Proceedings of the
2007 IEEE Intelligent Vehicles Symposium
Istanbul, Turkey, June 13-15, 2007

Odometry-Based Structure from Motion

P. Woock, F. Pagel, M. Grinberg, D. Willersinn
Fraunhofer IITB
Fraunhoferstr. 1, 76131 Karlsruhe
{philipp.woock|frank.pagel|michael.grinberg|dieter.willersinn}@iitb.fraunhofer.de

Abstract -- **Structure from motion refers to a technique to obtain 3D information from consecutive images taken with a moving monocular camera. In order to do this, the camera motion performed between two consecutive images needs to be known.**

In the work reported in this contribution, we investigated the precision of the odometry data of a commercially available passenger car.

In order to identify the required precision, we developed an error model based on camera parameters and the bicycle model. We investigated two options, both being based on speed measurements. The first one uses steering angle measurements, the second one uses measurements of the yaw rate.

Concluding, we found out that the specified precision of all odometry data available is sufficient to solve structure from motion. Long-term measurements empirically confirm the precision values given in the specification.

This result encouraged us to actually implement a structure-from-motion approach which yields depth information as predicted from the theoretical considerations.

Further work needs to be carried out in order to compensate for roll motions.

Index terms—**structure from motion, scene reconstruction, advanced driver assistance systems, pre-crash sensing, integrated safety systems**

I. INTRODUCTION

In order to support car drivers in everyday traffic situations optical systems are often used to recognize the environment of the car. In many cases 2D image information is not sufficient because of missing depth information. Therefore a binocular set-up is frequently used. Alternatively a single camera mounted on a moving vehicle can be used to extract the 3D-structure of the scene. This method is called Structure-from-Motion (SfM) [Jer91]. SfM can also be used as a fallback solution for binocular set-ups, e. g. if one camera is affected due to dirt or technical breakdown.

For the extraction of absolute 3D data using SfM one needs to know the distance and the relative orientation between two consecutive images acquisition positions (Figure 1). The knowledge of these parameters allows the usage of the same algorithms for monocular set-ups as being used for binocular set-ups.

The estimation of the camera motion is usually done via the F-Matrix ([Har00], [Luo97]). It provides only a scaled translation vector and requires thus assumptions about the scenery (e.g. no independently moving objects) and is computationally expensive. In order to obtain the actual length of the vector without any a-priori knowledge about the observed scene, odometry data is required. Having these data an absolute 3D-reconstruction is possible.

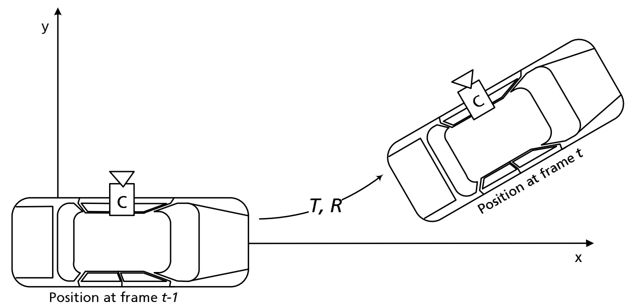

Figure 1: Platform motion yields stereo vision geometry.

The objective of the work reported in this contribution is to investigate whether the precision of the odometry data available from a commercially available passenger car is sufficient to perform this task without adding any additional sensors to the car. As shown in Figure 1 we consider a side-looking video camera which is used for a side pre-crash application.

As a first step we developed an error model which enables us to derive the required precision of the odometry data (Section II.B). We then experimentally determined the actual precision by means of a dead reckoning path reconstruction. The resulting accumulated errors confirm that the available precision is sufficient to perform SfM.

II. REQUIREMENT ANALYSIS

A. Basic requirement consideration

A.1 Refresh rate

To assess the minimum frame rate that is necessary to operate the system, we made the following assumptions (Figure 2):

To detect an object it needs to be fully visible in at least two consecutive frames. With a given camera aperture angle of 80°, and a maximum velocity of the camera-carrying vehicle of 100 km/h, we assume observed objects of width of 2 m or less at a distance of at least 2 m. We obtain a minimum frame rate of 20.48 Hz as a lower bound for such a system.

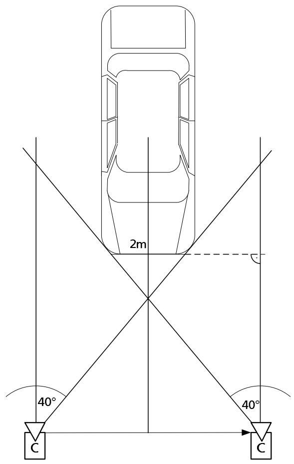

Figure 2: Object size and distance to determine minimum sampling rate.

A.2 Camera

The camera we use is a Basler A601f-HDR. The sensor has a resolution of 640x480 (VGA). Each pixel measures 9.9 x 9.9 µm.

B. Error Model

Our error model has two parts. One describes the rotational requirement, the other one describes the translational precision requirements. The required precision is based on the assumption that the image on the camera sensor should not be displaced by more than one pixel from the calculated position. The rotational error model is based on the bicycle motion model [Zom87] (Figure 3).

For the translational error model we first have to specify a worst case scenario as the required one-pixel accuracy is depth-dependent. We defined 2 m distance from the side-looking camera as worst case.

The vehicle provides the following odometry data: velocity v, yaw rate $\dot{\psi}$ and steering angle δ. Other known quantities are the wheel base l and the length l_h between the yaw rate sensor position S and the rear axle as shown in Figure 3.

B.1 Bicycle motion model

According to the bicycle model we assume that the car is performing a circular motion around the ICR (instantaneous centre of rotation) with a fixed radius R_h. R_h may vary between two consecutive frames. For straight-ahead motion, the ICR lies at infinity.

The model doesn't include any roll or pitch motion. Furthermore, it doesn't cover neither skidding nor tyre slip.

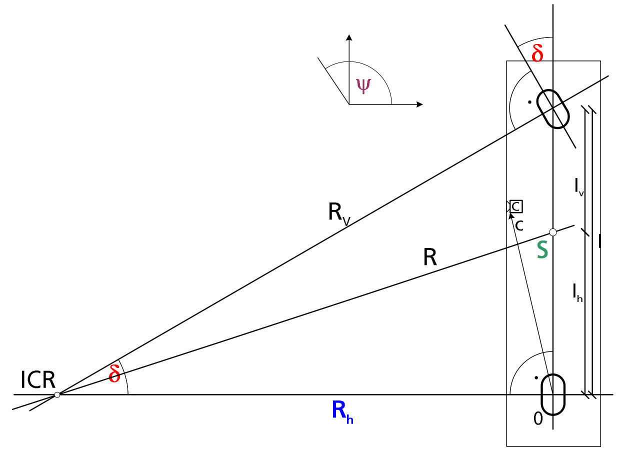

Figure 3: Bicycle motion model.

R_h can be derived from the odometry data in two different manners.

From the steering angle δ:

$$\tan\delta = \frac{l}{R_h} \quad \Leftrightarrow \quad R_h = \frac{l}{\tan\delta}$$

From the yaw rate $\dot{\psi}$:

$$\dot{\psi} = \frac{v}{R} \quad \Leftrightarrow \quad R = \frac{v}{\dot{\psi}}$$

$$R_h = \sqrt{R^2 - l_h^2} = \sqrt{\frac{v^2}{\dot{\psi}^2} - l_h^2}$$

B.2 Angular accuracy

To find the necessary angular precision, we calculate the camera rotation angle α that leads to a displacement of 1 pixel. α has different values for the central area of the camera sensor and its marginal area. In the remainder we use the stricter requirement from the sensor margin (Figure 4).

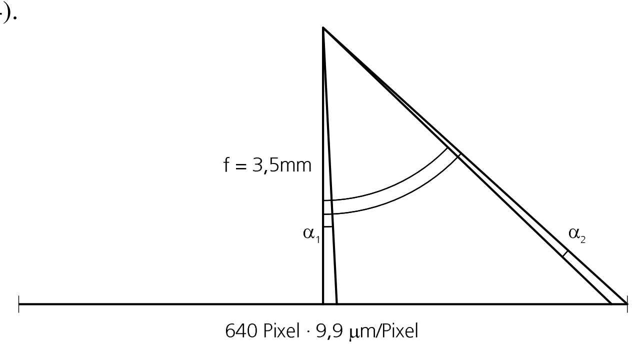

Figure 4: Angular precision from the geometry of the camera sensor.

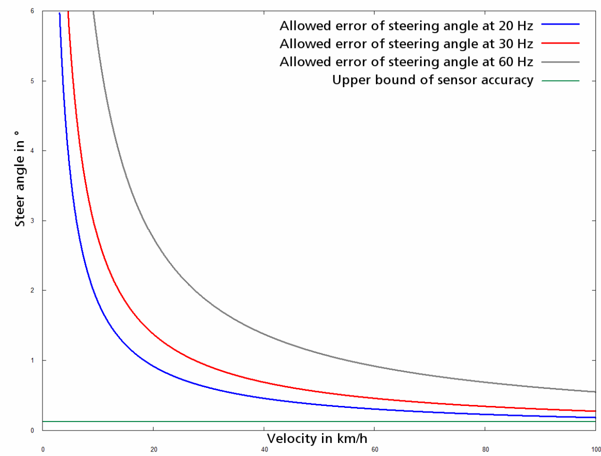

Figure 5: Required and actual precision using steering angle.

The required precision of the steering angle is

$$\delta = \tan^{-1}(\frac{\alpha l}{v\Delta t})$$

This leads to problems in the case of very fast motion combined with a very low refresh rate (e.g. more than 100 km/h and frame rates less than 20 Hz) as can be seen in Figure 5.

Similarly we obtain for the yaw rate

$$\dot{\psi} = \pm\frac{|\alpha||v|}{\sqrt{v^2(\Delta t)^2 + \alpha^2 l_h^2}}$$

Using yaw rate measurements, the difference between required and actual precision is higher than using steering angle measurements and it remains asymptotically constant for higher velocities as depicted in Figure 6.

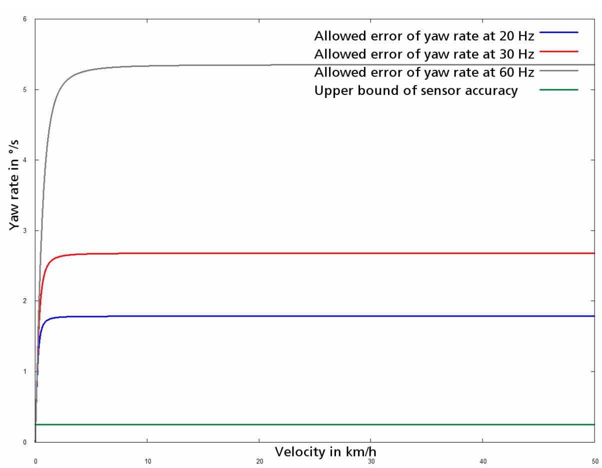

Figure 6: Required and actual precision using yaw rate.

B.3 Translational accuracy

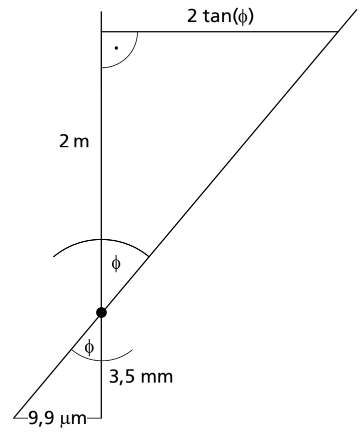

Figure 7: Required translational precision

The one-pixel displacement granularity is depth-dependent for translational motion. Assuming that objects are at a range of at least 2 m, we yield a required translational precision of 5.66 mm. At a speed of 100 km/h and an aperture angle of 40° at 20 Hz refresh rate the necessary precision of the estimated translation must be within 0.41%.

To illustrate how strong this requirement is one may perform the following calculation: The maximum translational error which may result from acceleration of 10 m/s² at the sampling frequency of 10 Hz offered by the vehicle may in the worst case become as large as 5 cm per sampling interval. However, we will see later that path reconstruction works quite well even if it is not perfectly met.

III. Experiments

A. Path reconstruction

To evaluate the data we did a path reconstruction using dead reckoning. We concatenated the circular trajectory arcs calculated with the linear bicycle model. This yields a non-differentiable path which could not be driven with a car in reality. But assuming a sufficiently high sampling rate, this is a good approximation to the real trajectory.

It is obvious that the better the reconstructed trajectory matches the reference track, the more precise is the position estimation between two consecutive frames.

Our first approach showed that the path reconstruction did not properly match reality. We found out that this is because of the non-linearities of the odometry data. Our first linear compensation approach helped to correct the velocity estimation but could not cope with the errors in estimation of the turning radius. Therefore we developed a nonlinear correction model that represents the actual situation (nonlinearites in the steering linkage) better. We found the yaw rate data also to exhibit nonlinear behaviour.

The nonlinear model we use for correcting the angles is “simple squashing” (SQ_1). A more sophisticated model would be “sigmoid squashing” (SQ_2). These functions are defined as

$$SQ_1 = 1 - \frac{1}{\frac{1}{a} + \frac{s}{|\delta|}} \qquad SQ_2 = 1 - a \cdot \frac{1}{1 + e^{(s|\delta - o|)}}$$

The parameter a denotes the range of correction that is applied, s describes the smoothness of the transition. The parameter o denotes the offset where the turning point of the sigmoid function is located.

The path reconstruction is done separately for steering angle and yaw rate measurements.

After calculating the calibration parameters the reconstruction matches the reference satisfyingly well (Figure 10).

B. Test scenarios and parameter estimation

We defined two scenarios for our experiments:

In the first scenario we drove 60 m straight forward and backward in a manner that the endpoint was exactly the same as the initial point. This has been done to check the measured length of the track and the behaviour at very little curvature. Results of the reconstruction of the forward-backward scenario using yaw rate are shown in Figure 8.

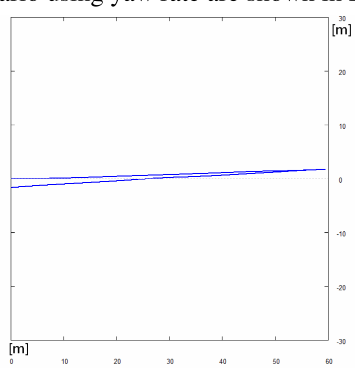

Figure 8: Path reconstruction of the straight scene with unified correction parameters (forward-backward scenario).

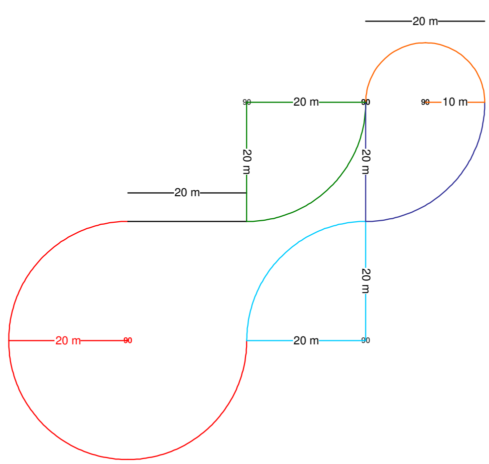

Figure 9: Test circuit to estimate the parameters of SQ_1 and SQ_2.

The second scenario was driving the circuit depicted in Figure 9. Here we wanted to test the performance on a curvy track. This test was performed four times. Slow and fast, each clockwise and counter-clockwise. In this way, all steering direction changes have been covered as this circuit contains several left to right and right to left changes as well as curvy to straight and straight to curvy changes. Those runs were used in order to calibrate the correction parameters.

Combining the parameters of each run, we obtained a unified parameter set that improves the results drastically with respect to the uncorrected results.

It turned out that the reconstruction based on steering angle was always less accurate than the reconstruction based on yaw rate. Even using the more versatile SQ_2 function did not yield a path reconstruction that was as good as the result based on the usage of the yaw rate.

Figure 8, Figure 10 and Figure 11 show some of the yaw rate-based results. Each sequence used for these tests had 2500 frames at a frame rate of 30 Hz. Figure 10 shows uncalibrated and calibrated results of a circuit run. The sequence used in Figure 11 is kind of a "real life" sequence because it contains strong acceleration as well as braking, parts with low curvature, parts with high curvature and different kinds of steering changes. At this run the true trajectory intersects itself. The calibrated reconstruction gives results which come much closer to reality.

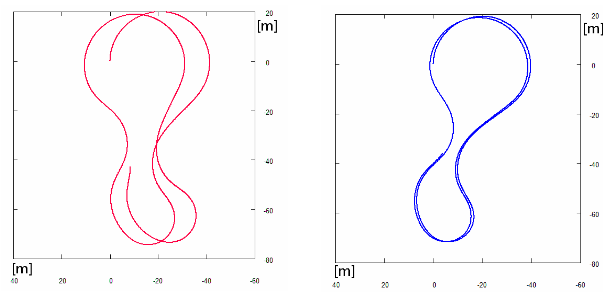

Figure 10: Path reconstructions via yaw rate of the circuit runs before (red) and after (blue) applying the calibration.

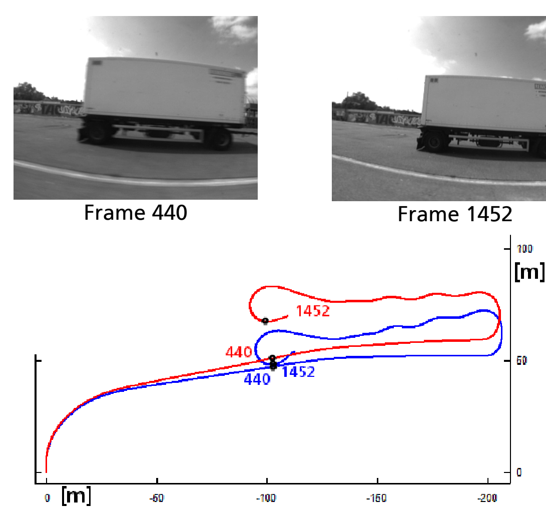

Figure 11: "Real life" test run and path reconstruction using uncalibrated yaw rate (red) and calibrated yaw rate (blue). Frame 440 and frame 1452 are the frames where the true trajectory intersects itself.

C. Practical Field Test: Structure from Motion

To test the odometry data in a practical manner we integrated these data into an already existing stereo system for side crash pre-sensing [Aprosys]. Instead of using two calibrated cameras with known relative orientation we used the odometry data to determine the extrinsic parameters between the image acquisition positions of two consecutive images of one single camera which is mounted on the side of the vehicle. Furthermore the camera has been calibrated with respect to the vehicle coordinate system. Thus the rotation and the translation of the vehicle can be transformed to the camera coordinate system. Once these parameters are known (in particular the translation and the rotation of the camera centre) we can calculate sparse 3D density maps with the same algorithm that is already implemented in the stereo system.

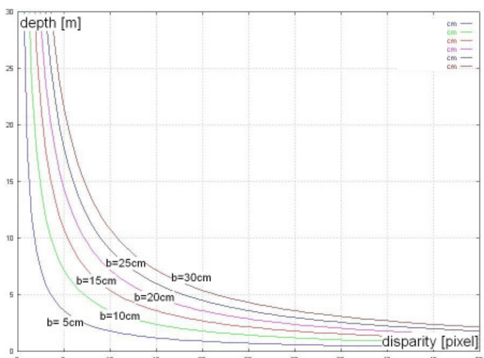

Figure 12: Reconstructed depth value subject to rectified disparity and baseline b **with fixed focal length** ***f*=360 [pixel].**

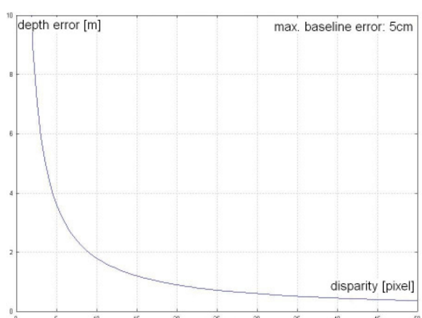

Figure 13: The reconstruction error subject to rectified disparity with fixed focal length *f*=360 [pixel] and a maximum baseline error of 5 cm.

Using rectification methods (e. g. [Tru00]), one can calculate the depth coordinate Z *w.l.o.g.* as

$$Z = \frac{f \cdot b}{d},$$

with focal length f, baseline b and disparity d.

Figure 12 demonstrates how Z behaves with respect to the disparity and to different baseline values. The focal length f is fixed at 360 pixel which is approximately the value of the wide-angle camera we used for the real tests. As one can see in the diagram a erroneous magnitude of the translational motion leads to different depth values. In addition Figure 13 shows how the error of the calculated depth value strongly increases as the disparity decreases. This graph may be seen as an upper boundary as we established a maximum translational error of 5 cm.

We verified the utility of the odometric data by displaying corresponding epipolar curves in two consecutive frames, using the extrinsic parameters resulting from the motion of the vehicle. As corresponding points in two images have to lie exactly on these curves, it is easy to see whether the calibration parameters are good or not. So once the correspondent epipolar curves are known, reconstruction can be done using standard algorithms ([Dang02]). Figure 14 shows an example of a scene on a parking lot with a straight forward motion and a velocity of 24 km/h. Further analysis has shown that even in curvy motion the epipolar curves corresponded well.

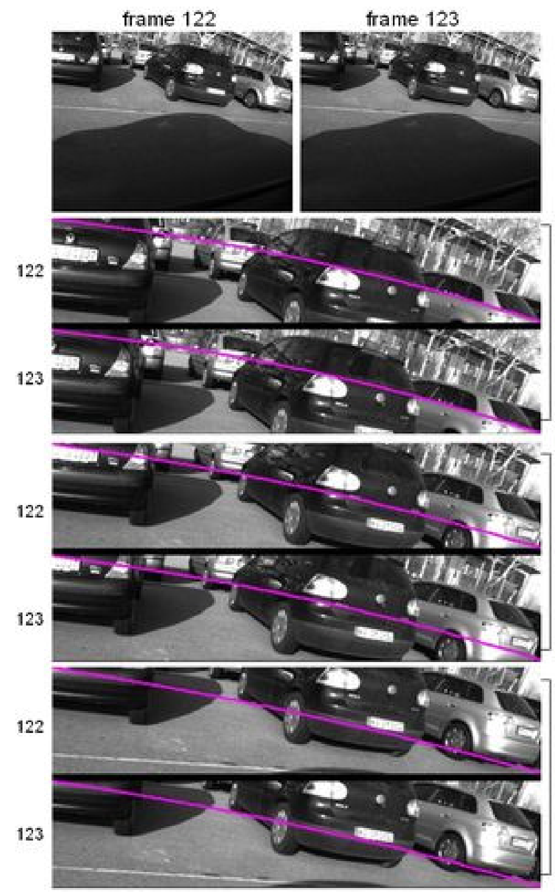

Figure 14: Correspondent epipolar curves, generated using odometry data. Frame 122 and 123 are taken from the *Karlsruhe parking lot* sequence.

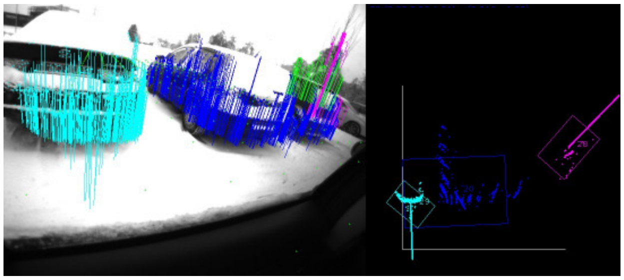

Figure 15: 3D analysis of the *Lindau parking lot* sequence with monocular SfM using odometry data.

Finally a real test performance run was done. To simplify matters we chose a rigid scene. The recordings for the test series were sampled with 40 Hz at a nearly constant speed of 25 km/h. This corresponds to b=~0.17 m. The whole parking lot sequence (Figure 15) contains 1200 images. Calculated 3D points and their projections onto the calibrated ground plane are displayed both in the top view (right part of Figure 15) and in the original image (left part). So every 3D point is visualized in the camera image as a line. This representation helps to identify outliers. The top view of the scene also shows the world coordinate frame with the horizontal *x*-axis headed to the tangential direction of motion. Additionally the [Aprosys] algorithm returns object hypotheses (illustrated as boxes in Figure 15) and the objects' estimated motion directions. The shape of the parking cars are well visible.

Figure 15 reveals a "fishbone" effect. That means that adjacent 3D points are not reconstructed continuously but spread as is visible in the top view. This artefact is caused by the usage of integer feature point coordinates and relatively small baseline values when calculating SfM. For this reason it is strongly recommended to calculate subpixel accurate feature point coordinates to avoid this phenomenon.

IV. Analysis

A. Data quality

Our investigations show that an assumed steering angle granularity of 1/6° as an upper bound is a reasonable value considering the nonlinearities in the steering mechanics. This is enough for velocities up to 100 km/h and at least a 20 Hz refresh rate.

The coarsest quantization in the yaw rate data provided by the vehicle is 0.24°/s which is far below the acceptable error.

The velocity values 2.2 km/h are all reported as zero. This is probably due to a quantizer dead zone. The resolution is 0.1 km/h. One has to keep in mind that the velocity data is only updated every 100 ms and therefore is reported as constant throughout several frames. Data quality does not meet the requirements seen in Section II.B.3.

Furthermore we showed that odometry data is sufficient to do an absolute, metric 3D reconstruction of a rigid scene.

V. Summary and Conclusion

We found out that the accuracy of the odometry data of a standard commercially available passenger car is sufficient to do Structure-from-Motion for regular motion (no skidding, standstill, etc.).

We obtained better results using the yaw rate data compared to the steering angle data. This may be because the steering angle is, unlike the yaw rate, sensitive to tire slip. Furthermore it is especially important that the steering angle around straight ahead (zero) position is reported as accurately as possible [Det05]. Probably a better understanding of the mechanical causes of the nonlinearities described in Section III.A would help.

It is important to remember that the path reconstruction as a whole is just a means to show how small the accumulated error is throughout the trajectory. This in turn implies the small error between two frames. We can rule out that these errors are cancelling out each other because the reconstruction works well on several different tracks with different characteristics.

Further studies must be done to recover the effect of full breaking, sliding, strong and fast curvatures and different velocities on the reliability of odometry data.

Acknowledgements

We would like to thank the APROSYS Subproject 6 consortium with the partners IST (Pt), Daimler-Chrysler (Ge), CIDAUT (Sp), Fraunhofer LBF (Ge), Warsaw University of Technology (Pl), TNO (Nl), Faurecia (Fr) and Siemens AG (Ge), Joachim Tandler from Siemens VDO and Jürgen Metzler from IITB.

References

[Aprosys] http://www.aprosys.com

[Dang02] T. Dang, C. Hoffmann, C. Stiller. Fusing Optical Flow and Stereo Disparity for Object Tracking. IEEE V. International Conference on Intelligent Transportation Systems, pp 112-117, Singapore, 2002.

[Det05] F. Dettki. Methoden zur objektiven Bewertung des Geradeauslaufs von Personenkraftwagen. Dissertation, Universität Stuttgart, 2005.

[Har00] R. Hartley, A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2000.

[Jer91] C. P. Jerian, R. Jain. Structure from Motion - A critical Analysis of Methods. IEEE Trans. on Systems, Man and Cybernetics 21(**3**), May/June 1991

[Luo97] Q. Luong, O. Faugeras. Self-Calibration of a Moving Camera from Point Correspondences and Fundamental Matrices. IJCV, 22(3):261--89, 1997.

[Tru00] E. Trucco, A. Verri, A. Fusiello. A Compact Algorithm for Rectification of Stereo Pairs. In Machine Vision and Applications 12, pp. 16-22, 2000.

[Zom87] A. Zomotor. Fahrwerktechnik: Fahrverhalten. Vogel Buchverlag, Würzburg, 1987.