Vision-based 3D Bicycle Tracking using Deformable Part Model and Interacting Multiple Model Filter

Hyunggi Cho, Paul E. Rybski and Wende Zhang

Abstract— **This paper presents a monocular vision based 3D bicycle tracking framework for intelligent vehicles based on a detection method exploiting a deformable part model and a tracking method using an Interacting Multiple Model (IMM) algorithm. Bicycle tracking is important because bicycles share the road with vehicles and can move at comparable speeds in urban environments. From a computer vision standpoint, bicycle detection is challenging as bicycle's appearance can change dramatically between viewpoints and a person riding on the bicycle is a non-rigid object. To this end, we present a tracking-by-detection method to detect and track bicycles that takes into account these difficult issues. First, a mixture model of multiple viewpoints is defined and trained via a Latent Support Vector Machine (LSVM) to detect bicycles under a variety of circumstances. Each model uses a part-based representation. This robust bicycle detector provides a series of measurements (i.e., bounding boxes) in the context of the Kalman filter. Second, to exploit the unique characteristics of bicycle tracking, two motion models based on bicycle's kinematics are fused using an IMM algorithm. For each motion model, an extended Kalman filter (EKF) is used to estimate the position and velocity of a bicycle in the vehicle coordinates. Finally, a single bicycle tracking method using an IMM algorithm is extended to that of multiple bicycle tracking by incorporating a Rao-Blackwellized Particle Filter which runs a particle filter for a data association and an IMM filter for each bicycle tracking. We demonstrate the effectiveness of this approach through a series of experiments run on a new bicycle dataset captured from a vehicle-mounted camera.**

I. INTRODUCTION

One of the ultimate goals in the automotive industry is to develop fully autonomous driving vehicles. One of key subsystems for achieving this goal is a robust perception system that will allow the vehicle to understand its current environment for the safety of people inside and outside of the vehicle [11]. Such a perception system must be able to detect and track other traffic participants such as cars as well as the class of objects called vulnerable road users (VRUs) [13] which includes entities such as bicyclists, motorcyclists, and pedestrians. For this purpose, we have been using the autonomous car "Boss" which won the 2007 DARPA Urban Challenge [7] as a test platform.

In this paper, we focus on the problem of identifying and extracting specific quantities of interest from the scene. We use the roof-mounted cameras on Boss to detect and track bicycles. Bicycles are a particularly challenging for an autonomous vehicle to track due to the fact that they are as unprotected as pedestrians but travel at higher speeds in very close proximity to vehicles. We have found that research into detecting and tracking bicycles for safety purposes shares a lot of similarities to detecting and tracking pedestrians but we believe the bicycle domain is more difficult than the pedestrian domain for two reasons. First, bicycles can change their appearances very drastically based on the viewing angle. Second, the relative speed of bicycles traveling on roads means that they can change their size and appearance in the camera's view in a very short time. Figure 1 shows some typical results of our bicycle tracking system.

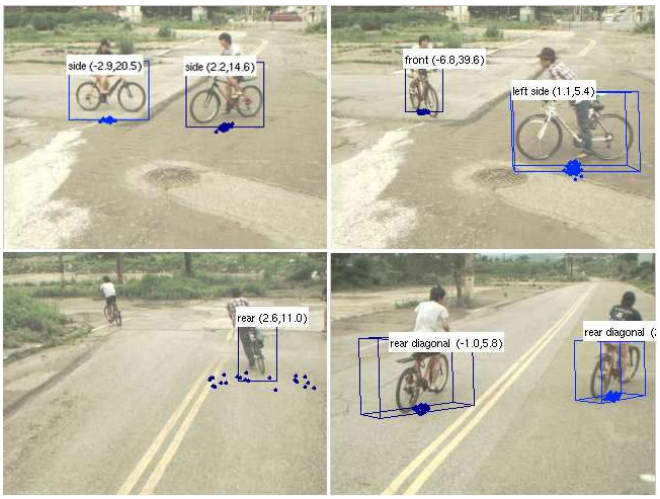

Fig. 1. Typical examples of our bicycle tracking system. A tracking result within (beyond) a certain distance (i.e., 8m) is visualized with a 3D (2D) bounding box. A text box syntax is {viewpoint (x coordinate, z coordinate)}. A set of points under a bounding box visualizes the uncertainty of tracking.

While other technologies such as LIDAR and RADAR are available to autonomous vehicles and all of these different technologies (including various combinations and fusion approaches) have been employed in various ways by other researchers [12], we are primarily considering the challenge of using a vision system to detect and track bicycles. Cameras and computer vision systems have a number of benefits as well as difficulties that must be contended with in order to be useful. On one hand, vision systems provide a high-resolution view of the world as compared to planar LIDAR or low-resolution scanning RADAR. Additionally, features such as color, texture, shape, and contours can all be extracted from vision systems which are unavailable to those other sensors. Another practical benefit of cameras

H. Cho and P. E. Rybski are with the Robotics Institute, Carnegie Mellon University, 5000, Forbes Ave., Pittsburgh, PA 15213, USA. {hyunggic, prybski}@cs.cmu.edu

W. Zhang is with the Electrical and Controls Integration Lab, General Motors R&D, 30500, Mound Rd, Warren, MI 48092, USA. wende.zhang@gm.com

is the relatively low cost of the sensor itself compared to LIDAR and RADAR. On the other hand, because the FOV of vision systems subtends a large area, objects of interest must be extracted from potentially complex backgrounds before they can be processed. Variations in lighting, object size, shape, and so forth mean that vision systems may be able to recognize a target in one set of conditions but may fail to recognize them in other situations.

The first contribution of this paper is that given a three-component bicycle model we built in our previous work [3], we exploit the unique characteristics of bicycle tracking. Clearly, bicycles do have their own kinematics and, in consequence, have more restrict constraints on their motion compared to that of pedestrians. We choose two motion models based on simplified bicycle's kinematics and fused them using a well-known IMM estimator. For each motion model, an extended Kalman filter (EKF) is used to estimate the position and orientation of a bicycle in the vehicle coordinates, hence 3D localization is possible with a proper scaling reasoning. The second contribution is the extension of the single bicycle tracking method to that of multiple bicycle tracking by incorporating a Rao-Blackwellized Particle Filter which runs a particle filter for data association and an IMM filter for each bicycle tracking. The final contribution of the paper is a new well-designed bicycle dataset which was made public for encouraging bicycle tracking research. The dataset is unique in that it is the first public domain bicycle dataset to our best knowledge. The dataset is valuable in terms of automotive applications in that it was collected based on the bicycle accident statistics [14].

The remainder of this paper is organized as follows. Section II reviews related work on detection and tracking of pedestrians. Our primary technical contributions in detection and tracking are described in Sections III and IV, respectively. We describe experimental results using the system in Section V and conclude in Section VI.

II. Related Work

Pedestrian Detection: There is a significant body of work on vision-based approaches for pedestrian detection. For a comprehensive survey of classical work, please see [12] and [16] while more recent work is surveyed in [11], [6], [5]. For the detection of pedestrians, there are roughly two main approaches: single template and part-based. This classification is based on representation of a human body regardless of features and classifiers used. Historically, a single template based approach was studied first and showed better performance compared to part-based models. Recently, however, some part-based models have shown more promising performance while they have a flexible and rich model. In a single template approach, the model captures a whole human body pattern using a single detection window. Papageorgiou et al. [18] uses Haar wavelet features in combination with a polynomial Support Vector Machine (SVM). Viola et al. [21] augment space-time information to their simple Haar-like wavelet features for moving people detection. Dalal and Trigg [4] show excellent performance for detecting human in a static image using a dense HOG (Histogram of Oriented Gradient) representation and a linear SVM. On the other hand, a part-based approach captures the pattern of each part and then combines results to make a final decision for pedestrian detection. Generally, part-based approaches can handle varying appearances of pedestrians due to clothing, pose, and occlusion, and thus, provide a more complex model for a pedestrian detection problem. Mohan et al. [18] divide human body into four parts: head, legs, left, and right arm. Each part detector is trained using a polynomial SVM and outputs are fed into a final classifier after checking geometric plausibility. Mikolajczyk et al. [17] model humans as assemblies of parts that are represented by the Scale Invariant Feature Transform (SIFT)-like orientation features. Felzenszwalb et al [9] demonstrate that a part-based model human detector can outperform many of existing current single template based approaches. Based on a variation of HOG features, they introduce a latent SVM formulation for training a part-based model from overall bounding box information without part location labels.

Pedestrian Tracking: For tracking of pedestrians, a number of mathematical frameworks have been proposed. Statistical or probabilistic methods such as the (extended) Kalman filter and particle filter are often employed. For instance, one such approach [13] uses an $\alpha - \beta$ filter to overcome gaps in detection where the proposed tracker is a simplified Kalman filter with a constant velocity model and predetermined steady-state gains. In another example [19], particle filters have been used to track a number of interacting people from a fixed camera.

III. Bicycle Detection with a Deformable Part-Based Model

Bicycle tracking from a moving vehicle is generally a challenging task especially when using a single video camera as its sole sensor. With advances of general object detection algorithms, tracking-by-detection has been a promising candidate for this purpose. This approach, basically, runs a detector for objects of interest at every time instant and provides a sequence of measurements to a tracking framework. In this work, we take a bicycle detector as a virtual sensor which generates a sequence of measurements. The output of this virtual sensor is a list of 2D bounding boxes and plays a pivotal role in a tracking process. Given a sequence of images, however, a robust and continuous bicycle detection is challenging due to the facts that a bicycle presents dramatic appearance changes according to camera viewpoints and also has an intra-class variability (e.g., mountain bikes vs. racing cycles). The first problem can be solved by building multiple view-based detectors to overcome dramatic appearance change. One of the common solutions to tackle the second problem is to establish a part-based model for an object of interest. Rather than trying to capture a global pattern of an object with one template, part-based models focus on parts of an object and, in consequence, provide more flexible and robust representations. While part-based models have an elegant formulation in theory, they have not shown a better

performance compared to a single template based approach. Recently, however, Felzenszwalb et al. [9] demonstrate a part-based model which outperformed the single template model by using a pictorial structure formulation in combination with a variation of HOG features. In this paper, our work for bicycle detection is largely based on this method. We build a eight view-based bicycle model and analyze the statistics of bicycle detection responses. The following subsections discuss some important details of this model and how it was applied to the algorithm in our research.

A. Deformable Part-Based Model

The core ideas of the deformable part-based model [9] boil down to the following three factors: a deformable part representation, an efficient matching process, and a latent SVM training process. First, they define a star-structured part-based model which is composed of a root filter, n (usually six) part filters, and associated deformation parameters. A root filter is for capturing an overall shape of an object (shown in the second row in Figure 2) and part filters are for capturing the appearance of each part of an object (shown in the third row). Finally, deformation parameters are for measuring the deviation of the part from its ideal location (shown in the fourth row). Thus, the score of the star model at a particular position and scale is defined by the sum of root filter score and part filter scores (from the best possible placement of the parts) subtracted by a deformation cost. The method also introduces a mixture of this star model to handle with significant changes in appearance according to viewpoint variation. Second, an efficient matching process based on dynamic programming and generalized distance transforms is proposed. With the mixture of star models, since a matching process itself is a huge optimization problem, it is most important to incorporate a fast method [10] for a detection task. Finally, a latent SVM training process is formulated to train a mixture of star models from bounding box ground truth. As the ground truth does not include part labeling information, part locations are treated as latent variables during training and thus the whole problem boils down to an optimization task with two sets of variables. In practice, they solve this problem using a coordinate descent algorithm by alternating between finding better latent values and optimizing the latent SVM objective function. In a detection process each example x is scored by a function of the following form:

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z). \tag{1}$$

where β is a vector of model parameters, z are latent values, and $\Phi(x, z)$ is a feature vector. In one star model, β is the concatenation of the root filter, the part filters, and deformation cost weights, z is a specification of the object configuration, and $\Phi(x, z)$ is concatenation of subwindows from a feature pyramid and part deformation features. We refer the reader to [9] for more details.

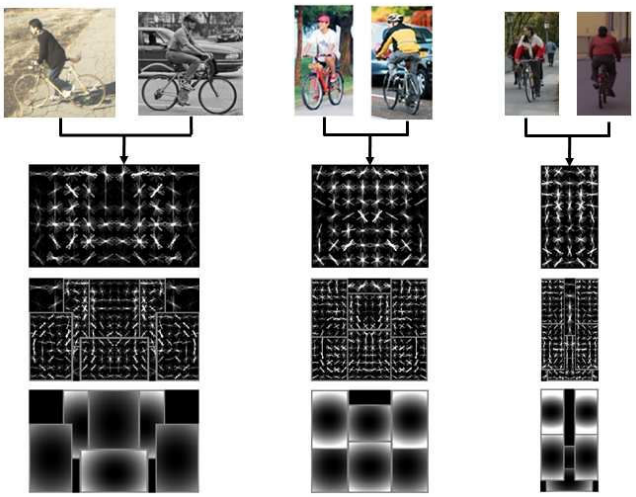

Fig. 2. Visualization of a three-component bicycle model. The top row shows several main viewpoints of a bicyclist and each column corresponds to one pair of specific view of a bicycle. Each row (from top to bottom) represents root filter, part filter, and deformation model, respectively.

B. Bicycle Detector as a Virtual Sensor

The physical sensor we are using is a monocular video camera which generates a sequence of images at a certain rate. At the arrival of a new image from the camera, our bicycle detector is applied and generates a set of bounding boxes if there are bicycles. Thus, in a tracking perspective, the bicycle detector is considered as a virtual sensor which generates a sequence of measurements. The measurement at time step k can be expressed by:

$$\begin{aligned} \mathbf{z}_k &= [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_n] \\ \mathbf{b}_i &= [t\ d\ l\ r\ v]^T \quad i = 1, ..., n \end{aligned} \tag{2}$$

where $\mathbf{b}_i$ indicates the y coordinates of the top(t) and bottom(d) borders of a bounding box, the x coordinates of the left(l) and right(r) borders, and an index of its view(v), respectively. The measurement set is fed into the Kalman filter's update process. In this sense, the idea using a bounding box as a measurement should be justified in terms of the following two aspects: continuity and consistency of responses.

Bicycle detection should be continuous as a viewpoint toward a bicycle is changing. Thus, the virtual sensor should be somehow equipped with the capability to handle this difficulty. Our baseline detector approaches this problem by building several models for different viewpoints of an object. However, this is a tricky trade-off problem between increasing the number of models and reducing the time complexity. Here, we propose to use eight view-based bicycle detector, paired and trained by its symmetric counterpart, in consequence, three physical models. This three-component bicycle model is visualized in Figure 2. The reasoning for using this combination will be discussed in Section V. Secondly, bicycle detection should be consistent. The quality of measurements in terms of its accuracy and consistency is

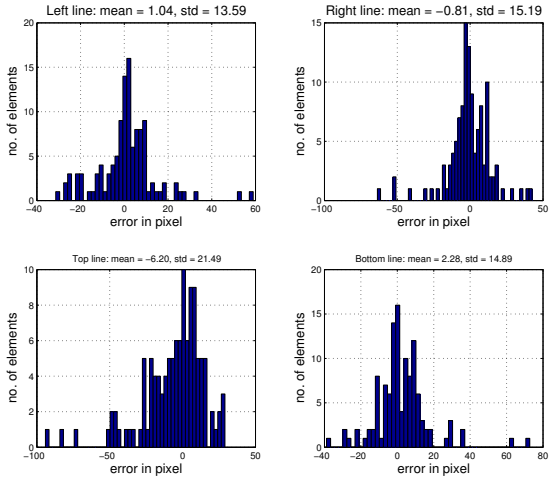

Fig. 3. Error distribution of bounding boxes obtained from the PASCAL VOC 2009 dataset `val`. Mean (m) and standard deviation (std) are also computed.

always an essential factor in a recursive filtering application. Thus, we provide an empirical justification for the quality of the response of our bicycle detector. Theoretically, as shown in Figure 2, the bicycle model itself manifests a quite tight bounding box regardless of its viewpoint, but in reality, detection results can show a significant variation depending on the configuration of its all parts. This fact can be easily revealed by analyzing statistics of detected bounding boxes. Figure 3 shows error distribution of four border lines of bounding boxes. This result is based on samples from the PASCAL VOC 2009 dataset `val` [1]. Out of 313 bicycle instances in the dataset, 118 bicycle instances were successfully detected with our bicycle detector and used in this analysis. Another interesting fact is that the error distribution can be well approximated by the Gaussian distribution, which is desirable for the Kalman filter.

IV. MULTIPLE BICYCLE TRACKING WITH AN IMM ALGORITHM AND A RAO-BLACKWELLIZED PARTICLE FILTER

With a set of measurements from our bicycle detector, a well defined tracking framework should be used to fuse the information from the object's motion model and measurements. Because the bicycle detector is a quite time demanding machine learning based module, we are interested in incorporating an algorithm with a lower complexity and providing certain indication about its uncertainty for tracking. For this reason, we chose to apply an extended Kalman filter (EKF) to our framework. We assume that real motion of a bicycle can be modeled by a set of simple motion models. In addition, as a measurement model, a nonlinear perspective projection equation is linearized with a flat ground assumption and used in the EKF update process. One key fact of our method is that we track the relative motion of a bicycle in the vehicle coordinate and thus, state variables are represented in the same coordinate accordingly. Specifically, after assuming that one bicycle track is already initialized, the tracking is conducted via the following two steps:

- **Predict:** Predict its next position in the vehicle coordinate using a set of motion models.
- **Update:** Update its state by incorporating current measurements, i.e., bounding boxes.

This process is illustrated in Figure 4. We discuss technical details of both a motion model set and a measurement model in the next subsections.

A. Bicycle Motion Model Set

As mentioned before, a bicycle has its own unique kinematics. Thus, at a first glance, it seems natural to use a bicycle's kinematics to model the real motion of a bicycle accurately. However, it become a fuzzy situation once considering the fact that the measurement in our case is a rough bounding box in the image space. From the sequence of these measurements, estimating all state variables (e.g., yaw angle and yaw rate) of a complicated model is a challenging task. Also, in information theoretic perspective, the level of accuracy between a motion model and a measurement model should be well balanced for a good tracking performance. We have been doing a comprehensive experiments to find the best solution for this issue. From our previous work [3], we have learned that a bicycle can be seen as a moving mass and tracked reasonably well using a constant velocity model. Here, we exploit a more complicated motion model based on simplified bicycle's kinematics and furthermore, improve tracking performance by fusing the information from both motion models. For this purpose, we use a well-known IMM (Interacting Multiple Model) filter. The underlying principle of the IMM filter is that the true motion of a bicycle cannot be exactly modeled by just one model, only be sufficiently approximated by using several motion models for representing dynamic driving behaviors of a target (i.e., maneuverings of a bicycle). The IMM filter runs several motion models in parallel and estimates a state by computing a weighted sum of several filter results which are based on different motion models. The derivation of the IMM estimator is well explained in [2] and omitted here for the brevity. Rather, we focus on a motion model set for the IMM filter.

For the model set, we use a combination of a constant velocity (CV) model and a simplified bicycle (SB) model. Since both motion models belong to a point model and a 3D bounding box is used as an object representation in real world scene, we chose to use the midpoint of the front bottom line of a 3D bounding box (displayed as a blue dot in Figure 4) as a representative point. Based on a flat ground assumption, the point can move freely only in the ground plane (i.e., X-Z plane) in the vehicle coordinates. First, in a CV model, the state of this moving point at time step k is expressed as a

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2009/#devkit/, accessed on Sep. 5 2009

vector:

$$\mathbf{x}_k = [x_k \quad z_k \quad \dot{x}_k \quad \dot{z}_k]^T \tag{3}$$

and the continuous-time state equation for this CV model [1] can be modeled as a linear, time-invariant system:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{w}(t) \tag{4}$$

where $\mathbf{w}(t)$ is a continuous time white Gaussian noise process with a power spectrum density $\mathbf{Q}_{cv}$. A discrete model of this state-space equation is used for the Kalman filter. Secondly, in a SB model, the state of the point is expressed by:

$$\mathbf{x}_k = [x_k \quad z_k \quad \psi_k \quad v_k \quad \omega_k \quad a_k]^T \tag{5}$$

where ψ_k, v_k, ω_k, and a_k are the yaw angle, forward velocity, yaw rate, and acceleration, respectively. The orientation of the forward velocity and acceleration vectors are defined with respect to the yaw angle as shown in Figure 4a. The continuous-time state equation for this SB model [15] assuming a constant acceleration and constant yaw rate is given by:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} v\cos(\psi) \\ v\sin(\psi) \\ \omega \\ a \\ 0 \\ 0 \end{bmatrix} \tag{6}$$

A discrete model of this state-space equation can be calculated by integration of the upper differential equation over one sampling period T and expressed by:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}[\hat{\mathbf{x}}_{k-1|k-1}] \\ &= \hat{\mathbf{x}}_{k-1|k-1} + \int_0^T \dot{\mathbf{x}}(\tau)d\tau \end{aligned} \tag{7}$$

The expression $\int_0^T \dot{\mathbf{x}}(\tau)d\tau$ in Equation 7 is represented in matrix form as:

$$\begin{bmatrix} \frac{v+aT}{\omega}SW + \frac{a}{\omega^2}CW - \frac{v}{\omega}\sin(\psi) - \frac{a}{\omega^2}\cos(\psi) \\ -\frac{v+aT}{\omega}CW + \frac{a}{\omega^2}SW + \frac{v}{\omega}\cos(\psi) - \frac{a}{\omega^2}\sin(\psi) \\ \omega T \\ aT \\ 0 \\ 0 \end{bmatrix} \tag{8}$$

where $SW = \sin(\psi + \omega T)$ and $CW = \cos(\psi + \omega T)$. The noise covariance matrix $\mathbf{Q}_{sb}$ of this discrete-time process can be computed using the direct discrete method [2] as $\Gamma D Q_c D^T \Gamma^T$ where the noise gain matrix Γ is the Jacobian of Equation 7, Q_c is the noise covariance matrix for the continuous-time process, and D is a mapping matrix.

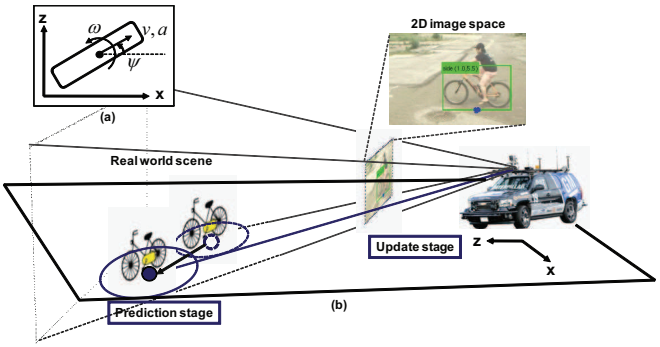

Fig. 4. Simplified bicycle motion model (a) and an illustration of the bicycle tracking process: prediction and update (b). Ellipses under bicycles represent uncertainty of the estimates.

B. Bicycle Measurement Model

In our work, since a bicycle detector is used as a virtual sensor device, the measurements are bounding box positions in the image space. In addition, the tracking process itself is executed in the state space (i.e., in the vehicle coordinate). Thus, a measurement model should be able to map the state variable $\mathbf{x}$ into its measurement space (i.e., in the image coordinate). To facilitate this mapping, we use only one representative point which is the midpoint of a bottom line of a 2D bounding box as a measurement. Then, the mapping process is simply done by a perspective projection equation. The nonlinear mapping of the state space into the measurement space of the video camera is given by:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, k) + \mathbf{v}_k \tag{9}$$

where $\mathbf{v}_k$ is the measurement noise at time step k and can be determined by analyzing the statistics of detection results as discussed in Section III. The nonlinear mapping function $\mathbf{h}$ is obtained by the following transformation:

$$\begin{aligned} \begin{bmatrix} u \\ v \\ w \end{bmatrix} &= \begin{bmatrix} f/s_x & 0 & u_c \\ 0 & f/s_y & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} & | & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \end{aligned} \tag{10}$$

where s_x and s_y are scale factors in x and y, respectively, and (u_c, v_c) is a camera optical center and f is the focal length of a camera. $\mathbf{R}$ is a rotation matrix and $\mathbf{t}$ is a translation vector for extrinsic parameters. The parameters a_{ij} are the corresponding entries of the final perspective projection matrix. Based on a flat ground assumption, the vector function $\mathbf{h}$ is expressed by:

$$h_1 = \frac{a_{11}X + a_{13}Z + a_{14}}{a_{31}X + a_{33}Z + a_{34}} \quad h_2 = \frac{a_{21}X + a_{23}Z + a_{24}}{a_{31}X + a_{33}Z + a_{34}} \tag{11}$$

C. Extension to Multiple Bicycle Tracking

The single bicycle tracking method discussed in the previous subsections is extended to the multiple bicycle tracking framework by incorporating a Rao-Blackwellized Particle Filter (RBPF). Since we have multiple measurements and no information about how many bicycles exist at a certain time step, the scope of the problem is quite broad, including solving a data association problem, estimating the number of bicycles, and each bicycle tracking. The RBPF framework provides a mathematical tool to handle this daunting task. The core idea of the RBPF is to break down a huge state estimation problem into two smaller problems, one can be solved by an analytical solution and the other one can be solved by a particle filter, hence better results than what could be obtained from a pure particle filter solution. Following the Rao-Blackwellized Monte Carlo data association (RBMCDA) algorithm given in [20], we apply the algorithm into our multiple bicycle tracking problem. Specifically, it relies on a Bayesian factorization to separate the posterior into two parts: 1) an estimation of the number of bicycles and data association problem and 2) a tracking problem. The RBPF solves the first problem via a particle filter and then, with known number of bicycles and data associations, tracks each bicycle using an IMM filter developed specifically for bicycles. Here, we briefly describe the problem formulation and how we applied the RBMCDA algorithm. In our case, the problem can be formulated by the following decomposition of the posterior:

$$p(\lambda_{0:k}, \mathbf{x}_{0:k}|\mathbf{z}_{1:k}) = p(\mathbf{x}_{0:k}|\mathbf{z}_{1:k}, \lambda_{0:k})p(\lambda_{0:k}|\mathbf{z}_{1:k}) \quad (12)$$

where $\mathbf{x}_k$ is the state variable of n bicycles at time step k, $\mathbf{z}_{1:k} \triangleq \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k\}$, and λ_k is the latent variable which contains the visibility indicator $\mathbf{e}_k$ and the data association indicator c_k at the time step k, i.e., $\lambda_k = \{\mathbf{e}_k, c_k\}$. It is important to understand that this factorization is always true, but only applicable to a problem when there is certain structure within the state variables. In our case, for example, the state variable is a mixture of state variables of n bicycles, a data association indicator meaning that which measurement corresponds to which track (i.e., bicycle), and a visibility indicator which controls target's initialization and termination. Once the second term in the right hand side of Equation 12 is determined, the first term can be easily solved by our single bicycle tracking method. The RBMCDA algorithm solves the second term using a particle filter by analyzing only one measurement at a time assuming that at most one target can terminate at any time step.

V. Experimental Results

We quantitatively evaluated our bicycle tracking method using various real world datasets. To evaluate our bicycle detector, we used the PASCAL VOC dataset [8] with a selected subset of a new bicycle dataset we collected from our experimental vehicle. As for the bicycle tracking evaluation in a real application context, we collected a new bicycle dataset of various scenarios based on the bicycle accident statistics [14]. This bicycle dataset consists of 6 sequences (three from the stationary ego-vehicle and three from the moving ego-vehicle) and is made publicly available to the community for research purposes[2]. Tracking experiments for all sequences are also conducted.

A. Detection Performance

To build a bicycle model, we used 357 positive training samples and 3300 negative samples. Based on samples from the PASCAL VOC 2009 dataset `train`, we discarded some bad samples (e.g. too small or too weird viewpoint) and then augmented the dataset with 160 positive samples from our bicycle dataset. We trained a three-component bicycle model which can capture eight different viewpoints of a bicycle (i.e., frontal and rear, four different diagonal views, and right and left side view) by pairing samples of one specific viewpoint with samples of its symmetric counterpart. Figure 2 visualizes this mixture model. For the test set, we used the same PASCAL dataset `val` plus 100 positive samples from ours. We run the detector equipped with three-component bicycle model over all images in the test set and draw a precision-recall (PR) curve for an evaluation. Some typical examples of bicycle detection are shown in Figure 5a and a PR-curve for our model is compared with that of the two-component model as well as that of the one-component model in Figure 5b. We used the same evaluation criterion of VOC PASCAL competition for accepting as a successful detection. The logical reasoning for using a three-component model can be explained with this result. As discussed in Section III, choosing the number of model is a trade-off between the detection rate and running time. From the PR curve, it is obvious that two and three-component model outperforms one-component model and two and three-component models show very similar performance. Since we use viewpoint information, which is contained as an element of a measurement, for a tracking process, we decided to use a three-component bicycle model.

B. Tracking Performance

Tracking experiments were conducted on all of our new bicycle dataset which consists of six video sequences from different scenarios. Half of them is from a stationary ego-vehicle and the other half of them is from a moving ego-vehicle. The sequences are challenging in that we tried to capture the true motion pattern of a bicycle with respect to an ego-vehicle, which is very close to real traffic situations. Some important statistics of the sequences are summarized in Table I. Since the dataset is quite challenging and the number of sequences is large (i.e., 6), it is hard to interpret the results quantitatively. Thus, we have listed the root mean square errors (RMSE) of position estimates in the last two columns in Table I to make a comparison between single motion model approach (CV model) and an IMM approach (CV and SB models). We also provide the result videos on six sequences at the same website as our bicycle dataset. Here, we only analyze three specific scenarios ('seq1', 'seq3', and 'seq6') in detail.

[2]http://www.cs.cmu.edu/~hyunggic/bicycle

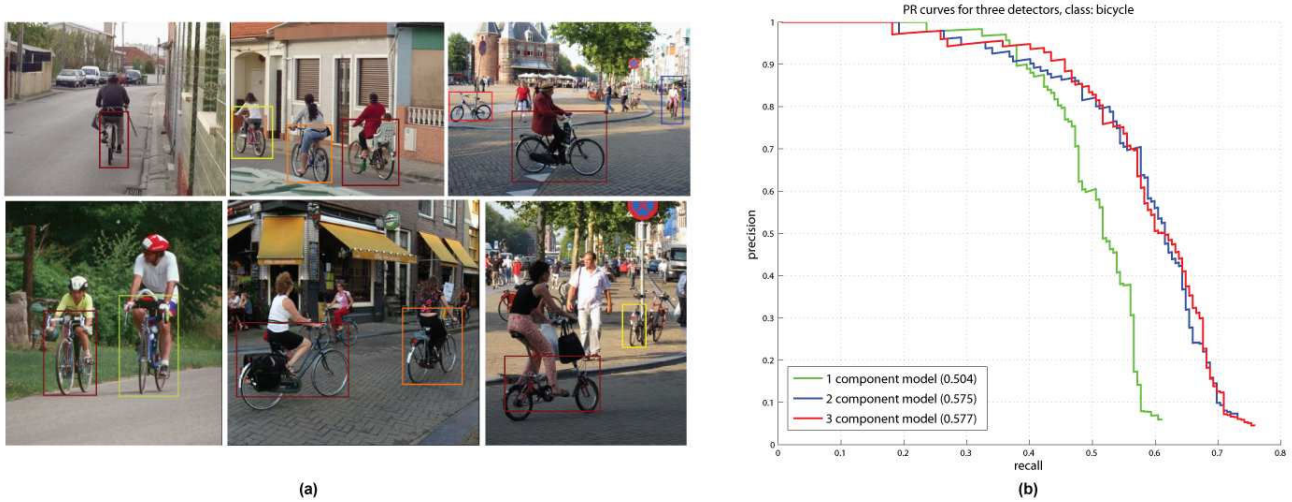

Fig. 5. Examples of detection with our three-component bicycle model (a), PR curves for three detectors (b). The red plot shows the response for the three-component model trained with PASCAL2009 and our dataset and the blue plot and green plot show the response for two-component model and one-component model, respectively, trained with the same dataset.

In the 'seq1' case, two bicyclists are moving laterally while the ego-vehicle is not in motion. This sequence is a relatively easy case, but an interesting point is that our tracking system can track the further bicyclist well even when there is quite amount of occlusion. The second row of Figure 7 shows this result.

In the 'seq3' case, a bicyclist comes across the road in front of the stationary ego-vehicle and makes a turn toward the vehicle so that the left side and frontal view of the bicycle are seen and must be tracked. The tracking result is shown in Figure 6 by plotting filtered state variables, especially positions in X and Z coordinates, at every time step. Selected tracking result images are also shown in the first row in Figure 7. As can be seen in the magnified region, we can clearly see the advantage of using a more complicated bicycle motion model in combination with a CV motion model.

In the 'seq6' case, three bicyclists are shown, one bicyclist is out of detection distance and two bicyclists become visible in the field of view (FOV), moving longitudinally along the ego-vehicle while the ego-vehicle itself moves forward. The third row of Figure 7 shows tracking results of our system. As can be seen, our system manages to track two bicycles despite significant egomotion and dynamic viewpoint changes of bicycles.

TABLE I

DETAILS OF THE SIX BICYCLE SEQUENCES. SM(SINGLE MODEL), IMM(INTERACTING MULTIPLE MODEL)

Seq.	ego-vehicle	bicycle	RMSE(SM)	RMSE(IMM)
'seq1'	stationary	laterally	0.0183	0.0216
'seq2'	stationary	longitudinally	6.6207	6.6196
'seq3'	stationary	randomly	0.1515	0.1443
'seq4'	moving	laterally	2.3493	2.3860
'seq5'	moving	longitudinally	7.0884	6.860
'seq6'	moving	randomly	11.0929	10.6281

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a vision-based 3D bicycle tracking method for intelligent vehicles using a deformable part model based detector and an IMM tracking algorithm. To robustly detect bicycles, we used *Felzenszwalb*'s deformable part-based detector [9] to construct a powerful three-component bicycle model. This robust bicycle detector provides a series of measurements (defined as bounding boxes) to the tracking module. We defined two different motion models to describe the kinematics of a bicycle. For each motion model, an extended Kalman filter (EKF) is used to estimate the position and velocity of a bicycle in the vehicle coordinate system. Finally, we show how we extend our IMM tracking method to allow it to track multiple bicycles by incorporating a Rao-Blackwellized Particle Filter to solve the data association problem. This complementary approach allows our system to effectively track a bicycle even when it changes orientation (and thus appearance) in the image. We have shown several experiments that illustrate the effectiveness of each component of the proposed method. As part of our future work, we intend to develop a new measurement mapping function to extract more information from the 2D bounding box and also apply our current method to other types of relevant objects such as pedestrians and vehicles.

VII. ACKNOWLEDGMENTS

This project was funded by General Motors through the General Motors-Carnegie Mellon Autonomous Driving Collaborative Research Lab.

REFERENCES

[1] Y. Bar-Shalom. *Tracking and data association*. Academic Press Professional, Inc., San Diego, CA, 1987.

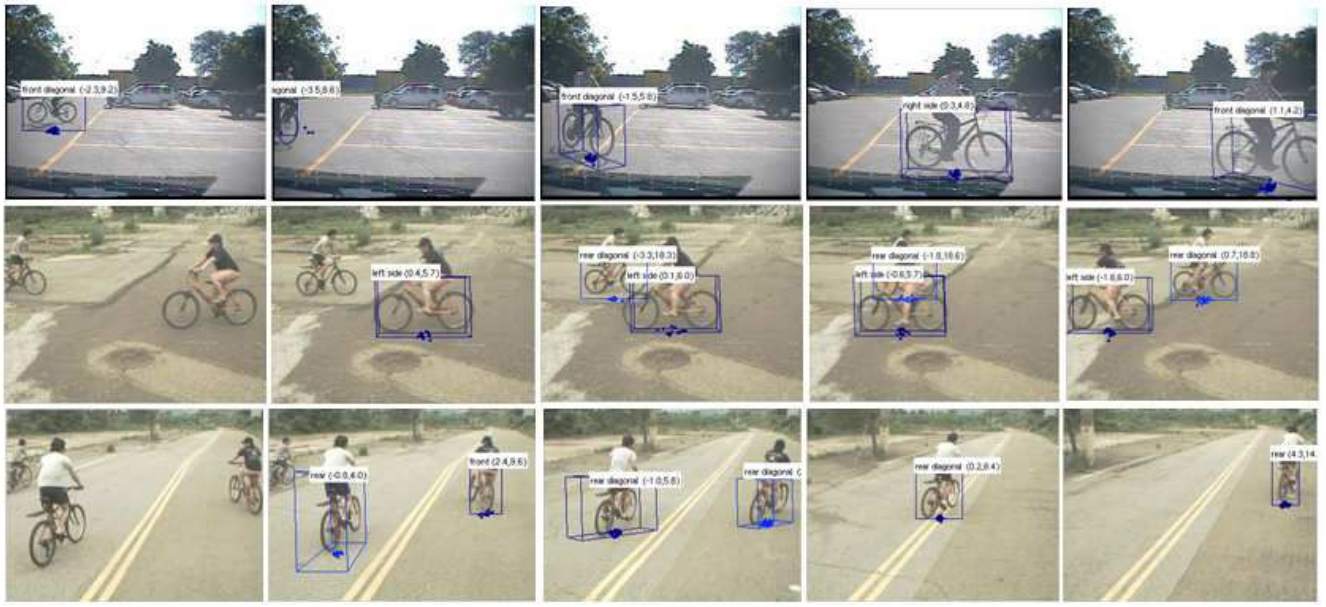

Fig. 7. Tracking results for three scenarios: 'seq1' (second row), 'seq3' (first row), and 'seq6' (third row). A tracking result within (beyond) a certain distance (i.e., 8m) is visualized with a 3D (2D) bounding box. 3D localization is possible by combining the state variable and a scale factor from the detection result.

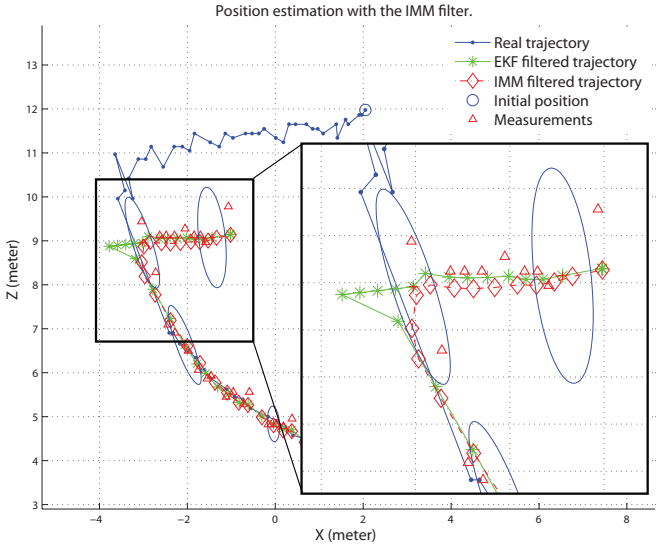

Fig. 6. The estimated path trajectory. The ellipses represent the 2-σ confidence regions for the bicycle position estimates. The magnified region clearly shows the advantage of using a SB model as well as a CV model.

[2] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation*. Wiley Interscience, 2001.

[3] H. Cho, P. Rybski, and W. Zhang. Vision-based bicycle detection and tracking using a deformable part model and an ekf algorithm. *IEEE Intelligent Transportation System Conference*, 2010.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Conference on Computer Vision and Pattern Recognition*, 2005.

[5] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. *Conference on Computer Vision and Pattern Recognition*, 2009.

[6] M. Enzweiler and D. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2008.

[7] C. U. et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part I*, 25(8):425–466, June 2008.

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results. http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99, 2009.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Intl. Journal of Computer Vision*, 61(1), 2005.

[11] T. Gandhi and M. M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transaction on Intelligent Transportation System*, 8(3):413–430, 2007.

[12] D. M. Gavrila. Sensor-based pedestrian protection. *IEEE Intelligent System*, 16(6):77–81, 2001.

[13] D. M. Gavrila, J. Giebel, and S. Munder. Vision-based pedestrian detection: The protector system. *IEEE Intelligent Vehicle Symposium*, pages 13–18, 2004.

[14] W. Hunter, W. Pein, and J. Stutts. Bicycle crash types: A 1990's informational guide. (FHWA-RD-96-104), 1997.

[15] N. Kaempchen, K. Weiss, M. Schaefer, and K. Dietmayer. Imm object tracking for high dynamic driving maneuvers. *IEEE Intelligent Vehicles Symposium*, pages 825–830, 2004.

[16] Z. Li, L. L. K. Wang, and F. Wang. A review on vision-based pedestrian detection for intelligent vehicles. *Conference on Vehicular Electronics and Safety*, 2006.

[17] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *European Conference on Computer Vision*, I:69–81, 2004.

[18] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[19] K. Smith, D. Gatica-Perez, and J. M. Odobez. Using particles to track varying numbers of interacting people. *Conference on Computer Vision and Pattern Recognition*, 2005.

[20] S. Srkk, A. Vehtari, and J. Lampinen. Rao-blackwellized particle filter for multiple target tracking. *Information Fusion*, 8(1):2–15, 2007.

[21] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, 63(2):153–161, 2005.