

Robust Tracking and Monocular 3D Reconstruction of Line-Based Features in Urban Environments

Kristin Schönherr,

Björn Giesler

Audi Electronics Venture GmbH

85080 Gaimersheim, Germany

kristin.schoenherr@audi.de,

bjoern.giesler@audi.de

Sonja Wahju

University of Stuttgart

sonjawahju@googlemail.com

Alois Knoll

Institute of Computer Science VI

University of Technology

Munich,

85748 Garching b. München,

Germany

knoll@in.tum.de

Abstract—Driving assistance and safety systems are based on data processing of different sensor information. The environment of a vehicle is detected by these sensing elements for accident avoidance or reduction of the accident severity. Due to package size and cost reasons, only selective sensors are used for standard-production of a vehicle. These integrated sensors have to fulfill several tasks at once and to serve different applications. Especially the processed data of a grayscale camera is used for parking assistant or lane detection system. But known from computer vision, a three-dimensional reconstruction of feature elements is possible by analyzing image sequences. This research considers corner detection to create point clouds in space. But the dominant elements of an urban environment, more precisely buildings, are characterized by edges. With this paper we focus on the robust tracking and three-dimensional reconstruction of line-based features. In this context we present our algorithm, defining and analyzing edge tolerances, which is called *Tube Principle*. The accuracy of this approach is determined by comparing the three-dimensional lines with landmarks of high-precise maps.

I. INTRODUCTION

The analysis and description of the vehicle environment are based on sensor data. Especially the aim of autonomous driving requires a reliable sensorial covering of the entire area around the vehicle. There are two possibilities to achieve a dense observation of the vehicle's environment: Increase the number of sensors or improve the sensitivity of available sensors. We take the second fact into account and analyze the possibilities of a grayscale camera to extract information about the surrounding urban area.

Known from computer vision the camera delivers more than two-dimensional image information. The pose of the camera can be estimated by comparing 3D model information with the image view, see [5] and [7]. This approach requires a certain edge extraction method.

Furthermore, the three-dimensional structure of an object can be found by analyzing image sequences over time, called structure from motion. The motion of an object, which is described by corner points, is tracked from one image to the next. The observed feature trajectories are used to reconstruct their three-dimensional position and the camera motion, compare [6]. In [13] one method of tracking and detection of point features is described. Tissainayagam et al. in [12] analyze the performance of different corner detectors

and tracking applications. The examples of pose estimation and 3D reconstruction show the potential of a single camera. Urban scenes show a complex traffic behavior and it is the challenge to evaluate such situations by driving assistance and safety systems. The classification of static and non-static objects like pedestrians is required. On closer examination of urban scenes, the line-based elements like buildings are very dominant. With our approach we focus on the detection and tracking of these line-based features for subsequent three-dimensional reconstruction.

As described in [11] and [1] existing literature distinguishes between two different types of tracking methods:

- 1) The tracking of edges based on their similarity between two images, i.e. edges are tracked by comparison of their attributes such as length, orientation and surrounding information.
- 2) The tracking of edges based on the analysis of the structure of all detected edges or a subset of those. For this kind of edge tracking considerably more geometric and topological information about the edges and their situation are necessary.

The last mentioned method is for multiple reasons not applicable. Some of those methods rely on intersection points of the edges, i.e. corner of objects to identify the structure of the edges. But due to occlusions like parking cars we only take vertical edges into account, which do not intersect by definition. Therefore, the developed tracking process fits into the first type of tracking method.

Fig. 1 gives an overview of our method which is handled by this paper. The innovative steps for more robustness and higher performance are gray highlighted. For evaluation, we compare the 3D reconstructed information with high-precise maps (GIS¹).

II. FEATURE DETECTION

There are several reasons to choose edges as features that are extracted from the images and tracked over an image sequence to eventually reconstruct them into the three-dimensional space. In the first place, they are easily identified in the images due to their change in contrast. Secondly, the

¹Global Information System

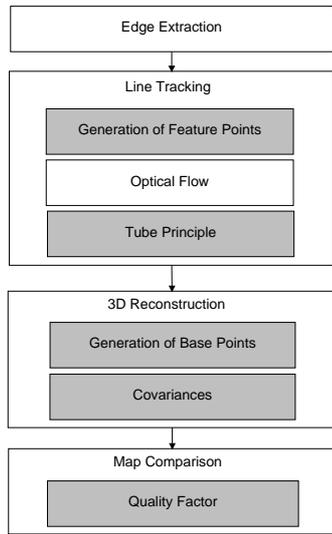


Fig. 1. Overview of robust line tracking and 3D reconstruction

images are taken by a camera which is mainly moving in the ground plane. Therefore, the limitation on vertical edge features has the advantage that the tracking direction is vertical to the characteristic contrast change. And eventually, edges allow a considerably denser description of the observed environment than for instance points do. Furthermore, in urban areas where a lot of man-made objects are located ideal conditions exist for an image processing algorithm working with nearly vertical edges.

A. Edge Extraction

For a reliable extraction of the line features different methods were investigated. The best solution for the edge extraction problem is a combination of Canny and Sobel operator. In a first step the Canny operator is used to extract all edges out of the grayscale image. Subsequently, the Sobel operator is applied to the grayscale image so that the slope direction can be received from the resulting Sobel image. Thus, every edge which slope is within a certain limited range is passed to the tracking algorithm which is described in the following section. Fig. 2 shows the reliable found image edges of an outdoor scene which illustrate a complex urban environment.



Fig. 2. Extracted edges of an outdoor scene by using Canny and Sobel operator

The only drawback of the described extraction method is

that an object edge can be detected as several edge parts due to insufficient lighting or a slight tilt. That is why the line tracking algorithm does not only have to find associations between edges of different images but also has to associate edge parts within a single image.

B. Edge Tracking by Points

A reliable detection of vertical edges in the two dimensional images is the first step to a three dimensional architectural model of the surrounding area. In a second step the detected lines have to be tracked between several images. These associations are essential for the later shown 3D reconstruction which uses the temporal associations to add a third dimension to the given two-dimensional images.

The line tracking algorithm of Tian et al. [11] does not track the edge itself but good trackable points on the edge. Feature points with a high probability of recognition in the next image are tracked and attached to the closest edge if their distances are less than one pixel. In this paper the approach of Tian et al. is extended so that points are directly distributed on the edge. The advantage is, that only points, which are already associated with an edge, have to be tracked. Moreover, these features are good to observe due to the contrast change in the orthogonal direction to the edge. By distribution of a sufficient number of points on the edge it is possible to avoid a negative influence of the aperture problem on the edge tracking.

A 2D-edge κ is defined by a start point

$$\mathbf{k}_{start} = [x_{start}, y_{start}] \quad (1)$$

and an end point

$$\mathbf{k}_{end} = [x_{end}, y_{end}]. \quad (2)$$

Additionally, every edge $\kappa_{n_\kappa}^t$ has a detection time t and a name n_κ which is an element of the set of names \mathbb{N}_K . With the help of this name the edges can be explicitly distinguished from each other and associated with edges from other time steps. Moreover, the start point is always defined by smaller y-value, i.e. $y_{start} < y_{end}$.

To track the edges from one image to the other, points $\mathbf{p}_{n_\kappa n_p}^t$ are equidistantly distributed on the edge as shown in Figure 3. At time t the gray points are scattered on the detected edges in an arbitrary distance a . Such a point on an edge is defined as

$$\mathbf{p}_{n_\kappa n_p}^t = [p_x, p_y] \quad (3)$$

at time t and can be distinguished from other points by its name $n_p \in \mathbb{N}_P^t \in \mathbb{N}_P$ (set of all point names) and the edge name $n_\kappa \in \mathbb{N}_K^t$ which allows a mapping of the point to an edge $\kappa_{n_\kappa}^t$.

The parameter a is a free selectable distance between the distributed points on the edges. In the developed algorithm a_{max} is provided to calculate the parameter a as

$$a = \frac{l_\kappa}{s} \leq a_{max} \quad (4)$$

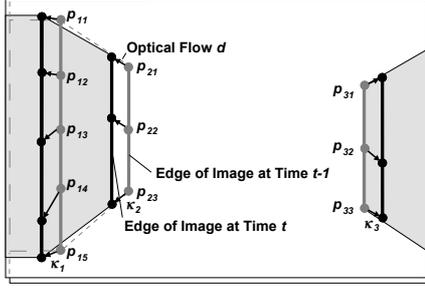


Fig. 3. Distribution of points that have to be tracked on edges

with the edge's length $l_\kappa = |\mathbf{k}_{end} - \mathbf{k}_{start}|$ and the point steps $s = \text{sup}(\frac{l_\kappa}{a_{max}})$ with $s \in \mathbb{N}$. In the presented algorithm a point is set at the start and at the end of the edge and the remaining points are distributed in between.

At the beginning of the edge tracking the edge $\kappa_{n_\kappa}^0$ is mapped to a set of points $\mathbb{P}_{n_\kappa}^0$ with the help of the difference vector between the start and end point of the edge

$$\mathbf{k}_{Diff} = \mathbf{k}_{end} - \mathbf{k}_{start} \quad (5)$$

$$\begin{aligned} \kappa_{n_\kappa}^0 &\mapsto \mathbb{P}_{n_\kappa}^0 : \\ \mathbb{P}_{n_\kappa n_p}^t &= \mathbf{k}_{start} + i \cdot a \cdot \mathbf{k}_{Diff} \end{aligned} \quad (6)$$

with $i = 0, 1, 2, \dots, s = |\mathbb{P}_{n_\kappa}^0| - 1$. Subsequently, the set of points is tracked by a pyramidal Lucas-Kanade algorithm that is described in [2]. This optical flow calculation method has the advantage that its tracking width is increased by the pyramidal implementation. In Fig. 3 the optical flow \mathbf{d} to the next image for each point is shown as an arrow so that the point set of an edge $\mathbb{P}_{n_\kappa}^t$ maps to a point set $\mathbb{P}_{n_\kappa}^{t+1}$

$$\begin{aligned} \mathbb{P}_{n_\kappa}^t &\mapsto \mathbb{P}_{n_\kappa}^{t+1} : \\ \mathbf{p}_{n_\kappa n_p}^{t+1} &= \mathbf{p}_{n_\kappa n_p}^t + \mathbf{d}_{n_\kappa n_p}^t = \begin{pmatrix} p_x \\ p_y \end{pmatrix}_{n_\kappa n_p}^t + \begin{pmatrix} d_x \\ d_y \end{pmatrix}_{n_\kappa n_p}^t. \end{aligned} \quad (7)$$

Finally, the point sets $\mathbb{P}_{n_\kappa}^{t+1} \subseteq \mathbb{P}^{t+1}$, i.e. all points that belong to an edge with the name n_κ , are mapped again with the RANSAC² [4] method to an edge

$$\begin{aligned} \mathbb{P}_{n_\kappa}^{t+1} &\mapsto \kappa_{n_\kappa}^{t+1} : \\ \text{dist}(\mathbf{p}_{n_\kappa n_p}^{t+1}, \kappa_{n_\kappa}^{t+1}) &\leq \epsilon \quad \forall \mathbf{p}_{n_\kappa n_p}^{t+1} \in \mathbb{P}_{RANSAC_{n_\kappa}}^{t+1} \subseteq \mathbb{P}_{n_\kappa}^{t+1}. \end{aligned} \quad (8)$$

From the second image on there are already points on the edges from former time steps. Those points can be shifted on the edge by the aperture problem so that new points have to be added. They are added if a part longer than a_{max} is not covered by the existing points.

C. Edge Grouping by Collection in Tubes

In some cases the edge tracking algorithm requires a combination of edges:

- 1) The chosen edge extraction method can produce a fragmented detection of an edge. Respectively, optical

²Random Sample Consensus

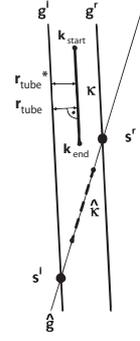


Fig. 4. Schematic demonstration of a tube

gaps or occlusions can lead to a detection of several edge parts instead of an entire edge. Due to the claim for edges as long as possible for the later 3D reconstruction, a method has to be found to merge the edge parts of the set \mathbb{K}_{parts}^t into a smaller set with complete edges $\mathbb{K}_{extract}^t$

- 2) The second case results from the edge tracking method explained before. From the second image on, both extracted edges $\mathbb{K}_{extract}^t$ and tracked edges $\mathbb{K}^{(t-1) \rightarrow t}$ are provided. The primary intention is to find the relation between the (tracked) edges from the previous image and the (extracted) edges from the current image. With an ideal calculation of the optical flow this would be possible by a simple comparison of the two sets. However, neither the given scene, the edge extraction nor the calculation of the optical flow are ideal so that the edges in the sets $\mathbb{K}_{extract}^t$ and $\mathbb{K}^{(t-1) \rightarrow t}$ belonging to the same edge in the three-dimensional space are not identical.

For these reasons the *tube principle* was developed to find associations between extracted edges of different time steps.

The edge segments are presorted in so-called *tubes*, which is an area placed around an already processed edge. If the next processed edge segment lies within such a tube it is grouped with the origin edge segment of the tube in a first step. The schematic description of a tube around an edge κ is shown in Fig. 4. It has to be mentioned that the roll movement of the vehicle can be compensated by using a gyro cluster.

For the origin edge segment with start and end point $\mathbf{k}_{start} = [x_{start}, y_{start}]$ and $\mathbf{k}_{end} = [x_{end}, y_{end}]$ the two tube boundaries \mathbf{g}^l and \mathbf{g}^r are determined with the distance of r_{tube} on the left and right side of the edge. So that the two limiting lines are defined each with two points:

$$\mathbf{k}_{start}^l = \begin{pmatrix} x_{start} - r_{tube} \\ y_{start} \end{pmatrix}, \mathbf{k}_{end}^l = \begin{pmatrix} x_{end} - r_{tube} \\ y_{end} \end{pmatrix} \in \mathbf{g}^l \quad (9)$$

$$\mathbf{k}_{start}^r = \begin{pmatrix} x_{start} + r_{tube} \\ y_{start} \end{pmatrix}, \mathbf{k}_{end}^r = \begin{pmatrix} x_{end} + r_{tube} \\ y_{end} \end{pmatrix} \in \mathbf{g}^r \quad (10)$$

For a reduction of computing time the horizontal distance r_{tube}^* is used instead of the vertical to the edge r_{tube} . This simplification can be made because all input edges are nearly

vertical so that $r_{tube}^* \approx r_{tube}$. The situation of the other edge segment $\hat{\kappa}$ to the origin edge segment κ is calculated by extending the edge to a line $\hat{\mathbf{g}}$ and intersecting it with the tube limits:

$$\mathbf{s}_l = \begin{pmatrix} x_l \\ y_l \end{pmatrix} = \hat{\mathbf{g}} \cap \mathbf{g}^l \quad (11)$$

$$\mathbf{s}_r = \begin{pmatrix} x_r \\ y_r \end{pmatrix} = \hat{\mathbf{g}} \cap \mathbf{g}^r \quad (12)$$

For the position of the two intersection points \mathbf{s}_l and \mathbf{s}_r the following possibilities exist:

- The edge is within the tube left tilted:

$$(y_l < \hat{y}_{start}) \wedge (y_r > \hat{y}_{end}) \quad (13)$$

- The edge is within the tube right tilted:

$$(y_r < \hat{y}_{start}) \wedge (y_l > \hat{y}_{end}) \quad (14)$$

- The edge is not entirely within the tube for all other combinations of the intersection points.
- For the case that there are no intersection points the two edge segments are parallel. In this case the distance between the two segments is calculated to determine whether the other edge segment is within the tube or not.

After all edge information are received and pre-sorted in edge groups like Fig. 5, we decide whether the edges are combined or partitioned as described in the following sections.

D. Combination of Extracted Edge Segments

If a new incoming edge $\hat{\kappa}_2$ is located within a tube range of a previously received edge $\hat{\kappa}_1$ a combination is done. The lowest y-value is chosen as the new start point and the highest y-value marks the new end point of the combined edge described with

$$\mathbf{k}_{start}^{12} = \begin{pmatrix} x_{start}^1 \\ y_{start}^1 \end{pmatrix}, \text{ if } y_{start}^1 < y_{start}^2 \quad (15)$$

$$\mathbf{k}_{start}^{12} = \begin{pmatrix} x_{start}^2 \\ y_{start}^2 \end{pmatrix}, \text{ if } y_{start}^2 < y_{start}^1 \quad (16)$$

$$\mathbf{k}_{end}^{12} = \begin{pmatrix} x_{end}^1 \\ y_{end}^1 \end{pmatrix}, \text{ if } y_{end}^1 > y_{end}^2 \quad (17)$$

$$\mathbf{k}_{end}^{12} = \begin{pmatrix} x_{end}^2 \\ y_{end}^2 \end{pmatrix}, \text{ if } y_{end}^2 > y_{end}^1 \quad (18)$$

Additionally, a tolerance factor can be used to allow small gaps in between the segments, which then lead to combined edges, shown as combination case in Fig. 5.

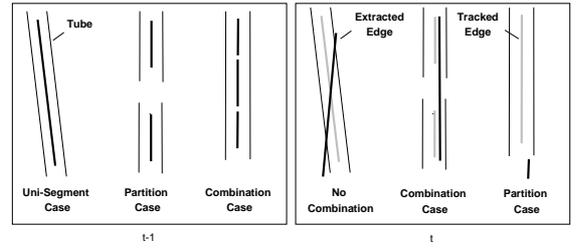


Fig. 5. Partitioning and combination of edges in single frame at $t-1$ and inter-frame analysis $(t-1) \rightarrow t$

E. Combination of Tracked and Extracted Edges

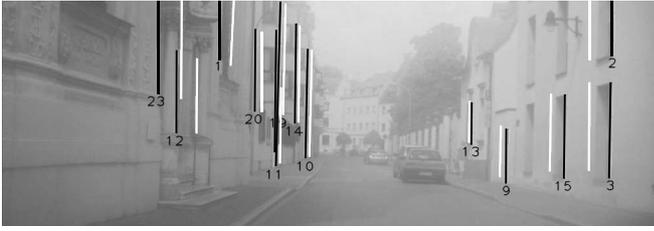
For the combination of the extracted edges out of the set $\mathbb{K}_{extract}^t$ and tracked edges out of the set $\mathbb{K}^{(t-1) \rightarrow t}$ also the tube method is used. The tolerance range is set around the tracked ones and analyzed in a different way, see Fig. 5 at time step t . Due to the fact that the extracted edge $\kappa_{extract}$ is more trusted than the tracked one $\kappa^{(t-1) \rightarrow t}$ it is used as basis and enlarged by the tracked edge. This combination method is used because the edge coordinates from the image are more reliable than the tracked points and their regression line. Furthermore, the tracking of points added during this step is easier because they are situated exactly on the edge.

F. Tracking Results

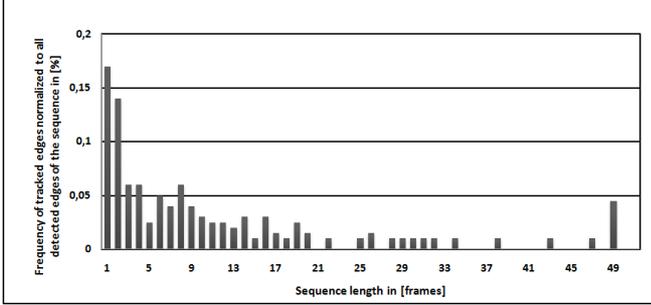
Fig. 6 a) shows the extracted edges of the current time step t related to their correspondences of a previous one. In diagram 6 b) the normalized frequency of tracked edges is visualized in conjunction to a single frame. Only few edges are not tracked especially why a subset of them already moved out of the visible area. In summary more than fifty percent of all recognized edges were tracked over at least 7 consecutive frames. The tube principle proved to be a robust tracking procedure. It was shown that the multi-purpose tube principle can be used to cope with gapped edges when analyzing single frames as well as the combination of edges when looking at multiple sequenced frames.

III. 3D RECONSTRUCTION

The robustness of tracking is a prerequisite step towards 3D reconstruction. The identification of static objects like buildings using prominent features is essential for the environmental capturing. In the following the term *edge* is transferred to *line* because of changing to geometrical considerations. The 3D reconstruction considers the associated lines of image sequences and ignores the generated feature points of the tracking process. The camera origin and the extracted edge of one image spans a plane, which can be obviously used for this step. But it has to be mentioned that the 3D reconstruction is very sensitive to roll movements of the car. Therefore the intersection of two defined planes, relating to different images of the sequence, generates a sloped line, which is difficult to handle. For more robustness, the crucial roll angle can be additionally determined by a gyro or an inertial sensor cluster, which are available in a car, see [8].



a)



b)

Fig. 6. Results of edge tracking a) Image with extracted edges (black) plus their identifier of time t and associated edges (white) of time step before; b) Frequency of tracked edges

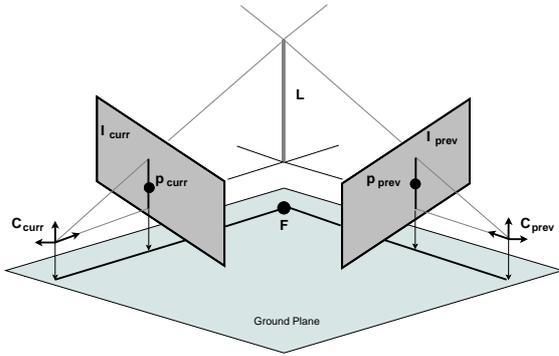


Fig. 7. 3D reconstruction by using baselines

Accordingly, we reduce the defined plane to a line which is projected to the ground.

A. Intersection of Baselines

The projection of the camera origin C and the vertical edge L of an image I generates base points on the ground, compare Fig. 7. The slight angle deviation of a vertical edge is minimized by calculating the central point p . The defined baselines of different images intersect at the base point F of the 3D reconstructed line. This method is preferred due to performance reasons.

B. Uncertainty of 3D Reconstruction

The accuracy of 3D reconstruction depends on the tracking accuracy and the camera distance of two positions from which to reconstruct. The bigger the position changes of the camera the better is the 3D reconstruction of the edges. That means also that features with a long distance to the camera are subjected to bigger errors. Fig. 8 shows the

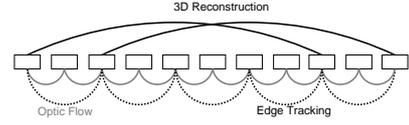


Fig. 8. Cascaded 3D reconstruction process of an image sequence

cascaded 3D reconstruction process of an image sequence. Skipping images reflects the required camera motion. The previous explanation regards to only one reconstruction step, but the features are tracked over a longer period. With the combination of the iterative results, the quality of the 3D reconstructions improves. Therefore, the generated lines in 3D space are considered as noisy measurements. We suppose a normal distribution of the measurements and use probabilistic basics to handle them. The expected value of one reconstructed base point L is defined by the average over the time period t from t_{begin} till t_{end} .

$$E[L] = \frac{1}{n} \sum_{t=t_{begin}}^{t_{end}} L_t \quad (19)$$

The variance is described by equation 20, which supposes a continuously process.

$$\sigma = \sqrt{\frac{1}{n} \sum_{t=t_{begin}}^{t_{end}} (L_t^2) - E[L]^2} \quad (20)$$

The comparison of the reconstructed results and the map material requires a description of the uncertainty relating to the ground. So the variance is transferred to the covariance in the x-y-dimension.

$$Cov_{xy} = \frac{1}{n} \left(\sum_{t=t_{begin}}^{t_{end}} L_x \cdot L_y \right) - E[L_x] \cdot E[L_y] \quad (21)$$

The uncertainty region of a reconstructed base point is elliptical with the major axis along the line-of-sight. The line approximation of the ellipsis and their intersection calculation relating to different point of views generates an extensive feature estimation on ground in a simple way.

IV. MAP COMPARISON

With a closer look onto an outdoor scene, vertical edges refer basically to buildings, their windows and doors. The precise map material, which is used for evaluation of the 3D reconstructions includes only footprints of the buildings and no texture data. In contrast the usage of feature points would produce an expanded data volume and their adaption to building information is more difficultly.

A. Evaluation of 3D Reconstructions

The evaluation step of the 3D reconstruction requires a precise knowledge of the vehicle position and its orientation. Whereas the car pose is given in a simulation environment, it has to be extensively determined in a real urban area by reference sensor technology. Fig. 9 shows the topview of the 3D reconstructions marked as blue crosses generated from

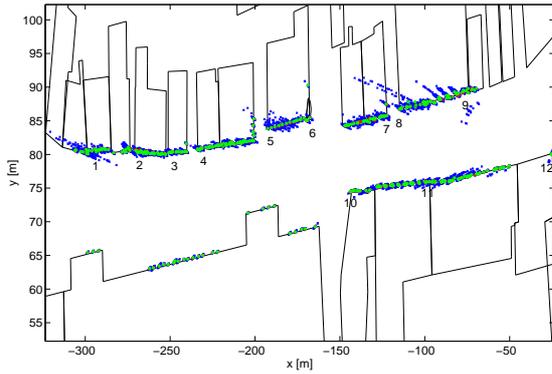


Fig. 9. Evaluation of 3D reconstruction by map comparison

TABLE I
DEVIATION OF THE 3D RECONSTRUCTIONS

Number of Baselines	Deviation of 3D Reconstruction in [m]	Average of all Deviations in [m]
1	0,2506	0,2542
2	0,1973	
3	0,1296	
4	0,2132	
5	0,0872	
6	0,1608	
7	0,1270	
8	0,0641	
9	1,1578	
10	0,1504	
11	0,2068	
12	0,3062	

different point of views. Therefore, the camera is moved along the street as if it integrated in a car. It has to be mentioned, that the exact analysis of the results is only possible due to simulated data. The averaged reconstructions (green), using the described approach above, are located near the map outlines. In Table I the deviation values of the 3D reconstructions relating to the associated outline are shown. Both the figure and the table confirm the performance and accuracy of our approach.

B. Prognosis - Evaluation of Pose Hypothesis

During a defined time period, the reconstruction process generates a local map of the line-based features in relation to the camera position. In reverse the direct comparison of the local and global map can be used for evaluation of pose hypotheses. Therefore, a quality factor, which describes the coverage of both maps has to be defined. We adapt our evaluation process, which is defined in [10], and transfer the distance calculation from image domain onto ground plane. Thus, the line of sights define the orientation of the search area. The distance values l between 3D reconstructed lines and the baselines of the global map form the quality factor W .

$$W = \frac{\left(\frac{n_{gef}^2}{n_{ges}}\right)^3}{\sum_{j=0}^{n_{gef}} l_j} \quad (22)$$

Whereas, the expression n_{ges} stands for all reconstruction points, n_{gef} describes the amount of reconstructed base points, where a corresponding outline of a building was found.

The usage of the determined quality factor in the image domain can badly handle wrong edge correspondences. But this drawback is hopefully reduced with the transformation onto ground plane and leads to a further improvement of vision-based localization extending e.g. [3] or [9].

V. CONCLUSION

In this paper the potential of image edges as a tracked and 3D reconstructed feature has been shown. The extraction of these elements extends the description of the vehicle environment by camera. This approach as applied above identifies the difficulties of each process step and delivers a balanced solution of performance, robustness and efficiency. We are able to generate local maps of reconstructed lines, which are automatically evaluated by global map material. In reverse this gives the opportunity of analyzing different pose hypotheses of the vehicle. Further work will handle the introduced two-dimensional map matching process in a deep way with the potential of lateral and longitudinal position improvement.

REFERENCES

- [1] C. Baillard, C. Schmid, A. Zisserman, A. Fitzgibbon, and O. O. England. Automatic line matching and 3d reconstruction of buildings from multiple views. 1999.
- [2] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm. Technical report, 2002.
- [3] Andrew J. Davison and David W. Murray. Mobile robot localisation using active vision. In *Proceedings of the 5th European Conference on Computer Vision*, pages 809–825, 1998.
- [4] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Readings in computer vision: issues, problems, principles, and paradigms*, pages 726–740, 1987.
- [5] C. Harris and C. Stennett. RAPID - a video rate object tracker. In *Proceedings of the first British Machine Vision Conference (BMVC90)*, 1990.
- [6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.
- [7] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, May 1991.
- [8] J. Ryu and J. C. Gerdes. Estimation of vehicle roll and road bank angle. In *Proc. Am. Contr. Conf.*, pages 2110–2115, Boston, 2004.
- [9] K. Schönherr, B. Giesler, and A. Knoll. Adaption of robotic approaches for vehicle localization. In *Proceedings of 7th International Workshop on Intelligent Transportation (WIT2010)*, Hamburg, 2010.
- [10] K. Schönherr, B. Giesler, and A. Knoll. Evaluation of pose hypotheses by image feature extraction for vehicle localization. In *Proceedings of 10th International Conference on Artificial Intelligence and Soft Computing (ICAISC2010)*, Zakopane, 2010.
- [11] Y.X. Tian, M. Gerke, G. Vosselman, and Q. Zhu. Automatic edge matching across an image sequence based on reliable points. In *ISPRS08*, pages 657–662, 2008.
- [12] P. Tissainayagam and D. Suter. Assessing the performance of corner detectors for point feature tracking applications. *Image and Vision Computing*, 22:663–679, 2004.
- [13] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.