

# A comparison of trajectories and vehicle dynamics acquired by high precision GPS and contemporary methods of digital image processing

K. Kozempel, H. Saul, M. Haberjahn & C. Kaschwich  
*Institute of Transportation Systems, German Aerospace Center, Germany*

## Abstract

Vehicle trajectories from recorded video sequences—acquired by several contemporary methods of digital image processing—are compared with high-precision GPS data serving as a reference. The raw data has been created by driving some scenarios with a car equipped with several sensors, i.e. DGPS, acceleration sensor, etc. At the same time, the car was recorded by a video camera system in order to derive trajectory data by computer vision methods. Thus, the car is tracked by an Extended Kalman Filter (EKF) preceded by a background estimator. To improve the accuracy of the tracking data it is combined with a model-based approach for object detection. This approach fits a 3-dimensional wire frame model of the car into the image. The paper presents the driving scenarios of the car, the implemented image processing methods and a quantitative evaluation of the extracted trajectories obtained by two different image processing methods. Accuracy and precision of the methods are determined by comparing their results with the DGPS reference data of the car.

*Keywords: trajectories, DGPS, accuracy, precision, evaluation, image processing, model-based.*

## 1 Introduction

The use of video cameras for traffic detection has become an easy and affordable alternative to induction loops. Cameras are comparatively cheap and easily to setup and a lot of research has been done in this field in the last years. However, the acquired data has to be geo-referenced, which means the detected objects must



be transformed from image space into world coordinates. A simple approach transforming the vehicles 2-dimensional image position into 3-dimensional world coordinates—provided the exterior orientation of the camera is known—is to project the object's centroid onto the road surface. This is not quite exact, as the centroid does not represent the center of the object and the further away it is, the less accurate the projection is. A more sophisticated approach is to fit the projection of a 3-dimensional car shaped model into the image. The adjustment of its orientation in world coordinates should provide a more accurate position. Although much research concerning model-based vehicle detection and tracking has been published, most of it only investigates the detection and classification accuracy or the adjustment convergence of the model. According to our knowledge, none of the publications evaluates the accuracy or rather the improvement compared to the simple centroid projection regarding the real world positioning.

Our approach is to compare the exact vehicle positions, provided by a high precision differential GPS (DGPS) and an inertial measurement unit (IMU), with both geo-referencing approaches, the centroid projection and the model adjustment.

## 2 Related work

Approaches for visual object detection or tracking—either in gray or colored video data—, that are based on a synthetic 3D-model are presented shortly hereafter. They all have in common the basic principle of projecting 3D-points or -edges into the image in order to find correspondences and, thus, finding the 3D-model equivalent in the image.

Koller *et al.* [1] tracks vehicles in traffic scenes by means of different vehicle models generated through twelve length parameters for 3D polyhedral models. Line segments in the image are matched with models' edges. The tracking process is supported by clustering of coherently moving image features. Example tracks from real video data are shown, including the results of positions of resulting trajectories and translational and angular velocity.

In Haag and Nagel [2], the same generic polyhedral vehicle models as Koller uses are applied. The tracking is again based on matching edge elements and is propped by the optical flow field. Rommel [3] examines different cost functions for model fitting by contour adaption and their convergence. Reinert [4] does not focus on vehicles, but studies different methods for adjustment of point and line correspondences for pose estimation in the context of Augmented Reality applications. A detailed evaluation is listed for solving 2D-3D point or line correspondences of the model fits by non-linear optimization. Less a-priori object model knowledge is used by Schwarzenberg [5] and no volume data is deducible from these models. Schwarzenberg introduces particle filtering for tracking. Current research concerning different cost functions and optimizations can be found in [6], [7] or [8], while our work focusses on determining positioning errors in world coordinates.





Figure 1: DLR Test vehicles *ViewCar* (left) and *UTRaCar* (right).

### 3 System description and methods

Two test vehicles were used to acquire video data on the one hand and GPS data on the other. The field test was conducted on June 25th, 2013.

#### 3.1 Video data acquisition

The Urban Traffic Research Car (*UTRaCar*, see figure 1), a Volkswagen Transporter especially adapted for traffic research purpose, has a 13 meters high extendable pole on which a mono camera system was mounted. The intrinsic camera orientation (focal length, lens distortion etc.) has been calibrated in advance in the laboratory while the extrinsic orientation had been calibrated on site using GPS measured ground control points.

#### 3.2 Reference GPS data

For the reference data set (ground truth) the *ViewCar* uses the inertial measurement system iDIS-FMS from the manufacturer iMAR which was created for motion analysis of cars, ships, airplanes. It records attitude angles, velocities and accelerations as well as GPS positions. Using post-processing and differential GPS (DGPS), iMAR claims to provide an accuracy of less than 1 cm in positioning. Unfortunately, the current accuracy values were not recorded during the tests. But both DGPS and post processing have been used, why a precision of 5 cm is assumed.

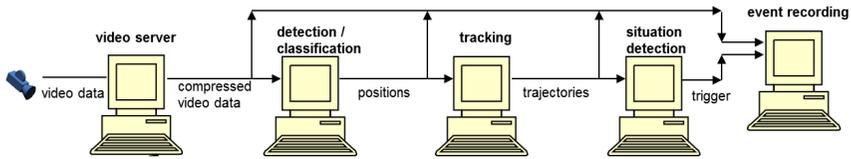


Figure 2: The traffic detection system MUSE.

### 3.3 Image processing

For the acquisition and processing of traffic data the Institute of Transportation Systems at German Aerospace Center (DLR) developed an own multi-sensor software (MUSE), which is sketched in figure 2. For the given purpose only a part of the system is used, which records and processes video data in order to detect, classify and track traffic objects.

The camera records video data, which is scaled and distributed by the video server to further clients (*detection and classification*). A Mixture-of-Gaussians background estimator detects moving regions and merges them to objects. Afterwards, the contour is Fourier transformed and the coefficients are put into a feature vector classified by a multi-class Support Vector Machine (SVM) to distinguish different vehicle types. In the last step the objects are tracked (*tracking*) by a Kalman Filter.

#### 3.3.1 Geo-referencing

For the first geo-referencing method simply the centroid is determined and projected onto the road surface. This intersection, assumed to be the vehicle's center point, causes obviously accuracy problems, as it lies behind the vehicle in most cases. The further away or the higher a vehicle is the less accurate its position can be determined. Of course, a lower projection point can be taken instead of the centroid, e.g. somewhere between the centroid and the lowest object's image point, but this causes projection errors for very close objects. Anyway, no fixed point in the 2D shape represents the real vehicle's center for all positions.

The second approach creates 3D wire models of different car classes according to Koller *et al.* [1] and projects them into the image. The initial values for position and direction are obtained by an ellipse fitting on the extracted foreground pixels. The ellipse is projected onto the road surface to get the values for east and north position as well as yaw angle in world coordinates. The vehicle is assumed to be on the plane road surface (altitude) and in an upright position (pitch=0, roll=0) which means that only three values (east, north and yaw angle) have to be adjusted. The test vehicle *ViewCar* (figure 1 left) is an Audi A6 C5 Avant, and the angular Koller model's parameters were especially adapted to fit the vehicle's curved hull best.

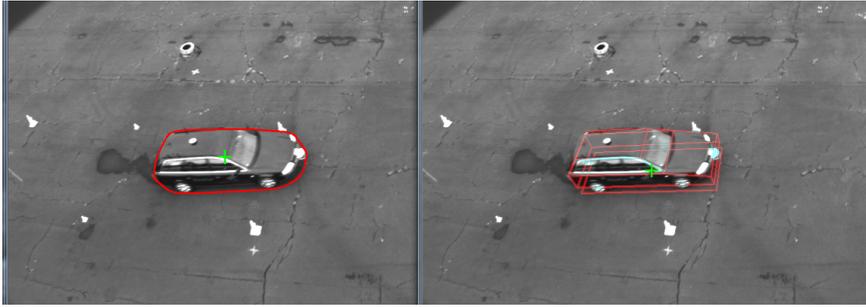


Figure 3: Centroid projection (left) and model fitting (right).

### 3.3.2 Adjustment

There are many different approaches for cost functions  $\Gamma(X)$  concerning model projection, some consider only point distances while some others compare line segments by angle and position. As mentioned by Rommel [3] the hull adaption is a rather simple cost function but very reliable in convergence. The cost function

$$\Gamma(X) = 1 - \frac{\text{area}(K(X) \cap C)}{\text{area}(K(X) \cup C)} \quad (1)$$

determines the difference area between the vehicle's detected ( $C$ ) and its projected hull ( $K(X)$ ) in dependence of the orientation parameter vector  $X$ . We slightly modified the function by taking only the projected lines instead of creating the convex hull, which caused faster processing and delivered even better results.

As adjustment a downhill simplex algorithm was implemented according to Nelder and Mead [9], which minimizes a nonlinear multidimensional problem without knowing any derivatives of the cost function.

### 3.3.3 Tracking

Subsequently, the positions are tracked with an Extended Kalman Filter, which in contrast to most implementations uses not only position ( $x, y$ ), angle ( $\alpha$ ), velocity ( $v$ ) and angular speed ( $v_\alpha$ ), but additionally the acceleration ( $a$ ) and angular acceleration ( $a_\alpha$ ). The state vector is defined as follows:

$$X = (x, y, \alpha, v, v_\alpha, a, a_\alpha). \quad (2)$$

The Kalman Filtering is done in order to smooth the position disparities and to get an optimal estimation between measured and predicted position depending on the motion model and covariance parameters [10].

## 4 Statistical analysis and validation of results

For the evaluation of accuracy and precision three test drives have been conducted in front of the camera. All of them describe a left-right chicane (see figure 4). The

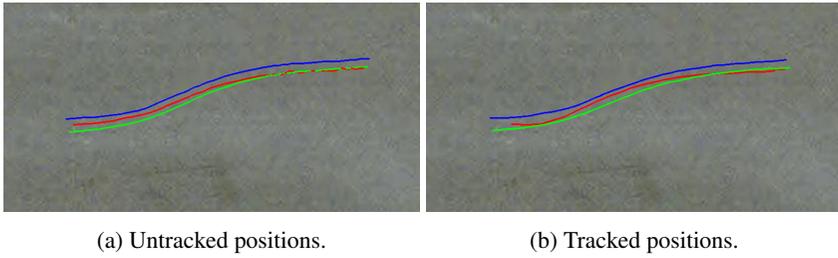


Figure 4: Trajectories from a scene: GPS (green), centroid-projection (blue), model-based (red).



Figure 5: Position errors, scene 1 (30 km/h).

*ViewCar* passes the camera view from left to right, twice with 30 km/h and once with 50 km/h.

The results of the image processing methods, which are presented in the preceding chapters, are validated by comparing them to high-precision DGPS positions from the observed car. Additionally, the extracted trajectories from the camera are validated untracked and filtered in order to see the influence of the EKF on the precision. The trajectories compared are shown in figure 4. Error graphs of the positions for each trajectory are shown in figures 5–6.

The statistical evaluation considers accuracy, which is the closeness of the positions determined from camera to the actual value received from DGPS, and the precision, which is the degree of repeatability and expresses how much the measurements scatter and vary under unchanged conditions. Only the horizontal (position) errors have been considered, since the altitude is assumed to be on the planar road surface. The evaluation is based on the errors of the measurements or observations from their corresponding GPS positions. Let  $(x_i, y_i)$  be the measured positions by image processing and  $(x_{GPS_i}, y_{GPS_i})$  their time-corresponding

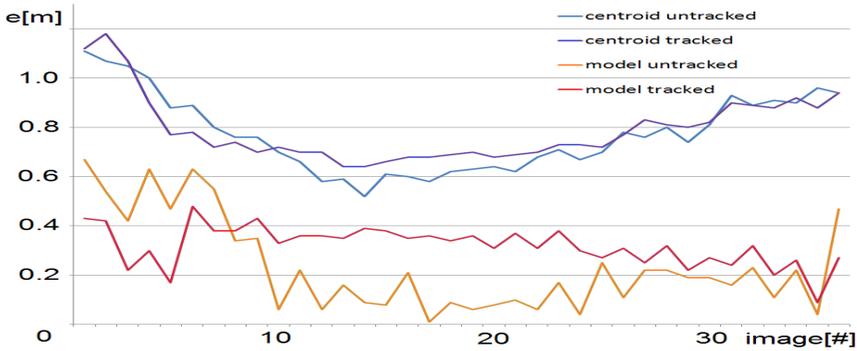


Figure 6: Position errors, scene 3 (50 km/h).

positions measured by GPS—providing as ground truth—for a number  $n$  of samples with  $1 \leq i \leq n, i \in \mathbb{N}$ . Then the horizontal error is the Euclidian distance between two associated measures:

$$e_{xy} = \sqrt{(\Delta x^2 + \Delta y^2)} \tag{3}$$

composed of the  $\Delta x$ , which is error in East-West-direction, and  $\Delta y$ , the error South-North-direction:

$$\Delta x = x - x_{GPS}, \tag{4}$$

$$\Delta y = y - y_{GPS}.$$

Since the positions are given in the UTM coordinate system the errors are consequently measured in meters.

The root-mean-square error (RMSE), also known as root-mean-square deviation (RMSD), was calculated in order to estimate GPS positional error in terms of accuracy. The RMSE is the square root of the mean of the pairwise squared deviations and measures how far on average the error is from 0. The RMSE for the horizontal position error  $RMSE_h$  is calculated as follows:

$$RMSE_h = \sqrt{\frac{1}{n} \sum_{i=1}^n \Delta x_i^2 + \Delta y_i^2}. \tag{5}$$

In addition to it, the RMSE can be determined for each component of the error, i.e. the  $RMSE_x$  for the x-axis and the  $RMSE_y$  for the y-axis:

$$RMSE_x = \sqrt{\frac{1}{n} \sum_{i=1}^n \Delta x_i^2} \tag{6}$$

$$RMSE_y = \sqrt{\frac{1}{n} \sum_{i=1}^n \Delta y_i^2}. \tag{7}$$



Table 1: Accuracy and precision of the methods for tracked trajectories in meters.

	Centroid			Model		
	Scene 1	Scene 2	Scene 3	Scene 1	Scene 2	Scene 3
Minimum	0.67	0.62	0.64	0.09	0.12	0.09
Maximum	1.21	1.17	1.18	0.65	0.62	0.48
Mean	0.9	0.85	0.8	0.35	0.38	0.32
Median	0.95	0.88	0.74	0.34	0.36	0.32
Standard deviation	0.14	0.13	0.14	0.14	0.15	0.08
95th percentile	1.15	1.06	1.12	0.6	0.59	0.43
RMSE <sub>H</sub>	0.91	0.86	0.81	0.38	0.40	0.33
RMSE <sub>x</sub>	0.23	0.17	0.25	0.21	0.20	0.18
RMSE <sub>y</sub>	0.88	0.84	0.77	0.31	0.35	0.27

The sample mean error of the data is calculated in the following way:

$$\bar{e}_{xy} = \frac{1}{n} \sum_{i=1}^n \sqrt{\Delta x_i^2 + \Delta y_i^2} = \frac{1}{n} \sum_{i=1}^n e_{x_i y_i}. \tag{8}$$

Other measures for the accuracy are percentiles. The  $P$ th percentile is the value below which a given percentage  $P$  of observations fall. For example, the 95th percentile has at least 95% of the observations being less than or equal to that value. A percentile is obtained by first computing the ordinal rank  $r_P$  of the  $P$ th percentile ( $0 \leq P \leq 100$ ) in an ascending sorted list of the samples with size  $n$ :

$$r_P = \frac{P}{100} \times n. \tag{9}$$

If  $r_P$  is no integer, the values of the closest ranks to  $r_P$  are interpolated. In this way the 25th (1st quartile  $Q_1$ ), the 50th (2-quantile and sample median), 75th (3rd quartile  $Q_2$ ) and the 95th percentile have been determined. In order to measure the spread of the errors and therefore to measure the precision, the sample standard deviation of the errors is used to indicate the variability of the differences:

$$s_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n e_{x_i y_i}^2 - \bar{e}_{xy}^2}. \tag{10}$$

In table 1, all given valuation methods are applied for the three scenes conducted by the car. Moreover, all collected data for the centroid-projection approach and the model-based approach for either the untracked and tracked scenario is pooled and shown in figure 7.



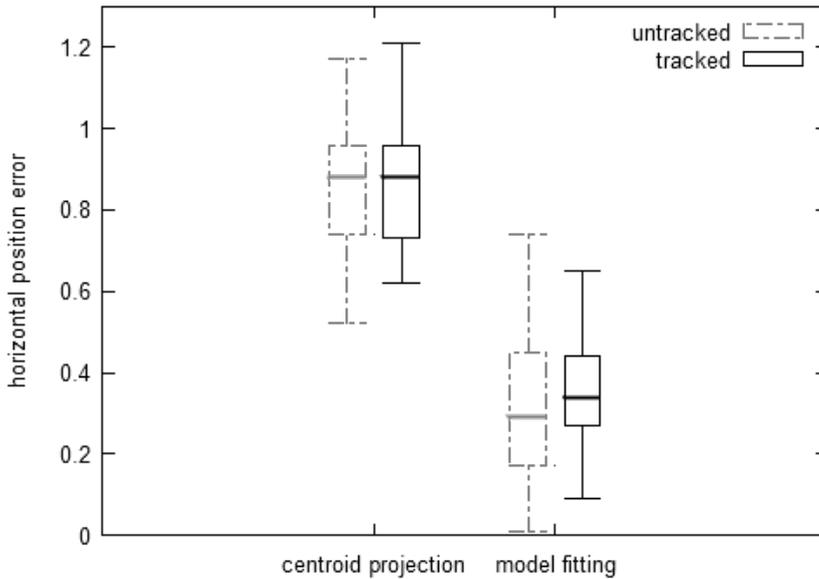


Figure 7: Errors of centroid projection and model fitting for untracked and tracked data of all scenes. The ends of the whiskers represent minimum and maximum errors.

As figures 4a and 4b show, the centroid positions (blue) have a systematic projection error, which locates all of them besides the real positions (green). The model fitting positions are closer to the GPS in all data sets, i.e. have smaller error values than the projected centroids. The systematic error of the centroid projection method could be reduced significantly—the box-and-whisker plot (figure 7) shows this clearly. The results table reveals the  $RMSE_y$  is reduced the most as expected: along the viewing direction the centroid projection produces the method-inherent offset. Thus, the model-based approach decreased the mean error by 51 cm, which is 59% less and the maximum error by 56 cm, which is 46% less. The reduction applies for all statistics except for the scatter. The sample standard deviation of errors is similar for both methods. Also the difference of maximum and minimum error (range of error values) remains almost unaffected, as well as mean and median errors lie close to each other, indicating the distribution of errors is not skewed.

The tracked position errors were expected to be even lower due to less outliers, but the graphs in figures 5 and 6 show that they are only a bit smoother (red) than the untracked ones (orange) and at certain positions even higher, as shown in figure 7.

## 5 Conclusion and outlook

As expected, the model projection considerably improved the accuracy compared to the simple centroid projection. The influence of the tracking is mainly an increase of precision, but also the elimination of the formerly smallest and biggest errors. This is due to the fact that errors tend more to the mean.

Further research can be done concerning the cost function and examining other ones considering their convergence, performance and accuracy, e.g. contour adaption [11, 12] or edge segment adaption [13]. Furthermore, the *Koller* model parameters have the convenient purpose to easily change the vehicle's vertices (e.g. shorter trunk, higher roof). These parameters can be adjusted as well using the minimization algorithm. On the one hand the model can be perfectly adapted, but as well errors could be pushed into the parameters. Thus, the impact of using more degrees of freedom on the accuracy is interesting to be examined. Instead of a simplex adjustment a Gauss–Newton or Levenberg–Marquardt algorithm can be used, which requires to formulate the rather complex cost function, containing projection, lens distortion and set operations, as a differentiable equation, which will be done soon. This is expected to yield faster convergence.

More experiments can be conducted to examine the dependency of errors from the distance of the observed car from camera.

## References

- [1] Koller, D., Daniilidis, K. & Nagel, H.H., Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, **10(3)**, pp. 257–281, 1993.
- [2] Haag, M. & Nagel, H.H., Combination of edge element and optical flow estimates for 3d-model-based vehicle tracking in traffic image sequences. *International Journal of Computer Vision*, **35(3)**, pp. 295–319, 1999.
- [3] Rommel, M., *Dreidimensionale Modellierung für die Objektverfolgung in Straßenverkehrsszenen*. Master's thesis, Humboldt-Universität zu Berlin, 2007.
- [4] Reinert, B., *Untersuchung nichtlinearer Methoden zur Berechnung der Kamerapose aus Punkt- und Geradenkorrespondenzen im Kontext der Analyse durch Synthese*. Master's thesis, Universität Koblenz-L., 2011.
- [5] Schwarzenberg, G., *Objektverfolgung mit Partikel-Filtern*. Master's thesis, Universität Karlsruhe, 2004.
- [6] Zhang, Z., Li, M., Huang, K. & Tan, T., 3D model based vehicle localization by optimizing local gradient based fitness evaluation. *2008 19th International Conference on Pattern Recognition*, pp. 1–4, 2008.
- [7] Reinbacher, C., Ruther, M. & Bischof, H., Pose Estimation of Known Objects by Efficient Silhouette Matching. *2010 20th International Conference on Pattern Recognition*, pp. 1080–1083, 2010.



- [8] Zheng, Y. & Peng, S., Model based vehicle localization for urban traffic surveillance using image gradient based matching. *15th International IEEE Conference on Intelligent Transportation Systems*, 2012.
- [9] Nelder, J.A. & Mead, R., A simplex method for function minimization. *Computer Journal*, **7**, pp. 308–313, 1965.
- [10] Welch, G. & Bishop, G., An introduction to the Kalman filter. Technical report, Chapel Hill, NC, USA, 1995.
- [11] Dahlkamp, H., Ottlik, A. & Nagel, H.H., Comparison of edge-driven algorithms for model-based motion estimation. *First International Workshop on Spatial Coherency for Visual Motion Analysis*, 2004.
- [12] Pece, A.E.C., A comparison of active-contour models based on blurring and on marginalization. *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.
- [13] Koller, D., Moving object recognition and classification based on recursive shape parameter estimation. *Proc. 12th Israel Conference on Artificial Intelligence, Computer Vision*, 1993.

