

Vehicle Tracking at Urban Intersections Using Dense Stereo

Alexander BARTH¹, David PFEIFFER and Uwe FRANKE
*Daimler AG, Group Research and Advanced Engineering,
Sindelfingen, Germany*

Abstract. A new approach for vehicle tracking at urban intersections based on stereo-vision is proposed. Objects are represented as rigid 3D point clouds and tracked by means of extended Kalman filtering. In this contribution we combine the advantages of a generic feature-based 3D point cloud model with vehicle specific geometrical and kinematical constraints to estimate the pose and motion state of oncoming vehicles at intersections. Real-time dense stereo disparity maps provide new opportunities in reconstruction of the 3D driving scene. An efficient and compact *Stixel World* representation is computed that segments the scene into drivable freespace and obstacles on the ground. Based on this data we derive the silhouette of an object in the image and constrain its pose in space during turning maneuvers. The system has been successfully tested on various real-world scenarios and runs in real-time on VGA images in our demonstration car.

Keywords. Vehicle Tracking, Driver Assistance Systems, Dense Stereo Vision, Kalman filtering

Introduction

Detecting and tracking other traffic participants at urban intersections has attracted special attention in the intelligent vehicle domain due to the large number of accidents still occurring day-to-day. Monitoring moving objects at such accident hotspots with stationary cameras, typically from elevated position, has been addressed by many researchers in the past, e.g. [1,2,3].

Previous work on vision-based vehicle tracking from a moving platform mainly concentrates on highway scenarios. However, precise information on the behavior of the oncoming and cross traffic at intersections provides a fundamental basis for future driver assistance and safety applications.

In general, one can distinguish between geometric and feature-based vehicle tracking approaches. Geometric approaches try to fit a geometric model, e.g. a cuboid [4,5] or more sophisticated vehicle models [3], to the given sensor data. Such approaches perform well as long as the model is a sufficient approximation of the real object, and the data is reliable.

¹Corresponding Author: Alexander Barth, Daimler AG, GR/PAP, HPC 050/G024, Sindelfingen, Germany; E-mail: alexander.barth@daimler.com.

Feature-based methods, as for example [6,7,8,9], model an object by a set of characteristic features, e.g. gray value or color statistics, edges, corners, etc. These statistics can be determined online and are usually more flexible than geometric models. A drawback of such methods is that, without any geometric constraints, detected objects may be incomplete, i.e., parts of the object that are not covered by a feature are missing, or features belonging two different physical objects are merged, i.e., the separation between two close objects can fail.

In [9], we have proposed a feature-based vehicle tracking approach that simultaneously estimates the pose and motion parameters of a rigid point cloud, representing the vehicle’s shape. Object points are detected and grouped on the Gestalt principle of common fate, i.e., points of common motion are likely to belong to the same object. The system has been successfully applied to predict the driving path of oncoming vehicles at country road scenes.

In this contribution, we will extend this feature-based point cloud model by a geometric model to overcome two fundamental assumptions made in the original approach. First, it is not longer required that the vehicle dimension can be reconstructed sufficiently well from the observed point cloud. Secondly, a good estimate of the vehicle’s center rear axle is essential to predict the object pose at highly dynamic turn maneuvers. However, it cannot be observed from motion as long as the vehicle is moving straight or not moving at all. The geometric model allows for adding direct measurements for the center rear axle position based on an object’s image silhouette and dense stereo disparity maps.

To be able to process the large amount of dense stereo data in real-time, an efficient *Stixel World* representation, which has been recently proposed by Badino et al [10], is used. This representation models both the drivable freespace and its boundaries, corresponding to obstacles over ground.

In Section 1, we will first briefly summarize our feature-based vehicle tracking approach and then extend this model in Section 2 by geometric constraints. Section 3 gives a short overview on the stixel representation, that is used in Section 4 and 5 for an initial pose refinement and to derive additional filter measurements respectively. Experimental results on real-world scenes are shown and discussed in Section 6.

1. Feature-Based Vehicle Tracking Approach

The object model consists of a state vector \mathbf{x} , including pose and motion parameters of an observed object, and a rigid 3D point cloud Θ :

$$OBJ \models \{\mathbf{x}, \Theta\} \tag{1}$$

The main idea is to estimate \mathbf{x} based on the 3D displacement of the rigid point cloud in a stereo image sequence using an extended Kalman filter [11]. Fig. 1(a) gives an overview on the general system.

Formally, vehicles are modeled as rigid body, whose pose relative to the ego vehicle is defined by the transformation of a local *object coordinate system* with respect to the *ego coordinate system* attached to the ego vehicle. The object pose can be fully described by an (arbitrary) reference point on the object, \mathbf{P}_{ref} , representing the object origin, and the Euler angle ψ , indicating the rotation around the height axis (see Fig. 1(b)).

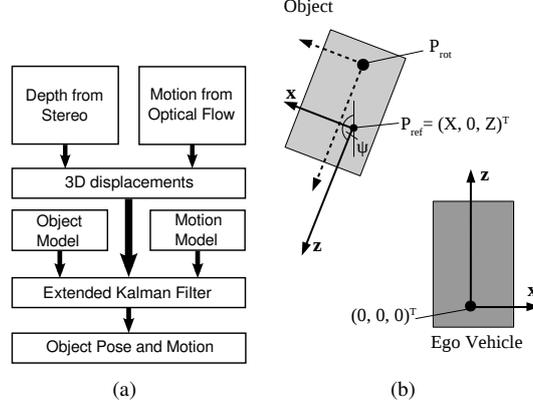


Figure 1. (a) System overview. (b) Object coordinate systems and pose parameters.

Incorporating vehicle specific characteristics, it is further assumed that the Z-axis of the object coordinate system is ideally aligned with the longitudinal axis of the vehicle, i.e., ψ corresponds to the moving direction. Lateral movements are restricted to circular path motion based on a simplified bicycle motion model. This motion model is parametrized by velocity v and acceleration \dot{v} in the moving direction as well as the yaw rate $\dot{\psi}$, i.e., the change of orientation. The model further requires the object origin to be located at the center rear axle of the vehicle. We will denote this characteristic point as *rotation point*, P_{rot} , in the following. Its relative position to the reference point is typically not known at initialization and has to be estimated.

Pose and motion parameters are summarized in the following state vector:

$$\mathbf{x} = \left[\underbrace{{}^e X_{ref}, {}^e Z_{ref}, {}^o X_{rot}, {}^o Z_{rot}}_{pose}, \underbrace{\psi, v, \dot{\psi}, \dot{v}}_{motion} \right]^T. \quad (2)$$

with reference point ${}^e \mathbf{P}_{ref} = [{}^e X_{ref}, 0, {}^e Z_{ref}]^T$ in ego coordinates and rotation point ${}^o \mathbf{P}_{rot} = [{}^o X_{rot}, 0, {}^o Z_{rot}]^T$ in object coordinates. For simplicity, a planar ground is assumed, i.e., ${}^e Y_{ref} = {}^o Y_{rot} = 0$ independent of the lateral and longitudinal position. This model can be easily extended by incorporating height information on the 3D geometry of the ground, e.g., using a height varying road model as proposed in [12]. The ego-motion is estimated and compensated, before the filter prediction step, using the method proposed in [13].

The object dimension is not explicitly modeled. Instead it is assumed that the object's shape is sufficiently represented by a set of 3D points. Each point ${}^o \mathbf{P}_m = [{}^o X_m, {}^o Y_m, {}^o Z_m]^T \in \Theta$, $1 \leq m \leq M$, has a fixed position within the object coordinate system and can be observed in terms of an image coordinate $\langle u_m(t), v_m(t) \rangle$ and stereo disparity $d_m(t)$ at time t . These point measurements build the measurement vector \mathbf{z}_p with

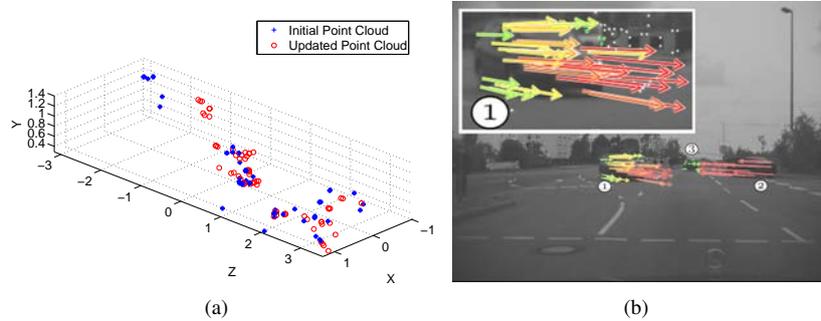


Figure 2. (a) Object point cloud at initialization and after 10 time steps. The initial point cloud, spread over 6 m in longitudinal direction, is successively refined. (b) Example 6D-Vision motion field used for object detection. The color encodes lateral velocity (red fast, green slow). Points at the front of vehicle no. 1 show larger lateral velocities compared to points at the rear, since the vehicle is turning.

$$\mathbf{z}_p(t) = [u_1(t), v_1(t), d_1(t), \dots, u_M(t), v_M(t), d_M(t)]^T. \quad (3)$$

The projection of each point onto the image plane is tracked using a feature tracker, e.g. the well-known KLT-tracker [14], to be able to reassign measurements of the same 3D point over a sequence of images. The nonlinear measurement model directly follows from the transformation between object and camera coordinates and the well-known projection equations of a finite perspective camera.

Since the exact position of a given object point is typically not known at initialization, it has to be estimated from the noisy measurements. For real-time applicability, the problem of motion estimation is separated from the problem of shape reconstruction. Thus, instead of estimating shape and motion simultaneously by integrating Θ into \mathbf{x} , the point positions are refined outside the Kalman filter.

For each object point oP_m , observed several times in terms of ${}^o\tilde{P}_m(t)$, a maximum likelihood estimation, assuming uncorrelated measurements and zero-mean Gaussian measurement noise [15], is given by

$${}^oP_m(t+1) = \left[\sum_{j=t_m}^t C_m^{-1}(j) \right]^{-1} \sum_{j=t_m}^t C_m^{-1}(j) {}^o\tilde{P}_m(j) \quad (4)$$

where $C_m(t)$ denotes the 3×3 covariance matrix of ${}^o\tilde{P}_m(t)$, and t_m the discrete time step point oP_m has been added to the model. An example for an initial noisy object point cloud and the same point cloud refined over 10 time steps is given in Figure 2(a).

Moving objects are detected from the 3D motion field of a number of feature points, distributed over the whole image (see Figure 2(b)). The motion vectors are estimated based on the principle of *6D-Vision*, i.e., the pointwise fusion of depth and motion by Kalman filtering [16].

The vehicle tracking is initialized from a cluster of points moving in the same direction with equal velocity. The average velocity vector gives the moving direction (initial orientation) and the initial vehicle speed. Both the reference point and the rotation point are initialized at the centroid of the initial point cloud.

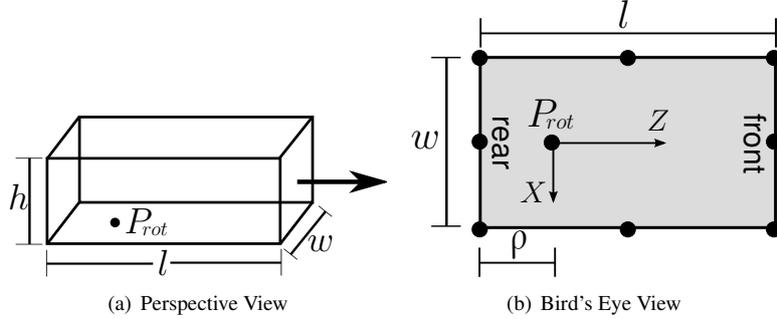


Figure 3. Geometric box model with the rotation point at the center rear axle located at a constant distance ρ from the vehicle's rear. The object corners as well as the side centers have a fixed position with respect to the rotation point describable in terms of w , h , l , and ρ .

2. Geometric Object Model

We now extend the object model in Eq. 1 by a cuboid D , yielding

$$OBJ \models \{\mathbf{x}, \Theta, D\} \quad (5)$$

with $D = [w, l, h]^T$, approximating the object dimension in terms of width w , length l , and height h , independent of the currently observed point cloud.

This has several advantages. A cuboid covers a certain region in space or on the image plane that can be used to associate new points to an object. Decoupling the object dimension from the point cloud is extremely helpful if parts of the point cloud are occluded or lost by the feature tracker. In addition, restricting the cuboid dimension to the expected size of road vehicles allows for rejecting points with similar motion pattern belonging to different, close-by objects in dense traffic scenes.

The cuboid model is used to integrate a basic semantical meaning on vehicle sides (front, rear, left, right) or characteristic points, e.g. front left corner or center of the right side. The objective is to align the sides of the cuboid with the physical sides of the vehicle. As can be seen in Fig. 3, all points on the cube sides have a fixed position with respect to a virtual *vehicle coordinate system* with the origin at the rotation point. The Z-distance ρ between rotation point and rear side is a given (vehicle specific) constant. In practice, $\rho = 1$ m is a good approximation for most road vehicles.

Accordingly, if the dimension of the object is known correctly, it is straight forward to describe the rotation point relative to a given object side or corner. This means, all corners or sides observable by a sensor can be used to constrain the position of the rotation point.

Based on this idea, we introduce a second vector \mathbf{z}_{rot} of K direct measurements for the rotation point, with $\mathbf{z}_{rot} = [{}^o\tilde{P}_{rot,1}, \dots, {}^o\tilde{P}_{rot,K}]^T$, and concatenate this vector with the point measurement vector in Eq. (3) to the total measurement vector \mathbf{z} with

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_p \\ \mathbf{z}_{rot} \end{bmatrix}. \quad (6)$$

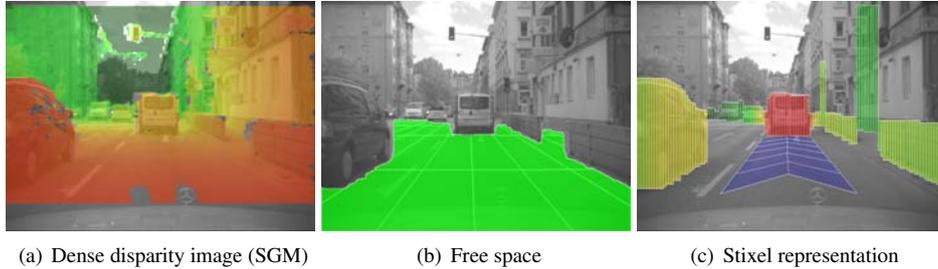


Figure 4. (a) Dense stereo results overlaid on the image of an urban traffic situation. The colors encode the distance, red means close, green represents far. Note that SGM delivers measurements even for most pixels on the road. (b) Free space obtained from an evaluation of the stereo occupancy grid. (c) Stixel representation for this situation, the colors encode the lateral distance to the expected driving corridor shown in blue.

The additional measurements are intended to stabilize the rotation point position and to prevent that the filter wrongly compensates ambiguities due to error-prone point positions in the object model by shifting the rotation point outside the object at rotational movements. This problem will be addressed in detail in the experimental results (see Section 6). The unknown object dimension is updated outside the Kalman filter by low pass filtering in this approach, when dimensional measurements are available.

At this point the extended model is independent of where the actual measurements for the rotation point or cuboid dimensions come from. In the following, we present an example realization based on dense stereo data.

3. Stixel-based Scene Representation

Real-time implementations of dense stereo algorithms, such as Semi-Global Matching (SGM) [17], on dedicated hardware [18] provide significantly more information on the 3D environment compared to sparse stereo methods. The gain in information and precision allows for improved scene reconstruction and object modeling.

However, more information means there is also more data to process. Thus, efficient data representations are beneficial to ease further interpretation. We use a very comprehensive but powerful scene representation that has been recently proposed by Badino et al. [10] for the intelligent vehicle domain.

It is based on the fact that traffic scenes typically consist of a relative planar free space which is limited by 3D obstacles that are nearly perpendicular to the ground. The so called *Stixel World* represents the 3D scene by a set of rectangular sticks, named “stixels”, as shown in Fig. 4(c). Each stixel is defined by its 3D position relative to the camera and stands vertically on the ground, having a certain height. Each stixel limits the free space and approximates the object boundaries. As illustrated in Fig. 4, the Stixel World is created by the following steps:

First, we compute a dense disparity image using SGM. Fig. 4(a) shows that SGM is able to model object boundaries precisely. In addition, the smoothness constraint used in the algorithm leads to smooth estimates in low contrast regions, exemplary seen on the street and the untextured parts of the vehicles and buildings.

In the second step, a stochastic occupancy grid is generated from the stereo disparities using the method presented in [19]. An occupancy grid is a two-dimensional array

which models occupancy evidence of the environment. Only those 3D measurements lying above the road are registered as obstacles in the occupancy grid. From this grid the freespace shown in Fig. 4(b) is computed. The used dynamic programming approach turns out to be highly robust w.r.t. disparity noise.

Each free space point of the polygon in Fig. 4(b) indicates not only the interruption the free space but also the base-point of a potential obstacle located at that position. Following the considered image column upward, the disparity is nearly constant but jumps to a smaller value above the object. This allows us to compute the upper boundary (i.e. the height) of the obstacles in a second pass of dynamic programming.

Given base and top point of an obstacle in a certain image column, all object disparities are averaged using a robust estimator to determine the distance of this stixel. This averaging significantly reduces the disparity noise and improves the depth accuracy. In praxis we use a stixel width of 3-7 pixel, which further improves the results. The final result is depicted in Fig. 4(c).

The sketched stixel representation features the following properties:

- *Compactness*: A significant reduction of the data volume is offered. If, for example, the width of the stixels is set to 5 pixels, a scene from a VGA image can be represented by $640/5 = 128$ stixels instead of 300.000 disparities.
- *Completeness*: The geometrical information contained in this representation is sufficient for many recognition tasks in driver assistance.
- *Robustness*: Outliers in the data have minimal or no impact on the resulting representation.

It is straight forward to cluster groups of stixels to objects based on depth discontinuities. Throughout this article it is assumed that vehicles at intersections are sufficiently isolated, i.e., the left and right end of a stixel cluster as well as all stixels in between correspond to a single object.

4. Initial Pose Refinement

The motion-based object detection method initializes objects according to the initial point cloud and with an expected dimension of typical road vehicles. As can be seen in Fig. 5(b), the white box, indicating the projection of the initial object hypothesis onto the image plane, is not accurately aligned with the object boundaries. At the same time, the corresponding stixel cluster provides a very good segmentation as visualized in Fig. 5(a). Therefore, the motion based object detection and initialization method is extended by an initial pose refinement step. The objective is to compute an improved initial vehicle pose that is consistent with the stixel data and that places the rotation point closer to the actual center rear axle.

The idea is as follows: The image columns of the left and right stixel of the corresponding stixel cluster, denoted as u_l and u_r respectively, define the viewing range and constrain the expected vehicle position in lateral direction (see Fig. 5(c)). At the same time, the disparity values d_l and d_r of the boundary stixels introduce constraints on distance.

Depending on the given object pose hypothesis $\langle \hat{x}, \hat{D} \rangle$, the left and right most stixel are directly linked to corresponding object corners. The inner stixels cannot be assigned

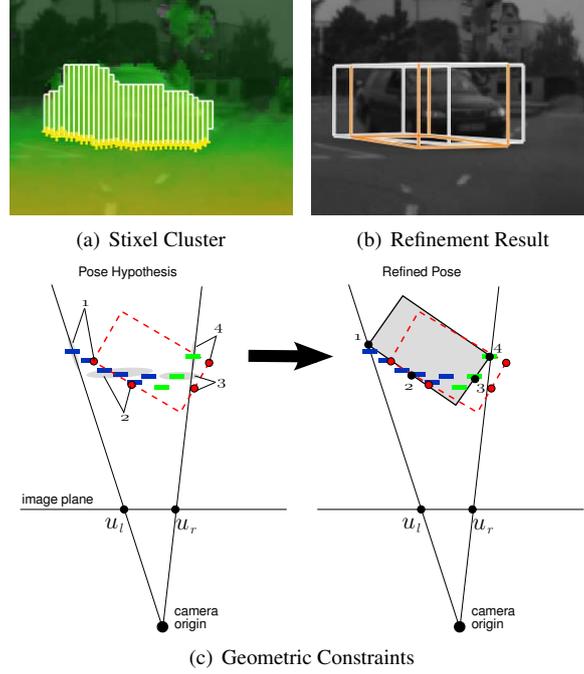


Figure 5. Six constraints on depth and lateral position are derived from a stixel cluster and assigned to four characteristic object points, that depend on visibility properties of the pose prior. The initial pose is iteratively refined by a maximum likelihood estimation.

to a concrete point at the visible object sides that easy, although they also provide valuable information on depth. Due to perspective, one or two vehicle sides are visible in the image at one time. Thus, we divide the inner stixels based on an expectation on the number of image columns to be covered by a visible side. The median disparity d_{s_i} over all stixels assigned to a given object side s_i , $i \in \{1, 2\}$, is taken as additional depth constraint on the center of that side. Since the median is more robust to outliers compared to the mean, inaccuracies in assigning the inner stixels to object sides are acceptable.

Formally, we can summarize the constraints in a vector \mathbf{c} as

$$\mathbf{c} = [u_l, u_r, d_l, d_r, d_{s1}, d_{s2}]^T. \quad (7)$$

Now, $f_{\langle \hat{\mathbf{x}}, \hat{\mathbf{D}} \rangle}$, with

$$f_{\langle \hat{\mathbf{x}}, \hat{\mathbf{D}} \rangle}(\mathbf{y}) = \mathbf{c}, \quad (8)$$

defines the functional model between the constraints and the parameters \mathbf{y} to be refined:

$$\mathbf{y} = [{}^o\mathbf{X}_{rot}, {}^o\mathbf{Z}_{rot}, \underline{\psi}, \underline{\Omega}(\hat{\mathbf{x}})]^T. \quad (9)$$

Here, $\underline{\Omega}(\hat{\mathbf{x}})$ denotes the size of the object side presumably covered by more stixels based on the pose prior $\hat{\mathbf{x}}$, i.e., only one dimension, width or length, is estimated at one time. With an increasing stereo uncertainty at larger distances (e.g. > 40 m for a 0.3 m stereo baseline), the parameter vector is reduced to contain only the rotation point position, since reliable size measurements cannot be obtained and the object motion gives a much better estimate on orientation.

The parameters are estimated using a maximum likelihood estimation [15]. Since $f_{(\hat{\mathbf{x}}, \hat{\mathcal{D}})}$ is nonlinear, it has to be linearized at \mathbf{y}_0 derived from the pose prior. Then, the parameter updates $\Delta \mathbf{y}$, with $\mathbf{y} \approx \mathbf{y}_0 + \Delta \mathbf{y}$, are computed as

$$\Delta \mathbf{y} = \left(\mathbf{A}^\top \Sigma_{cc}^{-1} \mathbf{A} \right)^{-1} \mathbf{A}^\top \Sigma_{cc}^{-1} \left(\mathbf{c} - f_{(\hat{\mathbf{x}}, \hat{\mathcal{D}})}(\mathbf{y}_0) \right), \quad (10)$$

where the matrix \mathbf{A} indicates the Jacobian $\left(\frac{\partial f_{(\hat{\mathbf{x}}, \hat{\mathcal{D}})}}{\partial \mathbf{y}} \right) \Big|_{\mathbf{y}_0}$, and Σ_{cc} the covariance matrix of the constraints.

This estimation procedure is iterated a few times (typically three iterations are sufficient). If the updates converge to approximately zero, the initial object pose is updated by the refined pose. Otherwise it is not changed to prevent a degradation. Fig. 5(b) shows the result of the refinement step for the poor initialization example discussed before. As can be seen, the orange box approximates the object much better and the rotation point has been moved from the centroid of the point cloud towards the rear.

Note that the refinement approach proposed above assumes an object to be completely visible in the image. Special handling of partly occlusions is outside the scope of this article.

5. Stixel Measurements

The pose refinement method proposed above, is used not only at initialization, but also to yield direct measurements of the rotation point during the tracking phase. The measurement vector \mathbf{z}_{rot} (see Sec. 2) is set to $\mathbf{z}_{rot} = [{}^o\mathbf{X}_{rot}, {}^o\mathbf{Z}_{rot}]^\top$. The Kalman filter state prediction is used as pose prior for computing the rotation point measurements.

The measurement noise can be derived from the estimation procedure, since $\Sigma_{\Delta \mathbf{y} \Delta \mathbf{y}} = \left(\mathbf{A}^\top \Sigma_{cc}^{-1} \mathbf{A} \right)^{-1}$ gives the covariance of the parameter updates. The final covariance matrix is given by error propagation.

The simultaneously estimated size $\underline{\Omega}(\hat{\mathbf{x}})$ is used to update the corresponding cuboid dimension, width or length, by a slow low pass outside the Kalman filter. In addition, height measurements are easily obtained from the stixel cluster.

6. Experimental Results

The proposed system has been tested on various real-world intersections scenes. The tracking results of an example sequence are superimposed in Fig. 6. The bounding box indicates the object pose and the carpet on the ground the predicted driving path based on the current motion state assuming constant yaw rate and constant acceleration. Optical

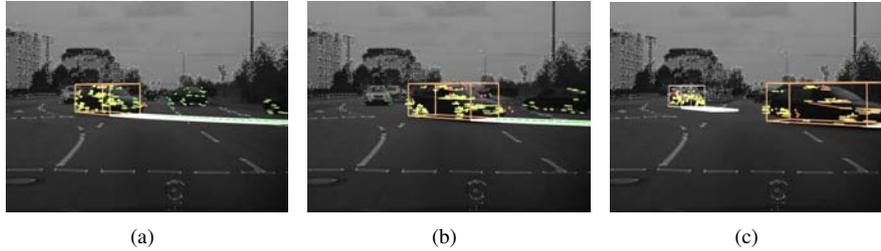


Figure 6. Example tracking results of an intersection scene with the estimated pose and predicted driving path superimposed. A second oncoming object is detected as soon as it starts moving (right image).

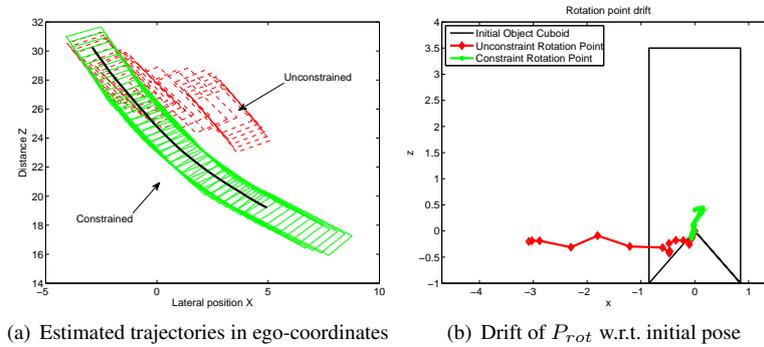


Figure 7. Additional rotation point measurements prevent that the Kalman filter minimizes the measurement error of the observed noisy object points by shifting the rotation point to a position outside the object.

flow vectors are also visualized. The object is successfully tracked through the maneuver until it leaves the visual field of the camera.

The estimated trajectory of this sequence (black solid line) and the corresponding object poses (green solid boxes) are shown in Fig. 7(a) from a bird's eye view. Without additional rotation point measurements the filtering fails in this sequence, indicated by the second trajectory in this figure (red dashed boxes).

The Kalman filter minimizes the residual between predicted and measured object points. If the filter is allowed to change the rotation point without any geometric constraints, it is possible that the rotation point drifts away from the object point cloud as occurring in this example (see Fig. 7(b)). A reliable prediction of the turn maneuver is not possible anymore, thus, the object track is rejected. This demonstrates the importance of the geometric constraints on the rotation point.

Further example results are depicted in Fig. 8. The Kalman filter state estimates of the yaw rate and velocity of the first row sequence are shown in Fig. 9. The velocity profile shows a typical acceleration behavior at turn maneuvers. The driver first slightly reduces the velocity and then accelerates after the maximum turn rate is reached.

The system stably runs at 80 ms cycle time on 640×480 images, including 40 ms for the freespace and stixel computation, 25 ms for feature tracking and ego-motion computation, as well as 2 – 3 ms for tracking and pose refinement of a single object on a current Quad Core processor, without exploitation of massive parallel computing, e.g., on the graphics card.

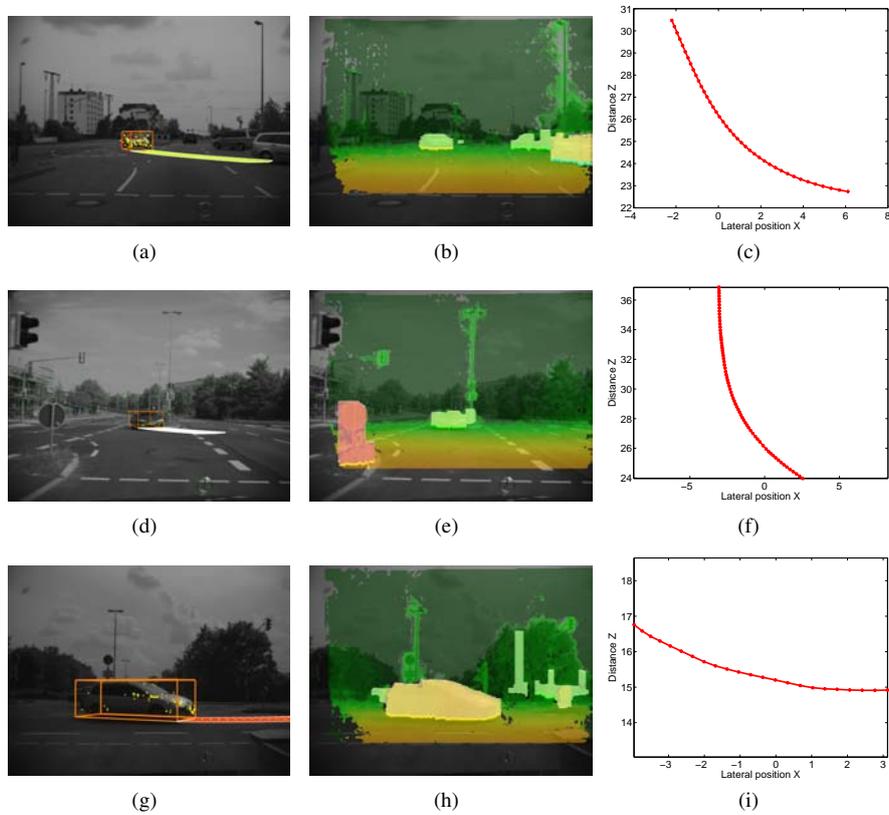


Figure 8. Example results with bounding box and predicted driving path superimposed (first column), stixel clusters and SGM stereo (second column), and resulting trajectories from bird's eye view (third column).

7. Conclusion

We have presented a hybrid vehicle tracking approach that combines a feature-based point cloud model with geometric constraints for application of tracking turning vehicles at intersections.

The Stixel World provides a powerful and efficient representation for the precise localization and segmentation of object boundaries. This information has been used for an initial pose refinement step and to derive additional measurements that constrain the position of the rotation point to the object's lateral center during filtering. The experimental results have shown a significant improvement of the tracking of oncoming vehicles at intersections.

The realization based on dense stereo stixels has demonstrated the practical usage of the generic measurement model. Alternative image-based methods for extracting the object silhouette in the image or other range sensors, such as lidar scanners, could be integrated accordingly.

Further investigations will include the direct integration of the single stixels into the object shape model and to address the problems arising at extreme dense traffic scenes that have been excluded in this contribution.

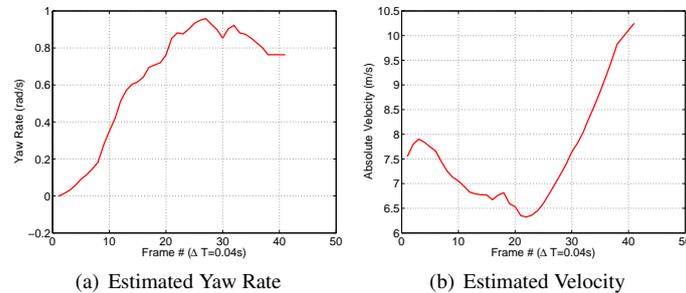


Figure 9. Motion state estimation results. The precise yaw rate and velocity estimate of the oncoming traffic provides useful information for future driver assistance systems.

References

- [1] H. Veeraraghavan and N. Papanikolopoulos, "Combining multiple tracking modalities for vehicle tracking at traffic intersections," in *IEEE Conf. on Robotics and Automation*, 2004.
- [2] S. Atev, H. Arumugam, O. Masoud, R. Janardan, and N. Papanikolopoulos, "A vision-based approach to collision prediction at traffic intersections," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 4, pp. 416–423, Dec. 2005.
- [3] A. Ottlik and H. H. Nagel, "Initialization of model-based vehicle tracking in video sequences of inner-city intersections," *Int. J. Comput. Vision*, vol. 80, no. 2, pp. 211–225, 2008.
- [4] R. Danescu, S. Nedevschi, M. Meinecke, and T. Graf, "Stereovision based vehicle tracking in urban traffic environments," *Intelligent Transportation Systems, IEEE Conference on*, pp. 400–404, 2007.
- [5] B. Barrois, S. Hristova, C. Woehler, F. Kummert, and C. Hermes, "3D pose estimation of vehicles using a stereo camera," in *Intelligent Vehicles Symposium, IEEE*, 2009.
- [6] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik, "A real-time computer vision system for measuring traffic parameters," in *Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997, pp. 495–501.
- [7] T. Dang, C. Hoffmann, and C. Stiller, "Fusing optical flow and stereo disparity for object tracking," *IEEE 5th International Conference on Intelligent Transportation Systems*, pp. 112–117, 2002.
- [8] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool, "Dynamic 3D scene analysis from a moving vehicle," in *Computer Vision and Pattern Recognition, CVPR. IEEE Conference on*, 2007, pp. 1–8.
- [9] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *Intelligent Vehicles Symposium, IEEE*, 2008.
- [10] H. Badino, U. Franke, and D. Pfeiffer, "The stixel world - a compact medium level representation of the 3D world," in *DAGM Symposium, Jena, Germany*, September 2009.
- [11] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications To Tracking and Navigation*. John Wiley & Sons, Inc, 2001.
- [12] A. Wedel, U. Franke, H. Badino, and D. Cremers, "B-spline modeling of road surfaces for freespace estimation," in *IEEE Intelligent Vehicles Symposium*, 2008, pp. 828–833.
- [13] H. Badino, "A robust approach for ego-motion estimation using a mobile stereo platform," in *1st Intern. Workshop on Complex Motion (IWCM04)*, Guenzburg, Germany, October 12 - 14 2004.
- [14] C. Tomasi and T. Kanade, "Detection and tracking of point features," Carnegie Mellon University, Tech. Rep. CMU-CS-91-132, April 1991.
- [15] C. McGlone, Ed., *Manual of Photogrammetry*, 5th ed. Amer. Soc. Photogrammetry, 2004.
- [16] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-vision: Fusion of stereo and motion for robust environment perception," in *27th DAGM Symposium*, 2005, pp. 216–223.
- [17] H. Hirschmüller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Computer Vision and Pattern Recognition, CVPR*, vol. 2, June 2005, pp. 807–814 vol. 2.
- [18] F. E. S. Gehrig and T. Meyer, "A real-time low-power stereo engine using semi-global matching," in *International Conference on Computer Vision Systems, ICVS*, 2009.
- [19] H. Badino, T. Vaudrey, U. Franke, and R. Mester, "Stereo-based free space computation in complex traffic scenarios," in *IEEE Southwest Symposium on Image Analysis and Interpretation*, 2008.