tilde{y}$ (m)	0.03	0.14	0.042	0.146
$\tilde{\psi}$ (deg)	0.22	0.67	0.13	0.59

The state estimation results for the two datasets are provided in Table II where $\tilde{x}, \tilde{y}$ and $\tilde{\psi}$ is the error in estimated position and yaw angle. From the table, it can be observed that for both datasets, the RMSE for position is less than 4 cm, and for yaw angle, it is less than 0.25 deg. The maximum error for the position is less than 15 cm, and that of the yaw angle is less than 0.7 deg. Hence, state estimation based on GNSS can provide accurate estimates for both position and yaw angle for image synthesis using 3D reconstruction.

B. PSR Vs NKSR

The image synthesized using PD depends on the accuracy of the state estimates and the quality of the mesh generated using 3D reconstruction. Two methods, PSR and NKSR, were implemented to create a mesh and will be analyzed in this subsection. Often, the mesh generated by PSR is such that there is an artificial lid over the scene, blocking the viewpoint because of which the synthesized image is completely black. Sometimes, the mesh in PSR bulges up, which also partially blocks the viewpoints. Fig 7 shows the artificial lid that was generated in the KITTI dataset, thus creating a completely black image at around 0.2 s. On the other hand, the image synthesized using NKSR is consistent over the whole time period for both KITTI and nuScenes data.

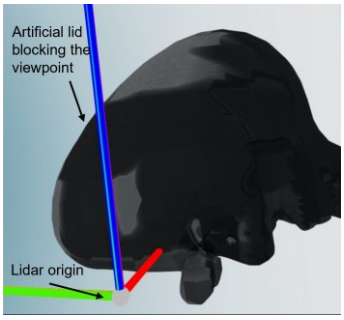


Fig. 7. Artificial lid generated using PSR for KITTI dataset

TABLE III. NKSR and PSR Comparision

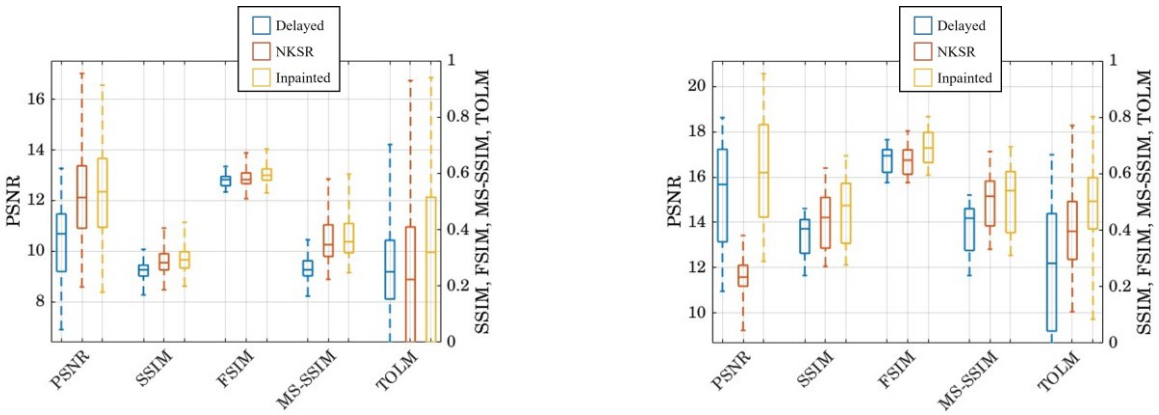
Metric	KITTI				nuScenes			
	PSR		NKSR		PSR		NKSR	
	μ	σ	μ	σ	μ	σ	μ	σ
PSNR	10.54	2.00	10.54	0.7	11.60	2.00	10.40	0.67
SSIM	0.20	0.07	0.19	0.03	0.23	0.08	0.25	0.04
FSIM	0.49	0.04	0.49	0.04	0.57	0.04	0.58	0.02
MS-SSIM	0.26	0.07	0.29	0.06	0.34	0.06	0.33	0.04

Table III shows the numeric comparison of PSR and NKSR results for the two data sets. In this table μ and σ refers to the mean and standard deviation for a given image metric. It can be observed that although the mean value of PSR and NKSR are almost same, PSR has larger standard deviation thus indicating the variability of the results obtained from PSR.

C. Image Inpainting

The use of NKSR for image synthesis leads to many black patches in the image. To fill these, image inpainting using MAT has been used. The results obtained are for three delay cases: 0.5 s, 1 s, and 0.7 s/0.75 s for both KITTI and nuScenes datasets. For the KITTI dataset, 0.7 s delay was considered, and for nuScenes, 0.75 s delay was considered. In all the subsequent plots and tables, 'NKSR' and 'Inpainted' refer to the inpainting of image synthesized using NKSR with the nearest inpaint and with MAT, respectively.

The box plot for various metrics for KITTI and nuScenes data for 0.5 s delay is shown in Fig. 8. The plot shows the PSNR values (greater than 1) on the left y-axis and other metrics values (ranging from 0 to 1) on the right y-axis. From the plot, it is clear that image synthesis using NKSR is greater for almost all the metrics as compared to delay cases. Also,



(a) KITTI dataset (b) nuScenes dataset

Fig. 8. Box plot for 0.5 s delay

TABLE IV. IMAGE METRIC COMPARISION FOR KITTI DATASET

Metric	PSNR ↑			SSIM ↑			FSIM ↑			MS-SSIM ↑			TOLM ↑		
Delay	0.5 s	0.7 s	1.0 s	0.5 s	0.7 s	1.0 s	0.5 s	0.7 s	1.0 s	0.5 s	0.7 s	1.0 s	0.5 s	0.7 s	1.0 s
Delayed	10.39	9.84	9.29	0.25	0.23	0.22	0.58	0.56	0.55	0.26	0.23	0.20	0.25	0.17	0.10
NKSR	12.28	**11.56**	**10.82**	0.29	0.27	0.25	0.58	0.57	0.55	0.36	0.32	0.28	0.26	0.18	0.14
Inpainted	**12.35**	11.53	10.74	**0.30**	**0.28**	**0.26**	**0.60**	**0.58**	**0.57**	**0.38**	**0.33**	**0.29**	**0.33**	**0.25**	**0.20**

TABLE V. IMAGE METRIC COMPARISION FOR NUSCENES DATASET

Metric	PSNR ↑			SSIM ↑			FSIM ↑			MS-SSIM ↑			TOLM ↑		
Delay	0.5 s	0.75 s	1.0 s	0.5 s	0.75 s	1.0 s	0.5 s	0.75 s	1.0 s	0.5 s	0.75 s	1.0 s	0.5 s	0.75 s	1.0 s
Delayed	15.22	14.56	14.03	0.38	0.36	0.35	0.65	0.64	0.63	0.41	0.39	0.37	0.26	0.19	0.15
NKSR	11.67	11.40	11.16	0.44	0.39	0.37	0.65	0.63	0.62	0.50	0.46	0.43	0.41	0.30	0.24
Inpainted	**16.37**	**15.29**	**14.66**	**0.47**	**0.43**	**0.40**	**0.70**	**0.66**	**0.64**	**0.51**	**0.46**	**0.43**	**0.48**	**0.36**	**0.29**

image synthesis using NKSR inpainted with MAT gives better results for all metrics than delayed image and NKSR. It can be easily observed that the TOLM metric is higher for inpainted images, indicating that deep learning-based inpainting methods improve object detection and localization accuracy. The improvement in PSNR shows less noise in the synthesized image, while that in SSIM indicates better structural similarity of synthesized with ground truth. Increased MS-SSIM for inpainted synthesized images indicates better similarity with ground truth over multiple resolutions. Improvement in the FSIM metric indicates that the synthesized image has better perceived quality, especially in terms of edges and textures, which are significant for human visual perception.

Tables IV and V show the Image metrics and TOLM comparison for KITTI and nuScenes datasets for various delay cases. The tables show the mean value of the various metrics over the whole-time interval. From the table, it is clear that in almost all the metrics, the performance of inpainted image synthesis is the best. The only exception is the PSNR values in Table IV for 0.7 s and 1 s delay cases, where the performance of NKSR is better.

It is very important to note that the value of the TOLM index for inpainted image synthesis is about double that of delayed image feed, which clearly indicates that the estimation-based PD system with image inpainting is able to increase the object detection and localization performance by 50 % as compared to delayed cases. The same trend is observed for both datasets, which proves the generality of the algorithm over various driving conditions.

D. Qualitative Results

The qualitative result for a sample time for both scenes and KITTI dataset for 0.5 s delay is shown in Fig. 9. From the figure, it is clear that the estimation-based PD can recreate images that match the ground truth. For the nuScenese dataset, the black Sedan on the right is partially visible in the ground truth, NKSR, and inpainted but fully visible in the delayed image. In the KITTI dataset, a black car is visible in the delayed image but not in ground truth, NKSR, and inpainted images. This fact points out that using delayed images for teleoperation results in incorrect perception of reality, and the estimation-based PD algorithm with deep learning-based image inpainting is able to create realistic images, which compensates for teleoperation delay. However, it can be observed that PD cannot capture the high-frequency components (for example the logs in the wooden fence as shown in Fig. 9 for KITTI dataset) in the image due to limitations on the mesh quality and raycasting, although it can accurately capture nearby buildings and vehicles on the road.

The qualitative results for the TOLM metric are shown in Fig. 10 for a sample time in the KITTI dataset. The results are for the 0.5 s delay case where the figure shows the delayed image, ground truth, NKSR results, and inpainted results. From the figure, it is clear that the car in the delayed image is far away from the ego vehicle compared to the ground truth, which is why the bounding box for the delayed image is smaller than the ground truth bounding box. On the other hand, for NKSR and inpainted cases, the bounding boxes for

(a) nuScenes Dataset (b) KITTI Dataset

Fig. 9. Qualitative results on nuScenes and KITTI datasets

the car are similar to that in the ground truth image. However, due to the artifacts created using the nearest inpainted method in NKSR, there is an additional bounding box for a person in the image. But using deep-learning-based inpainting methods, no such artifacts are present in the inpainted image, thus allowing for a more realistic image synthesis of the ground truth. For this sample time, the TOLM index for the delayed image was 0.3, while for NKSR and inpainted, it was 0.8 and 0.87, respectively, which proves the significant superiority of estimation-based PD and that of inpainting using deep learning.

E. Robustness against Variable Time Delays

In this subsection the estimation-based PD system is validated against real-time transmission issues like delays in sending GPS and IMU data and the occurrence of variable time delays for sending camera data. Due to certain network, constraints sometimes the transmission of even the GPS and IMU data may encounter brief delays of the order of 100 milliseconds. Hence, the performance of the PD system in such cases is evaluated. Table VI presents the perk of the PD system when there is a delay of 10 ms and 100 ms in the

Delayed NKSR

Ground Truth Inpainted

Fig. 10. Qualitative results for TOLM metric

transmission of GPS and IMU data for the nuScenes dataset. From the table it can be observed that the PD system has comparable performance to the no delay case although there is a slight increase in error and a slight decrease in PSNR. Although the metrics decrease a bit, the system is largely at par with the no delay case.

TABLE VI. EFFECT OF DELAY ON GPS AND IMU

Delay (ms)	$\tilde{x}$ (m)	$\tilde{y}$ (m)	$\tilde{\psi}$ (deg)	PSNR
0	0.029	0.042	0.13	16.37
10	0.045	0.06	0.28	16.3
100	0.34	0.33	1.5	15.73

Furthermore, the effects of variable time delay in the camera data along with a 10 ms and 100 ms delay in GPS and IMU data on the PD system are illustrated using Table VII, where the performance of the PD system is compared with that of the constant delay case.

TABLE VII. EFFECT OF VARIABLE DELAY IN CAMERA DATA

Delay in GPS and IMU data (ms)	Average delay in camera image(s)	Variability (s)	PSNR	FSIM
10	0.5	0	16.3	0.67
10	0.4583	0.0417	16.34	0.68
100	0.4583	0.0417	15.82	0.65

From Table VII, it can be observed that the image metrics for the case of variable time delay are actually higher than that of average delay case, this is due to the fact that the average value for the variable time delay is less as compared to the constant delay case. Thus, it can be concluded that the system is able to perform adequately even in the case of delay in GPS and IMU signals and variable time delay in camera and Lidar data.

F. Comparison with Video Prediction Methods

This subsection details the comparison of the estimation-based PD system with a state-of-art deep learning video prediction method DMVFN [17] to predict future images based on delayed images. DMVFN is an advanced video prediction framework that incorporates dynamic routing mechanisms alongside multi-scale motion estimation techniques. Furthermore, the quality of images generated by DMVFN is claimed to be better than other methods [15,16]. The deep learning network takes two consecutive images at time t and $t-1$ to predict images for time $t_2 = t_1 + \Delta t$. Table VIII shows the average PSNR, SSIM and TOLM metric for the proposed method (NKSR) and DMVFN for the nuScenes dataset for a 0.5 s delay. From the table it is observed that NKSR is able to perform better than DMVFN on all the three metrics. It is also important to note that although DMVFN is able to produce images, since it doesn't utilize any information for the current position of the ego vehicle, the TOLM metric is indeed very low as compared to NKSR. The NKSR is able to increase the vehicle detection and localization accuracy by 96 % compared to DMVFN.

Another important drawback of deep learning networks is that they fail to capture the complete scene when the ego vehicle is turning but the estimation-based PD system alleviates this problem by transforming the delayed images from both the front and side camera using the same estimation-based PD system. Fig 11 shows the synthesized image for both DMVFN and NKSR and it can been seen that DMVFN is unable to synthesize the white car in the right portion of the image when the ego vehicle is turning right, whereas NKSR is able to do so. The slight difference in the illumination of the white car in NKSR as compared to ground truth image is due to the fact that the side portion of the NKSR image was inpainted by synthesizing the new image from the right-side camera which had an illumination different from the front camera.

TABLE VIII. COMPARISION WITH VIDEO PREDICTION METHODS

Method	PSNR	SSIM	TOLM
DMVFN	20.6	0.56	0.26
NKSR	**21**	**0.58**	**0.51**

Fig. 11. Image comparison for DMVFN and NKSR

G. Robustness against GPS outage

This subsection details the robustness of the estimation-based PD system in cases when there is GPS outage resulting in the loss of GPS signal. In such cases, Lidar-based localization methods can be used to obtain the measurements for the EKF.

TABLE IX. EFFECT OF GPS OUTAGE

Case	$\tilde{x}$ (m)	$\tilde{y}$ (m)	$\tilde{\psi}$ (deg)	PSNR
GPS Available	0.017	0.014	0.620	12.01
GPS Outage (EKF-KISS-ICP)	0.270	0.120	0.640	12
GPS Outage (EKF-FAST-LIO)	0.200	0.160	1.160	11.98

Two such methods are evaluated. In KISS-ICP, ego-localization is done using Lidar odometry while in FAST-LIO it is done by fusing the IMU and Lidar data. Both methods were operated at 10 Hz on the ego vehicle and the obtained position and yaw angle of the ego vehicle was then transmitted to the teleoperation station where they were used

as measurement for the EKF in cases when there was GPS outage. Consider the case, when a sample trajectory for KITTI dataset encountered a GPS loss. Table IX shows the performance of the EKF-based KISS-ICP and FAST-LIO as compared to the case when GPS measurements are available. From table IX it can be seen that the accuracy of the EKF based on KISS-ICP and FAST-LIO SLAM is less as compared to the case when the GPS readings are available. Although there is decrease in accuracy yet there is a negligible decrease in the PSNR values of the generated images. Hence, the estimation-based PD system with GPS outage is at par with the case when GPS signals are available.

V. Conclusion

In this paper, an estimation-based PD system was designed based on 3D reconstruction of the environment around the ego-vehicle using novel sensor fusion of delayed camera and Lidar data. Two techniques, one based on Poisson reconstruction (PSR) and the other based on deep learning (NKSR), were evaluated for creating a mesh representation of the 3D environment. Raycasting was then performed to synthesize images based on the estimated non-delayed position and orientation of the ego vehicle obtained through ego vehicle state estimation. Although the synthesized images were better than the delayed image feed, they had many black patches due to unmeshed regions and uncolored triangles. Hence, deep learning-based image inpainting was applied to the generated images to fill the black patches and generate realistic images for AV teleoperation. Furthermore, a new metric for evaluating the performance of PD algorithms for object detection and localization (TOLM) was developed. The developed PD algorithm was then applied to real-world experimental data from KITTI and nuScenes datasets. The results indicated the superior performance of estimation-based PD algorithms with image inpainting compared to delayed image feed over various image comparison metrics like PSNR, SSIM, FSIM, and MS-SSIM. Furthermore, the PD algorithm was also evaluated on a new TOLM metric, and it showed an improvement of 50 % over the delayed image feed, thus proving the effectiveness of PD systems in detecting and locating other vehicles on the road. This work focused on a static environment around the ego-vehicle. Future work would incorporate various dynamic objects like moving vehicles or pedestrians in the scene.

References

[1] S. Neumeier, N. Gay, C. Dannheim and C. Facchi, "On the Way to Autonomous Vehicles Teleoperated Driving," AmE 2018 - Automotive meets Electronics; 9th GMM-Symposium, Dortmund, Germany, 2018, pp. 1-6.

[2] Startup Vay's Autonomy Workaround: Teledrivers to Operate Cars from Remote Location, URL: https://www.caranddriver.com/news/a37648114/vay-autonomous-teledriver-startup/

[3] Remote operation and the future of trucking, URL: https://einride.tech/insights/remote-operation-and-the-future-of-trucking

[4] Frank, L. H., Casali, J. G., & Wierwille, W. W. (1988). Effects of Visual Display and Motion System Delays on Operator Performance and Uneasiness in a Driving Simulator. Human Factors, 30(2), 201-217.

[5] G. Sharma and R. Rajamani, "Teleoperation Enhancement for Autonomous Vehicles Using Estimation Based Predictive Display," in IEEE Transactions on Intelligent Vehicles, vol. 9, no. 3, pp. 4456-4469, March 2024, doi: 10.1109/TIV.2024.3360410.

[6] Sato etal. Implementation and evaluation of latency visualization method for teleoperated vehicle. In 2021 IEEE Intelligent Vehicles Symposium (IV), pages 1–7, 2021.

[7] Frederic Chucholowski. Evaluation of display methods for teleoperation of road vehicles. Journal of Unmanned System Technology, 3:80–85, 02 2016. doi: 10.21535/just.v3i3.38.

[8] James Davis, Christopher Smyth, and Kaleb McDowell. The effects of time lag on driving performance and possible mitigation. IEEE Transactions on Robotics, 26(3):590–593, 2010.

[9] Yingshi Zheng, Mark J. Brudnak, Paramsothy Jayakumar, Jeffrey L. Stein, and Tulga Ersal. Evaluation of a predictor-based framework in high-speed teleoperated military ugvs. IEEE Transactions on Human-Machine Systems, 50(6):561–572, 2020.

[10] Frederic Chucholowski. Eine vorausschauende Anzeige zur Teleoperation von Straßenfahrzeugen. PhD thesis, 03 2016.

[11] Gaetano Graf, Hao Xu, Dmitrij Schitz, and Xiao Xu. Improving the prediction accuracy of predictive displays for teleoperated autonomous vehicles. In 2020 6th International Conference on Control, Automation and Robotics (ICCAR), pages 440–445, 2020.

[12] Henrikke Dybvik, Martin Løland, Achim Gerstenberg, Kristoffer Bjørnerud Sl˚attsveen, and Martin Steinert. A low-cost predictive display for teleoperation: Investigating effects on human performance and workload. International Journal of Human-Computer Studies, 145: 102536, 2021.

[13] MD Moniruzzaman, Alexander Rassau, Douglas Chai, and Syed Mohammed Shamsul Islam. High latency unmanned ground vehicle teleoperation enhancement by presentation of estimated future through video transformation. J Intell Robot Syst, 106, 2022.

[14] Ming, Ruibo, et al. "A Survey on Video Prediction: From Deterministic to Generative Approaches." *arXiv preprint arXiv:2401.14718* (2024).

[15] Wu, Yue, et al. "Future video synthesis with object motion prediction." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[16] Liu, Ziwei, et al. "Video frame synthesis using deep voxel flow." *Proceedings of the IEEE international conference on computer vision*. 2017.

[17] Hu, Xiaotao, et al. "A dynamic multi-scale voxel flow network for video prediction." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[18] Hu, Anthony, et al. "Gaia-1: A generative world model for autonomous driving." *arXiv preprint arXiv:2309.17080* (2023).

[19] H. Liu, T. Lu, Y. Xu, J. Liu, W. Li and L. Chen, "CamLiFlow: Bidirectional Camera-LiDAR Fusion for Joint Optical Flow and Scene Flow Estimation," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, 2022, pp. 5781-5791, doi: 10.1109/CVPR52688.2022.00570.

[20] W. Jeon, A. Zemouche and R. Rajamani, "Tracking of Vehicle Motion on Highways and Urban Roads Using a Nonlinear Observer," in *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 2, pp. 644-655, April 2019, doi: 10.1109/TMECH.2019.2892700.

[21] D. Simon, "Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches," John Wiley & Sons, Hobo- ken, 2006. doi:10.1002/0470045345

[22] Zhao, Y.-L.; Hong, Y.-T.; Huang, H.-P. Comprehensive Performance Evaluation between Visual SLAM and LiDAR SLAM for Mobile Robots: Theories and Experiments. *Appl. Sci.* **2024**, *14*, 3945. https://doi.org/10.3390/app14093945

[23] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley and C. Stachniss, "KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way," in IEEE Robotics and Automation Letters, vol. 8, no. 2, pp. 1029-1036, Feb. 2023, doi: 10.1109/LRA.2023.3236571.

[24] W. Xu and F. Zhang, "FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter,"

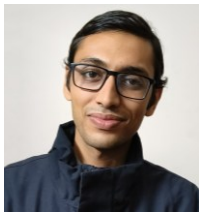
in *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3317-3324, April 2021, doi: 10.1109/LRA.2021.3064227.

[25] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson surface reconstruction. In Proceedings of the fourth Eurographics symposium on Geometry processing (SGP '06). Eurographics Association, Goslar, DEU, 61–70.

[26] Huang, Jiahui, et al. "Neural kernel surface reconstruction." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023.

[27] Li, Wenbo, et al. "Mat: Mask-aware transformer for large hole image inpainting." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022.

[28] Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLO (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics.



Gaurav Sharma obtained his B.Tech. degree in mechanical engineering from the Birla Institute of Technology and Sciences, Pilani in 2021. He is currently pursuing his Ph.D. degree at the University of Minnesota, Twin Cities, MN, USA. His research interests are in state estimation, and control, including applications in intelligent transportation and robotic systems.

Raunak Manekar obtained his Ph.D. degree from the University of Minnesota, Twin Cities, MN, USA. He is currently an Assistant Professor at BITS Pilani K K Birla Goa Campus, India. Prior to joining BITS Pilani, he worked as an AI Scientist at GE Healthcare. Dr. Manekar's research interests include machine learning, computer vision, and computational imaging, with a particular focus on applications in scientific imaging and robotic systems.

Rajesh Rajamani (*Fellow, IEEE*) obtained his M.S. and Ph.D. degrees from the University of California at Berkeley in 1991 and 1993 respectively and his B.Tech degree from the Indian Institute of Technology at Madras in 1989. Dr. Rajamani is currently the Benjamin Y.H. Liu-TSI Endowed Professor of Mechanical Engineering and Associate Director of the Minnesota Robotics Institute at the University of Minnesota. His active research interests include sensing, estimation and control for smart/autonomous systems.

Dr. Rajamani has co-authored over 190 journal papers and is a co-inventor on 16 granted patents. He is the author of the popular book "Vehicle Dynamics and Control" published by Springer. Dr. Rajamani has served as Chair of the IEEE CSS Technical Committee on Automotive Control and is currently Senior Editor of the *IEEE Transactions on Intelligent Transportation Systems*. He is a Fellow of IEEE and ASME and has been a recipient of the CAREER award from the National Science Foundation, the Ralph Teetor Award from SAE, the Charles Stark Draper Award from ASME, the O. Hugo Schuck Award from the American Automatic Control Council, and a number of best paper awards from conferences and journals. Several inventions from his laboratory have been commercialized through start-up ventures co-founded by industry executives. One of these companies, Innotronics, was recently recognized among the 35 Best University Start-Ups of 2016 by the US National Council of Entrepreneurial Tech Transfer.