Diminished Reality System Based on Open-source Software for Self-driving Mobility

Sei Ikeda* Ritsumeikan University
Iwao Takemura† Ritsumeikan University
Asako Kimura‡ Ritsumeikan University
Fumihisa Shibata§ Ritsumeikan University

ABSTRACT

The diminished reality (DR) techniques that visualize blind areas in road environments are expected to prevent accidents and to reduce passengers' stress or anxiety. However, the feasibility of such techniques is still unclear because most researches on DR for road environments are based on the assumption of the availability of specific sensor arrangements and infrastructures, which are not guaranteed to spread in the future. In this research, we propose a novel design to implement a DR system for rendering ghosted hidden background areas using various sensor data for self-driving. Our major assumption is that a number of automotive vehicles run around the world in the near future and their sensors and program modules are available for other purposes. In our experiments, we confirmed that hidden area can be visualized by using such data and modules.

Index Terms: Human-centered computing—Human computer interaction—Interaction paradigms—Mixed/augmented reality; Modeling and simulation—Computer graphics—Graphics systems and interfaces—Mixed/augmented reality

1 INTRODUCTION

The diminished reality (DR) techniques [9] in car, which visualize road scenes occluded by buildings and other automobiles [13], potentially contribute to providing the driver and fellow passengers with a sense of security as well as the safety. In the case of SAE J3016 (2016) autonomy levels 3 or 4, drivers are required to partially operate or response to the system and responsible for accidents in Japan. Such visualization techniques allow to inform the existence of pedestrians and cars to passengers, and potentially prevent accidents in advance. Even in the case of level 5, which does not require the existence of a driver, DR techniques can inform causes of sudden stops to the passengers and potentially provide a sense of security [14].

Most of the previous studies on DR applications on roads have been performed experiments using dedicated devices and infrastructures. Kojima et al. proposed a system that visualizes the dead angle which is not visible through side and rear-view mirrors [8]. In this research, they assume a road surveillance camera installed in a utility pole. Rameau et al. proposed a different type of system that visually eliminates vehicles running in front of the ego-vehicle and visualizes the hidden scene [13]. For the visualization, they performed texture-mapping to a point cloud obtained from a stereo camera mounted on a leading vehicle and sharing the point cloud with the ego-vehicle via wireless network. There are several researches similar to this as mentioned later. In such researches, ghosting algorithms and hardware configurations are specialized in image synthesis of the front vehicle.

*e-mail: ikeda.sei.jp@ieee.org
†e-mail: takemura@rm.is.ritsumei.ac.jp
‡e-mail: asa@is.ritsumei.ac.jp
§e-mail: fshibata@is.ritsumei.ac.jp

The goal of this study is to confirm the applicability of a software platform for self-driving to DR-based visualization in road environments. The software platform we adopt is Autoware [6], which is designed not only to simulate self-driving algorithms but also to control real autonomous vehicles. Autoware is based on the robot operating system (ROS), which is the robotics middleware consisting of various program modules with high reusability. We designed a ROS-based DR module and its connections to other modules of Autoware to demonstrate that DR applications can be implemented without installing additional special sensors in environment or vehicles in the near future where many self driving vehicles move around the environment. To confirm the feasibility of our design, we applied our system to two sets of image and LiDAR (Laser Imaging Detection and Ranging) point cloud containing static and dynamic objects to be ghosted.

2 RELATED WORK

2.1 Mixed/Diminished Reality in Road Environments

The group of Kitahara, Kameda, and Ohta in Tsukuba University have explored various applications of mixed/diminished reality techniques to road/vehicle environments [15]. For example, Sasai et al. [14] proposed a system to visualize road surface with predicted wheel trajectories. They demonstrated one evidence of the fact that this kind of visualization allows to reduce passengers' anxiety.

Another typical application of diminished reality is visualization of far-front view occluded by the front vehicle [2, 5, 11]. Real-time rendering using camera images acquired from a real vehicle has achieved [13]. Our research does not aim to such a specific situation.

One similar research field is rendering point cloud data of LiDARs [7, 10]. We are in a position not to study such basic rendering techniques themselves, but to adopt appropriate ones. Our current experiments rely on an original point-based rendering method which is easy to implement.

2.2 Framework for Self-driving Car

In addition to Autoware [6], several software platforms for self-driving simulation have been released. Carla-simulator[1] is also an open-source simulator for self-driving research [3]. This platform provides Python interface and virtual environment model. A similar platform AirSim[2] has been released by Microsoft [16]. An online educational service Udacity's Self-Driving Car Nanodegree Program also release open source programs based on Autoware. Since it is possible to acquire additional dataset by ourselves, we selected Autoware as a platform compatible with Tier IV's real sensing data acquired in Japan.

Autoware adopts various existing basic technologies such as localization [1, 17], mapping [19], object detection [4], object recognition [20], pass planning, and vehicle controlling. This research is not in a position not to discuss improvements of individual technologies, but to challenge implementing a DR technique on the platform with high practicality and re-usability.

[1]https://github.com/carla-simulator/carla
[2]https://github.com/Microsoft/AirSim

3 Diminished Reality System based on Autoware

3.1 Autoware Modules Related to DR

Autoware is a combination of various basic modules implemented on Robot Operating System (ROS). ROS follows the publish/subscribe model, which allows to connect additional modules. The main modules in Autoware are summarized as follows:

Sensing modules acquire data from various sensors such as LiDAR, RGB camera, GPS, and inertial sensor.

SLAM modules compute a single integrated point cloud (3-D map) and pose of the LiDAR device from range data of the LiDAR.

Localization modules also estimate the pose of the LiDAR device by matching the current range data to the existing 3-D map.

Dynamic object detection modules detect vehicles and pedestrians from the current range data and RGB image.

Static object detection modules extract static objects such as signals on image from current ego-vehicle pose, the current RGB image and a vector map, which stores the positions of static objects such as signals, buildings and lanes. The modules return fine positions of those and signal status.

Route finding modules find the detailed routes and appropriate velocity from a vector map and the current pose of the ego-vehicle.

Vehicle control modules control the ego-vehicle by using the pose of the ego-vehicle, the suitable velocity, and the decided route.

We reuse the underlined modules listed above for the DR system. Besides these, the static and dynamic recognition modules rely on the projection module that projects the range data or point cloud of the LiDAR onto an image and can be used for image generation in DR.

3.2 Design of DR Module

The DR module is designed to receive data from three information sources as input. First, the 3-D target area and the RGB image with its camera parameters are acquired directly from the ego-vehicle itself. Second, the hidden background image with its camera parameters are received through the V2V or V2I network. Finally, a required part of the 3-D map is received from a server through the V2I network. Given these pieces of input information, the DR module outputs a video see-through image where the area of the target object should be semi-transparent. The video see-through image superimposed on the image is output. We assume it is known which part of the global point cloud and which image frame of all the images in the database correspond to the current view.

The hidden background rendering implemented in the DR module is based on a traditional point-based rendering method. The point cloud in the 3-D map is projected onto the hidden background image to obtain texture as patches. The same point cloud is also projected onto the camera image of the ego-vehicle to overlay the patches. The LiDAR's pose is obtained from the SLAM module and the camera pose is calculated by multiplying the LiDAR-camera calibration matrix with the LiDAR pose matrix.

In order to support depth order perception, the projected patches are blurred by a Gaussian kernel with a standard deviation 10 pixels, and the edges of the target object are overlaid.

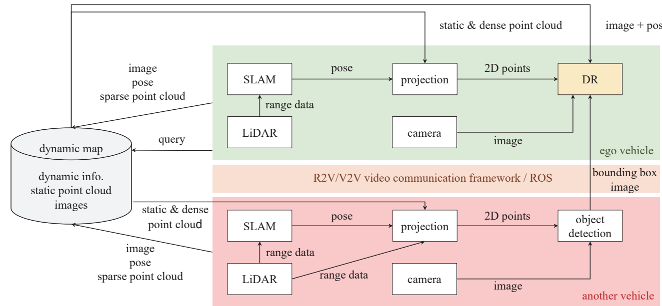


Figure 1: Dataflow of the future system.

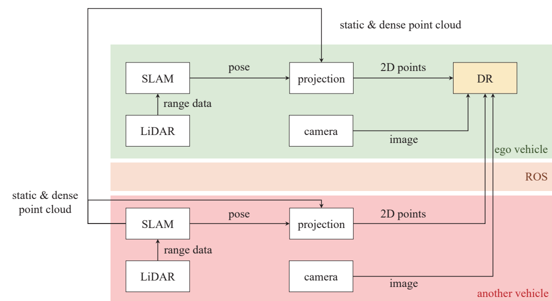


Figure 2: Dataflow of the current experiment.

3.3 Design of Connections & Simplification

Figure 1 shows our proposed framework using Autoware software modules. We assumed that there is a database server which provides and send a subset of the whole dynamic map and image frames according to a request form the ego-vehicle. The dynamic map consist of multiple layers, each of which stores either 3-D point cloud, images, vector map, or dynamic object information. The DR module mainly relays on SLAM[3] modules and projection module to render a hidden background image form the point cloud data and images sent from the database and another vehicle as an observer of the background.

We simplified this framework for our early experiment without the database server or the observer vehicle. For this purpose, the current experiment follows the framework shown in Figure 2. Additionally, we replaced the data acquired by the observer vehicle to the data acquired at the different frame in the same sequence. We use one frame where the target object occludes the background, and another frame where the background can be seen from the same vehicle.

[3] Simultaneously localizing and mapping [18]

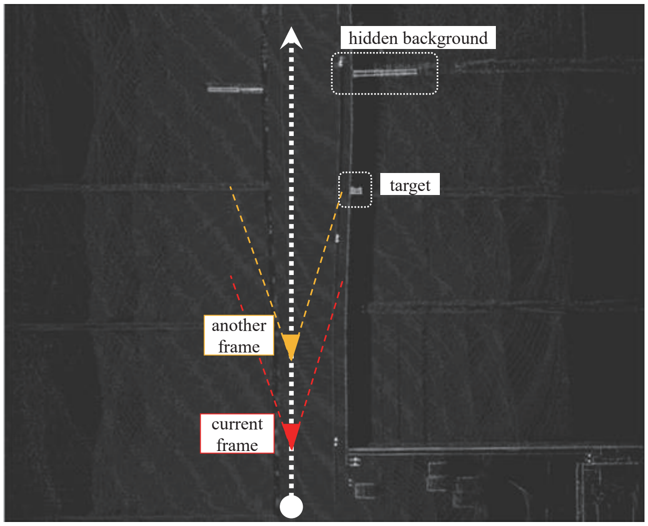
4 Experiment

4.1 Input Data

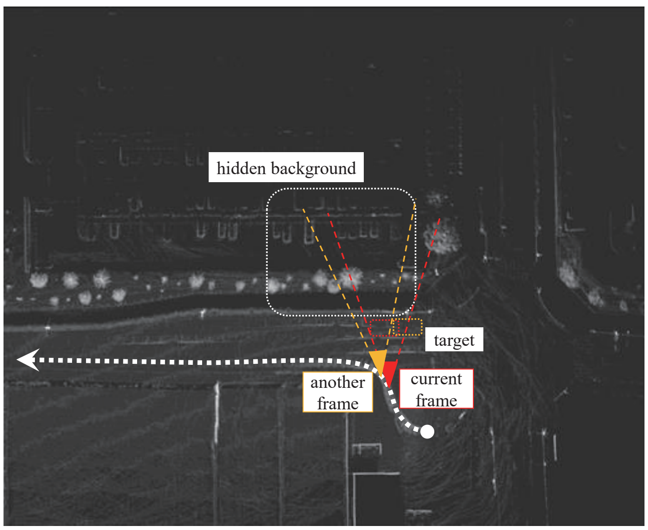
In order to confirm that the ghosted image can be generated only from the data that a vehicle acquired for self-driving, we used an existing dataset provided by Tier IV Inc. for testing self-driving algorithms. Tier IV provides a number of datasets containing sequences of sensor data such as images, point clouds of LiDAR, and GPS positions acquired by the automotive vehicle in road environments in the BAG format of the ROS. As of July 20, 2018, there are 28 BAG files, all of which contain camera images for object recognition. The data sequence in the BAG file contains time stamp and can be simulated using the actual data acquisition time.

We visually checked the video sequences contained in these files and selected the BAG file according to the following criteria: (i) The object is hiding the background from the camera position of a certain frame, and the background can be seen from the camera position of another frame. (ii) The background is static and has a characteristic texture. Following these criteria, we chose two parts in the same BAG file named anjo_01_2017-02-23.bag.

In the first part, the 963-th frame contains an advertisement board (target) hiding a background board (hidden background), as shown in Figure 3 (a). In the 989-th frame, the background can be seen from



(a) Static object.



(b) Dynamic object.

Figure 3: Experimental setup.

the camera. Therefore, we use the 963-th frame as the ego-vehicle viewpoint and the 963-th frame as the background observer one for the ghosted view of a static object. In the second part, the 407-th frame contains a vehicle (target) hiding guardrail and parked vehicles (background) as shown in Figure 3 (b). In this case, the target vehicle moved between the 417-frame and 419-frame. Although the camera displacement is quite small, the background becomes visible from the 419-frame viewpoint.

4.2 LiDAR-camera Calibration

The calibration parameters representing a rigid transformation from the LiDAR coordinate system to the camera one was computed by solving the PnP problem using multiple frames instead of using the original parameters. This is because the accuracy of the original parameters was not enough for overlaying patches. Additionally, the calibration process using the original calibration tool of the Autoware requires much time and a number of manual steps to obtain accurate parameters.

We manually pointed more than twenty pairs of 3-D LiDAR point and corresponding 2-D image point. The same process was performed on multiple frames. The PnP problem using these pairs were solved using a function solvePnP of the OpenCV. Such manual processing can be replaced by a self-calibration method [12].

4.3 Results & Discussions

Figure 4 and Figure 5 show the process of generation of ghosted images of static and dynamic objects, respectively. In Figure 4 (a), the rectangle shows 2-D region of interest (ROI) manually specified. The white advertising board containing a red curved arrow is the target object to be semi-transparent. (b) shows the extracted edges in the same region. The red dots in (C) indicate the projected point cloud. In (d), the background patches were projected. The same region was blurred and the textured edge pixels were overlaid in (e).

What can be judged from these figures, especially (c), is lack of the accuracy of the camera pose. A part of the point cloud corresponding to the electric pole was overlaid at a slightly shifted position. As shown in (d), the result rendered by our simple patch-based method contains the artifacts due to patch boundaries. Such artifacts became inconspicuous by blurring and foreground overlaying as seen in (e). For the purpose of grasping the rough state of the background scene, this level of image quality seems sufficient. In Figure 5, the similar things can be observed.

However, in each experiment, two frames were picked up from a single image sequence, and we dealt with these frames as the camera images of the ego vehicle and another vehicle. This set-up works advantageously in such image generation. Specifically, since temporal difference of the two frames was small, the changes in the illumination condition and camera position were small. Therefore, it is required to test the same conditioned experiments using two sets of sensing data from completely different vehicles

5 Conclusions & Future Work

We proposed a novel implementation design of a DR module which allows to present a ghosted view of a static background scene occluded by a static/dynamic object. The DR module was designed based on the Autoware, open-source software for self-driving mobility. Our experiments were conducted by using only an existing dataset provided for simulation of self-driving using the Autoware. This demonstrates the feasibility of the diminished reality system. In other words, in the environments where self-driving vehicles are popular enough, the DR can be realized only by adding a simple DR module without adding any new devices.

We will extend the current design for the case where there are dynamic objects in the background scene.

Acknowledgments

This work is partly supported by JSPS Kakenhi Grant Numbers 17H01747.

References

[1] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. IEEE. doi: 10.1109/IROS.2003.1249285

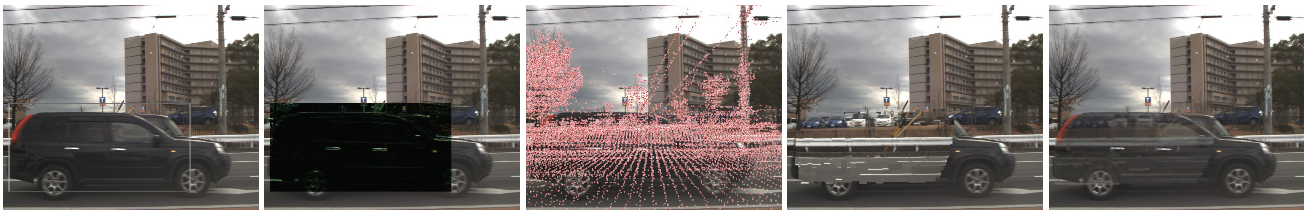
[2] H.-I. Chen, Y.-L. Chen, W.-T. Lee, F. Wang, and B.-Y. Chen. Integrating dashcam views through inter-video mapping. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015. doi: 10.1109/ICCV.2015.356

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, Sept 2010. doi: 10.1109/TPAMI.2009.167

(a) Input & ROI. (b) Extracted edges. (c) Projected point cloud. (d) Overlaid patches. (e) Result ghosted image.

Figure 4: Ghosted image of a static object.



(a) Input & ROI. (b) Extracted edges. (c) Projected point cloud. (d) Overlaid patches. (e) Result ghosted image.

Figure 5: Ghosted image of a dynamic object.

[5] P. Gomes, F. Vieira, and M. Ferreira. The see-through system: From implementation to test-drive. In *2012 IEEE Vehicular Networking Conference (VNC)*, pp. 40–47, Nov 2012. doi: 10.1109/VNC.2012.6407443

[6] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. An open approach to autonomous vehicles. *IEEE Micro*, 35(6):60–68, Nov 2015. doi: 10.1109/MM.2015.133

[7] A. Kelly, N. Chan, H. Herman, D. Huber, R. Meyers, P. Rander, R. Warner, J. Ziglar, and E. Capstick. Real-time photorealistic virtualized reality interface for remote mobile robot control. *The International Journal of Robotics Research*, 30(3):384–404, oct 2010. doi: 10.1177/0278364910383724

[8] K. Kojima, A. Sato, F. Taya, Y. Kameda, and Y. Ohta. Naviview: visual assistance by virtual mirrors at blind intersection. In *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.*, pp. 592–597, Sept 2005. doi: 10.1109/ITSC.2005.1520120

[9] S. Mori, S. Ikeda, and H. Saito. A survey of diminished reality: Techniques for visually concealing, eliminating, and seeing through real objects. *IPSJ Transactions on Computer Vision and Applications*, 9(1):17, Jun 2017. doi: 10.1186/s41074-017-0028-1

[10] S. Nebiker, S. Bleisch, and M. Christen. Rich point clouds in virtual globes – a new paradigm in city modeling? *Computers, Environment and Urban Systems*, 34(6):508–517, nov 2010. doi: 10.1016/j.compenvurbsys.2010.05.002

[11] C. Olaverri-Monreal, P. Gomes, R. Fernandes, F. Vieira, and M. Ferreira. The see-through system: A vanet-enabled assistant for overtaking maneuvers. In *IEEE Intelligent Vehicles Symposium*, pp. 123–128, June 2010. doi: 10.1109/IVS.2010.5548020

[12] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice. Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pp. 2053–2059. AAAI Press, 2012.

[13] F. Rameau, H. Ha, K. Joo, J. Choi, K. Park, and I. S. Kweon. A real-time augmented reality system to see-through cars. *IEEE Transactions on Visualization and Computer Graphics*, 22(11):2395–2404, Nov 2016. doi: 10.1109/TVCG.2016.2593768

[14] S. Sasai, I. Kitahara, Y. Kameda, Y. Ohta, M. Kanbara, Y. Morales, N. Ukita, N. Hagita, T. Ikeda, and K. Shinozawa. Mr visualization of wheel trajectories of driving vehicle by seeing-through dashboard. In *IEEE International Symposium on Mixed and Augmented Reality Workshops*, pp. 40–46, Sept 2015. doi: 10.1109/ISMARW.2015.17

[15] A. Sato, I. Kitahara, Y. Kameda, and Y. Ohta. Visual navigation system on windshield head-up display. In *Proc. 13th World Congress on Intelligent Transort Systems (ITS World Congress)*, 2006.

[16] S. Shah, D. Dey, C. Lovett, and A. Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[17] A. Sujiwo, E. Takeuchi, L. Y. Morales, N. Akai, Y. Ninomiya, and M. Edahiro. Localization based on multiple visual-metric maps. In *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 212–219, Nov 2017. doi: 10.1109/MFI.2017.8170431

[18] T. Taketomi, H. Uchiyama, and S. Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, Jun 2017. doi: 10.1186/s41074-017-0027-2

[19] E. Takeuchi and T. Tsubouchi. A 3-d scan matching using improved 3-d normal distributions transform for mobile robotic mapping. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3068–3073, Oct 2006. doi: 10.1109/IROS.2006.282246

[20] M. Yoshioka, N. Suganuma, K. Yoneda, and M. Aldibaja. Real-time object classification for autonomous vehicle using lidar. In *2017 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, pp. 210–211, Nov 2017. doi: 10.1109/ICIIBMS.2017.8279696