Stereo Visual Tracking Within Structured Environments for Measuring Vehicle Speed

Junda Zhu, *Student Member, IEEE,* Liang Yuan, *Member, IEEE,* Yuan F. Zheng, *Fellow, IEEE,* and Robert L. Ewing, *Senior Member, IEEE*

***Abstract*—We present a novel visual tracking method for measuring the speed of a moving vehicle within a structured environment using stationary stereo cameras. In the proposed method, visual stereo tracking and motion estimation in 3-D are integrated within the framework of particle filtering. The visual tracking processes in the two views are coupled with each other since they are dependent upon the same 3-D motion and correlated in the observations. Considering that the vehicle's motion is physically constrained by the environment, we further utilize the path constraint reconstructed from stereo views to reduce the uncertainty about the vehicle's motion and improve the accuracy for both tracking and speed measuring. The proposed method overcomes the challenges arising from the limitation of depth accuracy in a long-range stereo, and the experiments on the synthetic and real-world sequences have demonstrated its effectiveness and accuracy in both the tracking performance and the speed measurement.**

***Index Terms*—Object tracking, particle filtering, stereo vision, structured environments.**

I. Introduction

MEASURING the speed of a moving vehicle plays an important role in both civilian and military applications. Today, active sensor-based solutions, such as RADAR and LIDAR, are widely in use. Despite their popularity, there are two major issues in reality.

1) Detectable energy is emitted into the surrounding, which is not desirable for stealth operations.
2) Only the line-of-sight motion, i.e., the motion component along the incident direction affects the Doppler effect for RADAR and the time of flight for LIDAR. The speed measurement when the object is not moving directly toward the observer and/or in curved motion is not accurate.

With the recent advancement of vision systems and visual tracking technologies [1]–[8] (see the detailed review in the next section), vision-based solutions can now provide an important alternative to active sensors. A camera emits no energy and is not easily detectable by the target. A vehicle's 3-D location can be calculated given its corresponding projections in the views of calibrated stereo cameras. By continuously tracking the vehicle in stereo video streams, the speed of a moving vehicle can be measured.

Using stereo vision for speed measurement faces a great challenge, i.e., the depth accuracy is critical since the speed is the derivative of the depth with respect to time. Unfortunately, the depth error ϵ_z grows quadratically as the distance between the object and the cameras increases [9]

$$\epsilon_z \approx \frac{z^2}{bf} \cdot \epsilon_d$$

where z, b, f, ϵ_d are the distance, the stereo baseline, the focal length in pixels, and the pixel error, respectively. Since the baseline and focal length cannot be very large due to practical restrictions, the depth accuracy deteriorates at the far range and can be very noisy. Reducing the pixel error ϵ_d appears to be the only effort one can make to improve the depth accuracy, which has not been effectively studied before. Tracking corresponding feature points, applying triangulation to the points, and then filtering the noisy triangulation results are commonly done by the existing methods [1], [2], [10]–[12] (see the detailed review in the next section). There are two drawbacks associated with such methods. First, the stability of feature points on 3-D objects decreases significantly as the view angle changes [13]. The accuracy of feature point tracking is thus reduced when an object undergoes large rotation and translation in the 3-D. Second, the long-range stereo triangulation is subject to a range-dependent statistical bias, and directly applying filtering techniques to the 3-D coordinates obtained from stereo triangulation results in overestimated ranges [14].

In this paper, we propose a novel approach that integrates visual tracking and 3-D motion estimation under the framework of particle filtering [15], for accurate speed measurement with stereo vision. The cameras are assumed to be static in this paper. The proposed new method has a number of advantages. First, the capability of particle filtering for nonlinear models and non-Gaussian noises enables our stereo tracking method to handle nonlinearities arising from perspective projections and observations. As a result, accuracy and robustness of vehicle tracking in each of the stereo views are improved. Second,

Manuscript received September 16, 2011; revised December 5, 2011 and January 26, 2012; accepted February 9, 2012. Date of publication June 1, 2012; date of current version September 28, 2012. This paper was recommended by Associate Editor L. Zhang.

J. Zhu, L. Yuan, and Y. F. Zheng are with the Department of Electrical and Computer Engineering, Ohio State University, Columbus, OH 43210 USA (e-mail: zhuj@ece.osu.edu; yuanl@ece.osu.edu; zheng@ece.osu.edu).

R. L. Ewing is with the Air Force Research Laboratory, Wright-Patterson Air Force Base, Dayton, OH 45433 USA (e-mail: robert.ewing@wpafb.af.mil).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2012.2202074

both 2-D and 3-D information is included into the state space model to jointly estimate the 2-D tracks of the vehicle in each of the two views, as well as the 3-D location and motion. The 2-D visual tracking in the two views is now dependent upon the vehicle's 3-D motion, and is also correlated across the two views. This improves the tracking accuracy compared with conventional tracking methods using only one view, and does not require explicit stereo triangulation at each step after initialization since the correspondences between 2-D and 3-D locations are embedded in the state update. Finally, considering that the vehicle's movement is constrained by its path in reality, and the stereo vision enables us to extract the 3-D reconstruction of the path, we further include the path constraint into the particle filtering as the prior information about the vehicle's motion. This reduces the uncertainty about the vehicle's motion significantly as well as the dimension of the feasible state space, and therefore improves the accuracy of visual tracking and speed measurement.

The rest of this paper is organized as follows. Section II reviews previous works related to our work. Section III introduces our formulation of stereo tracking using particle filtering. Section IV studies the vehicle's 3-D motion under the path constraints for both straight and curved paths. Section V describes our tracking and speed measuring results on both synthetic and real-world sequences. Section VI concludes this paper.

II. Related Work

Estimating the 3-D motion of an object from stereo vision has been studied in the past [10]–[12], [16]. Reference [10] is one of the earliest works that addressed this problem. The motion parameters of a set of points are estimated from the noisy observations of their 3-D locations by an extended Kalman filter (EKF) or an iterated extended Kalman filter with a motion model assuming constant acceleration in translational motion and constant precession in rotational motion. Reference [11] studied the problem of computing the 3-D path of a moving rigid object by calibrated stereo vision. Feature points are detected and tracked in stereo views and their 3-D coordinates are calculated by linear triangulation. Reference [12] studied the equivalent problem of estimating the relative camera motion by tracking feature points in stereo views from a static scene. All the above methods rely on low level features, such as feature points, edges, and are targeted at close range applications.

Stereo motion estimation over long ranges has been studied in the intelligent transportation community for the applications in advanced driver assistance. In [1], the relative speed of an object with respect to the moving vehicle is estimated from stereo vision. The disparity and motion field are first obtained for each frame, and an EKF is applied to track the points over time and provide estimations about their velocity. The estimated velocities show large initial errors and slow convergence on simulation results, and the results on real sequences are not accurate. Barth and Franke [2] studied estimating the motions of oncoming vehicles using stereo vision. The point cloud associated with a vehicle is tracked using an extended Kalman filter to estimate the velocity and yaw rate of the vehicle. Their test results on synthetic data with a static ego-vehicle showed a total RMSE in the velocity of 2.2538 m/s for distance between 50 and 15 m and 0.4934 m/s thereafter, which are equivalent to about 5 miles per hour (mph) and 1.1 mph, respectively. The filtered depth appears to be overestimated. The results on real-world sequences taken from moving vehicles need the inputs from radar for initialization, and otherwise are slow in convergence and have large errors.

Our work is built upon the framework of particle filtering. Particle filtering, also known as sequential Monte Carlo methods, has received attention in the computer vision area in recent years [3]–[6], [17]–[20], since the seminal work of Isard and Blake [3] which first applied particle filtering to visual tracking. Only a few works [21]–[24] studied the stereo tracking using particle filtering. Reference [22] proposed a stereo tracking method with the stereo cameras mounted on a moving vehicle. Features are tracked using Lucas–Kanade optical flow, and their corresponding 3-D coordinates are calculated. Particle filtering is then applied to refine the results from feature point tracking, since the depth cannot be effectively estimated using particle filtering alone in their method. Quantitative results on the accuracy about the estimated motions are not shown. The paper in [24] proposed an adaptive multimodal stereo people tracking algorithm based on particle filtering. Depth information obtained from the stereo disparity map is used in addition to color and gradient cues for evaluating the particle weights. A person under tracking is modeled as two planar ellipsoidal surfaces, which correspond to the head and torso, respectively, and are assumed to be parallel to the camera. The method operates in real time and is able to track multiple people between a 0.5–6 m range with errors around 0.25 m.

Vehicle tracking requires vehicle detection to initialize the tracking process and/or recover from tracking failures. Vehicle detection is itself an active research area and there are a number of recent works that showed good vehicle detection results for different applications [7], [25], [26]. The paper in [25] proposed an approach for moving-vehicle detection and cast shadow removal in traffic monitoring videos using conditional random field. Spatial and temporal constraints are considered as contextual dependences for neighboring pixels, and each pixel in the scene is classified as belonging to a moving vehicle, cast shadow, or background based on the maximum *a posteriori* estimation. The paper in [7] presented a vehicle detection method based on dynamic background modeling. The challenges in vehicle detection, such as illumination changes and camera motions, are addressed by dynamically updating the background representation. Although in our current work we relied on manual input for detecting the vehicles in the first frames, existing vehicle detection algorithms as discussed above can be integrated for automated vehicle tracking and speed measuring.

Compared with the existing tracking literature, our proposed method differs in two important ways: 1) we integrate the visual tracking in stereo views and the motion estimation in 3-D within the particle filtering framework, and jointly estimate the object's 2-D tracks and 3-D motion, and 2) to improve the tracking performance and the speed measurement

accuracy for moving vehicles within structured environments, we utilize the path constraint reconstructed from stereo views to reduce the motion uncertainty and promote the accuracy.

III. Stereo Tracking With Particle Filtering

Measuring the speed of an object from stereo video is essentially an estimation problem in which the state of the object is estimated based on noisy video observations. The object state includes the information about the object position and motion, and is denoted by the state vector $\mathbf{x}_t$, where $t \in \mathbb{N}$ is the time index. The observation represents the available information from the video useful for estimating the state at time t and is represented by $\mathbf{z}_t$. The system is characterized by two models: the dynamics model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ and the observation model $p(\mathbf{z}_t|\mathbf{x}_t)$. From the Bayesian point of view, the problem is equivalent to constructing the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{0:t})$, where $\mathbf{z}_{0:t} = \{\mathbf{z}_0, \ldots, \mathbf{z}_{t-1}, \mathbf{z}_t\}$ denotes all the available observations up to time t. When both the dynamics and the observation models are linear and Gaussian, Kalman filtering provides the closed-form representation of the posterior distribution. However, in our application, both the dynamics and the observation models are nonlinear as can be seen in the following sections.

Particle filtering provides an approximate solution to the problem by recursively updating a discrete approximation to the posterior probability using weighted random samples, i.e., particles. The posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{0:t})$ is approximated by a set of N weighted random samples $\{\mathbf{x}_t^i, w_t^i\}_{i=1}^N$ with $\sum_{i=1}^N w_t^i = 1$, that is

$$p(\mathbf{x}_t|\mathbf{z}_{0:t}) \approx \sum_{i=1}^{N} w_t^i \cdot \delta(\mathbf{x}_t - \mathbf{x}_t^i) \tag{1}$$

where $\delta(\cdot)$ is the Dirac delta function. Random samples are drawn from the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)$. Since the optimal proposal distribution requires the inclusion of the current observation and is not always easy to be modeled and sampled from, suboptimal distributions are often used. The most common case is to use the dynamics model $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ [15], which is employed in this paper. In this case, the computation of the particle weights is directly based on the observation likelihood

$$w_t^i \propto w_{t-1}^i p(\mathbf{z}_t|\mathbf{x}_t^i). \tag{2}$$

The minimum mean-square-error estimation of the object state is the posterior mean, which is given by

$$\hat{\mathbf{x}}_t = \sum_{i=1}^{N} w_t^i \cdot \mathbf{x}_t^i. \tag{3}$$

Particles are resampled to avoid the degeneracy problem, and the weight calculation is further simplified to $w_t^i \propto p(\mathbf{z}_t|\mathbf{x}_t^i)$.

A. Dynamics Model for Stereo Tracking

The state vector $\mathbf{x}$ in particle filtering contains the information about the underlying dynamic process, and is recursively updated and estimated upon the arrival of new observations.

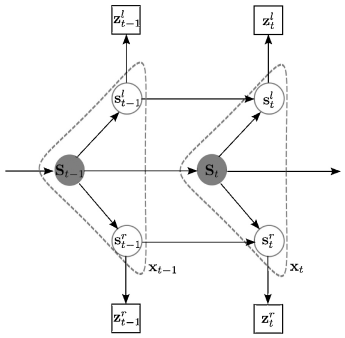


Fig. 1. Graphic model of the proposed algorithm.

For our problem, we construct the state vector $\mathbf{x}_t$ as a combination of 2-D and 3-D substates, i.e., $\mathbf{x}_t = \{\mathbf{s}_t^l, \mathbf{s}_t^r, \mathbf{S}_t\}$. $\mathbf{s}_t^{l,r}$ are the 2-D substates, which represent the 2-D tracks of the object of interest in the left and right views respectively. $\mathbf{S}_t$ is the 3-D substate, which contains the information about the object's location and motion in 3-D. Since we need to estimate the object's 3-D motion and measure its speed, the 3-D information is included into the filtering state and directly estimated in the filtering process, instead of being calculated from the stereo triangulation based on the independent 2-D tracking results from stereo views. This utilizes the prior information about the object's motion and also correlates the visual tracking processes in both views, as will be made clear in the following discussions. Introducing the 2-D substates provides us with the flexibility to track salient regions in each view, which may not exactly cover the same part of the object, and allows independent feature updates between consecutive frames in each view. Although one alternative way is to use only the 3-D state by estimating the object's width and height in the 3-D and by including the width and height into the state vector, we have chosen the 2-D–3-D state configuration because of two reasons.

1) Due to the inherent noise with long range stereo triangulation, the estimated dimensions of the object will have large uncertainty. Since the triangulated 3-D coordinates of the vertices have significant triangulation noises, and the estimates involve taking the difference between the noisy coordinates which are statistically independent, the uncertainty in the estimates of the width and height is large.
2) For a distant object, the detailed feature cannot be exactly matched in the videos due to the limitations in the resolution and quality of the image. Initial feature pairs may be projected onto different image locations as the object approaches, and cause mismatched regions in the stereo views. Although including 2-D substates increases the dimension of the state vector, it does not increase the searching space and computational complexity. The 2-D substates are dependent variables and are only propagated depending on the change of the 3-D substate, and no uncertainty and dynamics noise are added in the 2-D state update.

To prevent the stereo tracking from diverging, we employ the cross-likelihood between two views in the measurement model

to ensure that the corresponding regions of interests are being continuously tracked in both views, which will be discussed in detail in Section III-B. The graphical model of the state and the observation is shown in Fig. 1.

For our tracking approach, we choose a plane attached to the object of interest as the entity to be continuously tracked. Since the object under tracking is rigid, tracking the plane is equivalent to tracking the 3-D motion and pose change of the object itself. It overcomes the difficulty for reconstructing a full 3-D model of the object when it is far away, and also reduces the computational cost. For vehicle objects, the front of the vehicle is approximated by a plane and tracked by our algorithm. The plane is considered to be perpendicular to the motion direction. Note that although not all types of vehicles have a flat front, we consider that at extended distances, such geometric details cannot be distinguished and a plane can be used as a good approximation.

The 3-D state $\mathbf{S}_t$ accounts for the equation of the plane and its motion. Without loss of generality, the coordinate system associated with the left camera is chosen as the reference frame. At time t, the plane is parameterized by a 4-vector $\pi_t = [n_x, n_y, n_z, d]^\mathsf{T}$, where $\mathbf{n} = [n_x, n_y, n_z]^\mathsf{T}$ and d are the normal vector and the perpendicular distance to the origin, respectively. For any point P on the plane it satisfies the plane equation $\mathbf{n}^\mathsf{T} P = d$. As the object moves between times t and $t+1$, the plane undergoes the rotation $\Delta\mathbf{r}_t = [\phi_t, \theta_t, \psi_t]^\mathsf{T}$ and the translation $\Delta\mathbf{t}_t = [\Delta t_x, \Delta t_y, \Delta t_z]^\mathsf{T}$, where ϕ_t, θ_t, and ψ_t are the Euler angles of the rotation, and Δt_x, Δt_y, and Δt_z are the translations along x, y, and z axes, respectively. The 3×3 rotation matrix equivalent to $\Delta\mathbf{r}_t$ is represented by $\Delta\mathrm{R}_t$. The homogeneous transformation, including both the rotation and translation, is written as

$$\Delta\mathrm{T}_t = \begin{bmatrix} \Delta\mathrm{R}_t & \Delta\mathbf{t}_t \\ \mathbf{0} & 1 \end{bmatrix}.$$

The 2-D states $\mathbf{s}_t^{l,r}$ correspond to the projections of the planar regions in stereo views. We use quadrilaterals to delineate the regions of interest in each view, i.e., $\mathbf{s} = \{p_i^\mathsf{T}\}_{i=1,\ldots,4}$, where p_i is the ith vertex's coordinates and superscripts l, r are omitted. Note that we do not require the quadrilaterals in stereo views to cover exactly the same area on the tracked plane.

Consider that the plane moves from time t to $t+1$. The regions of interest in the stereo views undergo perspective transforms and we need to track them in order to estimate the plane's motion. The stereo cameras are calibrated with known intrinsic and extrinsic parameters. The intrinsic camera matrices for the left and right cameras are denoted as K^l and K^r, respectively, and the rotation matrix and translation vector between them are written as R and $\mathbf{t}$. The reference frame for the object motion is chosen to be identical to the coordinate system of the left camera, and therefore, for a given point $P = [X, Y, Z]^\mathsf{T}$ or its homogeneous form $\underline{P} = [X, Y, Z, 1]^\mathsf{T}$, it projects onto the left and right cameras by the camera projection equations

$$\begin{cases} \underline{p}^l = \mathrm{M}^l \underline{P} = \mathrm{K}^l[\mathrm{I} \quad \mathbf{0}]\underline{P} \\ \underline{p}^r = \mathrm{M}^r \underline{P} = \mathrm{K}^r[\mathrm{R} \quad \mathbf{t}]\underline{P} \end{cases} \tag{4}$$

where $\underline{p} = [x, y, 1]^\mathsf{T}$ denotes the homogeneous image coordinates, M is the 3×4 projection matrix, and K is the camera intrinsic matrix assuming zero skew [27].

For the left view, when the plane π_t rotates and translates by $\Delta\mathrm{R}_t$ and $\Delta\mathbf{t}_t$ from time t to $t+1$, its projection is subject to a perspective transform or homography H^l as follows:

$$\mathrm{H}^l = \mathrm{K}^l(\Delta\mathrm{R} - \frac{\Delta\mathbf{t}^\mathsf{T}\mathbf{n}}{d})\mathrm{K}^{l-1} \tag{5}$$

where the temporal index t is omitted from the subscripts for clarity. For the right view, the object's motion needs to be transformed into the coordinate system associated with the right camera before the homography can be derived. Given that the object is transformed by $\Delta\mathrm{T}_t$ from time t to $t+1$, or

$$\underline{P}_{t+1} = \Delta\mathrm{T}_t \underline{P}_t \tag{6}$$

and the rigid transformation between the left and right cameras as $\mathrm{T} = \begin{bmatrix} \mathrm{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}$ such that the homogeneous coordinates $\underline{P}_t^r$ with respect to the coordinate system of the right camera are

$$\underline{P}_t^r = \mathrm{T}\underline{P}_t \tag{7}$$

the object's motion with respect to the right camera can be derived as

$$\begin{aligned} \underline{P}_{t+1}^r &= \Delta\mathrm{T}_t^r \underline{P}_t^r \\ &= \mathrm{T}\ \Delta\mathrm{T}_t\ \mathrm{T}^{-1} \underline{P}_t^r. \end{aligned} \tag{8}$$

The rotation matrix $\Delta\mathrm{R}_t^r$ and the translation vector $\Delta\mathbf{t}_t^r$ with respect to the right camera can now be decomposed from the homogeneous transformation matrix $\Delta\mathrm{T}_t^r = \mathrm{T}\ \Delta\mathrm{T}_t\ \mathrm{T}^{-1}$, and the homography H^r for the right view is derived in the similar way as for (5).

Based on the above derivations, the dynamics model for stereo tracking can now be established in the following. For the 3-D substate $\mathbf{S}_t = \{\Delta\mathbf{r}_t, \Delta\mathbf{t}_t, \pi_t\}$, we assume a 3-D motion with constant rotational and translational velocity corrupted by additive Gaussian white noises, that is

$$\begin{bmatrix} \Delta\mathbf{r}_{t+1} \\ \Delta\mathbf{t}_{t+1} \end{bmatrix} = \mathrm{I}_{6\times6} \begin{bmatrix} \Delta\mathbf{r}_t \\ \Delta\mathbf{t}_t \end{bmatrix} + \mathcal{N}(\mathbf{0}, \Sigma_{r,t}) \tag{9}$$

where $\Sigma_{r,t} = \mathrm{diag}(\sigma_\phi^2, \sigma_\theta^2, \sigma_\psi^2, \sigma_{t_x}^2, \sigma_{t_y}^2, \sigma_{t_z}^2)$ is the covariance matrix for the rotational and translational velocities. The plane $\pi_t = \begin{bmatrix} \mathbf{n}_t^\mathsf{T} & d_t \end{bmatrix}^\mathsf{T}$ under tracking is updated as

$$\begin{bmatrix} \mathbf{n}_{t+1} \\ d_{t+1} \end{bmatrix} = \begin{bmatrix} \Delta\mathrm{R}_t & \mathbf{0} \\ -\Delta\mathbf{t}_t^\mathsf{T}\Delta\mathrm{R}_t & 1 \end{bmatrix} \begin{bmatrix} \mathbf{n}_t \\ d_t \end{bmatrix}. \tag{10}$$

For the 2-D substate $\mathbf{s}_t = \{p_{i,t}^\mathsf{T}\}_{i=1,\ldots,4}$, each vertex $p_{i,t}$ is transformed into its new location at time $t+1$ by

$$\underline{p}_{i,t+1} = \mathrm{H}\underline{p}_{i,t} \tag{11}$$

where $\underline{p} = [x, y, 1]^\mathsf{T}$ denotes the homogeneous image coordinates, and H is the homography induced by the planar motion in the form of (5), for which H^l and H^r are used for the left and right views, respectively. The update of 2-D substate $\mathbf{s}_t$ is random since the homography H is a random matrix computed from the random rotation and translation in 3-D. The above updating equations correspond to the dynamics model $p(\mathbf{x}_{t+1}|\mathbf{x}_t)$ for stereo tracking.

B. Observation Model for Stereo Tracking

The design of the observation model is an important issue for the particle filtering based tracking method. After a particle is propagated between consecutive frames, the observation model is responsible for determining its likelihood, which reflects how well it is keeping track of the object. For our stereo tracking method for speed measuring, the observation is comprised of the individual observations from the stereo views, i.e., $\mathbf{z}_t = \{\mathbf{z}_t^l, \mathbf{z}_t^r\}$. The stereo observations are correlated with each other since they are the projections of the same object. We utilize the stereo observations and their correlation in the observation model to promote its discriminative power and thus to improve the tracking accuracy. The observation model $p(\mathbf{z}_t|\mathbf{x}_t)$ is the product of three components

$$p(\mathbf{z}_t|\mathbf{x}_t) = l(\mathbf{z}_t^l)l(\mathbf{z}_t^r)c(\mathbf{z}_t^l, \mathbf{z}_t^r) \tag{12}$$

where $l(\mathbf{z}_t^l)$ and $l(\mathbf{z}_t^r)$ represent the single-view likelihood function based on the individual observations from the left and right views, respectively, and $c(\mathbf{z}_t^l, \mathbf{z}_t^r)$ models the cross likelihood function with respect to the correlation between the stereo observations. As the observation likelihood is the product of the single-view likelihoods and the cross-view likelihood, it provides better discriminative power and will be less affected by occlusions or clutters in a single view. The formulations of the likelihood functions are discussed in detail as follows.

1) *Single-View Likelihood:* The single-view likelihood function evaluates the likelihood of a given particle by comparing its extracted feature in a single view with the reference feature. It is similar to the conventional likelihood function employed in the single-view visual tracking methods. Considering that features, such as color histograms, gradient orientation histograms, extracted for the entire particle without utilizing spatial information do not meet the accuracy requirement of our tracking application, we employ partwise wavelet features to provide good discrimination power and localization accuracy. The region of interest of a given particle within a single view is divided into multiple parts, and wavelet features are extracted for each part and then concatenated to represent the combined feature, i.e., $f_t = \{f_{t,i}\}_{i=1,\ldots,N_p}$, where t is the temporal index, i is the part index, and N_p is the number of parts, respectively. We have chosen N_p to be 10 for the vehicle front by experiments such that each part contains enough pixels, and meanwhile the neighboring parts have different features to provide better spatial localization capability. It should be noted that in this paper the regions of interest are initialized by manual selection for the first frames and continuously tracked afterward. We partition the region of interest by mapping a regular grid onto the quadrilateral area, i.e., by applying a perspective transform to the grid to take its four corners into alignment with the vertices of the quadrilateral region of interest. The grid's dimensions are predefined and kept fixed throughout the tracking process. For a given part i, the feature vector $f_{t,i}$ is calculated by a set of Haar-like wavelet features, the same as in [28] and [29]. Although the entire vehicle front has slightly more complex geometries than a planar surface and undergoes perspective transformations, by partitioning it into small parts, each part can be considered as approximately planar and Haar-like feature can still be a good representation. Three types of Haar-like features (up-down, left-right, and diagonal) are extracted at two resolution levels, and a feature vector with a dimension of 16 is extracted for each block since we operate on gray-scale images in this paper. The Haar-like features are shown in Fig. 2. The extracted feature f_t is compared with the reference feature g_t for the Euclidean distance between them, i.e., $D = \|f_t - g_t\|_E$, and the likelihood function is given by

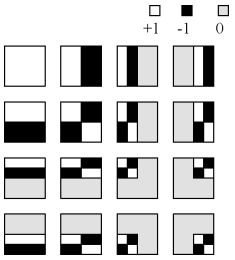


Fig. 2. Haar feature extraction.

$$l = \exp(-\lambda_D D^2) \tag{13}$$

where λ_D is the evaluation parameter. The reference feature g_t is initially extracted in the first frame and updated in each frame, that is

$$g_t = \epsilon g_{t-1} + (1 - \epsilon)\hat{f}_{t-1} \tag{14}$$

where $\epsilon \in (0, 1)$ controls the updating rate, and $\hat{f}_{t-1}$ is the extracted feature corresponding to the estimated state at time $t-1$. Because the likelihood is evaluated by taking the product of the single-view likelihoods from both views and the cross likelihood, only the particles which keep good track of the object in both views and have accurate stereo correspondences can have significant weights and large contribution to the estimated state; $\hat{f}_{t-1}$ will not have large deviations from the correct feature. Since we recursively update the feature in each frame and the view angle and scale changes between consecutive frames are not significant, linear update is sufficient for robust tracking and causes little feature drift and accumulative errors as justified in the experiments.

2) *Cross Likelihood:* In addition to utilizing the visual observation in each view independently for evaluating the single-view likelihood, we further make use of the correlation between the observations across the stereo views to achieve more discriminative power for accurate tracking. Given the calibration parameters of the stereo rig and the 3-D state of the object, the mapping between the object's projections between the stereo views can be established. By warping the object's projection in one view into the other using the established mapping, the matching pixels from two views should correspond to the same point on the object if the assumed 3-D state is accurate, and the pixel values are therefore correlated. Such a correlation between the pixel values across the stereo views can be utilized for evaluating the cross likelihood, and the detailed formulation is discussed as follows.

Let the intensity image patches in the stereo views be $\mathcal{I}_1(p), \forall p \in R_1$ and $\mathcal{I}_2(p), \forall p \in R_2$, where R_1 and R_2 are the regions of interest as defined by the 2-D states of a given particle. Given the camera intrinsic matrices K^l, K^r and the relative rotation and translation R, $\mathbf{t}$ between the stereo cameras, the projections of the plane $\pi = [\mathbf{n}, d]^\mathsf{T}$ under tracking are related by a cross-view homography H_c

$$\mathrm{H}_c = \mathrm{K}^r \left(\mathrm{R} - \frac{\mathbf{t}^\mathsf{T}\mathbf{n}}{d} \right) \mathrm{K}^{l-1}. \tag{15}$$

By applying the homography H_c to the image patch $\mathcal{I}_1$ in the left view, the image patch after warping is denoted as $\mathcal{I}'_1(p), \forall p \in R'_1$, where R'_1 is the warped region of interest. Should the warping be perfect, the matching pixels in $\mathcal{I}'_1$ and $\mathcal{I}_2$ are projected from the same object point, and the intensity values should be identical given the same camera settings for stereo views, that is [18]

$$\mathcal{I}_2(p) = \mathcal{I}'_1(p) + \epsilon(p) \quad p \in R_\cap \tag{16}$$

where $\epsilon(\cdot)$ is the imaging noise with zero mean, and $R_\cap$ is the intersected region between the warped image region R'_1 and the region of interest in the right view R_2. We use the normalized cross-correlation (NCC) between $\mathcal{I}'_1$ and $\mathcal{I}_2$ as the measure of the correlation between stereo observations

$$\mathrm{NCC}(\mathcal{I}'_1, \mathcal{I}_2) = \frac{1}{N-1} \sum_{p \in R_\cap} \frac{(\mathcal{I}'_1(p) - \bar{\mathcal{I}}'_1)(\mathcal{I}_2(p) - \bar{\mathcal{I}}_2)}{\sigma_{\mathcal{I}'_1} \sigma_{\mathcal{I}_2}} \tag{17}$$

where the intensity means $\bar{\mathcal{I}}'_1, \bar{\mathcal{I}}_2$ and standard deviations $\sigma_{\mathcal{I}'_1}, \sigma_{\mathcal{I}_2}$ are calculated within $R_\cap$ of N pixels. The cross likelihood $c(\mathbf{z}^l_t, \mathbf{z}^r_t)$ is calculated by

$$c(\mathbf{z}^l_t, \mathbf{z}^r_t) \propto \exp(-\lambda_c D_c^2) \tag{18}$$

where $D_c = 1 - \mathrm{NCC}(\mathcal{I}'_1, \mathcal{I}_2)$.

In summary, the stereo observation model employed in our work is comprised of three likelihood terms: the single-view likelihoods in the left and right views, and the cross likelihood which models the correlation of the two observations. Only the particles that have their underlying states accurately depict the actual location and motion of the object can have large values in all the three terms and thus a large value in the overall likelihood. In other words, taking the product of the likelihood terms creates a more peaked likelihood function, and thus provides better localization accuracy.

IV. Path Constraints for 3-D Motion

Visual tracking in the stereo views is dependent upon the object's 3-D motion, as discussed in the previous section. In general, an object going with unconstrained motion in 3-D space has six degrees of freedom. The large uncertainty in object motion results in large searching space for tracking. Meanwhile, different spatial configurations of an object may result in similar projections in the videos when it is far away, which results in ambiguities in recovering the 3-D motion to undermine the accuracy in the speed measurement. In many daily applications, an object of interest moves along a certain path that can be extracted from stereo views and reconstructed in 3-D. The information about the path can then be integrated into the stereo tracking as the constraint upon object motions to help reduce the motion uncertainty. In the remainder of this section, we will discuss how we can take advantage of the path information to improve the efficiency of stereo tracking and stereo measurement. We will first discuss the case when an object moves along a straight path, and then generalize the constraint to the curved path. The formulations for the curved case are also applicable to generic path motions. We base our discussion on the constant velocity model in both cases since it is most common in vehicle tracking applications. It can be extended to the constant acceleration model and other higher order motion models for more complex vehicle motions, which are beyond the scope of this paper and we refer the reader to [30] for a comprehensive survey.

A. Constraint With Straight Path

The case for which an object of interest moves along a straight path is common and important in reality such as measuring the speed of a vehicle on the freeway. Consider a straight path with its projections in the left and right views as $\mathbf{l}^l$ and $\mathbf{l}^r$, respectively. Given the camera projection matrices M^l and M^r, the image lines back project to two planes Q^l and Q^r

$$\begin{cases} Q^l = \mathrm{M}^{l\mathsf{T}} \mathbf{l}^l \\ Q^r = \mathrm{M}^{r\mathsf{T}} \mathbf{l}^r. \end{cases} \tag{19}$$

The intersection of these two planes corresponds to the 3-D line defining the straight path. The Plücker representation of the 3-D line L by the intersection of two plane equations is written as

$$L = Q^l Q^{r\mathsf{T}} - Q^r Q^{l\mathsf{T}} \tag{20}$$

which is a 4×4 matrix. Let the entry at the ith row and jth column of L be l_{ij}, the direction of the 3-D line $\mathbf{k} = [k_1 \quad k_2 \quad k_3]^\mathsf{T}$ is written as

$$\mathbf{k} = \begin{bmatrix} -l_{23} & l_{13} & -l_{12} \end{bmatrix}^\mathsf{T} / \| \begin{bmatrix} -l_{23} & l_{13} & -l_{12} \end{bmatrix}^\mathsf{T} \|. \tag{21}$$

For an object of interest moving along the straight path, it involves only the translation in the direction of $\mathbf{k}$ and no rotation, i.e., $\Delta\mathrm{R} = \mathrm{I}_{3\times3}$ and $\Delta\mathbf{t} = v_t\mathbf{k}$, where v_t is the translational velocity. Substitute them into (5) and (11), the 2-D state update equation can now be written as

$$\begin{aligned} \underline{p}_{t+1} &= \mathrm{K} \left(\mathrm{I} - \frac{v_t}{d} \mathbf{k}\mathbf{n}^\mathsf{T} \right) \mathrm{K}^{-1} \underline{p}_t \\ &= \begin{bmatrix} x_t - \dfrac{v_t}{Z_t + v_t k_3} [\alpha k_1 + (u_0 - x_t) k_3] \\ y_t - \dfrac{v_t}{Z_t + v_t k_3} [\beta k_1 + (v_0 - y_t) k_3] \\ 1 \end{bmatrix} \end{aligned} \tag{22}$$

or equivalently in the nonhomogeneous form

$$p_{t+1} = p_t - \frac{v_t}{Z_t + v_t k_3} \begin{bmatrix} \alpha & 0 & u_0 - x_t \\ 0 & \beta & v_0 - y_t \end{bmatrix} \mathbf{k} \tag{23}$$

where α, β, u_0, and v_0 are the horizontal and vertical focal lengths, the x and y coordinates of the principle point in the

(a) (b)

Fig. 3. Examples of curved path constraints for synthetic and actual traffic sequences. (a) Synthetic sequence. (b) Real sequence.

camera matrix K, and Z_t is the depth of the given point p_t by back projecting the point to the plane π_t. Since the tracked plane undergoes pure translation and the normal vector $\mathbf{n}$ is fixed, only the perpendicular distance d from the plane to the origin and velocity v need to be updated for the 3-D substate, for which we assume a constant velocity model with additive Gaussian noises

$$\begin{bmatrix} d_{t+1} \\ v_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_t \\ v_t \end{bmatrix} + \mathcal{N}(\mathbf{0}, \Sigma_m) \tag{24}$$

where $\Sigma_m = \text{diag}(\sigma_d^2, \sigma_v^2)$ is the covariance matrix for the distance and velocity. Compared with the unconstrained case that has six degrees of freedom in 3-D rotation and translation, the dimensionality and the uncertainty of the state space is greatly reduced.

B. Constraint With Curved Path

We represent a curved path by a *space curve*, which is defined as follows [31].

Definition 1: Let I be an interval of $\mathbb{R}$. A space curve C is a continuous function $C : I \to \mathbb{R}^3$ where $C(\tau) = [X(\tau), Y(\tau), Z(\tau)]^\mathsf{T}$ for all $\tau \in I$.

With calibrated cameras, a space curve can be reconstructed from its projections across two or more views. Earlier works [32] and [27] studied the reconstruction of low order planar curves, such as lines, conic sections, and the reconstruction of higher-order general space curves has also been studied recently [33]. Since the reconstruction of space curves from multiple views is an intensively studied topic and beyond the scope of this paper, we refer the reader to [33] for a thorough study. In our work, road structures, such as road boundaries and lane markings, are first extracted from the images and the corresponding 3-D curves are then reconstructed using the methods in [32] and [33]. The curved path constraints for individual lanes are then obtained by translating the reconstructed boundary curves in the 3-D. Considering the radius of practical road curves, the small offset of the vehicle from the center of the lane does not cause notable difference in its traveled distance. Fig. 3 shows the examples of the re-projections of curved path constraints in stereo views.

Due to safety considerations and other engineering issues, roads are always designed to have smooth transitions using arcs, clothoids, and so on. Therefore, we further assume that the object's motion path is continuous to the second order. Based on such assumptions, the *Frenet frame* is used to facilitate the modeling of the object motion.

Definition 2: The Frenet frame F is a local orthonormal basis $(\vec{T}, \vec{P}, \vec{B})$ attached to each point on the space curve C, given that C is a continuous space curve of class C^2.

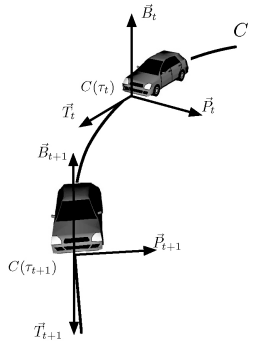


Fig. 4. 3-D motion with a curved path.

The vectors $\vec{T}$, $\vec{P}$, and $\vec{B}$ represent the tangent vector, principle normal vector, and binormal vector, respectively, as shown in Fig. 4, They can be derived from the parametric form of the space curve C as discussed in [31]. Consider the case for which we keep track of the front plane of a vehicle. The plane passing through the vehicle front intersects with the space curve C at a point $C(\tau)$, at which the current Frenet frame is constructed. In order to follow the path, the instantaneous velocity of the vehicle needs to be in the same direction as the tangent vector $\vec{T}$. The normal vector $\mathbf{n}$ of the plane can be considered as parallel to $\vec{T}$, and the plane contains the principle normal and binormal vectors $\vec{P}$ and $\vec{B}$. In other words, the tangent, principle normal, and binormal vectors are aligned with the vehicle's moving direction, its horizontal, and vertical axes, respectively. Note that when the object is maneuvering, for example when a car is changing its lane, the above analysis does not hold strictly. However, when the object is moving at a high speed within a limited path, the maneuvering direction may not deviate much from the tangent direction and the maneuvering is usually transient. Therefore, we consider such a motion constraint to be useful most of the time and can help reduce the uncertainty about the object motion.

Based on the motion constraint imposed by the motion path, we can now formulate the object's 3-D motion model. When the object moves along the space curve C, it undergoes both rotation and translation. Consider the object of interest at time t with its frontal plane intersecting the space curve at the point $C(\tau_t) = [X(\tau_t), Y(\tau_t), Z(\tau_t)]^\mathsf{T}$, as shown in Fig. 4. Suppose that it is traveling at speed v_t, at time t+1 it moves to a new location $C(\tau_{t+1}) = [X(\tau_{t+1}), Y(\tau_{t+1}), Z(\tau_{t+1})]^\mathsf{T}$ with the arc length from τ_t to τ_{t+1} being v_t, i.e., $v_t = \int_{\tau_t}^{\tau_{t+1}} \|C'(\tau)\| d\tau$. Considering that it is not always easy to solve for τ_{t+1} from the integral form and the time difference between the two frames is small, we use the chord length to approximate the arc length in order to calculate the object's location at time $t+1$. In this case, τ_{t+1} is obtained by solving the equation

$$\|C(\tau_{t+1}) - C(\tau_t)\| = v_t \quad \text{s.t.} \quad \tau_{t+1} > \tau_t \tag{25}$$

which is equivalent to finding the intersection point between the sphere centered at τ_t with radius v_t and the space curve C. In general, there may be two solutions for τ_{t+1}, the constraint $\tau_{t+1} > \tau_t$ ensures that the point along the forward direction is kept.

After the object's new location $C(\tau_{t+1})$ is determined given its motion speed v_t, the Frenet frame at τ_{t+1} can now be constructed as $F_{t+1} = (\vec{T}_{t+1}, \vec{P}_{t+1}, \vec{B}_{t+1})$. Since the Frenet frame is aligned with the object body as discussed in the previous subsection, the relative transformation between F_{t+1} and F_t corresponds to the transformation of the object of interest. For the object's rotation $\Delta\mathrm{R}_t$ between times t and $t+1$, it takes F_t into the same orientation with F_{t+1}, that is

$$\begin{bmatrix}\vec{T}_{t+1} & \vec{P}_{t+1} & \vec{B}_{t+1}\end{bmatrix} = \Delta\mathrm{R}_t \begin{bmatrix}\vec{T}_t & \vec{P}_t & \vec{B}_t\end{bmatrix}. \tag{26}$$

For the translation of the object between times t and $t+1$, it can be ideally written as $\Delta\mathbf{t}_t^0 = C(\tau_{t+1}) - \Delta\mathrm{R}_t C(\tau_t)$. However, in reality, it may not always strictly follow the exact path and the deviations need to be modeled for accurate tracking. Firstly, the object may have horizontal drift along the principle normal direction $\vec{P}_{t+1}$. Second, there is vertical fluctuation along the binormal direction $\vec{B}_{t+1}$, which may be resulted from uneven surfaces, and so on. Therefore, in addition to the ideal translation $\Delta\mathbf{t}_t^0$, we model the translations from horizontal drift and vertical fluctuation as $\Delta\mathbf{t}_t^{PB}$, which is zero mean white Gaussian with standard deviations σ_P and σ_B along $\vec{P}_{t+1}$ and $\vec{B}_{t+1}$ axes, respectively. Since the horizontal drift caused by the driver's behavior is usually greater than the vertical fluctuation due to uneven pavement, σ_P is given a larger value than σ_B. Note that $\Delta\mathbf{t}_t^{PB}$ also helps account for the difference in the translation when the reconstructed curve does not pass exactly through the central axis of the vehicle. The translation $\Delta\mathbf{t}_t$ as the composition of $\Delta\mathbf{t}_t^0$ and $\Delta\mathbf{t}_t^{PB}$, that is

$$\Delta\mathbf{t}_t = \Delta\mathbf{t}_t^0 + \Delta\mathbf{t}_t^{PB}. \tag{27}$$

Since $\mathbf{t}_t^{PB}$ only introduces a translation within the plane spanned by $\vec{P}_{t+1}$ and $\vec{B}_{t+1}$, it does not affect the intersection point τ_{t+1} between the object's plane and the space curve, and the rotation $\Delta\mathrm{R}_t$ derived above needs no change.

In summary, for the case with a curved path constraint, the rotation $\Delta\mathrm{R}_t$ and translation $\Delta\mathbf{t}_t$ and the plane equation π_t are dependent upon the current location τ_t of the object on the space curve $C(\tau)$ and the velocity v_t. Only two scalars τ_t and v_t need to be updated in the 3-D substate as compared with the unconstrained case in (9) and (10). We also assume a constant velocity model with additive Gaussian noise for the curved case.

V. Experiment Results

A. Experiment Setup

Our algorithm is implemented mainly using the Python scripting language. Since the major bottleneck of our algorithm lies in the feature extraction and evaluation, we have offloaded the feature computation to the graphical processing unit (GPU) using PyCUDA [34]. All the tests are performed on a laptop with a 2.4 GHz Intel Core 2 Duo processor with 2 GB RAM and an Nvidia GeForce 8600M GT GPU with 256 MB VRAM. Currently, we have an average processing time around 400 ms for each pair of stereo frames with 400 particles. Consider the computation overhead involved with the scripting language and the use of more powerful hardware, it is possible to bring the computation to 5 frames per second (f/s), which is the typical video acquisition rate in our application in order to obtain enough image displacements for tracking and speed measuring of a distant object.

We have used both synthetic and real-world video sequences in our experiments. The synthetic sequences are created using Autodesk 3ds Max software. The videos are rendered at a resolution of 1024×768 at 5 f/s. The real-world video sequences are recorded using our custom-built stereo rig consisting of two ImagingSource DFK41BU02.H cameras fixed on a 0.5 m horizontal bar. The two cameras are synchronized using a hardware timer trigger. The videos are recorded at resolutions of 1280×960 with frames rates of 5 f/s. The reference speed reading is obtained from GPS measurements and the calibrated speedometer. Since it is difficult to obtain accurate reference readings for time-varying speed due to the limitation of GPS and speedometer, for both straight and curve cases, we have assumed a constant velocity model for the vehicle motion and used the sequences with constant speed for testing. The test vehicles travel under cruise control over a long range to minimize the errors and variations in the reference readings. The standard deviation for the constant velocity model is chosen to be 0.05 m/s for the synthetic sequences. For the real sequences, the standard deviation is chosen to be 0.15 m/s, i.e., 0.33 mph, and we consider it to be sufficient for our tests since the vehicles in our tests were under cruise control. σ_P and σ_B for the horizontal and vertical drifts in the curved case are 0.10 m and 0.03 m, respectively. The observation parameters λ_D, λ_c, and ϵ are chosen to be 15, 20, and 0.3, respectively. Our tracking method is robust throughout the sequences, except having slightly degraded accuracy in the lane change cases.

For performance comparison, we have included the results obtained from feature point triangulation, Kalman filtering, and multimodal stereo tracking [24]. Feature point triangulation has been tested on the synthetic sequences, which have no imaging noises or compression artifacts. SIFT feature points [35] are matched across the stereo views, and the optimal triangulation scheme [27] is used to compute their 3-D locations. The centroid of all the triangulated 3-D points is computed, and the displacement of the centroid between consecutive frames is considered as the velocity. A Kalman filter is used to obtain a better estimation based on the raw triangulation results. For the dynamics model, we include the 3-D location and velocity into the filtering state and assume a constant velocity motion model. For the measurement model, we use the triangulated position as the measurement of the 3-D location with a Gaussian measurement noise. The covariance matrix for the measurement noise is determined using the image Jacobian as discussed in [36], for which we assume a 0.5 pixel standard deviation in the feature point matching. The computed interframe velocities from the raw triangulation results are not included in the measurement model. We have implemented the multimodal stereo tracking algorithm in [24] with two minor modifications to suit our application.

1) The vehicle front is modeled as a quadrangle bounding box instead of two ellipsoidal regions that were designed for person tracking.
2) Since the original method assumes the object to be parallel to the cameras, which is not the case in our

application, we transform the depth map using the known rotation angle between the camera and the straight path.

We have tested the method on the sequences with straight paths, since it is not directly applicable to the cases with curved paths where the orientation of the vehicle is varying and unknown.

B. Results on Synthetic Sequences

We first test our method on the synthetic sequences to verify its performance. Fig. 5 illustrates the stereo tracking on one of the synthetic straight sequences with the vehicle traveling at an actual speed of 55 mph. The results from both the unconstrained or constrained cases of our method, as well as the results from multimodal stereo tracking are shown for comparison. The number of particles is fixed to be 800 to compare the performances of different approaches. At the beginning, the constrained and unconstrained cases show similar tracking performance as shown for frames 1, 3, and 5. However, when the object further approaches, the unconstrained case shows inaccurate tracking results while the tracking of the constrained case is still reliable. The degraded performance of the unconstrained tracker is due to the higher degrees of freedom in the motion and thus a larger state space. The particle distributions for the constrained and unconstrained cases are shown in Fig. 5(a) and (b), where only the results for one view are shown since the results for the other view are similar. One can see that in the unconstrained case, particles are distributed in a larger area [Fig. 5(b)] than the constrained case [Fig. 5(a)]. The latter is therefore more accurate. For the multimodal stereo tracking, we have applied equivalent dynamics noises as the constrained case for a better comparison. Since the object has distinctive color compared with the background and strong gradient cues along the edges, the qualitative tracking results from multimodal stereo tracking are quite satisfactory.

Fig. 6(a) shows the results on speed estimates for the same sequence. We have also included the raw speed estimates computed from the triangulation of feature points, as well as the Kalman filtered speed estimates. The raw results from triangulation have large noises. The temporal smoothing of Kalman filtering provides a smoother speed trace; however, the results are still higher than the actual values. In spite of its satisfactory qualitative tracking performance, the quantitative speed estimates obtained from the multimodal stereo tracking method are not accurate and have considerable fluctuations. This is primarily due to the limitation to the subpixel accuracy of the stereo disparity algorithms for long-range applications, which is still a research issue in itself [37]. Note that even for the synthetic sequence that has no imaging noises, the disparity maps are not perfect as shown in Fig. 5(c). In addition, the method only uses the depth map as measurement information and takes the mean depth for evaluating the depth likelihood at each time step, the inherent noises in the individual depth map cannot be effectively filtered through the recursive state update. Our method without path constraint converges to a smaller value of velocity. Our method with a path constraint gives the best speed estimates among all the results. Because the differences in the image motions caused by different motion speeds when the object is far away are not significant, particles with inaccurate speed values may still get considerable weights in the first few steps. Therefore, it takes a few steps for the constrained case to converge to the actual value, and the subsequent measurements are quite accurate and have little fluctuation.

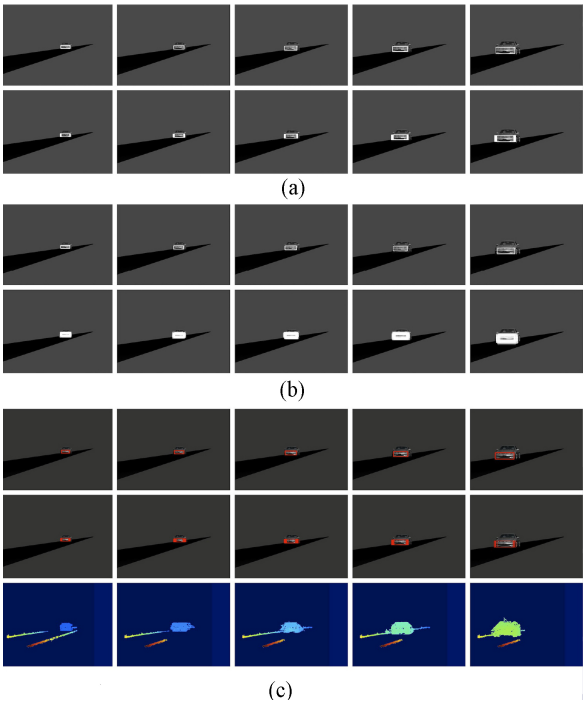


Fig. 5. Stereo tracking results on the synthetic straight sequence with the vehicle traveling at 55 mph. Frames shown: 1, 3, 5, 7, 9. (a), (b) Top rows show the tracking results and the bottom rows show the particle distribution. (c) Results from multimodal stereo tracking [24]. Top row: tracked locations. Center row: particle distributions. Bottom row: disparity maps. This figure is best viewed with magnification.

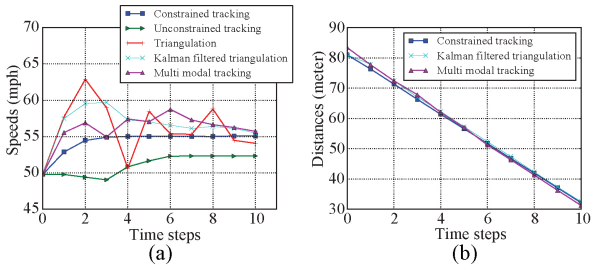


Fig. 6. Speed and depth estimates for the synthetic 55 mph sequence. For clarity, only the results from three methods are shown for the depth estimates. 800 particles are used for the constrained, unconstrained, and multimodal tracking. (a) Speed estimates. (b) Depth estimates.

Fig. 6(b) shows the depth estimates for this sequence. For clarity, only the results from our method with path constraint, Kalman filtered triangulation, and multimodal stereo tracking are shown. The Kalman filtered triangulation results have slightly larger depths, which are consistent with the analysis in [14] that the long-range stereo triangulation is biased toward larger distances. The multimodal stereo tracking has larger depths before frame 5 and lower depths thereafter. The depth estimates from our method have an almost constant slope except for the first frame.

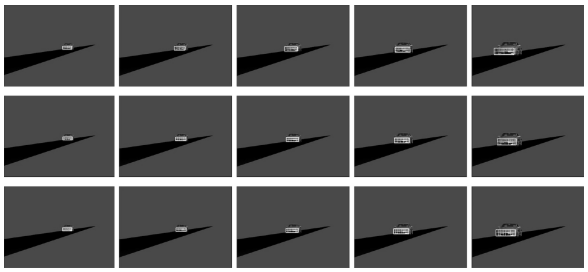


Fig. 7. Comparison of tracking results with different numbers of parts N_p in the single-view likelihood. The partition grids are overlaid on the tracking results. Top/middle/bottom rows: $N_p = 4/10/16$, respectively. This figure is best viewed with magnification.

We have also evaluated the performance of our constrained tracking approach with different numbers of particles, as well as different numbers of parts N_p for the single-view likelihood on this synthetic sequence. Each test is repeated for 20 times with the same initial conditions. For the tests with different numbers of particles, we fix the standard deviation of the constant velocity model to 0.05 m/s and $N_p = 10$. The average estimated speed, maximum error, and the average computational time are shown in Table I(a). With 200 particles, the average speed from the constrained tracking method is close to the actual value and the maximum error in the speed estimates is small. Further increasing the number of particles does not improve the results much. Because for the synthetic sequences, the simulated vehicle strictly follows the path constraint with constant speed and there is no imaging noise, our constrained tracking method does not need a large number of particles to cover the state space still providing reliable tracking. Therefore, we fix the number of particles to 200 for the remaining tests on the synthetic sequences in this paper. The average computational time increases approximately linearly with the number of particles. Table I(b) shows the average estimated speed and maximum error with different N_p values. Fig. 7 shows the qualitative tracking results from $N_p = 4$, 10, and 16, respectively. When $N_p = 4$, the individual parts are relatively large and particles with small offsets may still get considerable weights. Therefore, the localization accuracy of the extract feature is limited, which leads to inaccurate and unstable tracking results as shown in the top row of Fig. 7. When N_p is increased to 10, the tracking performance is much improved, as shown in the middle row of Fig. 7. Further increasing N_p does not improve the tracking performance. The results from $N_p = 16$ are slightly less accurate compared with those for $N_p = 10$. When the object is far away, its appearance details are limited in the image and parts with reduced size may not provide distinctive features. We have fixed N_p to 10 in all the remaining experiments in this paper.

We have also tested our stereo tracking algorithm using the synthetic sequences with curved paths. Only the constrained tracking results are shown here. Fig. 8 shows the constrained tracking results and particle distributions on a sequence for which a vehicle travels at 36 mph along a circular path. The vehicle undergoes both rotation and translation while traveling along the path, and its projections in the stereo views show both scale changes and view angle variations. Despite these challenges for visual tracking, the constrained tracking manages to continuously keep track of the vehicle front. Although when the vehicle comes close to the cameras its front cannot be well approximated as a plane in a strict sense, by recursively updating the reference features in each frame and utilizing the correlations between stereo views, the drifts of nonplanar feature locations do not cause significant

TABLE I
ERROR ANALYSIS ON THE SYNTHETIC 55 MPH SEQUENCE WITH DIFFERENT NUMBERS OF PARTICLES AND N_p VALUES

(a)

# Particles	Avg. Speed (mph)	Max Error (mph)	Avg. Time (ms/frame)
100	54.50	0.82	127.5
200	54.93	0.33	213.1
400	54.95	0.28	405.3
600	55.01	0.31	575.2
800	54.95	0.25	761.4

(b)

N_p	Avg. Speed (mph)	Max Error (mph)
4	54.01	2.72
6	54.24	1.33
8	54.71	0.65
10	54.93	0.33
12	54.95	0.35
16	55.05	0.52

(a) Results with different numbers of particles. Dynamics noise is 0.05 m/s and $N_p = 10$.
(b) Results with different N_p values. Number of particles is fixed at 200.

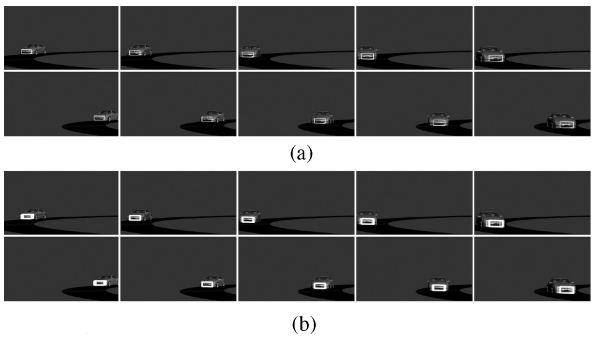


Fig. 8. Stereo tracking results on the synthetic curved sequence with the vehicle traveling at 36 mph. Frames shown: 1, 3, 5, 7, 9. Top and bottom rows in each figure show the left and right views, respectively. This figure is best viewed with magnification. (a) Tracking results for the constrained case. (b) Particle distributions for the constrained case.

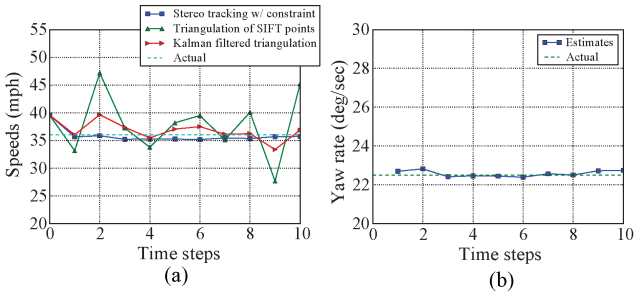


Fig. 9. Speed estimates for the synthetic curve sequence of 36 mph. For the angular velocities, only the results from our method are shown. (a) Translational velocities. (b) Angular velocities.

tracking errors. The path constraint plays a central role in providing reliable tracking results. With the path constraint, the rotation and translation are dependent upon the velocity, and the particles are propagated in such a way that they only correspond to the plausible 3-D motions and poses of the object within the current environment. This greatly reduces the search space and thus the number of particles required for effective tracking. As shown in Fig. 8(b), the particles follow the motion path and are concentrated around the actual locations of the vehicle.

Fig. 9(a) shows the speed estimates of the constrained tracking alongside with the raw and filtered results from the stereo triangulation of feature points. Because the vehicle's initial distance is smaller than those in the straight cases, the speed estimate approaches the actual value at the first step. The results are quite steady in the subsequent steps and are only slightly below the actual values. The raw results calculated from the feature point triangulation show large fluctuations, and the results from Kalman filtering are still not very accurate. Fig. 9(b) shows the estimations of angular velocities from our method. The estimated angular velocities are close to the actual value of 22.5°/s, although such a high angular velocity is not practical in reality due to the physical constraints of the ground vehicle. Since the simulated path is circular, we can compute the error introduced in (25), which approximates the arc using the chord length. At 5 f/s rate, the vehicle turns about 4.5° between two frames and the chord length is approximately 99.97% to the arc length. The modeling error is thus negligible.

C. Results on Real-World Sequences

In addition to testing our proposed method on the synthetic sequences, we further test it on real-world sequences. We have determined the number of particles to be 400 for our constrained tracking method by experiments and keep it constant in the tests on the real-world sequences unless otherwise stated. More particles are used in comparison with the synthetic sequences since there are fluctuations in the actual vehicle motions and imaging noises in the recorded videos. Fig. 10 shows the tracking results and particle distributions for a sequence with a straight path. The vehicle was traveling at 36 mph. Compared with the synthetic sequence, the background is more complex and the road surface is slightly tilted. Both our method with a path constraint and the multimodal stereo tracking method are tested. Equivalent dynamics models are applied to both methods for a meaningful comparison. For our method, the translations and scale changes of the vehicle in stereo views are tracked quite accurately throughout the sequence and the particle distributions are concentrated around the object's actual locations, as shown in Fig. 10(a). For the multimodal stereo tracking method, the tracking results also with 400 particles, and the disparity maps are shown in Fig. 10(b). Since our stereo rig is not in a parallel configuration, the left and lower parts of the disparity maps do not have valid values after stereo rectification. Compared with the synthetic sequence, the disparity computation for this sequence is more challenging and prone to errors. The tracking results are good at the beginning and drift to the right later on. Since

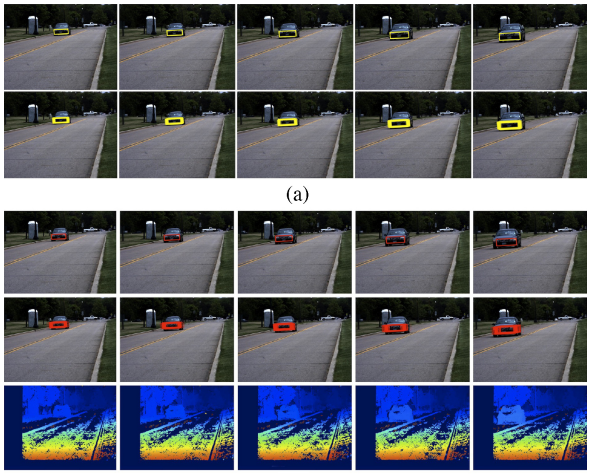


Fig. 10. Experiment results on the real sequence of 36 mph with straight path using 400 particles. (a) Results from our method. (b) Results from multimodal stereo tracking [24]. Top row: tracking results. Center row: particle distributions. Bottom row: the disparity maps.

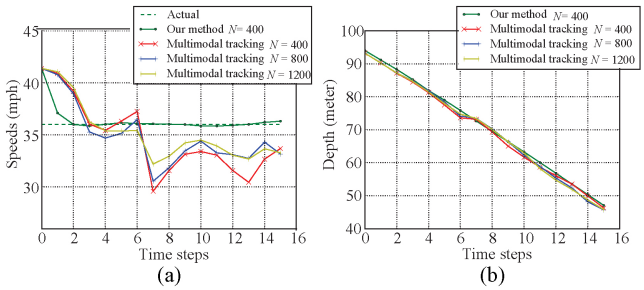


Fig. 11. Speed and depth estimates for the real straight sequence of 36 mph. N represents the number of particles. (a) Speed estimates. (b) Depth estimates.

the drifted particles have strong gradient responses along the top and bottom edges that are much longer than the side edges, the gradient likelihood values are not very low. Due to the subpixel accuracy limitation of the stereo disparity map, the side pixels cannot be effectively distinguished from the frontal ones, as shown in the bottom row in Fig. 10(b). The kernel smoothing employed in the method may play an adverse role in accurate tracking in this case. The particles appear slightly more dispersed compared with our method, although the same initial distribution and equivalent dynamics model are applied. Fig. 11(a) shows the comparison of speed estimates. Although the initial speed value is away from the actual value by over 10%, our method manages to approach the actual speed in two steps, and the subsequent values have small fluctuations. The speed estimates from multimodal stereo tracking with 400, 800, and 1200 particles show similar patterns that have abrupt drops at frame 7, and are not able to get close to the actual value thereafter. The results from 1200 particles have slightly lower fluctuations among the three. Fig. 11(b) shows the depth estimates. The depth estimates from our method are approximately along a straight line; however, the results from multimodal tracking show sudden changes at frame 7 and do not have a constant slope.

We have also tested the tracking algorithm on the sequences with curved paths. Fig. 12 shows the tracking results, as well

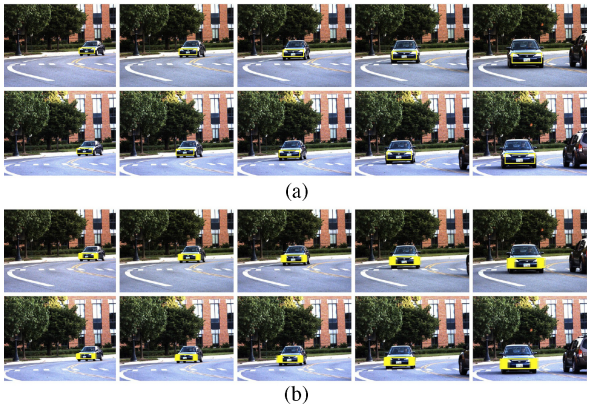


Fig. 12. Tracking results and particle distributions on the *Neil* sequence. (a) Tracking results. (b) Particle distributions.

as the corresponding particle distributions for the sequence *Neil*. The sequence was collected at a curved road segment on the OSU campus. The vehicle was traveling at 28 mph. The constrained stereo tracking shows good tracking results. The particles in both tests are distributed along the curved path, and effectively cover the motions of the vehicles. The speed estimates and the angular velocities are shown in Fig. 13. The speed estimates converge to the actual value after the first frame and are quite stable as shown in Fig. 13(a). The estimated angular velocities are shown in Fig. 13(b), which are not constant due to the varying curvatures of the actual path. Note that the angular velocity is not a state variable and is calculated by taking the difference between the estimated orientations of the vehicle between two frames, and therefore, no value is shown for frame 0. We have also evaluated the performance with different number of particles and different choices of noise magnitudes on the *Neil* sequence. Each test is repeated for 20 times with the same initial conditions, and the average value of the final estimated speed as well as the maximum error from 28 mph are calculated. For the tests with different numbers of particles, we fix the standard deviation of the constant velocity model to 0.15 m/s. The average estimated speed, maximum error, and the average computational time are shown in Table II(a). Since the initial speed value is much lower than the actual speed, with only 100 or 200 particles, the algorithm is not able to converge to the actual speed in some runs, and therefore, the average speed is lower than actual and the error is large. With more than 400 particles, the average speed is close to the actual value and the maximum error is reduced. The computational time increases approximately linearly with respect to the number of particles. For the tests with different dynamics noises, we fix the number of particles to be 400, and the results are shown in Table II(b). The average speed is not affected much by the increased dynamics noises. However, the maximum error increases as the dynamics noise gets larger.

Fig. 14 shows the tracking results with multiple vehicles on the sequence *OH-315*. The sequence was collected on the highway OH-315 near the OSU campus. Two independent particle filters are used for simultaneous tracking of both vehicles. Results for one view are shown here since the performance is similar in the other view. With the path constraints, the

TABLE II
ERROR ANALYSIS ON THE *Neil* SEQUENCE WITH DIFFERENT NUMBERS OF PARTICLES AND DYNAMICS NOISES

(a)

# Particles	Avg. Speed (mph)	Max Error (mph)	Avg. Time (ms/frame)
100	27.03	3.129	142.5
200	27.16	1.892	233.2
400	27.86	0.961	415.7
600	27.83	0.665	598.0
800	27.87	0.575	790.7

(b)

Noise Mag. (m/s)	Avg. Speed (mph)	Max Error (mph)
0.05	27.63	0.601
0.10	27.71	0.731
0.15	27.86	0.961
0.20	27.69	1.409
0.25	27.89	1.513

(a) Results with different numbers of particles. Dynamics noise is fixed at 0.15 m/s.
(b) Results with different dynamic noises. Number of particles is fixed at 400.

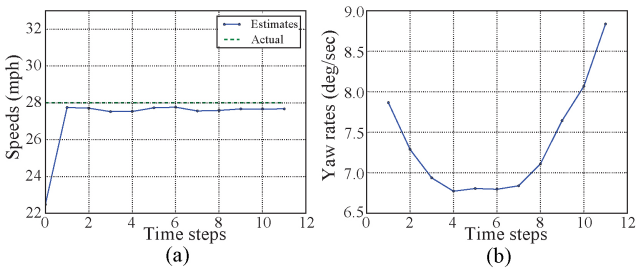


Fig. 13. Speed estimates for the *Neil* sequence at 28 mph. (a) Translational velocities. (b) Angular velocities.

Fig. 14. Tracking results and particle distributions with two vehicles on the *OH-315* sequence. The figure is best viewed in color with magnification.

particles for each tracker cover the motion paths of individual vehicles and the tracking results are quite accurate. Although the reference speed reading of 50 mph is only available for our test vehicle on the left, we have also measured the speed for the vehicle on the right. The speed estimates are shown in Fig. 15(a), where the initial speeds are both chosen to be the speed limit of 55 mph at this location. The speed measurements for the left vehicle are slightly higher than the reference speed reading.

In addition to testing our algorithm on the sequences with vehicles undergoing regular motions, we have also tested its performance for the lane change case. Fig. 16 illustrates the tracking results on two sequences with different speeds and lane change rates taken at the same location. For the

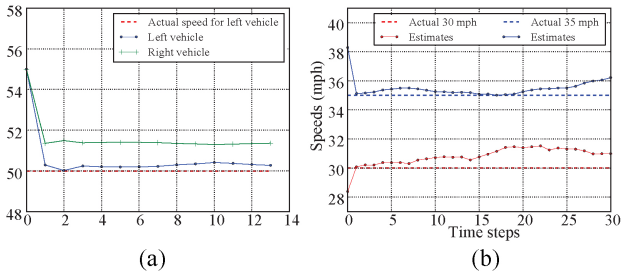


Fig. 15. Speed estimates for (a) *OH-315* sequence and (b) sequences with lane change.

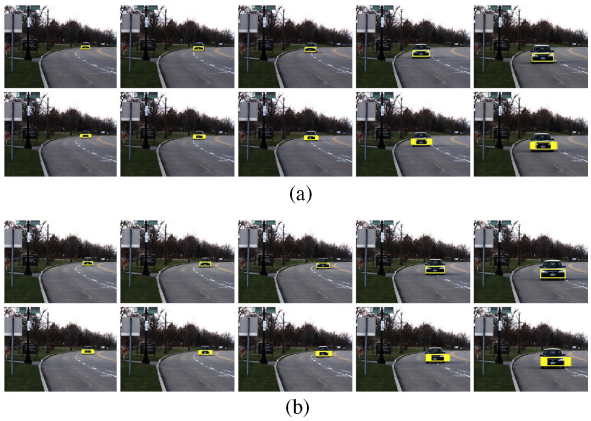


Fig. 16. Tracking results with lane change. Frames shown: 5, 10, 15, 25, 30. Top rows: tracking results. Bottom rows: particle distributions. Only results from one view are shown. (a) 35 mph sequence. (b) 30 mph sequence.

first sequence, the vehicle traveled at a constant speed of 35 mph and changed to the outer lane with a duration of around 5 s. For the second sequence, the test vehicle was instructed to travel at a lower speed of 30 mph but switch its lane at a faster than normal rate within 3 s. For both cases, the qualitative tracking results are quite good. The particle distribution for the second case is more biased toward the original path. With a lane width of 3.7 m, the horizontal drifts for each frame are around 0.15 and 0.25 m for the first and second sequences, respectively, which are still within the $3\sigma_P$ range given $\sigma_P = 0.10$. Therefore, the generated particles are still able to capture the vehicle's motion. However, for the quantitative speed estimates, they are not as accurate as those obtained from the regular sequences. Since the horizontal drift is no longer zero mean and the orientation of the vehicle front is not parallel to the principle normal direction of the original path constraint, the discrepancy between the assumed motion model and the actual vehicle motion leads to a degraded performance in the speed estimates.

VI. Conclusion

In this paper, we presented a novel stereo tracking method for measurement of the speed of moving objects within structured environments. The stereo tracking and motion estimation in 3-D were integrated within the particle filtering framework. For the state space, the 3-D location and motion were modeled using the 3-D substate and the 2-D tracks were represented by 2-D substates. By tracking a plane attached to the object of interest instead of its full 3-D model, the projected motions of the object in the stereo views were modeled by homographies. The observation likelihood was taken to be the product of individual observation likelihoods and cross likelihoods, and therefore provided better localization accuracy compared with single-view tracking. To further improve the accuracy in tracking and speed measurement, we utilized the path constraint to reduce the uncertainty about object motion and limit the dimension of the feasible state space. Both straight and curved paths were considered. The tracking performance and speed accuracy of our proposed method have been verified by the experiments on both synthetic and real-world sequences. In particular, the path constraint was effective in limiting the distribution of particles and was shown to be necessary for reliable tracking and speed measurement. Although in this paper we focused on the application to a vehicle traveling at a constant speed, the stereo tracking method with path constraint can be generalized to adaptive and time-varying speed measurement by incorporating more sophisticated dynamics models.

References

[1] T. Dang, C. Hoffmann, and C. Stiller, "Fusing optical flow and stereo disparity for object tracking," in *Proc. IEEE Conf. Intell. Transportation Syst.*, Sep. 2002, pp. 112–117.

[2] A. Barth and U. Franke, "Estimating the driving state of oncoming vehicles from a moving platform using stereo vision," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 560–571, Dec. 2009.

[3] M. Isard and A. Blake, "Condensation: Conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[4] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. 7th Eur. Conf. Comput. Vision*, vol. 2350. Jun. 2002, pp. 661–675.

[5] N. Apostoloff and A. Zelinsky, "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2003, pp. 558–563.

[6] J. Zhu, Y. Lao, and Y. F. Zheng, "Object tracking in structured environments for video surveillance applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 223–235, Feb. 2010.

[7] A. Doulamis, "Dynamic background modeling for a safe road design," in *Proc. Int. Conf. Pervasive Technologies Related to Assistive Environments*, 2010, pp. 1–8.

[8] N. D. Doulamis, "Coupled multiobject tracking and labeling for vehicle trajectory estimation and matching," *Multimedia Tools Applicat.*, vol. 50, no. 1, pp. 173–198, Sep. 2009.

[9] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2008, pp. 1–8.

[10] G.-S. J. Young and R. Chellappa, "3-D motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness results," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 8, pp. 735–759, Aug. 1990.

[11] R. Laganière, S. Gilbert, and G. Roth, "Robust object pose estimation from feature-based stereo," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 4, pp. 1270–1280, Aug. 2006.

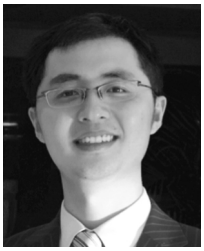
[12] Y. K. Yu, K. H. Wong, S. H. Or, and M. M. Y. Chang, "Robust 3-D motion tracking from stereo images: A model-less method," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 3, pp. 622–630, Mar. 2008.

[13] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," *Int. J. Comput. Vision*, vol. 73, no. 3, pp. 263–284, Jul. 2007.

[14] G. Sibley, L. Matthies, and G. Sukhatme, "Bias reduction and filter convergence for long range stereo," in *Robotics Research* (Springer Tracts in Advanced Robotics, vol. 28), S. Thrun, R. Brooks, and H. Durrant-Whyte, Eds. Berlin/Heidelberg, Germany: Springer, 2007, pp. 285–294.

[15] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[16] Y. Sumi, Y. Ishiyama, and F. Tomita, "Robot-vision architecture for real-time 6-DOF object localization," *Comput. Vision Image Understanding*, vol. 105, no. 3, pp. 218–230, Mar. 2007.
[17] J. Czyz, B. Ristic, and B. Macq, "A particle filter for joint detection and tracking of color objects," *Image Vision Comput.*, vol. 25, no. 8, pp. 1271–1281, Aug. 2007.
[18] J. Odobez, D. Gatica-Perez, and S. O. Ba, "Embedding motion in model-based stochastic tracking," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3514–3530, Nov. 2006.
[19] P. Pan and D. Schonfeld, "Video tracking based on sequential particle filtering on graphs," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1641–1651, Jun. 2011.
[20] L. Yuan, Y. F. Zheng, J. Zhu, L. Wang, and A. Brown, "Object tracking with particle filtering in fluorescence microscopy images: Application to the motion of neurofilaments in axons," *IEEE Trans. Med. Imag.*, vol. 31, no. 1, pp. 117–130, Jan. 2012.
[21] P. Barrera, J. M. Canãs, and V. Matellán, "Visual object tracking in 3D with color based particle filter," *Int. J. Information Technol.*, vol. 2, no. 1, pp. 61–65, Feb. 2005.
[22] G. Catalin and S. Nedevschi, "Object tracking from stereo sequences using particle filter," in *Proc. 4th Int. Conf. Intell. Comput. Commun. Process.*, Aug. 2008, pp. 279–282.
[23] A. Sabbi and M. Huber, "Particle filter based object tracking in a stereo vision system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2006, pp. 2409–2415.
[24] R. Muñoz-Salinas, M. García-Silvente, and R. M. Carnicer, "Adaptive multi-modal stereo people tracking without background modelling," *J. Visual Commun. Image Representation*, vol. 19, no. 2, pp. 75–91, 2008.
[25] Y. Wang, "Real-time moving vehicle detection with cast shadow removal in video based on conditional random field," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 437–441, Mar. 2009.
[26] X. Cao, C. Wu, J. Lan, P. Yan, and X. Li, "Vehicle detection and motion analysis in low-altitude airborne video under urban environment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1522–1533, Oct. 2011.
[27] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, MA: Cambridge Univ. Press, 2004.
[28] F. Zuo and P. H. N. de With, "Real-time facial feature extraction using statistical shape model and haar-wavelet based feature search," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 2. Jun. 2004, pp. 1443–1446.
[29] Y. Lao, J. Zhu, and Y. F. Zheng, "Sequential particle generation for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1365–1378, Sep. 2009.
[30] X. R. Li and V. Jilkov, "Survey of maneuvering target tracking: Part I. Dynamic models," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1333–1364, Oct. 2003.
[31] T. Banchoff and S. Lovett, *Differential Geometry of Curves and Surfaces*. Natick, MA: A. K. Peters, 2010.
[32] L. Quan, "Conic reconstruction and correspondence from two views," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 2, pp. 151–160, Feb. 1996.
[33] J. Kaminski and A. Shashua, "Multiple view geometry of general algebraic curves," *Int. J. Comput. Vision*, vol. 56, pp. 195–219, Feb. 2004.
[34] A. Klöckner. (2011, Jan.). *PyCUDA* [Online]. Available: http://mathema.tician.de/software/pycuda
[35] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[36] L. Matthies and S. A. Shafer, "Error modeling in stereo navigation," *IEEE J. Robot. Autom.*, vol. 3, no. 3, pp. 239–248, Jun. 1987.
[37] S. Gehrig and U. Franke, "Improving stereo subpixel accuracy for long range stereo," in *Proc. IEEE ICCV*, Oct. 2007, pp. 1–7.



Junda Zhu (S'08) received the B.S. and M.S. degrees from the Department of Information Science and Electronics Engineering, Zhejiang University, Hangzhou, China, in 2004 and 2006, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Ohio State University, Columbus.

His current research interests include image and video processing, with an emphasis on video object tracking.

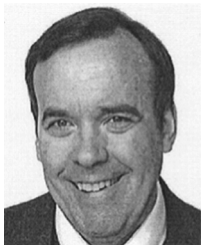
Liang Yuan (S'00–M'12) received the B.S. degree from Xinjiang University, Urumqi, China, in 1995, and the M.S. and Ph.D. degrees in electrical and computer engineering from Ohio State University, Columbus, in 2010 and 2011, respectively.

He is currently with the Department of Electrical and Computer Engineering, Ohio State University. His current research interests include computer vision and imaging processes, robotics, and automation in life science.

Yuan F. Zheng (F'97) received the B.S. degree from Tsinghua University, Beijing, China, in 1970, and the M.S. and Ph.D. degrees in electrical engineering from Ohio State University, Columbus, in 1980 and 1984, respectively.

He was with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, from 1984 to 1989. He has been with Ohio State University since August 1989, where he is currently a Winbigler Professor of electrical and computer engineering with the Department of Electrical and Computer Engineering. He served as the Chairman of the Department of Electrical and Computer Engineering from 1993 to 2004. He was on leave from Shanghai Jiao Tong University (SJTU), Shanghai, China, from 2004 and 2005, where he continues to collaborate with the SJTU faculty in the same fields. His current research interests include robotics (including humanoid robots), and robots and automation for life science, video, image, and signal processing.

Dr. Zheng was the Vice President for Technical Affairs of the IEEE Robotics and Automation Society from 1996 to 1999, an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION from 1995 to 1997, and the Program Chair of the IEEE International Conference on Robotics and Automation, Detroit, MI, in 1999, and Shanghai, China, in 2011. He received the Presidential Young Investigator Award from President Reagan in 1986.



Robert L. Ewing (S'77–M'87–SM'02) received the B.S.E.E. and M.S. degrees in physics from the University of Cincinnati, Cincinnati, OH, and the Ph.D. degree in electrical engineering from the University of Dayton, Dayton, OH.

He began his career with the Propulsion Laboratory, Wright-Patterson Air Force Base (AFB), Dayton, in 1970s with the development of jet engine control systems and the initial control system used on the F-15 aircraft. In the mid-1970s, he was with the Medical School, University of Cincinnati, where he worked in the area of electronic control and regeneration of peripheral (sciatic) nerves used in walking. He was a Medical Research Scientist with the Biodynamic Effects Division, Aerospace Medical Research Laboratory, Pensacola, FL, from 1977 to 1982. He became an Instructor with the Army, Air Force Institute of Technology (AFIT), and an Adjunct Instructor with Wright State University, Dayton, in 1982. He joined the Solid-State Electronic Devices Directorate, Wright Laboratory, Air Force Research Laboratory, Wright-Patterson AFB, in 1993, working in the area of hardware description language for very large scale integration synthesis. He has been a Technical Advisor with the Embedded Information Systems Branch, Information Directorate, as well as the Director of the Computer Engineering Research Consortium of local universities in the area of embedded system design since 1996. He is currently with the Sensor Directorate and is an Adjunct Professor with AFIT and Wright State University.

Dr. Ewing has been a Registered Professional Engineer with the State of Ohio since 1984.