2004 IEEE Intelligent Vehicles Symposium
University of Parma
Parma, Italy • June 14-17, 2004

Motion Based Vehicle Surround Analysis Using an Omni-Directional Camera

Tarak Gandhi and Mohan M. Trivedi
Computer Vision and Robotics Research Laboratory
University of California at San Diego, USA
{tgandhi,mtrivedi}@ucsd.edu
http://cvrr.ucsd.edu

Abstract

Omni-directional cameras which give 360 degree panoramic view of the surroundings have recently been used in many applications such as robotics, navigation and surveillance. This paper describes the application of motion estimation on omni camera to perform surround analysis using an automobile mounted camera. The system detects and tracks the surrounding vehicles by compensating the ego-motion and detecting objects having independent motion. Prior knowledge about ego-motion and calibration is optimally combined with the information from the image gradients to get better motion compensation.

1 Introduction

A vehicle surround analysis system that monitoring the presence of other vehicles in all directions is important for on-line as well as off-line applications. On-line systems are useful for intelligent driver support. On the other hand, off-line processing of video sequences is useful for study of behavioral patterns of the driver in order to develop better tools for driver assistance. However, since on-line systems have stringent requirements on camera placement and algorithm speed, we are currently working on off-line processing to develop robust algorithms, and then optimize the efficiency and camera placement to get on-line systems.

Omni-directional cameras give a 360 degree field of view of the surroundings and have been recently used in many applications [3]. Use of omni cameras from moving platforms is particularly interesting, since motion estimation ambiguities which exist with narrow field of view cameras are reduced. This paper explores the use of omni-directional cameras for complete surround analysis from a vehicle test platform. The system detects and tracks other vehicles and separates them from extraneous features such as road marks and shadows.

The current emphasis is on developing an off-line system which analyzes video sequences obtained from test runs in order to study the driver behavior patterns. To get full 360 degree surround view with least amount of FOV occupied by the car itself, a specially designed fixture was used to mount the camera two feet above the roof of the car. The vehicle test platform mounted with omni-directional camera for is shown in Figure 1 (a). A typical image from the camera is shown in Figure 1 (b). It is seen that the camera covers a 360 degrees field of view around its center. However, such a fixture would not be suitable for on-line systems, hence alternative camera placements are being explored.

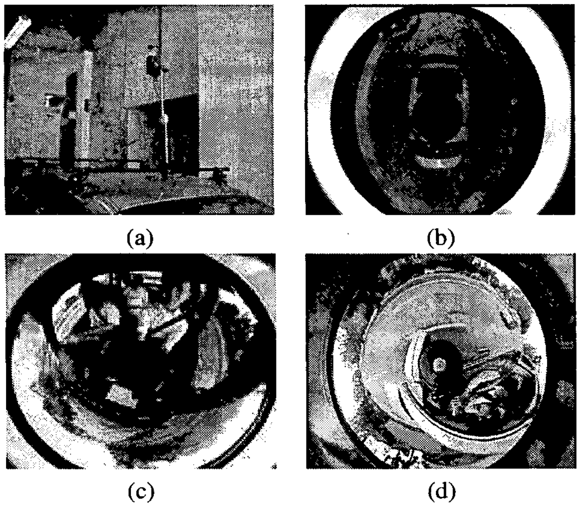
For example, in our earlier work [10], we had used omni cameras mounted inside the car and on vehicle side in order to get the view of both inside and outside of the car. Figure 1(c) shows the camera mounted inside the car. This configuration could simultaneously capture the driver face, as well as the surroundings. The face pose of the driver was analyzed and the view where the driver is looking was generated from the same camera. For side surround, configuration in Figure 1 (d) was more suitable, but the driver's view was less clear. Preliminary results of vehicle detection on this configuration were described in [10]. However, only one side of the car could be observed, and the car itself covered a large part of FOV.

1.1 Previous Work

Motion analysis has been used to separate ego-motion of the background and detect obstacles. Motion-compensated difference images were used for obstacle detection by [4]. Robust real time motion compensation for ground plane is also described in [14]. In [5], a system for video-based driver assistance involving lane and obstacle detection using rectilinear camera is described.

Motion estimation from moving omni cameras has recently been a topic of great interest. As noted by Gluckman and Nayar [8], omni cameras alleviate the motion ambiguity problems commonly encountered with narrow field of view cameras. Ego-motion of a moving platform is usually determined by projecting the image motion on a spherical [8] surface using Jacobians of transformations. Shakernia et

0-7803-8310-9/04/$20.00 © 2004 IEEE


(a) (b)

(c) (d)

Figure 1: (a) Vehicle Test Platform with omni camera mounted two feet above the roof using a specially designed fixture. (b) Omni-directional image acquired the camera. (c) and (d) Image from omni-camera mounted inside the car and on side of the car respectively.

al. [15] use the concept of back-projection flow, where the image motion is projected to a virtual curved surface in place of spherical surface and make the Jacobians of transformation simpler.

Parametric motion estimation based on image gradients, also known as the "direct method" has been used for rectilinear cameras in for planar motion estimation, obstacle detection and motion segmentation [12, 13]. The advantage of the direct methods is that they can use motion information not only from corner-like features, but also edges, which are usually more numerous in an image. The direct estimation approach was generalized in [7, 10] for motion compensation using omni cameras, where parameters of planar homography are estimated. A modification of that approach is used here with the omni camera mounted on an automobile to detect nearby vehicles and generate a complete surround view showing the position and tracks of the vehicles.

2 Vehicle Surround Analysis System

The block diagram of the surround analysis system is shown in Figure 2. The initial estimates of the the road plane motion parameters are obtained using the approximate knowledge about the camera calibration and speed. Using these parameters, one of the frames is warped towards another to compensate the motion of the ground plane. However, the motion of features having independent motion or height above the ground plane is not fully compensated. To detect these features, the normalized image difference between the two images is computed using temporal and spatial gradients. Morphological and other post-processing is performed to further suppress the ground features due to any residual motion and get the positions of the objects. The detected objects are then tracked over frames. To account for the inaccuracy in prior knowledge of ego-motion, the parameters are iteratively corrected using the spatial and temporal gradients of the motion compensated images using optical flow constraint in coarse to fine framework. The motion information contained in these gradients is optimally combined with the prior knowledge of motion parameters using a Bayesian framework similar to [14]. Robust estimation is used to separate the ground plane features from other features.

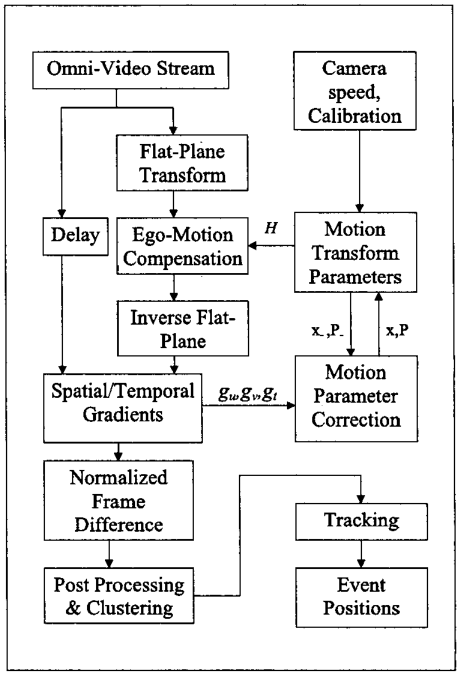

Omni-Video Stream
Camera speed, Calibration
Flat-Plane Transform
Delay
Ego-Motion Compensation
H
Motion Transform Parameters
Inverse Flat-Plane
x_,P_
x,P
Spatial/Temporal Gradients
g_u, g_v, g_t
Motion Parameter Correction
Normalized Frame Difference
Tracking
Post Processing & Clustering
Event Positions

Figure 2: Block diagram for event detection and recording system based on ego-motion compensation from a moving platform.

3 Motion Estimation from Omni Camera

The omni-directional camera used in this work consists of a hyperbolic mirror and a camera placed on its axis. It belongs to a class of cameras known as central panoramic catadioptric cameras [3]. These cameras have a single viewpoint that permits the image to be suitably transformed to obtain perspective views. To compensate the motion of the omni-directional camera, the omni transformation is combined with that due to motion.

3.1 Planar Motion Transformation

The motion of the road is modeled as a planar surface. Let P_a and P_b denote the perspective transforms of a point on the ground plane in the homogenous coordinate systems corresponding to two positions A and B of the moving camera. Then, P_a and P_b are related by a projective transform:

$$\lambda_b P_b = \lambda_a (R + DK^t) P_a = \lambda_a H P_a \tag{1}$$

where H is the homography matrix expressed in terms of rotation matrix R, translation vector D, and plane normal vector K.

Let C denote a nominal camera coordinate system, based on the known camera calibration. The actual camera system at any given time is assumed to have small rotation w.r.t. this system due to vibrations. Use of nominal system allows one to treat small rotations as angular displacement vectors. The ego-motion state is then expressed as:

$$\mathbf{x} = \begin{pmatrix} V \\ W \\ A \end{pmatrix} \tag{2}$$

where V is the camera linear velocity, W is the angular velocity, and angular displacement between nominal and actual camera system is A, all expressed in nominal camera system. The homography matrix H can be approximately expressed in terms of V, W, A as shown in Table 1.

3.2 Flat plane transformation

The geometry of a hyperbolic omni-directional camera is shown in Figure 3. According to the mirror geometry, a the light ray from the object towards the viewpoint at the first focus O is reflected so that it passes through the second focus, where a conventional rectilinear camera is placed.

Let $P = (X, Y, Z)^T$ denote the homogenous coordinates of the perspective transform of any 3-D point λP on ray OP, where λ is the scale factor depending on the distance of the 3-D point from the origin. It can be shown [1, 11, 15] that the reflection in mirror gives the point $(-x, -y)^T$ on the image plane of the camera using the flat-plane transform F:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{q_1}{q_2 Z + q_3\sqrt{X^2+Y^2+Z^2}} \begin{pmatrix} X \\ Y \end{pmatrix} \tag{3}$$

The pixel coordinates $(u, v)^T$ are then obtained by using the calibration matrix of the conventional camera.

For performing motion compensation using omni-directional cameras, the motion transforms are combined with the flat-plane transform as in [7] to warp every point in image B towards image A.

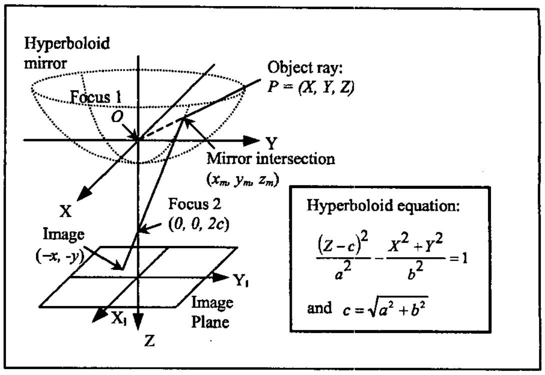


Figure 3: Omni-directional camera geometry.

3.3 Estimation

To estimate the ego-motion parameters, the parametric image motion is substituted into the optical flow constraint[9]:

$$g_u \Delta u + g_v \Delta v + g_t = 0 \tag{4}$$

where g_u, g_v are spatial gradients, and g_t is the temporal gradient. Since the image motion $(\Delta u, \Delta v)$ at each point i can be represented as a function of the incremental state vector $\Delta\mathbf{x}$, the optical flow constraint (4) for image points $1 \ldots N$ can be expressed as:

$$\Delta\mathbf{z} = \mathbf{c}(\Delta\mathbf{x}) + \mathbf{v} \simeq \mathbf{C}\Delta\mathbf{x} + \mathbf{v} \tag{5}$$

where

$$\mathbf{c}(\Delta\mathbf{x}) = \begin{pmatrix} (g_u\Delta u + g_v\Delta v)_1 \\ \vdots \\ (g_u\Delta u + g_v\Delta v)_N \end{pmatrix}, \Delta\mathbf{z} = -\begin{pmatrix} (g_t)_1 \\ \vdots \\ (g_t)_N \end{pmatrix} \tag{6}$$

and $\mathbf{v}$ is the vector of measurement noise in the time gradients, and $\mathbf{C} = \partial\mathbf{c}/\partial\mathbf{x}$ is the Jacobian matrix computed using chain rule as in [7]. The function $\mathbf{c}(\mathbf{x})$ is a non-linear. The i^{th} row of its the Jacobian is given by the chain rule: The function $\mathbf{c}(\mathbf{x})$ is a non-linear. The i^{th} row of its the Jacobian is given by the chain rule:

$$\mathbf{C}_i = \left(\frac{\partial \mathbf{c}_i}{\partial \mathbf{x}}\right) = \left(\frac{\partial \mathbf{c}}{\partial w_b}\frac{\partial w_b}{\partial p_b}\frac{\partial p_b}{\partial P_b}\frac{\partial P_b}{\partial h}\frac{\partial h}{\partial \mathbf{x}}\right)_i \tag{7}$$

where $P_b = (X_b, Y_b, Z_b)^T$, $p_b = (x_b, y_b)^T$ and $w_b = (u_b, v_b)^T$ are the coordinates of the point in the camera, image, and pixel coordinate systems for camera position B, and h is the vector of elements of H. The individual Jacobians are computed similar to [7]. The relationship between these variables, and their Jacobians are shown in Table 1.

Since the points having very low texture do not contribute much to the estimation of motion parameters, only those

$\mathbf{x} = \begin{pmatrix} V \\ W \\ A \end{pmatrix}$	$H = R + DK^T$	$\partial H = \partial R + \partial D.K^T + D(\partial K)^T$
$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix}$	$R \simeq I - W_\times \Delta t$ $D \simeq [I - A_\times] V \Delta t$ $K \simeq [I - A_\times] K_0$	$\partial R = \partial W_\times \Delta t$ $\partial D = (I - W_\times \Delta t - A_\times)\Delta t \partial V - (\partial W_\times \Delta t + \partial A_\times) V \Delta t$ $\partial K = -A_\times K_0$ $\partial V / \partial V_i = e_i, \partial W_\times / \partial W_i = \partial A_\times / \partial A_i = (e_i)_\times$
$h = \begin{pmatrix} h_1 & \dots & h_9 \end{pmatrix}^T$	$\begin{pmatrix} X_b \\ Y_b \\ Z_b \end{pmatrix} \equiv \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} X_a \\ Y_a \\ Z_a \end{pmatrix}$	$\frac{\partial P_b}{\partial h} = \begin{pmatrix} X_a & Y_a & Z_a & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X_a & Y_a & Z_a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Y_a & Z_a \end{pmatrix}$
$P = \begin{pmatrix} X & Y & Z \end{pmatrix}^T$	$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} = \frac{q_1}{q_2 Z + q_3 \lVert P \rVert} \begin{pmatrix} X \\ Y \end{pmatrix}$	$\frac{\partial p}{\partial P} = \frac{1}{(q_2 Z + q_3 \lVert P \rVert) \lVert P \rVert} \cdot \begin{pmatrix} q_3 x X - q_1 \lVert P \rVert & q_3 x Y & q_3 x Z \\ q_3 y X & q_3 y Y - q_1 \lVert P \rVert & q_3 y Z \end{pmatrix}$
$p = \begin{pmatrix} x & y \end{pmatrix}^T$	$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_u & s & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$	$\frac{\partial w}{\partial p} = \begin{pmatrix} f_u & s \\ 0 & f_v \end{pmatrix}$
$w = \begin{pmatrix} u & v \end{pmatrix}^T$	$\mathbf{c} = \begin{pmatrix} g_u & g_v \end{pmatrix} \begin{pmatrix} u_b - u_a \\ v_b - v_a \end{pmatrix} = -g_t + \eta$	$\frac{\partial \mathbf{c}}{\partial w_b} = \begin{pmatrix} g_u & g_v \end{pmatrix}$

Table 1: Chain of functions and Jacobians leading from state vector **x** to optical flow constraint **c**.

image points having gradient magnitude above a threshold value are selected for performing estimation. Alternatively, a non-maximal suppression is performed on the image gradients, and the image points with local maxima are used. This way, instead of computing Jacobians using multiple image transforms over the entire image, the Jacobians are computed only at the selected points which have significant information for estimating parameters.

The estimates of the state **x** and its covariance **P** are iteratively updated using the measurement update equations of the iterated extended Kalman filter [2],

$$\mathbf{P} \leftarrow \left[\mathbf{C}^T \mathbf{R}^{-1} \mathbf{C} + \mathbf{P}_-^{-1}\right]^{-1} \tag{8}$$

$$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} + \Delta\hat{\mathbf{x}} = \hat{\mathbf{x}} + \mathbf{P}\left[\mathbf{C}^T \mathbf{R}^{-1} \Delta\mathbf{z} - \mathbf{P}_-^{-1}(\hat{\mathbf{x}} - \mathbf{x}_-)\right] \tag{9}$$

However, the optical flow constraint equation is satisfied only for small image displacements up to 1 or 2 pixels. To estimate larger motions, a coarse to fine pyramidal framework [16] is used. Also, the estimation process is highly sensitive to the presence of outliers, i.e. points not satisfying the ground motion model. A region of interest of road is determined using calibration information, and the processing is done only in that region to avoid as many extraneous features as possible. To reduce the effect of outliers, Robust-M estimation [6] is used to reduce the effect of outliers by iteratively reweighting the contribution of samples according to their error residuals.

4 Vehicle Detection and Tracking

After motion compensation, the features on the road plane would be aligned between the two frames, whereas those due to obstacles would be misaligned. Image difference between the frames would therefore enhance the obstacles, and suppress the road features. To reduce the dependence on local texture, the normalized frame difference [17] is used. This is given at each pixel by:

$$\frac{\langle g_t \sqrt{g_u^2 + g_v^2} \rangle}{k + \langle g_u^2 + g_v^2 \rangle} \tag{10}$$

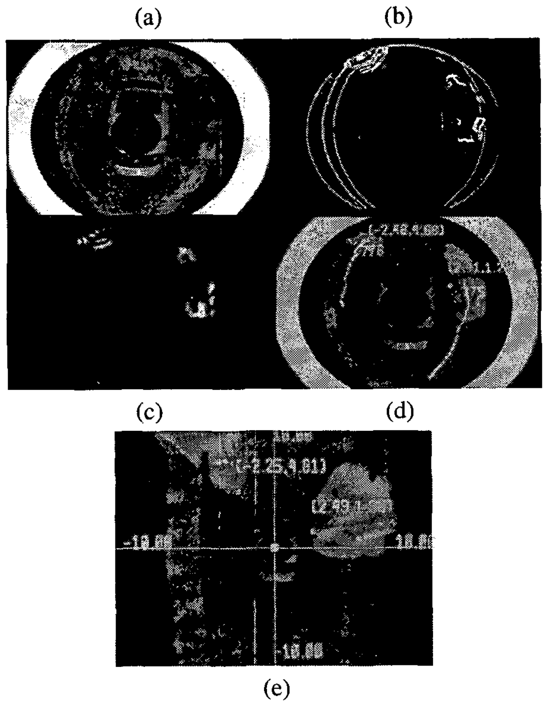
where g_u, g_v are spatial gradients, and g_t is the temporal gradient after motion compensation, and $\langle\cdot\rangle$ denotes a Gaussian weighted averaging performed over a $K \times K$ neighborhood of each pixel. In fact, the normalized difference is a smoothed version of the normal optical flow, and hence depends on the amount of motion near the point.

Due to untextured interior of a vehicle, blobs are usually detected at the sides of the vehicle. To get the full vehicle, it is assumed that if two blobs are within a threshold distance (5.0 meters) in the direction of car's motion, they constitute a vehicle. To detect this situation, the original image is unwarped using the flat plane transform, and a morphological closing is performed on the transformed image using a $1 \times N$ vertical mask.

After the blobs corresponding to moving objects are identified, nearby blobs are clustered and tracked over frames using Kalman filter [2]. The points on the blob that are nearest to the camera center usually correspond to the ground plane, and are marked as obstacle map. The vehicle position on the road is computed by projecting the track location on the obstacle map. Since the obstacle map is assumed to be on ground plane, the location of the vehicle can be obtained by inverse perspective transform.

5 Experimental Results

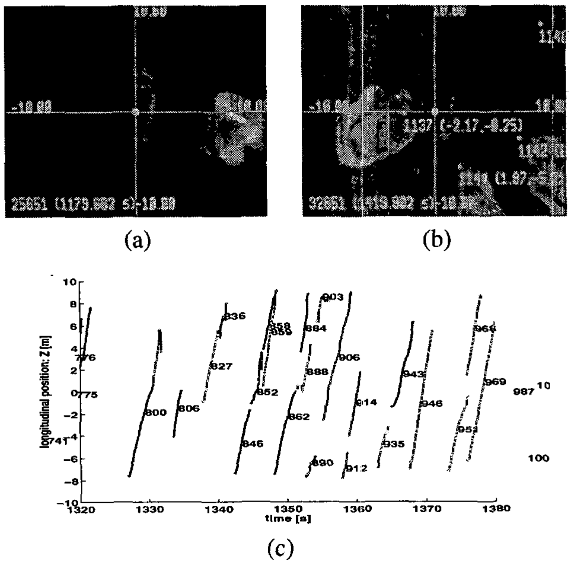
The vehicle detection approach was applied to an omni camera mounted on an automobile test-bed used for intelligent vehicle research. The test-bed is instrumented with a number of cameras and computers to capture synchronized video of the surroundings. In addition, the CAN bus of the vehicle gives information on vehicle speed, pedal and brake positions, radar, etc. The vehicle was driven with a speed up to 65 miles per hour. The actual vehicle speed, obtained from CAN bus was used for initial motion estimate. A video sequence of 36000 frames (20 minutes) was processed and vehicles on both sides of the car were detected as shown in Figure 4. The flat plane transform was applied to the omni image to perform surround analysis. Figure 5 (a), (b) shows the detection of moving vehicles in different road conditions. Figure 5 (c) shows the plots of track positions against time for a segment of the video. The algorithm was also applied to a video sequence from an omni camera mounted on the side of the vehicle as in [10]. Figure 6 shows the result of the detection algorithm. Note that the outside view covers only one side of the car, whereas a large part is occupied by the car itself and its driver.


(a) (b)

(c) (d)

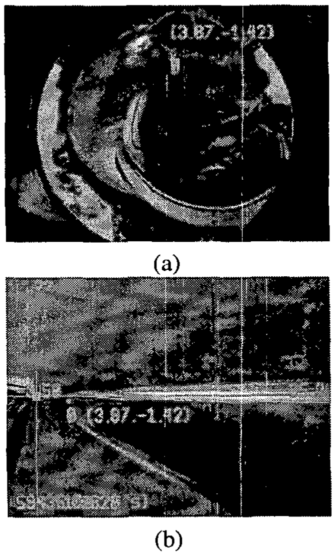
(e)

Figure 4: (a) Image from a sequence using omni camera mounted on a moving car with estimated parametric motion of ground plane. (b) Classification of points into inliers (gray), outliers (white), and unused (black). (c) Normalized difference between motion-compensated images. (d) Detection and tracking of moving vehicles marked with track id and the coordinates in road plane. (e) Surround view generated by dewarping omni image.


(a) (b)

(c)

Figure 5: Surround analysis in different situations with the top mounted camera: (a) City road (b) Freeway (c) Plot of the longitudinal position of vehicle tracks against time. The tracks are color coded as red, yellow and green according to increasing lateral distance from the ego-vehicle.


(a)

(b)

Figure 6: (a) Detection of a moving vehicle from a side mounted omni camera. (b) Surround view generated by dewarping the omni image.

6 Summary

This paper described a system to generate a 360 degree surround view using an omni-directional camera mounted on an automobile. The system detects and tracks vehicles by estimating and compensating the ego-motion of the road surface. For motion estimation, the planar motion transform is combined with the omni camera transform, and optical flow constraint is used to optimally combine the prior knowledge of ego-motion parameters with the information in the image. Experimental results demonstrate the vehicle detection in two configurations. The top mounted camera configuration was optimized for monitoring the entire surround for off-line studies of driver behavior. On the other hand, the side mounted configuration with some modifications in camera placement may be useful for monitoring the driver as well as the driver's side surround. The latter may also be more suitable for realistic placement in series cars. The current implementation operates at about 3 frames per second on a Pentium 4 system, hence suitable for use in off-line processing of video sequences, but algorithm optimization may yield better speeds.

Acknowledgments

We are thankful for the grant awarded by the Nissan Motor Company LTD, Japan and the DiMi California which provided the primary sponsorship of the reported research. We also thank our colleagues from the CVRR laboratory for their contributions and support, as well as the SIO workshop who specially designed the fixture for omni camera. In particular, we would like to thank Joel McCall for his support and Dr. Erwin Boer for his insightful suggestions on visualizing the results. We thank the reviewers for their insightful comments for improving the paper.

References

[1] O. Achler and M. M. Trivedi. Real-time traffic flow analysis using omnidirectional video network and flatplane transformation. In *Workshop on Intelligent Transportation Systems*, Chicago, IL, 2002.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation.* John Wiley and Sons, 2001.

[3] R. Benosman and S. B. Kang. *Panoramic Vision: Sensors, Theory, and Applications.* Springer, 2001.

[4] S. Carlsson and J. O. Eklundh. Object detection using model-based prediction and motion parallax. In *European Conference on Computer Vision*, pages 297–306, April 1990.

[5] W. Enkelmann. Video-based driver assistance: From basic functions to applications. *International Journal of Computer Vision*, 45(3):201–221, 2001.

[6] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* 2003.

[7] T. Gandhi and M. M. Trivedi. Motion analysis of omni-directional video streams for a mobile sentry. In *First ACM International Workshop on Video Surveillance*, pages 49–58, Berkeley, CA, November 2003.

[8] J. Gluckman and S. Nayar. Ego-motion and omnidirectional cameras. In *Proc. of the International Conference on Computer Vision*, pages 999–1005, 1998.

[9] B. Horn and B. Schunck. Determining optical flow. In *DARPA81*, pages 144–156, 1981.

[10] K. Huang, M. M. Trivedi, and T. Gandhi. Driver's view and vehicle surround estimation using omnidirectional video stream. In *IEEE Intelligent Vehicles Symposium*, Columbus, OH, June 2003.

[11] K. C. Huang and M. M. Trivedi. Video arrays for real-time tracking of persons, head and face in an intelligent room. *Machine Vision and Applications*, 14(2):103–111, 2003.

[12] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(6):577–589, June 1998.

[13] Q. Ke and T. Kanade. Transforming camera geometry to a virtual downward-looking camera: Robust ego-motion estimation and ground-layer detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages I:390–397, June 2003.

[14] W. Kruger. Robust real time ground plane motion compensation from a moving vehicle. *Machine Vision and Applications*, 11:203–212, 1999.

[15] O. Shakernia, R. Vidal, and S. Sastry. Omnidirectional egomotion estimation from back-projection flow. In *IEEE Workshop on Omnidirectional Vision*, June 2003.

[16] E. P. Simoncelli. Coarse-to-fine estimation of visual motion. In *Proc. Eighth Workshop on Image and Multidimensional Signal Processing*, pages 128–129, Cannes, France, 1993.

[17] E. Trucco and A. Verri. *Computer vision and applications: A guide for students and practitioners.* Prentice Hall, March 1998.