3D Model-Based Dynamic Feature Planning for Laser Scanner-Based Navigation of Vehicles

Roland Stahn[1,2] and Andreas Stopp[1]

[1] DaimlerChrysler AG, Group Research, Assistance and Safety Systems, Berlin, Germany
[2] Technical University of Berlin, Computer Vision and Remote Sensing Group

***Abstract*—In this paper we present an approach to dynamic model-based feature prediction and feature planning for our laser scanner-based navigation system. The system enables autonomous precise maneuvering of vehicles relative to target objects based on 2D laser range data. An adaptive tracking algorithm uses predicted features to localize the target. Jump edges, line segments, line intersections and free space areas are considered. Feature prediction is based on an attributed polygonal 3D object model representing object surfaces, masking volumes and free space volumes. The model is intersected by a virtual scan plane to dynamically determine all detectable features for the current sensor pose. In order to further increase range, robustness and precision of navigation, feature planning is performed prior to or dynamically during the vehicle's approach towards the target. A rating function allows to compare feature sets with regard to visibility and quality. This enables active localization, i.e. controlling and optimizing the pose of an actuated sensor, which is especially useful for long range approaches and compensation of ground unevenness. The system has been evaluated and in part it has already been tested successfully with a *Mercedes-Benz* truck on an outdoor test yard under varying environmental conditions.**

I. Introduction

Precise navigation relative to single or multiple target objects is a basic task during vehicle operation, e.g. while docking at ramps, maneuvering into parking positions, coupling trailers or underriding swap bodies. In [1] we have introduced a general approach of laser scanner-based navigation that aims at automating these procedures. It is based on 2D laser range scanners, which provide an unmatched combination of measurement precision, fast data acquisition, robustness to varying environmental conditions and reasonable hardware costs. Fig. 1 shows our test vehicle (*Mercedes-Benz Actros*) with a rear, fix mounted *SICK LMS200* laser scanner and 180° field of view. We successfully demonstrated the capabilities of our system by implementing an assistance system for swap body interchange [1]. It works very reliable and with impressive precision for distances up to 30 m, speeds up to 1 m/s and approach angles up to 100°. In [2] we further investigated object-related navigation of vehicles with strong non-holonomic constraints, e.g. truck-trailer combinations. Our motion planning algorithms generate feasible trajectories for both, maneuvering within limited space and long distance approaches. In this paper we describe precise and robust localization from even longer range and on uneven ground.

The performance of any object tracking algorithm mainly depends on the set of visible features. Current laser scanners acquire range data within one or multiple scan planes. Object features are extracted from each individual scan. The alignment of the scan plane of a fix-mounted sensor is usually not stationary. It depends on the mounting place of the sensor as well as on the vehicle's motion on the ground along the planned trajectory. Features might be too small, get occluded or leave the sensor's field of view during the approach. On the contrary, an *actuated sensor* allows to control the sensor orientation independently of the vehicle's motions. This way the alignment of the sensor scan plane and thus the set of visible object features can be optimized by *feature planning*. Sensor actuation can be performed by a dedicated actuator (e.g. tilt actuator) or even by utilizing the level control system of vehicles. Feature planning can be implemented by rating, comparing and optimizing sets of visible features for different possible sensor orientations. This requires *feature prediction* for any 6-DOF sensor pose relative to the target object. Our approach is based on a 3D object model which is intersected by a virtual sensor plane in order to extract potential features.

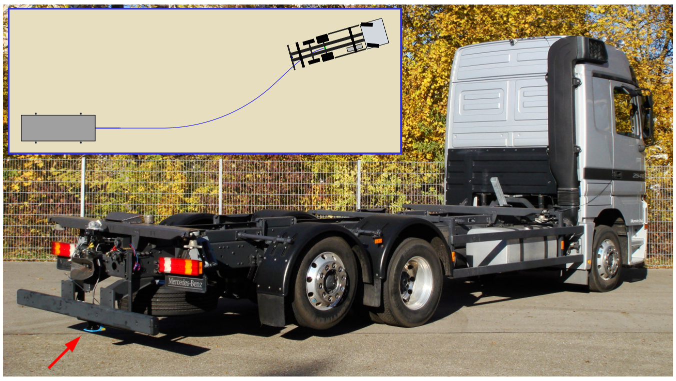


Fig. 1: Test vehicle with rear 2D laser range scanner approaching a target object on a planned trajectory.

Feature planning for visual servoing tasks is an open research field [3]. This applies to (indoor) industrial robotics and even more to outdoor vehicle navigation with much more complex environmental conditions. [4] and [5] present feature planning approaches for camera-based indoor visual servoing. In [6] 3D model based feature prediction is performed for a 3D laser range sensor, but only depth values are predicted and the relocalization algorithm is mainly based on lines extracted from an additional video camera. In this paper we apply feature planning to outdoor laser scanner-based visual servoing. Our approach utilizes results from localization, object modelling and robotic motion planning methods as described in the following sections.

II. System architecture

The basic idea of the laser scanner-based navigation system is to guide a vehicle relative to a target object using laser range data. The navigation process starts with scene analysis, target selection and motion planning followed by online feature planning, object tracking and motion control. Fig. 2 shows the architecture of the navigation system. The overall architecture has been described in [1] in detail. Newly added is the *feature planning* module. Input values are the type of the target object detected by the *scene analysis*, the trajectory of the vehicle calculated by the *motion planning* module and an estimation of the last known or the predicted current vehicle pose provided by the *object tracking* module. Output values are the predicted set of features to be tracked by the object tracking and an optimized pose for the actuated sensor, if present.

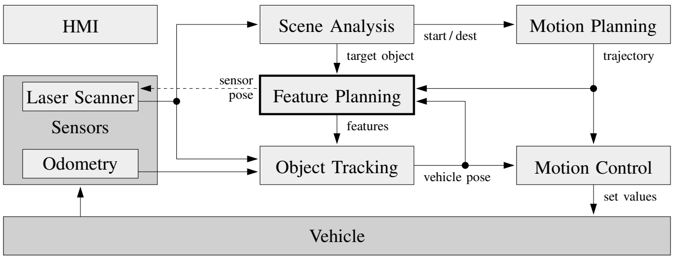


Fig. 2: Laser scanner-based navigation system architecture.

Fig. 3 depicts the inner architecture of the feature planning module subdivided into *Feature Prediction* and *Sensor Pose Optimization*, both of which consist of two submodules respectively. The *3D object model* provides an abstract representation of the current target object. Given a specific sensor pose, the *feature extraction* calculates the intersection between the 2D sensor plane and the 3D object model and extracts all relevant features (see section IV). Sensor pose optimization starts with the evaluation of these features by the *feature rating* module (see section VI). The resulting rated feature vector is passed to the *sensor pose planning* module, which optimizes the set of visible features by adjusting the sensor orientation based on the current vehicle pose and its pre-planned trajectory.

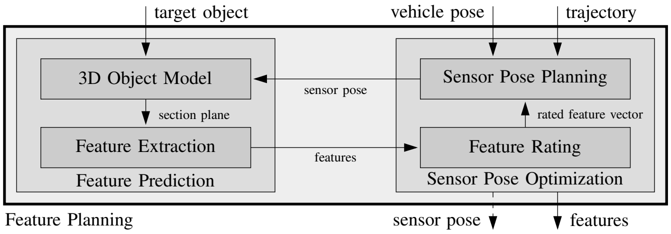


Fig. 3: Architecture of the feature planning module.

The system has been designed to support different use cases. It can be used for *online feature prediction*, i.e. during the approach the set of detectable features is continuously predicted from the current sensor pose. These features are used for adaptive object tracking (see section V). Performing *offline feature planning* determines optimized features for a complete approach in advance. It uses the pre-planned motion trajectory to determine the most favorable sensor orientation or to plan points along the trajectory to change the alignment of the sensor plane. Finally the system enables *dynamic feature planning*, where the sensor orientation is actively re-adjusted to optimize the set of visible features.

III. Features

We first examine the problem, which features are actually relevant to identifying and relocating objects within 2D laser scans. Basic features extractable from range images are discussed in [7]. This includes *jump edges*, *lines* and *line intersections*, all of which have been used (separately) in SLAM applications [8] indoor and outdoor. Furthermore the concept of *free space* is known from various collision avoidance systems, e.g. [9]. For long range outdoor relocalization usually only few features of an object are visible. Therefore in our approach *all* of above features are taken into account. Following we briefly summarize their individual properties.

A. Jump Edges

An elemental feature within laser scans are discontinuities of the depth measurements. These are commonly called *jump edges* and typically mark the borders of object faces. They are visible to the sensor as long as the originating faces are detected. A single jump edge is described by the tuple $\mathsf{J} = (x, y, depth)$ composed of its position (x, y) and its minimum $depth$. It does not yield any information on the object's orientation – but one can use pairs of jump edges for that [1]. To extract jump edges from a laser scan the difference of adjacent depth readings is compared to a threshold.

B. Line Segments

Another basic feature are *line segments*. They result from the pointwise scanning of planar surfaces (e.g. walls) by the laser sensor [10]. A line segment is characterized by the tuple $\mathsf{L} = (x_s, y_s, x_e, y_e)$ describing it's starting and end point within the 2D scan plane. From these coordinates the orientation ϕ_L and length l_L of the line segment can be calculated. Various algorithms have been developed for line segment extraction – see [11] for a comparison. We use an enhanced variant of the PSA algorithm [12] that detects lines with at least 4 scan points. It allows small gaps to be more robust regarding singular measurement errors.

C. Line Intersections

A rather complex feature in 2D laser scans are *intersections* of line segments [13]. They result from spatial intersections of object faces. A line intersection is represented by the tuple $\mathsf{S} = (x_s, y_s, \phi_s, l_A, l_B)$. It describes the intersection location

(x_s, y_s), the intersection angle ϕ_s and the distances l_A and l_B to the intersecting line segments. If both l_A and l_B are zero, we call that intersection *real*. Otherwise it is called *virtual*. Feature extraction is done by intersecting all possible pairs of extracted line segments. Setting thresholds for ϕ_s, l_A and l_B limits the number of results.

D. Free Space

Any area scanned by the laser sensor without detecting any object is called *free space*. It is described by a set of vertices forming a closed polygon $\mathsf{F} = \{(x_0, y_0), (x_1, y_1), \dots\}$. By definition there must not be any scan data within the polygon area. Free space is mainly used by obstacle avoidance applications [9]. Nevertheless it is a very useful additional feature for object recognition - especially when tracking objects with a sparse set of features only [1]. Free space requirements significantly reduce the number of candidates during object classification. It does not improve the estimation of the object pose though - therefore we call it an *indirect* feature.

IV. Feature Prediction

Our approach to feature prediction is to have an abstract representation of both target object and laser scanner. Given a 6-DOF pose for both sensor and target we predict all types of features discussed in the previous section by simulating the sensor measuring the target.

A. 3D Object Model

The model of the target object needs to be an efficient and flexible representation of the target's appearance. Model-based object representations are known from computer graphics and CAD [14]. For our application we use a 3D object model consisting of closed polygons. All polygons are directed, differentiating between front and back side. The direction is specified by the order of the vertices (mathematically positive). Three polygon categories are distinguished within the model: *Object polygons* approximate the physical surface of the target object. They can be additionally attributed by color, reflectivity and texture. *Free space polygons* define free space volumes - that are regions of the object space that are definitely free of any object parts. Free space polygons are transparent, i.e. object polygons located behind them are still visible. Finally, *mask polygons* enclose areas of the target object that should be explicitly ignored for feature prediction and object relocation. Any feature located within such a masking volume is not matched. This allows to define object models for whole classes of objects by hiding individual variations within masking volumes. Fig. 4 shows an example for a target object and its appropriate object model. The object is a box swap body. An European Norm [15] defines the location and size of the box, the guide rails and the support legs. These parts are modelled using object polygons (colored blue in Fig. 4). Additionally, non-standardized parts are masked-out with a masking volume (red) and a free space volume (green) is added in-between the support legs, where the vehicle docks in. The resulting model matches any type C745 box swap body.

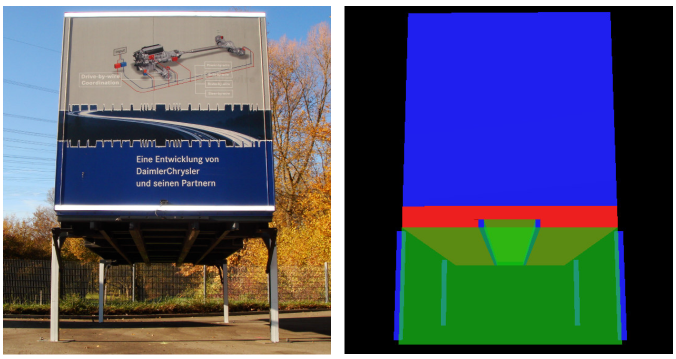


Fig. 4: A box swap body and its 3D object model with object (blue), masking (red) and free space (green) polygons.

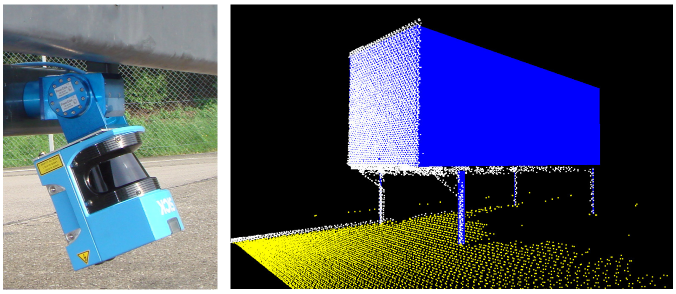


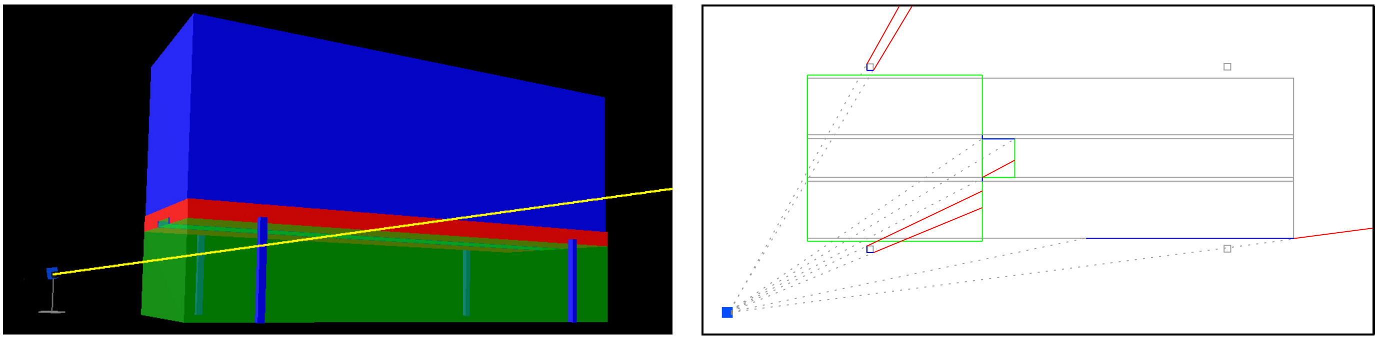
Fig. 5: Actuated sensor and 3D scan of a box swap body acquired by it. Scan data classified as ground is shown yellow.

The type of the target object is determined during scene analysis (usually 2D [1]) and target selection. In this paper we assume the availability of a corresponding 3D object model. Due to its polygonal structure it may be easily imported from existing CAD data. Masking and free space volumes usually have to be added manually though. Alternatively object and free space polygons may be reconstructed or even learned from 3D laser scan data [16]. This can be done during a 3D scene analysis while the vehicle is stopped. Fig. 5 shows an example recorded using a tilting scanner mounted at the vehicle's rear.

B. Feature Extraction

The system is able to extract all feature types described in section III from the 3D object model for any arbitrary 6-DOF sensor pose. This is much more generalized compared to model-based indoor localization algorithms proposed in [13] and [17] that use a floor parallel sensor plane and extract line segments only. Our algorithm performs two major steps: In a first step the 3D object model is intersected with the 2D sensor plane and all (theoretically) *visible* features are computed. In a second step we apply a sensor model to determine all *detectable* features. This depends on both depth and angular resolution of the laser scanner as well as on the distance of the sensor to the target and the sensor's field of view.

In order to determine all visible features we assume an ideal laser scanner with a 360° field of view and an unlimited measurement resolution. First of all the object model is transformed into the sensor's local coordinate system and all



(a) Calculation of visible features by intersecting the 3D object model with the 2D scan plane (yellow).

(b) Extracted features: jump edges (red), object line segments (blue) and free space (green). Intersections are not shown for better clarity.

Fig. 6: Model based feature prediction. One can easily recognize line segments from the support legs, the box and free space.

non-transparent polygons directed away from the sensor are removed (*back-face culling*). Then all polygons that intersect the sensor plane are determined by checking each polygon edge. For each intersecting polygon the visible line segments within the scanner plane are extracted. At the end points of each line segment potential jump edges and their minimal depths are determined. Jump edges have either a known depth (in case of a transition between two model polygons) or an unknown depth (in case of an object-background transition). Finally line intersections are computed for all pairs of line segments. The resulting feature vector $\mathbf{f} \in \mathbb{F}$ describes all visible features and their spatial relation.

Fig. 6 shows an example of the feature extraction. The sensor is located near the front of the target object with the sensor plane tilted upwards (see Fig. 6a). In Fig. 6b the extracted features (except for line intersections) are shown. One can easily identify the line segments and jump edges originating from the object's support legs and the box superstructure.

V. Adaptive Object Tracking

During an approach the visibility of object features changes significantly depending on the vehicle's trajectory and the sensor's field of view. In [1] we proposed a multi-phase object tracking algorithm to handle different sets of features. The addition of feature prediction to our system enables to further generalize this approach. Starting with a pose estimation of the vehicle by a Kalman filter the set of currently visible features is predicted using the 3D object model. This feature set is then used to localize the target object within the current laser scan (matching) and to calculate the current vehicle pose. If the measured pose differs significantly from the originally estimated pose, the procedure of feature prediction and matching is repeated iteratively in order to optimize the feature set used. Finally the Kalman filter is updated. With this *adaptive* object tracking algorithm the target object is tracked using an optimized set of features all the time.

The set of visible features depends on the alignment of the scan plane too. This can be problematic if the alignment is not known, e.g. when moving on uneven ground. Using a multilayer sensor, it is possible to compensate for pitch motions of the vehicle. Assuming three scan planes object tracking is normally performed using the middle plane. If tracked features are found within the lower or upper plane instead, the change of the sensor tilt angle can be estimated. This is either used to update the feature prediction or to re-adjust the tilt angle of the sensor actuator, if installed.

VI. Feature Planning

The basic idea of feature planning is to optimize the alignment of the sensor plane for a given sensor position. Using feature prediction we can estimate the detectable features for any arbitrary sensor pose. First of all we define a feature rating function in order to be able to compare the features of different sensor plane alignments to each other. This function then can be used for the sensor pose planning algorithm.

A. *Feature Rating*

The feature rating function $\mathsf{r} : \mathbb{F} \mapsto \mathbb{R}^+$ maps a feature vector $\mathbf{f} \in \mathbb{F}$ to a numerical non-negative rating value $\mathbf{g} \in \mathbb{R}^+$:

$$\mathbf{g} = \mathsf{r}(\mathbf{f}) \tag{1}$$

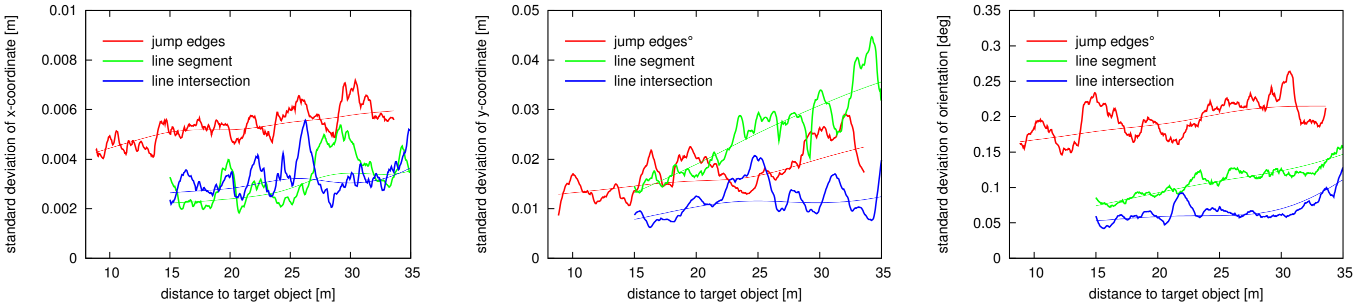
$\mathbf{g}$ defines a relative rating with no upper limit. The higher the score the better the examined set of features. In [18] local image features are characterized and [4] defines a set feature fitness measures for camera-based visual servoing. In the following we describe a rating system that is applicable to all laser scanner-based features types described in section III. We define the rating function $\mathsf{r}(\mathbf{f})$ to be a weighted sum of the feature *visibility* $\mathsf{v}(\mathbf{f})$ and the feature *quality* $\mathsf{q}(\mathbf{f})$:

$$\mathsf{r}(\mathbf{f}) = w_v \cdot \mathsf{v}(\mathbf{f}) + w_q \cdot \mathsf{q}(\mathbf{f}) \tag{2}$$

$\mathsf{v}(\mathbf{f})$ describes how well a feature is seen by the laser sensor. It is further composed of rating functions for the *detectability* $\mathsf{t}(\mathbf{f})$ and the *stability* $\mathsf{s}(\mathbf{f})$ of a feature vector $\mathbf{f}$:

$$\mathsf{v}(\mathbf{f}) = w_t \cdot \mathsf{t}(\mathbf{f}) + w_s \cdot \mathsf{s}(\mathbf{f}) \tag{3}$$

The detectability indicates how well the features can be extracted from laser scan data. This mainly depends on the angular resolution of the sensor and its distance to the target object. For outdoor vehicle navigation this is very important



(a) Standard deviation of x-coordinate. (b) Standard deviation of y-coordinate. (c) Standard deviation of orientation.

Fig. 7: Comparison of object relocation using different object features extracted from real scan data. The thick curves show the mean value of measurements from four independent approaches. Thin lines show a smooth spline interpolation.

since object tracking often has to be performed for ranges above 30 m. The stability rating $\mathsf{s}(\mathbf{f})$ estimates how robust a feature vector is to variations of the assumed sensor pose due to ground unevenness or relocation errors.

The quality rating function $\mathsf{q}(\mathbf{f})$ describes the usefulness of a feature or feature vector for estimating the vehicle pose. It is further differentiated into *distinctiveness* $\mathsf{d}(\mathbf{f})$, *position accuracy* $\mathsf{p}(\mathbf{f})$ and *orientation accuracy* $\mathsf{o}(\mathbf{f})$:

$$\mathsf{q}(\mathbf{f}) = w_d \cdot \mathsf{d}(\mathbf{f}) + w_p \cdot \mathsf{p}(\mathbf{f}) + w_o \cdot \mathsf{o}(\mathbf{f}) \tag{4}$$

$\mathsf{d}(\mathbf{f})$ estimates how well the target object is identifiable using the predicted feature vector. For example a single jump edge has a low distinctiveness, two long lines with a free space area in-between on the other hand have a high distinctiveness. $\mathsf{p}(\mathbf{f})$ and $\mathsf{o}(\mathbf{f})$ measure the contribution of each feature to a precise relocalization (see section VII for experimental results).

The evaluation of the rating functions inevitably depends on sensor properties and the respective feature type. A detailed description of the rating functions for each feature type will be part of a future publication. The choice of the weighting coefficients w_i depends on the requirements of the respective application. For example, precise docking demands high feature quality while fast driving emphasizes feature stability. The coefficients may also be dynamically adjusted. For example, object docking might require good visibility while far away from the target and high quality close to it.

B. Sensor Pose Planning

Based on the rating function for comparing different sensor poses, we are able to define a cost function. Such a function enables the application of well known planning algorithms [19] to the feature planning problem. We utilize our multi-dimensional, grid-based planning system which is used for vehicle motion planning as described in [2]. The dimension of planning space depends on the degrees of freedom of the sensor or actuator. The simplest case is to plan along the vehicle's motion trajectory with a single vertical joint actuator. This one-dimensional planning can be performed very fast. The planning cost function needs to optimize the sensor pose by maximizing the feature rating while minimizing sensor actuation since sensor movement requires time too. This is done by adding costs for every sensor motion.

VII. Results

In order to further examine the properties of the features introduced in section III, we have performed a series of experiments using a box type swap body (see Fig. 4) and a fixed mounted single plane *SICK LMS200* laser scanner. We set up three different configurations of our generic tracking algorithm – each based on one basic feature type. The first set-up tracks the pattern of four jump edges originating from the front pair of support legs of the swap body. This is the same set of features used in the first phase of the tracking algorithm described in [1]. Each leg has a width of 10 cm and is visible up to 30 m at max. The second set-up tracks the front face of the swap body box by extracting the corresponding line segment from the laser scan. The box has a width of 2.60 m and is visible for up to 80 m, if the scan plane is aligned favorably. The third set-up calculates the intersection of the line segments originating from the front and one side of the swap trailer box. For the latter two configurations the sensor has been tilted upwards by 4.5 °. Additional free space definitions have been used for all three set-ups to identify the target object unambiguously.

Fig. 7 shows a comparison of the data obtained for each feature set for the range from 8 m to 35 m. We measured the pose (x, y, ϕ) of the target object within the right-handed sensor coordinate system with the x-axis pointing along the longitudinal axis of the vehicle. Since no reference measurement system has been available we cannot compare the pose measurements of the three set-ups directly. Instead we use the standard deviation of the pose estimation for a relative comparison. It gives an indication of the stability and quality of this estimation. For each coordinate of the object pose and each feature set the standard deviation has been calculated from unfiltered data of four independent approaches using a moving mean with a temporal window size of ±1 s. We have used odometry data to compensate for the vehicle ego-motion within this window. Note that each diagram uses different scales since we want to compare the features to each other.

Furthermore, some peaks in this measurements are systematic errors caused by uneven ground and changing reflectivity of the target surface.

The results show in general that the deviation in x is much better than that in y. That is typical since current laser scanners have a better depth resolution compared to their angular resolution. Regarding the x-component, line-based features perform slightly better than jump edges (Fig. 7a), because they use more scan points averaging sensor noise. This is much more significant for the estimation of the orientation ϕ though (Fig. 7c). The y-measurements of line segments and jump edges are limited by the angular resolution of the scanner – their y-deviation increases with the distance (Fig. 7b). Line intersections provide nearly constant y-deviation since their location mainly depends on the ϕ-estimation of the intersecting lines. In [1] we have determined the absolute precision of the jump edge-based pose estimation to be $\pm 1\,\mathrm{cm}$ and $\pm 0.2\,^{\circ}$ during dock-in of the vehicle into the target swap body (measured manually). The results in Fig. 7 imply that line segments and line intersections perform even better. Together with the considerations in section III this leads to the following appraisal: Jump edges are easily identified and found on nearly every object. They are the basic feature for object relocation. Long line segments are detectable even at long ranges and provide a better ϕ-estimation. They are not always available though. Line intersections provide the best pose estimation but require two suitable line segments. Detection of small objects from long ranges is difficult due to the limited angular resolution of the laser scanner and insufficient reflectivity of the target's surface area. This can be improved using a next-generation multilayer laser scanner with variable angular resolution and extended range [20].

Regarding our application example, tracking a box swap body, the results imply that it would be best to track the box front as long as possible for high precision and stability and switch to the support legs at close range and while docking, where the box leaves the field of view of the sensor. This could be done to certain extend even without sensor actuation by using a multilayer laser scanner with a ground-parallel plane tracking the support legs and at least one more plane tilted towards the box superstructure.

VIII. Conclusion

This paper extends our outdoor laser scanner-based navigation system in two ways: First, adding dynamic feature prediction and adaptive model-based object tracking generalizes object recognition. Second, the introduction of dynamic feature planning and actuated sensor control further improves the range, precision and robustness of the system. The feature rating system enables online optimization of feature set and sensor pose using planning algorithms. A 3D attributed polygonal object model is used to determine the set of detectable features for any arbitrary 6-DOF sensor pose. The addition of free space and masking volume definitions increases the general usability of this representation significantly. The system has been designed to meet practical requirements of autonomous or semi-autonomous outdoor vehicle navigation, including varying environmental conditions and ground unevenness. The developed algorithms are applicable to known and announced future laser scanners with a single or with multiple scan planes. Future work includes further practical experiments with an actuated laser scanner as well as with a multilayer sensor on more uneven ground.

Acknowledgments

This work has been partly supported by *DaimlerChrysler Truck-Product-Engineering*.

References

[1] R. Stahn, G. Heiserich, and A. Stopp, "Laser scanner-based navigation for commercial vehicles," in *IEEE Intelligent Vehicles Symposium (IV'07)*, June 2007, pp. 969–974.

[2] R. Stahn, T. Stark, and A. Stopp, "Laser scanner-based navigation and motion planning for truck-trailer combinations," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, September 2007, accepted.

[3] E. Malis, "Survey of vision-based robot control," in *European Naval Ship Design, Captain Computer IV Forum*, April 2002.

[4] M. Paulin and H. G. Petersen, "Automatic feature planning for robust visual servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, August 2005, pp. 205–211.

[5] F. Janabi-Sharifi and M. Ficocelli, "Formulation of radiometric feasibility measures for feature selection and planning in visual servoing," in *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 34, no. 2, April 2004, pp. 978–987.

[6] M. R. Stevens and J. R. Beveridge, "Interleaving 3D model feature prediction and matching to support multi-sensor object recognition," Colorado State University, Tech. Rep. CS-96-107, December 1995.

[7] X. Jiang and H. Bunke, *Threedimensional Computer Vision - Acquirement and Analysis of Range Images*. Springer, Berlin, 1997, [german].

[8] D. J. Spero and R. A. Jarvis, "A review of robotic SLAM," Department of Electrical and Computer Systems Engineering, Monash University, Tech. Rep. MECSE-4-2007, January 2007.

[9] S. Thrun *et al.*, "Stanley: The robot that won the DARPA grand challenge," *Journal of Robotic Systems*, vol. 23, no. 9, pp. 661–692, 2006.

[10] A. Garulli, A. Giannitrapani, A. Rossi, and A. Vicino, "Simultaneous localization and map building using linear features," in *2nd European Conference on Mobile Robots (ECMR'05)*, September 2005.

[11] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 1768–1773.

[12] A. Ekman, P. Klöör, and D. Strömberg, "Incremental map-making in indoor environments," April 1993, iFAC workshop on Intelligent Autonomous Vehicles.

[13] J. Gomes-Mota, "Localisation of a mobile robot using laser scanner and reconstructed 3D models," Ph.D. dissertation, Universidade Técnica de Lisboa, Portugal, November 2001.

[14] F. Langbein, "Geometric modelling - scene and object representation," Lecture notes, 2003.

[15] "EN 284 - Swap bodies. Swap bodies of class C. Dimensions and general requirements." European Committee for Standardization (CEN), 1992.

[16] D. Hähnel, W. Burgard, and S. Thrun, "Learning compact 3D models of indoor and outdoor environments with a mobile robot," *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 15–27, 2003.

[17] A. Arsénio and M. I. Ribeiro, "Active range sensing for mobile robot localization," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, 1998, pp. 1066–1071.

[18] G. Carneiro and A. D. Jepson, "The distinctiveness, detectability, and robustness of local image features," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 296–301.

[19] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion - Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, 2005.

[20] IBEO Automobile Sensor GmbH, "ALASCA XT," 2006, datasheet.