3D Pose Estimation of Vehicles Using a Stereo Camera

Björn Barrois, Stela Hristova, Christian Wöhler
Daimler AG
Group Research, Environment Perception
P.O. Box 2360, D-89013 Ulm, Germany
{Bjoern.Barrois, Stela.Hristova, Christian.Woehler}@daimler.com

Franz Kummert, Christoph Hermes
Bielefeld University
Faculty of Technology, Applied Computer Science Group
P. O. Box 100 131, D-33501 Bielefeld, Germany
{Franz, CHermes}@techfak.uni-bielefeld.de

***Abstract*—This study introduces an approach to three-dimensional vehicle pose estimation using a stereo camera system. After computation of stereo and optical flow on the investigated scene, a four-dimensional clustering approach separates the static from the moving objects in the scene. The iterative closest point algorithm (ICP) estimates the vehicle pose using a cuboid as a weak vehicle model. In contrast to classical ICP optimisation a polar distance metric is used which especially takes into account the error distribution of the stereo measurement process. The tracking approach is based on tracking-by-detection such that no temporal filtering is used. The method is evaluated on seven different real-world sequences, where different stereo algorithms, baseline distances, distance metrics, and optimisation algorithms are examined. The results show that the proposed polar distance metric yields a higher accuracy for yaw angle estimation of vehicles than the common Euclidean distance metric, especially when using pixel-accurate stereo points.**

I. Introduction

For future advanced driver assistance systems the pose and motion of oncoming and intersecting vehicles represent an important information. Some approaches for detection of obstacles around the ego-vehicle can be found in the literature, but the existence of an obstacle is not enough information to evaluate the situation entirely. The knowledge about pose and motion of oncoming and intersecting vehicles is important to avoid accidents. Especially at intersections, analysis of the behaviour of other road users is important to help the driver to pass through the intersection in a safe and comfortable manner.

A laser scanner and a video camera are used in [1] to estimate the pose, size, and velocity of vehicles. The detection of vehicles is based on the laser scanner data, after that a cascade classifier uses the camera images to classify the objects in front of the ego-vehicle. The evaluation in [1] is limited to the classification performance of the system, while the geometric detection accuracy is not assessed.

Stereo analysis based on affine warping is used in [2] to detect obstacles in front of the ego-vehicle. The image of the left camera is warped and mapped onto the image of the right camera using affine warping and edge comparison. Dynamic Programming estimates a boundary between the road surface and obstacles on the road. The poses of the obstacles are not estimated.

In [3] the scene in front of the ego-vehicle is reconstructed using both stereo vision and structure from motion. The objects are detected and classified using the approach introduced in [4]. The results are only reliable in a depth range between 15 and 30 metres. The pose of the objects is only roughly estimated and not compared to ground truth data.

In contribution [5] a stereo camera system is used for several applications (e.g. detection and tracking of vehicles and pedestrians). First, an occupancy grid is created using the 3D points of an initial stereo calculation to detect objects in front of the car. When an object is found, the size and aspect ratio of the object are used to find a corresponding model in an object model database. The 3D model from the database is projected into the image and fitted to the image of the object using chamfer matching. The distance range for reliable results is about 35 m. A statement about the pose accuracy of the objects is not given.

The approach introduced in [6] uses a sparse scene flow field which is established by a Kalman filter based fusion of stereo vision and tracked optical flow vectors. An Extended Kalman Filter tracks the point cluster representing the vehicle. An explicit vehicle model is not utilised, but the size of the vehicle is estimated regarding the size of the point cloud. The settling time of the filter amounts to about 25 frames for the sequences examined in [6].

The approach in this study aims for a high accuracy of the pose estimation without relying on temporal filtering. We thus perform a tracking-by-detection approach where the pose estimation is performed individually for each frame. The method is evaluated using ground truth data and real-world sequences. We use a sparse scene flow field to have reliable depth information combined with motion information about the investigated scene. After clustering of the four-dimensional point cloud (three-dimensional spatial coordinates and one-dimensional horizontal motion information), object clusters are used to estimate the object pose. The approach is evaluated on seven different real-world sequences including slowly and fast moving vehicles on an intersection, captured with three different stereo baseline distances and three different stereo algorithms.

II. 3D Tracking System

The tracking system is based on the combination of stereo and optical flow computation. First, the input images from a

calibrated camera system are rectified to standard geometry with epipolar lines parallel to the image rows. We use three different algorithm combinations which are described in the following sections to calculate a sparse incomplete scene flow field. Afterwards, the 3D points are clustered into separate objects. The 3D points of an object are used for the ICP algorithm to estimate the object pose, where the classical Euclidean distance or the polar distance metric introduced in this study can be used.

A. Spacetime Stereo

The spacetime stereo algorithm is based on local intensity modelling and yields 3D points with the associated motion component parallel to the epipolar lines [7]. Image regions corresponding to a sufficiently high vertical intensity gradient are extracted in the left and right camera image, and their local spatio-temporal neighbourhood is modelled by the model function $h(\vec{P}, u, v, t)$, where u and v denote the pixel coordinates, t the time coordinate in a spatio-temporal region of interest, and $\mathbf{P}$ the vector of function parameters:

$$h(\vec{P}, u, v, t) = p_1(v,t) \tanh\left[p_2(v,t)u + p_3(v,t)\right] + p_4(v,t) \quad (1)$$

The $\tanh$ function approximates the shape of an ideal edge blurred by the point spread function of the optical system. Correspondence analysis is then based on a comparison of the modelled edges in the left and the right image. The SSD values on each epipolar line are analysed, where different constraints can be taken into account: uniqueness, ordering, or the minimum weighted matching constraint [8]. Analysis of the model function parameters yields the velocity component along the epipolar line and subpixel accurate disparity values.

B. Feature-based stereo and optical flow

The utilised feature-based method for computing stereo and optical flow is based on Haar Wavelets to obtain a feature representation of the investigated scene [9]. The wavelet features are established in both images (left and right camera or current and previous timestep), then the comparison is performed using a hash table technique. This approach provides a fast implementation of the algorithm. The correspondences have pixel accuracy, which is a disadvantage in comparison with the other regarded stereo algorithms. The combination of the stereo and optical flow results yields 3D points associated with the motion components parallel to the image plane.

C. Correlation Stereo and feature-based optical flow

The combination of correlation-based stereo with the feature-based optical flow technique yields optical flow and more precise depth information in real-time. The correlation-based stereo algorithm is based on the SSD comparison of left and right image patches [10]. An interest operator (Prewitt filter) is used to separate reliable from unreliable depth information. The combination yields 3D points with two-dimensional motion vectors.

D. Clustering

An initial segmentation of the attributed 3D point cloud extracted with any of the sparse scene flow techniques (cf. Fig. 1a) is obtained by means of a graph-based unsupervised clustering technique [11] in a four-dimensional space spanned by the spatial coordinates and the horizontal velocity component of the 3D points. This clustering stage generates a scene-dependent number of clusters, essentially separating the moving object from the (stationary or differently moving) background.

E. Pose Estimation using ICP and Euclidean Metric

For pose estimation, a cuboid is utilised as a weak geometric model representing several types of vehicles. An initial pose is estimated based on the centroid and the first principal component of the vehicle point cloud. Afterwards, the iterative closest point (ICP) algorithm [12] fits the geometric model to the point cloud. Thus the translational pose parameters t_x and t_z and the yaw angle θ are updated by minimising the mean squared distance between the scene points and the model. The distance d_i is the perpendicular between the visible model side and the 3D point in Euclidean space (cf. Fig. 2a), where p_i denotes the 3D point and p_{m_i} the corresponding point on the model plane:

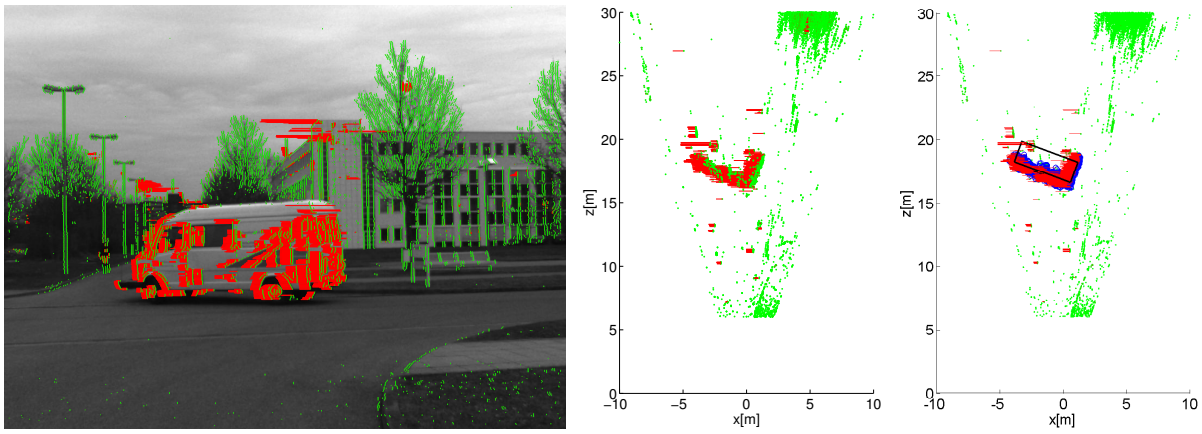
$$d_i(t_x, t_y, \theta) = \|p_i - p_{m_i}\|. \quad (2)$$

Different nonlinear optimisation methods can be used for the minimisation of the error function. In this study we use the Levenberg-Marquardt with numerically calculated derivatives and the Downhill-Simplex algorithm [13].

In [12], the ICP algorithm is applied to the registration of point sets, curves, and surfaces. Since this approach can only be used in situations where all scene points belong to the object, it is a pose estimation rather than a scene segmentation technique. In the ICP algorithm proposed in [14], the scene points and the object model are represented as sets of chained points. During each iteration step the pose parameters are updated while at the same time some scene points are assigned to the object model and others are discarded, based on the distance to the object and the similarity of the tangent directions of the scene and model curves. Thus, outliers in the 3D point cloud are automatically rejected, and the algorithm is robust with respect to disappearing and re-appearing object parts as well as partial occlusions. As a result, the subset of scene points belonging to the object, i. e. a scene segmentation, is inferred along with the 3D object pose.

F. Polar Distance Metric

The ICP algorithm with the Euclidean metric does not take into account the properties of the stereo measurement process. When reconstructing 3D points from two cameras, the relation between the disparity and the depth is nonlinear. This results in a low accuracy for points which are far away from the camera, while points close to the camera have a higher accuracy. The noise of the point positions is neither Gaussian nor symmetric. The Euclidean metric used in the



(a) Input image with 3D points. (b) Bird's eye view. (c) Clustered points.

Fig. 1. Example of an incomplete sparse scene flow field. (a) Original image with projected 3D points (green) and associated motion vectors (red). (b) Bird's eye view of the same scene. (c) Clustered points belonging to the vehicle (blue) together with the adapted model (black).

standard ICP assumes at least a symmetric error distribution of the measurements. To overcome this problem one might think of transforming the geometric model into disparity space (pixel coordinates and disparities), which is very complex depending on the model. We thus prefer using a distance metric which takes into account the nonlinearity of the measurement error.

The 3D points are represented in polar coordinates (cf. Fig. 2b) on the xz plane, where the distance is $r(x,y,z) = \sqrt{x^2+y^2+z^2}$ and the polar angle amounts to $\varphi = \arctan(x/z)$. Assuming an ideal pinhole camera model, the pixel coordinates (u,v) can be transformed into the 3D coordinates according to $u/f = x/z$ and $v/f = y/z$, where f denotes the camera constant in pixels. Assuming a standard epipolar geometry, the depth coordinate z can be calculated by $z = bf/d$, where b is the baseline distance in metres and d the disparity in pixels. Using these equations, the distance to the camera $r(u,v,d)$ is given by

$$r(u,v,d) = \sqrt{\left(\frac{bu}{d}\right)^2 + \left(\frac{bv}{d}\right)^2 + \left(\frac{bf}{d}\right)^2}. \tag{3}$$

An error calculation based on the total differential dr is used to analyse the relations of the depth measurement process:

$$dr = \frac{\partial r}{\partial u}du + \frac{\partial r}{\partial v}dv + \frac{\partial r}{\partial d}dd \tag{4}$$

Converting these terms into three-dimensional world coordinate expressions results in the following relations:

$$\frac{\partial r}{\partial u} = \frac{x}{f}\frac{z}{r}, \quad \frac{\partial r}{\partial v} = \frac{y}{f}\frac{z}{r}, \quad \frac{\partial r}{\partial d} = \frac{zr}{fb}. \tag{5}$$

The term z/r is smaller than or equal to 1. The terms x/f, y/f and z/f all have the same order of magnitude. The measurement errors du and dv of the pixel coordinates u, v and the disparity measurement error dd are of the order $0.1-1$ pixels and independent of z and r. In our scenario one can assume that $r/b \gg 1$, whereby the measurement error dr largely depends on the disparity error dd. Accordingly, dr is approximately proportional to zr.

Hence, this error analysis yields the result that the normalised measurement error $dr/(zr)$ has an approximately Gaussian distribution. The error distribution of the polar angle φ is also approximately of Gaussian shape. This results in an error function E for the model fit which consists of an error term regarding the distance of a 3D point to the model and a second error term regarding the polar angle difference, combined by the user-defined weight factor λ:

$$E(r_i, z_i, \varphi_i) = \sum_{i=1}^{N} \left[E_r^2(r_i, z_i) + \lambda E_\varphi^2(\varphi_i)\right] \tag{6}$$

The value of λ depends on the relative magnitudes of the error terms E_r and E_φ. In all our experiments we have set $\lambda = 0.03$. The first error term E_r only depends on distances from the camera, where r_i is the distance of the object point i to the camera and r_{m_i} the distance to the camera of the intersection point of the same line of sight with the model:

$$E_r(r_i, z_i) = \frac{r_i - r_{m_i}}{z_i r_i}. \tag{7}$$

The second error term E_φ only depends on the polar angles:

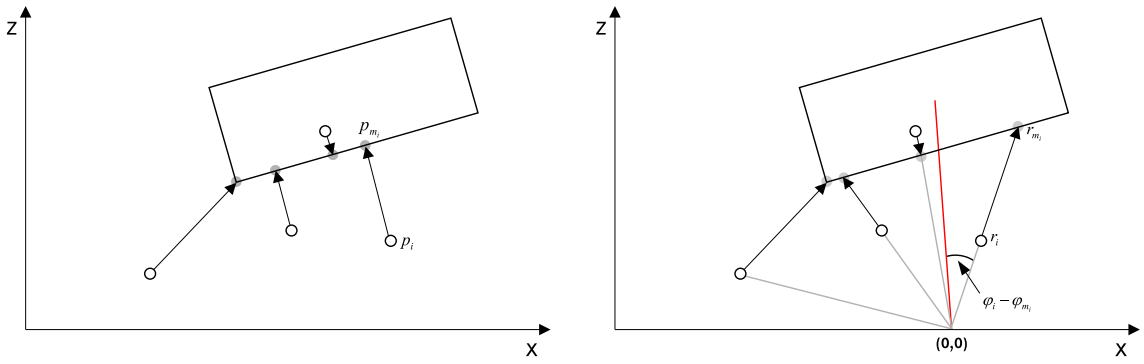
$$E_\varphi(\varphi_i) = \frac{\varphi_i - \varphi_m}{2}\left(1 + \tanh\left|\alpha(|\varphi_i - \varphi_{m_i}| - \beta)\right|\right), \tag{8}$$

where φ_m describes the polar angle of the centre of the model.

The hyperbolic tangent enforces continuity of the angular error function and is required for the convergence behaviour of the optimisation. The parameter β corresponds to about half the angular width of the model projected into the image plane, whereas α is a user defined parameter which influences the optimisation behaviour. Error function (6) can be minimised in the same manner as the Euclidean metric using a nonlinear optimisation method.

III. Experimental Evaluation

To obtain a reliable statement about the accuracy of the pose estimation results, ground truth data are generated for



(a) Common Euclidean distance metric.

(b) Polar distance metric.

Fig. 2. Comparison of the Euclidean distance metric (a) and the polar distance metric (b).

the sequences[1]. For this purpose, several coloured markers have been attached to the vehicle. We used colour cameras for image acquisition and a colour classifier to extract the markers in the images. Afterwards a highly accurate corner detector [15] estimates the corner positions in each image at subpixel accuracy, which are used for bundle adjustment to compute the three-dimensional world coordinates of each corner point. Four markers are attached to each side of the vehicle, such that the corresponding plane is overdetermined. Three colour cameras with a resolution of 1034×776 pixels were used for image acquisition. The cameras were mounted side by side at different displacements, resulting in three different baseline distances of 102, 228, and 380 mm. The frame rate of the sequences is about 14 fps. The colour information is only used for ground truth calculation, whereas for the stereo algorithms the images were converted to greyscale images. Our system performs a frame-by-frame pose estimation without any temporal filtering (tracking-by-detection). The 3D point cloud is clustered to obtain the scene part which represents the vehicle. The first principal component of this point cloud is used to initialise the angular pose parameter. The ICP-based optimisation then yields the vehicle pose.

A pose estimation is accepted if the extension of the 3D point cloud in x direction is at least 20% larger then the vehicle width, otherwise the result is regarded as unreliable because only a very small number of 3D points are present on the longer side of the vehicle. We use seven different sequences which represent typical intersection scenarios: a vehicle passing straight through the intersection, and a vehicle turning left or right with different velocities. The evaluation is based on two geometric indicators: the yaw angle difference $\Delta\Theta$ between the true yaw angle and the pose estimation result and the mean distance of the ground truth points to the corresponding model planes. The differences are visualised using error bars which are based on the median of the full sequence and the 25% and the 75% quantiles, respectively. For most of the sequences the pose estimation yields reliable results for about 90% of timesteps. Only in sequence 6 the vehicle drives nearly parallel to the z axis, leading to a small projected extension of the 3D point cloud in x direction, which yields unreliable pose estimation results for more than 90% of the frames of that sequence.

[1]Stereo image sequences and ground truth data are publicly available at http://aiweb.techfak.uni-bielefeld.de/files/VehicleSequences2009.tgz

A. Baseline Distance

The first part of the evaluation analyses the different baseline distances of the sequences. Fig. 3 shows the yaw angle error and Fig. 4 the distance error for the three different baseline distances for all seven sequences using the described three different stereo approaches. The comparison between the different baseline distances shows that the smallest baseline is not sufficient for our application. Especially for vehicles turning left or right in front of the camera (sequences 3 and 4) the errors are large. The largest baseline also produces larger errors, since due to the large camera displacement and the resulting parallax effects the stereo algorithms produce a smaller number of correct correspondences.

B. Stereo Algorithm

To evaluate the different stereo algorithms, tracking results are computed for all sequences with the intermediate baseline distance and the Levenberg-Marquardt algorithm. Figs. 5 and 6 show the results of the different stereo algorithms, where both distance metrics (Euclidean and polar) are evaluated.

The results for the yaw angle error show that the difference between the pixel accurate feature-based stereo and the other two algorithms with subpixel accuracy is negligible. The results are similar for all three stereo algorithms.

However, regarding the vehicle position, a difference between pixel and subpixel accuracy is noticeable. The distances of the feature-based stereo results show an up to three times higher error for sequences 1, 2, and 3. No significant difference between the spacetime stereo and the correlation-based stereo can be found.

C. Distance Metric

Figs. 5 and 6 show the pose estimation errors for the two distance metrics. For the position error both metrics have a similar performance, whereas the polar distance metric produces a smaller yaw angle error for most sequences. The

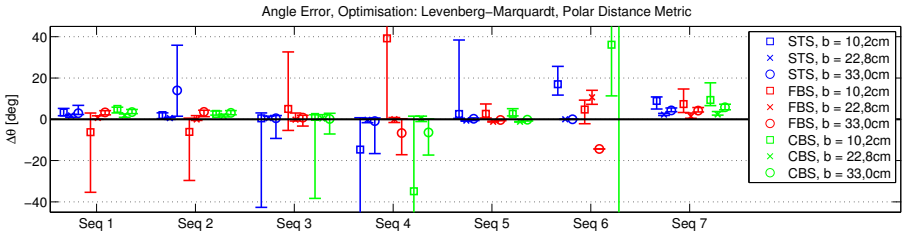


Fig. 3. Dependence of the yaw angle error on the stereo baseline, Levenberg-Marquardt optimisation, polar distance metric. Blue colour denotes the spacetime stereo technique (STS), red colour the feature based stereo (FBS), and green colour the correlation-based stereo (CBS). The small baseline is marked by squares, the intermediate baseline by crosses, and the large baseline by circles. The intermediate baseline distance is most favourable with respect to the yaw angle error.

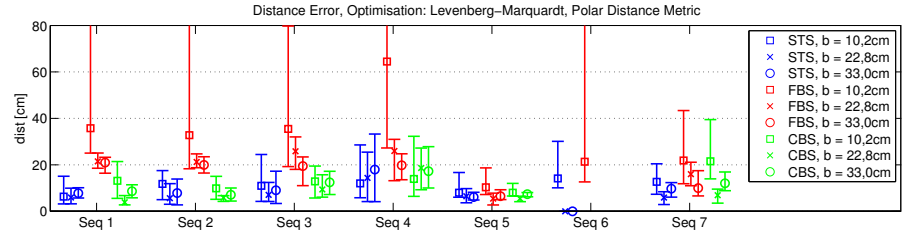


Fig. 4. Dependence of the distance error on the stereo baseline, Levenberg-Marquardt optimisation, polar distance metric. Colours and markers as in Fig. 3.

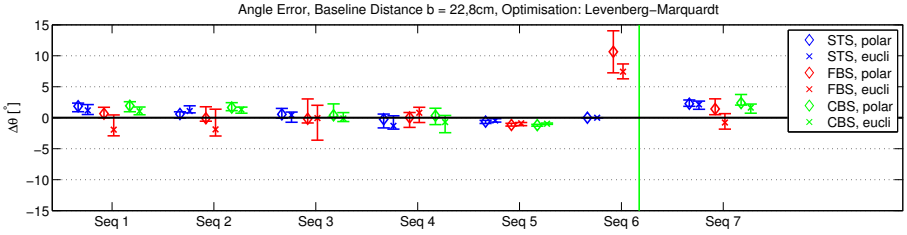


Fig. 5. Dependence of the yaw angle error on the stereo algorithm, Levenberg-Marquardt optimisation, intermediate stereo baseline. The colours (blue, red, and green) denote the stereo algorithm, while diamonds represent the polar metric and stars the Euclidean metric.

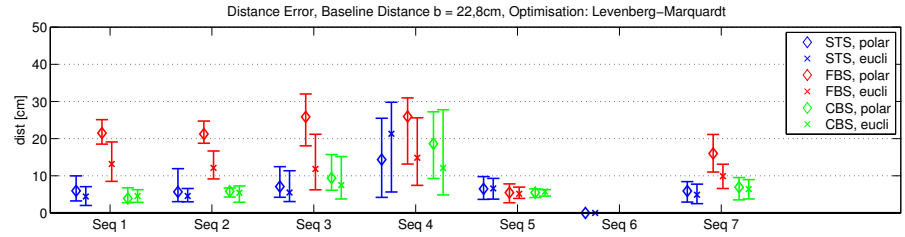


Fig. 6. Dependence of the distance error on the stereo algorithm, Levenberg-Marquardt optimisation, intermediate stereo baseline. Colours and markers as in Fig. 5.

computation time for the polar metric is the same as for the Euclidean metric.

D. Optimisation Algorithm

The results of the Downhill-Simplex optimisation are shown in Fig. 7 for the yaw angle error and in Fig. 8 for the error of the average distance between the ground truth points and the corresponding model planes.

The yaw angle errors are all higher for the Downhill-Simplex algorithm. Considering that the number of function calls of the Levenberg-Marquardt algorithm is of the same order of magnitude (about 100 function calls) as for the Downhill-Simplex algorithm with a better pose estimation accuracy, the Levenberg-Marquardt algorithm appears to be preferable. The distance errors show a similar behaviour.

IV. Summary and Conclusion

In this study we have introduced an approach for three-dimensional vehicle pose estimation using a stereo camera. After stereo and optical flow computation on the investigated scene, a four-dimensional clustering approach separates the static from the moving objects in the scene. The iterative

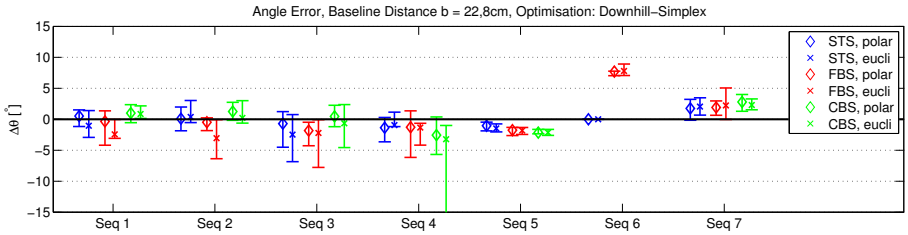


Fig. 7. Yaw angle error for Downhill-Simplex optimisation, intermediate stereo baseline. Colours and markers as in Fig. 5.

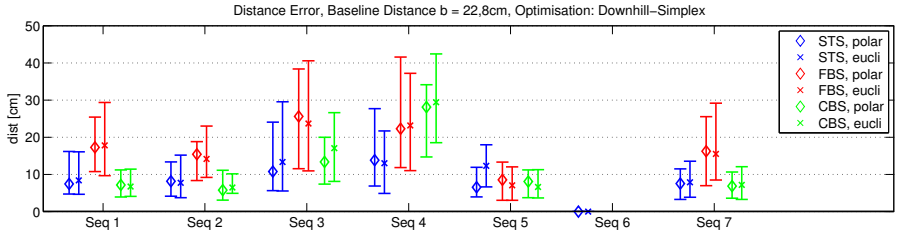


Fig. 8. Distance error for Downhill-Simplex optimisation, intermediate stereo baseline. Colours and markers as in Fig. 5.

closest point algorithm (ICP) estimates the vehicle pose using a cuboid as a weak vehicle model. In contrast to classical ICP optimisation a polar metric was used which takes into account the properties of the stereo measurement process. We have followed the tracking-by-detection approach such that no temporal filtering (e.g. Kalman Filtering) has been applied. The algorithm has been tested on seven different real-world sequences where different stereo algorithms, different baselines, different distance metrics and different optimisation algorithms have been evaluated.

The evaluation shows that for obstacles in a range between 10 and 30 m in front of the ego-vehicle, the intermediate baseline distance of 228 mm is preferable. In that scenario a subpixel accurate stereo algorithm yields up to three times higher distance accuracies when compared to a pixel accurate algorithm. To minimise the point distances in the ICP algorithm, the Levenberg-Marquardt algorithm yields better accuracies than the Downhill-Simplex optimisation while the number of function calls is comparable. The proposed polar distance metric is preferable to the Euclidean distance metric as for most sequences it yields a higher accuracy of the estimated yaw angle, especially for the pixel accurate stereo algorithm. If this most favourable configuration is employed, the pose estimation yields yaw angle errors typically smaller than 3 degrees and absolute distance errors of about 0.1 m, corresponding to relative distance errors around 0.5%.

Acknowledgments: The work described in this contribution was carried out within the research initiative AKTIV-AS supported by the German Bundesministerium für Wirtschaft und Technologie (grant no. 19S6011A).

REFERENCES

[1] S. Wender and K. Dietmayer, "3d vehicle detection using a laser scanner and a video camera," in *In Proceedings of 6th European Congress on ITS in Europe*, Aalborg, Denmark, June 2007.

[2] S. Kubota, T. Nakano, and Y. Okamoto, "A global optimization algorithm for real-time on-board stereo obstacle detection systems," in *IEEE Intelligent Vehicles Symposium*, 2007, pp. 7–12.

[3] B. Leibe, N. Cornelis, K. Cornelis, and L. V. Gool, "Integrating recognition and reconstruction for cognitive traffic scene analysis from a moving vehicle," in *DAGM Annual Pattern Recognition Symposium*. Springer, 2006, pp. 192–201.

[4] B. Leibe and B. Schiele, "Scale-invariant object categorization using a scale-adaptive mean-shift search," in *DAGM Annual Pattern Recognition Symposium*, 2004, pp. 145–153.

[5] S. Nedevschi, R. Danescu, T. Marita, F. Oniga, C. Pocol, S. Sobol, C. Tomiuc, C. Vancea, M. M. Meinecke, T. Graf, T. B. To, and M. A. Obojski, "A sensor for urban driving assistance systems based on dense stereovision," in *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 276–283.

[6] A. Barth and U. Franke, "Where will the oncoming vehicle be the next second?" in *IEEE Intelligent Vehicles Symposium*, 2008.

[7] J. Schmidt, C. Wöhler, L. Krüger, T. Gövert, and C. Hermes, "3D scene segmentation and object tracking in multiocular image sequences," in *The 5th International Conference on Computer Vision Systems Conference Paper*, 2007. [Online]. Available: http://biecoll.ub.uni-bielefeld.de/volltexte/2007/29/

[8] G. Fielding and M. Kam, "Applying the hungarian method to stereo matching," in *IEEE Conference on Decision and Control*, 1997, pp. 549–558.

[9] F. Stein, "Efficient computation of optical flow using the census transform," in *DAGM04*, 2004, pp. 79–86.

[10] U. Franke and A. Joos, "Real-time stereo vision for urban traffic scene understanding," in *Procs. IEEE Intelligent Vehicles Symposium 2000*, Dearborn, USA, 2000, pp. 273–278.

[11] B. Barrois and C. Wöhler, "Spatio-temporal 3d pose estimation of objects in stereo images," in *6th International Conference on Computer Vision Systems (ICVS)*, 2008.

[12] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, February 1992. [Online]. Available: http://portal.acm.org/citation.cfm?id=132022

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The art of scientific computing*. Cambridge University Press, 1992.

[14] Z. Zhang, "Iterative point matching for registration of free-form curves," INRIA, Tech. Rep., 1992. [Online]. Available: ftp://ftp.inria.fr/INRIA/publication/publi-ps-gz/RR/RR-1658.ps.gz

[15] L. Krüger and C. Wöhler, "Accurate chequerboard corner localisation for camera calibration and scene reconstruction," *Submitted to Pattern Recognition Letters*, 2009.