“Seeing” the Invisible: Under Vehicle Reconstruction (UVR) for Surround View Visualization

Feng HU,PhD · Shuping YU · Wenjia ZHOU

Autonomous Vehicle Software, NVIDIA

E-mail: {fengh, shupingy, wenjiaz}@nvidia.com

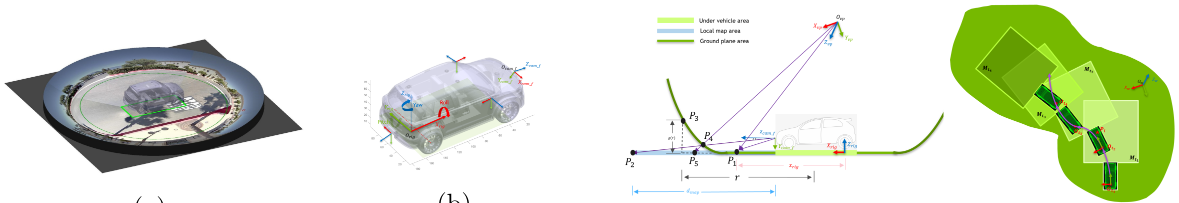
Abstract. Providing blind-spot-free vehicle surround view to the driver is important for many driving maneuvers such as parking. Existing vehicle Surround View System (SVS) can only visualize front, left, rear and right side of the vehicle but leaves the under vehicle area unknown. However, perceiving the under vehicle area is critical for many tasks such as passing through speed bumps, avoiding potholes, driving on narrow roads with high curbs or the unpaved terrain. In this paper, we propose a novel Under Vehicle Reconstruction (UVR) algorithm which utilizes what the vehicle sees in the past and vehicle egomotion to “see” through the original invisible under vehicle area. First, front or back fisheye cameras, are utilized to build a local textured map for future usage. Second, vehicle’s precise location and orientation within the local map is estimated using the vehicle egomotion. Finally, correspondent under vehicle area texture is retrieved from the map using vehicle’s pose and stitched together with traditional Surround View System to provide a new blind-spot-free visualization. As far as we know, our work is the first solution that can provide full under vehicle area reconstruction which empowers many Advanced Driving Assistant System (ADAS) functionalities such as transparent hood or transparent vehicle. Experiments on both simulated and real data are presented to show the effectiveness and robustness of the proposed algorithm.

1. Introduction

Surround View System (SVS) enables a driver to observe the surrounding area of the ego vehicle without the need of getting off the seat, especially the blind-spots area where the line of sight is occluded by the vehicle itself. This visualization is beneficial for many driving maneuvers such as smoothly parking into or out of a spot without hitting road curb, pedestrians, or other vehicles.

Existing Surround View System usually utilizes four fisheye cameras mounted at the front, left, rear and right sides of the vehicle body to perceive the surrounding from the four directions, though more cameras can be added in special cases such as for long trucks or vehicles with trailers. Fisheye video frames are then stitched together using camera parameters and blending techniques to provide a horizontal 360 degree seamless surround visualization.

Under vehicle area therefore is usually ignored by traditional SVS and is a large (e.g. multiple square meters) blind-spot for the existing system. However, perceiving the under vehicle area is critical for many tasks such as passing through high speed bumps, avoiding potholes on the road, driving on narrow surface with high curbs or the unpaved terrain. Capacity of visualizing the under vehicle area can also enable Advanced Driving Assistive System (ADAS) features such as “Transparent Hood”– seeing the relative positions of your two front wheels and the ground as



(a) (b) (c) (d)

Figure 1: (a) An Under Vehicle Reconstruction example; (b) the coordinate system diagram; (c) intersection view of the geometrical model; (d) map creating and updating example

if the vehicle hood is transparent, or "Transparent Vehicle"–seeing through your entire vehicle, an example of which is shown in Fig. 1a.

If the vehicle is backing up, the rear fisheye camera can be used to cache the surround information for future UVR usage, rather than the front fisheye camera. In some extreme cases, such as long trucks making sharp turns, side fisheye cameras can also be involved. In this paper, we will mainly focus on UVR with single front fisheye camera for vehicle moving forward case, since the principles apply to the backing up and other extreme cases easily.

We summarize our contributions as below: (1) a first-of-its-kind full under vehicle reconstruction solution that goes beyond traditional Surround View System for blind-spot-free vehicle visualization; (2) a local mapping procedure to retrieve vehicle surround visual information for future reconstruction; (3) an egomotion based under vehicle area texture retrieval approach utilizing vehicle geometric model and camera calibration parameters.

2. Related work

Advanced Driver Assistance System (ADAS) is becoming more and more popular with various features [1][2]. Displaying surrounding information of a vehicle for assistive parking and low-speed maneuvering is one of the ADAS functions and of interest to both academia and industrial communities [3][4][5]. A classical Surround View System consists of three components: (1) four fisheye cameras mounted surrounding the vehicle for image capturing; (2) a central computing unit where raw video frames from all cameras are processed; and (3) a visualization device, such as a built-in display, where the rendered results from specified perspectives are shown to the driver.

Typical rendering views include a "bird view" where a virtual camera is placed on top of the vehicle and all four sides of the vehicle are shown, or a "helicopter view" where the virtual camera is placed arbitrarily to show any specific viewing angle. For existing systems, since there is no visual input for the under vehicle area, artificial pixels are usually filled. Using pure black pixels to hint that the information is missing or more aesthetically using a computer graphic generated vehicle 3D model are two common ways.

Different camera types or positions will affect the final visualization result, therefore calibrating all camera intrinsic parameters and their relative poses are necessary for the Surround View Systems. Camera calibration is a well studied problem and many previous work are conducted [6][7][8][9]. Camera intrinsic parameters, such as focal length and distortion coefficients, can be obtained using Zhang's method [6] with a simple chessboard. Fisheye camera poses can be further obtained using overlapped pattern images [10]. Recent work also explores additional online calibration besides the above calibration procedure (i.e. factory or static calibration) to further enhance visualization accuracy.

Robust and accurate estimation of the egomotion of the vehicle relative to the road is a critical component and available to most vehicles, especially those with various levels of autonomous functionalities. A widely used method for determining egomotion is combining vehicle speed sensors, gyroscopes, and accelerators [11], though egomotion can also be enhanced with other sensors besides Inertial Measurement Unit (IMU), such as cameras [12] or Radar [13]. Depending on whether a global reference system is used or not, egomotion can be separated into two groups: absolute position measurements and relative position measurements (i.e. dead-reckoning). In our work, we use absolute egomotion for simulated experiments, and relative egomotion for real data experiments to illustrate the robustness of our UVR solution under various egomotion availability configurations.

3. Methodology

In our configuration, four fisheye cameras are installed at the front, left, rear and right side of a vehicle, where surrounding videos are continuously captured and transferred to the central process unit. Egomotion information are provided and synchronized with the video frame timestamps.

The major idea of Under Vehicle Reconstruction (UVR) is to preprocess the cached history frames associated with egomotion information and build a local map about vehicle's surrounding environment for future reconstructing. After the local map is built, at each future timestamp, the under vehicle area is retrieved using the map and the vehicle location within the map.

3.1. Geometric modeling

Fig. 1b is an illustration of our definitions for multiple involved coordinate systems. We first define a vehicle rig coordinate system and set its origin at the ground projection of the center of the rear vehicle axis. Its X_{rig} axis points to vehicle front, Y_{rig} axis points to the left, and Z_{rig} axis upward. In addition, we assume the vehicle egomotion coordinate system aligns with the vehicle model rig coordinate system. Moreover, all four fisheye cameras coordinate systems are calibrated relative to the vehicle rig coordinate system.

We model vehicle surrounding geometry as a symmetric 3D bowl shape, where the inner part is a circular ground plane and the outside is a bent slope whose height increases proportionally to its distance to the bowl center, as shown in Fig. 1a. Note that vehicles usually drive on flat surface and the places close to the vehicle are more important than further away parts from visualization perspective of view. With this bowl shape assumption, we not only can visualize the vehicle's close surrounding in a distortion-free way, but also can encapsulate further away scenes as much as possible at the cost of some distortion.

A sample view of the bowl shape is shown in Fig. 1c. The inner part is a circle with radius R_{inner}, whose height value (i.e. Z_{rig} value in the vehicle rig coordinate system) is zero. The outer side is a monotonic increasing curve defined by function $g(r)$ where r is the distance from the bowl center to the ground projection point of a 3D surface point. The height value of a point $P(x_p, y_p)$ in the rig coordinate system can be calculated with Equ. 1.

$$g(r) = c_1 * r^{c_2} \tag{1}$$

where $r = \sqrt{(x_p - x_o)^2 + (y_p - y_o)^2)}$, and x_o as well as y_o are the bowl center's coordinate system in the vehicle rig coordinate system. c_1 is an experimental constant adjusting bowl's height scale and c_2 is a parameter to control the curve shape (e.g. $c_2 = 2$ for a quadratic curve).

The bowl shape is colored into light green, dark green and blue for different areas as shown in Fig. 1c. The light green area is the targeted under vehicle area we want to reconstruct. The dark green is the area where we assume that it is flat in the bowl shape, and the blue area extends the dark green area used for local map building.

3.2. Map creating

A map is defined as a textured 2D planar surface consisted of grids for the surrounding of a vehicle. The size of the grid is defined as the map resolution and is configurable. In this paper we assume each grid is a square with the same pre-defined length e.g. 1cm $*$ 1cm.

If a local map already exists before a vehicle moves into the mapped area, and we also know the vehicle's position as well as orientation within the map, we can retrieve the under vehicle area information using the pre-built map.

For each location inside a mapping area at any timestamp, e.g. the blue area in Fig. 1c, we can calculate its visual information value by retrieving from the correspondent fisheye image pixel. Since the unit of the map grid is in metrics, e.g. in centimeter, but the unit of the fisheye image is in pixel, we need to build the relationship between these two units.

Note that each pixel in the fisheye image defines a ray, and if we back project the ray onto the ground it will hit a map grid. Therefore we can determine the grid's value using the correspondent ray (pixel) value. By repeating this process to each and every grid, we can create a local map using pixel information from a fisheye image.

3.3. Map updating and vehicle area retrieving

Denote at time t_i , $i = 0, 1, ..., T$, the new local map created from fisheye image as M_{t_i} and the updated map which merges all previous local maps as M'_{t_i}. At time t_{i+1}, map updating is the process of creating $M'_{t_{i+1}}$ by combining the new local map $M_{t_{i+1}}$ on top of M'_{t_i} using Equ. 2.

$$M'_{t_{i+1}} = M'_{t_i} \cup M_{t_{i+1}} \tag{2}$$

where $\cup$ is the map union operator that merges two aligned maps together e.g. using blending.

An illustration of the map updating is shown in Fig. 1d. Assume the vehicle is moving along the purple line and both vehicle position and orientation change along the time. O_{t_i} is vehicle rig coordinate system origin at the time t_i. M_{t_i} is the green rectangular area in front of the vehicle at time t_i. When the vehicle moves forward, the map updating procedure continuously digests the newly perceived local maps into a wider map, and when the vehicle moves on top of the wider map, its under vehicle area visual information is already prepared for retrieving.

Vehicle egomotion provides the relative pose relationship between vehicle rig origin O_{t_i} and $O_{t_{i+1}}$ at time t_i and t_{i+1}, which can be used to geometrically align two maps $M_{t_{i+1}}$ and M'_{t_i}. Denote the translation vectors of the vehicle rig coordinate system at time t_i and t_{i+1} as T_{t_i} and $T_{t_{i+1}}$, also denote the respective rotation matrices as R_{t_i} and $R_{t_{i+1}}$. For each point in the world coordinate system P_w, we can represent its coordinates P_{t_i} and $P_{t_{i+1}}$ in vehicle rig coordinate system at time t_i and t_{i+1} using Equ. 3

$$\begin{cases} P_w = R_{t_i} P_{t_i} + T_{t_i} \\ P_w = R_{t_{i+1}} P_{t_{i+1}} + T_{t_{i+1}} \end{cases} \tag{3}$$

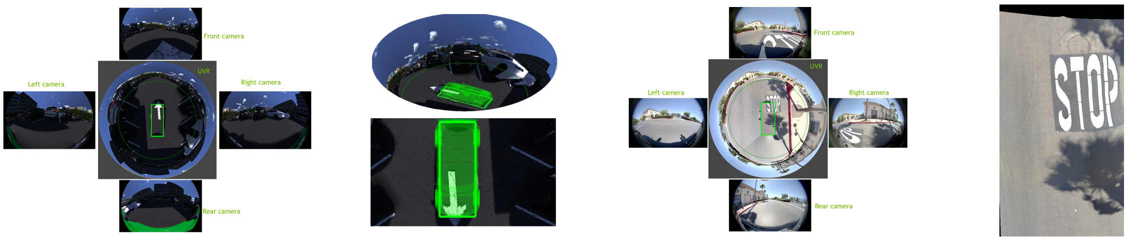
Therefore, by eliminating the P_w the value of $P_{t_{i+1}}$ at time t_{i+1} can be retrieved using Equ.4

$$P_{t_{i+1}} = R^{-1}_{t_{i+1}} (R_{t_i} P_{t_i} + T_{t_i} - T_{t_{i+1}}) \tag{4}$$

where T_{t_i}, $T_{t_{i+1}}$, R_{t_i} and $R_{t_{i+1}}$ are known via vehicle egomotion.

Equ.4 establishes the geometric transformation relationship between $M_{t_{i+1}}$ and M'_{t_i}, which allows us to find for any point in $M_{t_{i+1}}$ its correspondence in M'_{t_i} and vice versa. Once the map is updated, we can fill the under vehicle area in the 3D bowl shape using the updated map and completes under vehicle reconstruction. When the entire 3D bowl surface is fully textured, "helicopter view" or "bird view" can be rendered with specified viewport.

Note that at the very beginning of a trip, there is no history map cached for UVR usage, therefore UVR cannot be completed due to missing information. However, once the vehicle



(a) (b) (c) (d)

Figure 2: (a) Sample UVR result using simulated fisheye images; (b) transparent vehicle and transparent hood applications; (c) sample UVR result using real vehicle fisheye images; (d) local map for under vehicle area information retrieval

starts to move and the under vehicle ground was perceived by previous camera frames, we will be able to complete the UVR.

4. Experiments

In our experiments, we define the inner radius of the 3D bowl shape as 6 meters and outer radius 9 meters. Moreover, we set the map grid size as 1 cm $*$ 1cm.

4.1. Simulated data experiments

One sample result of Under Vehicle Reconstruction along with the four input fisheye images using simulated data is shown in Fig. 2a. The center image is Under Vehicle Reconstruction result combined with traditional Surround View System, within which the green rectangular in the middle is the under vehicle area, and the green circle is the inner circle of the 3D bowl shape. A demo video showing results of UVR for vehicle's passing through the ground landmarks can be found with link: https://shorturl.at/gtJQ2.

With the under vehicle area reconstructed, it empowers many other applications for vehicle surrounding visualization. Fig. 2b shows two sample applications. The top image shows a helicopter view of the UVR with a semi-transparent 3D vehicle model embedded into the scene. The viewport can move freely while the vehicle moves inside the simulated world. A demo video of transparent vehicle can be found with link: https://shorturl.at/bkmAH. The bottom image of Fig. 2b shows an example of transparent hood application, where the driver can intuitively see the ego vehicle's positions in regard to the ground. Transparent hood is helpful in cases where we need to pass through bumps or holes on the ground for smooth driving experience. A demo video of transparent hood can be found with link: https://shorturl.at/iuxBY.

4.2. Real data experiments

One sample result of Under Vehicle Reconstruction using real vehicle data is shown in Fig. 2c, and one map example where the UVR area is retrieved is shown in Fig. 2d.

UVR quality can be evaluated by examining how landmarks pixels from various places, e.g. both front camera area and under vehicle area, are stitched together. For example, in Fig. 2c, the "STOP" sign is divided into two parts. The left bottom part is under the vehicle and invisible to all the four fisheye cameras, while the rest are inside the front and right fisheye camera Field of View. A demo video showing how the ego vehicle passing through the "STOP" sign can be found with the link: https://shorturl.at/apvI6.

Under Vehicle Reconstruction not only can help with the missing information blind-spot problem for the under vehicle area, but also empowers other visualization applications in real world. One application is to provide a helicopter view of the vehicle as shown in Fig. 1a (A video demo can be found with link: https://shorturl.at/rsHKT). By configuring with appropriate transparency level, we can "see through" the vehicle at any angle around the vehicle or even inside the vehicle, which is beneficial for many cases such as assisting lane changing in the freeway.

5. Conclusion

In this paper, we propose a novel solution for blind-spot-free vehicle surround visualization by adding under vehicle reconstruction into the traditional surround view system.

For UVR's limitation, since we assume that the physical under vehicle area at present is the same as in the past, the solution will fail if there are dynamic objects moving under the vehicle, such as a pet running around. However, we argue that such extreme cases rarely appear in our daily driving. Moreover, the drivers are expected to stop the vehicle to check the under vehicle area rather than relying on assistive visualization system if these edge cases happen.

References

[1] Feng Hu, Niranjan Avadhanam, Yuzhuo Ren, Sujay Yadawadkar, Sakthivel Sivaraman, Hairong Jiang, and Wu Siyue. Gaze detection using one or more neural networks, U.S. Patent 11144754, Oct. 2021. 2

[2] Feng Hu. Robust seatbelt detection and usage recognition for driver monitoring systems. *AAAI 2022 Workshop on Trustworthy Autonomous Systems Engineering*, 2022. 2

[3] Jose Domingo Esparza Garca. *3D Reconstruction for Optimal Representation of Surroundings in Automotive HMIs, Based on Fisheye Multi-Camera Systems.* PhD thesis, 2015. 2

[4] Vikram Appia, Hemant Hariyani, Shiju Sivasankaran, Stanley Liu, Kedar Chitnis, Martin Mueller, Umit Batur, and G Agarwa. Surround view camera system for adas on ti's tdax socs. *Texas Instruments Technical Note*, 2015. 2

[5] Tianjun Zhang, Nlong Zhao, Ying Shen, Xuan Shao, Lin Zhang, and Yicong Zhou. Roecs: A robust semi-direct pipeline towards online extrinsics correction of the surround-view system. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3153–3161, 2021. 2

[6] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000. 2

[7] Yuzhuo Ren and Feng Hu. Camera calibration with pose guidance. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2180–2184. IEEE, 2021. 2

[8] Tuotuo Li, Feng Hu, and Zheng Geng. Geometric calibration of a camera-projector 3d imaging system. In *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*, pages 187–194, 2011. 2

[9] Feng Hu, Yuzhuo Ren, Niranjan Avadhanam, and Ankit Pashiney. System and method for optimal camera calibration, U.S. Patent 2021/010575, April. 2021. 2

[10] Buyue Zhang, Vikram Appia, Ibrahim Pekkucuksen, Yucheng Liu, Aziz Umit Batur, Pavan Shastry, Stanley Liu, Shiju Sivasankaran, and Kedar Chitnis. A surround view camera solution for embedded systems. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 662–667, 2014. 2

[11] Florian Raudies and Heiko Neumann. A review and evaluation of methods estimating ego-motion. *Computer Vision and Image Understanding*, 116(5):606–633, 2012. 3

[12] Jwu-Sheng Hu and Ming-Yuan Chen. A sliding-window visual-imu odometer based on tri-focal tensor geometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 3963–3968. IEEE, 2014. 3

[13] Patrick Wallrath and Reinhold Herschel. Egomotion estimation for a sensor platform by fusion of radar and imu data. In *2020 17th European Radar Conference (EuRAD)*, pages 314–317. IEEE, 2021. 3