

Bicycle Tracking Using Ellipse Extraction

Tohid Ardeshiri, Fredrik Larsson, Fredrik Gustafsson, Thomas B. Schön, Michael Felsberg
Department of Electrical Engineering
Linköping University
Linköping, Sweden
e-mail: {tohid, larsson, fredrik, schon, mfe}@isy.liu.se

Abstract—A new approach to track bicycles from imagery sensor data is proposed. It is based on detecting ellipsoids in the images, and treat these pair-wise using a dynamic bicycle model. One important application area is in automotive collision avoidance systems, where no dedicated systems for bicyclists yet exist and where very few theoretical studies have been published. Possible conflicts can be predicted from the position and velocity state in the model, but also from the steering wheel articulation and roll angle that indicate yaw changes before the velocity vector changes. An algorithm is proposed which consists of an ellipsoid detection and estimation algorithm and a particle filter. A simulation study of three critical single target scenarios is presented, and the algorithm is shown to produce excellent state estimates. An experiment using a stationary camera and the particle filter for state estimation is performed and has shown encouraging results.

Keywords: Tracking, Particle Filter, Computer Vision, Ellipse Extraction, Bicycle.

I. INTRODUCTION

Automotive safety is an active research [2] and development area in the industry of highest importance for society. There are numerous collision avoidance (CA) and mitigation systems on the market, and the first systems for protecting pedestrians are appearing right now [11]. Such a driver assistance system can highlight pedestrians to the driver using infrared cameras, autonomously warn the driver, brake the car to avoid or mitigate collision, steer away, or, as the last resort when the collision is unavoidable, lift the hood and fire off external airbags in front of the A pillars to make the best of the situation.

Bicyclists are another group of vulnerable trafficants, where much less has been done so far, but where the aforementioned avoidance and mitigation principles for pedestrians can be applied once a good detection and tracking filter exists. One of the few publications in this field [10] applies computer vision algorithms based on trained classifiers. In this contribution, we use a model-based approach by studying the most characteristic features of a common bike: the two wheels. The proposed algorithm first processes image data in order to extract ellipsoids, see Figure 1. These ellipsoid parameters are then input to a tracking filter based on a bicycle model. The position and velocity state of the filter can be used for collision avoidance decision algorithms as done in conventional automotive CA systems [7]. Even more interesting is to use the articulation of the front wheel and the roll angle of the bicycle in the decision process. These two states show the intention of the driver, and is a promising indicator for assessing potential risk and issue an early conflict warning.

To the best of authors knowledge there are few publications focused on bicycle and cyclist tracking and collision avoidance. One of the few publications is [10]. However,



Figure 1. The green ellipses indicate measurements obtained from the two bike wheels. The ellipse parameters are later fed through a particle filter framework in order to estimate the bicycle state.

a lot of interesting algorithms are proposed for pedestrian detection and tracking using stereo vision and infrared cameras which can be tailored for the cyclists. In [3], [13] and [4] algorithms for pedestrian detection and tracking are proposed and in [15] an overview on vision-based pedestrian detection for intelligent vehicles is provided.

This article is structured as follows: in the subsequent section we describe a dynamic model for the bicycle's motion. In the third section we will describe a method for detecting ellipses in the images and we derive the mapping between extracted parameters of an ellipse in the image to the pose of the projected wheel. We extend these methods for a bicycle with two wheels. A particle filter is formulated for tracking a bicycle in simulation, using three scenarios with a moving camera. The tracking performance in simulation is evaluated against three simulated maneuvers in Section IV. A particle filter [6] is formulated for tracking of a bicycle using image data captured by a stationary camera and evaluated in Section V. The concluding remarks will be presented in the last section.

II. DYNAMIC MODEL

A. Coordinate Systems

In a collision avoidance application the camera is mounted on a moving platform. The pose of the platform relative to the global frame may be measured or estimated using complementary sensors. In this paper we assume that the platform is moving with known speed in the XZ -plane and that it has zero angular velocity. Therefore the global coordinate system coincides with the camera's coordinate system. In future work the pose of the camera can be merged into the algorithm as an input. In other words, we are interested in the relative pose of

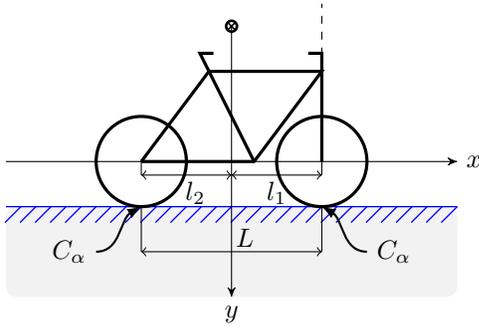


Figure 2. Illustration of the coordinate system and the bicycle parameters. The wheelbase L and the distance of center of gravity to the wheel centers is denoted by l_1 and l_2 . The y -axis goes through the center of gravity and the x -axis goes through the wheel centers.

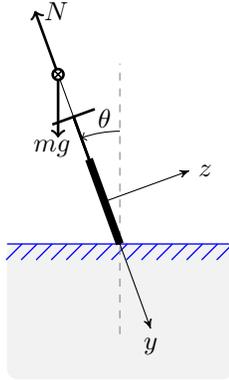


Figure 3. Illustration of the inclination θ of the bicycle. The inclination angle can be calculated using Newton's second law of motion. The gravitational force is denoted by mg and the reaction force of the ground is denoted by N .

the camera's coordinate system and the bicycle's coordinate system which will be introduced here.

The origin of the bicycle's coordinate system is attached to a point below its center of gravity and along a line connecting the centers of the two wheels. The x -axis of the bicycle's coordinate system is oriented along the bicycle's frame and passes through the centers of both wheels. The y -axis is pointing downwards and the z -axis is pointing in the direction perpendicular to the plane containing the rim of the rear wheel. See Figure 2 and Figure 3.

B. Projected Bicycle

In order to describe the bicycle's pose in the global coordinate system we will parametrize the bicycle. The bicycle is parametrized by the position of the origin of its coordinate system $P_c = (X_c, Y_c, Z_c)$, the normal vector to the rear wheel $n_r = (n_{1r}, n_{2r}, n_{3r})^T$, the steering angle δ , the slope of bicycle's track α , and its wheel base L . The caster angle of the front wheel is neglected and the rotation axis of the front wheel is assumed to be vertical to the flat ground along the global Y -axis. The normal vector to the front wheel can be calculated using the steering angle δ . The error for small inclination angle θ and small slopes is very small.

The bicycle's inclination in the roll direction while turning

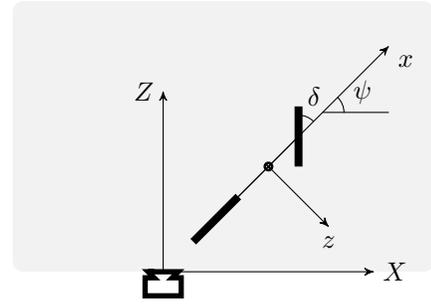


Figure 4. An extended bicycle model is used as motion model where ψ and δ are shown in this figure. The orientation of the camera at the origin of the global coordinate system is shown.

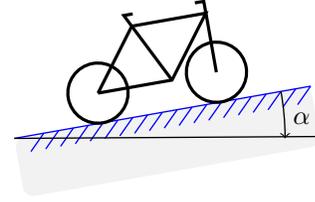


Figure 5. The slope of the bicycle's track is denoted by α

is denoted by θ and is given as

$$\tan \theta = \frac{v_x \dot{\psi}}{g}, \quad (1)$$

where g is the gravitational acceleration, v_x is the longitudinal speed of the bicycle and $\dot{\psi}$ is the yaw rate of the bicycle. Equation (1) can be derived by writing Newton's second law of motion in the lateral direction and in the vertical direction, respectively. The speed of the bicycle in the lateral direction will be denoted by v_z .

By assembling the aforementioned parameters of the bicycle's pose into one vector the state vector

$$x = (\psi, X_c, Z_c, \dot{\psi}, v_x, v_z, \delta, Y_c, \alpha)^T \quad (2)$$

is created and will be used later on to establish a motion model.

Using the state variables stated in (2), the position of the center of both wheels and their respective normal vector is calculated according to

$$P_r = \begin{pmatrix} X_c - l_2 \cos \psi \cos \alpha \\ Y_c + l_2 \sin \alpha \\ Z_c - l_2 \sin \psi \cos \alpha \end{pmatrix} \quad (3)$$

$$P_f = \begin{pmatrix} X_c + l_1 \cos \psi \cos \alpha \\ Y_c - l_1 \sin \alpha \\ Z_c + l_1 \sin \psi \cos \alpha \end{pmatrix} \quad (4)$$

$$n_r = (\sin \psi \cos \theta, \sin \theta, -\cos \psi \cos \theta)^T \quad (5)$$

$$n_f = (\sin(\psi + \delta) \cos \theta, \sin \theta, -\cos(\psi + \delta) \cos \theta)^T. \quad (6)$$

We assume that the radius of the wheel, r , l_1 and l_2 , where $l_1 + l_2 = L$, are known.

C. Motion Model

A motion model using the bicycle model [1] is created. The nonlinear state space representation of the motion model in continuous-time is given below

$$\dot{x} = \begin{pmatrix} \dot{\psi} \\ (\cos(\psi)v_x - \sin(\psi)v_z) \cos \alpha - V_X \\ (\sin(\psi)v_x + \cos(\psi)v_z) \cos \alpha - V_Z \\ \dot{\psi} \\ 0 \\ -\frac{2C_\alpha v_z}{mv_x} - \frac{v_x + C_\alpha(l_1 - l_2)}{mv_x} \dot{\psi} + \frac{C_\alpha}{m} \delta \\ u \\ v_x \sin \alpha - r \sin(\theta) \dot{\theta} \\ 0 \end{pmatrix} \quad (7)$$

where

$$\ddot{\psi} = -\frac{C_\alpha(l_1 - l_2)}{I_z v_x} v_z - \frac{C_\alpha(l_1^2 + l_2^2)}{I_z v_x} \dot{\psi} + \frac{C_\alpha l_1}{I_z} \delta \quad (8)$$

and V_X and V_Z are X and Z component of the host vehicle's speed. The modeling parameters are listed in Table I and are illustrated in Figures 2–5. The cornering stiffness of the tires is denoted by C_α , the bicycle's mass is denoted by m . The rotational inertia of the bicycle is shown by I_z . In this motion model the steering angle δ is driven by the input u .

Parameters	C_α	m	I_z	l_1	l_2	r
Value	1000	80	26	0.60	0.49	0.32
Unit	N	kg	kgm^2	m	m	m

III. MEASUREMENT MODEL

The measurement model describes the mapping between the state variables and the measured values. We are using a camera as a sensor and the measured signals which will be used for state estimation are the parameters of the two extracted ellipses from the image of the two bicycle wheels. In this section we describe how the measurement vector y relates to the state variables x and how the ellipses are extracted from the image data.

The measurement vector,

$$y_k = (a_f, b_f, \text{Re}\{z_{0f}\}, \text{Im}\{z_{0f}\}, \phi_f, a_r, b_r, \text{Re}\{z_{0r}\}, \text{Im}\{z_{0r}\}, \phi_r)^T + e, \quad (9)$$

is composed of parameters of the two ellipses corresponding to the front wheel and the rear wheel, respectively. An ellipse can be parametrized using two semi-axes, a and b , and the position of its center, $z_0 = \text{Re}\{z_0\} + i\text{Im}\{z_0\}$ and the angle ϕ which denote the angle between the largest semi-axis a and the real axis in the complex image plane. The subscripts f and r denote the association with the front and the rear wheel, respectively.

A. Projected Wheel

In this section we derive the equations of the projection from 3D pose of a bicycle wheel (parametrized by the position of the wheel's center and a normal vector) to the corresponding ellipse in the image plane. Let $n = (n_1, n_2, n_3)^T$ denote the normal vector to the wheel and let $P_0 = (X_0, Y_0, Z_0)$ denote the position of the center of the wheel. See Figure 6 for an illustration. For a point (X, Y, Z) belonging to the bicycle's wheel with radius r we have the following relations,

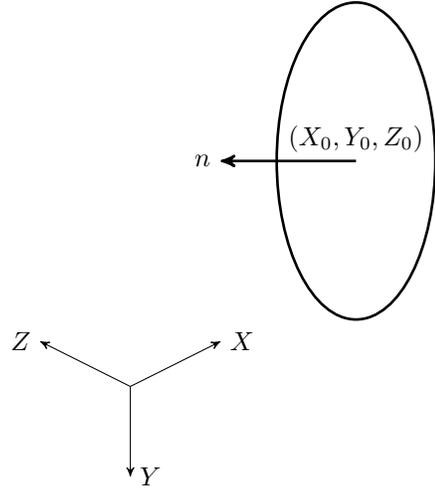


Figure 6. The position of the center of the wheel and the normal vector to its plane are used to describe the ellipse parameters.

$$(X - X_0)^2 + (Y - Y_0)^2 + (Z - Z_0)^2 - r^2 = 0, \quad (10)$$

$$n^T \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix} = 0, \quad (11)$$

where the latter equation can be written as

$$n_3 Z(X, Y) = -n_1 X - n_2 Y + n_1 X_0 + n_2 Y_0 + n_3 Z_0. \quad (12)$$

Using the pinhole camera model, a point (X, Y, Z) is projected onto the image coordinate $z = (x + iy)$ according to

$$z = \frac{(X + iY)f}{Z(X, Y)} = \frac{n_3(X + iY)f}{-n_1 X - n_2 Y + \bar{Z}_0}, \quad (13)$$

where

$$\bar{Z}_0 = n_1 X_0 + n_2 Y_0 + n_3 Z_0. \quad (14)$$

The projection of the points on the edge of the wheel onto the image plane gives us a conic section. This can be represented as a second degree polynomial or equivalently using homogeneous coordinates as

$$(\text{Re}\{z\} \ \text{Im}\{z\} \ 1) \ C \begin{pmatrix} \text{Re}\{z\} \\ \text{Im}\{z\} \\ 1 \end{pmatrix} = 0, \quad (15)$$

where

$$C = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix} \quad (16)$$

is the symmetric matrix and its components are given by

$$c_{11} = n_1^2(D^2 - r^2) - 2n_1 X_0 \bar{Z}_0 + \bar{Z}_0^2, \quad (17)$$

$$c_{22} = n_2^2(D^2 - r^2) - 2n_2 Y_0 \bar{Z}_0 + \bar{Z}_0^2, \quad (18)$$

$$c_{33} = n_3^2(D^2 - r^2)f^2 - 2n_3 Z_0 \bar{Z}_0 f^2 + \bar{Z}_0^2 f^2, \quad (19)$$

$$c_{12} = n_1 n_2 (D^2 - r^2) - (n_2 X_0 + n_1 Y_0) \bar{Z}_0, \quad (20)$$

$$c_{13} = n_1 n_3 (D^2 - r^2)f - (n_1 Z_0 + n_3 X_0) \bar{Z}_0 f, \quad (21)$$

$$c_{23} = n_2 n_3 (D^2 - r^2)f - (n_3 Y_0 + n_2 Z_0) \bar{Z}_0 f, \quad (22)$$

where

$$D^2 = X_0^2 + Y_0^2 + Z_0^2. \quad (23)$$

The resulting conic is an ellipse. In general, a conic is an ellipse if the corresponding matrix C has two positive singular values and one negative [9]. All z points calculated from (13) will form an ellipse if (X, Y, Z) belong to the set described by (10) and (11). This can be verified by inserting z from (13) into (15) for points on the bicycle wheel, such that

$$\begin{aligned} n_3^2(X - X_0)^2 + n_3^2(Y - Y_0)^2 + \\ (n_1X + n_2Y - n_1X_0 - n_2Y_0)^2 - r^2 = 0. \end{aligned} \quad (24)$$

The center of the ellipse, z_0 is given by

$$\begin{pmatrix} \text{Re}\{z_0\} \\ \text{Im}\{z_0\} \end{pmatrix} = \bar{C}^{-1} \begin{pmatrix} -c_{13} \\ -c_{23} \end{pmatrix}, \quad (25)$$

where

$$\bar{C} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}. \quad (26)$$

Let λ_1 and λ_2 be the eigenvalues of \bar{C} , and V_1 and V_2 be their respective normalized eigenvectors. The length of the semi-axes and their directions, introduced in Section III-B, are given by

$$a = \sqrt{\frac{-\det C}{\lambda_1 \det \bar{C}}} \quad (27)$$

$$b = \sqrt{\frac{-\det C}{\lambda_2 \det \bar{C}}} \quad (28)$$

$$\phi = \arcsin(V_2(1)), \quad (29)$$

where a, b, z_0, ϕ are the parameters of the ellipses that are given by the projection of the front and the rear wheel into the camera. The measurement model can therefore be written in nested format, where $y = h(P_r, P_f, n_r, n_f) + e_1$ is given by (9)–(29). The mapping between P_r, P_f, n_r and n_f and the state variable x is given by (3)–(6) respectively. We model the noise as additive white Gaussian noise. This assumption can be further evaluated using real data in a real world implementation of the algorithm.

B. Ellipse Extraction

Extracting ellipses from an image is a well studied problem within the computer vision community. Numerous approaches have been suggested e.g. Hough transform approaches [14], point based approaches [8] and approaches based on statistical moments [5].

For our real world experiments a simplified setup has been used, i.e. a static camera, a dark surrounding, a single bicycle and reflective rims on the tires. These simplifications allow us to extract points on the tires by background subtraction and subsequent thresholding followed by standard morphological operations. As a first step, points for connected components are given the same label followed by a second step, where components with different labels are merged based on proximity. This results in a number of clusters each of which have a unique label.

Motivated by the knowledge that we are looking for only one bike, thus expecting two wheels, we extract the two largest clusters. Given points for these two clusters we estimate the

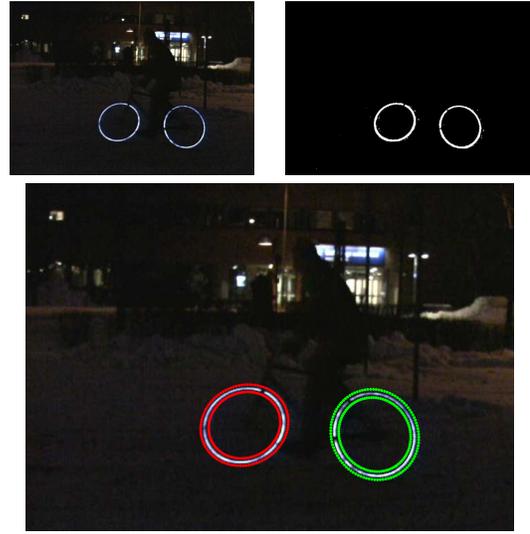


Figure 7. Ellipse extraction. Top left: Query image, Top right: Query image after background subtraction. Bottom: Ellipses plotted with 0.9 and 1.1 times the estimated size, the actual ellipse estimated are halfway between the lines.

ellipses corresponding to the wheels by the method by Halir and Flusser [8]. The process of extracting ellipses from images is illustrated in Figure 7.

IV. NUMERICAL SIMULATION

In order to evaluate the feasibility of the algorithm the motion model introduced in Section II is simulated for three trajectories which are all listed here and are illustrated in Figure 8.

- 1) Lane change in front of the host vehicle
- 2) Crossing the host vehicles path with constant velocity
- 3) Left turn in front of the host vehicle

It should be noted that for the purpose of the simulation the motion model is simulated in its most general form while another motion model can be used for the tracking purposes.

Measurements are created using the measurement model introduced in Section III. We added noise to the simulated measurements according to

$$e \sim \mathcal{N}(0, R_1) \quad (30)$$

where

$$R_1 = 5 \times 10^{-7} I_{10 \times 10}. \quad (31)$$

A. Setup

The extended Kalman filter is used for tracking of the bicycle in simulation. The extended Kalman filter uses a simplified motion model with the state

$$x = (\psi, X_c, Z_c, \dot{\psi}, v_x, v_z, \delta)^T \quad (32)$$

and dynamics

$$\dot{x} = \begin{pmatrix} \dot{\psi} \\ (\cos \psi v_x - \sin \psi v_z) - V_X \\ (\sin \psi v_x + \cos \psi v_z) - V_Z \\ \ddot{\psi} \\ 0 \\ -\frac{2C_\alpha v_z}{m v_x} - \frac{v_x + C_\alpha(l_1 - l_2)}{m v_x} \dot{\psi} + \frac{C_\alpha}{m} \delta \\ 0 \end{pmatrix} \quad (33)$$

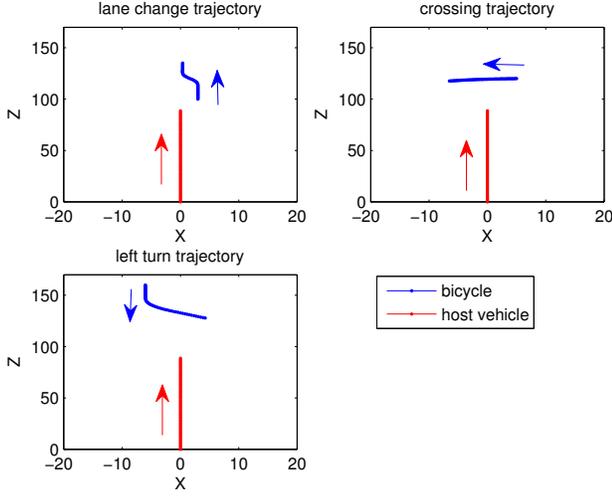


Figure 8. In order to create a stream of frames which will be further on used as measurements a bicycle model is created in MATLAB. The bicycle model is simulated for three trajectories which are labeled accordingly.

where

$$\ddot{\psi} = -\frac{C_\alpha(l_1 - l_2)}{I_z v_x} v_z - \frac{C_\alpha(l_1^2 + l_2^2)}{I_z v_x} \dot{\psi} + \frac{C_\alpha l_1}{I_z} \delta. \quad (34)$$

In this formulation of the motion model, the steering angle δ is modeled as a disturbance. The measurement model is the same as what we introduced in Section III save for the fact that the slope α , is set to zero. The inclination of the bicycle θ is not a state variable and is calculated using (1) in the nested measurement function definition (9)-(29).

The motion model is discretized according to

$$x_{k+1} = f(x_k) + e_k \quad (35)$$

$$y_k = h(x_k) + w_k \quad (36)$$

and the noise and the initial states is assumed to be according to

$$e_k \sim \mathcal{N}(0, R_2) \quad (37)$$

$$w_k \sim \mathcal{N}(0, Q) \quad (38)$$

$$x_0 \sim \mathcal{N}(\bar{x}_0, P_{x0}). \quad (39)$$

where the \bar{x}_0 is the true initial state. The noise covariance matrices are chosen according to

$$Q = T^2 \text{Diag}(0, 0, 0, 0, 0, 0, 0.01, 0) \quad (40)$$

$$R_2 = 2 \times 10^{-5} I_{10 \times 10} \quad (41)$$

where T is the sampling time in the extended Kalman filter implementation which in this simulation was set to 0.1s. Four elements of x_0 , that is, $\psi_0, X_{c0}, Z_{c0}, \delta_0$, can be numerically calculated using the inverse of the mapping introduced in the measurement step with a number of iterations.

B. Results

Tracking performance for the three trajectories is presented in Figures 9–11. Tracking performance is satisfactory with respect to the lateral position, but not equally well in the longitudinal direction. It is known that the depth estimates

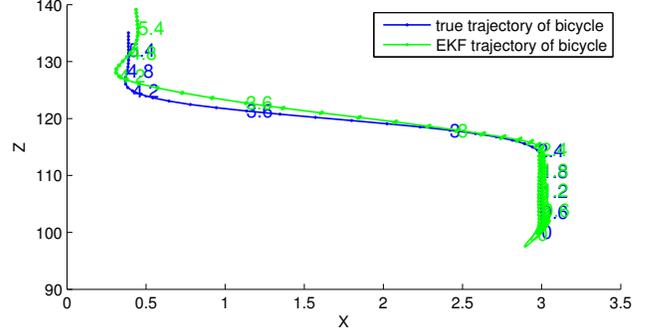


Figure 9. The lane change maneuver: The true and estimated trajectory by the extended Kalman filter and the confidence of the estimates are shown for $t=0$ s up to 6 s.

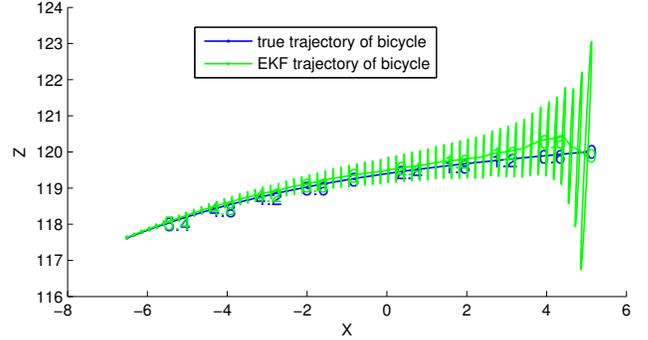


Figure 10. The crossing maneuver: The true and estimated trajectory by the extended Kalman filter and the confidence of the estimates are shown for $t=0$ s up to 6 s.

using image sensors under forward motion are not very precise [12]. In order to further evaluate the filtering performance the true value for the states and the estimated value from the extended Kalman filter implementation are overlaid in Figure 12 for the lane change trajectory. The state estimates produced by the extended Kalman filter were equally good for the other two trajectories.

Evaluation of simulation results and study of the simulation parameters showed the following shortcomings of the algorithm and created new open issues:

- The assumptions for this simulation such as additive Gaussian noise and the covariance of the measurement vector, the ellipse parameters, is to be verified by real world test results.
- The tracking algorithm showed to be very sensitive to the presence of slope in the simulated trajectory. In the real world test scenarios there where some degree of inclination of the ground is expected, a remedy is needed.
- Due to the nonlinearities in the measurement model h and the motion model the convergence of the extended Kalman filter is affected by the initial state.
- The extended Kalman filter did not perform well when realistic noise and covariances were used and its good performance was limited to the narrow distribution of noise and covariance shown in (31), (40) and (41). Therefore a particle filter was used for the real world experiment. The tracking algorithm based on particle

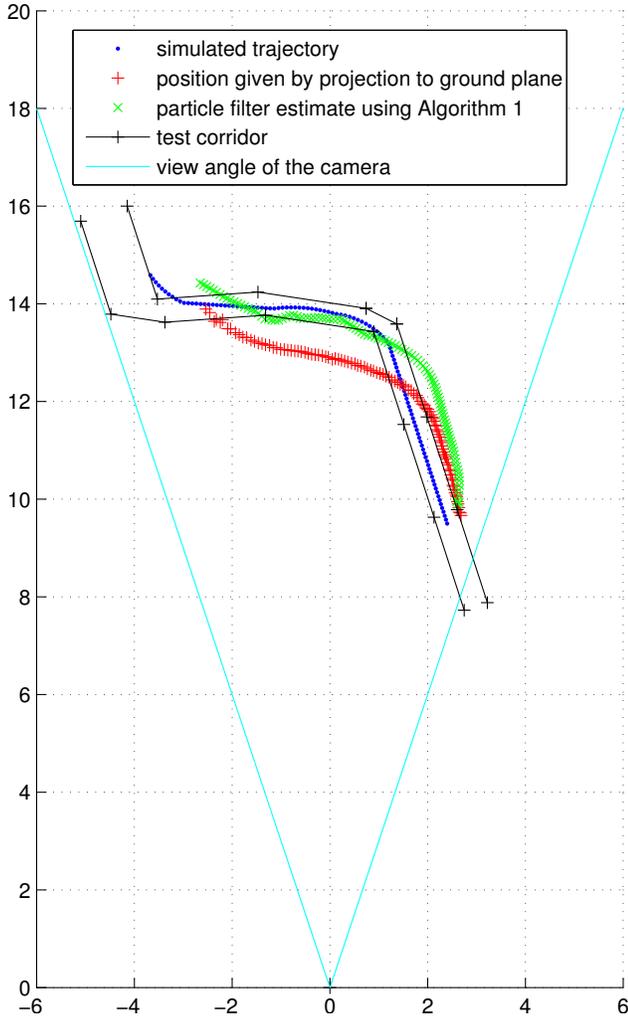


Figure 13. Estimated position for the S-turn maneuver using real image data is shown. The particle filter estimates follows the experiment's virtual corridor very well.

The noise covariance matrices are chosen according to

$$Q = T^2 \text{Diag}(0.01, 0.01, 0.01, 0.01, 0.01, 0.01, 400, 0), \quad (49)$$

$$R_3 = 0.36 \times I_{10 \times 10}, \quad (50)$$

where $T = 0.04s$.

In order to evaluate the filtering performance knowledge of the true trajectory of the bicycle and the other state variables such as the steering angle and the speed of the bicycle would have been beneficial. In order to create reference values to compare the estimated states with, a track was drawn on the ground and the bicycle was ridden within a track of 50 cm width. By measuring the spacial distances to the camera a rough estimate of the bicycle's trajectory was achieved. In order to create a reference for the other states the bicycle model was simulated and the steering input that would move the bicycle within the trajectory was used as the reference for the steering input. Other states are also estimated using the bicycle model in the same fashion. Tracking performance for the two trajectories is presented in Figures 13-16. The

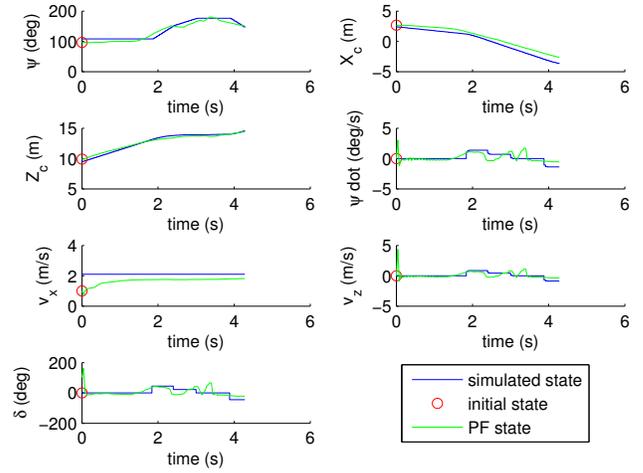


Figure 14. Estimated states for the S-turn maneuver using real image data is compared with the quasi-true states created using simulation.

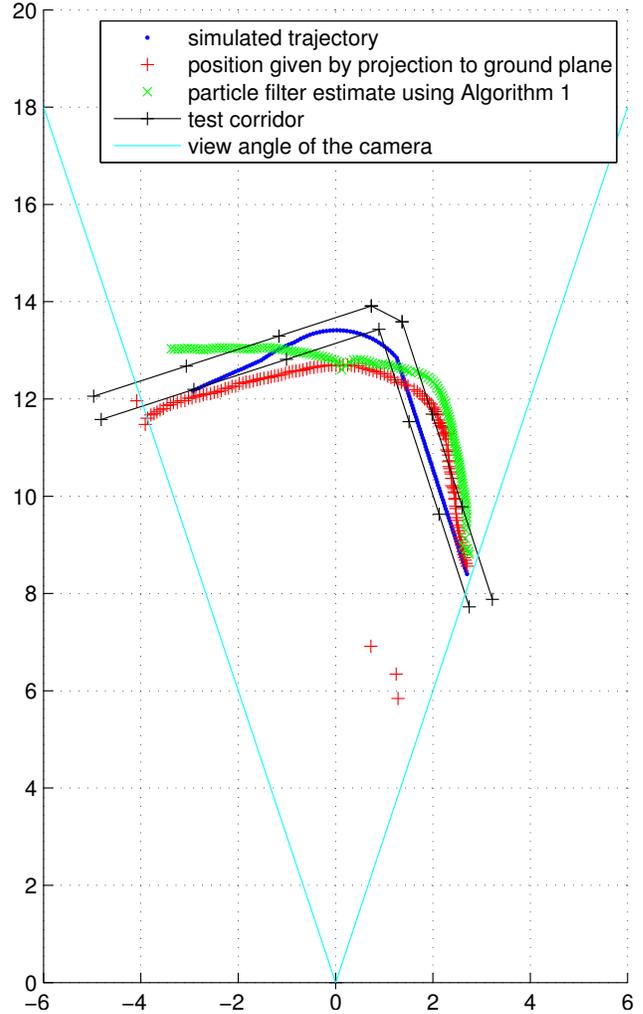


Figure 15. Estimated position for the L-turn maneuver using real image data is shown. The particle filter estimates follows the experiment's virtual corridor very well.

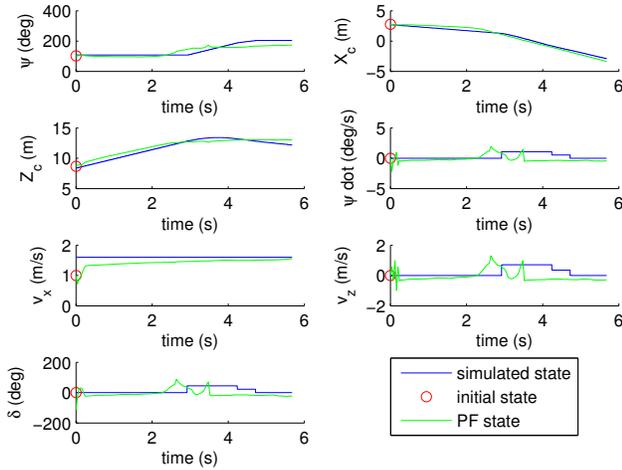


Figure 16. Estimated states for the L-turn maneuver using real image data is compared with the quasi-true states created using simulation.

results presented in this work should be viewed as a proof of concept for tracking a bicycle using ellipse extraction. There are of course other competing approaches for the problem such as [10] where the cyclist and the bicycle are treated as an extended target. The advantage of our approach which can be further enhanced if combined together with other approaches is that the cyclist's intention to turn can be detected from the steering input before the speed vector is affected. Another usage of this result is to combine the detection of the steering input with a simpler motion model in a jump Markov linear model framework.

VI. DISCUSSION

A bicycle tracking algorithm using the particle filter and ellipse extraction is presented. The algorithm consists of an ellipsoid detection and estimation algorithm and a particle filter. A simulation study of three critical single target scenarios is presented, and despite the nonlinear state space equation and measurement model the algorithm is shown to produce excellent state estimates. An experiment using a stationary camera and the particle filter for state estimation is performed and has shown encouraging results. Future work includes how to compute a covariance uncertainty from the ellipse extraction algorithm, and how to handle the association and track handling steps in a multi-target tracking framework, in particular the unique ellipsoid pairing problem.

ACKNOWLEDGEMENT

The authors gratefully acknowledge funding from the Swedish Research Council for the project Extended Target Tracking. Further fundings have been received through EL-LIIT, the Strategic Area for ICT research, funded by the Swedish Government and from the EC's 7th Framework Programme (FP7/2007-2013), grant agreement 247947 (GAR-NICS).

REFERENCES

- [1] K. J. Åström, R. E. Klein, and A. Lennartsson. Bicycle dynamics and control. *IEEE Control Systems Magazine*, 25(4):26–47, August 2005.
- [2] D. Bernstein, editor. *Special Issue on Active Safety*, volume 30. Control Systems Magazine, IEEE, aug. 2010.

- [3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the 18th Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, San Diego, CA, USA, volume 1 of *IEEE*, pages 886–893 vol. 1, 2005.
- [4] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Proceedings of the 22nd Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, Miami, Florida, USA., IEEE, pages 304–311, 2009.
- [5] P.-E. Forssén and A. Moe. View matching with blob features. *Image and Vision Computing, Canadian Robotic Vision Special Issue*, 27(1-2):99–107, January 2009.
- [6] N.J. Gordon, D.J. Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, April 1993.
- [7] F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, first edition, 2010.
- [8] R. Halif and J. Flusser. Numerically stable direct least squares fitting of ellipses. In *Proceedings of the Sixth International Conference in Central Europe on Computer Graphics and Visualization'98 (WSCG'98)*, Plzen - Bory, Czech Republic, pages 125–132, 1998.
- [9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
- [10] C. Hyunggi, P. Rybski, and W. Zhang. Vision-based bicycle detection and tracking using a deformable part model and an EKF algorithm. In *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, 2010.*, pages 1875–1880.
- [11] L. Qingfen, F. Tjärnström, J. Roll, and B. Wass. Developing a far infrared based night-vision system with pedestrian detection. In *Proceedings of the VDI Optische Technologien in der Fahrzeugtechnik, Leonberg, Germany.*, June 2008.
- [12] A. Vedaldi, G. Guidi, and S. Soatto. Moving forward in structure from motion. In *Proceedings of the 20th Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, Minneapolis, Minnesota, USA, IEEE, jun. 2007.
- [13] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *9th IEEE International Conference on Computer Vision (ICCV 2003)*, 14-17 October, Nice, France, pages 734–741 vol.2, 2003.
- [14] H. K. Yuen, J. Illingworth, and J. Kittler. Detecting partially occluded ellipses using the Hough transform. *Image Vision Computing*, pages 31–37, 1989.
- [15] L. Zhenjiang, W. Kurfeng, L. Li, and W. Fei-Yue. A review on vision-based pedestrian detection for intelligent vehicles. In *IEEE International Conference on Vehicular Electronics and Safety, (ICVES) 2006, Shanghai, China*, pages 57–62, 2006.