Video Stabilization for a Camcorder Mounted on a Moving Vehicle

Yu-Ming Liang, Hsiao-Rong Tyan, Shyang-Lih Chang, Hong-Yuan Mark Liao, and Sei-Wang Chen

***Abstract*—Vision systems play an important role in many intelligent transportation systems (ITS) applications, such as traffic monitoring, traffic law reinforcement, driver assistance, and automatic vehicle guidance. These systems installed in either outdoor environments or vehicles have often suffered from image instability. In this paper, a video stabilization technique for a camcorder mounted on a moving vehicle is presented. The proposed approach takes full advantage of the *a priori* information of traffic images, significantly reducing the computational and time complexities. There are four major steps involved in the proposed approach: global feature extraction, camcorder motion estimation, motion taxonomy, and image compensation. We begin with extracting the global features of lane lines and the road vanishing point from the input image. The extracted features are then combined with those detected in previous images to compute the camcorder motion corresponding to the current input image. The computed motion consists of both expected and unexpected components. They are discriminated and the expected motion component is further smoothed. The resulting motion is next integrated with a predicted motion, which is extrapolated from the previous desired camcorder motions, leading to the desired camcorder motion associated with the input image under consideration. The current input image is finally stabilized based on the computed desired camcorder motion using an image transformation technique. A series of experiments with both real and synthetic data have been conducted. The experimental results have revealed the effectiveness of the proposed technique.**

***Index Terms*—Image compensation, intelligent transportation systems (ITS), in-vehicle vision systems, motion estimation, motion taxonomy, video stabilization.**

I. Introduction

OVER the past several years, there has been a growing interest in intelligent transportation systems (ITS) [13], [19], [24]. ITS, which integrates technologies from diverse areas, attempts to effectively utilize limited resources while maximizing safety, efficiency and comfort of transportation. Three major areas constitute surface transportation uses of ITS; they are, intelligent roads (e.g., automatic traffic signal control, electrical toll collection, vehicle guidance, and automatic accident/incident detection), intelligent vehicles (e.g., automatic vehicle location, automatic collision avoidance, route planning, and driver assistance), and intelligent systems (e.g., travel services, demand management and operations, emergency management, and integrated service networks). Vision systems [12] have played an important role in the first two areas.

Unlike vision systems developed for applications of intelligent roads, in which the visual sensors (e.g., cameras and camcorders) utilized are commonly static with respect to the ground, those sensors employed in intelligent vehicles are typically dynamic in the sense that they are mounted on moving vehicles. Both static and dynamic visual sensors have suffered from image/video instability. The instability associated with static sensors may result from wind blowing, passing objects (e.g., trucks and airplanes), and earthquakes, while the instability associated with dynamic sensors is caused essentially by movements of vehicles. Several vision systems have been installed in vehicles for assisting drivers in detecting, tracking and recognizing such objects as roads [8], [26], lane markings [5], [20], [22], [33]–[35], traffic signs [1], [17], [18], [29], road conditions (e.g., dry, wet, fog, freezing, and snow) [2], [35], and obstacles (e.g., pedestrians, vehicles, motorcycles, and other intruders) [6], [16], [22]. Their performances have been significantly impaired by the effect of image/video instability. In this paper, we present a technique to deal with this issue for a camcorder mounted on a moving vehicle.

There are typically three major stages constituting a video stabilization process, *motion estimation*, *motion taxonomy*, and *image compensation*. In the motion estimation stage, the camcorder motion is estimated from the input images. Since the estimated motion may contain both expected and unexpected motions, they are discriminated in the motion taxonomy stage. The image effect caused by unexpected camcorder motion is eliminated from the input image in the image compensation stage. The resulting image is a stabilized version of the input image. In the above process, motion estimation is ill-conditioned because the camcorder motion, which is three-dimensional (3-D), is to be estimated on the basis of input images that are two-dimensional (2-D). This stage actually dominates the entire stabilization process in both time complexity and accuracy. Many techniques, both hardware and software, have been proposed for improving the performance of motion estimation.

Hardware devices, including mechanical (e.g., gyros, accelerometers, and elastic dampers) [32] and electronic [15], have been developed for fast motion estimation. Mechanical equipment capable of dealing with large instabilities is typically not precise and are physically bulky and heavy. On the contrary, electronic devices are better in terms of precision, size and weight, but are limited in flexibility. In this paper, we prefer firmware approaches because of their flexibilities. However, the most crucial problem with the firmware approaches may be the

Manuscript received July 10, 2003; revised July 20, 2004.

Y.-M. Liang and H.-Y. M. Liao are with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan.

H.-R. Tyan is with the Institute of Information and Computer Engineering, Chung Yuan University, Chungli 32023, Taiwan.

S.-L. Chang and S.-W. Chen are with the Graduate Institute of Computer Science and Information Engineering, National Taiwan Normal University, Taipei 116, Taiwan (e-mail: schen@csie.ntnu.edu.tw).

Digital Object Identifier 10.1109/TVT.2004.836923

time complexity. We compensate for this drawback by utilizing *a priori* information of traffic scenes.

As mentioned, 3-D motion estimation from 2-D images is ill-conditioned. Additional information should be included in order to transform the problem into a well-conditioned one. Several motion models, serving as an important source of a priori information, have been introduced, including a 2-D rigid model with four parameters [11], [27], a 2-D affine model with six parameters [7], [9], [21], [36], a 2.5-D model with seven parameters [23], and a 3-D model with nine parameters. Intuitively, the higher dimension a model is used, the better accuracy is achieved. However, this may not be always the case because high dimensional models involve more complicated computations, which themselves may incur numerical instability.

There are two important tasks when using motion models: interframe motion calculation and model fitting. The interframe motions between successive images are calculated first. The calculated motions are next fitted to the preselected motion model, from which an overdetermined system of equations of motion parameters is obtained. By finding an optimal solution to this system, motion parameters are determined. The techniques for calculating interframe motions can be classified into two categories, differential and matching (or correlation) approaches. The differential approaches, primarily for computing optical flow [11], [14], [31] and based on the assumption of image brightness constancy, have been known to be sensitive to both high-order derivative errors and the aperture problem. Although recently there has been considerable progress in improving the derivative errors, the approaches are still annoyed with the aperture problem. Most previous researchers employed the matching approach to calculate interframe motions.

Matching techniques can be divided into two classes, block matching [30], [36] and feature-based matching [14], [28]. Typical block matching has been known to be time consuming. Many strategies have been proposed for improving the time complexity of the block matching method, such as multiresolution matching [23], gray-coded bit-plane matching [25], integral projection matching [36], Laplacian pyramid-based matching [21], and winner-update matching [10]. Although recently the time complexity of the block matching method has been greatly improved, the method is intrinsically weak at handling the motions involving rotation and scale change. On the other hand, feature-based matching can cope with motions involving translation, rotation, and scaling. However, the effectiveness of the feature-based method depends heavily on the features employed.

Features can be roughly grouped into two categories, local and global. Local features (e.g., points, edges, corners, faces, textures, and colors) generally possess a higher degree of tolerance to occlusion but lower tolerance to noise than global features. For interframe motion calculations, it is common that inappropriate local features (e.g., those extracted from moving objects) are extracted from images. Inadequate features could significantly degrade the accuracy of motion calculations. In this paper, the global features chosen are lane lines and road vanishing points. These features enable us to directly compute 3-D camcorder motions without precalculating 2-D interframe motions. Moreover, only a small number of features are adequate for calculating camcorder motions.

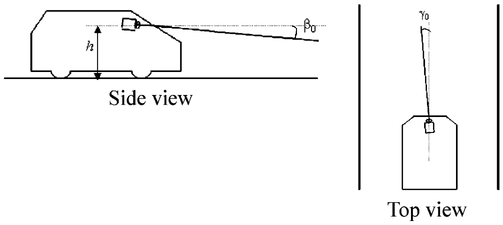


Fig. 1. Configuration of the mounted camcorder.

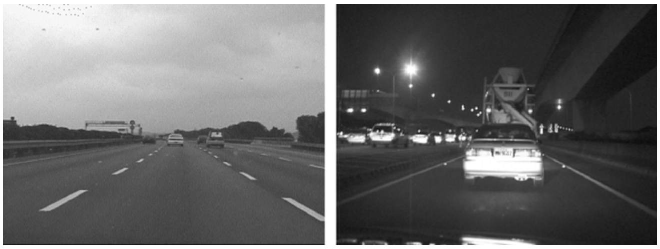


Fig. 2. Examples of scene images taken in daytime (left image) and at night (right image).

The rest of this paper is organized as follows. In Section II, we address the problem under consideration and give the process for solving it. The critical techniques for implementing the process are given in Sections III–VII. These include global feature extraction in Section III, imaging geometry in Section IV, camcorder motion estimation in Section V, motion taxonomy and smoothing in Section VI, and image compensation in Section VII. Experimental results are presented in Section VIII. Finally, Section IX presents our conclusions and suggests related future work.

II. Video Stabilization

In this section, the problem under consideration is described. The process for solving the problem is then presented. The details of implementation will be discussed later.

A. Problem Description

Recently, a driver-assistance system (DAS) for increasing driving safety for professional drivers (e.g., taxicab, bus, freight truck, and tanker drivers) as well as for reducing driving risk for the handicapped and elderly has been developed in our laboratory. Several subsystems, including a road sign recognition subsystem, a driving environmental change detection subsystem, and a subsystem to detect risky behaviors of nearby moving vehicles, have been developed for the DAS. There are still assorted subsystems for the DAS remaining to be developed, such as for detecting lanes, road markings, traffic signals, obstacles, and road conditions. Clearly, all the aforementioned subsystems need an in-vehicle vision system for acquiring and processing traffic images. Such a vision system should be able to cope with video instability originating from the vibrations of moving vehicles. Although video instability can be dealt with in separate subsystems, the redundancy of processing would degrade the performance of the entire system. It seems better to deal with the issue before video sequences are fed

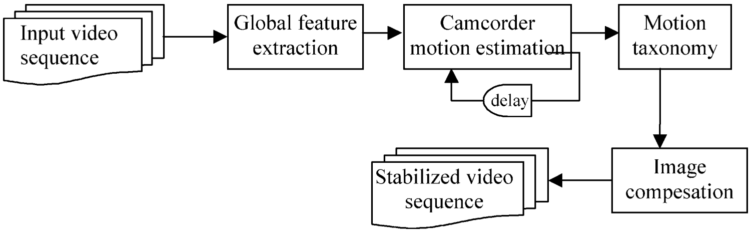

Input video sequence
Global feature extraction
Camcorder motion estimation
delay
Motion taxonomy
Image compesation
Stabilized video sequence

Fig. 3. Block diagram of the proposed video stabilization process.

into subsystems. However, different subsystems may demand different levels of video stability. How to develop a technique which could be suitable for all subsystems while maximizing the overall performance of the entire system, is one of the major challenges in this study.

Referring to Fig. 1, our video camcorder is mounted in the front of a vehicle that may be driving on a freeway or highway. Fig. 2 shows two examples of scene images acquired with the camcorder; the left image was taken in daytime and the right one was taken in nighttime. The proposed video stabilization process will be able to operate in both day and night. Several parameters regarding the configuration of the camcorder are given in Fig. 1, including the depression angle β_0, the viewing angle γ_0, and the height h of the camcorder. These parameters and the focal length f of the camcorder are known beforehand.

B. Stabilization Process

Fig. 3 depicts a block diagram for the proposed video stabilization process. There are four major stages constituting the process, global feature extraction, camcorder motion estimation, motion taxonomy, and image compensation. Consider an input video sequence. For each image of the sequence, we detect the features of lane lines and road vanishing point in the image. We refer to these features as global features because they characterize the road, which is the most important object in a traffic image. Also, the structure of the road can be completely reconstructed from the features. To detect lane lines, we first extract lane markings from the image based on the fact that lane markings typically correspond to the bright pixels in a traffic image. A traffic image is displayed in Fig. 4(a), and the lane markings extracted from the image are shown in Fig. 4(b). Clearly, noise affects the lane markings. A technique, combining connected component labeling and size filtering for eliminating noise is then applied to the image of Fig. 4(b). However, perfect noise removal is rarely achieved [see Fig. 4(c)]. Subsequent steps should be able to tolerate this imperfection. Next, we locate the middle points of the lane markings as shown in Fig. 4(d). Based on the located points, lane lines are determined by line fitting. Fig. 4(e) depicts the result of line fitting. As shown in this figure, outliers originating from noise have been overlooked. Finally, we define the road vanishing point simply as the intersection of the lane lines [see Fig. 4(f)]. This completes the first stage of feature extraction.

Having obtained the lane lines and the road vanishing point, we directly estimate 3-D camcorder motion parameters without precalculating 2-D interframe motions. This becomes feasible because only a subset of camcorder motion parameters of interest is estimated and the global features utilized to estimate the parameters provide enough information for the purpose. The global features used include features from both the current input image and the stabilized immediately preceding image. Hence, only two images are required at the motion estimation stage. This stage can therefore be performed on an image-by-image basis. Since no additional image is required at the other stages, the entire process can be conducted in the same manner. This characteristic is important to our DAS application because it greatly reduces the waiting times of the DAS subsystems for the stabilized version of any input image.

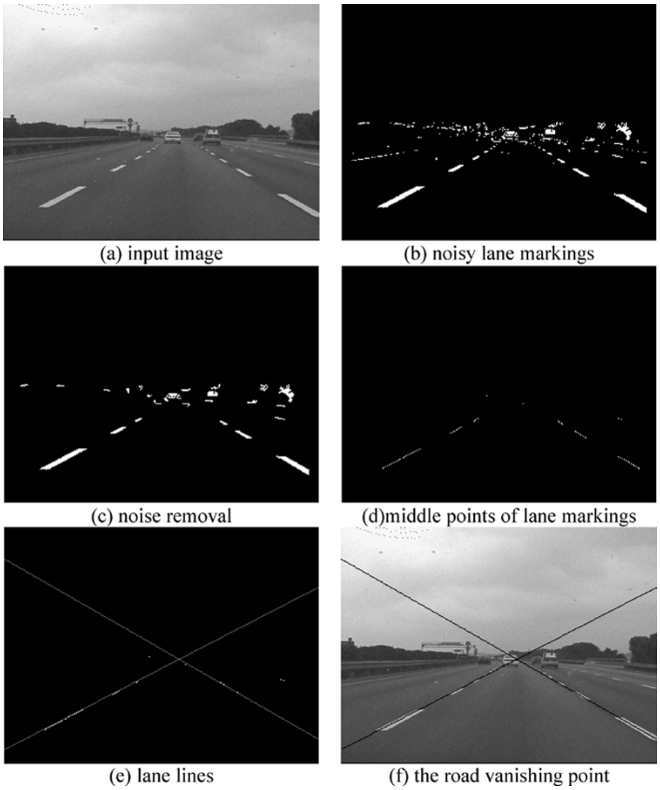

(a) input image
(b) noisy lane markings
(c) noise removal
(d)middle points of lane markings
(e) lane lines
(f) the road vanishing point

Fig. 4. Examples illustrating the stage of lane line detection and road vanishing point determination.

The camcorder motion parameters estimated in the previous stage include the translation distance in the x direction, T_x, and the rotation angles, α, β, γ, about the x, y, z axes, respectively. The camcorder motions corresponding to these parameters contain expected and unexpected motions. They are discriminated at the motion taxonomy stage. The expected motion component is further smoothed. The resultant motion is next integrated with a predicted motion, which is derived from the previous desired camcorder motions using a simplified Kalman filter. After this, the desired camcorder motion is obtained. The undesired motion

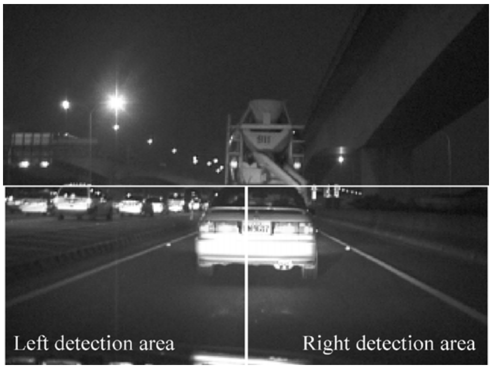


Fig. 5. The appearances of lane markings and their search space.

is then taken to be the difference between the desired and the estimated motions. At the final stage of image compensation, the image effect caused by the undesired camcorder motion is eliminated from the input image using an image transformation technique. The resulting image is what we call the stabilized version of the input image; it is then delivered to the DAS subsystems. After completing the current image, the process continues with the next input image.

There are several features that characterize the proposed process. First, the global features, lane markings, and road vanishing points, are utilized to estimate motion parameters. Second, the global features provide enough information for directly estimating 3-D motion parameters without first calculating 2-D interframe motions. Third, the process can be performed on an image-by-image basis instead of on a batch basis. Finally, the process can work in both day and night.

III. Global Feature Extraction

This section addresses the feature extraction of lane markings and the road vanishing point from an input image. We first consider the characteristics of real world lane markings: (a) lane markings are painted on roads; (b) lane markings can be white, yellow or red; (c) lane markings are shaped like ribbons; (d) lane markings may be continuous or segmented; (e) lane markings are aligned or parallel to each other. These characteristics together with the known configuration of our camcorder (see Fig. 1) provide useful information for extracting lane markings from images. Referring to Fig. 5, lane markings present in an image correspond to bright pixels and, as a whole, look like strips appearing in the lower portion of the image with an orientation of about 45 degrees.

According to the above observations, we develop an approach for extracting lane markings from images. To illustrate this method, we refer to the example shown in Fig. 4. Let I denote the input image and $i(x,y)$ represent the gray value of the pixel located at (x,y). Define the horizontal right and left gradients for pixel (x,y) as

$$\begin{aligned} d_{+m}(x,y) &= i(x,y) - i(x+m,y) \\ d_{-m}(x,y) &= i(x,y) - i(x-m,y) \end{aligned} \tag{1}$$

where m specifies the maximum horizontal width of lane markings present in an image. Based on the above definitions, we construct an image, denoted by R, whose entries $r(x,y)$ are computed as follows [see (2) at the bottom of the page]. Equation (2) picks an image pixel whose gray level is higher than both of its two horizontal neighbors located at distances $-m$ and m. Such a pixel is likely to be located on a lane marking. We retain the combined horizontal gradient, i.e., $d_{+m} + d_{-m}$, of the pixel in image R. We then filter out from R the pixels with weak accumulated gradients, where the average of the gradients of all the picked pixels serves as the threshold t. The surviving pixels in R are next set to 1 in another image B

$$b(x,y) = \begin{cases} 1 & \text{if } r(x,y) > t \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

The above process is only applied to the lower half of the input image due to the aforementioned observation that lane markings appear in the lower portion of an image. We hence define the lower half of an image as the search space of lane markings, denoted by $\mathbf{S}$.

The result of applying the above process to the input image of Fig. 4(a) is shown in Fig. 4(b). As we can see in this result, there is significant noise accompanying the lane markings. In order to get rid of the noise, we apply a connected component labeling procedure, followed by a size filtering procedure to image B. Size filtering eliminates small components based on a threshold determined by the average number of pixels in all the connected components. Since noise removal is rarely perfect, our subsequent steps will tolerate this imperfection. See Fig. 4(c) where the result of noise removal for the image of Fig. 4(b) is depicted.

Next, we want to locate the middle points of lane markings in the resultant image. We do this by examining the image row by row from the bottom within the search space $\mathbf{S}$. In each row, we look for bright segments whose lengths are less than m defined earlier. We next mark the central points of the segments, if found, as the candidate middle points of lane markings. Actually, several strategies have been introduced to improve the performance of this procedure. For instance, the points marked in the previous row are used to guide the search in the present row. A row has to be fully examined only if fewer than two marked points are provided by the previous row, or fewer than two segments are found in the present row around the points marked in the previous row. In this way, many noisy components are ignored. Very short segments are simply ignored. We do not attempt to connect adjacent short segments, which may result from broken lane markings, into a longer one. Although

$$r(x,y) = \begin{cases} d_{+m}(x,y) + d_{-m}(x,y), & \text{if } (d_{+m}(x,y) > 0) \text{ and } (d_{-m}(x,y) > 0) \\ 0, & \text{otherwise} \end{cases}. \tag{2}$$

we may lose some segments of lane markings, this usually is not a problem because only a few points are enough for line fitting. Fig. 4(d) shows the candidate middle points of lane markings located in the image of Fig. 4(c).

Thereafter, we want to find lines which best fit the candidate middle points located in the previous step. The method of least squares is commonly employed for finding linear approximations to data. However, this method is known to be sensitive to outliers. In this paper, a technique based on the idea of median of intercepts [4] is developed. Let the equation of a line be

$$y = ax + b. \tag{4}$$

Suppose that this line passes through two distinct points (x_1, y_1) and (x_2, y_2). The coefficients of the equation are

$$\begin{cases} a = \frac{(y_2 - y_1)}{(x_2 - x_1)} \\ b = \frac{(x_2 y_1 - x_1 y_2)}{(x_2 - x_1)} \end{cases}. \tag{5}$$

Let $P = \{p_i, i = 1, \ldots, n\}$ be the set of candidate middle points extracted in the previous step. For each pair of points (p_s, p_t) in P, we can compute two coefficients (a_l, b_l) according to (5). These two coefficients can be represented as a point in a 2-D space, called the parameter space, with its two axes corresponding to the two coefficients (a, b) of the line equation.

In a continuous parameter space, point pairs selected from a straight line should give identical coefficients of the line equation or equivalently correspond to the same point in parameter space. However, this is usually not the case for a line in the discrete image space; generally, a cluster of points, rather than a single point, is formed in the parameter space. Therefore, during line fitting we keep searching for clusters in the parameter space. If a cluster containing points, $\{(a_l, b_l), l = 1, \ldots, m\}$ where m is larger than a prescribed number, is found, the median point (a_M, b_M) of the cluster is determined immediately by

$$\begin{cases} a_M = \text{median}\{a_i\}, & i = 1, \ldots, m \\ b_M = \text{median}\{b_j\}, & j = 1, \ldots, m \end{cases}. \tag{6}$$

A virtual line is then generated in the image space according to (a_M, b_M). Along this line, we count and then delete the image points near the line. The number of deleted image points and the median point are then recorded somewhere and the cluster is removed from the parameter space. The above process is repeated until no point remains in the image space. The median points associated with the two largest numbers of deleted image points are then chosen to construct two lines in the image space. Typically, these two lines delineate the lane in which our vehicle is driving. Fig. 4(e) illustrates where the two constructed lane lines are drawn. Finally, the road vanishing point is simply determined by intersecting the two lane lines.

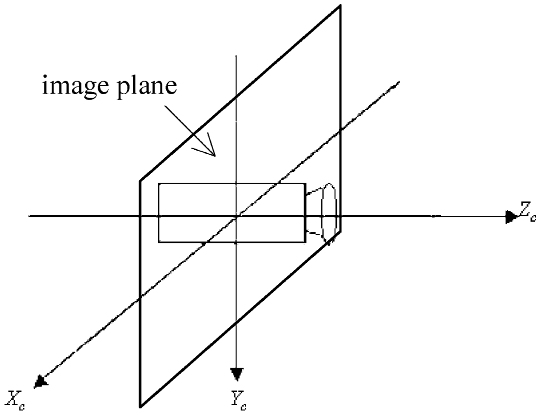


Fig. 6. The camcorder coordinate system.

IV. Imaging Geometry

In this section, a motion model for the camcorder mounted on a moving vehicle is presented. The imaging of the lane markings and the vanishing point of a road characterized by the model is then addressed. The parameters embedded in the camcorder motion model will be estimated in the next section based on the formulations provided in this section.

A. Camcorder Motion Model

We establish a coordinate system for the camcorder as shown in Fig. 6. The camcorder coordinate system (X_c, Y_c, Z_c) has the image plane coincident with the X_c-Y_c plane and the viewing direction along the Z_c-axis. So, the center of the image plane is at the origin of the coordinate system.

The camcorder moving in space performs a series of 3-D rigid motions, which can be translations and/or rotations. In terms of homogeneous coordinates, translation of the camcorder from a location p_1 to a new location p_2 can be performed using the matrix multiplier

$$Trans(T_x, T_y, T_z) = \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & -T_y \\ 0 & 0 & 1 & -T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

Rotation of the camcorder about any axis D by an angle θ can be performed using the matrix multiplier, as shown in (8) at the bottom of the page. The above transformation rotates the coordinates by an angle α about the Z_c axis, following by a rotation of β about the new coordinate $X_c{}'$ axis, and then rotating through an angle γ about the new Y_c'' axis, namely

$$\begin{aligned} Rot(D, \theta) &= Rot(\alpha, \beta, \gamma) \\ &= Rot(Y_c'', \gamma) \cdot Rot(X_c', \beta) \cdot Rot(Z_c \alpha). \end{aligned}$$

$$Rot(\alpha, \beta, \gamma) = \begin{bmatrix} \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\beta\sin\gamma & 0 \\ \sin\alpha\cos\beta & \cos\alpha\cos\beta & -\sin\beta & 0 \\ \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\beta\cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{8}$$

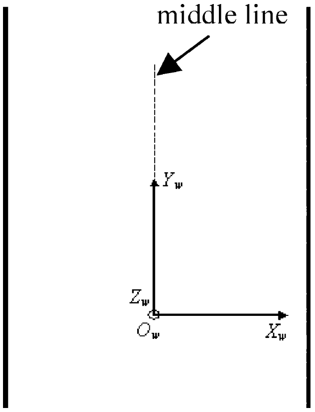
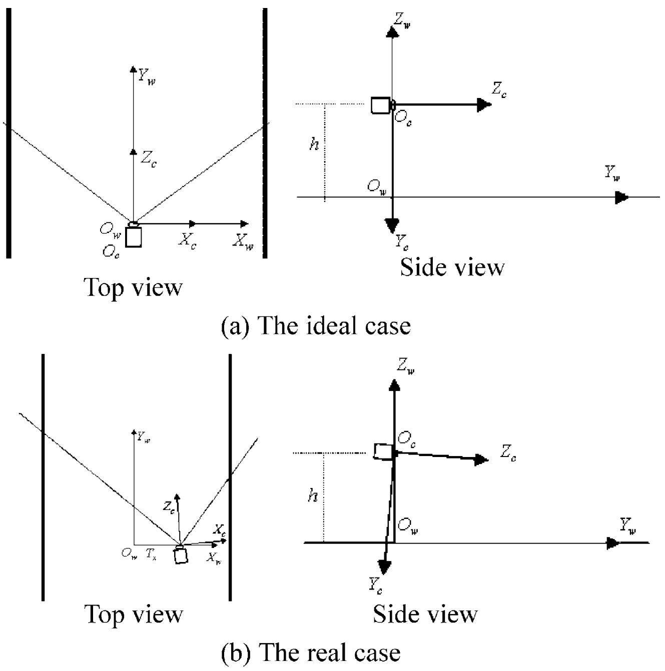


Fig. 7. The real world coordinate system.



Top view Side view

(a) The ideal case

Top view Side view

(b) The real case

Fig. 8. Relationships between the camcorder and real world coordinate systems.

Next, consider the real world coordinate system. Refer to Fig. 7, which shows the real world coordinate system at a certain time instant. The origin O_w of the coordinate system (X_w, Y_w, Z_w) is located at a point on the middle line of the lane, in which the vehicle is driving. The Y_w-axis of the coordinate system is tangent to the middle line and points forward, the X_w-axis perpendicular to the Y_w-axis points to the right, and the Z_w-axis points vertically upward. Rather than fixed, the real world coordinate system moves along the middle line of the lane at the same speed as the vehicle.

See Fig. 8; there are two states, ideal and real, for the relationship between the camcorder and the real world coordinate systems. In the ideal state [Fig. 8(a)], the origin O_c of the camcorder coordinate system is situated at a distance h directly above the origin O_w of the real world coordinate system, where h approximates the height of a vehicle. The X_c-Z_c plane of the camcorder coordinate system is parallel to the X_w-Y_w plane of the real world coordinate system. The transformation describing the above relationship between the two coordinate systems is

$$T_i = \mathrm{Trans}(0,0,h) \cdot Rot(0, 90^\circ, 0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{9}$$

In the real case [Fig. 8(b)], in addition to the height (i.e., h) different between the origins of the camcorder and the real world coordinate systems, the camcorder may have a rotation $Rot(\alpha, \beta, \gamma)$ with respect to the camcorder coordinate system and a translation T_x in the X_w direction of the real world coordinate system. The transformation specifying such a relationship between the camcorder and the real world coordinate systems is shown in (10) at the bottom of the page.

The scene images acquired by the camcorder in the ideal case will serve as reference images for those taken in the real case. The objective of image stabilization is to transform the real images into reference ones.

B. Imaging of Lane Markings

According to the definition of the real world coordinate system, the road surface is coincident with the X_w-Y_w plane of the real world coordinate system. Lane markings painted on the road surface all have $z_w = 0$. Let p be a point located on a lane marking and have real world coordinates $(x_w, y_w, 0)$. In the ideal case [Fig. 8(a)], the real world coordinates of the point can be transformed into camcorder coordinates (x_c, y_c, z_c) by the transformation of (9)

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & h \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} x_w \\ h \\ y_w \\ 1 \end{bmatrix}. \tag{11}$$

Next, the image coordinates (x_i, y_i) of the point can easily be found by perspective projection

$$\begin{cases} x_i = f\frac{x_w}{y_w} \\ y_i = f\frac{h}{y_w} \end{cases} \tag{12}$$

where f is the focal length of the camcorder.

$$\begin{aligned} T_a &= \mathrm{Trans}(T_x, 0, 0) \cdot Rot(\alpha, \beta, \gamma) \\ &= \begin{bmatrix} 1 & 0 & 0 & -T_x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\beta\sin\gamma & 0 \\ \sin\alpha\cos\beta & \cos\alpha\cos\beta & -\sin\beta & 0 \\ \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma & \cos\beta\cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \end{aligned} \tag{10}$$

Considering the camcorder having been translated by T_x in the X_w direction, the image coordinates (x_i', y_i') of point p becomes

$$\begin{cases} x_i{}' = f\frac{x_w - T_x}{y_w} = x_i - \frac{T_x}{h} y_i \\ y_i{}' = f\frac{h}{y_w} = y_i \end{cases}. \tag{13}$$

Furthermore, if the camcorder has been rotated by angles α, β, γ about the X_c', Y_c', Z_c' axes, the point will be projected onto the image plane at $(x_i''$, $y_i'')$, where [see (14) at the bottom of the page].

C. Imaging of the Road Vanishing Point

Parallel lines in the real world can be imagined to meet at infinity. The projection of this meeting point onto an image plane is called the vanishing point [3] of the parallel lines. Suppose that the image coordinate of a vanishing point is (x_v, y_v). Since the vanishing point is the projection of an infinitely far 3-D point, the y_w coordinate of the point is far larger than its x_w coordinate (i.e., $y_w \gg x_w$) and h (i.e., $y_w \gg h$). Considering the ideal case [Fig. 8(a)], from (12) the coordinate (x_v, y_v) of the vanishing point can be obtained

$$\begin{cases} x_v \underset{y_w \gg x_w}{=} f\frac{x_w}{y_w} \approx 0 \\ y_v \underset{y_w \gg h}{=} f\frac{h}{y_w} \approx 0 \end{cases}. \tag{15}$$

In other words, the vanishing point is close to the center of the image plane.

If the camcorder is further translated by T_x along the X_c axis, the vanishing point (x_v', y_v') can be obtained from (13) and (15)

$$\begin{cases} x_v' = x_v - \frac{T_x}{h} y_v \approx 0, \\ y_v' = y_v \approx 0 \end{cases}. \tag{16}$$

From this result, we know that the coordinates of the vanishing point are not affected by camcorder translations. However, if the camcorder is rotated by angles α, β, γ about the X_c, Y_c, Z_c axes, respectively, the vanishing point (x_v'', y_v'') can be obtained according to (14) and (15) [see (17) at the bottom of the page]. The above result tells us that the coordinates of the vanishing point can only be affected by the rotation parameters β and γ, but not α.

V. Camcorder Motion Estimation

In this section, camcorder motion parameters are estimated. The input image is assumed to have been taken by the camcorder when it is affected by a transformation that is controlled by the camcorder motion parameters. In the most general case, there are six parameters involved in a rigid transformation, three for translation (T_x, T_y, T_z), and three for rotation (α, β, γ). In this study, the three rotation parameters are the main concern. We neglect the translation parameters during image compensation because of the following reason. Note that camcorder translation (T_x, T_y, T_z) is measured with respect to the real world coordinate system, which moves with the vehicle. Both the camcorder translation and the vehicle motion induce effects on the images. However, the camcorder translation is much smaller than the vehicle movement. The effect on images caused by the camcorder translation is much less significant than that induced by the vehicle movement. Although we do not consider the camcorder translation during image compensation, the translation component T_x is required for the estimation of the rotation parameters. Therefore, four parameters, T_x, α, β, and γ, need to be estimated.

Provided that the lane markings and the associated road vanishing point have been extracted from the input image, to estimate the camcorder motion parameters, we first calculate the rotation parameters β and γ based on the road vanishing point. The calculated parameters are then used to compensate for their effects on the lane markings. The compensated lane markings are next used to calculate parameters T_x and α.

A. Motion Parameters β and γ

Recall that (17) relates the image coordinates of the road vanishing point (x_v, y_v) and the camcorder rotation parameters β and γ; i.e.

$$\begin{cases} x_v = f \tan\gamma \\ y_v = -f\frac{\tan\beta}{\cos\gamma} \end{cases}. \tag{18}$$

$$\begin{cases} x_i'' = f\frac{x_w(\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma) + h(\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma) + y_w\cos\beta\sin\gamma}{x_w(\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma) + h(\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) + y_w\cos\beta\cos\gamma} \\ \quad = f\frac{x_i(\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma) + y_i(\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma) + f\cos\beta\sin\gamma}{x_i(\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma) + y_i(\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) + f\cos\beta\cos\gamma} \\ y_i'' = f\frac{x_w\sin\alpha\cos\beta + h\cos\alpha\cos\beta + y_w(-\sin\beta)}{x_w(\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma) + h(\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) + y_w\cos\beta\cos\gamma} \\ \quad = f\frac{x_i\sin\alpha\cos\beta + y_i\cos\alpha\cos\beta + f(-\sin\beta)}{x_i(\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma) + y_i(\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) + f\cos\beta\cos\gamma} \end{cases}. \tag{14}$$

$$\begin{cases} x_v'' = f\frac{x_v(\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma) + y_v(\cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma) + f\cos\beta\sin\gamma}{x_v(\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma) + y_v(\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) + f\cos\beta\cos\gamma} \\ \approx f\tan\gamma \\ y_v'' = f\frac{x_v\sin\alpha\cos\beta + y_v\cos\alpha\cos\beta + f(-\sin\beta)}{x_v(\sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma) + y_v(\cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma) + f\cos\beta\cos\gamma} \\ \approx -f\frac{\tan\beta}{\cos\gamma} \end{cases}. \tag{17}$$

Rearranging these equations, we obtain

$$\begin{cases} \gamma = \tan^{-1}\left(\frac{x_v}{f}\right) \\ \beta = -\tan^{-1}\left(\frac{y_v \cos\gamma}{f}\right) \end{cases}. \tag{19}$$

Since x_v, y_v and f are known, β and γ can be calculated by (19).

Next, we want to compensate the lane markings for the effects caused by β and γ. To do so, we rotate the camcorder about its Y_c-axis by angle $-\gamma$ and then rotate about the new X_c'-axis by angle $-\beta$. The associated transformation is

$$\begin{aligned} R(-\gamma,-\beta) &= Rot(X_c',-\beta)\cdot Rot(Y_c,-\gamma) \\ &= \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ \sin\beta\sin\gamma & \cos\beta & \sin\beta\cos\gamma \\ \cos\beta\sin\gamma & -\sin\beta & \cos\beta\cos\gamma \end{bmatrix}. \end{aligned} \tag{20}$$

Let (x_c', y_c', z_c') be the new coordinates after transforming the camcorder coordinates (x_c, y_c, z_c) of a point by rotation $R(-\gamma,-\beta)$. Then

$$\begin{bmatrix} x_c' \\ y_c' \\ z_c' \end{bmatrix} = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ \sin\beta\sin\gamma & \cos\beta & \sin\beta\cos\gamma \\ \cos\beta\sin\gamma & -\sin\beta & \cos\beta\cos\gamma \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}. \tag{21}$$

Suppose that (x_i, y_i) and (x_i', y_i') are the image coordinates produced by the perspective projection of (x_c, y_c, z_c) and (x_c', y_c', z_c') onto the image plane, respectively. From (12) and (21), (x_i, y_i) and (x_i', y_i') are related by

$$\begin{cases} x_i' = f\frac{x_i\cos\gamma - f\sin\gamma}{x_i\cos\beta\sin\gamma - y_i\sin\beta + f\cos\beta\cos\gamma} \\ y_i' = f\frac{x_i\sin\beta\sin\gamma + y_i\cos\beta + f\sin\beta\cos\gamma}{x_i\cos\beta\sin\gamma - y_i\sin\beta + f\cos\beta\cos\gamma} \end{cases}. \tag{22}$$

Therefore, given an image point (x_i, y_i) on a lane marking, (22) will give the new coordinates (x_i', y_i') of the point after compensating rotation. Note also that the road vanishing point (x_v, y_v) will become (0, 0) after compensating rotation [see Section IV-C].

B. Motion Parameters T_x and α

The compensated lane markings are then used to calculate parameters T_x and α. Let (x_i, y_i) be a point on a lane marking. Suppose that the camcorder is translated by T_x along the X_w-axis of the real world coordinate system, followed by a rotation of an angle α about the new Z_c'-axis of the camcorder coordinate system. The new coordinates (x_i', y_i') of the point can be calculated according to (13) and (14)

$$\begin{cases} x_i' = (x_i - dy_i)\cos\alpha - y_i\sin\alpha \\ y_i' = (x_i - dy_i)\sin\alpha + y_i\cos\alpha \end{cases} \tag{23}$$

in which $d = T_x/h$. To know T_x, we need to calculate d first, which is determined in the following.

Let s and s' be the slopes of a lane marking before and after the above transformation of the camcorder, respectively. Since the coordinates of the road vanishing point are (0, 0) both before and after the transformation, s and s' can be calculated by

$$\begin{cases} s = \frac{0-y_i}{0-x_i} = \frac{y_i}{x_i} \\ s' = \frac{0-y_i'}{0-x_i'} = \frac{y_i{}'}{x_i{}'} \end{cases}. \tag{24}$$

From (23) and (24), we get

$$s' = \frac{y_i'}{x_i'} = \frac{(x_i - dy_i)\sin\alpha + y_i\cos\alpha}{(x_i - dy_i)\cos\alpha - y_i\sin\alpha} = \frac{(x_i - dy_i)\tan\alpha + y_i}{(x_i - dy_i) - y_i\tan\alpha}. \tag{25}$$

And, from (24) and (25) we obtain

$$\tan\alpha = \frac{s'(1-ds) - s}{(1-ds) + s's}. \tag{26}$$

Let l_{t-1} and r_{t-1} be the slopes of the two lane markings in the immediately preceding image after stabilization and l_t' and r_t' be the slopes of the two lane markings in the current image after compensating rotation. From (26), we have

$$\begin{cases} \tan\alpha = \frac{l_t'(1-dl_{t-1}) - l_{t-1}}{(1-dl_{t-1}) + l_t' l_{t-1}} \\ \tan\alpha = \frac{r_t'(1-dr_{t-1}) - r_{t-1}}{(1-dr_{t-1}) + r_t' r_{t-1}} \end{cases}. \tag{27}$$

By equating the two equations in (27), we obtain a quadratic equation in d

$$pd^2 + qd + r = 0 \tag{28}$$

where

$$\begin{cases} p = (l_t' - r_t')l_{t-1}r_{t-1}, \\ q = -(l_t' - r_t')(l_{t-1} + r_{t-1}) \\ r = (l_t' - r_t')(1 + l_{t-1}r_{t-1}) + (r_{t-1} - l_{t-1})(1 + l_t' r_t') \end{cases}$$

.

Solving (28) for d gives two solutions. We choose the one closer to that in the immediately preceding image.

Having determined d, we can calculate T_x by

$$T_x = dh. \tag{29}$$

Next, substituting d into (27), we get

$$\alpha = \tan^{-1}\left(\frac{l_t'(1 - dl_{t-1}) - l_{t-1}}{1 - dl_{t-1} + l_t' l_{t-1}}\right). \tag{30}$$

We still need to compensate the lane markings for the effects of T_x and α because the slopes of the compensated lane markings will be used in the stabilization of the subsequent input image. To compensate the lane markings , we first rotate the camcorder about the Z_c-axis by angle $-\alpha$

$$Rot(Z_c, -\alpha) = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{31}$$

Using this transformation, the new coordinates (x'_c, y'_c, z'_c) of a point with camcorder coordinates (x_c, y_c, z_c) can be calculated by

$$\begin{bmatrix} x'_c \\ y'_c \\ z'_c \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}. \tag{32}$$

Let (x_i, y_i) and (x'_i, y'_i) be the image coordinates produced by the perspective projections of (x_c, y_c, z_c) and (x'_c, y'_c, z'_c) onto the image plane, respectively. From (32) we obtain the relationship between (x_i, y_i) and (x'_i, y'_i)

$$\begin{cases} x'_i = x_i \cos\alpha + y_i \sin\alpha \\ y'_i = -x_i \sin\alpha + y_i \cos\alpha \end{cases}. \tag{33}$$

Therefore, given a point (x_i, y_i) on a lane marking, (33) will give the new coordinates (x'_i, y'_i) of the point after compensating the camcorder rotation by α.

Next, we translate the camcorder by T_x along the new X'_c axis. The new coordinates $(x''_i y''_i)$ of the point can be calculated using (13)

$$\begin{cases} x''_i = x'_i - \frac{T_x}{h} y'_i \\ y''_i = y'_i \end{cases}. \tag{34}$$

After compensating the lane markings for the effects caused by T_x and α, we then calculate the slopes of the lane markings. Let (x'', y'') be a point on a lane marking. Since the coordinates of the road vanishing point are (0, 0), the slope of the lane marking is $s'' = (0 - y'')/(0 - x'') = y''/x''$.

VI. Motion Taxonomy and Smoothing

The estimated camcorder motion characterized by parameters T_x, α, β, and γ, contains both expected and unexpected motions. The objective of motion taxonomy is to discriminate between expected and unexpected camcorder motions. Note that the expected motions estimated for successive images are generally inconsistent with each other because of noise, computational error and imperfect feature detection. Therefore, the expected motions computed at sequential stages are further smoothed.

Rather than deal with motion taxonomy and motion smoothing separately, we do them in one step. The method is motivated by the idea of the Kalman filter. In brief, the Kalman filter determines the state of a system based on both the state predicted from the previous states and the state derived from the current observation. The differences between the predicted and observed states computed at different stages are accumulated in terms of a state covariance matrix and a measurement covariance matrix for predicting system states at later stages. A characteristic of the Kalman filter during its integration of the predicted and the observed states is that the larger the difference between the predicted and observed states, the more the filter tends toward the predicted state because the observed state may result from a noisy or erroneous measurement. As a result, the Kalman filter smoothes out the system states determined at sequential stages.

Consider a motion parameter x ($x = T_x$, α, β or γ). Let x_t be the parameter estimated for the current image and x'_{t-2} and x'_{t-1} be the smoothed parameters determined for the previous two images. We first predict the current smoothed parameter x'_{pt} based on x'_{t-2} and x'_{t-1} simply by extrapolation, i.e., $x'_{pt} = 2x'_{t-1} - x'_{t-2}$. For initial conditions, let $x'_0 = x_0$ and $x'_1 = x_1$. For subsequent images, we determine the current smoothed parameter x'_t based on the predicted smoothed parameter x'_{pt} and the current estimated parameter x_t by using a convex linear combination, i.e., $x'_t = (1-k)x'_{pt} + kx_t$, where $k = e^{-(|x_t - x'_{pt}|)1/2}$. By this definition of k, the larger the difference $|x_t - x'_{pt}|$, the smaller the value of k and, in turn, the closer x'_t is to x'_{pt}. This equation deals with the issues of motion taxonomy and motion smoothing in one step. We refer to the smoothed expected motion parameters as the desired motion parameters.

VII. Image Compensation

While we have computed four desired camcorder motion parameters T_x, α, β, and γ, only the rotation parameters α, β, γ are of interest at the image compensation stage. As mentioned before, the image effect caused by T_x is much smaller than that caused by vehicle movement. So, we ignore the translation parameter T_x here.

During image compensation, an image transformation process involving rotation is applied to the input image, thereby transforming the image into its stabilized version. Imagine that the input image is obtained by rotating the camcorder about the X_c, Y_c and Z_c axes by angles α, β and γ, respectively, and that the stabilized image is obtained by rotating the camcorder about the same axes by angles α', β' and γ'. Let C and C' denote the camcorder coordinate system before and after rotation. Consider a 3-D world point p with the coordinates (x_c, y_c, z_c) and (x'_c, y'_c, z'_c) in C and C', respectively. Then, (x_c, y_c, z_c) and (x'_c, y'_c, z'_c) are related by

$$\begin{bmatrix} x'_c \\ y'_c \\ z'_c \end{bmatrix} = R_1 R_2 \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{35}$$

where $\mathrm{R}_1 = Rot(Y''_C, \gamma) \cdot Rot(X'_{C'}, \beta) \cdot (Z_{C'}, \alpha)$ and $\mathrm{R}_2 = Rot(Z''_C, -\alpha') \cdot Rot(X'_C, -\beta') \cdot (Y_C, -\gamma')$.

Let (x_i, y_i) and (x'_i, y'_i) be the image coordinates produced by the perspective projections of (x_c, y_c, z_c) and (x'_c, y'_c, z'_c) onto the image plane in the respective spaces C and C'. Then, the relationship between (x_i, y_i) and (x'_i, y'_i) can easily be computed from (35). During image transformation, a pixel in the output image (x'_i, y'_i) is transformed back into the input image location (x_i, y_i), and then the pixel in the output image is set to the value of the corresponding pixel in the input image.

VIII. Experimental Results

A series of experiments were carried out to examine the effectiveness of the proposed stabilization approach. Both synthetic and real data were utilized in the experiments. Real data was collected using a camcorder mounted on a moving vehicle. Since there is no simple and inexpensive way to reliably measure the camcorder motions that are necessary for evaluating the

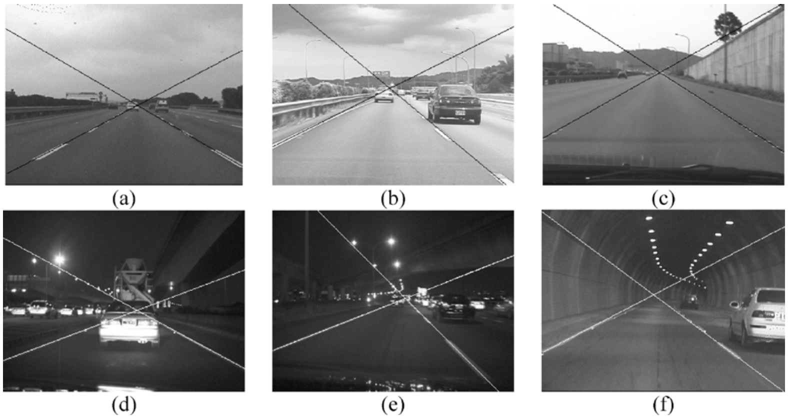


Fig. 9. Images are chosen from distinct video sequences acquired at different times of day and in different road segments. The lane lines extracted from the images are shown as well.

accuracies of the calculated camcorder motions, we turn to simulated data that were generated using software. For generating simulated video sequences, camcorder motions were provided beforehand so that we could later perform accuracy analysis for the calculated motions.

In the experiments with real data, a number of video sequences in color were collected. The frame size is 320 by 240 pixels. Fig. 9 shows six images chosen from six different representative video sequences in our database which were taken at different times of day on different road segments. For each video sequence we first applied two algorithms, lane marking detection and line fitting, to each image of the sequence. Refer to Fig. 9 in which the lane lines extracted from the images are shown. Note that the images in the figure include those acquired in both daytime and nighttime. The results demonstrate that our algorithms can perform well in a wide range of lighting conditions. This is important because the following stabilization process relies on the outcomes of these algorithms. This means that our stabilization process can operate in wide ranges of both time and lighting conditions.

Having extracted lane lines from the current image, the road vanishing point is determined simply by finding the point at which the lines would intersect. The lane lines and the road vanishing point from the current image are combined with those from the previous image to compute the camcorder motion corresponding to the current image. Then, the expected camcorder motion is separated from the computed motion and further smoothed. The resulting motion is referred to as the estimated camcorder motion corresponding to the current image. The estimated motion will later be integrated with the predicted one leading to the final desired camcorder motion. Accordingly, the input image is to be stabilized. The predicted camcorder motion is extrapolated from the desired motions corresponding to the preceding images. During image stabilization, the current image is transformed based on the information provided by the desired camcorder motion. Several experimental results of video stabilization for the video sequences are available in our database available on our web site http://www.iis.sinica.edu.tw/~ulin/. Currently, our stabilization program runs on a Pentium IV 2.8-GHz PC at a rate of about 20 images per second.

TABLE I
ESTIMATED ERRORS OF β AND γ WHEN T_x AND α ARE FIXED

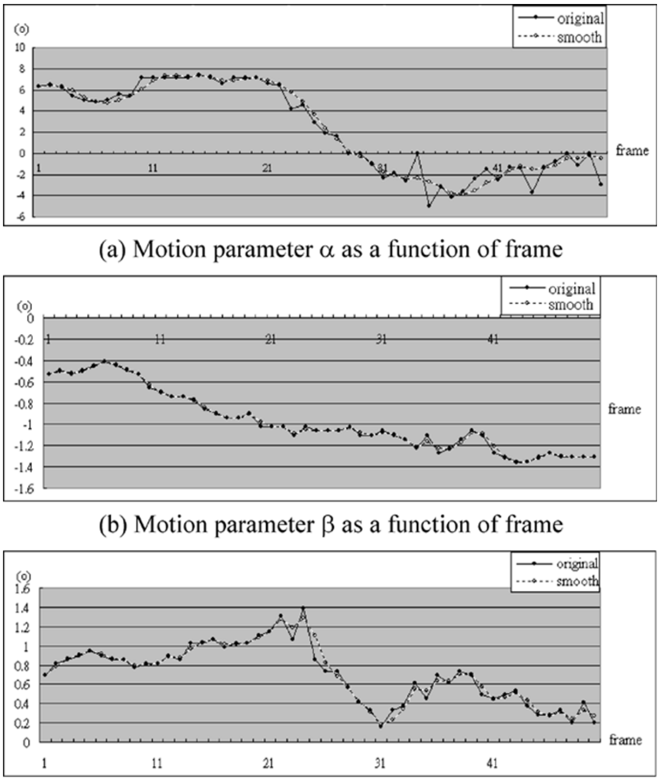
T_x=0 , α=0° , β= -10° ~10° , γ= -10° ~10°			
Average error in β	0.019712°	Average error in γ	0.022273°
Maximum error in β	0.040042°	Maximum error in γ	0.035346°

For the simulated video sequence, we assumed that there was a lane with a width of three meters in 3-D space and that a camcorder with the focal length of 2.8 mm was mounted on a vehicle at the height above the ground of one meter. Before generating a video sequence, we predetermined a series of camcorder motions. Each camcorder motion was composed of two components, one corresponding to vehicle motion and one corresponding to the relative motion between the camcorder and the vehicle. The former was simulated by y_w value and the latter by a set of T_x, α, β and γ values. While a camcorder motion is easily defined by assigning values to the motion parameters, a consistent series of motions should be carefully selected. Empirically, we fixed the values of certain motion parameters and gradually altered the values of the remaining parameters within some prescribed intervals. For example, we fixed T_x and α and kept increasing y_w. Then β and γ were charged each time by one degree from -10 to 10 degrees. In total, a series of 441 camcorder motions was obtained. In another example, we fixed β and γ and kept increasing y_w. Then each time T_x was varied by 0.1 m from -1 to 1 m and α by one degree from -10 to 10 degrees. Again, a series of 441 camcorder motions was generated.

Given a series of predetermined camcorder motions, we computed a series of camcorder poses. For each pose, we project the presumed lane onto the image plane to obtain an image of the lane. Repeating the above process for the camcorder poses in the series, a video sequence of the lane was obtained. We then computed the camcorder motions corresponding to the individual images from the generated video sequence. Table I gives the computational errors of β and γ when T_x and α are fixed. The average and maximum errors of β and γ were all under 0.05 degrees. Table II shows the computational errors of T_x and α when β and γ were fixed. In this case, the average error and maximum error of T_x were both under 0.0004 m and those of α were below 0.006 degrees. Other experimental results (not

TABLE II
ESTIMATED ERRORS OF T_x AND α WHEN β AND γ ARE FIXED

$\beta=0^o$, $\gamma=0^o$, T_x= -1~1 , α= -10° ~10°			
Average error in T_x (m)	0.0000015	Average error in α	0.0000284 °
Maximum error in T_x (m)	0.000317	Maximum error in α	0.005591 °



(a) Motion parameter α as a function of frame

(b) Motion parameter β as a function of frame

(c) Motion parameter γ as a function of frame

Fig. 10. Curves of α, β, and γ before and after applying stabilization.

shown) have also given similar outcomes. For our application, the above errors can be tolerable for our DAS subsystems.

Finally, we applied the proposed video stabilization process to the generated video sequences and recomputed the camcorder motions corresponding to the individual images based on the stabilized video sequences. Fig. 10 shows the curves of α, β, and γ before and after applying stabilization. Clearly, the curves after stabilization are much smoother than those before stabilization.

IX. Concluding Remarks and Future Work

A video stabilization technique for a camcorder mounted on a moving vehicle was presented. The technique possesses a number of intriguing features. First, it directly computes 3-D camcorder motions without precalculating 2-D interframe motions. Second, it computes camcorder motions based on only a few global features that can be reliably detected. Third, it can operate in both daytime and nighttime. Fourth, it can be performed on an image-by-image basis instead of a batch basis. The first two characteristics make our stabilization process very effective in computation, and hence applicable for real time applications. The third feature is critical for in-vehicle vision systems because they are usually required to be able to work during daytime and nighttime. The fourth characteristic is important to our driver assistance system (DAS) since it greatly reduces the waiting times of the DAS subsystems for the stabilized version of any input image.

Since the proposed technique heavily relies on lane markings, the technique would not work on a roadway without lane markings. Besides, if lane markings of a road are occluded due to heavy traffic or are covered by mud, water or snow, our system will not work either. However, once vanishing lane markings emerge again, our system resurges after a few input video images. As mentioned, the proposed technique can work in both daytime (high intensity environments) and nighttime (low intensity environments). Both lane markings located inside and outside a shadow can be detected. Only a few points near the boundaries of shadows may be missed by our lane marking detector. A few missing marking points would not harm the subsequent lane line fitting. Although straight lines were used for line fitting, they will be able to apply to the lane markings detected from curved roads. The reason is twofold. First, freeways and highways were considered in this study. Curves existing in such roadways are typically gentle. Second, the search space for lane markings has been restricted in the lower half of each input image. Curves present in such a limited space visually approximate to straight lines.

While we preferred to deal with video instability before video sequences are fed into our DAS subsystems, this may not be the best strategy because different subsystems demand different levels of video stability. Individual subsystems possessing their own stabilization process may not be so bad. Or, we can think of endowing the current stabilization process with a mechanism that will adaptively transmit intermediate results of the process to different subsystems. Any strategies that can maximize the overall performance of the entire DAS system will determine our future research direction.

References

[1] Y. Aoyagi and T. Asakura, "A study on traffic sign recognition in scene image using genetic algorithms and neural networks," in *Proc. IEEE IECON Int. Conf. Industrial Electronics, Control, and Instrumentation*, Taipei, Taiwan, 1996, pp. 1838–1843.

[2] T. Bachmann and K. Naab, "On-board detection of friction between tyre and road as an example of a driver assistance system," in *Proc. 6th Intelligent Transport Systems World Congress*, Toronto, ON, Canada, 1999, 2141.

[3] E. K. Bas and J. D. Crisman, "An easy to install camera calibration for traffic monitoring," in *Proc. IEEE Conf. Intelligent Transportation Syst.*, 1997, pp. 362–366.

[4] K. P. Behzad, K. P. Behrooz, and N. S. Netanyahu, "A nonparametric method for fitting a straight line to a noisy image," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 998–1001, Sept. 1989.

[5] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Processing*, vol. 7, pp. 62–81, Jan. 1998.

[6] M. Bertozzi, A. Fascioli, and A. Broggi, "Performance analysis of a low-cost solution to vision-based obstacle detection," in *Proc. IEEE Int. Conf. Intelligent Transportation Syst.*, Tokyo, Japan, 1999, pp. 350–355.

[7] M. Betke, "Recognition, resolution, and complexity of objects subject to affine transformations," *Int. J. Comput. Vision*, vol. 44, no. 1, pp. 5–40, 2001.

[8] A. Broggi and S. Berte, "Vision-based road detection in automotive systems: A real-time expectation-driven approach," *J. Artif. Intell. Res.*, vol. 3, pp. 325–348, 1995.

[9] P. J. Burt and P. Anandan, "Image stabilization by registration to a reference mosaic," in *Proc. ARPA Image Understanding Workshop*, 1994, pp. 425–434.

[10] Y. S. Chen and C. S. Fu, "Winner-update algorithm for nearest neighbor search," in *Proc. Int. Conf. Pattern Recognition*, vol. 2, Barcelona, Spain, Sept. 2000, pp. 708–711.

[11] L. S. Davis, R. Bajcsy, M. Herman, and R. Nelson, "RSTA on the move," in *Proc. ARPA Workshop on Image Understanding*, 1994, pp. 435–456.

[12] E. D. Dickmanns, "The development of machine vision for road vehicles in the last decade," in *Proc. IEEE Int. Symp. Intelligent Vehicles*, July 2002.

[13] C. Drane and C. Rizos, *Positioning Systems in Intelligent Transportation Systems*. Boston, MA: Artech House, 1998.

[14] Z. Duric and A. Rosenfeld, "Image sequence stabilization in real time," *Real-Time Imaging*, vol. 2, no. 5, pp. 271–284, 1996.

[15] Y. Egusa, H. Akahori, A. Morimura, and N. Wakami, "An application of fuzzy set theory for an electronic video camera image stabilizer," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 351–356, 1995.

[16] S. Ernst, C. Stiller, J. Goldbeck, and C. Roessig, "Camera calibration for lane and obstacle detection," in *Proc. Int. Intelligent Transportation Systems Conf.*, Tokyo, Japan, 1999, pp. 356–361.

[17] A. D. L. Escalera and L. Moreno, "Road traffic sign detection and classification," *IEEE Trans. Ind. Electron.*, vol. 44, pp. 848–859, Dec. 1997.

[18] C. Y. Fang, C. S. Fuh, and S. W. Chen, "Detection and tracking of road signs," *Pattern Recog. Image Anal.*, vol. 11, pp. 304–308, 2001.

[19] S. Ghosh and T. Lee, *Intelligent Transportation System: New Principles and Architectures*. New York: CRC, 2000.

[20] A. Guiducci, "Parametric model of the perspective projection of a road with applications to lane keeping and 3-D road reconstruction," *Compu. Vision Image Understand.*, vol. 73, pp. 414–427, 1999.

[21] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt, "Real-time scene stabilization and mosaic construction," in *Proc. 2nd IEEE Workshop Applications of Computer Vision*, 1994, pp. 54–62.

[22] G. Y. Jiang, T. Y. Choi, S. K. Hong, J. W. Bae, and B. S. Song, "Lane and obstacle detection based on fast inverse perspective mapping algorithm," in *IEEE Int. Conf. Systems, Man, Cybernetics*, vol. 4, 2000, pp. 2969–2074.

[23] J. S. Jin, Z. Zhu, and G. Xu, "A stable vision system for moving vehicles," *IEEE Trans. Intell. Transport. Syst.*, vol. 1, pp. 32–39, 2000.

[24] L. A. Klein, *Sensor Technologies and Data Requirement for Intelligent Transportation Systems*. Boston, MA: Artech House, 2001.

[25] S. J. Ko, S. H. Lee, S. W. Jeon, and E. S. Kang, "Fast digital image stabilizer based on gray-coded bit-plane matching," *IEEE Trans. Consumer Electron.*, vol. 45, pp. 598–603, 1999.

[26] W. Li, X. Jiang, and Y. Wang, "Road recognition for vision navigation of an autonomous vehicle by fuzzy reasoning," *Fuzzy Sets Syst.*, vol. 93, pp. 275–280, 1998.

[27] C. Morimoto and R. Chellappa, "Fast electronic digital image stabilization for off-road navigation," *Real-Time Imaging*, vol. 2, no. 5, pp. 285–296, 1996.

[28] J. K. Paik, Y. C. Park, and D. W. Kim, "An adaptive motion decision system for digital image stabilizer based on edge pattern matching," *IEEE Trans. Consumer Electron.*, vol. 38, pp. 607–616, 1992.

[29] G. Piccioli, E. D. Micheli, P. Parodi, and M. Campani, "A robust method for road sign detection and recognition," *Image Vision Comput.*, vol. 14, pp. 209–223, 1996.

[30] K. Ratakonda, "Real-time digital video stabilization for multi-media applications," in *Proc. IEEE Int. Symp. Circuits Systems*, vol. 4, 1998, pp. 69–72.

[31] S. Srinivasan and R. Chellappa, "Image stabilization and mosaicking using the overlapped basis optical flow field," in *Proc. Int. Conf. Image Processing*, vol. 3, 1997, pp. 356–359.

[32] T. Viéville, E. Clergue, and P. E. D. S. Facao, "Computation of ego-motion and structure from visual and internal sensors using the vertical cue," in *Proc. IEEE Int. Conf. Computer Vision*, Berlin, Germany, 1993, pp. 591–598.

[33] Y. Wang, D. Shen, and E. K. Teoh, "Lane detection using spline model," in *Pattern Recogn. Lett.*, vol. 21, 2000, pp. 677–689.

[34] S. M. Wong and M. Xie, "Lane geometry detection for the guidance of smart vehicle," in *Proc. IEEE Int. Conf. Intell. Transport. Syst.*, Boston, MA, 1999, pp. 925–928.

[35] M. Yamada, K. Ueda, I. Horiba, and N. Sugie, "Discrimination of the road condition toward understanding of vehicle driving environments," in *Proc. IEEE Int. Conf. Intell. Transport. Syst.*, Tokyo, Japan, 1999, pp. 20–24.

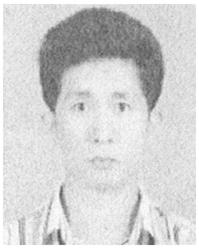
[36] Y. M. Yeh, H. C. Chiang, and S. J. Wang, "A digital camcorder image stabilizer based on gray coded bit-plane block matching," in *Proc. 13th IPPR Conf. Computer Vision, Graphics and Image Processing*, Taipei, Taiwan, 2000, pp. 244–251.

Yu-Ming Liang received the B.S. and M.S. degrees in information and computer education from National Taiwan Normal University in 1999 and 2002, respectively.

He is currently a Research Assistant with the Institute of Information Science, Academia, Sinica, Taipei, Taiwan. His areas of research interest include pattern recognition, image processing, and computer vision.

Hsiao-Rong Tyan received the B.S. degree in electronic engineering from Chung-Yuan Christian University, Chung-Li, Taiwan, in 1984 and the M.S. and Ph.D. degrees in computer science from Northwestern University, Evanston, IL, in 1987 and 1992, respectively.

She is an Associate Professor with the Department of Information and Computer Engineering, Chung-Yuan Christian University, where she currently conducts research in the areas of computer networks, computer security, and intelligent systems.



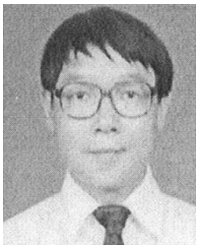
Shyang-Lih Chang was born in Taipei, Taiwan, in 1962. He received the B.S. degrees in electronics engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 1987. He received the M.S. degree from the Department of Computer Science and Information Engineering, National Chiao-Tung University, Hsinchu, Taiwan, in 1990.

Currently, he is a Ph.D. degree candidate with the Department of Information and Computer Education of National Taiwan Normal University, Taipei, Taiwan. He has also been a lecturer with the Department of Electronics Engineering, St. John's and St. Mary's Institute of Technology since August 1993. His research interests include image processing, computer vision, and the design of microprocessor systems.

Hong-Yuan Mark Liao received the B.S. degree in physics from National Tsing-Hua University, Hsin-Chu, Taiwan, in 1981, and the M.S. and Ph.D. degrees in electrical engineering from Northwestern University, Evanston, IL, in 1985 and 1990, respectively.

He was a Research Associate with the Computer Vision and Image Processing Laboratory at Northwestern University during 1990–1991. In July 1991, he joined the Institute of Information Science, Academia, Sinica, Taipei, Taiwan, as an Assistant Research Fellow. He was promoted to Associate Research Fellow and then Research Fellow in 1995 and 1998, respectively. From August 1997 to July 2000, he served as the Deputy Director of the Institute. From February 2002 to January 2004, he was the Acting Director of Institute of Applied Science and Engineering Research. He is jointly appointed as Professor of the Computer Science and Information Engineering Department of National Chiao-Tung University. His current research interests include multimedia signal processing, wavelet-based image analysis, content-based multimedia retrieval, and multimedia protection.

Dr. Liao is the Managing Editor of the *Journal of Information Science and Engineering*. He is on the Editorial Board of the *International Journal of Visual Communication and Image Representation*, *Acta Automatica Sinica*, and the *EURASIP Journal on Applied Signal Processing*. He was an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA during 1998–2001. He was the recipient of the Young Investigators' award from Academia Sinica in 1998; the Excellent Paper Award from the Image Processing and Pattern Recognition society of Taiwan in 1998 and 2000. In 2003, he received the Distinguished Research Award from the National Science Council of Taiwan. He served as the Program Chair of the International Symposium on Multimedia Information Processing (ISMIP'97), the Program Co-Chair of the Second IEEE Pacific-Rim Conference on Multimedia (2001), and the Conference Co-Chair of the Fifth IEEE International Conference on Multimedia and Exposition (ICME).



Sei-Wan Chen received the B.Sc. degree in atmospheric and space physics, the M.Sc. degree in geophysics from National Central University in 1974 and 1976, respectively, and the M.Sc. and Ph.D. degrees in computer science and engineering from Michigan State University, East Lansing, in 1985 and 1989, respectively.

From 1977 to 1983, he worked as a Research Assistant in the Computer Center of Central Weather Bureau, Taiwan, Republic of China. In 1990, he was a researcher with the Advanced Technology Center of the Computer and Communication Laboratories at the Industrial Technology Research Institute, Hsinchu, Taiwan. From 1991 to 1994, he was an Associate Professor and from 1995 to 2001, he was a Full Professor with the Department of Information and Computer Education, National Taiwan Normal University, Taipei, Taiwan. He is currently a Professor of the Graduate Institute of Computer Science and Information Engineering at the same university. His areas of research interest include neural networks, fuzzy systems, pattern recognition, image processing, and computer vision.