3-D Model-Based Vehicle Tracking

Jianguang Lou, Tieniu Tan, *Fellow, IEEE*, Weiming Hu, Hao Yang, and Steven J. Maybank, *Member, IEEE*

***Abstract*—This paper aims at tracking vehicles from monocular intensity image sequences and presents an efficient and robust approach to three-dimensional (3-D) model-based vehicle tracking. Under the weak perspective assumption and the ground-plane constraint, the movements of model projection in the two-dimensional image plane can be decomposed into two motions: translation and rotation. They are the results of the corresponding movements of 3-D translation on the ground plane (GP) and rotation around the normal of the GP, which can be determined separately. A new metric based on point-to-line segment distance is proposed to evaluate the similarity between an image region and an instantiation of a 3-D vehicle model under a given pose. Based on this, we provide an efficient pose refinement method to refine the vehicle's pose parameters. An improved EKF is also proposed to track and to predict vehicle motion with a precise kinematics model. Experimental results with both indoor and outdoor data show that the algorithm obtains desirable performance even under severe occlusion and clutter.**

***Index Terms*—Model-based vision, occlusion reasoning, pose refinement, tracking filter, traffic surveillance.**

I. Introduction

In THE LAST two decades, visual traffic surveillance has attracted more and more interest in the area of computer vision because of its tremendous application prospect. Efficient and robust vehicle localization and tracking from monocular intensity image sequences play a key role in traffic surveillance, and can provide intermediate results to derive semantic descriptions for moving vehicles in traffic scenes, such as "Car no. 2 turned left at the junction," "The white car stopped in the park," etc. Based on these descriptions, not only can routine management be performed, but also abnormal events can be detected and control decisions be made in real time.

In the tracking and localization process, experience suggests that better performance can be obtained by exploiting task-based a *priori* knowledge. In most traffic scenarios, the target objects are known and three-dimensional (3-D) geometric descriptions for these objects can be established in advance through measurement, CAD models or computer vision techniques (i.e., structure from motion [3]). In this paper, we concentrate on the 3-D model-based vehicle tracking problem and we assume that the geometric descriptions (for example, 3-D wireframe models) of vehicles have already been established. We present a 3-D wireframe model-based vehicle tracking approach. As in our previous work, the ground-plane constraint (GPC) [1] is exploited (that is, vehicles are constrained to be in contact with the ground plane (GP) under normal conditions). The number of degrees of freedom (dof) of vehicle pose is, thus, reduced from 6 to 3, which can be described by the location (X, Y) on the GP and the orientation (θ) about the normal of the GP.

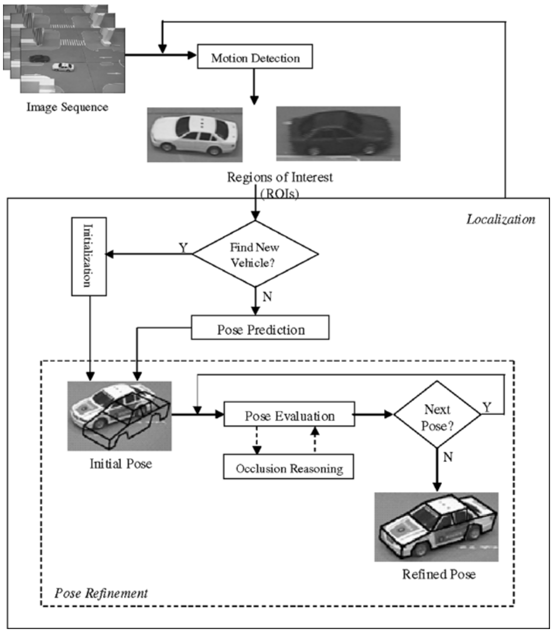

Fig. 1. Schematic diagram of a model-based vehicle tracking system.

The work described in this paper is part of a vision-based vehicle tracking system for automatic monitoring and surveillance of road traffic. A schematic diagram of the system is shown in Fig. 1. We assume that the camera is static and calibrated, and 3-D wireframe models of vehicles have already been established. In our system, image sequences captured from a CCTV camera are first fed into the motion detection module to identify image regions where significant motion occurs. These regions are called regions of interest (ROI) because they are likely to contain road vehicles. For each detected ROI in a specific frame, either the predictive tracking module or the pose initialization module is activated according to whether it is occurring for the first time. An initial pose for the vehicle in the ROI is generated in both cases, which is further refined by the pose refinement module to deliver the final result. This paper concentrates on the tracking module and the pose refinement algorithm (the small dotted block in Fig. 1), though advances in other parts of

Manuscript received June 11, 2003; revised September 1, 2004. This work was carried out at the NLPR. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bruno Carpentieri.

J. Lou is with Microsoft Research Asia, Beijing 100080, China (e-mail: t-jlou@microsoft.com).

T. Tan, W. Hu, and H. Yang are with the The National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China (e-mail: tnt@nlpr.ia.ac.cn; wmhu@nlpr.ia.ac.cn; hyang@nlpr.ia.ac.cn).

S. J. Maybank is with the Department of Computer Science, University of Reading, Reading, Berkshire RG6 6AY, U.K. (e-mail: s.j.maybank@reading.ac.uk).

Digital Object Identifier 10.1109/TIP.2005.854495

the system have also been made. Details related to the pose initialization algorithm can be found in our previous paper [1], [2].

The remainder of the paper is organized as follows. In Section II, we introduce some previous related approaches. A new pose evaluation function is defined in Section III. Section IV focuses on pose refinement, where we refine the pose parameters by an iterative processe. A new predictive tracking filter with a vehicle motion model is provided in Section V. In Section VI, we introduce a simple occlusion reasoning strategy and show how to actively crop the model projection to allow matching under occlusion. Experimental results and discussions are presented in Section VII.

II. Previous Related Work

The aim of model-based localization is to derive an object's 3-D pose by mapping image data to corresponding model descriptions. This involves issues such as how to choose appropriate features in the image and model domains, respectively, and how to match these features together. Previous work in this area includes graph matching [5], indexing and invariants [6], viewpoint consistency constraint [7], gradient-based approach [2], [8], [9], self-similarity [10], etc., which are based on matching sets of two-dimensional (2-D) image features (such as points, line segments, and conic sections) and sets of corresponding 3-D features. Optical flow has also been exploited [11].

In many approaches, it is necessary to establish the correspondence between the image features and the model features [12], which is far from trivial [1]. However, hypothesis driven approaches can successfully avoid the feature correspondence problem, as the matching can be implicitly determined as a byproduct of the hypothesis [13]. The approach presented in this paper is of this category.

How to generate, verify and refine a hypothesis (hypothetical pose) plays a key role here. In some previous work [1], [9], the pose verification process is related to the definition of a metric to measure the similarity between an image and a 3-D model instantiation.

Another issue in object tracking is occlusion. Model-based approaches essentially have an advantage in dealing with occlusion due to the explicit supervision of 3-D models, as shown in previous work [1], [9]. For real applications in traffic surveillance, it is inadequate. Thus, we introduce some special occlusion processing strategies as is done by Koller *et al.* in their 2-D contour tracking algorithm [14]. Frank *et al.* [15], [16] and Haag *et al.* [17] have also proposed some valuable ideas about the handling of occlusion. In this paper, we only adopt a simple but effective strategy for occlusion handling which avoids explicit scene modeling as required in [15]–[17].

The Kalman filter and the extended Kalman filter (EKF) are widely used in visual tracking systems [9], [14], [18]–[23], because they are linear recursive filters which can be implemented in real time, and under certain conditions the Kalman filter is optimal [24]. In [14], a visual vehicle tracking system using Kalman filter is discussed, and in Koller *et al.* [9], an iterated extended Kalman filter (IEKF) is used. Koller *et al.* [9] also adopt a simple model that assumes that the car carries out a circular motion with constant translational and constant angular velocity. However, the EKF needs a precise dynamic model and prior knowledge about the statistical characteristics of the measurement noise. Therefore, as claimed by Maybank *et al.* [25], [26], the traditional EKF does not perform well when the car carries out a complicated maneuver (in passing, one should also note that Haag *et al.* [27] argued that IEKF can obtain reasonably accurate predictions under a high sampling rate such as 50 half-frames per second, but, in fact, most existing 3-D visual tracking algorithms can not work at such a high speed because of their high computational cost; thus, we would rather follow the argument of Maybank *et al.*).

To overcome the deficiencies of the traditional EKF, Maybank *et al.* [25], [26] have proposed a covariance-updating filter (CUF) in which the mean and the covariance of the system states are propagated with errors of order $O(t^3)$. However, the CUF is still sensitive to the uncertainty of motion model and fails when the car carries out a complicated motion. In fact, the dynamic model changes over time, so a simple Brownian motion cannot easily model the change of steering and acceleration. Furthermore, the statistical characteristics of the noise are often unknown.

Monte Carlo filters [28] are also widely adopted in this area, such as the bootstrap filter [29], and condensation [30]. Because of the sampling required by Monte Carlo filters, their computational cost is much higher than a Kalman filter, which limits their application in real-time 3-D visual object tracking.

III. Pose Evaluation

For pose refinement, a pose evaluation function which measures the quality of a given pose must be defined. In this section, we introduce a simple pose evaluation function based on all the edge points in the ROI extracted by the Canny operator.

A. Point-to-Line Segment (PLS) Distance

Given a point a and a line segment l with endpoints b_1 and b_2, let a' be the orthogonal projection of a onto the line through b_1 and b_2. The PLS distance between a and l is defined as (see Fig. 2) (1), shown at the bottom of the page.

The PLS distance has an explicit physical meaning in that it is the minimum distance that a must move in order to exactly lie on l.

B. Pose Evaluation Function

Based on the PLS distance, we can define a metric to evaluate the similarity between an image region and a model projection

$$D(a,l)=\begin{cases}|\vec{aa'}|, & \text{if the projection } a' \text{ of } a \text{ lies between } b_1 \text{ and } b_2\\ \min\limits_{b\in\{b_1,b_2\}}|\vec{ab}|, & \text{otherwise}\end{cases} \quad (1)$$

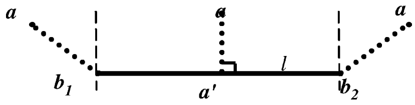

Fig. 2. PLS distance between a point and a line segment.

and, hence, the quality of the corresponding pose. Given an image region with edge point set $I = \{I_{j,k}\}$ and a model projection under a given pose with 2-D line segment set $L = \{L_p | 1 \leq p \leq N\}$, the pose evaluation function is defined as

$$H(I, L) = \sum_{j,k} \min_p((\vec{g}_{j,k} \cdot \vec{n}_p) \cdot D(I_{j,k}, L_p))^2 \qquad (2)$$

where $D(I_{j,k}, L_p)$ is the PLS distance defined in Fig. 1, $\vec{g}_{j,k}$ denotes the unit image gradient vector at point $I_{j,k}$ (In the implementation of Canny operator, $\vec{g}_{j,k}$ can be simultaneously estimated.) and $\vec{n}_p$ denotes the unit normal of the line segment L_p.

Careful readers may find that the basic idea here is somewhat similar to the Chamfer distance [31] in that it utilizes the form of quadratic sum to improve its robustness to noise compared to the traditional Hausdorff distance [32], [33]. Another similar method is the iterative closest point (ICP) algorithm [34] which uses closest point pairs to measure the similarity between two point sets. All of these distances are some formulations of topological distance between two point sets.

This evaluation function can effectively evaluate the similarity between an image region and a model projection. Fig. 3 shows an example of the evaluation function, where (a) is an image containing ROI and (b) the surface of the pose evaluation function around the model instantiation in (a), where the evaluation function is reversely normalized to 0–1. It is obvious that the surface of such an evaluation function is smooth and the peak of the function is very conspicuous, indicating that the evaluation function has desirable properties for optimization. A comparison with the previous evaluation function from Reading can be found in [35].

IV. Pose Refinement

A. Decomposition of the Pose Refinement Process

The aim of the pose refinement process is to start from an initial pose and refine it gradually to approximate the correct pose of the target. Therefore, this process can be considered as a series of pose evolutions, which correspond to a series of virtual motions of the vehicle from the initial pose to the refined pose in the 3-D space. An arbitrary 3-D motion of a rigid object can be separated into two independent motions: translation of the centroid and rotation around the centroid.

Under the GPC, the translation is limited to be parallel to the GP, and the rotation is about the vehicle's central axis perpendicular to the GP. Under the weak perspective assumption (which is valid in most traffic scenes because the camera is relatively far away from the targets), the problem, thus, becomes one of finding parameters for two kinds of motions (translation and rotation) to arrive at a best match between the image region and the model projection, and can be resolved alternatively.

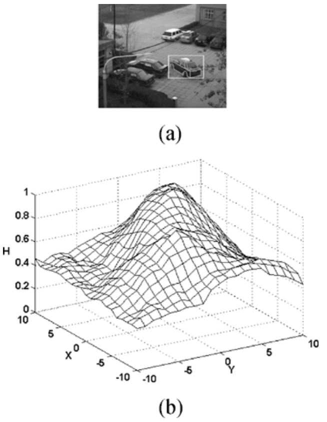

Fig. 3. Illustration of pose evaluation function. (a) An image containing ROI. (b) Surface of the evaluation function.

B. Determination of Translation Parameters

When the model projection translates on the 2-D image plane by an amount of (u, v), the new model projection line segment set is $L' = \{L'_p | 1 \leq p \leq N\}$, where L'_p is the counterpart of L_p after translation. If L_p is represented by the equation $a_p j - k + b_p = 0$, the translated version becomes L'_p: $a_p j - k + (b_p - a_p u + v) = 0$. The pose evaluation function between the image region I and the translated model projection L' is

$$H(I, L') = \sum_{j,k} \min_p((\vec{g}_{j,k} \cdot \vec{n}_p) \cdot D(I_{j,k}, L'_p))^2. \qquad (3)$$

The determination of translation parameters is equivalent to finding a translation vector (u, v) in the image plane to optimize the pose evaluation function $H(I, L')$ defined in (3) to reach its minimum. For each line segment L_p, we define a subset of I as

$$I_p = \{I_{j,k} \in I | D(I_{j,k}, L_p) = \min_n D(I_{j,k}, L_n)\}. \qquad (4)$$

For the sake of simplicity, we assume that $I_p = I'_p$ (in reality, the assumption does not strictly hold, but experimental results show that even if some points are misclassified after translation, the result will not be impaired severely). By introducing this assumption to fix the point classification results caused by the $\min$ function, we can easily obtain the analytic form of the derivative of the pose evaluation function. In fact, we can see that (3) yields a quadratic function $H(I, L') = a_1 u^2 + a_2 v^2 + 2a_3 uv + 2a_4 u + 2a_5 v + a_6$, where all the coefficients are independent of the value of (u, v).

The new pose evaluation function in (3) reaches its minimum value when its partial derivatives are zero. Finally, we obtain the closed-form solution as follows:

$$u = \frac{(a_2 a_4 - a_3 a_5)}{(a_3^2 - a_1 a_2)}, \; v = \frac{(a_1 a_5 - a_3 a_4)}{(a_3^2 - a_1 a_2)}. \qquad (5)$$

Multiple iterations of the above algorithm can be performed to compensate for the side effects of the assumption $I_p = I'_p$ and improve the precision of the solution. As the resultant pose is drawn closer to the correct pose, the misclassification of points in I'_p caused by this assumption will gradually diminish as seen in the experimental results. Once u and v are obtained, the translation parameters can easily be determined using the GPC and known camera parameters. Like similar existing methods [36], it is very hard to prove in theory that such an iterative scheme is guaranteed to converge to the global optimal point. However, experiments show that this scheme can often obtain very good solution only after 3–5 iterations.

C. Determination of Rotation Parameter

1) Geometry-Based Method: We first give the definitions of some coordinate systems that are used. There are three 3-D coordinate systems in the following analysis: model coordinate system ($\mathcal{M}$), world coordinate system ($\mathcal{W}$), and camera coordinate system ($\mathcal{C}$).

The transform between $\mathcal{W}$ and $\mathcal{M}$ consists of rotation matrix $\mathbf{R}_1$ and translation vector $\mathbf{t}_1$ which is obtained from the process described in Section IV-B. The transform between $\mathcal{C}$ and $\mathcal{W}$ consists of rotation matrix $\mathbf{R}_2$ and translation vector $\mathbf{t}_2$ which are determined by camera calibration. The composite transform between $\mathcal{C}$ and $\mathcal{M}$ consists of rotation matrix $\mathbf{R} = \mathbf{R}_2\mathbf{R}_1$ and translation vector $\mathbf{t} = \mathbf{R}_2\mathbf{t}_1 + \mathbf{t}_2$.

We define the origin in $\mathcal{M}$ to be the vehicle's centroid and the XOY plane in $\mathcal{M}$ to be parallel to the GP. Therefore, the rotation axis becomes the Z axis in $\mathcal{M}$. To determine the rotation parameter $\Delta\theta$, we try to establish some correspondences between angles in the image plane and angles in the vehicle model. If the orientation of the tracked vehicle at current and last image frame are marked as θ_b and θ_a, respectively, the rotation parameter $\Delta\theta$ can be defined as the difference between θ_b and $\theta_b(\Delta\theta = \theta_b - \theta_a)$. When the vehicle rotates by angle $\Delta\theta$, the projection of a known angle in the vehicle model will vary with $\Delta\varphi$. This variation provides enough information to compute the rotation angle $\Delta\theta$.

For an arbitrary point $P_{\mathcal{M}}(X_{\mathcal{M}}, Y_{\mathcal{M}}, Z_{\mathcal{M}})$ in the model coordinate system, the change in its 3-D world coordinates due to rotation $\Delta\theta$ is given by

$$\begin{bmatrix} \Delta X_{\mathcal{W}} \\ \Delta Y_{\mathcal{W}} \\ \Delta Z_{\mathcal{W}} \end{bmatrix} = R_1 \begin{bmatrix} \cos\Delta\theta - 1 & -\sin\Delta\theta & 0 \\ \sin\Delta\theta & \cos\Delta\theta - 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X_{\mathcal{M}} \\ Y_{\mathcal{M}} \\ Z_{\mathcal{M}} \end{bmatrix} \tag{6}$$

where R_1 depends on θ_a which has been determined at last frame.

The weak perspective projection equation is

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \alpha' & 0 & 0 \\ 0 & \beta' & 0 \end{bmatrix} \left(R_2 \begin{bmatrix} X_{\mathcal{W}} \\ Y_{\mathcal{W}} \\ Z_{\mathcal{W}} \end{bmatrix} + t_2 \right) + \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} \tag{7}$$

where u_0 and v_0 are the intrinsic parameters of the camera. For a weak perspective projection, we assume that α' and β' are constants.

Combining (6) and (7), we can obtain the change in the 2-D image coordinates of point $P_{\mathcal{M}}$ on the image plane

$$\begin{aligned} \Delta u =& \alpha'((r_{12}X_{\mathcal{M}} - r_{11}Y_{\mathcal{M}})\sin\Delta\theta \\ &+ (r_{11}X_{\mathcal{M}} + r_{12}Y_{\mathcal{M}})(\cos\Delta\theta - 1)) \qquad (8) \\ \Delta v =& \alpha'((r_{22}X_{\mathcal{M}} - r_{21}Y_{\mathcal{M}})\sin\Delta\theta \\ &+ (r_{21}X_{\mathcal{M}} + r_{22}Y_{\mathcal{M}})(\cos\Delta\theta - 1)) \qquad (9) \end{aligned}$$

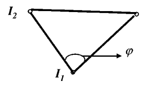

Fig. 4. Angle defined by three points on the image plane.

where r_{11}, r_{12}, r_{21}, r_{22} are the coefficients of rotation matrix $\mathbf{R}$, Δu and Δv are the variation of point $P_{\mathcal{M}}$'s projection on the image plane due to rotation $\Delta\theta$.

Consider a set of three points $(X_{\mathcal{M}1}, Y_{\mathcal{M}1}, Z_{\mathcal{M}1})$, $(X_{\mathcal{M}2}, Y_{\mathcal{M}2}, Z_{\mathcal{M}2})$ and $(X_{\mathcal{M}3}, Y_{\mathcal{M}3}, Z_{\mathcal{M}3})$ whose supporting plane is nonparallel to the GP. Their projection points (u_1, v_1), (u_2, v_2) and (u_3, v_3) define an angle on the image plane (see Fig. 4)

$$\cos\varphi = \frac{\overrightarrow{(u_2-u_1, v_2-v_1)} \cdot \overrightarrow{(u_3-u_1, v_3-v_1)}}{\left|\overrightarrow{(u_2-u_1, v_2-v_1)}\right| \cdot \left|\overrightarrow{(u_3-u_1, v_3-v_1)}\right|}.$$

Let $c_i = \partial f/\partial u_i$, $d_i = \partial f/\partial v_i$. Then the differential of the function is

$$-\sin\varphi d\varphi = \sum_{i=1}^{3}(c_i du_i + d_i dv_i). \tag{10}$$

By substituting difference for differential, the above equation can be rewritten as

$$-\sin\varphi\Delta\varphi = \sum_{i=1}^{3}(c_i\Delta u_i + d_i\Delta v_i). \tag{11}$$

From (8) and (9), we can see that u_i and v_i (i = 1, 2, 3) only consist of linear combinations of $\sin\Delta\theta$ and $\cos\Delta\theta$. After dividing by the coefficient of $\cos\Delta\theta$, a very simple equation $\cos\Delta\theta + k\sin\Delta\theta = q$ is acquired. It is solvable if and only if $|q| \leq (1 + k^2)^{1/2}$. If it is solvable, its solution in $[-\pi/2, \pi/2]$ (the vehicle's rotation angle between two consecutive frames rarely exceed this range) is

$$\Delta\theta = \operatorname{sgn}(k) \cdot \left(\arcsin\frac{q}{\sqrt{1+k^2}} - \arcsin\frac{1}{\sqrt{1+k^2}} \right). \tag{12}$$

Since all the coefficients are independent of the angle $\Delta\theta$ and can be determined in advance, the solution here is also closed-form. The point (u, v) on the image plane is used in the procedure described above, but we do not need to know the correspondences between the image edge points and the model vertices, only the correspondences between image angles and model angles are needed.

From the above derivation, we find that in order to determine the rotation parameter, all the information that we need is the angle defined by three known points in the image plane; locations of these points are not utilized at all. There are many angles satisfying this condition in a ROI containing a vehicle, because all the vertices in the 3-D model are known. In the rest of this subsection, we discuss how to extract such an angle in the image.

We only consider the neighborhoods of vertices since we need not know their exact locations. For a specific vertex, we deem that the vertex neighborhood constraint (VNC) is valid, which is stated as follows: For a vertex Q that is visible from the viewpoint, if the set of image lines extracted in its neighborhood is denoted by L_I, the set of model projection lines starting from this vertex is denoted by L_M, there must exist a subset L_C of set

L_I which satisfies the following: 1) L_C has the same cardinality as L_M (cardinality is the number of elements in a set); 2) there is a one-to-one mapping $h : a \approx b \Leftrightarrow |a - b| < \delta$ between L_C and L_M, where a and b are the slope angle of elements in set L_C and L_M, respectively.

Based on the VNC, the extraction of angles defined by three known points in the image can be described as follows: line extraction is performed on the neighborhood (typically 16 × 16) of a specific vertex to determine set L_I; set L_M is obtained from model projection; subset L_C is chosen from L_I by establishing one-to-one mapping h between L_C and L_M. The angles constituted by every two lines in L_C are precisely what we need, because h describes the correspondences among lines in L_C and L_M.

In theory, one such angle is adequate to compute the rotation parameter and we do not need to perform this process on every angle in the 3-D model. Instead, we only focus on those angles whose vertices are incident to several long edges, such as the incident vertices of the roof and the front window, or the side vertices of the bonnet of a car.

2) Small-Range Search: Due to the presence of image noise, the edge points (u_1, v_1), (u_2, v_2) and (u_3, v_3) are often noisy, and the rotation angle recovered in Section IV-C.I may be noisy. Therefore, a small-range search of pose parameter $\Delta\theta$ (the orientation angle) may be introduced, if necessary, to further refine the pose by optimizing the pose evaluation function defined in (2) along the $\Delta\theta$ axis of the pose parameter space.

In addition, the translation parameters can be further refined by applying the process described in Section IV-B again after the rotation parameters have been estimated. When the initial pose is relatively far from the ground truth, the first time estimated translation parameters are often not perfect. In this case, this further refinement step would be necessary.

V. Vehicle Tracking

A. Motion Model

In general, the performance of tracking also depends on the structure of tracking filters that contain the dynamic model for the car motion. The more accurate the dynamic model, the better the performance of the tracker. In this subsection, we describe the motion model used in our system. Koller *et al.* [9] introduce a simple circular motion model. A more precise dynamic model is studied in Maybank *et al.* [26]. The car is controlled by the driver by varying the steering angle φ and changing the speed v. In this paper, we use a two-point-bicycle model to describe the motion of the car. As shown in Fig. 5, (x, y) is the position of the car's centroid on the GP, v is the translational speed of the rear wheel, θ is the orientation of the whole car, ψ is the steering angle of the front wheel's hub, and l is the wheelbase of the car (typically $l \approx 2.5$ m). In this model, it is assumed that the vehicle is rigid, so the distance between the front wheel and the rear wheel does not change over time. It is also assumed that the wheels of the car cannot slip sideways.

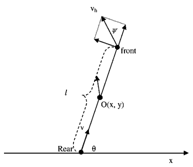

Fig. 5. Bicycle model of vehicle motion.

Therefore, the motion can be modeled by a dynamic process $X = [x, y, v, \theta, \psi]^T$, where X is the state vector and the state space is five-dimensional. We assume that the measurement noise e is white noise. The dynamic model of the motion can then be described as follows:

$$\begin{cases} \dot{x} = \frac{v}{2\cos\psi}\sqrt{1+3\cos^2\psi}\frac{2\cos\psi\cos\theta+\sin\psi\sin\theta}{\sqrt{\sin^2\psi+4\cos^2\psi}} \\ \dot{y} = \frac{v}{2\cos\psi}\sqrt{1+3\cos^2\psi}\frac{2\cos\psi\sin\theta+\sin\psi\cos\theta}{\sqrt{\sin^2\psi+4\cos^2\psi}} \\ \dot{v} = a \\ \dot{\theta} = \frac{v\cdot\tan\psi}{l} \\ \dot{\psi} = b \end{cases} . \tag{13}$$

In this dynamic model, a and b are used to describe the behavior of the driver. a reflects acceleration, braking, or change of gear; b represents the turn of the steering wheel. a and b are the parameters which depend on the driver's behavior and cannot be modeled easily. In our system, we do not explicitly model them, and only estimate them over time by introducing an orthogonality constraint which will be described in the next subsection.

B. Motion Tracking

Our filter is based on EKF. In order to improve the robustness of the filter against the model's inaccuracy, we modify the EKF by adding a new objective function [37]. The same objective function was used elsewhere by Zhou *et al.* [38] in the field of control system's diagnosis. The idea is that once the model has changed, the residual error series would change immediately, and then we adapt the filter to satisfy the orthogonality condition (just like white noise) in order that the filter's estimated states can track the system's real states quickly and accurately. If the model's parameters match the real system, the orthogonality condition will be self-satisfying for the EKF. We assume that the measurement noise is white noise, and if the parameters can be estimated properly, the residual error process should be white noise too. But if the model changes over time, the traditional EKF's residual errors do not satisfy the orthogonality constraint, and this indicates that the estimated parameters are not accurate. We adapt the filter to make sure that the residual error series has similar characteristics as white noise in order that the estimated states of the filter can track the system's states as quickly as system parameters change. For a real nonlinear system, the orthogonality condition can only be satisfied approximately.

We can adapt the filter to satisfy the orthogonality condition by adjusting the EKF's gain matrix on-line. This is achieved by using a fading parameter. The covariance matrix prediction is updated to

$$P_{k+1|k} = \lambda_{k+1} F_k(\hat{X}_{k|k}) P_{k|k} F_k^T(\hat{X}_{k|k}) + V_k \tag{14}$$

where λ is the fading parameter, P is covariance matrix, F is a discrete form of $f(X)$ and V_k is the covariance of the measurement error. The suboptimal estimation of λ is discussed in [39].

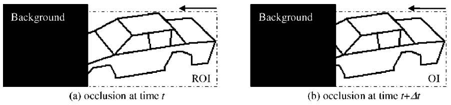

Fig. 6. Simple illustration of occlusion detection.

In fact, this means to find a tradeoff between the observation and the prediction of system model.

VI. Occlusion Reasoning

Up until now, we have described our pose refinement and tracking algorithm, but a practical visual tracking system should have certain ability to handle occlusion, because occlusions widely exist in the real world. Although model-based approaches are capable of dealing with occlusion due to the explicit supervision of models, most existing work focuses on the approaches' inherent robustness to the loss of image information caused by occlusion. We believe that a special occlusion reasoning strategy should be explicitly incorporated into the pose refinement algorithm to guarantee its performance under varying occlusion conditions.

In this paper, a simple occlusion reasoning strategy is presented which can actively crop the model projection to be matched when occlusion is detected. The basic idea is to determine a visible area for the model instantiation and only allow the model projection in the visible area to participate in matching. This can simply be performed by comparing the locations and sizes of ROIs in consecutive frames, see Fig. 6. In general, there is no need to constrain the visible area to accurately reflect the vehicle's real appearance under occlusion. It is sufficient to ensure that the marked visible area contains enough information for pose refinement. We will show examples in Section VII (Fig. 8) to demonstrate that this simple strategy is effective in handling occlusion.

In the future, we intend to develop a new and integrated framework to handle various occlusion situations.

VII. Experimental Results

Numerous experiments have been conducted and experimental results are presented in this section to demonstrate the performance of the proposed pose refinement and tracking algorithms. Motion detection is first performed on each frame to obtain ROIs that contain significant motion. Edge points in these ROIs are extracted using the Canny operator. Localization and tracking of vehicles are then performed based on these edge points. Due to space constraints, only a limited number of typical cases are given in the following.

1) Occluded Car: We also tested the proposed tracking algorithm with explicit occlusion reasoning on an image sequence that contains significant occlusion to demonstrate the algorithm's robustness to occlusion. The results obtained from this image sequence are shown in Fig. 7. The experimental results show that even if the vehicle is severely occluded, robust results can still be achieved due to the incorporation of the occlusion reasoning strategy. Special attention should be paid to the resultant pose from frame 120 to frame 200, where the toy car is severely occluded and occupies only a small region in the image, which constitutes a heavy challenge to the refinement algorithm. After frame 260, parts of the toy car are outside the field of view, but the tracker continues to work well.

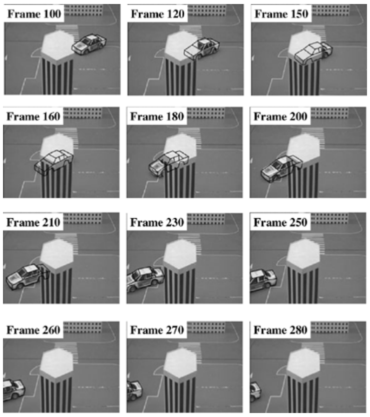

Fig. 7. Tracking results with significant occlusion.

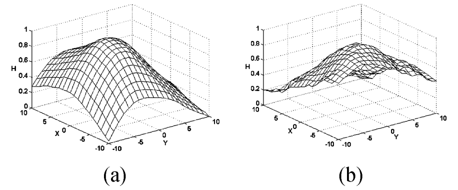

Fig. 8. Surfaces of evaluation functions for frame 200 in Fig. 7 where the vehicle is under occlusion. (a) The surface of evaluation function with explicit occlusion reasoning. (b) The surface of function without any occlusion reasoning.

The essential reason for the successful tracking at these frames lies in the successful use of occlusion reasoning. For example, in Fig. 8, (a) is the surface of evaluation function with explicit occlusion reasoning on Frame 200 in Fig. 7, and (b) is the surface of function without any occlusion reasoning. It is easy to see that the pose evaluation function with occlusion reasoning produces much smoother surface and more significant peak than the other one.

A. Real World Sequence

1) Small Viewing Angle Sequence: In order to further test our algorithm, several real world sequences are utilized. In this subsection, we present the tracking result in an image sequence of a parking area with a small viewing angle shown in Fig. 9. In this scene, a black saloon is tracked. Although the car is of distinct intensity from the background, the skeleton of the car is not well defined in the image because all parts of the car are black. As we know, a basic assumption behind 3-D wire-frame model-based algorithms is that vehicles can be represented by their skeletons. Thus, it is a big challenge for wire-frame model-based methods to track a car without a clear skeleton in an image sequence. In addition, perspective projection with small viewing angle will increase tracking difficulty for its inherent uncertainty [9]. Despite of these difficulties, we can still obtain accurate results as shown in Fig. 9. The resultant trajectory of the tracked vehicle on GP is shown in Fig. 10, where the variances of vehicle's pose parameters (x, y, θ) were estimated as $\sigma_x = 0.093$ m, $\sigma_y = 0.171$ m and $\sigma_\theta = 0.017$ rad, respectively, by our tracking filter. From this, we can see that the tracker can track the vehicle very stably under a small viewing angle.

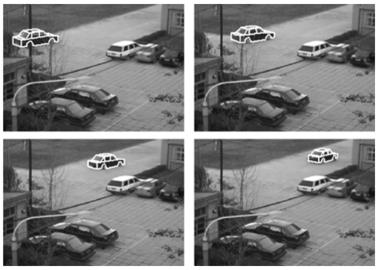

Fig. 9. Image sequence with small viewing angle.

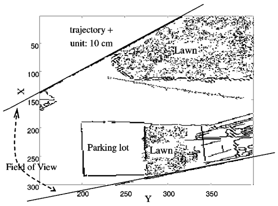

Fig. 10. Resultant trajectory of tracked vehicle in Fig. 9 as shown on GP.

2) Road Scenario: Besides the parking lot scenario described above, our algorithm is also widely tested in real world city roads. In the tested sequences, all typical vehicle behaviors in city roads have been covered. In Fig. 11, a digital camera is installed to watch a city road for a long time at frame rate of 15 frames/s. It is worthy to point out the results of frame #14 004 and frame #15 160 where the parts of the vehicles are partially overlapped by a tree. The branches of the tree often bring a lot of clutter edge points which is a big challenge for the tracking algorithms using edge points as its observation in the image. In frame #14 004 and frame #15160, our algorithm can work very well which depends on two aspects. One is the good prediction of our tracking algorithm which can reduce the possibility of falling into a local minimum, and the other is that most of the edge points still elicit positive values for the algorithm. The overall variances of the parameters (x, y, θ) are (0.53 m, 0.39 m, 0.05 rad) which is much larger than the small view angle sequence described in the above subsection. One of important factors which influence the performance is that the wire-frame model. On the city road, there are too many types of vehicle. Even for the saloon car, we can find more than ten types in the sequences. On the other hand, the number of our stored models is relatively very small. In practice, we apply one saloon model for all saloon cars. Sometime, the model can not precisely match the moving car which degrades the system's performance. The model acquirement will be discussed in the next subsection again as a weakness of 3-D model-based algorithms.

B. Discussion and Comparison

Noise and complex lighting condition are two key adverse factors in the performance of most vision algorithms. In this subsection, we will discuss the advantages and the limitations of our algorithm under different noise and lighting conditions.

Our method is based on a natural intuition that a vehicle can be represented by its skeleton, namely a wire-frame model, which has been widely used in our daily life. Based on this observation, the pose evaluation can be performed by matching the projection of the wire-frame model with the image edges in a ROI. It is well known that image edges are less sensitive to lighting changes than other image features. This characteristic guarantees the algorithm's robustness under different lighting conditions and different levels of image quality. In Fig. 12, an example is presented to support our claim where the quality of the image has been largely reduced by applying a heavy compression with JPEG. From Fig. 12(c), we can see that the algorithm can work very well under this situation because the edge points extracted in the ROI [Fig. 12(b)] can still provide enough information for our pose evaluation.

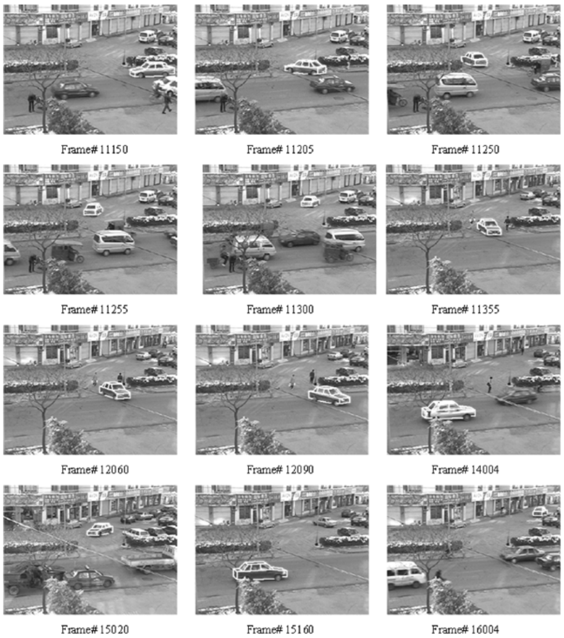

Fig. 11. Tracking result in a road scene.

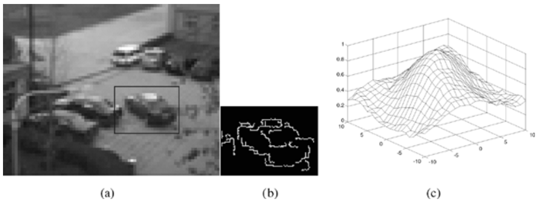

Fig. 12. Pose evaluation function for a car with low resolution image. (a) The low resolution image with a rectangular ROI, (b) edge points in the ROI, and (c) the surface of the pose evaluation function.

However, our wireframe model-based algorithm also brings some limitations. Structural clutters which are not distinct from vehicle skeletons will influence the performance of the algorithm. In the worst cases, these clutters can make the refine process fall into a local minimum. For example, in Fig. 13, we show a vehicle is occluded by serious clutters. Since the vehicle's skeleton is totally submerged by clutters, the pose evaluation fails. We can find that most of these clutters are part of the static background. Fortunately, in our algorithm, only the edges in the ROI are involved in the pose estimation, and a good ROI detection step can largely eliminate such clutters. In other words, we can determine whether each detected edge point belongs to the foreground or the background by exploiting the motion information, and the pose evaluation can be preformed

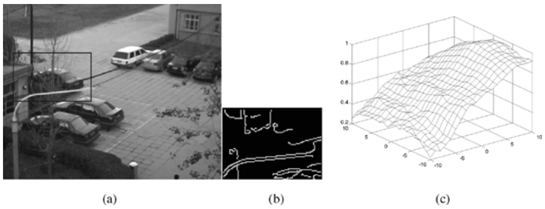

Fig. 13. Pose evaluation function for a car with serious clutters. (a) Car image with occlusion and clutters, (b) edge points in the ROI, and (c) the surface of the pose evaluation function.

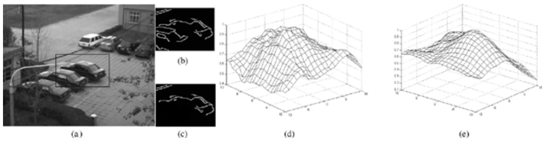

Fig. 14. Pose evaluation function for a car under occlusion. (a) Car image with occlusion, (b) edge points around the car, (c) foreground edge points, (d) the surface of the pose evaluation function based on edge points around the car, and (e) the surface of the pose evaluation function based on foreground edge points.

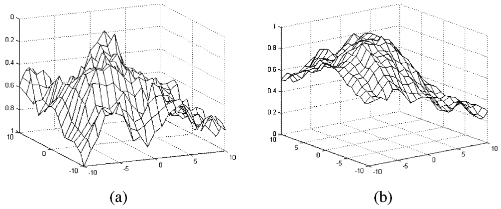

Fig. 15. Evaluation function surfaces for frame #14 004. (a) Surface of ICONIC function. (b) Surface of our proposed PLS function.

only by the foreground edge points. In Fig. 14(a), a moving car is occluded by a parked car, and the edge points around the moving car is shown in Fig. 14(b). If we use all these edge points to calculate the pose evaluation function, the pose evaluation function would contain some local minimums as shown in Fig. 14(d). On the other hand, Fig. 14(c) and (e) shows the foreground edge points and their result pose evaluation function which has a smooth surface and a significant peak.

Sometimes, a wire-frame model can not precisely depict an object (e.g., a modern roadster) whose surface is a very smooth curved face without any significant edges and corners. Our wireframe model-based method may not suitable for such kind of objects. In addition, the model acquirement for these smooth objects is also not a trivial task.

Comparing to other methods is always a good way to illustrate the strengths and weaknesses of an algorithm. Here, we compare our proposed PLS function to the evaluation function named ICONIC from Reading [1], [2] where the gradient information around the projection of wireframe model is utilized. Both methods are tested in the toy scene and real world scenarios under different conditions. First, we focus on the smoothness of surface, because it is always one of desirable properties for optimization. By analyzing the results, we found that the surfaces of PLS evaluation function are much smoother than the ICONIC. Fig. 15 shows a typical example where the surfaces are from evaluation of frame #14 004. The reason might be that the quadratic form of PLS makes it much smooth.

VIII. Conclusion

The work described in this paper forms part of a visual traffic surveillance system which is intended to smoothly track vehicles in traffic scenes and to analyze the behavior of vehicles. We have presented algorithms for two important issues of the system, namely pose refinement and vehicle tracking. Under the weak perspective and GPC assumption, we can refine translation parameters and rotation parameters, respectively. Closed-form solutions to both subproblems are obtained, thus reducing the computational cost of our model-based approach. We also improve the widely used EKF by adding an orthogonality condition to the estimated measurement errors. This reduces the sensitivity to the model uncertainty, and extends the car motion model for obtaining a good prediction performance in vehicle tracking. In addition, a simple partial match-based occlusion reasoning strategy is explicitly integrated in the pose refinement procedure, which aims to detect the occurrences of occlusion and actively modify the model to accommodate the resulting changes in vehicles' appearances. Experimental results have shown that that the presented algorithms work well even under partial occlusion.

References

[1] T. N. Tan, G. D. Sullivan, and K. D. Baker, "Model-based localization and recognition of road vehicles," *Int. J. Comput. Vis.*, vol. 27, no. 1, pp. 5–25, 1998.

[2] T. N. Tan and K. D. Baker, "Efficient image gradient based vehicle localization," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1343–1356, Nov. 2000.

[3] T. N. Tan, K. D. Baker, and G. D. Sullivan, "3-D structure and motion estimation from 2-D image sequences," *Image Vis. Comput.*, vol. 11, no. 4, pp. 203–210, 1993.

[4] H. Yang, J. G. Lou, H. Z. Sun, W. M. Hu, and T. N. Tan, "Efficient and robust vehicle localization," in *Proc. Int. Conf. Image Process.*, Sep. 2001, pp. 355–358.

[5] G. Kogut and M. Trivedi, "Efficient and robust vehicle localization," presented at the 9th World Congr. Intelligent Transport Systems, 2002.

[6] J. W. Lee, M. S. Kim, and I. S. Kweon, "A Kalman filter based visual tracking algorithm for an object moving in 3-D," in *Proc. Int. Conf. Intelligent Robots and Systems*, Sep. 1995, pp. 355–358.

[7] M. S. Costa and L. G. Shapiro, "3-D object recognition and pose with relational indexing," *Comput. Vis. Image Understand.*, vol. 79, no. 3, pp. 364–407, 2000.

[8] H. Kollnig and H. H. Nagel, "3-D pose estimation by directly matching polyhedral models to gray value gradients," *Int. J. Comput. Vis.*, vol. 23, no. 3, pp. 283–302, 1997.

[9] D. Koller, K. Daniilidis, and H. H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *Int. J. Comput. Vis.*, vol. 10, no. 3, pp. 257–281, 1993.

[10] R. Cutler and L. S. Davis, "Model-based object tracking in monocular image sequences of road traffic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 781–796, Aug. 2000.

[11] M. Haag and H. H. Nagel, "Combination of edge element and optical flow estimates for 3-D-model-based vehicle tracking in traffic image sequences," *Int. J. Comput. Vis.*, vol. 35, no. 3, pp. 295–319, 1999.

[12] W. M. Wells, "Statistical approaches to feature-based object recognition," *Int. J. Comput. Vis.*, vol. 21, no. 1, pp. 63–98, 1997.

[13] D. G. Lowe, "Robust model-based motion tracking through the integration of search and estimation," *Int. J. Comput. Vis.*, vol. 8, no. 2, pp. 113–122, 1992.

[14] D. Koller, J. Weber, and J. Malik, "Efficient and robust vehicle localization," in *Proc. 3rd Eur. Conf. Computer Vision*, Stockholm, Sweden, Sep. 1994, pp. 186–196.

[15] T. Frank, M. Haag, H. Kollnig, and H.-H. Nagel, "Characterization of occlusion situations occurring in real-world traffic scenes," in *Proc. ECCV Workshop on Conceptual Descriptions from Images*, Apr. 1996, pp. 43–57.

[16] ——, "Tracking of occluded vehicles in traffic scenes," in *Proc. 4th Eur. Conf. Computer Vision*, Apr. 1996, pp. 485–494.

[17] M. Haag, Th. Frank, H. Kollnig, and H.-H. Nagel, "Influence of an explicitly modeled 3-D scene on the tracking of partially occluded vehicles," *Comput. Vis. Image Understand.*, vol. 65, no. 2, pp. 206–225, 1997.

[18] I. J. Cox and S. L. Hingorani, "An efficient implementation and evaluation of reid's multiple hypothesis tracking algorithm for visual tracking," in *Proc. Int. Conf. Pattern Recognition*, Apr. 1994, pp. 437–442.

[19] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Who? when? where? what? A real time system for detecting and tracking people," in *Proc. 3rd Int. Conf. Face and Gesture Recognition*, Apr. 1998, pp. 222–227.

[20] A. Blake, M. Isanc, and D. Reynard, "Learning to track the visual motion of contours," *Artif. Intell.*, vol. 78, no. 2, pp. 101–133, 1995.

[21] R. Rosales and S. Sclaroff, "Improved tracking of multiple humans with trajectory prediction and occlusion modeling," in *Proc. CVPR Workshop Interpretation of Visual Motion*, Apr. 1998, pp. 437–442.

[22] C. Bregler, "Learning and recognizing human dynamics in video sequences," in *Proc. Int. Computer Vision and Pattern Recognition*, Apr. 1997, pp. 568–574.

[23] T. C. Robert, J. L. Alan, and K. Takeo, "A system for video surveillance and monitoring," in *Proc. American Nuclear Society (ANS) 8th Int. Topical Meeting on Robotics and Remote Systems*, Apr. 1999, pp. 568–574.

[24] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Second ed, R. M. Osgood Jr., Ed. Upper Saddle River, NJ: Prentice-Hall, 2000.

[25] S. J. Maybank, A. D. Worrall, and G. D. Sullivan, "Filter for car tracking based on acceleration and steering angle," in *Proc. Brit. Machine Vision Conf.*, Sep. 1996, pp. 615–624.

[26] ——, "A filter for visual tracking based on a stochastic model for driver behavior," in *Proc. 4th Eur. Conf. Computer Vision*, Apr. 1996, pp. 540–549.

[27] M. Haag and H. H. Nagel, "Tracking of complex driving manoeuvres in traffic image sequences," *Image Vis. Comput.*, vol. 16, no. 8, pp. 517–527, 1998.

[28] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamical systems," *J. Amer. Stat. Assoc.*, vol. 93, no. 8, pp. 1032–1044, 1998.

[29] N. J. Gordon, "A hybrid bootstrap filter for target tracking in clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 1, pp. 353–358, Jan. 1997.

[30] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. Eur. Conf. Computer Vision*, Apr. 1996, pp. 343–356.

[31] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 849–865, Jun. 1988.

[32] W. J. Rucklidge, "Efficiently locating objects using the Hausdorff distance," *Int. J. Comput. Vis.*, vol. 24, no. 3, pp. 251–270, 1997.

[33] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.

[34] P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[35] Q. F. Liu, J. G. Lou, W. Hu, and T. Tan, "Comparison of model-based pose evaluation algorithm in traffic scenes," in *Proc. 2nd Int. Conf. Image and Graphics*, Aug. 2002, pp. 343–356.

[36] G. D. Sullivan, "Visual interpretation of known objects in constrained scenes," *Phil. Trans. Roy. Soc.*, vol. B, no. 337, pp. 361–370, 1992.

[37] J. G. Lou, T. N. Tan, and W. M. Hu, "Visual vehicle tracking algorithm," *Electron. Lett.*, vol. 29, no. 18, pp. 1024–1026, 2002.

[38] D. H. Zhou, Y. G. Xi, and Z. J. Zhang, "A nonlinear adaptive fault detection filter," *Int. J. Syst. Sci.*, vol. 22, no. 12, pp. 2563–2571, 1991.

[39] J. G. Lou, H. Yang, W. M. Hu, and T. N. Tan, "Visual vehicle tracking using an improved EKF," in *Proc. 5th Asia Conf. Computer Vision*, Jan. 2002, pp. 296–301.

Jianguang Lou received the B.Sc. and M.Sc. degrees in automation from Zhejiang University, Hangzhou, China, in 1997 and 2000, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, in 2003.

He is an Associate Researcher at Microsoft Research Asia, Beijing. His main research interests include computer vision, human computer interaction, image processing, and pattern recognition.

Tieniu Tan (M'92–SM'97–F'03) received the B.Sc. degree in electronic engineering from Xian Jiaotong University, China, in 1984 and the M.Sc., DIC, and Ph.D. degrees in electronic engineering from the Imperial College of Science, Technology, and Medicine, London, U.K., in 1986 and 1989, respectively.

He joined the Computational Vision Group, The University of Reading, Reading, U.K., in October 1989, where he was a Research Fellow, Senior Research Fellow, and Lecturer. Currently, he is a Professor and the Director of the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing. He is an Associate Editor of *Pattern Recognition*. His current research interests include image processing, computer vision, pattern recognition, multimedia, and robotics.

Dr. Tan is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He was an Elected Member of the Executive Committee of the British Machine Vision Association and Society for Pattern Recognition (1996 to 1997) and is a Founding Co-Chair of the IEEE International Workshop on Visual Surveillance.

Weiming Hu received the Ph.D. degree from the Department of Computer Science and Engineering, Zhejiang University, Hangzhou, China.

From April 1998 to March 2000, he was a Postdoctoral Research Fellow at the Institute of Computer Science and Technology and Founder of the Research and Design Center, Peking University, Peking, China. Since April 1998, he has been with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, as an Associate Professor. His research interests include visual surveillance and monitoring of dynamic scenes, neural networks, and filtering of objectionable Internet images.

Hao Yang received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1998, and the M.Sc. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, in 2001.

Steve J. Maybank (M'97) received the Ph.D. degree in computer science from Birkbeck College, University of London, London, U.K., in 1988.

He joined the Pattern Recognition Group, Marconi Command and Control Systems, Frimley, U.K., in 1980 and then joined the GEC Hirst Research Centre, Wembley, U.K., in 1989. From 1993 to 1995, he was a Royal Society/EPSRC Industrial Fellow with the Department of Engineering Science, University of Oxford, Oxford, U.K. In 1995, he joined the University of Reading, Reading, U.K., as a Lecturer in the Department of Computer Science. In 2004, he became a Professor with the School of Computer Science and Information Systems, Birkbeck College. His research interests include the geometry of multiple images, camera calibration, visual surveillance, information geometry, and the applications of statistics to computer vision.