This is a preprint. The DOI for the final published version is available at https://doi.org/10.23919/ICCAS63016.2024.10773150.

Metaverse aided Teleoperation Scene Prediction Optimization using Depth Camera Transformation based on Relative Velocities

Jonghyeok An[1], Changyeong Jeong[2], Hyunrok Cha[3], Myeonghwan Hwang[3], Seungha Yoon[3], and Eugene Kim*

[1]Robot Engineering, Korea National University of Science and Technology, Daejeon, 34113, South Korea (ajh5265@kitech.re.kr)
[2] Mechanical Engineering, Gwangju Institute of Science and Technology, Gwangju, 61005, South Korea
[3]Purpose-Based Mobility group, Seonam Division, Korea Institute of Industrial Technology, Gwangju 61012, South Korea

* Corresponding author with the Purpose-Based Mobility group, Seonam Division, Korea Institute of Industrial Technology, Gwangju 61012, South Korea(egkim@kitech.re.kr)

Abstract: The biggest problem with Vehicle Teleoperation is that its performance is greatly influenced by communication delay. In particular, considering its importance, the heterogeneity of camera information felt by remote drivers inherently involves human error that can lead to incorrect commands. Focusing on these issues, we aim to generate future Camera RGB frames considering communication delay and compare their performance. For this purpose, a teleoperation environment with communication delay was implemented on the metaverse, and future frames corresponding to the delay time were predicted. In particular, selective future frame prediction was implemented through an algorithm that clusters only surrounding vehicles from the surrounding environment and calculates the relative speed at each moment. From the results, it was confirmed that when predicting these future frames, it is better to predict by taking into account the relative speed of the surrounding environment, rather than simply predicting based on one's own speed.

Keywords: Teleoperation, Predictive display, Metaverse, Latency, Perspective projection

1. INTRODUCTION

Recent advancements in autonomous driving technology have been substantial; however, it is anticipated that the commercialization of this technology will require additional time due to the ongoing reports of accidents.[1]. For this reason, recent researchers have shown significant interest in remote driving technology as a means to supplement the still-imperfect autonomous driving systems [2]. The literature on teleoperation can be broadly classified into two categories: human-in-the-loop studies [3], [4], [5], [6] and driving performance evaluations [7], [8], [9]. Among them, a predictive display technique has been developed to anticipate visual information, thereby improving drivers' awareness during teleoperation [10], [11], [12]. In particular, a recent study utilizing perspective projection explored a novel method for reconstructing RGB data based on the vehicle's movement, taking communication delays into account [13].

1.1 Contribution

The perspective projection method predicted the next frame using the depth map image of the current frame. However, it did not consider the movement state of objects or vehicles in the external environment. If the velocity of the front vehicle is faster than the velocity of the ego vehicle, the front vehicle should appear smaller in the predicted image, but this was not the case with the perspective projection method. The proposed method predicts the next frame by considering the relative velocity of the ego vehicle and the front vehicle. We have named this method Relative Velocity-Informed Projection (RVIP).

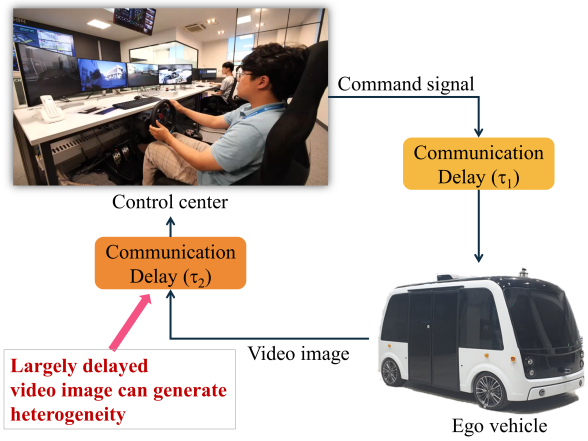


Fig. 1.: Teleoperated driving system workflow diagram

2. METHODOLOGY

The primary challenge in teleoperated driving is the communication delay that occurs between the control center and the ego vehicle. This causes a heterogeneity between the teleoperator's view and the actual environment. In Fig. 1, the process that occurs when the teleoperator performs teleoperated driving is illustrated. When a command is input by the teleoperator at the control center, the first communication delay (τ_1) occurs during transmission to the ego vehicle. Additionally, the video

*This work was supported by "Development of Core Technologies for a Working Partner Robot in the Manufacturing Field" (KITECH EO-24007)

This is a preprint. The DOI for the final published version is available at https://doi.org/10.23919/ICCAS63016.2024.10773150.

images generated by the camera installed on the ego vehicle experience a second communication delay (τ_2) when transmitted back to the control center, causing difficulties for the teleoperator in driving. The perspective projection alleviates the heterogeneity perceived by the teleoperator.

2.1 Perspective Projection

Ref. [13] shows the predicted screen presented to the teleoperator using perspective projection. After receiving the depth map image of the current frame, it is converted into a point cloud based on this information. Eq. (1) shifts the origin of the pixels from the top-left corner to the center of the depth map image.

$$\begin{aligned} u_d &= x_d - \frac{W+1}{2} \\ v_d &= y_d - \frac{H+1}{2} \end{aligned} \tag{1}$$

In Eq. (1), x_d, y_d, and z_d represent the horizontal, vertical, and depth values in a coordinate system with the origin at the top left. u_d, v_d, and w_d are the x_d, y_d, and z_d values based on the shifted origin. W and H are the width and height of the image resolution.

Eq. (2) describes the process of converting the depth map into a point cloud.

$$\begin{aligned} u_p &= w_p \left[u_d \left\{ \frac{\tan(\frac{\text{fov}H}{2})}{W/2} \right\} \right] \\ v_p &= w_p \left[v_d \left\{ \frac{\tan(\frac{\text{fov}V}{2})}{H/2} \right\} \right] \end{aligned} \tag{2}$$

fov$H$ and fov$V$ represented the horizontal and vertical fields of view (FOV) of the depth map image, respectively. Since the depth map image provides the z-coordinates, or depth values, these did not need to be calculated separately.

The obtained point cloud is translated to the ego vehicle's position in the next frame using the current velocity and steering angle values of the ego vehicle. The matrix T in Eq. (3) is a transformation matrix created considering the rotation and translation of the ego vehicle.

$$T = \begin{bmatrix} C_\Psi & 0 & -S_\Psi & C_\Psi \Delta X_{\text{cam}} - S_\Psi \Delta Z_{\text{cam}} \\ 0 & 1 & 0 & 0 \\ S_\Psi & 0 & C_\Psi & S_\Psi \Delta X_{\text{cam}} + C_\Psi \Delta Z_{\text{cam}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

T is used to predict the point cloud for the next frame. Here, the Ψ value denotes the yaw angle of the vehicle. C_Ψ and S_Ψ are $\cos(\Delta\Psi_{\text{cam}})$ and $\sin(\Delta\Psi_{\text{cam}})$, respectively, and consider the rotational movement of the vehicle. ΔX_{cam} and ΔZ_{cam} consider the translational motion of the vehicle.

Eq. (4) obtains the $u_{p\text{Next}}$, $v_{p\text{Next}}$, and $w_{p\text{Next}}$ values of the point cloud for the next frame by applying the transformation matrix T to the u_p, v_p, and w_p.

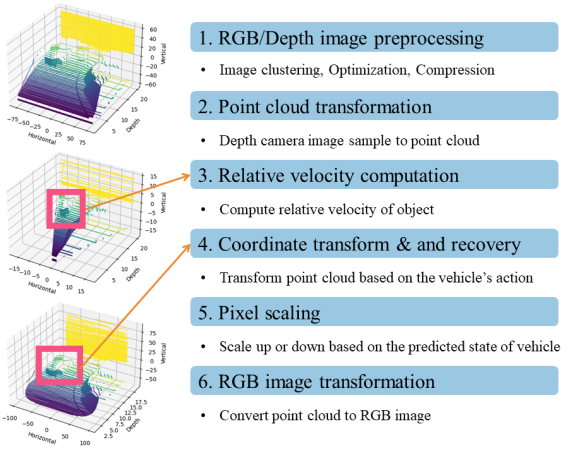


Fig. 2.: Process flow chart of proposed method

$$\begin{bmatrix} u_{p\text{Next}} \\ v_{p\text{Next}} \\ w_{p\text{Next}} \\ 1 \end{bmatrix} = T \cdot \begin{bmatrix} u_p \\ v_p \\ w_p \\ 1 \end{bmatrix} \tag{4}$$

The point cloud values at the new position need to be converted back to a depth map image, and the origin has to be shifted back from the center of the image to the top-left corner. This is inversely converted using the equations described in Eq. (1) and Eq. (2). Using Eq. (5), the depth value of each pixel in the current frame is divided by the depth value at the new (next frame) position to create a scaling factor.

$$S = \frac{w_p}{w_{p\text{Next}}} \tag{5}$$

This scaling factor is then used to adjust the size of each pixel, resulting in the generation of the predicted RGB image. The perspective projection process is performed only on the stationary environment point cloud. This is because the stationary environment is affected solely by the velocity and steering angle of the ego vehicle.

2.2 Relative Velocity-Informed Projection

The perspective projection method predicts and displays the position of the ego vehicle to the teleoperator assuming that only the ego vehicle is moving and the environment remains stationary. However, in real-world scenarios, parts of the environment are in motion. Notably, there are other vehicles driving on the road alongside the ego vehicle. Fig. 2 illustrates the process by which the predicted frames are delivered to the teleoperator. The conventional perspective projection method lacks the third step, which is the relative velocity computation. The newly proposed method, RVIP, considers the state of moving objects within the environment to provide the teleoperator with a predictive display.

In this process, a newly utilized element is the semantic image. Using both the semantic image and the depth map image, the vehicles and the environment are distinguished and converted into a clustered point cloud. The

relative velocity is then calculated using the clustered vehicle point cloud. Eq. (6) and Eq. (7) are used to calculate the relative velocity between the ego vehicle and the front vehicle, which only travel straight.

$$d_{\text{rel}} = w_{p\text{Prev,min}} - w_{p,\text{min}} \tag{6}$$

$$V_{\text{rel}} = \frac{d_{\text{rel}}}{\Delta t} \tag{7}$$

The term d_{rel} represents the difference in distance between the front vehicle and the ego vehicle across consecutive frames, while V_{rel} denotes the relative velocity. In Eq. (6), $w_{p\text{Prev}}$ is the smallest depth value among the front vehicle point clouds in the previous frame, and $w_{p,\text{min}}$ represents the smallest value in the current frame. In Eq. (7), Δt represents the time interval between the previous frame and the current frame. If the front vehicle is moving faster than the ego vehicle, the relative velocity is negative; if it is slower, the relative velocity is positive. V_{rel} is calculated by taking the point with the smallest depth value (w) from the point cloud of the front vehicle in both the previous and current frames, subtracting the value of the previous frame from the current frame, and then dividing by Δt. Because the relative velocity between frames is assumed to be constant, we use the relative velocity of the previous frame as the relative velocity of the current frame.

Eq. (8) is a transformation matrix that expresses the rotation change and movement change of the vehicle considering the relative velocity in Eq. (7). Equation (9) calculates the vehicle point cloud of the next frame using T_{rel} and the vehicle point cloud of the current frame.

$$T_{\text{rel}} = \begin{bmatrix} C_\Psi & 0 & -S_\Psi & C_\Psi \Delta X_{\text{cam}} - (S_\Psi \Delta Z_{\text{cam}} + V_{\text{rel}} \Delta t) \\ 0 & 1 & 0 & 0 \\ S_\Psi & 0 & C_\Psi & S_\Psi \Delta X_{\text{cam}} + (C_\Psi \Delta Z_{\text{cam}} + V_{\text{rel}} \Delta t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} u_{\text{veh},p\text{Next}} \\ v_{\text{veh},p\text{Next}} \\ w_{\text{veh},p\text{Next}} \\ 1 \end{bmatrix} = T_{\text{rel}} \cdot \begin{bmatrix} u_{\text{veh},p} \\ v_{\text{veh},p} \\ w_{\text{veh},p} \\ 1 \end{bmatrix} \tag{9}$$

$u_{\text{veh},p\text{Next}}$, $v_{\text{veh},p\text{Next}}$, $w_{\text{veh},p\text{Next}}$ are the vehicle point cloud values in the next frame, and $u_{\text{veh},p}$, $v_{\text{veh},p}$, $w_{\text{veh},p}$ are the vehicle point cloud values in the current frame. The inverse transformations, including converting the point cloud to depth and shifting the pixel origin to the top-left corner, are performed as in the perspective projection method. The relative velocity pixel size scale factor is determined using Eq. (10).

$$S_{\text{rel}} = \frac{w_{\text{veh},p}}{w_{\text{veh},p\text{Next}}} \tag{10}$$

If the front vehicle is moving faster than the ego vehicle, $w_{\text{veh},p\text{Next}}$ is greater than $w_{\text{veh},p}$, resulting in an S_{rel} value between 0 and 1, thus having a diminishing effect. Similar to the perspective projection method, the relative scale factor varies for each pixel, causing the range of each pixel to differ.

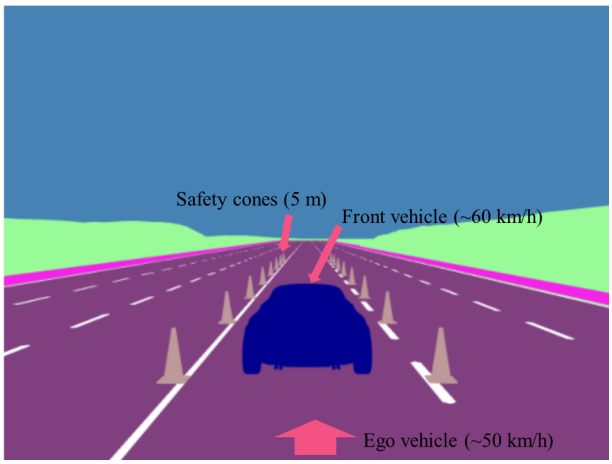


Fig. 3.: Semantic image provided by MetaDrive. It is used to distinguish and cluster the vehicle and other environmental elements.

3. EXPERIMENT

3.1 General setup

The computer's operating system was Linux Ubuntu 20.04 LTS, and Python version 3.8.19 was used. The simulator used was MetaDrive version 0.4.2.3. The semantic images, RGB images, and depth map images were provided by MetaDrive [1] [14].

3.2 Performance validation

3.2.1 Simulation conditions

The ego vehicle used in the experiment was defined as a vehicle teleoperated by a driver. It was provided by MetaDrive and has dimensions of length = 4.52 m, height = 1.19 m, and width = 1.85 m. The velocity limits were set to 60 km/h for the front vehicle and 50 km/h for the ego vehicle. This setup was intentionally designed to use relative velocity to predict the position of the front vehicle as it moves away from the ego vehicle.

It is important to note that the maximum depth value of the depth camera is 20 m. This is to ensure that the same equation used in the perspective projection method for depth data compression can be applied. If the maximum depth is not 20 m, the constant values in the equation would need to be adjusted.

A total of 16 safety cones were used on the road, with 8 cones on the left line and 8 cones on the right line. The distance between the cones on the left and right lines was 3.8 m, and the distance between cones in the same line was set to 5 m. To intuitively understand the effectiveness of this method, the ego vehicle and the front vehicle were set to drive in the same lane and could only move straight. In our experimental setup, fov$H$ and fov$V$ values are set to 80.

The simulation was conducted for a total of 100 steps. The interval between physical calculations for each step is 0.07 seconds, and the physical calculation is performed only once per frame.

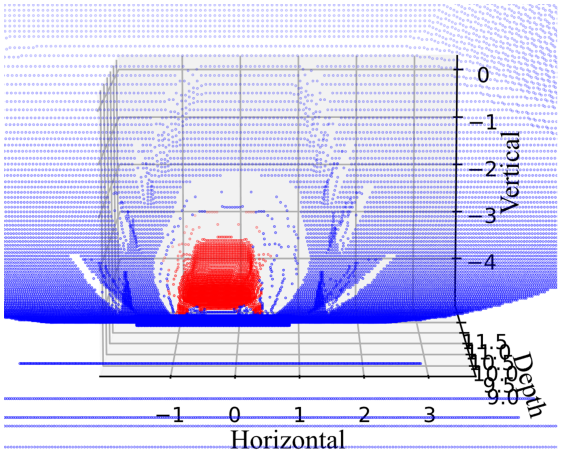
[1] https://metadriverse.github.io/metadrive/

This is a preprint. The DOI for the final published version is available at https://doi.org/10.23919/ICCAS63016.2024.10773150.

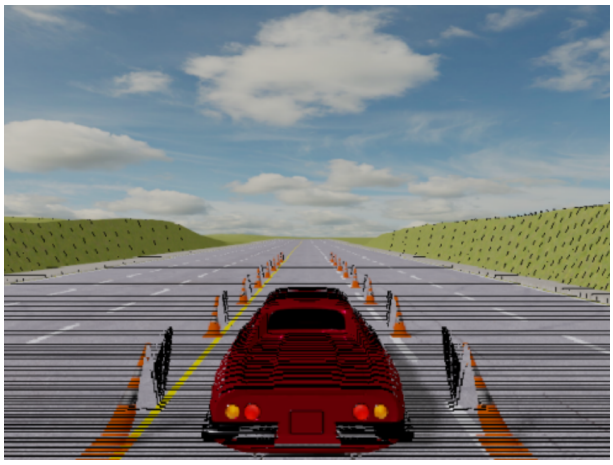
3.2.2 Evaluation of result

The predicted next frame vehicle point cloud and the actual next frame's points with the smallest depth values were extracted. Values from 30 seconds to 80 seconds were collected, and the error was calculated by subtracting the extracted values from the actual frame values for each method. The errors were tested using the Mann-Whitney u test rank sum method, and the p-value obtained from this process was used to determine the significance of the proposed method.

4. RESULT & DISCUSSION



(a) Point cloud (P.P)

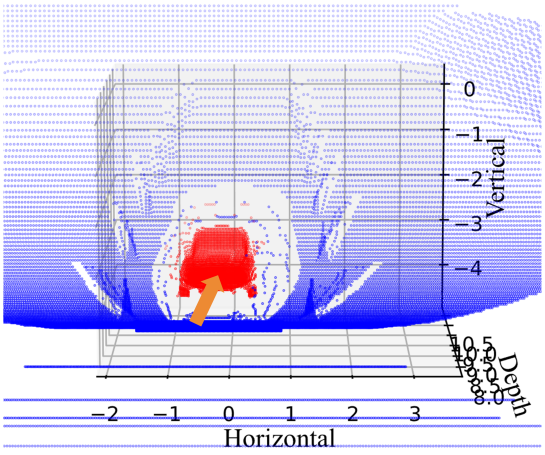


(b) RGB (P.P)

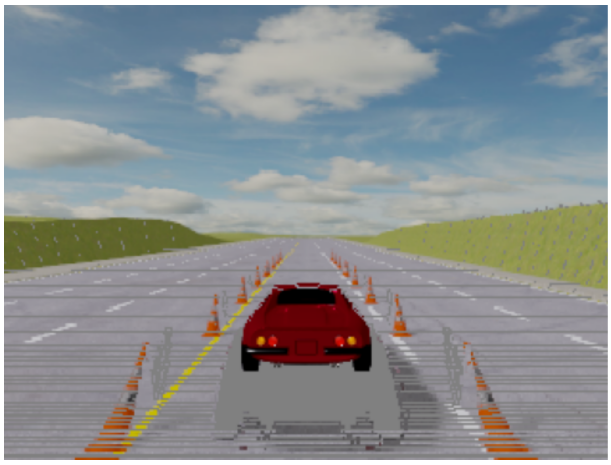
Fig. 4.: Comparison of point clouds and RGB images (perspective projection method). (a) is the point cloud of the predicted frame using the P.P. method. Red dots indicate points corresponding to the front vehicle. Blue dots indicate points that correspond to other points that do not belong to the vehicle. (b) is the RGB image of the predicted frame using the P.P. method.

4.1 Point cloud and RGB of the predicted frame

Fig. 4 and Fig. 5 show the perspective projection prediction and our prediction expressed as a point cloud and RGB image, respectively. In our method, the vehicle appears further ahead in the depth direction on the point cloud because the relative velocity of the vehicle in front is considered. Accordingly, the image of the vehicle in the predicted image is scaled down to appear farther away than expected. From the relative positions of the front vehicle and the cones on both sides, it can be intuitively confirmed that the faster vehicle in front is further ahead. The lines visible below the point cloud are pixels lost during the image encoding/decoding process. Empty pixels appear in the image predicted by the perspective projection method as perspective projection is applied, but they are filled with gray similar to the road color to improve image quality.



(a) Point cloud (Ours)



(b) RGB (Ours)

Fig. 5.: Comparison of point clouds and RGB images (Ours). (a) is the point cloud of the predicted frame using the ours method. Red dots indicate points corresponding to the front vehicle. Blue dots indicate points that correspond to other points that do not belong to the vehicle. (b) is the RGB image of the predicted frame using the ours method.

4.2 Predicted RGB comparison by frame

Fig. 6 shows images where successive frames and predictions were extracted from the MetaDrive simulation. Each row represents the actual images and predicted im-

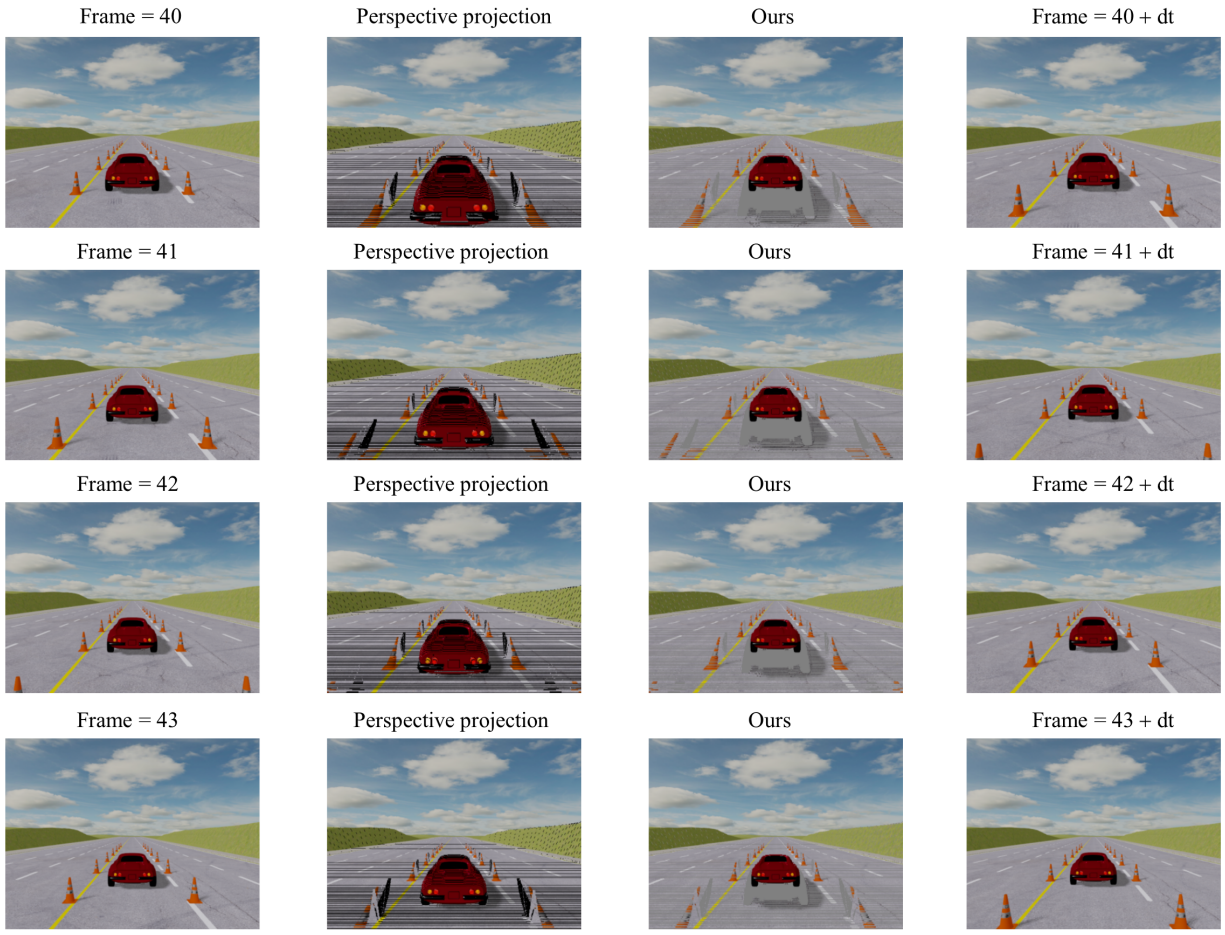


Fig. 6.: RGB images of current frame, perspective projection, proposed method, and actual frame after dt.

ages from Frame 40 to Frame 43. The first column shows the actual frame at the given time (e.g., Frame = 40). The second column displays the prediction using the Perspective Projection method. The third column presents the prediction using our proposed method. The fourth column depicts the actual frame at the next time step (e.g., Frame = 40 + dt).

As previously mentioned, because the limit velocity of the front vehicle is set higher than that of the ego vehicle, the gap between the vehicles should increase over time. When using the P.P. method for prediction, the gap with the front vehicle narrowed, whereas with the proposed method, the gap widened. When compared to the next frame, it can be seen that the proposed method is more similar.

4.3 Distance comparison between ego vehicle and front vehicle

In Fig. 7, the result of error comparison with and without the proposed method is shown. A significance test (Mann-Whitney U test-rank sum) was used (p:3.30 e-18) for two groups (W/WO). The result indicates that by using the proposed method that takes the relative velocity into account, the prediction of the video frames can be improved.

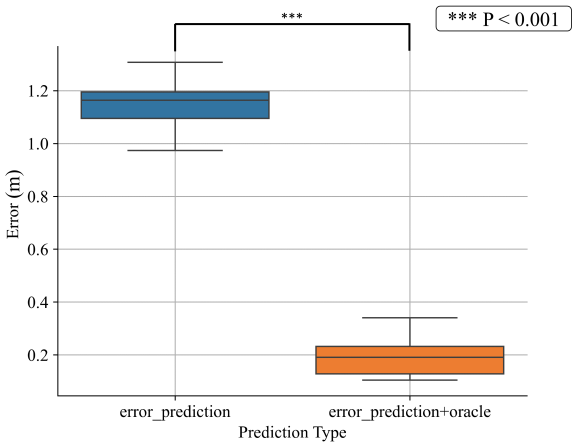


Fig. 7.: Comparison of distance error between the ego vehicle and the front vehicle with and without the proposed method. It can be known that the error when using the proposed method is significantly smaller than that of the prediction using perspective projection method ($p<0.001$)

5. CONCLUSION

In this paper, we propose a new method to make predictive displays for remote driving of vehicles more realistic. The proposed method improves accuracy by not only considering the ego vehicle's expected moving position but also considering the expected position of surrounding vehicles based on their relative velocities. To evaluate the proposed method, we used metaverse simulation, which artificially created communication delay, and compared the difference between using the proposed method and not using it while driving at different velocities on a straight road. The results confirmed that the proposed method better represents the actual vehicle distance and that there was a significant difference compared to when the proposed method was not used. The proposed method can be applied not only to remote driving of vehicles but also to other fields involving remote control.

In future work, the experimental environment was simulated only on a straight road; it will be necessary to evaluate the system on curved roads that consider the rotational motion of the ego vehicle and surrounding vehicles. Subsequently, the performance of the proposed algorithm should be evaluated on actual vehicles or in other applications.

REFERENCES

[1] A. Bhardwaj, A. H. Ghasemi, Y. Zheng, H. Febbo, P. Jayakumar, T. Ersal, J. L. Stein, and R. B. Gillespie, "Who's the boss? arbitrating control authority between a human driver and automation system," *Transportation research part F: traffic psychology and behaviour*, vol. 68, pp. 144–160, 2020.

[2] D. Majstorović, S. Hoffmann, F. Pfab, A. Schimpe, M.-M. Wolf, and F. Diermeyer, "Survey on teleoperation concepts for automated vehicles," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1290–1296, IEEE, 2022.

[3] T. B. Sheridan, "Space teleoperation through time delay: Review and prognosis," *IEEE Transactions on robotics and Automation*, vol. 9, no. 5, pp. 592–606, 1993.

[4] J. Y. Chen, E. C. Haas, and M. J. Barnes, "Human performance issues and user interface design for teleoperated robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1231–1245, 2007.

[5] J. Prakash, M. Vignati, E. Sabbioni, and F. Cheli, "Vehicle teleoperation: Human in the loop performance comparison of smith predictor with novel successive reference-pose tracking approach," *Sensors*, vol. 22, no. 23, p. 9119, 2022.

[6] J. Prakash, M. Vignati, E. Sabbioni, and F. Cheli, "Vehicle teleoperation: Successive reference-pose tracking to improve path tracking and to reduce time-delay induced instability," in *2022 IEEE Vehicle Power and Propulsion Conference (VPPC)*, pp. 1–8, IEEE, 2022.

[7] J. Storms and D. Tilbury, "Equating user performance among communication latency distributions and simulation fidelities for a teleoperated mobile robot," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4440–4445, IEEE, 2015.

[8] D. J. Gorsich, P. Jayakumar, M. P. Cole, C. M. Crean, A. Jain, and T. Ersal, "Evaluating mobility vs. latency in unmanned ground vehicles," *Journal of Terramechanics*, vol. 80, pp. 11–19, 2018.

[9] M. Cross, K. A. McIsaac, B. Dudley, and W. Choi, "Negotiating corners with teleoperated mobile robots with time delay," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 6, pp. 682–690, 2018.

[10] F. E. Chucholowski, "Evaluation of display methods for teleoperation of road vehicles," *Journal of Unmanned System Technology*, vol. 3, no. 3, pp. 80–85, 2016.

[11] M. J. Brudnak, "Predictive displays for high latency teleoperation," in *Proc. NDIA Ground Veh. Syst. Eng. Technol. Symp.*, pp. 1–16, 2016.

[12] M. Moniruzzaman, A. Rassau, D. Chai, and S. M. S. Islam, "High latency unmanned ground vehicle teleoperation enhancement by presentation of estimated future through video transformation," *Journal of Intelligent & Robotic Systems*, vol. 106, no. 2, p. 48, 2022.

[13] J. Prakash, M. Vignati, D. Vignarca, E. Sabbioni, and F. Cheli, "Predictive display with perspective projection of surroundings in vehicle teleoperation to account time-delays," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[14] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou, "Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, pp. 3461–3475, March 2023.