2013 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)
June 23-26, 2013, Gold Coast, Australia

Fast and Precise Localization at Stop Intersections

Alexander Barth[1], Jan Siegemund[2], and Julian Schwehr[1]

Abstract— **This article presents a practical solution for fast and precise localization of a vehicle's position and orientation with respect to stop sign controlled intersections based on video sequences and mapped data. It consists of two steps. First, an intersection map is generated offline based on street-level imagery and GPS data, collected by a vehicle driving through an intersection from different directions. The map contains both landmarks for localization and information about stop line positions. This information is used in the second step to precisely and efficiently derive a vehicle's pose in real-time when approaching a mapped intersection. At this point, we only need coarse GPS information to be able to load the proper map data.**

The experimental results show this approach is able to successfully localize a vehicle at stop intersections with decimeter accuracy under various weather conditions and after several months between the map data collection and the test drive.

The proposed method enables a variety of functions for future driver assistance systems, such as stop line violation warning, advanced Adaptive Cruise Control/ Traffic Jam Assist functions in cities, or autonomous driving.

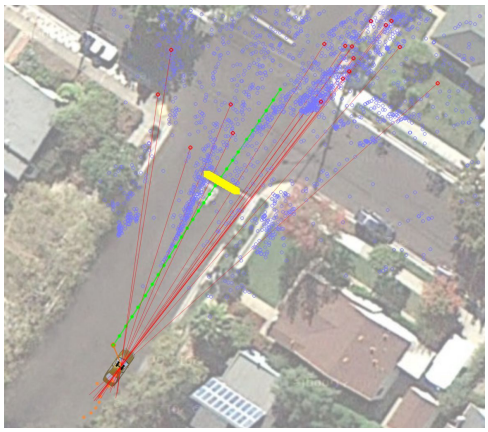


Fig. 1. Bird's Eye view of our vision-based localization method. The objective is to refine a vehicle's pose with respect to a stop line (yellow line) based on video images and mapped waypoints (red and blue circles) along a reference trajectory (green line).

I. Introduction

Future advanced driver assistance systems require information about the host vehicle's relative position and orientation, not only with respect to a given lane, as used in lane keeping or recent Traffic Jam Assist functions, but also with respect to the traffic environment and road infrastructure. Traffic intersections in particular are highly challenging environments. Precisely estimating the host vehicle's pose, for example, with respect to traffic lights, stop lines, or crosswalks enables a variety of new functions, making cars smarter in terms of obeying the traffic rules. This is especially essential with increased automation of the driving task in future cars, or in terms of augmented reality applications.

In this article we focus on stop intersections, but the proposed methods can easily extend to other traffic scenarios. The objective is to derive the host vehicle's pose, i.e. position and orientation, with respect to an intersection and in particular with respect to a given stop line the host vehicle is approaching.

Instead of analyzing the complex environment in front of a vehicle purely based on the real-time video data, e.g. by detecting stop signs in the scene to identify potential intersections, we decrease the complexity of the problem by storing static information about the intersection in a map ahead of time from a previous drive through the same intersection. This gives us an expectation of what the sensors are going to see at a certain coarse location. The real-time task reduces to matching the image content with the expectation from the map and refining a pose prior based on a set of observed *waypoints* in the scene (see Fig. 1). A Kalman filter is applied to smooth the estimation results and integrate additional information about the ego motion of the host vehicle.

Mapping an unknown 3D environment based on sensor data has been well addressed in the robotics domain in recent decades, resulting in numerous publications about the so called Simultaneous Localization and Mapping (SLAM) problem [1], [2]. In particular, graph-based SLAM approaches have been established as practicable and accurate solution even for large scale problems [3]. Methods that only use video cameras and information about the robot's motion are referred to as Visual SLAM. These methods typically aim for a consistent 3D model of the local environment over a larger area or route. For a recent survey on visual SLAM approaches in the intelligent vehicles domain, see [4]. Our work is closely related to the work published in [5], that uses a self-generated map of 3D visual landmarks for estimating all six degrees-of-freedom of the host vehicle with respect to the map. However, we further reduce the complexity of the problem by adding several constraints, reasonable for typical road scenes, to get a very fast result. Besides mapping characteristic landmarks suitable for localization, there has also been work done in the field of mapping static road objects such as traffic lights [6], [7] or road signs [8].

We will first introduce the map generation steps in Sec. II and then explain how this data is used for real-time pose esti-

[1]Mercedes-Benz Research and Development North America, Driver Assistance & Chassis Systems U.S., Palo Alto, CA, USA firstname.lastname@daimler.com

[2]University of Bonn, Department of Photogrammetry, Bonn, Germany jansiegemund@uni-bonn.de

mation in Sec. III. Experimental results and our conclusions are given in Sec. IV and Sec. V, respectively.

II. Map Generation

In our approach we decouple the estimation of the reference poses at which the video images are captured from the computation of the waypoints used for real-time localization. This significantly reduces the complexity of the problem.

A. Data Collection and Reference Trajectory Generation

In the data collection phase a vehicle equipped with a forward looking stereo camera mounted behind the windshield is driven through an intersection from all possible directions. We record images at a fixed rate of 16 Hz and assign information about the current host vehicle's ego motion state to each image, such as speed and yaw rate, as well as GPS data from the on-board navigation system (latitude, longitude, heading) if available.

Once all data has been acquired, any coarse and noisy GPS positions are adjusted by fitting the most likely cubic B-spline model to the data points for each trajectory. The pose refinement also incorporates the host vehicle's speed and yaw rate as constraints between two adjacent positions to yield consistent reference trajectories. Outliers in the GPS data are detected and removed from the estimation in an iterative procedure. Finally, the resulting continuous spline function allows for assigning each recorded image a refined *reference pose* on this reference trajectory.

B. Waypoint Computation

We select only a subset of images, denoted as *reference images*, for the waypoint computation. We keep the sequential order of the images and make sure the selected images are a minimum distance apart according to the assigned position on the reference trajectory. This reduces the amount of data to be stored in the map. For each reference image, we extract image features using a feature detector and describe the features with a descriptor. There is a variety of feature detectors and descriptors. We use a combination of the *FAST* corner detector [9] and the *BRIEF* descriptor [10] in our experiments, however, any other suitable combination might be used as well.

Next, feature matching between K consecutive reference images is performed in a sliding window manner. Features that can be matched in all K reference images are considered stable. Since the camera poses are known from the reference trajectory, one can estimate a 3D waypoint and its covariance matrix from the corresponding image points using triangulation in a maximum likelihood sense. We typically use $K = 3$ images. See Fig. 2 for an example.

For each waypoint a new map entry is generated. We only store a 2D Cartesian coordinate, i.e. we are ignoring the height coordinate and the corresponding covariance matrix of the waypoint. In addition, we maintain a link to all feature positions and descriptors of all images used to compute this point. It is therefore not only possible to query a list of waypoints for a given reference pose, but also the corresponding image features.

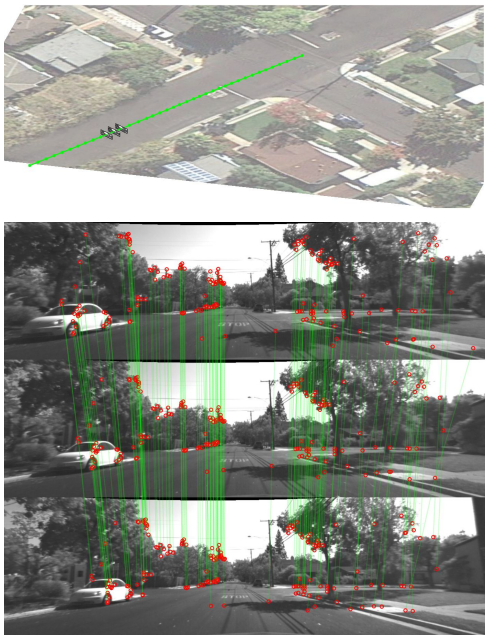


Fig. 2. Corresponding image features between neighboring images (*FAST-BRIEF*). Waypoints are computed based on $K = 3$ consecutive images.

C. Mapping of Stop Lines

We implemented a convenient semi-automated approach for mapping stop lines. During the data acquisition phase the human operator is asked to push a button on the steering wheel whenever the vehicle is stopped exactly at a stop line, i.e. with the front bumper right above the line on the ground. This internally marks the current image as the stop position.

In the offline processing step, the manually marked position generates a stop line hypothesis, given the known constant transformation from the camera to the vehicle front. An advantage of this approach is that both the stop line position and the reference images are stored in the same reference coordinate system.

To overcome deviations that human drivers have in stopping exactly at a stop line, we further refine the coarse stop line position as well as the stop line orientation based on the captured reference images. The objective is to align the projection of the stop line to strong image intensity edges in nearby reference images containing the stop line. We assume a good alignment if the projected line covers a region with a strong image gradient. A particle filter is used to sample several hypotheses of stop line poses around the initial approximate value. The particles are weighted according to the following equation:

$$w_i = I\Lambda_{\max} - \sum_{i=1}^{I} \Lambda(i) \tag{1}$$

where $\Lambda(i)$ corresponds to the entry of the Euclidean distance transform [11] of the thresholded gradient image at the i-th pixel of the projected line, $1 < i < I$, and $\Lambda_{\max}$ to the maximum of Λ. The final stop line position is then computed as the weighted average of the final particle set. An example

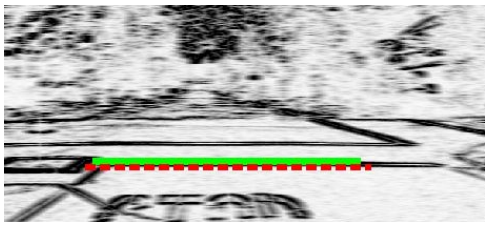


Fig. 3. Stop line refinement based on image gradient. The dashed red line represents the initial hypothesis, the solid green line is the refined result over 5 reference images.

for this refinement step is given in Fig. 3. An efficient stereo-vision based 3D representation, called Stixels [12], is used to analyze whether the area in front of the vehicle is occupied by an object, e.g. a lead vehicle, or whether the ground is visible. The refinement is skipped when the stop line is occluded in the reference images.

As an alternative approach to the manual annotation of stop line positions during the data acquisition phase, one could also initialize the stop line refinement based on detected stop signs in the image using computer vision techniques, or a road attribute map from an external source.

III. Real-Time Localization

Instead of computing a full bundle adjustment of all measurements and object poses, we reduce the problem to a planar resection of single vehicle poses based on fixed waypoint correspondences. This fast approximation still provides very accurate results in practice, given good knowledge of the coarse location from previous time steps. This section first introduces how we derive a single pose based on observed waypoints at one time step, then describes how the pose estimate is recursively refined using a Kalman filter.

A. Single Time Step Pose Estimation

Given a set of planar waypoints ${}^{\mathcal{I}}\boldsymbol{X}_n = [x_n, y_n, 0]^\mathsf{T}$, $2 < n < N$, with uncertainty ${}^{\mathcal{I}}\Sigma_{X_n X_n}$ in a 3D Cartesian reference coordinate system, denoted as *intersection system* $\mathcal{I}$, and noisy observations $[u_n, v_n]^\mathsf{T}$ of their image projection coordinates, we estimate the relative position and orientation of the host vehicle. It is defined by the 4×4 transformation matrix ${}^{\mathcal{V}}\mathsf{M}_{\mathcal{I}}$ of its local *vehicle coordinate system* $\mathcal{V}$ with respect to $\mathcal{I}$, with

$$ {}^{\mathcal{V}}\mathsf{M}_{\mathcal{I}} = \begin{bmatrix} {}^{\mathcal{V}}\boldsymbol{R}_{\mathcal{I}}(\psi) & \mathbf{0} & {}^{\mathcal{V}}\boldsymbol{T}_{\mathcal{I}} \\ \mathbf{0}^\mathsf{T} & 1 & 0 \\ \mathbf{0}^\mathsf{T} & 0 & 1 \end{bmatrix}, \tag{2} $$

i.e. we restrict the degrees of freedom of this transformation to a 2D rotation ${}^{\mathcal{V}}\boldsymbol{R}_{\mathcal{I}}$ around the height axis by angle ψ, and a 2D translation vector ${}^{\mathcal{V}}\boldsymbol{T}_{\mathcal{I}} = [T_x, T_y]^\mathsf{T}$. The intersection system and vehicle system coincide in the ground plane. All systems are right-handed with x pointing forward, y to the left, and z upwards. The camera is rigidly mounted to the host vehicle with a known six parameter transformation ${}^{\mathcal{C}}\mathsf{M}_{\mathcal{V}}$ of coordinates from the vehicle system $\mathcal{V}$ to the camera system $\mathcal{C}$ (homogeneous form).

The relationship between homogeneous image coordinates ${}^{\mathcal{C}}\mathbf{x}_n = [u_n, v_n, 1]^\mathsf{T}$ and intersection coordinates of a waypoint is given by the projection

$$ {}^{\mathcal{C}}\mathbf{x}_n = \lambda \left(\begin{bmatrix} \mathsf{K} & \mathbf{0}_{3\times 1} \end{bmatrix} {}^{\mathcal{C}}\mathsf{M}_{\mathcal{V}} \, {}^{\mathcal{V}}\mathsf{M}_{\mathcal{I}} \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix} \right), \tag{3} $$

with the 3×3 intrinsic camera calibration matrix K, and λ being the free scale parameter. For ease of readability we skip the point index n in the following equations.

Multiplying ${}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{V}}^\mathsf{T}\mathsf{K}^{-1}$ from left on both sides yields

$$ {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{V}}^\mathsf{T}\mathsf{K}^{-1} \, {}^{\mathcal{C}}\mathbf{x} = \lambda \left(\begin{bmatrix} I_3 & {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{V}}^\mathsf{T} \, {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{V}} \end{bmatrix} {}^{\mathcal{V}}\mathsf{M}_{\mathcal{I}} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \right). \tag{4} $$

With the substitutions

$$ {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{V}}^\mathsf{T}\mathsf{K}^{-1} \, {}^{\mathcal{C}}\mathbf{x} = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \tag{5} $$

$$ {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{V}}^\mathsf{T} \, {}^{\mathcal{C}}\boldsymbol{T}_{\mathcal{V}} = \boldsymbol{\Omega} \tag{6} $$

we receive

$$ \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \lambda \left(\begin{bmatrix} {}^{\mathcal{V}}\boldsymbol{R}_{\mathcal{I}} \begin{bmatrix} x \\ y \end{bmatrix} + {}^{\mathcal{V}}\boldsymbol{T}_{\mathcal{I}} + \begin{bmatrix} \Omega_x \\ \Omega_y \end{bmatrix} \\ z + \Omega_z \end{bmatrix} \right). \tag{7} $$

Since only waypoints in front of the camera are considered, $x' > 0$ holds and we may build the fraction of the first two rows

$$ \frac{y'}{x'} = \frac{\sin(\psi)x + \cos(\psi)y + T_y + \Omega_y}{\cos(\psi)x - \sin(\psi)y + T_x + \Omega_x} = g(\psi, T_x, T_y) \tag{8} $$

yielding a nonlinear function g relating the unknown pose parameters ψ, T_x, and T_y to the transformed measurements x' and y'. We solve for these unknown parameters in an iterative manner using Least-Squares optimization, considering all N measurements. Note that the unknown z-coordinate becomes obsolete by this step and thus does not have to be stored in the map.

To speed up the computation, we treat the waypoints coordinates ${}^{\mathcal{I}}\boldsymbol{X}$ as fixed throughout the optimization process. We also reduce the approximation error by considering the uncertainty of the waypoints in the stochastic model. In order to increase the robustness of the approach, an outlier test is applied before each new iteration of the estimation process, rejecting points that are not consistent with the current estimate.

B. Sequential Pose Estimation

The following steps are performed to make the above steps practicable for localization of a vehicle with respect to an intersection.

a) Initialization: In the very first step, an approximate value of the vehicle's global position is required to load the right map data and extract a list of waypoints that are assumed to be visible from the current pose. We use coarse GPS information about the host vehicle's position and heading. Alternatively, image retrieval approaches could be exploited to identify the current position purely based on similarity of the image content to reference images in the database [13]. However, using GPS significantly speeds up the process and increases the reliability of the initialization.

b) Waypoint Selection: Given an approximate value of the vehicle's location we search for the nearest reference pose in the database. Since each reference pose is linked to a list of waypoints, we obtain both the 2D positions $[x, y]$ of these points as well as a list of descriptors D of the image features that have been considered for computing the waypoint.

c) Feature Matching: We are using the same feature detector and descriptor that was used to compute waypoints to detect and describe features in the current image. The resulting image feature descriptors are then compared to the waypoint descriptors D. We require a feature to match at least $\lceil K/2 \rceil$ descriptors according to a matching cost function, if K is the number of images that were used to compute the waypoint. This process can be computed in parallel for further speed up.

d) Temporal Filtering: The single pose estimates are used as direct measurements for a Kalman filter with a linear motion model. The measurement noise results from the uncertainty of the estimated parameters. The Kalman filter not only smooths the results, it also allows for predicting the next pose based on the ego-motion and using it as the next approximate value. In our system the ego-motion is derived from the speed and yaw rate of the host vehicle. Any other method, such as visual odometry could be used instead. To be less sensitive to errors in the pose estimation, an outlier detection method based on the Mahalanobis distance between the prediction and the measurement is applied. Once initialized, the feature-based localization does not require additional GPS measurements. However, GPS is used to verify the estimation result if available. Increasing deviations from the GPS measurement can trigger a reinitialization.

The Kalman filter has the additional advantage of being highly efficient in terms of a real-time system, compared to methods that simultaneously adjust a sequence of poses, e.g. combining the information of multiple time steps into one huge equation system such as bundle adjustment.

C. Stop Line Distance Computation

Once the relative position of the host vehicle with respect to the intersection is known, it is straightforward to compute the vehicle's relative position to a given stop line, which is also defined in the intersection system. The uncertainty of this relative transformation results from the uncertainty of the filtered pose estimate and the uncertainty of the stop line position in the map. We define the stop line distance as the orthogonal distance from the host vehicle's front center position to the stop line.

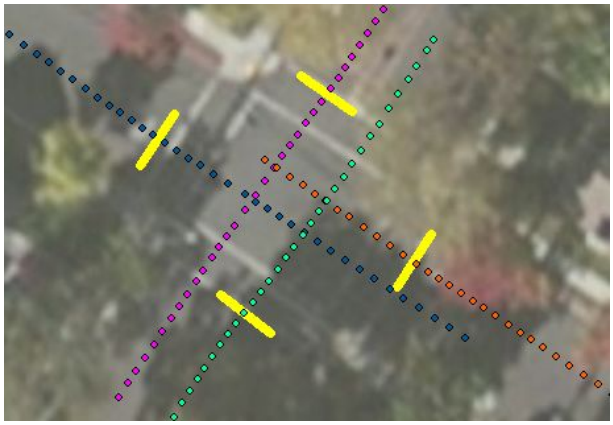


Fig. 4. Example intersection with four reference trajectories and mapped stop lines superimposed on areal imagery based on WGS84 coordinates (Source: Bing Maps).

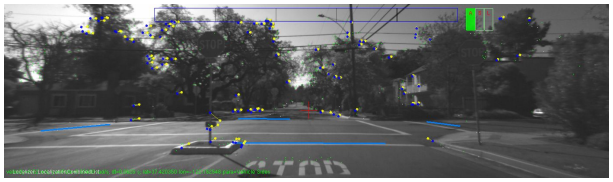
IV. Experimental Results

We have tested our system both on recorded image sequences and in live traffic. Fig. 4 shows a typical 4-Way Stop intersection at a residential area in Palo Alto, CA. Four reference trajectories (colored dots) and the mapped stop lines (yellow lines) for each intersection approach are overlaid on the image. The reference poses and stop lines, as well as a number of waypoints not shown here, were generated according to the method described in Sec. II in February 2013.

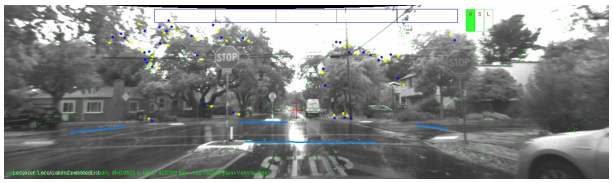
Using this intersection, we evaluated the accuracy and precision of our approach with recorded sequences from test drives where the driver was instructed to stop as close as possible to the stop line, i.e. with the vehicle front bumper above the line marking on the ground. The evaluation criteria is the estimated stop line distance at the time the recorded vehicle speed reaches zero. The stop line distance is positive if the vehicle front bumper is passed the stop line. A negative distance means the vehicle has not reached the stop line yet. We found that the standard deviation of a human driver stopping at a stop line is about 15 cm after some practice.

In the first experiment, we considered three different data sets. The first data set included 12 trajectories recorded two days after the reference trajectory was generated, while the second data set with 15 trajectories was recorded a week after. In the third data set, 12 trajectories have been considered, including samples from April, June, August, October, and December 2012, as well as February 2013. The October and December scenes were collected in heavy rain. Fig. 5 shows some example images of these sequences with the estimated stop line position as well as the matched features superimposed. In all experiments the North-South direction (Fig. 4, purple dots) was considered, which is in particular challenging since there are many trees along the street that make it more difficult to find stable waypoints. The recorded trajectories typically start 40-50 m before the stop line.

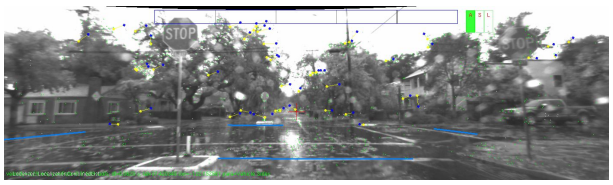
We compared our proposed feature-based localization approach to a Kalman filter approach that fuses the raw



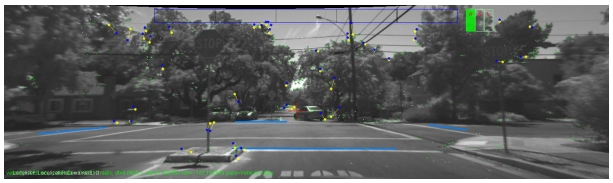
(a) February 2013



(b) December 2012



(c) October 2012



(d) August 2012

Fig. 5. Sample images from sequences in Palo Alto data set #3 with predicted stop line positions (blue lines) and matched features superimposed (blue crosses=reference image, yellow crosses=current image).

GPS measurements with the vehicle speed and yaw rate to estimate the vehicle position using the same linear motion model that is being used in our approach.

Fig. 6 shows the distribution of the resulting stop line distance estimates at the time the vehicle actually stopped. The red horizontal lines indicate the median, the upper and lower bound of the blue boxes indicate the 75th and 25th percentile respectively. It turns out that our approach outperforms the GPS-based solution in all three experiments in terms of accuracy. The median stop line distance for the first data set is -0.35 m. Given the fact that the stop line marking at this intersection is 30 cm wide, the data indicates the driver has stopped right in front of that marking most of the time. Please note that the proposed stop line refinement in the mapping phase tends to shift the stop line to the center of the line marking. The corresponding standard deviation is 0.19 m, which is within an order of magnitude of the variation of a human driver stopping at the stop line. Since the focus in the mapping phase is on the relative accuracy between the stop line position and the reference trajectory, the GPS-based approach can be more inaccurate and is strongly influenced by the current satellite configuration. This effect can be seen in the second data set. Here, the median GPS-based stop line distance is -1.1 m.

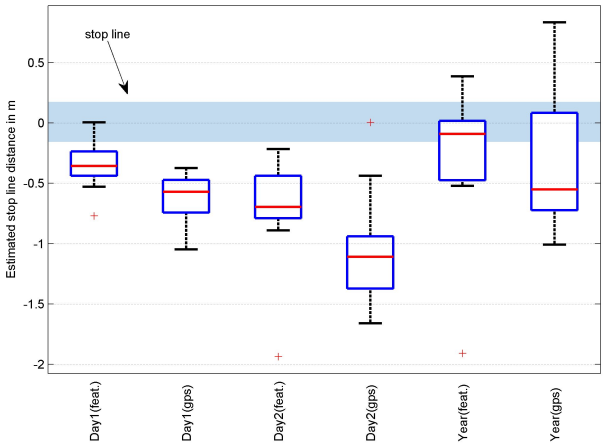


Fig. 6. Distribution of the estimated stop line distances for different data sets and localization methods at the Palo Alto intersection.

The feature-based approach gets closer to the truth, however, with a median of -0.69 m there is still a larger offset compared to the first data set. This indicates that a poor initialization, which is based on GPS, still has some effect on the result of the feature-based localization, which is only a refinement of the approximate values.

The long-term study shows the estimated distances are closer to zero for both approaches. This can be explained by the fact that these sequences include a larger variety of the actual stop positions, i.e. for some of the older sequences it is not guaranteed that the driver has stopped exactly at the same position. Thus, we expect a larger variance of the measurements. The median stop line distance for the feature-based localization on the long-term data set is less than 10 cm.

We repeated the experiment at an intersection in San Francisco, CA that does not follow the planar ground assumption. We have collected data from 25 trajectories approaching the same intersection on two different days, 15 on the first day, and 10 a week later. The results are shown in Fig. 7. The median of the estimated stop line distance is less than -1 cm for both data sets using the feature-based localization. For comparison, using the GPS-based method the median stop line distance is $+0.63$ m on the first day, and $+1.12$ m on the second day. The high accuracy of the feature-based localization can be explained by much more stable features on building facades compared to the Palo Alto data set, where most features were placed on trees. Example images from this data set are given in Fig. 8.

In total it turns out that our approach provides a more precise localization of the stop line compared to the GPS-based localization, i.e. the standard deviation is much smaller. Since we also knew the host vehicle was stopped exactly at the stop line most of the time, we can also conclude that our approach is more accurate in detecting when the vehicle has actually reached the stop line position than the GPS-based approach.

The overall computation time of the real-time localization is 21 ms per image on average, including 7 ms for feature detection and 11 ms for brute force feature matching. The

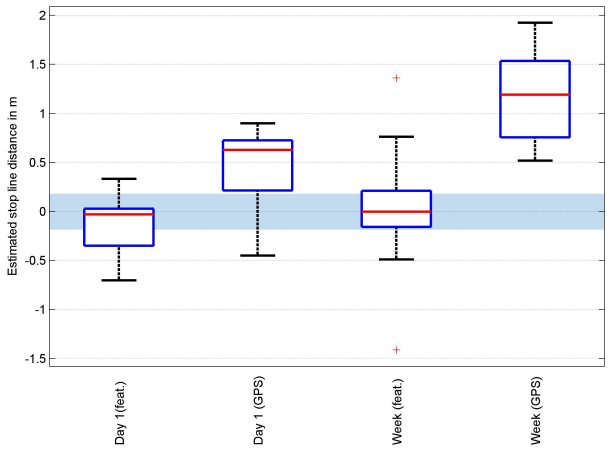


Fig. 7. Distribution of the estimated stop line distances for the San Francisco data set. The median stop line distance is less than 1 cm with the feature based localization.

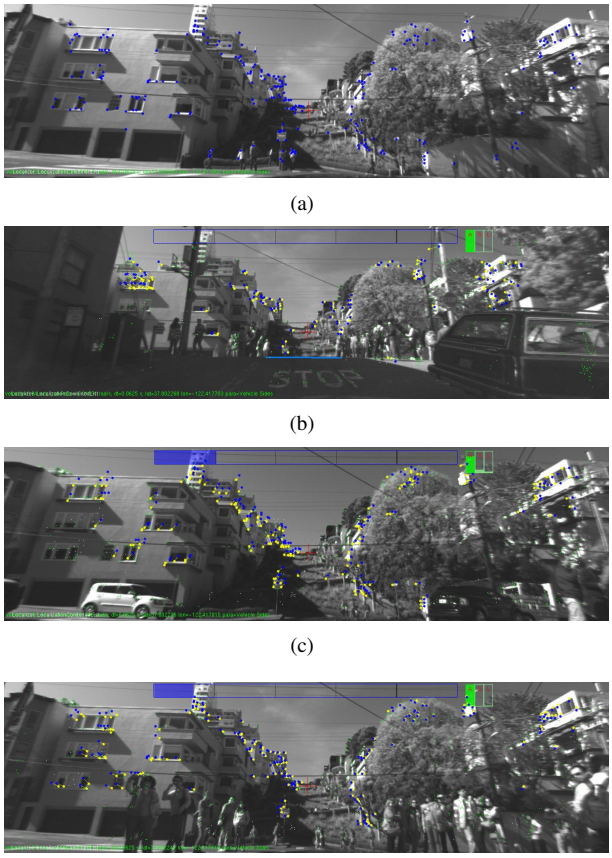
actual pose refinement takes less than 1 ms on a Quad Core PC (Intel i7-2600K). The remaining time is overhead, e.g. for loading the waypoint lists.

V. Conclusion

We have presented a real-time pose estimation approach that is both fast and precise for localizing a vehicle's pose with respect to an intersection, and in particular with respect to a stop line. The method is generic such that it could easily be applied to different traffic scenarios.

The relative localization based on image features outperforms the global GPS-based solution in terms of the stop line distance accuracy. Relative accuracy is much more essential than global accuracy in many applications, e.g., if the vehicle should stop automatically at a stop line.

Furthermore, due to the significant simplification of the pose estimation to a 2D problem, we both increase the robustness of the approach in real-world traffic scenes and significantly speed the process. The low computational costs of our approach is one key advantage compared to other real-time capable approaches that estimate more degrees of freedom of the vehicle's pose. The real-time pose refinement can be computed in less than 1 ms. The remaining processing time is mainly dominated by the feature detection and matching. These steps could be further optimized, e.g., by exploiting parallel processing capabilities.



(a)

(b)

(c)

(d)

Fig. 8. (a) Examples from San Francisco data set reference trajectory, containing only the mapped way points. (b)-(d) Sample images from test runs with matched features superimposed. Features on the buildings provide very stable waypoints, while features on pedestrians or cars are changing quickly.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i the essential algorithms," *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, vol. 2, p. 2006, 2006.

[2] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intell. Transport. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, 2010.

[3] S. Thrun and M. Montemerlo, "The graph slam algorithm with applications to large-scale mapping of urban structures," *I. J. Robotic Res.*, vol. 25, no. 5-6, pp. 403–429, 2006.

[4] G. Ros, A. Sappa, D. Ponsa, and A. M. Lopez, "Visual slam for driverless cars: A brief survey," in *IEEE Workshop on Navigation, Perception, Accurate Positioning and Mapping for Int. Veh.*, 2012.

[5] H. Lategahn and C. Stiller, "City gps using stereo vision," in *Vehicular Electronics and Safety (ICVES), IEEE International Conference on*, July 2012, pp. 1 –6.

[6] S. Gehrig, S. Wagner, and U. Franke, "System architecture for an intersection assistant fusing image, map, and gps information," in *Intelligent Vehicles Symp., Proc. IEEE*, June 2003, pp. 144–149.

[7] N. Fairfield and C. Urmson, "Traffic light mapping and detection," in *Robotics and Automation (ICRA), IEEE International Conference on*, May 2011, pp. 5421 –5426.

[8] S. Cavegn and S. Nebiker, "Automated 3d road sign mapping with stereovision-based mobile mapping exploiting disparity information from dense stereo matching," in *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2012, pp. 61–66.

[9] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, pp. 105–119, 2010.

[10] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2012.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," Cornell Computing and Information Science, Tech. Rep., 2004.

[12] D. Pfeiffer and U. Franke, "Towards a global optimal multi-layer stixel representation of dense 3d data," in *British Machine Vision Conference (BMVC)*, Dundee, Scotland, August 2011.

[13] H. Badino, D. Huber, and T. Kanade, "Real-time topometric localization," in *International Conference on Robotics and Automation*, May 2012.