3-D Motion Estimation Using Range Data

Hamid Gharavi, *Fellow, IEEE*, and Shaoshuai Gao

***Abstract*—Advanced vehicle-based safety and warning systems use laser scanners to measure road geometry (position and curvature) and range to obstacles in order to warn a driver of an impending crash and/or to activate safety devices (air bags, brakes, and steering). In order to objectively quantify the performance of such a system, the reference system must be an order of magnitude more accurate than the sensors used by the warning system. This can be achieved by using high-resolution range images that can accurately perform object tracking and velocity estimation. Currently, this is very difficult to achieve when the measurements are taken from fast moving vehicles. Thus, the main objective is to improve motion estimation, which involves both the rotational and translation movements of objects. In this respect, an innovative recursive motion-estimation technique that can take advantage of the in-depth resolution (range) to perform accurate estimation of objects that have undergone three-dimensional (3-D) translational and rotational movements is presented. This approach iteratively aims at minimizing the error between the object in the current frame and its compensated object using estimated-motion displacement from the previous range measurements. In addition, in order to use the range data on the nonrectangular grid in the Cartesian coordinate, two approaches have been considered: 1) membrane fit, which interpolates the nonrectangular grid to the rectangular grid, and 2) the nonrectangular-grid range data by employing derivative filters and the proposed transformation between the Cartesian coordinates and the sensor-centered coordinates. The effectiveness of the proposed scheme is demonstrated for sequences of moving-range images.**

***Index Terms*—Intelligent transport, ladar, laser scanners, object tracking, range image, three-dimensional (3-D) motion estimation, vehicle safety.**

I. Introduction

THE CLASSICAL motion-estimation techniques in computer vision use intensity images or stereovision to estimate three-dimensional (3-D) motion parameters. These techniques are not yet sufficiently robust or fast enough to be used for highly sensitive real-time systems such as crash-prevention operations [1]–[4]. Recently, with the rapid progress of range-camera technology, capturing what is referred to as two-and-a-half-dimensional (2.5-D) images is becoming possible. Currently, high-speed range cameras are capable of acquiring raster-depth measurements of an object with relatively high frame rates. These images can provide precise measurements of the geometry of the 3-D environment, including all three Cartesian coordinates of the points on an object. This can make motion estimation and object tracking much easier and more reliable compared with using only video-intensity images.

Range cameras can be classified into two distinct categories, namely 1) matrix raster imaging and 2) single-beam 3-D scanners. The most common sensor technique used in the first category is triangulation range scanners, which are suitable for short ranges. In this technology, a scene is illuminated with structured light. The range is then measured by triangulating corresponding points, using the baseline between the transmitter and receiver. Another example in this category is flash-ladar technology, which uses short flashes of modulated laser light to illuminate the field of view. A matrix of receiver elements senses the returned modulated light, which can then measure range via modulation shifts. This technology, which has not yet fully matured for industrial applications, can generate high frame rates; however, the sensitivity and resolution of the receiver elements is still quite low.

In single-laser-beam scanning technology, a deflection-mirror assembly scans a beam over the scene. This type of technique has been widely used for many tactical and industrial applications and uses different types of range-measurement technologies. One example is the use of an amplitude-modulated continuous wave (AMCW) laser-range module. It gives both intensity (reflectance) and range information. Another example is time of flight (pulsed) laser-range modules, which send short pulses that are reflected by surrounding objects. By detecting the reflections and measuring the time of flight, the system can then calculate distances to objects. Time-of-flight-based laser scanners, due to their lower cost, have been widely used for object tracking in automobile safety, navigation, and robot-vehicle applications. The frame rate and scan resolution of single-beam scanning technology systems depend on the data rate of the laser-range module, which determines the speed with which images can be scanned at high angular resolution. Note that with this technology, a 3-D scan of a scene is obtained by deflecting the laser beam in equal increments of angle in horizontal and vertical planes. A scanned scene can then be represented in terms of range ρ, horizontal angle θ, and elevation angle ϕ, which corresponds to a spherical (polar) coordinate system.

By converting a range image from the spherical coordinate system to a so-called Cartesian elevation map (CEM), Horn and Harris [5] developed a recovery system for the six degrees of freedom of motion of a vehicle, which has been a challenging problem in autonomous navigation. In CEM, depth Z is expressed as a function of X and Y, which corresponds to displacements in the horizontal plane. This time-varying CEM is used to estimate the translational and rotational movements of rigid objects.

Manuscript received February 16, 2006; revised April 26, 2006, June 12, 2006, and June 23, 2006. The Associate Editor for this paper was N. Zheng.

The authors are with the National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD 20899-8920 USA (e-mail: gharavi@nist.gov; sago@nist.gov).

Digital Object Identifier 10.1109/TITS.2006.883112

Although the optimized solution offered by Horn and Harris has been very effective, it does not always produce very accurate estimation of 3-D motion displacements, which is crucial for highly sensitive operations such as crash detection and prevention. Thus, we present here a recursive approach to enhance estimation accuracy. This iterative approach is based on minimizing the error between the new position of the object and its previous location, after being compensated using estimated-motion displacements. Details of the proposed algorithm are presented in Section II. Since a set of 3-D points obtained in the CEM coordinate may not be placed regularly on a rectangular grid, we have developed a method that uses a nonrectangular grid to reconstruct the displaced frame. This scheme employs derivative filters, together with transformation between the Cartesian coordinates and sensor-centered coordinates for image reconstruction. Details of this approach, together with the membrane-fit technique [20], which is normally used to map the CEM image into a regular rectangular grid, are also discussed in Section II. Their performances for the proposed recursive estimation are compared in the simulation section. In addition, we describe how sequences of range video images can be synthetically generated to evaluate and compare the performance of the motion-estimation techniques presented in this paper.

II. 3-D Motion Estimation

In general, there are two classes of motion-estimation algorithms for range images: Class one is for rigid-motion surfaces [5]–[14], and the other is for moving deformable surfaces [15]–[18]. We are concerned with rigid motion due to the transportation nature of our applications. Class one can be further divided into two categories: One is a feature-based algorithm [12]–[14] whose performance depends on the detection of reliable range image features and the establishment of interframe correspondence among them. The other is a direct area-based algorithm [5]–[11], which is more straightforward than the feature-based algorithm. Thus, in our approach, we have considered the direct area-based algorithms to estimate the motion parameters.

A. Recursive Algorithm

Recovery of the six degrees of freedom of motion displacement can be best accomplished by using time-varying CEM, as proposed by Horn and Harris [5]. Their algorithm is based on the assumption that most of the surface is smooth so that local tangent planes can be constructed. In addition, the motion between frames is smaller than the size of most features in the range image (e.g., the onboard sensors provide estimation of the motion, which can be used to approximately register the range maps). Furthermore, the environment is a single rigid assemblage, and only the motion of the sensor relative to the environment has to be recovered. In other words, the whole range image should have the same rigid-motion parameters.

A time-varying CEM can be expressed as a function of the form $Z(X, Y, t)$, where t denotes time, Z is the depth, and X and Y are the displacements in the horizontal and vertical planes, respectively. For a rigid-motion scene, the motion can be described as instantaneous translational velocity (a vector with three elements) and instantaneous angular velocity (a vector with three elements). For every 3-D point, an elevation-rate-constraint equation relating the derivatives of X, Y, and Z can be obtained as [5]

$$\dot{Z} = p\dot{X} + q\dot{Y} + Z_t \tag{1}$$

where $p = \partial Z/\partial X$, $q = \partial Z/\partial Y$, and $Z_t = \partial Z/\partial t$.

The components of velocity of a point in the range image are

$$\dot{X} = \frac{dX}{dt}, \quad \dot{Y} = \frac{dY}{dt}, \quad \dot{Z} = \frac{dZ}{dt}. \tag{2}$$

The vector to a point on the surface is $R = (X, Y, Z)^T$, and

$$\frac{dR}{dt} = -\mathbf{t} - \omega \times R \tag{3}$$

where $t = [U\ V\ W]^T$ is the translational velocity, and $\omega = [A\ B\ C]^T$ is the rotational velocity.

From (2) and (3)

$$\begin{cases} \dot{X} = -U - BZ + CY \\ \dot{Y} = -V - CX + AZ \\ \dot{Z} = -W - AY + BX \end{cases}. \tag{4}$$

From (1) and (4)

$$pU + qV - W + rA + sB + tC = Z_t \tag{5}$$

where $r = -Y - qZ$, $s = X + pZ$, and $t = qX - pY$.

Let us assume that there is a set of m pixels in the image, and for each such pixel, we define the following set of six-dimensional vectors for the nth pixel:

$$\Phi_n = \begin{bmatrix} p_n \\ q_n \\ -1 \\ r_n \\ s_n \\ t_n \end{bmatrix}, \quad D = \begin{bmatrix} U \\ V \\ W \\ A \\ B \\ C \end{bmatrix}.$$

From (5), the rate of change for elevation Z_t at pixel n can be shown as

$$(Z_t)_n = \Phi_n^T D. \tag{6}$$

Based on (6), we can estimate the motion iteratively, where at each iteration, the previous estimate is used in the process. Let us assume that in this process, two consecutive video frames (which were generated at a fixed frame rate) are used to measure the change of rate of elevation. After each iteration, the estimated-motion vectors are used to reconstruct the compensated first frame for the next iteration.

From (6), we can show that

$$(\gamma_n)^{i-1} = \left(\Phi_n^T\right)^{i-1}\left(D - \overline{D}^{i-1}\right) + \text{noise} \tag{7}$$

or

$$\text{noise} = (\gamma_n)^{i-1} - \left(\Phi_n^T\right)^{i-1}\left(D - \overline{D}^{i-1}\right) \tag{8}$$

where γ is the measurement of the displaced frame difference (DFD) between the second frame and the compensated first frame (i.e., the estimated second frame) using the estimated-motion vectors [21].

For a cluster of m moving pels, after carrying out the minimization, the least squares estimate of D is

$$\sum_{n=1}^{m}(\gamma_n)^{i-1}(\Phi_n)^{i-1} = \left(\overline{D}^{i} - \overline{D}^{i-1}\right)\left[\sum_{n=1}^{m}(\Phi_n)^{i-1}\left(\Phi_n^T\right)^{i-1}\right]. \tag{9}$$

Thus

$$\overline{D}^{i} = \overline{D}^{i-1} + \left[\sum_{n=1}^{m}(\Phi_n)^{i-1}\left(\Phi_n^T\right)^{i-1}\right]^{-1}\sum_{n=1}^{m}(\gamma_n)^{i-1}(\Phi_n)^{i-1}. \tag{10}$$

As previously mentioned, in the preceding iterative process, it is necessary to incorporate the estimated-motion displacement vector to reconstruct the displaced moving object in the frame. In order to obtain the new position of each displaced pixel on a nonrectangular grid in CEM, we used two approaches: In the first approach, we consider the membrane-fit interpolation model [20]. In the second, we developed a combination of derivative filters [20] and transformation between the Cartesian coordinates and the sensor-centered coordinates in a nonrectangular grid. Both methods are described in the succeeding sections.

B. Membrane Fit

This model can be used to place a set of 3-D points on the rectangular grid after transforming the range data from spherical (sensor-centered) to Cartesian coordinates [20]. Interpolation is based on computing an estimate e in region A that uses the measured data m on a nonrectangular grid. For an optimal solution, this would require minimizing the following energy functional:

$$\int_A h(e)\,dx\,dy \rightarrow \min \tag{11}$$

where

$$h(e) = \omega(e-m)^2 + \alpha\left(e_x^2 + e_y^2\right), \qquad e_x = \frac{\partial e}{\partial x}; \quad e_y = \frac{\partial e}{\partial y}$$

x, y are the sensor-grid (range image) index, and ω is the confidence value, which is one for nonzero areas in the image and zero elsewhere. Note that the smoothness is regulated via α [20]. In addition, the region of interest A could be the entire image or a previously selected subset.

The minimization of the preceding function is found by solving the Euler–Lagrange equation (for each pixel)

$$\frac{\partial h}{\partial e} - \frac{d}{dx}\frac{\partial h}{\partial e_x} - \frac{d}{dy}\frac{\partial h}{\partial e_y} = 0. \tag{12}$$

Thus, we have the following equations for each pixel:

$$\begin{aligned} 2\omega(e-m) - 2\alpha e_{xx} - 2\alpha e_{yy} &= 0 \\ \omega e - \omega m - \alpha\Delta e &= 0. \end{aligned} \tag{13}$$

In a discrete implementation, the Laplacian Δe in (13) can be estimated by the difference between a local average and the central value: $\Delta e = \bar{e} - e$. Using this Laplacian approximation, the Euler–Lagrange equation can be written as

$$(\omega + \alpha)e = \alpha\bar{e} + \omega m. \tag{14}$$

Solving this iteratively, we can show that

$$(e)^{i+1} = \frac{\alpha}{\omega+\alpha}(\bar{e})^i + \frac{\omega m}{\omega+\alpha}. \tag{15}$$

As we are dealing with a convex energy functional, the preceding iterative process is expected to converge. Thus, initialization can be accomplished with zero. However, since this may result in slow convergence, we initialize it instead with an estimate found by linearly interpolating the available data points. The iterations are run until either a maximum number of iterations (typically 1000) is reached or the mean change between iterations is less than a threshold (here, 1×10^{-6}).

C. Using the Nonrectangular-Grid Range Data

1) Derivative Filter: To use the range data on the nonrectangular sensor grid directly for motion estimation, a new version of the range flow constraint equation is derived in [20]. The three components of the motion vector for one point (i.e., on the X, Y, and Z directions) can be written as

$$\begin{cases} \dot{X} = X_x\,\dot{x} + X_y\,\dot{y} + X_t \\ \dot{Y} = Y_x\,\dot{x} + Y_y\,\dot{y} + Y_t \\ \dot{Z} = Z_x\,\dot{x} + Z_y\,\dot{y} + Z_t \end{cases} \tag{16}$$

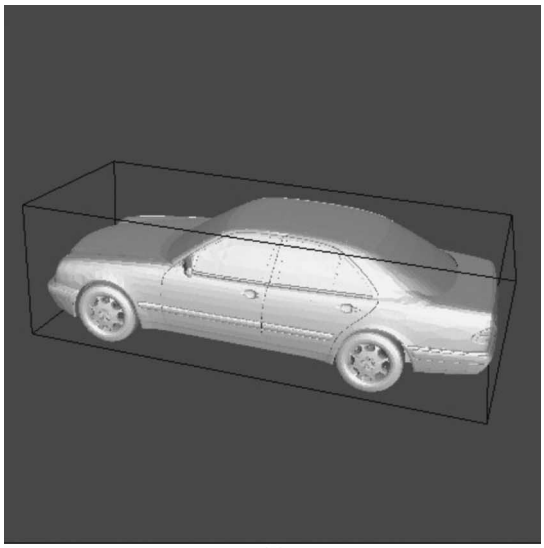
where $\dot{X}$, $\dot{Y}$, and $\dot{Z}$ have been shown in (2), and

$$X_x = \frac{\partial X}{\partial x}, \quad X_y = \frac{\partial X}{\partial y}, \quad X_t = \frac{\partial X}{\partial t}$$

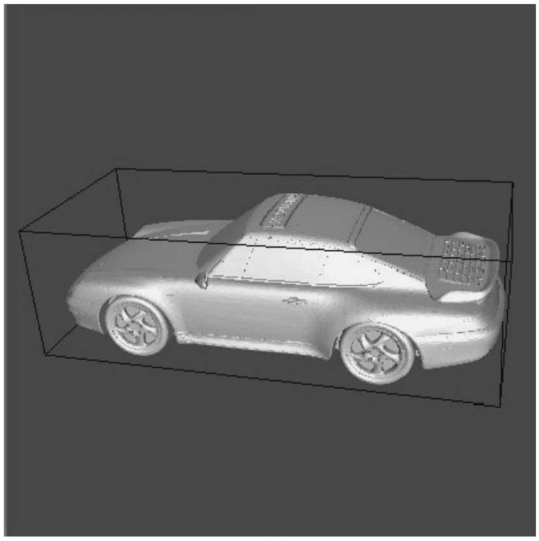
$$Y_x = \frac{\partial Y}{\partial x}, \quad Y_y = \frac{\partial Y}{\partial x}, \quad Y_t = \frac{\partial Y}{\partial t}$$

$$Z_x = \frac{\partial Z}{\partial x}, \quad Z_y = \frac{\partial Z}{\partial y}, \quad Z_t = \frac{\partial Z}{\partial t}$$

$$\dot{x} = \frac{dx}{dt}, \qquad \dot{y} = \frac{dy}{dt}.$$



(a)



(b)

Fig. 1. OOGL files. (a) Auto and (b) Porsche.

Eliminating $\dot{x}$ and $\dot{y}$ gives (17), shown at the bottom of the page.

Compared with (1), it can be seen that

$$\begin{cases} p = \frac{Y_y Z_x - Y_x Z_y}{X_x Y_y - X_y Y_x} \\ q = \frac{X_x Z_y - X_y Z_x}{X_x Y_y - X_y Y_x} \\ Z_t = \frac{X_x Y_y Z_t + X_y Y_t Z_x + X_t Y_x Z_y - X_x Y_t Z_y - X_t Y_y Z_x - X_y Y_x Z_t}{X_x Y_y - X_y Y_x} \end{cases}. \tag{18}$$

2) Proposed Reconstruction in a Nonrectangular Grid: In order to reconstruct the first frame after each iteration in a nonrectangular grid, we perform motion compensation directly

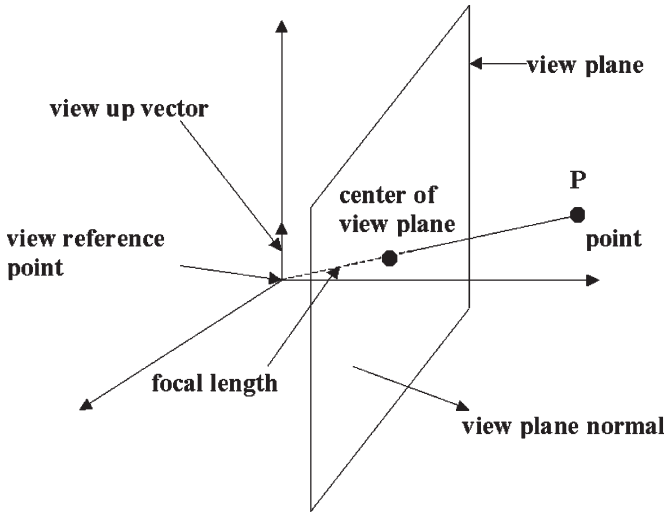


Fig. 2. Camera parameters.

to the spherical (polar) coordinate. This requires transformation between (ρ, θ, ϕ) and (X, Y, Z) each time the motion vector estimation is updated. The transformation from sensor-centered coordinates (ρ, θ, ϕ) to Cartesian coordinates (X, Y, Z) can be shown as

$$\begin{cases} X = \rho \sin\theta \cos\phi \\ Y = \rho \sin\phi \\ Z = \rho \cos\theta \cos\phi \end{cases}. \tag{19}$$

Similarly, the transformation from (X, Y, Z) to (ρ, θ, ϕ) can be shown as

$$\begin{cases} \rho = \sqrt{X^2 + Y^2 + Z^2} \\ \theta = \arctan\frac{X}{Z} \\ \phi = \arctan\frac{Y}{\sqrt{X^2 + Z^2}} \end{cases}. \tag{20}$$

Given the first frame F_1 and the estimated-motion vector MV, the estimated second frame $\hat{F}_2$ will be

$$\begin{cases} \hat{X}_2(x', y') = X_1(x, y) + MV_X \\ \hat{Y}_2(x', y') = Y_1(x, y) + MV_Y \\ \hat{Z}_2(x', y') = Z_1(x, y) + MV_Z \end{cases} \tag{21}$$

where x, y, x', and y' are the image index. For range data on the rectangular grid, we can directly obtain (x', y') as

$$\begin{cases} x' = x + \frac{MV_X}{\Delta X} \\ y' = y + \frac{MV_Y}{\Delta Y} \end{cases} \tag{22}$$

where $\Delta X = X(x+1, y) - X(x, y)$, and $\Delta Y = Y(x, y+1) - Y(x, y)$.

$$\dot{Z} = \frac{Y_y Z_x - Y_x Z_y}{X_x Y_y - X_y Y_x}\dot{X} + \frac{X_x Z_y - X_y Z_x}{X_x Y_y - X_y Y_x}\dot{Y} + \frac{X_x Y_y Z_t + X_y Y_t Z_x + X_t Y_x Z_y - X_x Y_t Z_y - X_t Y_y Z_x - X_y Y_x Z_t}{X_x Y_y - X_y Y_x} \tag{17}$$

TABLE I
PARAMETERS SET IN THE SIMULATION

Parameters \ Sequence	Left side of "Auto"	Back of "Auto"	Left side of "Porsche"	Back of "Porsche"
View Plane Normal	(0, 1, 0)	(-1, 0, 0)	(0, 1, 0)	(-1, 0, 0)
View Up Vector	(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	(0, 0, 1)
View Reference Point	(70.8, -201.2, 34.7)	(250.8, 31.2, 34.7)	(0, -240, 32.7)	(200, 0, 32.7)
Resolution of the output images	(400, 400)	(400, 400)	(400, 400)	(400, 400)
Size of the camera image plane	(0.05, 0.05)	(0.05, 0.05)	(0.05, 0.05)	(0.05, 0.05)
Focal Length	0.06	0.06	0.06	0.06

However, since the 3-D range data in the (X, Y, Z) coordinate system are not on the rectangular grid, we cannot directly incorporate the motion vector to reconstruct the motion-compensated frame. At the same time, the 3-D points in the sensor-centered coordinate (ρ, θ, ϕ) system has a property in which $\Delta\theta$ and $\Delta\phi$ are constant, where $\Delta\theta = \theta(x+1, y) - \theta(x, y)$ and $\Delta\phi = \phi(x, y+1) - \phi(x, y)$.

Therefore, each time the motion vector is estimated in the X, Y, Z coordinates, motion compensation is performed on the spherical coordinate, where

$$(X_1, Y_1, Z_1) \rightarrow (\rho_1, \theta_1, \phi_1)$$

$$(\hat{X}_2, \hat{Y}_2, \hat{Z}_2) \rightarrow (\hat{\rho}_2, \hat{\theta}_2, \hat{\phi}_2).$$

The compensated image is then transformed back to the Cartesian coordinate system for motion estimation, i.e.,

$$\begin{cases} x' = x + \frac{\hat{\theta}_2 - \theta_1}{\Delta\theta} \\ y' = y + \frac{\hat{\phi}_2 - \phi_1}{\Delta\phi} \end{cases}. \tag{23}$$

III. SIMULATIONS

In order to quantitatively analyze our proposed 3-D motion-estimation algorithm, we have developed a method to synthetically generate sequences of moving-range images. In particular, these moving images are produced in such away that a 3-D object can be displaced in accordance with the predefined-motion displacement parameters. These images can allow us to evaluate the accuracy of estimated-motion vectors with reference to the actual displacement parameters.

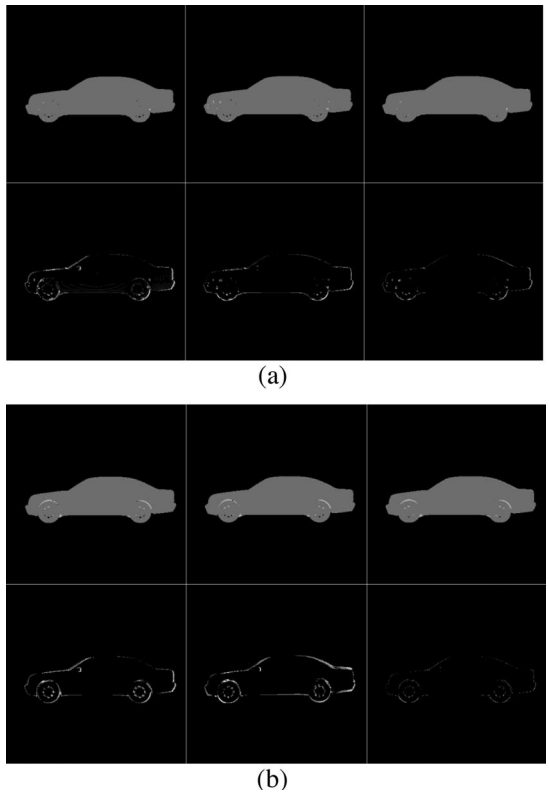


Fig. 3. Subjective comparison of different algorithms for the left side of Auto with $U = -1$, $V = -0.5$, and $W = -1$. (a) Approach 1. (b) Approach 2. From left to right, then top to bottom. (1) First image. (2) Second image. (3) Estimated second image using the motion parameters of the final iteration. (4) Difference image between the original first and second images. (5) Difference image between the second image and the estimated second image (DFD) using the estimated-motion parameters of the first iteration (Horn's algorithm). (6) Difference image between the second image and the estimated second image using the motion parameters of the final iteration.

A. Generating Range Video Data

Moving-range image sequences were constructed via 3-D object-oriented graphics library (OOGL) files. OOGL is a 3-D object data file in which an object is defined by vertices, lines, and surfaces. Fig. 1 shows two OOGL files that were selected to generate range video sequences for our simulation. One is called "Auto," and the other one is simply referred to as "Porsche" [19]. These 3-D OOGL images were then used to generate a sequence of 2.5-D moving-range image files (RIFs).

A RIF is a range image format that is based on the Cartesian coordinates (X, Y, and Z components) and consists of the object points and the mask map (which indicates the locations of object points). In this format, each frame is constructed by displacing first the object in the OOGL file and then transforming it into a RIF file. In this way, we can create a sequence of moving-range images (frames), where the object in each frame can be displaced by a predefined 3-D motion vector. Two consecutive transformed RIF images were then used as inputs to the motion estimator.

To get the RIF files from the OOGL files, we must set some parameters such as view plane normal, view up vector, camera position ("view reference point"), resolution of the output images, size of the camera image plane, and focal length. Fig. 2 shows some of these parameters.

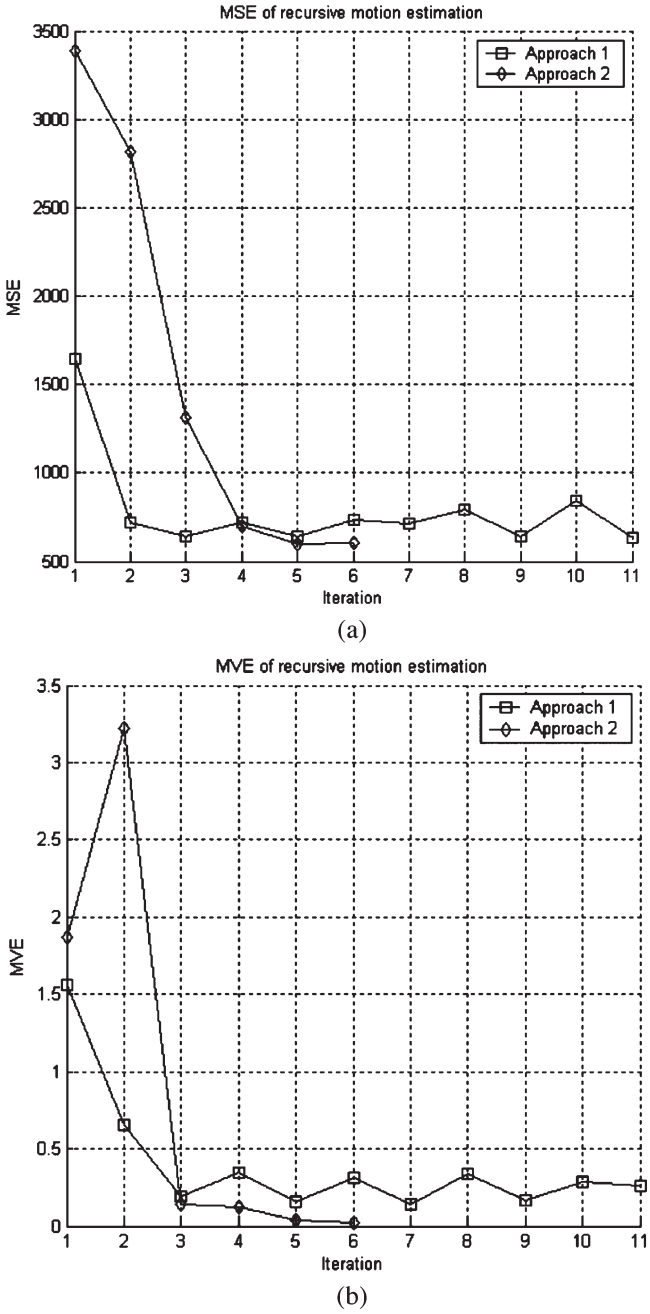


Fig. 4. Objective comparison of different algorithms for the left side of Auto with $U = -1$, $V = -0.5$, and $W = -1$. (a) MSE. (b) MVE.

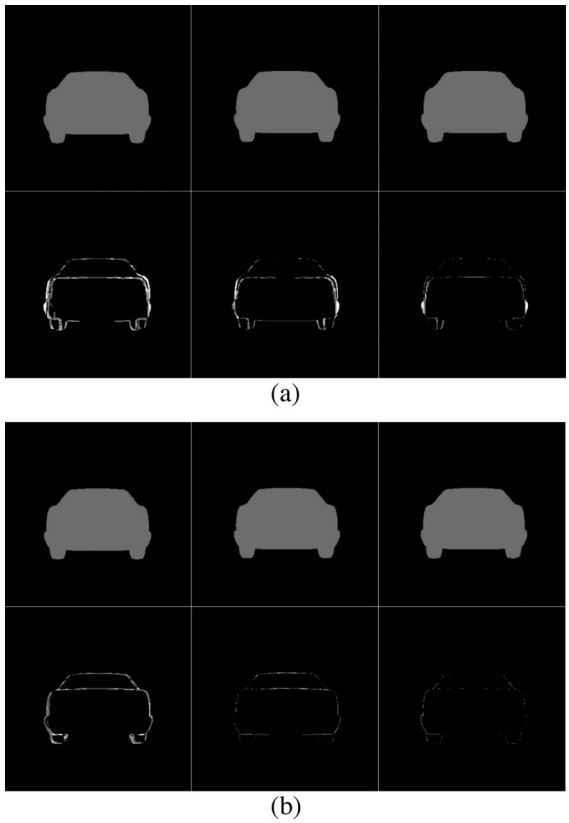


Fig. 5. Subjective comparison of different algorithms for the back of Auto with $U = 1$, $V = -0.5$, and $W = -1$. (a) Approach 1. (b) Approach 2. From left to right, then top to bottom. (1) First image. (2) Second image. (3) Estimated second image using the motion parameters of the final iteration. (4) Difference image between the original two images. (5) Difference image between the second image and the estimated second image (DFD) using the motion parameters of the first iteration (Horn's algorithm). (6) Difference image between the second image and the estimated second image (DFD) using the motion parameters of the final iteration.

To move objects, one method is changing the view reference point, up vector, and view plane normal when converting an OOGL file to a RIF file. The other method used in our paper is adding a transformation matrix in the OOGL file and then creating the second frame. The transformation matrix is a 4×4 real matrix for homogeneous object transformation. It can represent all of the 3-D transformations such as rotation, translation, scaling, shearing, and perspective. It acts by applying multiplication on the right of the vectors. Thus, if p is a four-element row vector that represents the homogeneous coordinates of a point in the OOGL object and M is the 4×4 matrix, then the transformed point is $p' = pM$. Suppose frame 1 $F_1 = [X_1, Y_1, Z_1]$ and transformation matrix M, frame 2 can be represented as

$$[F_2, 1] = [F_1, 1] \times M.$$

For rigid-motion objects with sensor-centered coordinates, M is in the form of

$$M = \begin{bmatrix} 1 & C & -B & 0 \\ -C & 1 & A & 0 \\ B & -A & 1 & 0 \\ U & V & W & 1 \end{bmatrix}.$$

By setting the motion parameters U, V, W, A, B, and C, we can get the second frame. Then, we use motion-estimation algorithms to get the estimated-motion parameters $\hat{U}, \hat{V}, \hat{W}, \hat{A}, \hat{B}$, and $\hat{C}$. In this case, we know the true motion parameters, which can be used to evaluate the exactness of the motion-estimation algorithms.

It should be noted that the transformation from the OOGL file to the RIF file is a sampling process, which inevitably introduces sampling errors. Consequently, even the true motion parameters cannot give the same reconstructed frame (estimated), compared with the original frame.

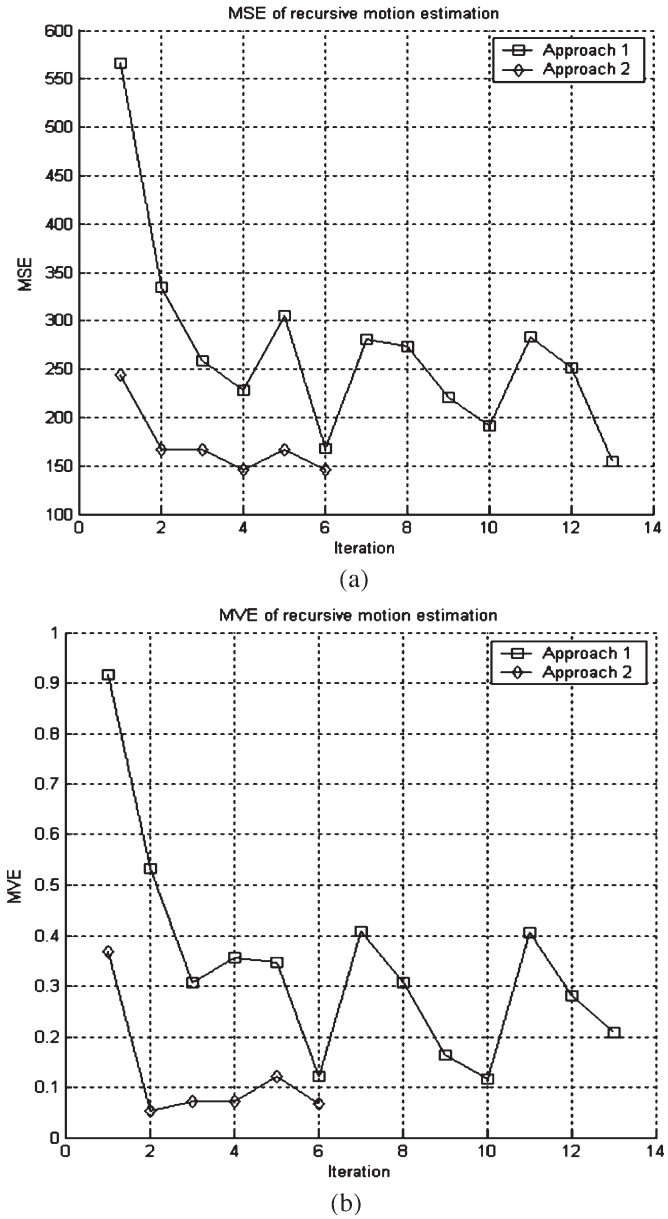


Fig. 6. Objective comparison of different algorithms for the back of Auto with $U = 1$, $V = -0.5$, and $W = -1$. (a) MSE. (b) MVE.

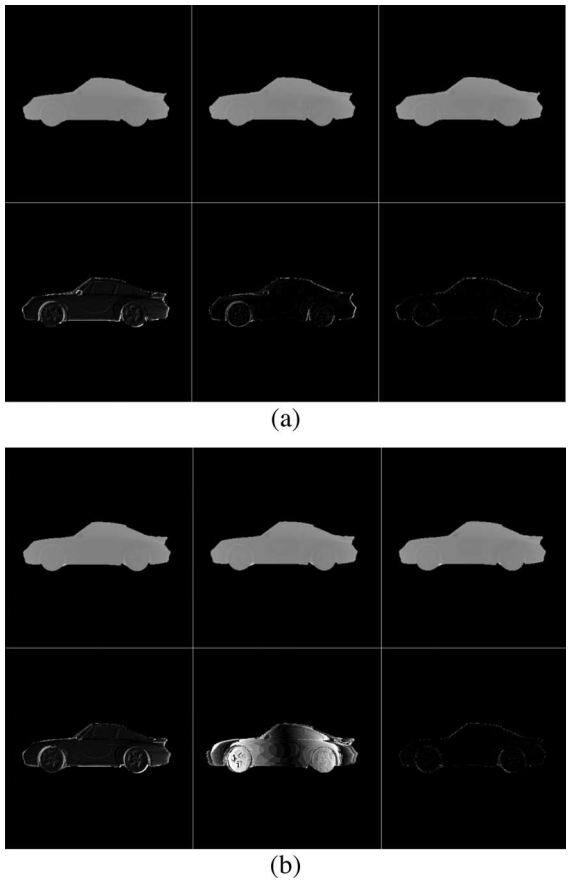


Fig. 7. Subjective comparison of different algorithms for the left side of Porsche with $U = -1$, $V = -0.5$, and $W = -1$. (a) Approach 1. (b) Approach 2. From left to right, then top to bottom. (1) First image. (2) Second image. (3) Estimated second image using the motion parameters of the final iteration. (4) Difference image between the original two images. (5) Difference image between the second image and the estimated second image (DFD) using the estimated-motion parameters of the first iteration (Horn's algorithm). (6) Difference image between the second image and the estimated second image using the motion parameters of the final iteration.

B. Results

In this section, we present the simulation results of the proposed motion-estimation technique in accordance with (10). From this equation, we can observe that for $i = 1$ (first iteration) and for the initial estimate $D^0 = 0$, (10) reduces to the Horn and Harris algorithm [5]. Therefore, any improvement after the first iteration is credited to the proposed recursive method and not to the Horn and Harris algorithm. Another factor that affects the performance of the estimation method is dealing with the nonrectangular grid that is typical of range images in the (X, Y, Z) coordinate system. As described in Section II, we have considered two distinct approaches: Approach 1 uses membrane fit to map a range image into a rectangular grid. Approach 2, which is also proposed in this paper, operates in the original nonrectangular-grid format and uses a combination of derivative filters and transformation between (ρ, θ, ϕ) and (X, Y, Z). Both methods have been implemented, and their performances will be presented in this section.

We use two criteria as a measure of performance: mean square error (MSE) and motion vector error (MVE). The MSE between frames 1 and 2 is defined as

$$\text{MSE} = \frac{1}{m} \sum_{R} \left[(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2 \right]$$

where R is the region that combines both objects in two frames $R = \text{MASK}_1 \cup \text{MASK}_2$, and m is the number of the points in region R.

Given the true motion parameters U, V, W, A, B, and C and the estimated ones $\hat{U}, \hat{V}, \hat{W}, \hat{A}, \hat{B}$, and $\hat{C}$, MVE is defined as

$$\text{MVE} = \frac{|U-\hat{U}|+|V-\hat{V}|+|W-\hat{W}|+|A-\hat{A}|+|B-\hat{B}|+|C-\hat{C}|}{|U|+|V|+|W|+|A|+|B|+|C|}.$$

It should be noted that in our recursive motion-estimation algorithm, we use MSE as the standard criterion to determine when iterations should be stopped. Although our recursive

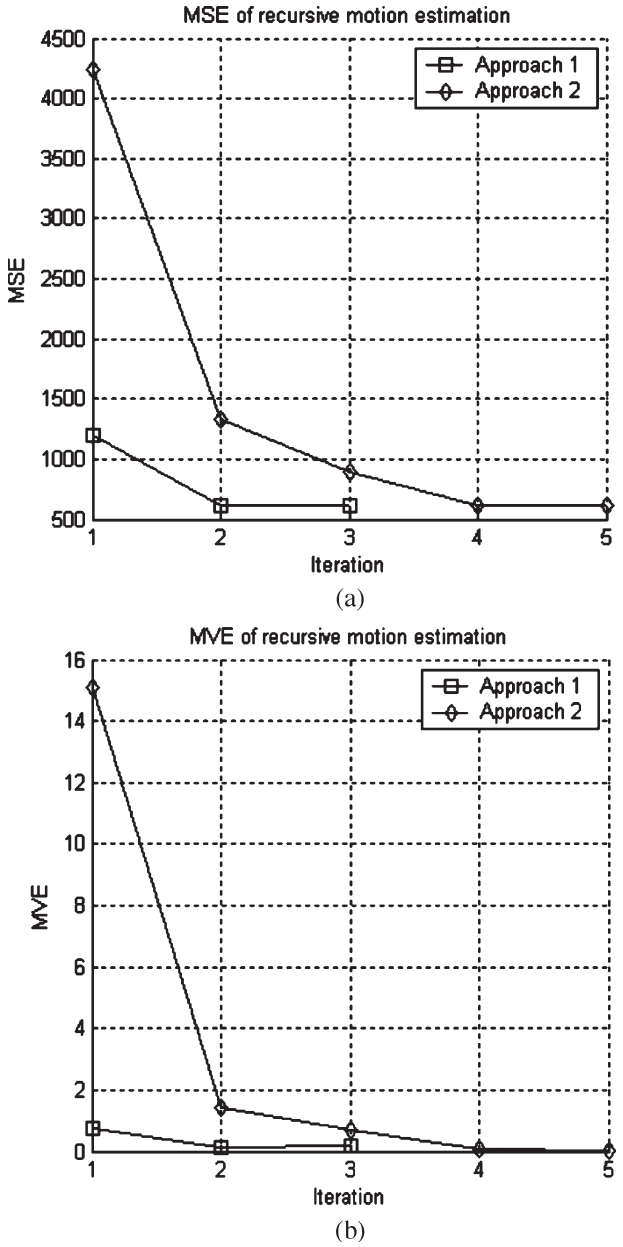


Fig. 8. Objective comparison of different algorithms for the left side of Porsche with $U = -1$, $V = -0.5$, and $W = -1$. (a) MSE. (b) MVE.

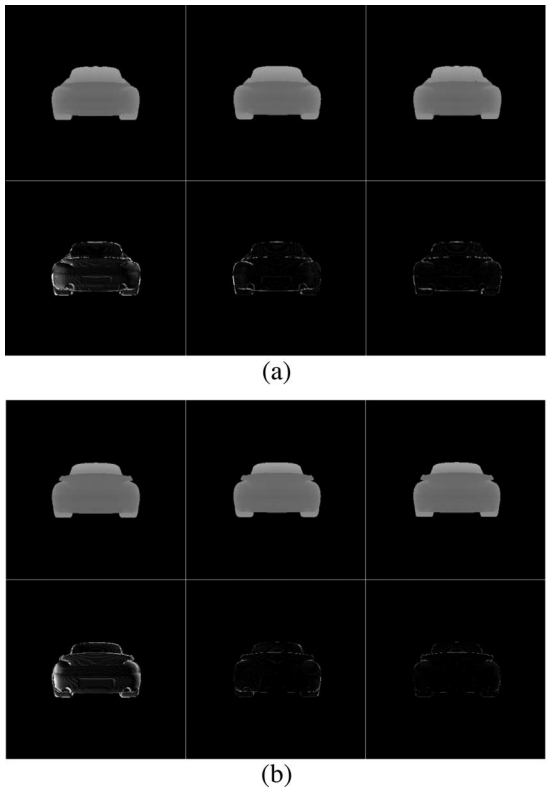


Fig. 9. Subjective comparison of different algorithms for the back of Porsche with $U = 1$, $V = -0.5$, and $W = -1$. (a) Approach 1. (b) Approach 2. From left to right, then top to bottom. (1) First image. (2) Second image. (3) Estimated second image using the motion parameters of the final iteration. (4) Difference image between the original two images. (5) Difference image between the second image and the estimated second image (DFD) using the estimated-motion parameters of the first iteration (Horn's algorithm). (6) Difference image between the second image and the estimated second image using the motion parameters of the final iteration.

method is based on minimizing the MSE, we use the MVE criterion to assess the estimation performance. It should be noted that in reality, the MVE criterion cannot be utilized, as the true motion displacement is unknown. Thus, the use of MVE in our experiments is simply to determine the accuracy of the motion-estimation schemes. In our experiments, we set the maximum number of iterations to 16. However, if the MSE difference between successive iterations is less than a threshold (i.e., 0.1) and the current MSE is larger than the previous one, the previous estimation will be selected.

We carried out these experiments under various scenarios and test conditions. For example, we used different parameters to transform a 3-D image (see Fig. 1) from OOGL to RIF. Based on the 3-D test images shown in Fig. 1, we created a large number of range video sequences with different view angles and different translation and rotational motions. For the two test images, we have examined six scenarios. The first four use translation motion only. The fifth uses rotation motion. The last uses both translation and rotation motions with different levels of synthetic noise. Table I gives the parameters used when converting OOGL files to RIF files in our simulation.

The results of our experiments are presented subjectively and objectively. In the subjective results, we show a difference between the second frame and the estimated second frame. Note that the estimated second frame corresponds to the reconstructed first frame that was compensated by an estimated-motion vector after each iteration. This frame difference, as shown by (7) in Section II, corresponds to the DFD. The first set of results (the first scenario), which shows two consecutive frames of the left side of the Auto image using only translational movement, is depicted in Figs. 3 and 4. It can be clearly observed that the results of the first iteration, which correspond to the Horn and Harris algorithm, are very poor. This is mainly because the surfaces of some objects are not smooth enough and there are many surfaces that are not always conjoined smoothly. However, after successive iterations, the estimated-motion parameters approach the actual-motion parameters.

It can also be seen that the combination of the recursive motion-estimation scheme and the proposed approach 2 can

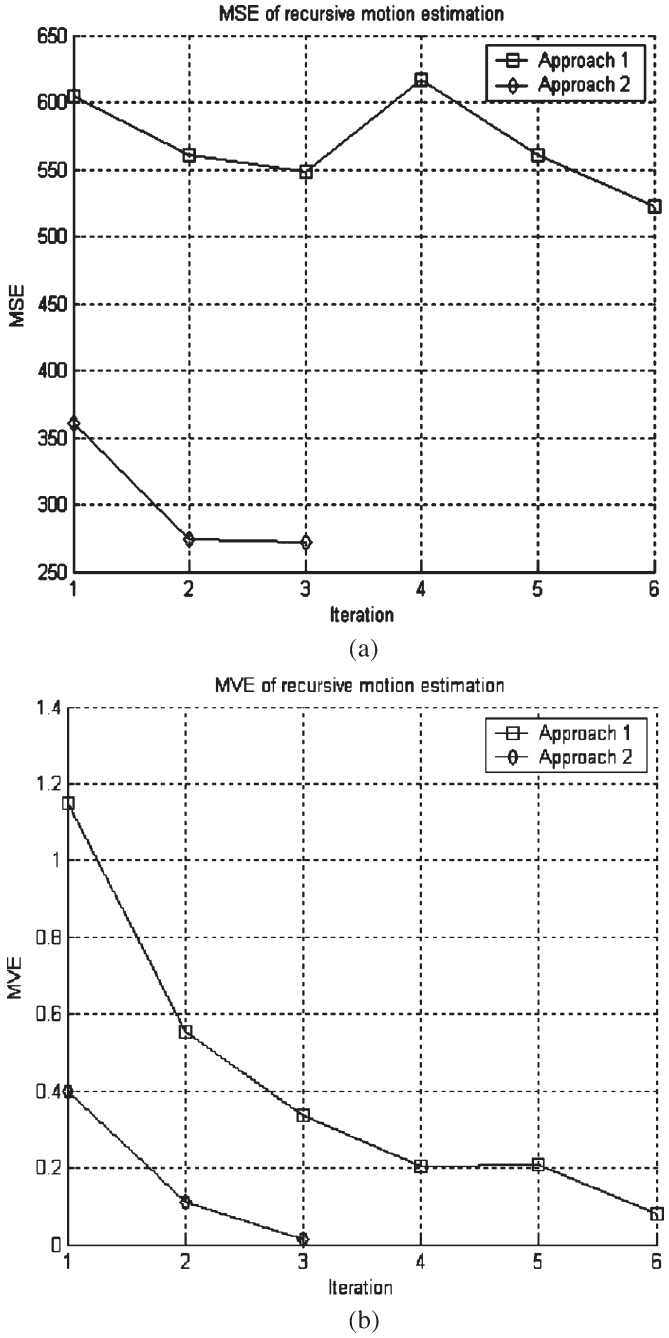


Fig. 10. Objective comparison of different algorithms for the back of Porsche with $U = 1$, $V = -0.5$, and $W = -1$. (a) MSE. (b) MVE.

outperform approach 1, particularly at a higher number of iterations. This is mainly because approach 1 uses membrane fit, and this will inevitably introduce interpolation errors. However, as shown in Fig. 4, approach 1 performs better than approach 2 in the first iteration. This is to a large extent due to the discontinuity and unevenness of the object surfaces. For approach 1, the membrane fit makes the surfaces smooth, which is helpful for the first iteration (Horn and Harris algorithm). Note that the Horn and Harris algorithm assumes that most surfaces in the scene are sufficiently smooth so that the estimated derivatives are reliable.

Thus, there are two factors that affect the results of motion estimation: One is the introduction of errors in the motion-estimation process, and the other one is the discontinuity and unevenness of surfaces considered in motion-estimation algorithms. From the MSE and MVE curves, it can be seen that the MSE and MVE are not always in accord. Sometimes, low MSE does not necessarily mean low MVE, and vice versa. This is because of the sampling error introduced in the transformation from the OOGL file to RIF file, especially on the border of the objects. For instance, some border points on the second image cannot be reconstructed from the first image. As a result, these points will affect the computation of MSE.

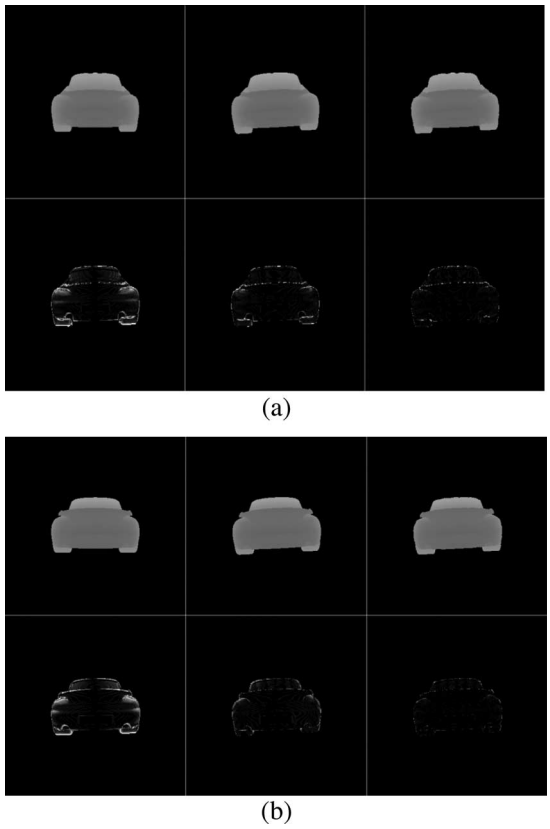


Fig. 11. Subjective comparison of different algorithms for the back of Porsche with $C = 0.05$. (a) Approach 1. (b) Approach 2. From left to right then top to bottom. (1) First image. (2) Second image. (3) Estimated second image using the motion parameters of the final iteration. (4) Difference image between the original two images. (5) Difference image between the second image and the estimated second image (DFD) using the estimated-motion parameters of the first iteration (Horn's algorithm). (6) Difference image between the second image and the estimated second image using the motion parameters of the final iteration.

For the second scenario, we use the back of Auto, where the second image is generated with a different translational movement (i.e., $U = 1$, $V = -0.5$, $W = -1$). The results of subjective and objective evaluations are shown in Figs. 5 and 6, respectively. In this case, we clearly see a significant improvement of the estimation scheme as the number of iterations increase. We can also observe that with approach 2, the best results can be reached with much fewer iterations. For the third and fourth scenarios, we run similar experiments but using the Porsche test image instead of the Auto image. The results, which are depicted in Figs. 7–10, verify the consistency in the estimation accuracy, particularly with the help of approach 2.

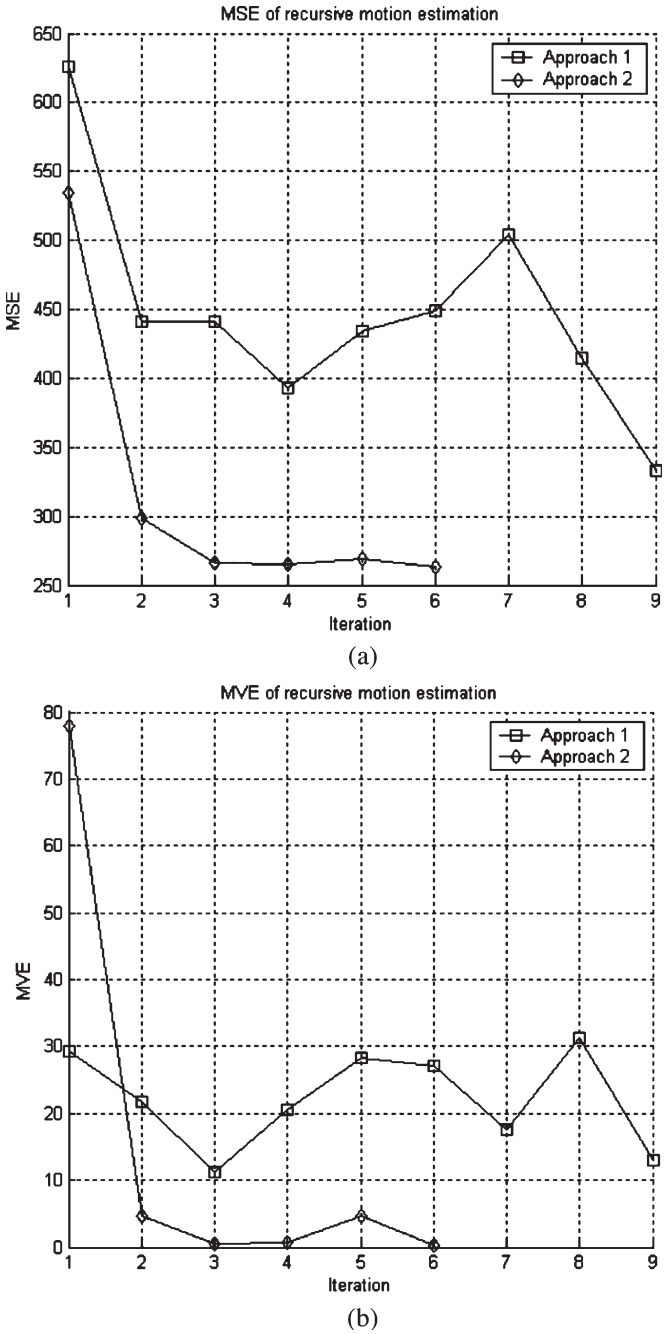


Fig. 12. Objective comparison of different algorithms for the back of Porsche with $C = 0.05$. (a) MSE. (b) MVE.

For the fifth scenario, we evaluate the performance of the estimation technique with respect to simple rotational movement in the XY plane ($C = 0.05$). Results shown in Figs. 11 and 12 clearly demonstrate that with approach 2, we can almost achieve near-perfect estimation [see Fig. 12(b)].

In our final scenario, all six components of the 3-D motion vector ($U = 1$, $V = -0.5$, $W = 1$, $A = -0.02$, $B = -0.02$, and $C = 0.02$) have been used to displace the object (Porsche) in the image. In addition, in order to assess the performance of the motion estimation, we deliberately corrupted the second range image with zero-mean additive Gaussian noise. Different levels of noise, as described by its standard deviation, are added to the range component ρ in the spherical coordinate (before transformation to the CEM coordinate). We then averaged the motion-estimation results by running each test 50 times. The results, which are depicted in Fig. 13 for the first and second approaches, show how effectively the motion estimation converges under more complex motion displacements at different noise levels. We can also observe that the performance superiority of approach 2 over approach 1 becomes more distinct at higher noise levels. Nevertheless, as can be expected, the estimation performance deteriorates with the increasing noise level. Note that in these experiments, we have not used any filtering to reduce the effect of Gaussian noise.

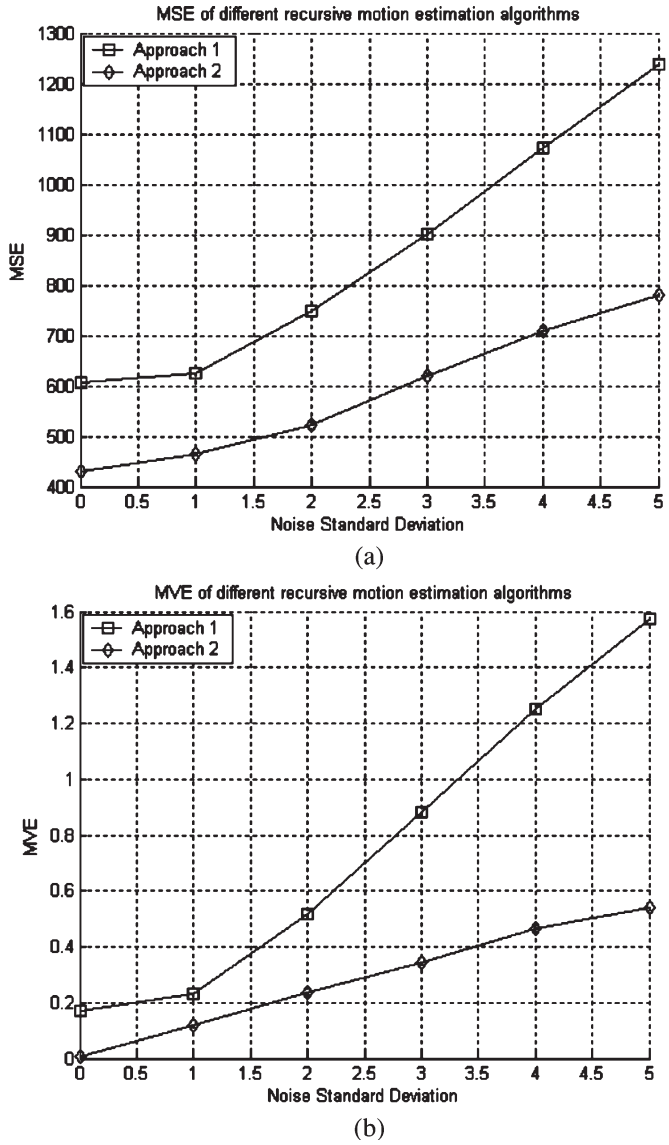


Fig. 13. Objective comparison of different algorithms for Porsche with different levels of noise. (a) MSE. (b) MVE.

Finally, we should point out that in the case of the first approach, which is based on membrane fit, the computational complexity (at every iteration) would almost be the same as that with Horn's algorithm. Obviously, a higher number of iterations would require a longer delay. The second approach, however, needs additional processing in each iteration, which includes the transformation between (ρ, θ, ϕ) and (X, Y, Z).

IV. Conclusion

Advanced vehicle-based safety and warning systems use 3-D sensors (radar, laser scanners, and stereo) and cameras to measure road geometry (position and curvature) and range to obstacles in order to warn a driver of an impending crash and/or to activate safety devices (air bags, brakes, and steering). In addition, recovering scene structure and camera motion from image sequences of rigid motion has been an important topic in computer vision and robotic vehicle applications. It is against this very important and challenging backdrop that we introduced a motion-estimation approach from motion formulations that yield stable and accurate estimation framework. We have shown that displacements of objects with complex 3-D motion in moving-range images can be accurately estimated by using the recursive approach presented in this paper. To further improve the estimation accuracy, we have also developed a method to reconstruct the range image on a nonrectangular grid that is typical of range images in the Cartesian coordinate system.

Acknowledgment

The authors would like to thank Dr. B. Leibe for his help in developing the 3-D software for generating moving-range video sequences.

References

[1] Z. Kim and T. Cohn, "Pseudoreal-time activity detection for railroad grade-crossing safety," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 319–324, Dec. 2004.
[2] G. Lefaix, T. Marchand, and P. Bouthemy, "Motion-based obstacle detection and tracking for car driving assistance," in *Proc. ICPR*, 2002, vol. 4, pp. 74–77.
[3] N. Ancona, G. Creanza, D. Fiore, R. Tangorra, B. Dierickx, G. Meynants, and D. Scheffer, "Real-time, miniaturized optical sensor for motion estimation and time-to-crash detection," in *Proc. SPIE—Adv. Focal Plane Arrays and Electron. Cameras*, 1996, vol. 2950, pp. 75–85.
[4] T. Camus, "Calculating time-to-collision with real-time optical flow," in *Proc. SPIE—VCIP*, 1994, vol. 2308, pp. 661–670.
[5] B. K. P. Horn and J. Harris, "Rigid body motion from range image sequences," *CVGIP, Image Underst.*, vol. 53, no. 1, pp. 1–13, Jan. 1991.
[6] K. Chaudhury, R. Mehrotra, and C. Srinivasan, "Detecting 3-D motion field from range image sequences," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 2, pp. 308–314, Apr. 1999.
[7] M. Harville, A. Rahimi, T. Darrell, G. Gordon, and J. Woodfill, "3-D pose tracking with linear depth and brightness constraints," in *Proc. ICCV*, 1999, pp. 206–213.
[8] Y. Liu and M. A. Rodrigues, "Correspondence less motion estimation from range images," in *Proc. ICCV*, 1999, pp. 654–659.
[9] L. Luchese, G. Doretto, and G. M. Cortelazzo, "Frequency domain estimation of 3-D rigid motion based on range and intensity data," in *Proc. Int. Conf. Recent Adv. 3-D Digital Imag. and Model.*, 1997, pp. 107–112.
[10] R. Szeliski, "Estimating motion from sparse range data without correspondence," in *Proc. ICCV*, 1988, pp. 207–216.
[11] J. Kolodko and L. Vlacic, "Fusion of range and vision for real-time motion estimation," in *Proc. IEEE Int. Conf. Intell. Veh. Symp.*, 2004, pp. 14–17.
[12] A. Hoover, D. Goldgof, and K. Bowyer, "Egomotion estimation of a range camera using the space envelope," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 4, pp. 717–721, Aug. 2003.
[13] K. Arun, T. Huang, and S. Blostein, "Least square fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, 1987.
[14] B. Sabata and J. K. Agarwal, "Estimation of motion from a pair of range images," *CVGIP, Image Underst.*, vol. 54, no. 3, pp. 309–324, Nov. 1991.
[15] M. Yamamoto, P. Boulanger, J. Beraldin, and M. Rioux, "Direct estimation of range flow on deformable shape from a video rate range camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 1, pp. 82–89, Jan. 1993.
[16] L. Tsap, D. Goldgof, and S. Sarkar, "Model-based force-driven nonrigid motion recovery from sequences of range images without point correspondences," *Image Vis. Comput.*, vol. 17, no. 14, pp. 997–1007, Nov. 1999.
[17] C. Kambhamettu, D. Goldgof, M. He, and P. Laskov, "3-D nonrigid motion analysis under small deformations," *Image Vis. Comput.*, vol. 21, no. 3, pp. 229–245, Mar. 2003.
[18] H. Spies, B. Jahne, and J. Barron, "Range flow estimation," *Comput. Vis. Image Underst.*, vol. 85, no. 3, pp. 209–231, Mar. 2002.
[19] G. Hetzel, B. Leibe, P. Levi, and B. Schiele, "3-D object recognition from range images using local feature histograms," in *Proc. CVPR*, 2001, vol. 2, pp. 394–399.
[20] H. Spies, "Analysing Dynamic Processes in Range Data Sequences," Ph.D. dissertation, Univ. Heidelberg, Heidelberg, Germany, 2001.
[21] H. Gharavi and H. Reza-Alikhani, "Pel-recursive motion estimation algorithm," *Electron. Lett.*, vol. 37, no. 21, pp. 1285–1286, Oct. 2001.

Hamid Gharavi (F'92) received the Ph.D. degree from Loughborough University, Loughborough, U.K., in 1980.

He joined AT&T Bell Laboratories, Holmdel, NJ, in 1982. He was then transferred to Bell Communications Research (Bellcore) after the AT&T-Bell divestiture, where he became a Consultant on video technology and a Distinguished Member of Research Staff. In 1993, he joined Loughborough University as a Professor and the Chair of communication engineering. Since September 1998, he has been with the National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD. He was a core member of the Study Group XV (Specialist Group on Coding for Visual Telephony) of the International Communications Standardization Body CCITT (ITU-T). He was selected as one of the six university academics to be appointed to the U.K. Government's Technology Foresight Panel in Communications to consider the future through 2015 and make recommendations for allocation of key research funds. He is the holder of eight U.S. patents. His research interests include video/image transmission, three-dimensional motion estimation, wireless multimedia, mobile communications and third-generation wireless systems, and mobile *ad hoc* networks.

Dr. Gharavi was a recipient of the Charles Babbage Premium Award of the Institute of Electronics and Radio Engineering, in 1986 and the IEEE CAS Society Darlington Best Paper Award, in 1989. He is a Distinguished Lecturer of the IEEE Communication Society. He has been a Guest Editor for a number of special issues. He is the Deputy Editor-in-Chief of the IEEE Transactions on Circuits and Systems. Since January 2003, he has been a Member of the Editorial Board of the Proceedings of the IEEE.

Shaoshuai Gao was born in Shandong Province, China, in 1976. He received the B.S. degree from Tianjin University, Tianjin, China, in 1998 and the Ph.D. degree from the Graduate School of Chinese Academy of Sciences, Beijing, China, in 2003.

From 2003 to 2004, he was a Research Fellow with the Nanyang Technological University, Singapore. Since September 2004, he has been a Guest Researcher with the National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD. His research interests include error resilient video coding, three-dimensional image processing, and range flow.