MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

Visual Reconstruction of an Intersection by Integrating Cameras on Multiple Vehicles

Daisuke Ota Shintaro Ono Katsushi Ikeuchi
The University of Tokyo
Institute of Industrial Science, 4-6-1 Komaba, Meguro-ku, Tokyo, JAPAN
{ota, onoshin, ki}@cvl.iis.u-tokyo.ac.jp

Abstract

A method is presented for generating a bird's-eye view of a road traffic scene via the integration of multiple images from in-vehicle cameras; the resulting bird's-eye view supports the goal of greater traffic safety. As there are typically many blind spots at any given intersection which could lead to accidents or traffic jams, if views can be obtained from multiple vehicles coming into an intersection from different directions, a bird's-eye view can be generated by integrating the images seen from each vehicle; images are projected onto a common ground plane to reproduce the intersection virtually. The presented method assumes that the geometry of the scene and all camera intrinsic and extrinsic parameters relative to the car body are known. We estimate the orientation of each vehicle using vanishing points in each image and the 3D position of the foreground in an image by noting the volume of the intersection contained in the view.

1. Introduction

One of the most important goals in any Intelligent Transport System (ITS) is to enhance driving safety. Several approaches for supporting safe driving using sight information have been proposed. Eliminating blind spots and supporting parallel parking using backward-pointing cameras has already been put to practical use. There have also been attempts to make drivers more likely to notice dangerous objects in blind spots using fixed infrastructure cameras.

Safe driving requires good perception of environmental conditions, appropriate judgment, and actions matched to the situation. Many traffic accidents are caused by a lack of perception, and since humans perceive the external world mainly by sight information it can be very useful to assist driver perception with camera images. However, a driver's field of view is not fundamentally strengthened by installing a single camera on the front of a car. Several approaches to alerting drivers to objects which are in blind spots using infrastructure camera images as well as images from in-vehicle cameras have been proposed in [1], [2] and [6].

We propose removing drivers' blind spots without infrastructure cameras. This is useful because the number of vehicles with onboard cameras has increased dramatically in recent years; there are potentially many views from multiple points in intersections. An image generated with a minimum of blind spots – by integrating cameras on multiple vehicles – could be distributed back to the drivers of those vehicles as a safety aid. A virtual bird's-eye view with the estimated 3D positions of all vehicles and pedestrians could therefore be very helpful; a method of constructing a bird's-eye view is presented in this paper.

It is very hard to achieve such a system using only infrastructure cameras because a lot of costly infrastructure investment and maintenance is needed. In contrast, our proposed system's usefulness is expanded with the addition of each driver, and our system could be self-organizing or even incorporate infrastructure if it is available. Real-time information about city driving conditions is collected from the bottom-up in this system, and because consumers increasingly tend to use the internet rather than published guide books (which themselves represent a top-down approach) such information when uploaded to the web could be extremely useful – not only as a convenience for slow-traffic avoidance but also in map-making and even disaster response. In all these ways, our proposed system could play a significant role.

This paper is organized as follows: Section 2 outlines our proposed system. Section 3 describes the generation of a virtual bird's-eye view. Section 4 describes the 3D position estimation of foreground objects. Section 5 describes a preliminary experiment using CG simulation images and results. Finally, Section 6 outlines our conclusions and potential future research.

2. Outline of proposed system

When two or more vehicles enter an intersection from different directions, images from cameras on each vehicle are to be transmitted to each other; each car is then to generate a virtual bird's-eye view by integrating the images. We then deliver the integrated result to each vehicle present to minimize the number of blind spots.

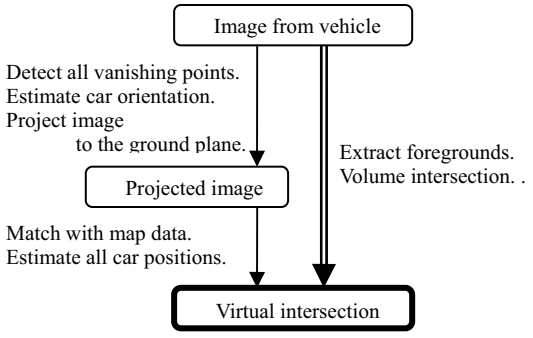

Figure 1: Proposed system outline

In this section, we describe the core of our system, which is the processing of multiple images. We assume that vehicle positions are roughly known via GPS. We assume that the geometry of roads and buildings, camera intrinsic parameters and the camera geometry relative to the car body are also known.

First, we detect the vanishing point in each of the images and thereby estimate the orientation of each vehicle. Next, we project each image onto the ground plane and integrate all the input images by matching them with known map information regarding the particular intersection. Next, we extract foreground objects such as vehicles that appear in each image from the background and estimate the 3D positions of these vehicles using a volume intersection method. Finally, we visually reconstruct virtual intersections using the obtained information.

3. Generating a virtual bird's-eye view

Generation of a virtual bird's-eye view is explained in this section. We assume that the following are known:

- vehicle rough position (which intersection a vehicle entered, and its direction of travel)
- all camera intrinsic performance parameters
- camera geometry relative to each vehicle's body
- map information for the particular intersection

We integrate images from multiple vehicles on the ground plane using these assumptions.

3.1. Estimating vehicle orientation using vanishing points

Vehicles entering an intersection should usually be driving roughly parallel with the road, so we can estimate the orientation of each vehicle relative to the road using vanishing points. In each image, we detect edge points and lines using a Hough transform; the intersection of these lines defines a locus of vanishing points. We remove any lines which are nearly perpendicular with respect to the gap between the orientation of the road and the direction of the locus of vanishing points in the image. By discarding these outliers, we can more easily detect the lines corresponding to the left and right sides of the road. Criteria which define useful vanishing points:

1) Retain vanishing points which lie in an image's vertical upper half and are "near" the horizontal middle of the image.
2) Retain vanishing points formed by the intersections of lines with edge points belonging equally to the left half and right half of the image.

The most useful vanishing point is formed by the intersection of the lines most closely matching the edges of the road. Projecting the image to the ground plane, we can estimate the angular gap of the view direction vs. the direction of the road.

3.2. Projection images

To project images seen from multiple vehicles to the ground plane and integrate them, a 3D point $\mathbf{x} = (x, y, \ldots)^{\mathrm{T}}$ is projected to a 2D point using (1), in

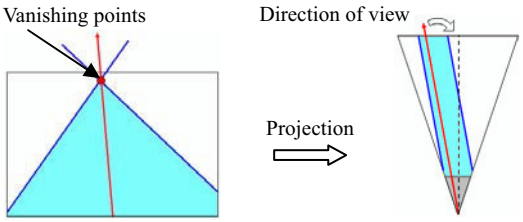

Figure 2: Estimating Vehicle Orientation.
Left: Vanishing points in the original image,
Right: Direction of View line in projected image.

which $\tilde{\mathbf{x}}$ represents a homogeneous coordinate vector. **P**, **A**, **R** and t (2) express a projection matrix. Note that (2) represents the known intrinsic parameters of a particular camera's orientation and translation, respectively.

$$\tilde{\mathbf{m}} \cong \mathbf{P}\tilde{\mathbf{X}} = \mathbf{A}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\tilde{\mathbf{X}} \tag{1}$$

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} \alpha_u & -\alpha_u \cot\theta & u_0 \\ 0 & \alpha_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix} = \begin{bmatrix} \mathrm{r}_{11} & \mathrm{r}_{12} & \mathrm{r}_{13} & \mathrm{t}_1 \\ \mathrm{r}_{21} & \mathrm{r}_{22} & \mathrm{r}_{23} & \mathrm{t}_2 \\ \mathrm{r}_{31} & \mathrm{r}_{32} & \mathrm{r}_{33} & \mathrm{t}_3 \end{bmatrix}$$

Matrix **A** is known because a camera's intrinsic parameters have been measured or are specified. The camera's rotation relative to a car body is assumed known, while the orientation of the car is derived as above using vanishing points, therefore matrix **R** is known and we can project camera images to the ground plane using the composite projection matrix **P**.

3.3. Matching known road information with projected images

Map information regarding the intersection is known, so by matching each projected image to the map we can estimate each car's position and also that car's camera position. The map data includes road elements and buildings on a 2D plane, as shown in Figure 3. We detect edge points in the images projected to the ground plane and search for the place where the derived edge points best match the known road parts on our map. We then add the building elements included in the map data to the projected image. Figure 3 shows a typical matching result. We can integrate multiple images projected to the ground plane in this way, as shown in Figure 4.

4. Estimating 3D positions of foreground objects

There would typically be other vehicles and pedestrians in actual input camera images. We have to estimate the positions of these objects to enhance traffic safety;

extracting foreground objects and estimating their 3D positions is described below.

4.1. Extracting foreground objects

Two methods are used to extract foreground objects such as cars from the background: matching edge points, and exploiting differences between the input images.

The first method involves searching for the best match between edge points and the known map data. This method is simple and effective when the image is primarily background. But when there are objects such as cars in the image, these foreground objects occlude edge points and therefore interfere with the match. Figure 5 shows such interference, resulting in mismatched road element areas, in green. One could nevertheless improve this method of extracting foreground objects using graphcut methods as in [3], [4].

The second method of extracting foreground objects uses the differences between input images, as shown in the bottom panes of Figure 5.

4.2. Estimating 3D position using the silhouette volume intersection technique

We estimate the 3D positions of foreground objects based on the silhouette volume intersection method [5]. We can reconstruct the 3D volume object from multiple-viewpoint silhouette images, as shown in Figure 8. The algorithm is as follows:

- When a 3D point is visible in silhouette in all images, it is a point inside an object body.
- All other points are outside the object body.

5. Experiments and Results

We performed two experiments. First, we generated a virtual intersection using CG. Second, we performed an indoor experiment in which two corridors in a building crossed at right angles, similar to a typical intersection.

5.1. Experimental Using CG Simulated Camera Images

In the first experiment, we placed the road and buildings in the CG intersection and cars. Four cars (No. 1-4) have cameras focused on the intersection and the car (No.5) amidst the intersection has a camera focused on the No.4 car, as shown in Figure 6.

We generated a virtual bird's-eye view (Figure 7, left) and reconstructed each view seen from each camera (Figure 7, right) as below:

1) We estimated the orientation of the No. 5 vehicle with vanishing points: We detected vanishing points in images obtained by cameras and estimated the orientation (yaw angle) of the car body.

2) We estimated each camera position by integrating all the images. The estimated camera position is almost equal to the true value and the estimation error margin is smaller than five pixels.

3) We selected foreground-object candidates in elements with mismatching edges and as being parts where the differences from other images are large. Graph-cut techniques were used to extract foreground objects from the background.

We estimated 3D object positions using the silhouette volume intersection algorithm.

5.2. Experimental Using Real Images

First we placed real cameras focused onto this "intersection" and an object (a red box) which simulates a car amidst this intersection, as shown in Figure 8, left. The ratio of the corridor's width to a typical road width is the same as the ratio of the box's size to a typical car's size. Each image seen from each camera is shown in Figure 8, right. As a result, we also generated a virtual bird's-eye view (Figure 9, left) and reconstructed each view (Figure 9, right).

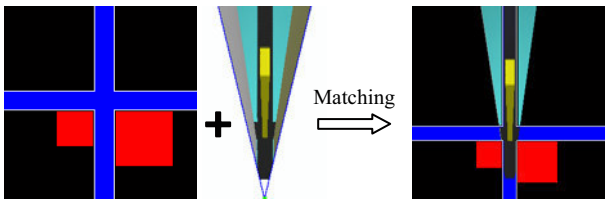

Figure 3: Typical Match of Map Data to an Input Image Left: Map data in which white, blue and red illustrate road edges, the road, and buildings, respectively. Middle: Input image projected to the ground plane. Right: Resulting integrated image.

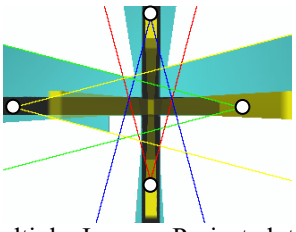

Figure 4: Multiple Images Projected to the Ground Plane and Integrated. White circles are estimated camera positions; cones indicate the field of view of input images.

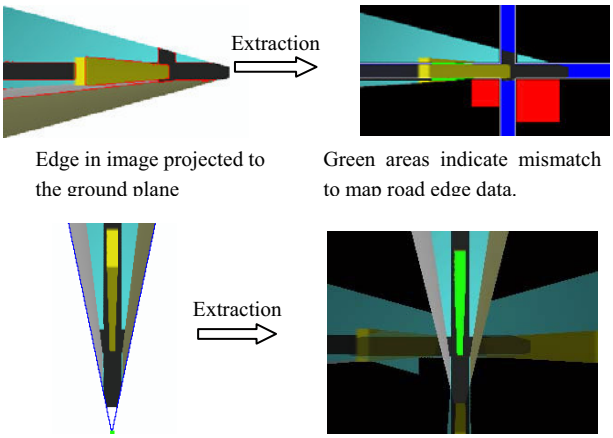

Figure 5: Extracting foreground objects. Top: Using edge points. Bottom: by differences from other images

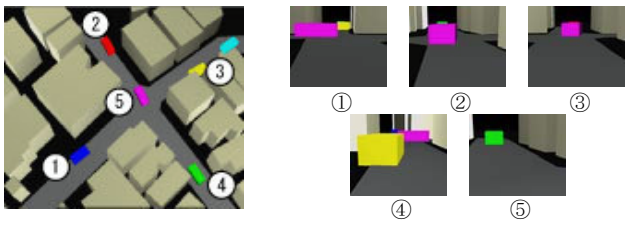

Figure 6: Experimental virtual environment. Left: No. 1-5 cars have cameras focused upon the intersection. Brown objects are buildings. Right: Images of the No.5 object are seen from the other cars.

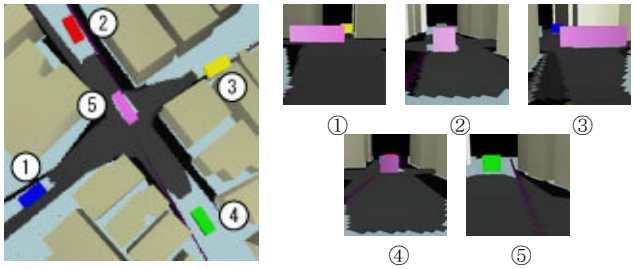

Figure 7: Reconstructed results. Left: A virtual bird's-eye view, Right: Reconstructed view seen from each cars.

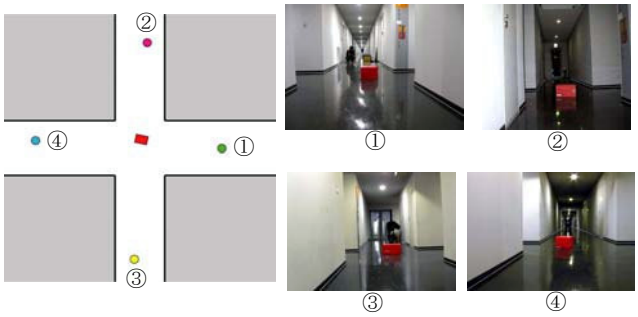

Figure 8: Experimental environment. Left: Four circles represent cameras focused onto an intersection. A red object is amidst the intersection. Gray Regions are in walls. Right: Images of the red object are seen from the each camera.

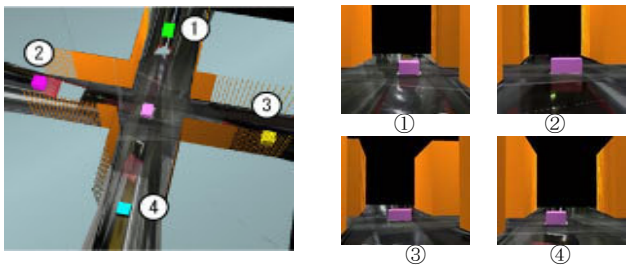

Figure 9: Reconstructed results. Left: A virtual bird's-eye view, Right: Reconstructed view seen from each camera.

5.3. Discussion

The effectiveness of this system is high in our artificial CG environment; however, this paper does not consider many real-world complications such as:

- Too many cars in an intersection: resulting images are too occluded and their distances too short to yield proper vanishing points,
- Too few cars in an intersection: not enough input data, or the camera viewpoints are not adequately orthogonal,
- Incidental occlusion by buildings, traffic signs, street trees and so on,
- Weather conditions which could affect camera noise,
- Lighting conditions which could affect camera noise,
- Certain styles of intersection (e.g., roundabouts, underpasses) which could introduce additional complexity,
- The "freshness" of the data: camera latency, Doppler effects; processing time; image redistribution time; etc.

6. Conclusions

In this paper, we propose a system that generates a bird's-eye view in order to minimize driving blind spots by integrating images seen from multiple vehicles near or in an intersection. We detect the direction of roads using vanishing points and calculate the orientations of camera-bearing cars. We project all the input images to the ground plane and estimate all the camera positions by matching the projected images to known map data. We estimate the 3D positions of foreground objects such as other cars or pedestrians using a silhouette volume intersection algorithm, possibly also requiring graphcut techniques. We then generate the resulting virtual bird's-eye view by integrating all this information.

We simulated a test of our proposed method in a CG environment and found its performance to be favorable. We plan to conduct more experiments using a real environment to enhance our methodology. In addition, we plan to evaluate this system quantitatively and characterize its telecommunication and computing needs for practical, real-world applications.

References

[1] K. Kojima, A. Sato, F. Taya, Y. Kameda and Y Ohta: "NaviVIew: Visual Assistance by Virtual Mirrors at Blind Intersection", ICTS 2005

[2] F. Taya, I. Kitahara, Y. Kameda and Y. Ohta: "NaviView: Driver Vision Enhancement by Sensing Live Environment", Proceedings of Systems and Information of SICE (SSI2005), pp.168-173, 2005.

[3] Y. Li, J. Sun, C. Tang and H. Shum: "Lazy Snapping", SIGGRAPH. 2004

[4] C. Rother, V. Kolmogorov and A. Blake: "Interactive Foreground Extraction using Iterated Graph Cuts", SIGGRAPH'04, 2004.

[5] T. Wada, X. Wu, S. Tokai and T. Matsuyama: "Parallel Volume Intersection Based on Plane-to-Plane Projection", IPSJ SIG-CVIM, Vol.42, No.SIG 6 (CVIM 2), pp.33-43, 2001

[6] T. Nakanishi, T. Yendo, T. Fujii and M. Tanimoto: "Right Turn Assistance System at Intersections by Vehicle-Infrastructure Cooperation", IV 2006

[7] M. Murao, Y. Matsushita, K. Ikeuchi and M. Sakauchi: "Visualization of Traffic Conditions for Drivers", UM3' 2000