The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

On the Bending Problem for Large Scale Mapping

I. Esteban, O. Booij, J.Dijk, F. Groen
TNO Defense, Security and Safety
Intelligent Systems Lab Amsterdam - University of Amsterdam

Abstract— **During Simultaneous Localization And Mapping, geometrical constraints are established between map features. These constraints, introduced through measurements and motion prediction, produce a bending effect in the event of closing a large loop. In this paper we present a discussion of the bending problem for trajectory based representations. Furthermore, we propose a generic approach to reduce the bending effect that exploits common geometrical constraints in human-made environments through the use of a Heading Based Curvature measurement. We show by means of experimental results that our approach increases significantly both the global and local map accuracy.**

I. Introduction and Motivation

Simultaneous Localization and Mapping (SLAM [1]) has seen a fruitful research during the last decade (see [2] for a complete survey). Most recently, special attention has been given to reduce the computational complexity of the task of estimating and updating a global map given a set of geometrical constraints and their uncertainties. Many techniques have been presented to overcome the computational limits of the standard Kalman Filter (KF) solution. State-augmentation, sparsification, particle filters or sub-mapping [2] are the most relevant successful approaches. Exactly Sparse Delayed State Filters (ESDSF) have gained particular attention due to their potential to solve very large mapping problems where thousands of states need to be estimated [3] [4]. However, the size of the environment is not the only limiting factor when consistently mapping large areas. The source of error for the family of Kalman Filter approaches is threefold: *linearization, data association and error models.*

KF based solutions for non linear processes are handled by linearizing about the mean of the state usually through a first order approximation. This is a well known problem specially relevant in large scale mapping. The error induced by linearization increases over time and produces an inconsistent estimation. Sub-mapping [5] and batch processing techniques [6] aim at reducing the effects of linearization and improving map consistency.

Bad data association introduces erroneous links between the states in the information matrix which can cause an inaccurate map estimation. Robust data association techniques based on appearance models [7][8] and uncertainty of pose estimation [3][9] aim to reduce the number of wrong associations with successful results.

The third source of error for KF based solutions is inaccurate error models for both the process to be estimated and the measurements taken in the environment. These models are used for the prediction and update step respectively. Usually these error models are simplified by estimating a single covariance from test data which is then used through the complete estimation. This simplification introduces artifacts in the event of closing a large loop. For delayed state approaches, this is a problem that has never been explicitly discussed in the literature to our knowledge. From here on we refer to it as the *trajectory bending* behavior. At every step in a ESDSF new links are introduced between the current pose and the previous poses through both control of the actuation, and measurements. Assuming that the number of measurements remains constant [10] and no loops are closed for a long period, the relational structure of the trajectory becomes *very strong locally*. In other words, there are more links connecting neighboring poses. It is reasonable to assume that when closing a large loop, the distance between loop closing points in the estimated trajectory is also large. When the loop-closing measurements are taken, those far away poses need to be connected. Due to its local structural strength the trajectory will bend reducing the global error in the same way as a wire bends when two extremes are forced together. The amount of bending and the shape of the bending are determined by the structure of the measurements and the error models.

Sub-mapping techniques as in [5] naturally reduce the bending behavior as the links between poses of different sub-maps are eliminated. However, they do not preserve the geometrical and statistical dependencies between different portions of the map. This usually requires after-loop closure optimization to re-locate the sub-maps.

In this paper, we explore the use of dynamic error models to reduce the bending problem in the event of closing a large loop. In the same spirit as [11], we exploit the common knowledge that the mapping vehicle drives over human-made environments where straight roads are the norm. As opposed to their approach, we do not enforce straightness between landmarks, rather we construct a dynamic observations error model based on the curvature of the estimated trajectory. We embed our dynamic models into an Omnivideo ESDSF [3] [4] [10] algorithm and introduce the metrics for measuring the improvement. Further, we present results of experiments in outdoor environments where large loops are closed.

The rest of the paper is organized as follows. In Section II we introduce our omnivision mapping approach. In Section III we introduce the conditions for the bending process to occur. Our proposed dynamic error model and the metrics to measure the improvement are also detailed. Section IV describes the results obtained. We close the paper

978-1-4244-3804-4/09/$25.00 ©2009 IEEE

with some conclusions and future work in Section V.

II. VIEW-BASED MAPPING

In order to test our approach we used an ESDSF to estimate the trajectory of our robotic vehicle, for which we assume a planar movement. The vehicle was equipped with two measuring devices. A differential GPS unit provided the ground truth data for evaluation. An omnivideo camera was mounted on the vehicle's roof which provided geometrical constraints for the mapping process. We used a standard vehicle with no odometry sensors except for the built-in speedometer.

ESDSFs work in two stages. First, motion prediction is used to obtain a rough estimation of the map, then measurements are taken and used to improve this estimation.

A. Motion Prediction

Delayed State filters adopt a state representation $\mathbf{x}_t = [X_t, M_t]^T$ where X_t represents the last vehicle pose and the map M_t consists of all previous vehicle poses: $M_t = [X_{t-1}, X_{t-2}, ..., X_0]^T$. Each vehicle pose is defined as the 2D position in the plane and orientation angle: $X_t = [x_t, y_t, \theta_t]^T$.

At every step t (image recording step), an additional robot pose is added to the state as follows: $\mathbf{x}_{t+1} = [X_{t+1}, \mathbf{x}_t] = [X_{t+1}, X_t, M_t]^T = [X_{t+1}, M_{t+1}]^T$. The new vehicle pose is calculated:

$$X_{t+1} = X_t + \mathbf{u}_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t + d \times cos(\theta_t) \\ y_t + d \times sin(\theta_t) \\ \theta_t \end{bmatrix}, \quad (1)$$

where d is the distance traveled since the last step. This distance is estimated from the observation of the average speed during the whole trajectory and considered constant. This motion prediction model is very inaccurate and only used to provide a rough scale of the map. Motion prediction with no world observations will yield a straight trajectory with equally spaced poses. Note that we are adopting a linear motion model where no rotation is considered for every step, hence no linearization error is introduced during motion prediction.

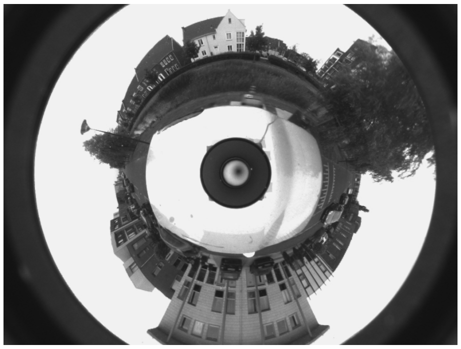

Fig. 1. Outdoors omnidirectional image taken from the roof of the vehicle

B. Omnivideo Observations

The advantages of omnidirectional images are twofold. Firstly, they convey information from the complete surroundings. Secondly, they are well suited for data association. Due to the omnidirectional point of view, image matching is more accurate and less limited compared to regular images. The lack of viewing direction is particularly interesting for loop-closure situations where image matching is now independent of the heading of the vehicle.

The goal is then to first match pairs of images and then to obtain geometrical constraints about their relative location.

1) Image Matching: Every new image needs to be compared to the previously taken images in order to obtain geometrical constraints about their poses. Comparing every new image with the rest of the set costs $O(n)$. We improve that using graph theory. We find key images based on image appearance [12] and build a connected dominating set [13]. When a new image is recorded, it is compared only with the key images, reducing the computational costs. In order to obtain more detailed measurements, the new image is compared with the images linked to the matching key image.

2) Epipolar geometry: Having obtained the image matches, we estimate the motion from the camera centers using the epipolar constraint [14]. SIFT image feature [15] correspondences are obtained for every pair of images. These correspondences can be related by the essential matrix E which describes the relative camera pose. E is then estimated with the planar version of the 8-point algorithm [14] using 3 of those correspondences. False feature matches are rejected using RANSAC and if the ratio between the inlaying correspondences and the number of features is larger than a certain threshold (which we set to 0.17) then we further decompose E to obtain a relative pose estimation using Horn‘s method [16], obtaining the heading of the translation ϕ (the baseline) and a 2D rotation θ. These two are used to construct the measurement vector $\mathbf{z}_t = [\phi_0, \theta_0, \phi_1, \theta_1, ..., \phi_n, \theta_n]^T$ used in the ESDSF.

C. Exactly Sparse Delayed State Filter

ESDSFs are part of the Information Filters family. Traditional Kalman Filter approaches estimate the mean state $\mu_{\mathbf{x}}$ and covariance $\Sigma_{\mathbf{x}}$. Information filters estimate the information vector $\eta_{\mathbf{x}}$ and information matrix $\Lambda_{\mathbf{x}}$ instead. These are related by $\Lambda_{\mathbf{x}} = \Sigma_{\mathbf{x}}^{-1}$ and $\eta_{\mathbf{x}} = \Lambda_{\mathbf{x}} \mu_{\mathbf{x}}$.

As described by [10], an information filter becomes naturally sparse if a Delayed State approach is taken as the filter equations can be implemented in constant time as shown in [10]. The state recovery is solved using the Cholmod2 function that is part of the SuiteSparse package [1] by Tim Davis to solve the sparse, symmetric, positive-definite linear system $\Lambda_{\mathbf{x}_t} \mu_{\mathbf{x}_t} = \eta_{\mathbf{x}_t}$ in close to linear time.

[1] SuiteSparse: collection of packages for large sparse matrices. http://www.cise.ufl.edu/research/sparse/SuiteSparse/

III. Dynamic Error Models and the Bending Problem

A. The Bending Problem

One of the fundamental requirements of consistent map building is loop-closure. The error estimation during the mapping procedure tends to accumulate over time. Closing loops bounds the growth of this estimation error by recognizing when a previously seen area is revisited.

Given that world observations and odometry measurements are noisy, it is reasonable to expect that in the event of closing a large loop, the estimated distance between connecting poses is large. After a sufficient number of loop-closure observations, the map is re-estimated in order to correct for the accumulated error, connecting those coincident poses.

There are two conditions for the bending to occur. First, a sufficiently *large loop needs to be closed*. Second, the *observations need to be local*. The bending is guaranteed if these two are met. Udo Frese first described in [17] the source of the bending problem as the *"Certainty of Relations despite Uncertainty of Positions"*. As the map is built, relations are established between neighboring poses through the odometry measurements and world observations. Motion prediction provides information between successive poses while world observations provide information between current and observed poses. The strength of these relations will depend on the selected odometry and observations error models. As we assume that the world *is not fully observable* the nature of the relations is *local*. Neighboring poses are more strongly related than far away poses. Due to the matching between images taken close by, there are more geometrical constraints between these poses than far away poses, enforcing certain probabilistic rigidity in the structure of the trajectory. This rigidity is highly visible in the event of closing a large loop where the gap in the loop closure point is reduced and the trajectory is bent accordingly.

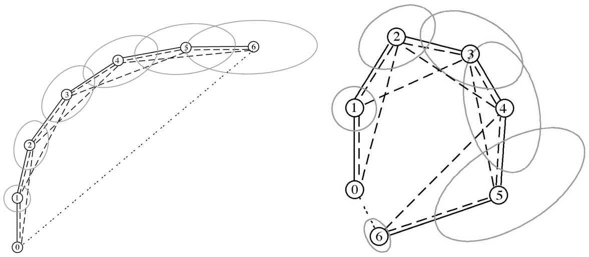

Fig. 2. LEFT: Local structure of the state. Black lines represent relations introduced by odometry. Dashed lines represent relations introduced by world observations. RIGHT: Trajectory after loop-closure.

In the example shown in Figure 2 (LEFT), seven robot poses are drawn. Image matches exist up to two images back in time. From pose 5, for example, observations are available of poses 4 and 3. It is easy to see that local poses are strongly related to neighboring poses. Pose 4 is for instance strongly related to 3 and 5. The strength of each node with other nodes weakens with the distance as there are less links. There is no link between poses 5 and 1 for instance, therefore the global relation of pose 5 with pose 1 is much weaker than the relation with pose 4. In the event of closing a loop (see Figure 2, RIGHT), geometrical constraints are observed between far away poses. These constraints are shown in the figure between poses 0 and 6 with a dotted line. If the number of observations is sufficiently large and accurate, poses 0 and 6 must be brought together. Given the relational local structure of the existing trajectory, the bending is distributed according to the error estimation of each pose (light gray ellipsoids) and the strength and number of relations of each pose with their neighbors. In our example, the bending is more severe on poses 4,5 and 6 than in poses 0, 1 and 2 due to their smaller uncertainty given than the uncertainty of the observations was kept constant (single covariance error model). If the number of links is not equally distributed and/or the strength of the links is changing, the bending will be more severe on those poses with higher flexibility. In other words, the fewer the links and the weaker the links, the more flexible a portion of the trajectory is in the case of bending. Note that in a perfectly consistent filter, the error estimation of pose 5 should cover the area where pose 5 is moved after loop closure. This example shows an inconsistent filter. Any EKF or alike will eventually become inconsistent (optimistic estimation of the error) due to both inaccurate error models and errors derived from the linearization process.

B. A Dynamic Measurement Error Model

In order to cope with the bending problem, we propose a dynamic error model (DEM) approach. Accurate error models for odometry or world observations are uncommon. Usually an accurate estimation is difficult to obtain as the true error depends on many factors such as speed, ground type, lighting conditions, etc. A dynamic approach will be more accurate, using different covariances depending on the external conditions.

We propose a generic approach where the measurement error model is built dynamically based on a *curvature* measure of the estimated trajectory. This idea is born from the experimental observation that the 8-point algorithm performs better in estimating the heading of the translation and the rotation when a straight trajectory is taken [18] (see section IV-A). In our loop closure experiments we observe that the bending is distributed according to both the number of relations between poses and the strength of these relations which is defined by the error models. In a human-made environment where corners and straight portions are the common thing to encounter, we would like the bending to occur mainly on the corner portions. If the straight sections are preserved and the corners are used as joints in the bending process, the global accuracy of the map will increase and the local structure will improve. Given that this is the precise behavior of the 8-point algorithm we construct a dynamic error model. In order to enforce the bending process after loop closure over the poses in the trajectory with highest curvature, we define the observations error model R_k at pose k as:

$$R_k = \begin{bmatrix} R_0(c_k+1)^2 B & 0 & \dots & 0 \\ 0 & R_0(c_k+1)^2 B & & 0 \\ \vdots & & & \\ 0 & 0 & \dots & R_0(c_k+1)^2 B \end{bmatrix}$$

where R_0 is the baseline error model, c_k is the curvature measure at the estimated pose k as defined below and B is a constant used to fine tune the model according to the density of the observations. R_0 represents the baseline error and it is estimated using the single covariance approach so that the performance is best. The quadratic form is used to enforce even further the straightness of the majority of the trajectory. During our experiments we observed that the constant B is not sensitive if the variance in the number of observations per meter is very low. On the other hand, if the number of observations changes over time due to environmental conditions, the constant B becomes more sensitive and a change of 50% can be the difference between success or failure.

It is clear from this definition of R_k that the error model changes according to c_k. The higher the curvature measure, the higher the error. This will give less credit to the observations in the event of driving through a corner. As the model is defined not only for the current pose but all the observed poses, the strength of the relations between current and observed poses will decrease accordingly. This relaxation of the links between poses in corner-like areas will enforce the bending over those portions after loop-closure, effectively maintaining the straight portions straight. Note that we use a diagonal error model that disregards the dependencies between the estimation of the rotation and translation and the estimation between multiple frames. Even though this is a simplistic approach, we show in our experiments that the idea of a dynamic error model based on an online curvature measure works. A more accurate error representation with dependency terms will most likely improve the estimation even further, though the dependencies in the covariance matrix between the rotation, translation and across frames is unclear at the moment.

C. Metrics

1) Heading-Based-Curvature: We define *Heading Based Curvature* (HBC) as the average change in heading θ of the last n poses:

$$c_k = \frac{\sum_{i=k-1}^{k-n} (\theta_k - \theta_i)}{n}, \tag{2}$$

where k is the current robot pose.

This measure allows the identification of curves and corners in the trajectory. Its purpose is the evaluation of the performance in the estimation of the trajectory at a local level. First we assign an HBC measure to every pose of the ground truth (GT), then we can compare this value with the HBC of the estimated trajectory. Observing this difference, we can then evaluate how close the estimation is from the GT *locally* regardless of the position, orientation and scaling. This will give an impression on how similar the trajectory is with respect to corners or straight sections.

It is important to note that we use the same metric for the definition of the error model and the evaluation of the local performance. However, this metric is only used to define the error model and has not direct influence on the final curvature of the estimated trajectory after loop-closure. Also, the metric used for evaluation is measured at the estimated trajectory after loop-closure, while the metric used for the definition of the error model is estimated online and might differ much from the HBC of the final estimated trajectory.

2) Best-Fit model: In order to measure the global similarity between the estimated trajectory and the GT, we employ a best-fit model. The nature of the estimation of the trajectory when mapping is relative, while the nature of the GPS data is global. Therefore, we need to be able to compare both maps. The scale of the estimated trajectory is provided by the motion model so only affine transformations H are allowed for the GT trajectory to fit the estimated data such that the sum of the square of the pose-to-pose euclidean distances $D_i = \sqrt{(x^i_{GT} - x^i_{SLAM}{}^2 + y^i_{GT} - y^i_{SLAM}{}^2)}$ is minimized.

IV. Results and Discussion

In this section we present the results of our approach to reduce the bending problem in a medium size loop-closure scenario. First we evaluate our dynamic error model using a simulated dataset, then we validate our results in a real world outdoors environment.

A. Estimating Rotation and Translation

In this experiment we show how the error in the estimation of the rotation and translation using the 8-point algorithm grows as the trajectory curvature increases. We use simulated point correspondences and a perfect trajectory to evaluate the error in the estimation.

Figure 3 shows the error distribution when estimating the motion of a simulated camera (640x480, 5 mm focal length). We simulated 10000 steps with a varying angle for turning to the right hand side (0, 10 and 20 degrees). For every step, the motion is calculated using 100 noisy image feature matches and the error is calculated. For the rotation, we use the simple subtraction between the true rotation and the estimated rotation. In the case of the translation, we measure the angular distance between the true translation direction and the estimated direction. It is clear that when estimating the translation the error distribution becomes wider as the turning angle increases. This shows that the performance of the motion estimation algorithm depends on the curvature of the trajectory, the more the camera turns, the less accurate the estimation of the translation direction is. In the case of the rotation, the difference in the error distribution is not that noticeable. However we can see that the error in the case of a straight trajectory is much smaller than when the camera is turning. Also, the noise in the estimation increases (left and right extremes of the plot) as the turning angle increases.

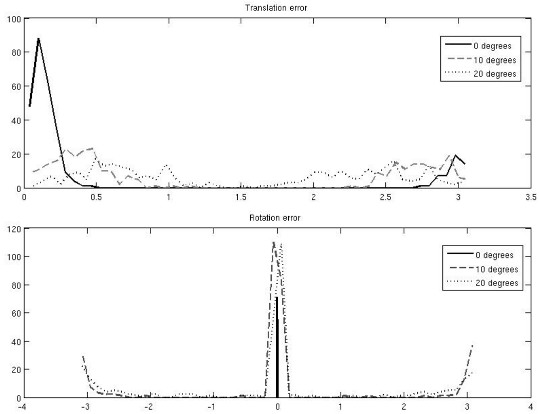

Fig. 3. [Artificial Dataset] Error distribution (histograms) in the estimation of rotation and translation using the plannar version of the 8-point algorithm. Y axis - density. X axis - radians. Deviation in image feature location: 0.001 pixels.

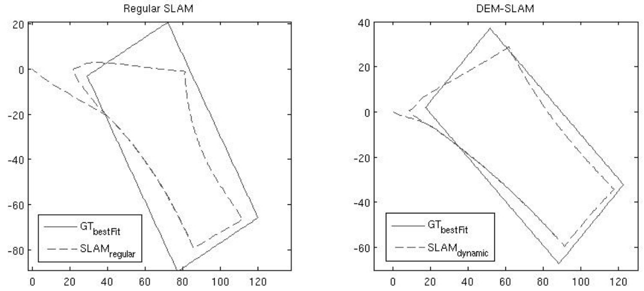

Fig. 4. [Artificial Dataset] Estimated trajectories and GT best fit.

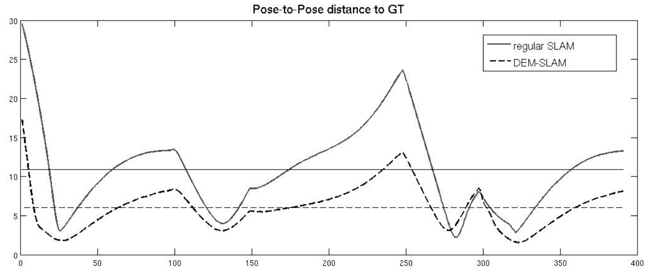

Fig. 5. [Artificial Dataset] Distance to GT best fit. Average regular estimation: 10.8602. Average DEM approach: 6.0338. X-axis shows the pose, Y-axis shows meters.

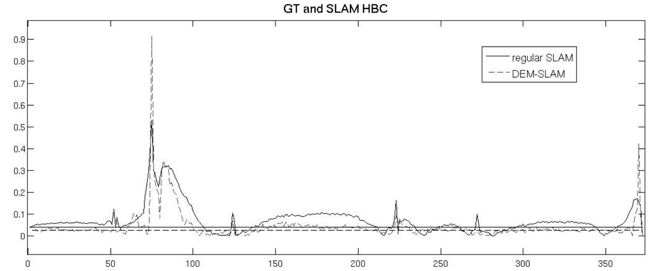

Fig. 6. [Artificial Dataset] GT and estimated trajectory HBC difference. The horizontal axis represents the pose number, the vertical axis is the absolute difference between GT HBC and the estimated trajectory HBC. The average differences are also shown: regular estimation is 0.0401 and DEM estimation is 0.0253

B. *Artificial Dataset*

We evaluate the performance of our approach in a simulated dataset where an imaginary vehicle drives over a perfect rectangle. The trajectory consists on 5 different straight portions, 4 for the rectangle plus 1 for the overlap (see Figure 4). The sides of the rectangle measure 100 meters for the long side and 50 meters for the short side. The overlap takes place on the long side. Observations are created with added Gaussian noise with standard deviation of $\sigma = 0.05$ radians. The number of observations per step are limited to the number of poses in the GT that lay within a 10 meter radius. A constant speed model similar to the one adopted in the real vehicle is employed. The speed is set to 1 image-meter per time unit. The diagonal elements of the process error matrix were set to $[1.15, 1.15, 0.0873]$ (m,m,radians). The diagonal elements of the baseline observation error matrix were set to 0.075 radians. For the dynamic error approach a constant $B = 0.5$ was used. Even though a deviation of 0.05 might seem a bit large, we tried to reproduce as best as possible a real scenario where the images are noisy (see Figure 3). Also, note that we do not perform a refinement step like Bundle Adjustment as we are interested in the performance of the loop closure process. A refinement step would further improve the overall result. The baseline error matrix is slightly larger than the true error in order to simulate an inconsistent filter with an overestimated error model.

Figure 4 shows the best fit GT trajectory for both the regular estimation and our DEM approach. It is easy to appreciate that our approach performs better globally as the estimated trajectory is in general closer to the GT trajectory. Also, the rectilinear portions of the trajectory are better preserved after loop-closure. The corners in the top end are also improved. Figure 5 shows quantitatively this improvement. Pose-to-pose euclidean distances from the GT to both the regular and DEM estimations are shown. Our approach not only displays a significant improvement around the corner points (poses 100, 150 and 250) but also on the average distance. The DEM trajectory is on average 44% closer to the *best fit* ground truth trajectory. Figure 6 shows the local performance in terms of HBC. The DEM trajectory shows better performance both in the straight portions and the corners, where the angles are closer to orthogonal . Only at two poses in the corners the performance is worse than the regular estimation. The DEM HBC difference shows on average a 37% improvement with respect to the regular approach.

C. *Mapping a Large City Block*

In order to test our approach in a more realistic scenario, we recorded a dataset in an urban outdoors environment. 826 images were recorded at an approximate rate of one image every 1.1 meters. The average observed driving speed was 10 km/h. The total length of the trajectory according to the GPS was 912.5 meters. A sample image is shown in Figure 1. We used the same baseline error model and value for the constant B as in the simulated dataset.

V. Conclussions and Future Work

We presented a generic solution based on the observation that most mapping approaches take place in human-made environments where geometrical constraints are known. By defining the observations model according to the heading-based-curvature of the estimated trajectory, we have shown

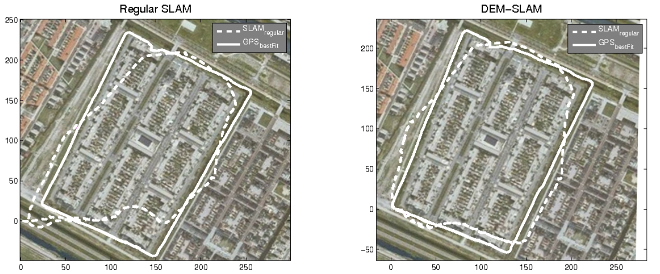

Fig. 7. [Outdoors Dataset] regular trajectories and GT best fit. A Google map image is show in the background. A dotted line shows the estimated trajectory.

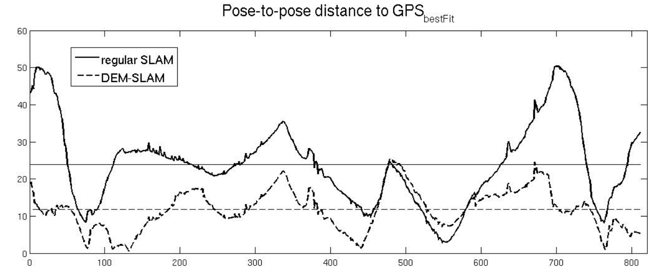

Fig. 8. [Outdoors Dataset] Distance to GT best fit. Average regular estimation: 23.9285. Average DEM approach: 11.8228

that good improvements can be achieved with respect to regular static error models approaches. Our DEM solution shows an important improvement in terms of pose-to-pose euclidean distances and a significant improvement with respect to the local structure of the trajectory. In particular, we show that the accuracy of straight and corner points in the estimated trajectory improves. The results we obtained are accurate given that we use a single omnivideo camera and a naked-eye observation of the average speed to reconstruct the trajectory of the vehicle. A more accurate motion prediction mechanism will further improve the estimation.

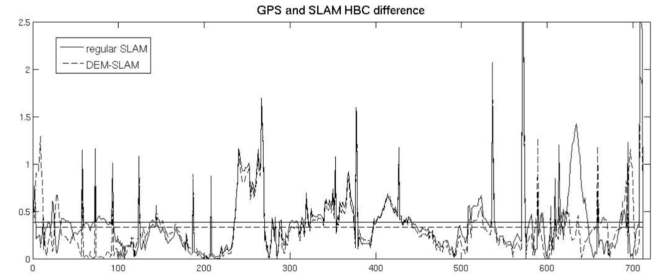

Fig. 9. [Outdoors Dataset] GT and estimated trajectory HBC difference. Horizontal axis is the pose, vertical is the absolute difference $abs(GT - HBC)$ and HBC of estimated trajectory. Average differences are also shown: regular estimation is 0.3816 and DEM approach is 0.3344

Figure 7 shows the GPS and estimated trajectories. Our technique performs better in capturing the shape of the real trajectory even in the presence of noisy observations. Figure 8 shows the pose-to-pose distances from the GPS data to the estimated trajectories. DEM estimation performs better over the complete trajectory showing only comparable results to the regular approach around pose number 500. On average, DEM estimation is 49% closer to GPS data. The local performance shown in Figure 9 also shows some improvement. In this case the average HBC difference is better by 10%.

In order to further develop our approach, larger experiments should be made with more complex trajectories. Also a refinement step such as BA should be added and a more complex covariance structure should be explored. In summary, we have presented two contributions to the task of mapping large human-made environments with a camera. Firstly our Dynamic Error model approach that shows a consiste improvement over standard techniques. Secondly our Heading Basec Curvature measure which is used to improve the map accuracy but can also be employed to evaluate trajectories in their local structure. Given the computational advantages of ESDFs, we believe that a consistent map can be built of large-size urban environments with a simple and inexpensive setup as we have shown.

ACKNOWLEDGMENT

This study was funded and supported by the Netherlands Forensisc Institute, TNO and the University of Amsterdam within the Netherlands project for large scale CCTV analysis.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (slam): Part i the essential algorithms," in *Robotics and Automation Magazine*, 2006.

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localisation and mapping (slam): Part ii state of the art," in *Robotics and Automation Magazine*, 2006.

[3] R. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation in an unstructured environment using a delayed state history," in *ICRA*, 2004, pp. 25–32.

[4] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended informationn filters," in *Int. J. Robot. Res.*, 2004.

[5] J. A. Castellanos, R. Martinez-Cantin, J. Tardós, and J. Neira, "Robocentric map joining: Improving the consistency of ekf-slam," in *Robotics and Autonomous Systems*, 2007.

[6] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous lozalisation and mapping," in *IEEE Transactions on Robotics*, 2005.

[7] F. Ramos, J. Nieto, and H. Durrant-Whyte, "Recognising and modelling landmarks to close loops in outdoor slam," in *ICRA*, 2007.

[8] P. Newman and K. Ho, "Slam- loop closing with visually salient features," in *ICRA*, 2005.

[9] L. Clemente, A. Davidson, I. Reid, J. Neira, and J. Tardós, "Mapping large loops with a single hand-held camera," in *Robotics: Science and Systems*, 2007.

[10] R. Eustice, J. Leonard, and H. Singh, "Exactly sparse delayed-state filters for view-based slam," in *IEEE Transactions on Robotics*, 2006.

[11] K. R. Beevers and W. H. Huang, "Inferring and enforcing relative constraints in slam," in *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2006.

[12] B. Kröse, O. Booij, and Z.Zivkovic, "A geometrically constrained image measure for visual mapping, localization and navigation," in *ECMR*, 2007.

[13] O. Booij, Z.Zivkovic, and B. Kröse, "Sampling in image space for vision based slam," in *RSS Workshop*, 2008.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[15] D. Lowe, "Distinctive image features from scale-invariant keypoints," in *Int. J. of Computer Vision*, 2004.

[16] B. Horn, "Recovering baseline and orientation from essential matrix," 1990. [Online]. Available: citeseer.ist.psu.edu/horn90recovering.html

[17] U. Frese, "A discussion of simultaneous localization and mapping," in *Autonomous Robots*, 2006.

[18] D. Nistér, "An efficient solution to the five-point relative pose problem." in *IEEE Transactions on Pattern Analisys and Machine Intelligence*, 2004.