Lane-Level Positioning With Sparse Visual Cues

Victoria Kogan[1], Ilan Shimshoni[2], and Dan Levi[3]

[1] Department of Computer Science, University of Haifa, Israel
[2] Department of Information Systems, University of Haifa, Israel
[3] General Motors Advanced Technical Center, Israel

***Abstract*— Vehicle localization and autonomous navigation consist of accurately positioning a vehicle in a lane. This paper presents topological localization methods by matching the visual cues from the on-board monocular camera images and the preprocessed database. We propose two methods for vehicle localization. The first exploits the 3D information of the sparse visual cues from the database. The relative vehicle rotation and translation are extracted using SOREPP method which is able to handle challenging scenarios with extremely low inlier fractions. The translation length is selected among multiple triangulation candidates. In order to select the best candidate, we suggest a robust soft-threshold estimation method which is not prone to local maxima even when the inliers' fragment is very small. The other method seeks for a refined test vehicle position estimation using a soft-threshold on Sampson distances and Cross-ratio measurements given a current noisy vehicle pose and several near-by database images. This method does not require any 3D knowledge for operation. The main challenge our optimization algorithm addresses is due to the camera and scene configurations. The database images and the test image are all taken from positions which are approximately co-linear. In addition, the scene points visible in all these images are almost co-linear with the camera positions. In this configuration, standard localization algorithms will exhibit difficulties in obtaining accurate results. The novel algorithm we present here is able to overcome this problem. The suggested localization methods are initialized by a rough position estimation thus require a regular vehicle GPS on-board. A monocular camera is required for the topological localization. We evaluate the proposed methods on real data from the KITTI database with the RTK-GPS output as ground truth.**

I. Introduction

Providing reliable and accurate car position estimation on the road is a necessary step towards autonomous driving. The Global Navigation Systems (GPS) become more and more popular these days. However, the accuracy of these devices depends on several factors. The quality of the navigation system hardware plays a huge role in the resulting positioning output and high accuracy is not available to the general public due to its high cost. GPS suffers from satellite blockage occurring in urban environments, under bridges, tunnels or in forests. The aiding systems like the Inertial Measurement Unit (IMU) allows a GPS receiver to work when GPS signals are unavailable and reduce the recovery time; however, in continuous GPS outages it is impossible to eliminate the drift. Various studies have been conducted to overcome these issues. Use of digital maps coupled with tracking algorithms is widely researched [1]. Modern navigation applications use map-matching techniques in order to localize the vehicle.

Another approach makes use of sensor information. The vehicle is equipped with sensory hardware such as IMU sensor [2], LIDAR and/or camera. The advantage of such systems is autonomous data acquisition which is usually not prone to the atmospheric changes and satellite visibility as with GPS. Various algorithms make use of this information, for example by 3D-maps matching [3]. However, many of the sensors are not widely used in the commercial automobile sector due to their high cost. In the last decade, the price and size of cameras is decreasing while the quality of their images is getting better. Nowadays, many cars have built-in cameras to monitor the front, rear and side view.

The vehicles are naturally equipped with different hardware sets. The ultimate goal is to develop a system that will use all the available information from each vehicle, while each sensor will contribute to the position approximation accuracy. In this paper we propose two methods of location approximation. Both methods use a database; it can be either prerecorded, self-updated or both. The first method is applicable for the database built using a vehicle equipped with a stereo camera, or a mono camera with a 3D depth scanner. Such equipment enables estimation of the feature point's 3D location. It is also possible to use the output from the structure from motion algorithms. The second method is applicable when no 3D information is provided. Both sources, however, can co-exist in a single localization system. Both databases must have a measure of the vehicle's position either in a local or the global Cartesian Coordinates System including an uncertainty estimation. The database can be built either using a GPS system on board and/or using the proposed algorithm.

The main challenge our algorithm addresses is due to the camera and scene configurations. Consider a vehicle driving on a road. The images taken during the database construction and the test image are all taken from positions which are approximately co-linear. In addition, the scene points visible in all these images are also close to co-linear with the camera positions. In this configuration, standard localization algorithms will exhibit difficulties in obtaining accurate results. The novel algorithm we present here is able to overcome this problem.

II. Related Work

The presented work is related to vehicle position estimation using an image database recorded on a given driving

path. This is a sub-problem of the Simultaneous Localization and Mapping (SLAM). SLAM algorithms such as in [4], [5] estimate the vehicle's location while building the map of the near-by area.

Most position estimation approaches start with topological localization [6], [7], [8], [9] which is a method that identifies the database image that is most similar to the current image. Most localization methods rely on the extraction of features from images. [7] evaluated the use of SIFT and SURF features for long-term seasonal topological localization. The results showed that SURF, or rather the upright version of SURF denoted U-SURF, had the best performance and found more relevant keypoints that generate valid matches and was much faster than SIFT. [10] showed that ORB features turn out to be less accurate and also less reliable than the SURF and SIFT-GPU features. The latest, however, requires a relatively advanced GPU.

The next step estimates the vehicle location given a set of landmarks and the noisy current and optionally all past vehicle poses. [6] suggests to use a Bayes filter. A similar approach with a Kalman filter is suggested by [11], and a particle filter in [1]. These methods may have complexity issues since the particle filter's time complexity is linear with respect to the number of particles. Naturally, the more particles, the better the accuracy, so there is a compromise between speed and accuracy. Scalability could be improved by considering only subproblems at each time step and postponing a global update as long as possible, however this makes these methods prone to drifting issues. Another approach [4] uses Bundle Adjustment (BA) to solve a maximum-likelihood estimation (MLE) over the space of the image features and camera poses by means of nonlinear least squares estimate. However, it requires high memory usage that scales with size of the map.

The complete framework for autonomous vehicle navigation is presented in [12]. The control guides the vehicle along the previously recorded visual route while calculating the direction to the next visual checkpoint by finding corresponding features in both views followed by the application of a robust estimation method as the RANSAC [13]. The main weakness of RANSAC is the necessity to sample a valid set. As the inlier fraction decreases, the probability to sample a valid set drops rapidly, increasing greatly the required number of iterations. In recent years considerable progress has been made in developing estimation algorithms that tackle these problems. Such algorithms include BEEM [14], BLOGS [15], and recently USAC [16]. However, scenarios with wide baseline images or small overlapping regions between the images still challenge even the current state-of-the-art algorithms due to the low inlier fractions. We find the relative vehicle location using the epipolar geometry estimator SOREPP [17]. For the close cameras scenarios with large number of correspondences SOREPP estimated the epipolar geometry more accurate than USAC with the five-point algorithm.

The proposed method is also a topological localization method. The database construction is described in the next section. At the beginning of the test drive we obtain a rough global vehicle position and match the test image to the database images. Then we provide a fast MLE position estimation algorithm given the landmarks set. This localization algorithm is described in Section IV. Finally, we review the accuracy of the suggested method in different road scenarios and suggest the improved optimization algorithm in Section V.

III. Database Construction

We start with the database construction step by recording an image dataset using a dedicated vehicle equipped with a calibrated camera and a high-precision GPS system. The database vehicle can also have a 3D sensor, like LIDAR, or a calibrated stereo system on board as an option. The recording software must synchronize all the sensors on board with a high-precision clock. Since the database is created off-line, further post processing steps like INS data tight coupling and GNSS corrections will be performed on the GPS data in order to improve the accuracy. Finally, all the images are tagged with an exact rotation and position with position uncertainty covariance matrix, in the absolute world coordinates.

We extract U-SURF and SIFT descriptors from each image. We store the U-SURF descriptors in randomized KD-Trees [18] for fast nearest neighbor searching in order to match the database images to the image taken from the test vehicle. The SIFT features are extracted for an accurate approximation of the test vehicle's 6-DOF position.

When the 3D information is available, the distance from the camera to each 3D point with its uncertainty is saved for each extracted feature. This information can be obtained using a 3D sensor, structure from motion or using a stereo camera system. In order to reduce the cost of the database recording vehicle and to simplify the algorithm we use a stereo camera system. The 3D points are extracted for each feature match between the perspective views of the same 3D scene taken by two calibrated cameras [19]. In addition to the distance from the camera to each 3D point, we also calculate its dispersion at a certain configuration. Our estimate is based on a first order approximation derived in [20]. We also calculate the Focus of Expansion (F.O.E.) for every database images pair as suggested in [21] in a certain radius. In the localization step, we use the F.O.E. point as a 4th image point required for the cross-ratio calculation, which represents a matching correspondent point in an image taken by a car at infinity.

IV. Localization

With a new image taken by the test drive vehicle's camera we obtain its rough location, either from a GPS or car odometry, or from the previous run of the algorithm. At the first run the vehicle position uncertainty is large and imitates a lost-in-space scenario, while in future runs (tracking mode) the uncertainty is much lower, enabling us to adjust the search radius accordingly. We query the database for all the close images in terms of vehicle's position in the world in

a certain radius from the approximated test vehicle location. We then narrow the search to the k closest images from the near-by images retrieved in the previous step using an image similarity measure. As described in Section III, the database contains the randomized KD-Trees with the U-SURF descriptors for a fast nearest neighbor searching, and the image similarity can be defined as a normalized sum of the descriptors' similarity. An overview of this and additional image comparison methods can be found in [22].

We seek to estimate the 6-DOF rotation and translation parameters which imply a position in the world from the static scene of the test vehicle image and a matched database image. Firstly we extract the test vehicle's relative rotation and the translation direction using SOREPP [17]. SOREPP uses pose priors which are naturally available in automotive scenarios due to restricted vehicle movement speed and direction. It has several advantages over the other methods, for example its running time is much faster, and its ability to handle challenging scenarios with extremely low inlier fractions. The next two subsections describe the estimation methods of the 6th DOF parameter translation distance α between the database and the test cameras. Finally, given the relative 6-DOF position estimation: rotation R_{AT} and translation direction $\hat{t}_{AT}$, the global test camera's rotation R_T and location t_T in the world coordinates is calculated:

$$R_T = R_A * R_{AT},\ t_T = t_A + \alpha(-R_A^{\intercal} * \hat{t}_{AT}). \quad (1)$$

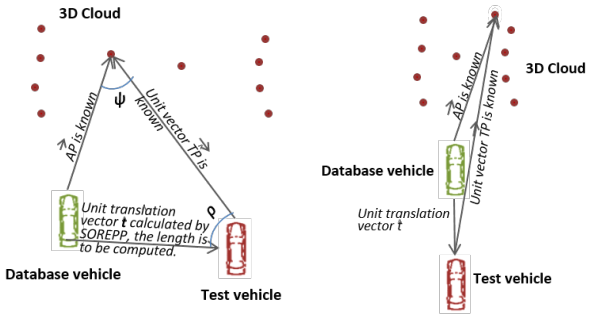
where $[R_A|t_A]$ is the extrinsic matrix of the database vehicle position.

The triangulation method described in the next subsection is based on matching the test vehicle image to single image from the database and assumes the availability of a 3D point cloud in the database. Relying on the 3D points triangulation or on the epipolar geometry, however, may introduce inaccuracy in a single-lane scenario, where both the database and the test vehicles are located on the same driving lane. In the subsection IV-B we introduce the cross ratio estimation between the test image and two database images in order to obtain an accurate translation distance estimation between the database and the test cameras when located on the same driving lane.

Finally, the subsection IV-C introduces a robust soft-threshold algorithm which is not prone to local maxima and robust to the outliers used by the two suggested methods for the translation distance estimation.

A. Exploiting 3D Information

Since a 3D point represents a physical feature in space, it can be observed from different viewpoints, as in Fig. 1a. Each 3D point correlated with a SIFT feature found both on the test and the database images implies a triangle with the following edges: the rays from the database camera and the test camera to the 3D point, $\overrightarrow{PA}$, $\overrightarrow{PT}$, and the translation vector between the cameras $\vec{t}_{AT}$. The directional vectors of these three edges are known as well as the distance $|\overrightarrow{PA}|$. Therefore, the translational direction length $\alpha = |\vec{t}_{AT}|$ can



(a) (b)

Fig. 1. (a) A triangle labeled with the components of the law of sines. ρ, ψ are the angles, and $\vec{AP}$, $\vec{t}$ are the sides opposite them. (b) A narrow triangle formed by the two vehicles driving closely on the same lane. A small error in the translation direction $\hat{t}$ between the database and the test vehicle and errors in the 3D points estimation result in a large error in the translation distance $|\vec{t}|$ calculation.

be computed using the law of sines:

$$\frac{\alpha}{\sin\arccos(\overrightarrow{PA}\cdot\overrightarrow{PT})} = \frac{|\overrightarrow{PA}|}{\sin\arccos(\overrightarrow{AP}\cdot\hat{t}_{AT})}.$$

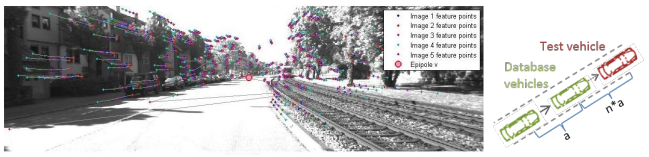
Since we construct many triangles, one for each 3D point which is visible both by the stereo pair and the test camera, the number of α candidates is equal to the number of the 3D points. We will exploit the available information and use a robust algorithm for ignoring the outliers described in Section IV-C.

However, there is a degradation in performance when the database and the test vehicle are related by translation only especially when the 3D points are located far away from the camera, as appears in Fig. 1b. In these cases the triangles are narrow and a small error in translation direction between the database and the test vehicle and errors in the 3D points estimation result in a large error in the translation distance estimation. The next subsection describes a method that overcomes the problem of images taken by the vehicles which lie on the same line.

B. Cross Ratio

R. Basri et al. [23] make two observations. First, that two images related purely by translation give rise to the same epipolar lines irrespective of the translation distance. Second, a continuous translation of the camera along the same direction results in a monotonic translation of points along their respective epipolar lines. Given two images taken from known positions of the vehicle (database images I_1, I_2), the algorithm is able to recover the position of the test vehicle from an additional image $\hat{I}$ taken by it.

These observations are illustrated in Fig. 2a. Five images were taken by a vehicle driving on a straight line. The extracted common features are shown in the image, where each color represents a different image's features. Each five features that represent the same 3D point are connected by a line, forming tracklets. The epipole is close to the image center and represents the translation direction. Each



(a) (b)

Fig. 2. (a) Features of several images related purely by translation (taken from a vehicle driving straight) lie on the same epipolar lines originating at the epipole $\mathbf{v}$. The number of steps can be extracted from the ratios between the three images' points and the epipole. (b) Cross ratio in the world: between the database image I_1 and the I_2 the robot performed a step of size a in that direction. Denote by n the remaining number of steps of size a separating the current position from the target.

five features (tracklets) almost lie on the same epipolar line. However, there is still a small rotation between the images, so the tracklets do not form perfect lines. In order to achieve a perfect camera alignment where all the cameras look forward we perform a rectification before calculating the cross-ratio.

Suppose the translational direction from the database vehicle at image I_1 to the test vehicle is given by $\hat{t}_1$, and that between the other database image I_2 and the I_1 the vehicle performed a step a in that direction. Denote by CR the remaining number of steps of size a separating the current position from the target, as illustrated in Fig. 2b. According to R. Basri et al. [23], the number of steps can be extracted from the ratios between all the matched points of the four images:

$$CR_{I12_i} = \frac{(\mathbf{p_{I_{1_i}}} - \mathbf{p_{\hat{I}_i}})(\mathbf{p_{I_{2_i}}} - \mathbf{v})}{(\mathbf{p_{I_{2_i}}} - \mathbf{p_{I_{1_i}}})(\mathbf{p_{\hat{I}_i}} - \mathbf{v})}, \tag{2}$$

where $\mathbf{p_{\hat{I}_i}}$ is a feature point in $\hat{I}$, $\mathbf{p_{I_{1_i}}}$ in I_1, $\mathbf{p_{I_{12_i}}}$ in I_2 and they lie along the same epipolar line in the three images. We suggest to use the epipole $\mathbf{v}$ to represent a 4th image taken by a car whose position is at infinity. Thus, the CR_{12_i} is obtained as a cross-ratio along this line.

Next we will rewrite the cross-ratio in world coordinates from [23] using the positions L_1 of image I_1, L_2 of image I_2, $L_{\hat{I}}$ of image $\hat{I}$ and the position of the image at the epipole $\mathbf{v}$ at infinity:

$$CR_{W1,2} = \frac{(L_1 - L_{\hat{I}})(L_2 - \infty)}{(L_2 - L_1)(L_{\hat{I}} - \infty)} = \frac{(L_1 - L_{\hat{I}})}{(L_2 - L_1)}. \tag{3}$$

Note, that the unknown variable is the distance α between the database vehicle position L_1 and the test position $L_{\hat{I}}$. The cross ratio in image space and in the world is the same, therefore, from (2) and (3):

$$\alpha_i = L_1 - L_{\hat{I}} = \frac{CR_{I12_i}}{L_2 - L_1}.$$

This method provides us with a set of α suggestions equal to the number of tracklets, or, 3D points that are recognized as corresponding features in the two database images and the test image. In order to select the best candidate, we apply the algorithm suggested in Section IV-C. Final position and the rotation of the test vehicle in the world coordinates are calculated by (1).

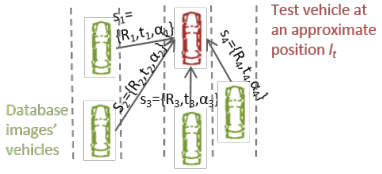


Fig. 3. A static scene of a test vehicle with 4 matched database vehicles. Since the database vehicles' positions are known we can estimate the relative position of the test vehicle.

C. Robust Distance Estimation with Soft-Threshold Method

We suggest using a robust algorithm based on the principles of MLESAC [24] which is not prone to local maxima and is robust to the outliers using a soft-threshold method. In the following we make an assumption that the noise for the each α candidate estimation is Gaussian with an estimated mean α and uniform standard deviation σ_α. Some of the features, however, are mismatched and the error in α is not Gaussian but uniform with $-\frac{v}{2}, ..., +\frac{v}{2}$ being the range within which outliers are expected to fall. The error for each α_i candidate from all the α candidates is $e_{\alpha_i} = \alpha_i - \hat{\alpha}$ and is modeled as a mixture model of Gaussian and uniform distributions. Therefore the error minimized is the negative log likelihood:

$$\underset{\hat{\alpha} \in A}{\operatorname{argmin}}(-L(\hat{\alpha})) = -\log \sum_{i=1}^{n} (\gamma \frac{1}{\sqrt{2\pi\sigma_{\alpha_i}^2}} \exp(-\frac{(\hat{\alpha} - \alpha_i)^2}{2\sigma_{\alpha_i}^2}) + (1-\gamma)\frac{1}{v}), \tag{4}$$

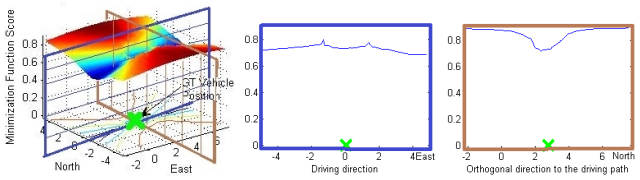
where γ is the mixing parameter implying the expected proportion of inliers. Per each candidate $\hat{\alpha}$, calculate its mixing parameter γ and then the error $-L(\hat{\alpha})$. Select the final α:

$$\alpha = \underset{\alpha}{\operatorname{argmin}}\{(-L(\alpha_i)) | i = 1..|A|\}. \tag{5}$$

The described robust method solves the outlier issues even when the inliers' portion is very small.

V. Global Position Optimization

We suggest an optimization process on the global test vehicle position that incorporates the knowledge obtained in the previous section. The initial approximation of the test vehicle location is estimated using the epipolar geometry and the cross ratio. Then, we perform the optimization. Consider the following static scene, shown in Fig. 3. For each acquired image by the calibrated test vehicle camera $\hat{I}$, consider the k matched database images. The initial rough position for the minimization function is calculated using the k database images, their global locations L_i and the test image. We use SOREPP for an each database image I_i, $i = 1..k$ with the test image $\hat{I}$. SOREPP provides an estimation for the relative rotation R_{AT_i} and translation direction $\hat{t}_{AT_i}$. There are k rays that start at locations $L_1, ..., L_k$ and intersect at some point which is a rough approximation for test vehicle position. The initial position for the optimization is the intersection point of these rays as in Fig. 3. However, the vehicles may drive



(a) (b) (c)

Fig. 4. (a) Sampson distance function evaluated around the actual vehicle position. (b) There is no absolute minimum along the driving direction which make it impossible to define the vehicle longitudinal position by the function minimization. (c) Sampson distance minimization function recognizes correctly the vehicle lateral position.

on the same line. In this case estimating the approximate intersection point of almost parallel lines will result in a very inaccurate position. We therefore estimate the initial position by additional C_2^k (choose 2 from k) calculations of the final test vehicle position using the cross ratio and selecting the median coordinates.

In [17] the following Gaussian score is defined:

$$g(c_{ij}) = \exp(-\frac{d(c_{ij})}{h}), \tag{6}$$

where $d(c_{ij})$ is a function that calculates the Sampson distance for every putative correspondence c_{ij} using the intrinsic and extrinsic matrices of the test and database cameras. This function is a type of an M-estimator on the Sampson distance associated with each correspondence with a soft threshold h. For the true relative pose structure, inliers would receive high scores, i.e., close to 1, while outliers far away from the model would receive low scores, i.e., close to 0. These scores determine the preliminary target minimization function:

$$v = \frac{1}{k}\sum_{i=1}^{k}\sum_{j=1}^{n}\hat{w}_{ij}(1 - g(c_{ij})), \tag{7}$$

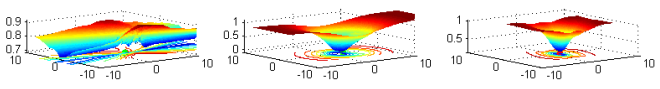
where $\hat{w_{ij}}$ are the normalized weights based on the SIFT ratio test:

$$w_{ij} = 1 - \frac{d_{1NN}^2(c_{ij})}{d_{2NN}^2(c_{ij})}, \tag{8}$$

where $d_{1NN}(c_{ij})$, $d_{2NN}(c_{ij})$ are the distances from the first and second nearest neighbors of the c_{ij}-th correspondence. Thus v is the score of a specific relative test vehicle pose s_i over all the putative correspondences from the k close database images and is in the range [0; 1].

Consider the same static scene, but with database vehicles and the test vehicle located in the same driving lane. In this case all the images imply a pure translation. Since the epipolar geometry enables us to find the rotation and translation direction up to scale, the optimization based on (7) will output a test location somewhere on that single line, where the vehicles are located but it will be a problem to find the exact position on that line as seen on Fig. 4b. Nevertheless, there is an added value in the Sampson distance function since it proved itself as a reliable position estimator in the lateral direction as seen in an Fig. 4c.

In order to refine the estimated lateral position, we suggest an additional term based on the cross-ratio between the



(a) (b) (c)

Fig. 5. Example of the optimization terms on $k = 4$ database images and a test image taken by a test car driving in the left part of the lane with $54cm$ offset from the lane center. (a) Sampson term optimization. (b) Cross-Ratio term optimization. (c) Final optimization.

feature points on the images and the vehicles' positions in the world. We start by calculating the cross-ratio $CR_{I_{ij}}$ for each two database images I_i, I_j out of the k and the test image, which are C_2^k combinations. For each three such images we calculate the cross-ratio for each feature visible on all the three images by (2) and then use the robust soft-threshold method presented in Section IV-C for finding the best cross-ratio CR_{Iij}. Finally we have C_2^k cross-ratio measures. The cross-ratio works well on the feature points correspondences that lie on the same epipolar lines, and this holds for the images taken by the vehicles positioned on the same straight lane. The cross-ratio's accuracy is compromised when the vehicles are acquiring images at a distance orthogonal to the driving direction, for example on the near-by lane. Therefore we define a weighting parameter cr_w_{ij} for each cross-ratio measure which is a measure of whether the three vehicles are located in the same driving lane. This is done in each iteration of the optimization process in which a suggested position of the test vehicle is given.

We define the following target function we would like to minimize:

$$y_{\hat{I}} = \frac{1}{C_2^k}\sum_{ij}^{C_2^k} cr\_w_{ij}(1 - \exp^{-(CR_{Ii,j} - CR_{Wi,j})^2}), \tag{9}$$

where $CR_{I_{ij}}$ was computed before running the optimization function by (2) and $CR_{W_{ij}}$ is computed by (3) in each optimization step. We finally add this term to (7) in order to define the following minimization problem:

$$\hat{s} = \underset{l_t}{\operatorname{argmin}} \frac{1}{k}\frac{1}{n_k}\sum_{i}^{k}\sum_{j_i}^{n_k}\hat{w}_j\,(1 - g(j)) + \frac{1}{C_2^k}\sum_{ij}^{C_2^k} cr\_w_{ij}(1 - \exp^{-(CR_{Ii,j} - CR_{Wi,j})^2}). \tag{10}$$

Optimization of the Sampson term in (7), cross-ratio term in (9) and the final minimization function from (10) were evaluated on different sets of near-by database images in different configurations where the database and the test vehicles were located on the same driving lane and the different driving lanes. We show the results of these optimization functions on a scenario with a test image matched to $k = 4$ database images on Fig. 5. The database images were taken in the same driving lane while the test image was taken by a car that drove at the right part of the lane with an offset of 54 cm from the lane's center. From looking at the Fig.

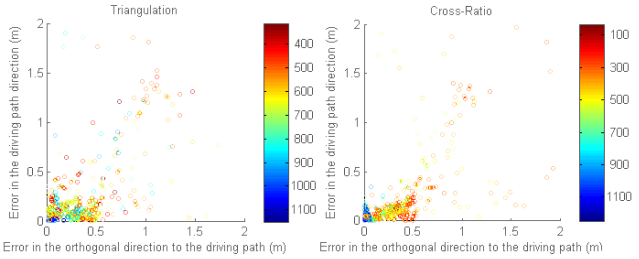


Fig. 6. Scatter plot of directional errors for single matching methods. The colors indicate the number of inlier feature matchings.

TABLE I
TRANSLATION LENGTH α ESTIMATION ERROR USING CROSS-RATIO AND TRIANGULATION METHODS.

	Triangulation	**Cross-Ratio**
Mean (m)	0.3	0.04
Median (m)	0.09	0.03
Min (m)	0	0
Max (m)	4.71	0.19

5b it is clear that the Cross-Ratio term copes well also with vehicles that are not positioned strictly on a straight lane. The contours under Cross-Ratio plot have an ellipse shape with the large axis in the driving direction. Contrariwise, the Sampson term in Fig. 5a provides an accurate lateral estimation of the vehicle position. Combining these together (Fig. 5c) refines the final estimation.

VI. EXPERIMENTAL RESULTS

We performed our experiments on the KITTI grayscale stereo sequences synchronized with GPS/IMU [25]. After rectification and cropping, the images size is 1242×375. The baseline of the stereo camera rig is approximately 54 cm. The cameras are triggered at 10 fps. In the experiments we use the 24 video sequences shot in an urban area with a total number of 6755 frames. For the experiment we use every third frame having a subsequence of 2251 stereo frames. 1501 stereo images were used to build the database. The other 750 left camera images equally spaced between the database images were used as a test set and the 750 right camera images were discarded. By dividing each driving session into the database and test sets we stage the test vehicle driving in the same lane with the database vehicle scenario.

The database stereo images were used to extract the 3D point cloud for the triangulation experiment where 480 points (minimum of 285 points) were visible both in the stereo pair and the test image out of 650 points visible in the stereo pair on average. In the cross-ratio and the optimization scenario, each database stereo camera was treated as a standalone camera having 4502 database images with 1430 SIFT features per image on average, resulting in average of 800 correspondences (minimum of 450) between the single database and test images.

The triangulation and the cross ratio methods' performance is given in Table I. When estimating the translation

TABLE II
TEST VEHICLE POSITION ESTIMATION DIRECTIONAL ERRORS FOR TRIANGULATION AND THE OPTIMIZATION WITH $k = 4$.

	Lateral Error (m)		**Longitudinal Error (m)**	
	Triang.	**Optimization**	**Triang.**	**Optimization**
Mean	0.43	0.18	0.48	0.46
Median	0.2	0.11	0.09	0.18
Min	0	0	0	0
Max	11.94	2.67	9.99	4.67

TABLE III
DIRECTIONAL ERRORS FOR CROSS-RATIO OPTIMIZATION FUNCTION FROM (9) AND THE FINAL OPTIMIZATION FUNCTION FROM (10) WITH $k = 4$ AND $k = 8$ DATABASE IMAGES.

	Lateral Error (m)				**Longitudinal Error (m)**			
	$k = 4$		$k = 8$		$k = 4$		$k = 8$	
	CR	**Fnl**	**CR**	**Fnl**	**CR**	**Fnl**	**CR**	**Fnl**
Mean	0.24	0.18	0.22	0.19	0.52	0.46	0.3	0.27
Median	0.09	0.11	0.17	0.14	0.34	0.18	0.2	0.14
Min	0	0	0	0	0	0	0	0
Max	4.96	2.67	2.5	1.4	5.5	4.67	1.98	2.05

distance α, cross-ratio outperforms the triangulation method. The test vehicle position is estimated by adding the translation vector $\vec{t} = \alpha\hat{t}$ to the matched database vehicle position. The translation direction $\hat{t}$ is calculated by SOREPP and suffers from inaccuracy in challenging scenarios with low number of corresponded features. Both this error and the α calculation error are accumulated and have a bad effect on the final test vehicle position estimation compared to the promising results in Table I.

For an accurate estimation of the driving lane we are interested in a small lateral error that is orthogonal to the driving direction. The directional errors are shown in Table II and in Fig. 6. As can be seen in the figure, the images with high numbers of feature matchings yield better results with directional errors close to 0. In order to improve the overall performance the results with low number of inliers can be discarded or paired with another information source in order to receive an uninterrupted localization service. The lateral errors appear to be a bit smaller than the longitudinal errors in the driving direction.

Next, we evaluate the optimization method. We perform the evaluation on $k = 4$ and $k = 8$. The evaluation results in Table III show that the lateral accuracy is almost unchanged, however, there is a significant improvement in the driving direction when the matched database images number k is larger. The optimization method provides better results compared to the triangulation and cross ratio methods.

We show several driving sessions with the suggested methods driving paths estimation in Fig. 8.

VII. CONCLUSION AND FUTURE WORK

We have presented a vehicle localization method by matching the visual cues found on the single on-board

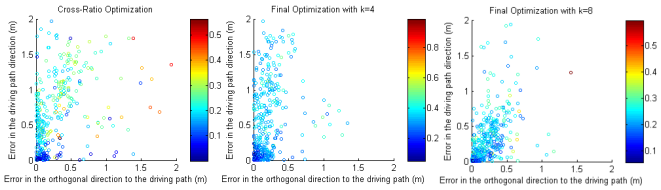


Fig. 7. Scatter plot of directional errors for multi-matching optimization method. The colors indicate the optimization function value.

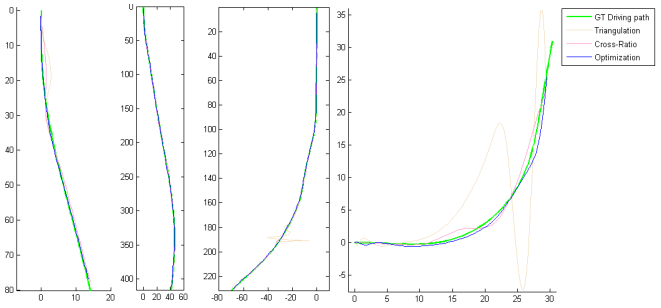


Fig. 8. Driving sessions with GT path and the estimated driving paths.

camera images. We reviewed two methods for vehicle localization; one estimates and exploits the 3D information while the other runs the optimization on the images alone. The epipolar geometry methods, based on minimizing the Sampson distance or the 3D points triangulation, alone are not able to provide the ultimate solution because they allow excessive freedom in the driving lane direction. Coupling such a method with a cross-ratio term provides an accurate solution for all the 6 DOFs.

We evaluated the proposed localization methods with RTK-GPS sensor output as the ground truth. The GPS sensor output is not perfectly synchronized with the images acquisition process. In addition, the GPS system outages are inevitable in a rural scenario so a drifting error is added to the synchronization error. The GPS position acquired per each frame and the 3D positions of the features can be refined using BA. Then an additional evaluation of the suggested algorithm should provide better results yielding lower errors.

The suggested optimization method can be expanded to enrich the database with freshly acquired images by the test vehicles' single camera. An additional optimization step will refine the initial database location and reduce its' uncertainty while saving the proximity of the new database vehicle location to the initial database location by the Mahalanobis distance. The database vehicles' position uncertainty is either provided by the GPS system or calculated using visual constrains. The advantage of such expansion is a fast adaptation to the environmental and seasonal changes.

ACKNOWLEDGMENT

We would like to thank the General Motors Advanced Technical Center in Israel which provided the authors with a research grant to perform this research.

REFERENCES

[1] A. U. Peker, O. Tosun, and T. Acarman, "Particle filter vehicle localization and map-matching using map topology," in *IV*. IEEE, 2011, pp. 248–253.

[2] L. Zhao, W. Y. Ochieng, M. A. Quddus, and R. B. Noland, "An extended kalman filter algorithm for integrating GPS and low cost dead reckoning system data for vehicle performance and emissions monitoring," *The journal of Navigation*, vol. 56, no. 02, pp. 257–275, 2003.

[3] I. Baldwin and P. Newman, "Road vehicle localization with 2D push-broom LIDAR and 3D priors," in *ICRA*. IEEE, 2012, pp. 2611–2617.

[4] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast-scale outdoor navigation using adaptive relative bundle adjustment," *IJRR*, vol. 29, no. 8, pp. 958–980, 2010.

[5] H. Strasdat, J. Montiel, and A. J. Davison, "Scale drift-aware large scale monocular SLAM." *RSS*, vol. 2, no. 3, p. 5, 2010.

[6] H. Badino, D. Huber, and T. Kanade, "Real-time topometric localization," in *ICRA*. IEEE, 2012, pp. 1635–1642.

[7] C. Valgren and A. J. Lilienthal, "SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments," *Robotics and Autonomous Systems*, vol. 58, no. 2, pp. 149–156, 2010.

[8] H. Kume, A. Suppé, and T. Kanade, "Vehicle localization along a previously driven route using an image database," in *IAPR ICMVA*, 2013.

[9] H. Lategahn, M. Schreiber, J. Ziegler, and C. Stiller, "Urban localization with camera and inertial measurement unit," in *IV*. IEEE, 2013, pp.719–724.

[10] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An evaluation of the RGB-D SLAM system," in *ICRA*. IEEE, 2012, pp. 1691–1696.

[11] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter based SLAM," *T-RO*, vol. 23, no. 5, pp. 1036–1049, 2007.

[12] J. Courbon and Y. Mezouar and P. Martinet, "Autonomous navigation of vehicles from a visual memory using a generic camera model," *TITS*, vol. 10, no. 3, pp. 392–402, 2009.

[13] A. M. Fischler, and C. R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[14] L. Goshen and I. Shimshoni, "Balanced exploration and exploitation model search for efficient epipolar geometry estimation," *TPAMI*, vol. 30, no. 7, pp. 1230–1242,2008.

[15] A. S. Brahmachari and S. Sarkar, "BLOGS: Balanced local and global search for non-degenerate two view epipolar geometry," in *ICCV*. IEEE, 2009, pp. 1685–1692.

[16] R. Raguram and O. Chum and M. Pollefeys and J. Matas and J. Frahm, "USAC: A universal framework for random sample consensus," *TPAMI*, vol. 35, no. 8, pp. 2022–2038, 2013.

[17] Y. Goldman, E. Rivlin, and I. Shimshoni, "Robust epipolar geometry estimation using noisy pose priors," *Submitted to Pattern Recognition*, 2015.

[18] I. Z. Emiris and D. Nicolopoulos, "Randomized kd-trees for approximate nearest neighbor search," CGL-TR-78, NKUA, Tech. Rep., 2013.

[19] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[20] R. M. Haralick, "Propagating covariance in computer vision," in *PCCV*. Springer, 2000, pp. 95–114.

[21] J. M. Zezhi Chen, Nick Pears and T. Heseltine, "Epipole estimation under pure camera translation," in *Proceedings of the 7th International Conference on Digital Image Computing: Techniques and Applications, Macquarie University, Australia*, 2003, pp. 849–858.

[22] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: an experimental comparison," *Information Retrieval*, vol. 11, no. 2, pp. 77–107, 2008.

[23] R. Basri, E. Rivlin, and I. Shimshoni, "Visual homing: Surfing on the epipoles," *International Journal of Computer Vision*, vol. 33, no. 2, pp. 117–137, 1999.

[24] P. H. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[25] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *IJRR*,2013.