Stereovision Based Vehicle Tracking in Urban Traffic Environments

R. Danescu, S. Nedevschi, M.M. Meinecke, T. Graf

Abstract— **This paper presents an algorithm for tracking the cuboids generated from grouping the 3D points obtained through stereovision. The solution described in the paper takes into consideration the particularities of the scenario and of the sensor, and brings considerable improvement in all the phases of tracking: initialization, prediction, measurement and update. The corner of the cuboid becomes the central working concept, thus improving the handling of partially occluded objects, of objects partially out of the field of view, and of objects whose measurement is fragmented by the sensor inaccuracies. After association at corner level, multiple measurements or validated parts of a measurement form a virtual object, the meta measurement, which is used for track update. The size of a vehicle is tracked using a histogram voting method. The resulted algorithm shows robustness and accuracy in the crowded urban scenario.**

I. Introduction

Urban area is a crowded place, not only because of cars, pedestrians and other mobile units, but also because of the complex scenery, which is never too far away from the traffic lanes. The success rate of a tracking algorithm depends greatly on its capacity to ensure that the same object is tracked in all the frames. This means no confusion between objects, no losing of the object, and no merging of independent objects. Therefore, the most difficult part of the urban traffic object-tracking algorithm is the measurement-track association.

Depending on the sensor and on the problem to be solved, the demands and the complexity of a tracking algorithm may vary. All tracking algorithms try to evaluate the state of a system over a period of time, using the available sensorial information and some knowledge about the intrinsic properties of the system. Many trackers rely on the mathematical framework of the Kalman filter [2], which provides a consistent and computation effective way of handling the stages of prediction, association and update. The use of Kalman filter for tracking is described in many works, the most representative being [1].

The Kalman filter requires that the probability models of all the tracking stages are known, and are of Gaussian type, described by mean value and covariance matrix. In order to overcome these restrictions, researchers have invented several variants of the Kalman filter, such as the Extended Kalman Filter (EKF) of the Unscented Kalman filter (UKF) [5], and even tracking methods that work with any type of probability density, such as the CONDENSATION algorithm [3], which describes the probability as the density of a set of particles.

Many of the works in the field of tracking focus on the mathematical aspects related to the manipulation of probability values, and say little about the way in which the tracking algorithm is adapted to a given sensor in order to make a functional system. This is especially true for the vision sensors, which deliver complex, but less reliable data. The error model of a ranging sensor is simple, and is related to the sensor's inaccuracies only, while the vision sensor may provide occasional false or incomplete results, which may not be suitable for association using probability laws only (or they would be if the error probabilities were perfectly modeled).

The purpose of this paper is not to present a new mathematical model for tracking, but to show how we can solve the problems associated to the output of a stereovision sensor (or any vision sensor) in a classical Kalman filter tracking framework. The work relies on the results of the stereovision algorithm for lane and object detection, developed by the Technical University of Cluj-Napoca and Volkswagen AG [4].

II. Measurement

The tracking algorithm is the final stage of a stereovision based object recognition system. The 3D points obtained through stereo processing of a synchronized pair of images are placed in two categories: road points and obstacle points. The road points are used for lane detection, and the obstacle points are grouped into cuboids, based on vicinity criteria. The result of the grouping algorithm, which is also the measurement data for tracking, is a list of non-oriented 3D cuboids.

Manuscript received April 1, 2007. This work was supported by Volkswagen AG.

R. Danescu and S. Nedevschi are with the Technical University of Cluj Napoca, Romania, e-mail: Radu.Danescu@cs.utcluj.ro, Sergiu.Nedevschi@cs.utcluj.ro, Department address: Computer Science Department, Str. C. Daicoviciu nr. 15, 400 020 Cluj Napoca, Romania, Phone: +40 264 401 457

M.M. Meinecke and T. Graf are with Volkswagen AG, Driving Assistance Electronics department, e-mail: marc-michael.meinecke@volkswagen.de, thorsten.graf@volkswagen.de

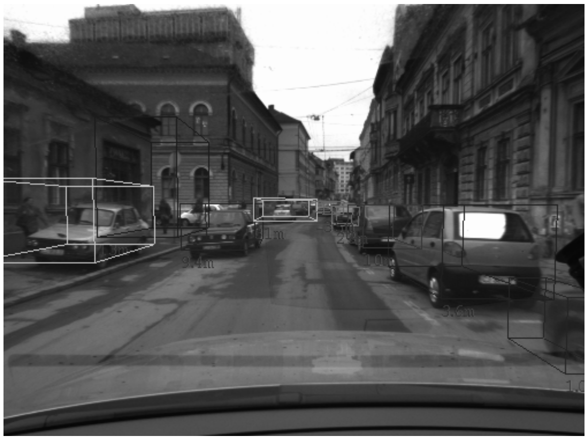

Fig. 1. Grouping the 3D points into cuboids – perspective view

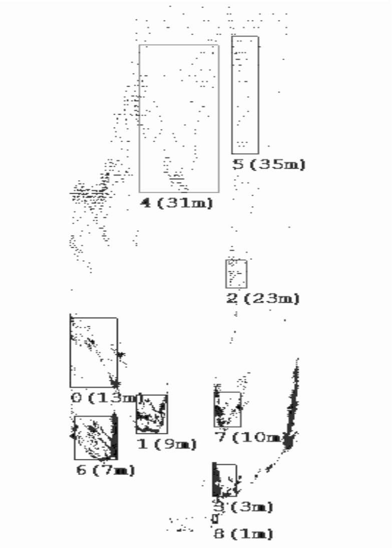

Fig. 2. Grouping the 3D points into cuboids – bird-eye view

III. Object Model

A model for the vehicle object must be general enough to account for all the object of interest, and yet simple enough to be handled in a robust fashion. The 3D cuboid is our proposed object model. Its components are:

- The size components Sx, Sy and Sz, along each of the coordinate axes

- The position components X and Z.

- The velocity components Vx and Vz, which also provide the object's orientation.

The Y coordinate will not be tracked, as the only objects that we want to track are the ones on the road.

Our coordinate system has the origin in the front of the ego vehicle, at the ground level. The X axis points to our left, the Y axis points down towards the road, and the Z axis points forward along our direction of travel. The size components Sx, Sy and Sz can also be called width, height and length of a tracked vehicle.

IV. Cuboids and Corners

The point grouping algorithm delivers a set of non-oriented cuboids, aligned with the coordinate axes. One of the problems we face is that we don't know if these objects represent whole objects in the real world, or they are fragments. Fragmented vision has many causes, some avoidable through algorithm optimization, and some unavoidable. One of the unavoidable fragmentation causes is the field of view clipping. If an object intersects with the field of view, we have no possibility of knowing whether it is a fragment or a full object, in the absence of a classification algorithm.

Figure 3 shows a set of cuboids, in bird-eye view. There is no indication whether these objects are whole or fragments. Figure 4 shows the position of these objects against the field of view, and the position of the 2D projection of these objects in the image space. We can easily notice that the objects touching the limits of the 3D field of view are also touching the limits of the image.

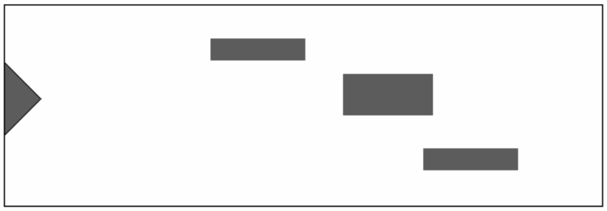

Fig. 3. Bird-eye view of a set of cuboids. Are they complete or partial views of larger cuboids?

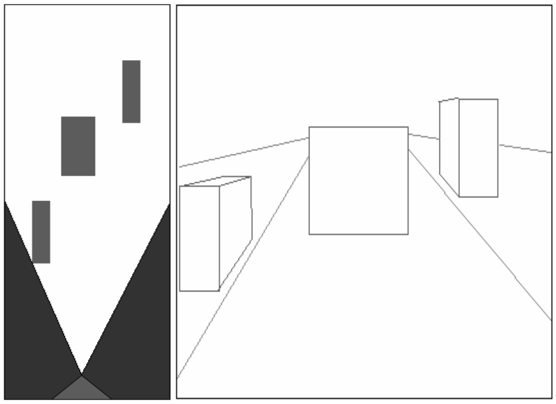

Fig. 4. Cuboids versus the field of view (left) and in the image space (right).

By analyzing the position of the object in the image space, we can decide which object has great chances of being fragmented. Even more, we can say which of the corners of the object are trustworthy.

We divide the image into several zones, as in figure 5. Depending on the position of the object with respect to these zones, we have five possibilities:

- Center object: the projection of the back side of the object falls within the central left and right limits

- Moderate left object: the projection of the back side of the object crosses the first left limit, but does not touch the second (extreme) left limit

- Extreme left: the projection of the back side of the object crosses the extreme left limit

- Moderate right object: same as the moderate left, but with respect to the right limits

- Extreme right object: same as the extreme left, but with respect to the right limits

In figure 5 we have exhibited three such situations: center, extreme left and moderate right.

Depending on what situation described above an object is in, we can establish its relevant corner list. The object viewed from above is a rectangle with four corners, but not all these corners are visible – the invisible corners are considered irrelevant, and will play no part in the stages of the tracking algorithm.

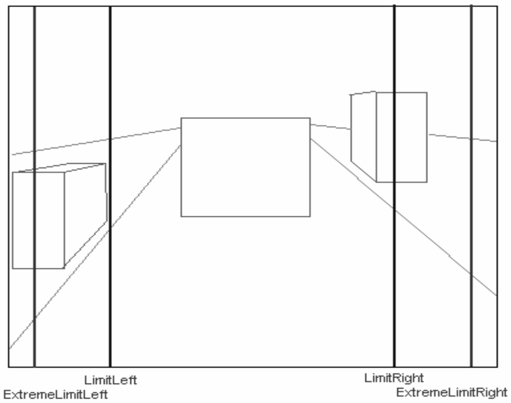

Fig. 5. Image zones used for establishing the list of visible corners

The relevant corners of a central object are the left and the right backside corners, as we cannot see the front of the vehicle. A moderate lateral object has both corners from the backside visible, and also one corner of the front side (the moderate left object will have the front right corner as relevant, and the moderate right object will have the right front corner as relevant). The extreme objects will have only one relevant corner, the front left or right corner.

The corner based reasoning will affect all the stages of the tracking algorithm, which will be presented in the following sections.

V. Track Initialization

A measurement cuboid may initialize a track if several conditions are met:

1. The cuboid is not associated to an existing track;

2. The cuboid is on the road (we compare its Y position with the profile of the road);

3. The cuboid's back side position in the image does not touch the image limits – the object is either a central object or a moderate side object (as described in the previous section)

4. The height and width of the cuboid must be consistent to the standard size of the vehicles we expect to find on the road. The classification based on size is also useful for initialization of the object's length, as in most cases the camera cannot observe this parameter directly.

If a cuboid obeys these conditions, it creates a track hypothesis, which becomes a confirmed track after three consecutive successful associations with the measurement data, in the next frames.

VI. Measurement-Track Association

The association (matching) process has two phases: a 3D matching of the predicted track cuboid against the measurements, which is performed as a simple intersection of rectangles in the bird-eye space, and a corner by corner matching in the image space, when each active corner is matched against the corners of the measurement cuboids that passed the 3D intersection test.

We refer to the 3D matching phase as the "coarse" matching, and to the corner-level matching as the "fine" matching.

For the coarse matching, the predicted cuboids are "enhanced" with the measurement error on the Z axis, error which can be evaluated from the parameters of the optical system.

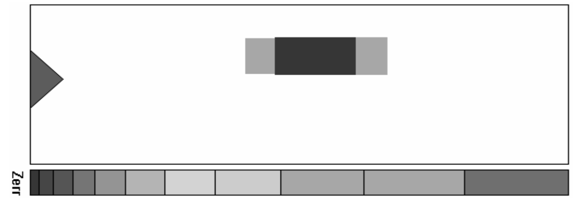

Fig. 6. The measurement error of the stereo reconstruction is added to the prediction length

The enhanced object is then verified against the measurement objects, by computing the intersections in the bird-eye view projection space.

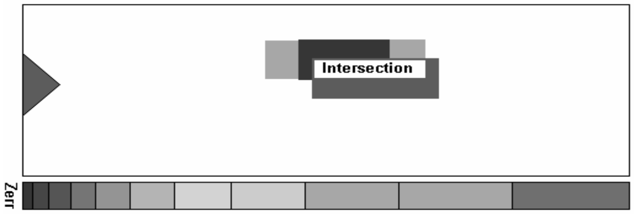

Fig. 7. Intersection between the enhanced prediction and the measurement

The ratio between the area of the intersection and the area of the enhanced prediction is a measure of the quality of the association. If a measurement intersects multiple predicted objects, it will be associated to the one it had the best intersection measure. A predicted object may associate to multiple measurement objects, but not the other way around.

After the 3D association is completed, each track prediction is compared to each of the associated measurements corner by corner, using their 2D projection, for the relevant corners only. The relevant coordinate is the image lateral coordinate (the x coordinate), as it is more stable than y, which is influenced by the random pitching of the ego vehicle. Once we have established that the objects that we consider are on the road (airborne objects are rejected in all stages of tracking), and once we have established the fact that the objects to be associated are in the same distance range, there is no need for an image y-coordinate test.

The comparison distance is the difference in the image x coordinate between the measurement and the prediction corners, normalized by the width of the predicted object in the image space. Only corners of the same type are compared (that is, the forward left corner of the prediction is compared only to a forward left corner of the measurement). If a relevant corner of the prediction associates to at least one relevant corner of a measurement, this corner becomes an "active" corner. The active corners form a virtual object which we'll call "meta-measurement". The meta measurement is the sum of all measurement objects associated to a predicted object. This process is described by the figures 8, 9 and 10. When two or more measurements associate at corner-level to the same side of the prediction, we expand the meta-measurement in such a way as to create an envelope of the associated measurements, and this helps overcome fragmentation.

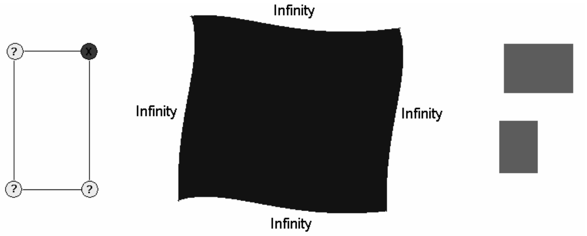

Fig. 8. The relevant corners (left), meta-measurement (middle) and unassociated measurement data (right)

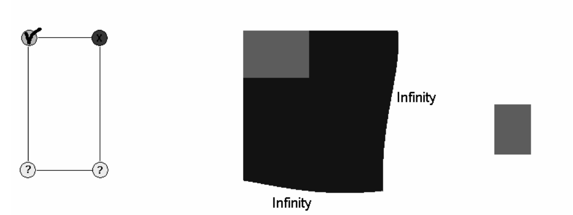

Fig. 9. Association of the first measurement object. The forward left corner becomes an active corner, and the meta-measurement is bounded on two sides

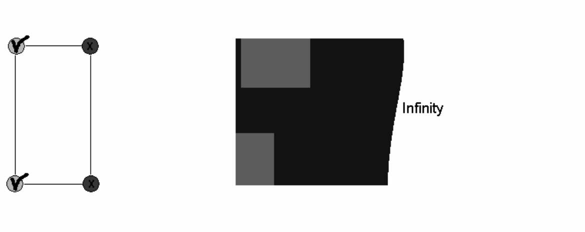

Fig. 10. All possible associations have been performed. The meta-measurement is bounded in three sides.

VII. Track State Update

The mathematical framework for updating the position and the speed of a vehicle is the Kalman filter, constructed upon a linear motion model and a linear measurement model.

Depending on the situation of the meta measurement, we can have a direct observation of the position, or an inferred one. For instance, if the meta measurement is bounded to the left and to the right, we have a measurement of the X coordinate of the center of the object. If the meta measurement is bounded to one side only, we use the size information to compute the center's position based on the position of the observed lateral side. The same reasoning applies for the Z coordinate of the object center. The speed of the object is not observed directly, as it is a hidden state variable. It will derive from the Kalman filter estimation.

The size of the tracked object, formed by three components, Sx, Sy, Sz, is not suited for tracking with the Kalman filter, as it is not a continuous parameter. However, due to the fact that at one point the size of an object may be incorrectly evaluated due to measurement errors or lack of visibility, a mechanism of refining the size estimation had to be devised. This mechanism is a voting system based on histograms, one histogram per size component. The positions in the histogram correspond to each possible size value, in a discrete system (we have chosen a 10 cm increment). When a size component observation is made, the corresponding histogram value is incremented by a value which takes into account the reliability of the observation (if the object is too far, the increment is smaller, if the object is near the increment is higher). The size corresponding to the histogram cell with the highest value is taken as the current perceived size of the object.

One size component is updated only if the meta measurement is bound on both sides of the corresponding axis.

The results are output as cuboids, for the purpose of display in 2D and/or 3D fashion, and for communicating through the CAN bus, to the driving assistance application. Due to the fact that the urban environment is unpredictable, both tracked and non-tracked objects need to be considered. The difference between tracked and non-tracked objects is persistence and speed. Tracked objects have persistence, which means that they do not disappear if they are not detected in one or two frames. Non-tracked objects, detected in the current frame, are included in the results, but they will be removed if they are not detected in the next frames. Tracked objects also have speed information, derived from temporal analysis through the Kalman filter.

VIII. Results, Conclusion and Future Work

The system has been tested extensively on real urban traffic scenes, in many situations where a "classical" tracking algorithm, designed for highways, would have failed. The problems included persistent ghost objects, false associations due to 3D only vicinity criteria which failed on the city streets, and false object size due to field of view clipping. The new algorithm, characterized by a more elaborate and slower initialization, a complex, 3D based and 2D corner-based measurement-track initialization that is aware of the limitations of the field of view, and mixed tracked/non tracked objects output, solved the above problems in almost all scenarios.

Figure 11 and 12 show some results in real world scenarios. The tracked vehicles are drawn with a lighter line, while the detected and not tracked objects are drawn with a dark line.

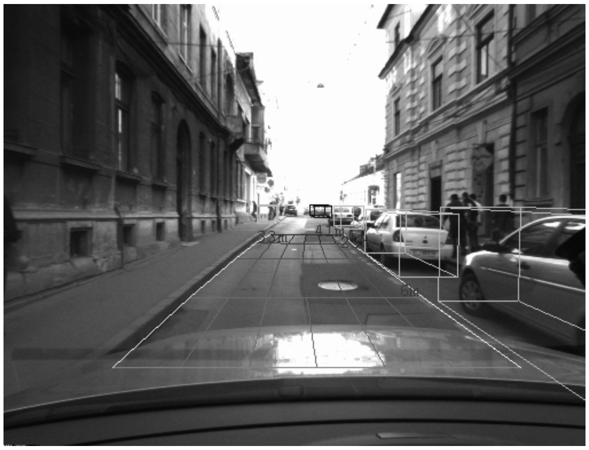

Fig. 11. Tracking results – the side objects are correctly tracked as we pass them.

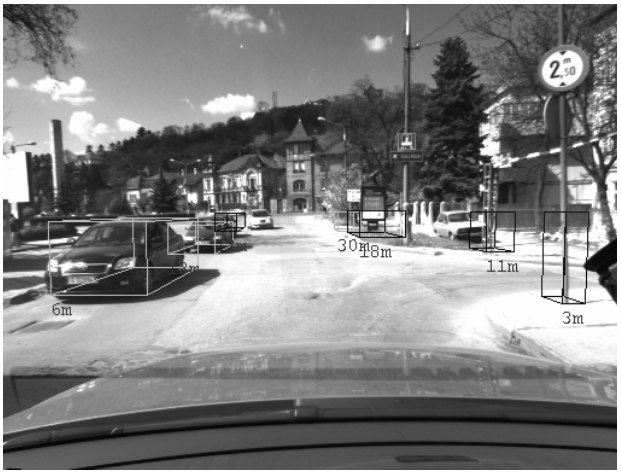

Fig. 12. Tracking results – are tracked, and the scenery is seen as non-tracked objects

Future work will address the problem of tracking the object orientation directly, instead of deriving it from the velocity components, and will combine the size classification with a shape classifier, for a more robust initialization.

References

[1] 1. Y. Bar-Shalom, T.E. Fortmann, "Tracking and Data Association", Academic Press Inc., 1988

[2] 2. R.E. Kalman, "A New Approach to Linear Filtering and Prediction Problems", Transactions of the ASME--Journal of Basic Engineering, vol. 82, pp. 35-45, 1960

[3] 3. M. Isard, A. Blake, "CONDENSATION -- conditional density propagation for visual tracking", Int. J. Computer Vision, 29, 1, 5--28, (1998)

[4] 4. S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, T. Graf, R. Schmidt, "Driving Environment Perception Using Stereovision", Procedeeings of IEEE Intelligent Vehicles Symposium, (IV2005), June 2005, Las Vegas, USA, pp.331-336

[5] 5. E.A. Wan, R. Van der Merwe, "The unscented Kalman filter for nonlinear estimation", Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC.