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

Intelligent Vehicle Localization Using GPS, Compass, and Machine Vision

Somphop Limsoonthrakul*, Matthew N. Dailey*, and Manukid Parnichkun†
*Computer Science and Information Management
†Mechatronics
Asian Institute of Technology
Klong Luang, Pathumthani, Thailand
{somphop, mdailey, manukid}@ait.ac.th

Abstract— **Intelligent vehicles require accurate localization relative to a map to ensure safe travel. GPS sensors are among the most useful sensors for outdoor localization, but they still suffer from noise due to weather conditions, tree cover, and surrounding buildings or other structures. In this paper, to improve localization accuracy when GPS fails, we propose a sequential state estimation method that fuses data from a GPS device, an electronic compass, a video camera, and wheel encoders using a particle filter. We process images from the camera using a color histogram-based method to identify the road and non-road regions in the field of view in front of the vehicle. In two experiments, in simulation and on a real vehicle, we demonstrate that, compared to a standard extended Kalman filter not using image data, our method significantly improves lateral localization error during periods of GPS inaccuracy.**

I. INTRODUCTION

Among the challenges involved in building a safe intelligent vehicle, localization is among the most important, because without precise knowledge of the vehicle's location with respect to its surroundings, autonomy is impossible. Although GPS devices are extremely useful for localization, they are not sufficient by themselves, because satellite signal quality varies with weather and proximity to trees and buildings. The problem is especially acute in urban areas. Under these circumstances, accurate and robust localization relies critically on additional sensors or filtering techniques.

There is a great deal of previous work using Kalman filters to improve GPS-based vehicle localization. Cooper et al. [1] propose an extended Kalman filter (EKF) model for vehicle navigation that incorporates a GPS device and an inertial navigation system (INS). Sadiadek et al. [2] improve the EKF for GPS/INS localization using fuzzy logic to adapt prediction and sensor noise strength. Thrapp et al. [3] and Bonnifait et al. [4] demonstrate EKFs that fuse GPS and odometry data, and Panzieri et al. [5] use an EKF to fuse GPS, INS, odometry, and laser scanner data. Machine vision techniques are also proving useful; Georgiev [6] presents a method using camera pose estimation to improve localization in urban environments when GPS performance becomes low. The method fuses GPS, odometry, and compass data using an EKF, but when the EKF's uncertainty grows too large, monocular vision is used instead of the GPS signal. Agrawal and Konolige [7] present a localization method using stereo vision and GPS. In their work, visual odometry is fused with GPS measurements using an EKF.

Although the EKF is efficient, linearizing the motion and sensor models can introduce inaccuracy, and its assumption of a Gaussian posterior distribution over vehicle poses means it can fail when the true distribution is non-Gaussian, especially when it is multi-modal [8], [9]. To solve this problem, Dellaert et al. introduce a localization method for indoor mobile robots using particle filter called Monte Carlo localization (MCL) [10] and apply the technique to the task of vision-based localization [11]. This work demonstrates the robustness of particle filters for localization with ambiguous sensor information.

In our work, we complement a GPS device, compass, and wheel encoders with machine vision to address GPS inaccuracy, and we use a particle filter to address linearization error and the limitations of the Gaussian posterior assumption. Our machine vision technique extracts road regions from the field of view in front of the vehicle. By comparing the observed road region with that expected based on a candidate vehicle position and a predefined map, we can compute the likelihood of the observation given the candidate vehicle position and, to the extent that the map and road region classification are accurate, thereby improve vehicle localization precision.

II. ROAD REGION CLASSIFICATION

We use a forward-pointing camera and road region classification to improve localization accuracy. As shown in the flowchart in Fig. 1, we perform Gaussian smoothing to reduce image noise then classify each pixel in the image as road or non-road using a H-S color histogram. We then transform the classification results from the image plane to the (robot-relative) ground plane using a pre-calculated planar homography. The resulting robot-relative road region measurement vector can be used for vehicle localization.

A. Hue-Saturation Histogram

We use a 2D histogram to represent the distribution of road pixels' color. Histograms are attractive because they are simple to calculate and easy to use. We use the hue and saturation components in the HSV color model [12] to determine whether each pixel is likely to be on the road or not because, unlike the RGB color space, HSV

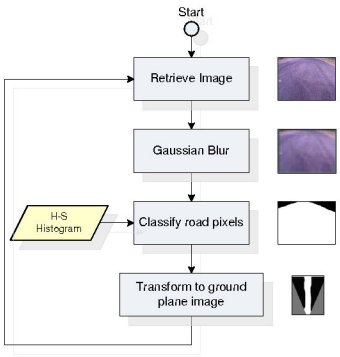


Fig. 1. Overview of road classification method.

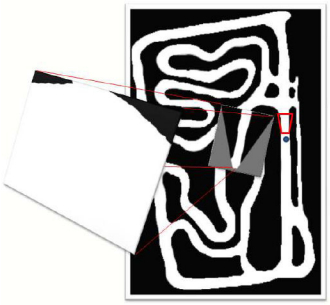


Fig. 2. Using a road classification image to measure a particle's likelihood.

represents intensity or brightness on a dimension orthogonal to color. Since only a 2D histogram needs to be estimated, much less traning data is typically needed for equally good performance.

To calculate a 2D histogram for road pixels, we simply count the number of occurrences of each H-S pair in a training data set. Then we normalize the histogram by the total number of pixels in the training data. We use road images acquired under a variety of different lighting conditions.

The H-S histogram gives us a model $P(H = h, S = s \mid C = \text{road})$, i.e., the class-conditional probability of obtaining particular H-S values (h, s) when the class C of the pixel in question is "road." We threshold this probability in order to classify each pixel in the region of interest. We use as a threshold that which is optimal in terms of the F_1 measure for separating road and non-road pixels over a training set.

B. Image Plane to Ground Plane Transformation

The road classification result is used as observation data in our localization module. The algorithm needs to compare the road classification results with a predefined map to measure the likelihood of candidate vehicle positions. To enable fast computation in the localization module, we transform the classification results from the image plane to the ground plane. We calibrate the camera with respect to the vehicle then pre-compute a 2D homography H between the camera's image plane and the ground plane.

III. LOCALIZATION

This section describes the system state, motion model, and measurement model comprising the design of our particle filter based localization algorithm.

A. System State

We assume a discrete time system with a constant sampling interval. The state of the system at time t is

$$\mathbf{x}_t = \begin{bmatrix} x_t & y_t & \theta_t \end{bmatrix}^T,$$

where (x_t, y_t) is the center of the vehicle's rear axle and θ_t is its orientation with respect to the world coordinate plane.

B. Motion Model

We use two drive wheel encoders and a steering wheel encoder to obtain odometry information

$$\mathbf{u}_t = \begin{bmatrix} d_t & w_t \end{bmatrix}^T,$$

where d_t is the distance traveled from time $t-1$ to time t (simply the average of the distances obtained from the two drive wheel encoders) and w_t is the steering wheel angle at time t. We treat the odometry measurement as a control action and model the motion error as Gaussian:

$$\begin{bmatrix} \hat{d}_t \\ \hat{w}_t \end{bmatrix} = \begin{bmatrix} d_t \\ w_t \end{bmatrix} + \begin{bmatrix} \varepsilon_d(d_t) \\ \varepsilon_w \end{bmatrix}$$

where $\varepsilon_d(d_t) \sim \mathcal{N}(0, \alpha_1 d_t)$ and $\varepsilon_w \sim \mathcal{N}(0, \alpha_2)$. The error variance for d_t is a linear function of d_t. α_1 and α_2 are vehicle-specific error parameters, which we currently obtain through trial and error.

Given an odometry measurement, we discretize the bicycle kinematic model [13], assuming no slip, to predict the vehicle's motion:

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} R_t(\sin\theta_t - \sin(\theta_t - \phi_t)) \\ R_t(\cos(\theta_t - \phi_t) - \cos\theta_t) \\ -\phi_t \end{bmatrix},$$

where $R_t = l/\tan(w_t)$ is the current turning radius (l is the distance between the front and rear axle) and $\phi_t = d_t/R_t$ is the change in heading angle.

C. Measurement Model

The GPS and compass yield a simple linear measurement model

$$\mathbf{z}_{G,t} = \mathbf{x}_t + \varepsilon_G$$

for which a Gaussian error model $\varepsilon_G \sim \mathcal{N}(\mathbf{0}, \mathtt{R})$, $\mathtt{R} = \text{diag}\left(\sigma_G^2, \sigma_G^2, \sigma_C^2\right)$, is appropriate.

The measurement model for the H-S histogram-based road region classifier is necessarily more complex. After transformation to the ground plane in the vehicle's coordinate system, we have a measurement

$$\mathbf{z}_{C,t} = \begin{bmatrix} c_{t,1} & c_{t,2} & \dots & c_{t,K} \end{bmatrix}^T,$$

where c_i is the predicted class (road or non-road) of the i-th pixel and K is the number of pixels in the vehicle coordinate ground plane region of interest.

To calculate the likelihood of a candidate vehicle position according to the road classification image, then, we compare each pixel in the image with the corresponding location in a binary map created using satellite images and manual labeling of the road and non-road regions. The general idea of this measurement is shown in Fig. 2. We assume conditional independence of the road image pixel measurements given the vehicle's position and the map. We set the assumed probability of true positive, false positive, true negative, and false negative measurements by trial and error.

Combining the two models and exploiting conditional independence assumptions, we define the weight $w_t^{(i)}$ for particle i at time t as

$$\begin{aligned} w_t^{(i)} &= p(\mathbf{z}_t \mid \mathbf{x}_t^{(i)}) \\ &= p(\mathbf{z}_{G,t} \mid \mathbf{x}_t^{(i)}) p(\mathbf{z}_{C,t} \mid \mathbf{x}_t^{(i)}) \\ &= p(\mathbf{z}_{G,t} | \mathbf{x}_t^{(i)}) \prod_{j=1}^{K} p(c_{t,j} \mid \mathbf{x}_t^{(i)}). \end{aligned}$$

To avoid numerical problems and to prevent the GPS and compass measurement from being overwhelmed by the image measurement, we use the modified measurement model

$$\log(w_t^{(i)}) = \alpha + \beta \log p(\mathbf{z}_{G,t} \mid \mathbf{x}_t^{(i)}) + \gamma \sum_{j=1}^{K} \log p(c_{t,j} \mid \mathbf{x}_t^{(i)}),$$

where α prevents the image weights from driving the total weight to 0, and β and γ adjust the relative importance of the GPS/compass and image terms. Since the weights $(w_t^{(i)})$ are normallized after inverting the log, α does not affect the result mathematically but does improve the numerical stability. We determine β and γ experimentally through trial and error.

IV. EXPERIMENTAL RESULTS AND EVALUATION

In this section we describe the results of our implementation of the proposed method in an intelligent vehicle control system based on the architecture described in [14]. We test a simple control system using the proposed localization method in both a virtual reality simulation and a real vehicle. In simulation, we precisely specify the characteristics of the GPS error distribution and compare to ground truth. In the real vehicle experiment, we evaluate the results qualitatively.

In our control system, the GPS, compass, and encoder readers run at a frequency of 10 Hz, and the road classifier processes images 13 fps. Since we use a particle filter based method, the speed of the localization process depends on the number of particles (samples) used in the system. In our experiments, with 1000 particles, the localization process can run at frequency of 4 Hz with the other processes.

As a baseline for comparison, we implemented a straightforward Kalman filter based localization method for comparison with our proposed method. The Kalman filter uses the same motion model described in the previous section along with the GPS and compass sensor data. We experimentally tuned the filter's parameters (e.g. the estimation and measurement noise covariances) to perform well under normal GPS error conditions.

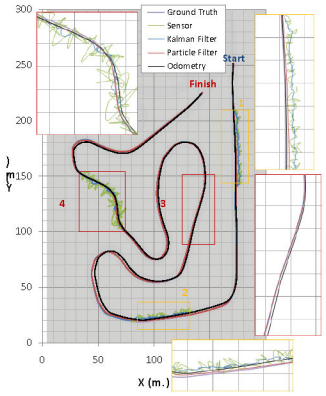
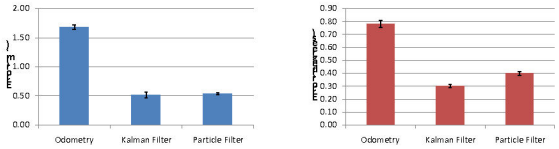


Fig. 3. Localization results in simulation.



(a) (b)

Fig. 4. Overall localization error in simulation. (a) Position error. (b) Heading error. Error bars denote 95% confidence intervals.

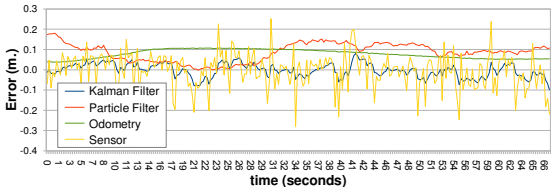
We present the simulation results first followed by the real-vehicle results. We omit results from particle filter based localization method without measurement data from vision, since even in the particle filter, the distribution of GPS and compass errors are assumed to be Gaussian, so the results are nearly identical to the Kalman filter based method.

A. Experiment I: Simulation

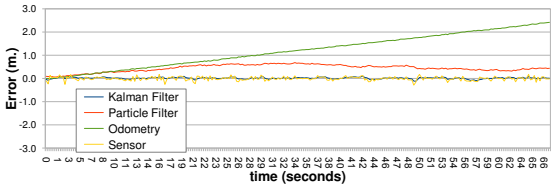
We simulated vehicle motion at 10 km/hr from a start point to a finish point along a pre-defined trajectory. The simulation setup and results are summarized in Fig. 3. We defined four areas in which the GPS error was increased or biased. In the first and second areas, the GPS error had a normal distribution with variance of 1.0 meters, and the mean was biased from the ground truth for 3.0 meters in the E-W direction (first area) or N-S direction (second area) accordingly. In the third area, the GPS was completely blocked, and in the fourth area, the GPS had an extremely high error (variance of 3.0 meters) with no bias. As shown in the result image, the localization results from our method are close to the ground truth for all cases of GPS error, and they are smoother than the results from the Kalman filter based localization method.

The localization error averaged over the entire simulation run is shown in Fig. 4. To compare the error rates in each condition, we performed paired two-tailed t-tests with a Type I error rate $\alpha = 0.05$. We found that the Kalman filter and particle filter were not significantly different in terms of position error and that the Kalman filter was significantly

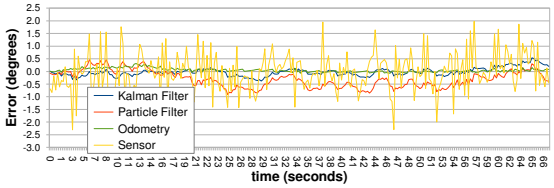
Fig. 5. Waypoints for simulated vehicle trajectory in Experiment I.



(a)



(b)



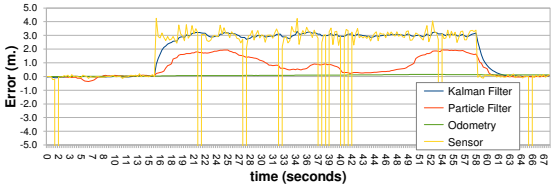
(c)

Fig. 6. Experimental results in simulation when the GPS error is small. (a) Lateral error. (b) Longitudinal error. (c) Heading error.

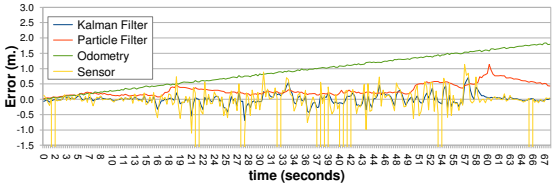
better than the particle filter in terms of heading error. This is because under normal circumstances, when the GPS error is small (variance of 0.06 meters), the Kalman based method is more precise than our method. However, from the results shown in Fig. 3, it was clear to us that the particle filter performs better than the Kalman filter during periods of unreliable GPS, so we simulated vehicle motion again from left to right along the trajectory shown in Fig. 5 under five GPS error conditions: small Gaussian error, lateral shift, longitudinal shift, large Gaussian error, and GPS signal loss.

1) Small GPS error: The GPS error was distributed as a 2D Gaussian with variance 0.06 meters and mean at the ground truth. The results in Fig. 6 show that the KF (shown in blue) performs best since the particle filter sample set may not contain a particle perfectly positioned at the ground truth.

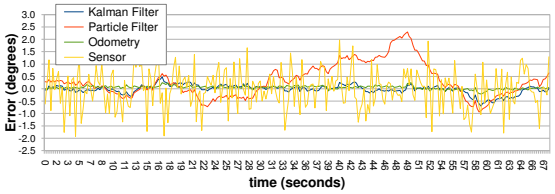
2) Shift of GPS in lateral direction: We shifted the GPS error distribution from the ground truth by 0.3 meters in the lateral direction (upward in Fig. 5) and added Gaussian noise with a variance of 0.3 meters. As shown in Fig. 7, with the addition of visual road region information, our particle filter-based method substantially decreases lateral error.



(a)



(b)



(c)

Fig. 7. Experimental results in simulation when the GPS is biased in the lateral direction. (a) Lateral error. (b) Longitudinal error. (c) Heading error.

3) Shift of GPS in longitudinal direction: We again shifted the GPS error distribution from the ground truth by 0.3 meters but this time in the longitudinal direction (rightward in Fig. 5). As shown in Fig. 8, our localization results become slowly biased by the GPS error. This is because the difference in appearance of the road regions in the longitudinal direction is small, so the distribution of the posterior depends strongly on the biased measurements from the GPS and compass.

4) Extreme GPS error: We set the GPS error variance to be high (5 meters). The results in Fig. 9 show that the PF's estimates are smoother and closer to the ground truth.

5) GPS signal loss: Finally, we blocked the GPS for some time. The only observation data used to measure the vehicle's position and orientation were from the compass and camera. The results in Fig. 10 show that our localization method is nevertheless close to the ground truth in the lateral direction, although the longitudinal error is high, since the road region images do not differentiate longitudinal positions well.

In each of these five cases, we checked whether the results of our method significantly decrease localization error or not. The graphs in Fig. 11 compare the localization error from our method and the EKF-based method in each condition. The blue vertical bars represent the absolute mean error of our method, the red bars represent the absolute mean error of EKF localization, and the vertical error bars indicate 95% confidence intervals. Paired t-test with Bonferroni correction at $\alpha = 0.05$ were used to test the statistical confidence of the results, and the evaluation shows that the error rates for our method are significantly different from the EKF error rates in every condition.

The results (Fig. 11(a)) show that our method can signif-

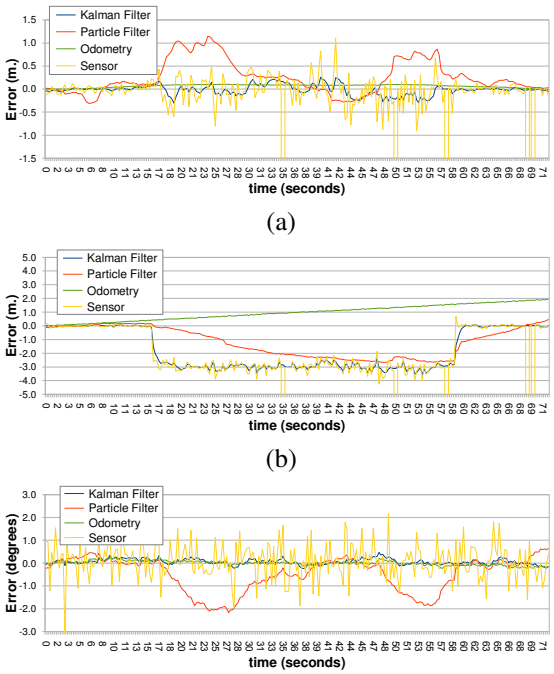


Fig. 8. Experimental results in simulation when the GPS is biased in the longitudinal direction. (a) Lateral error. (b) Longitudinal error. (c) Heading error.

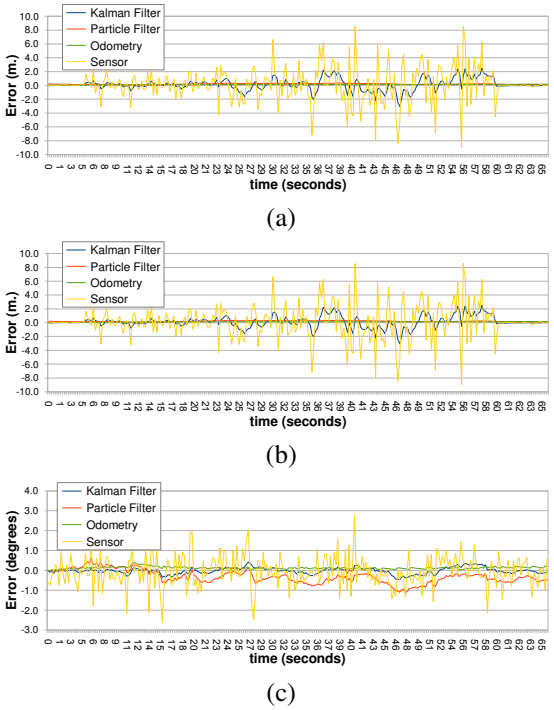


Fig. 9. Experimental result in simulation when the GPS error is extremely high. (a) Lateral error. (b) Longitudinal error. (c) Heading error.

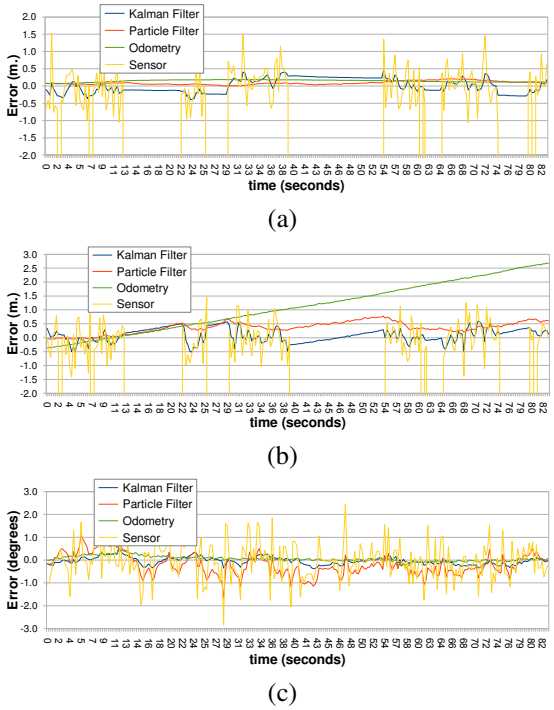


Fig. 10. Experimental results in simulation when the GPS signal is lost for some time. (a) Lateral error. (b) Longitudinal error. (c) Heading error.

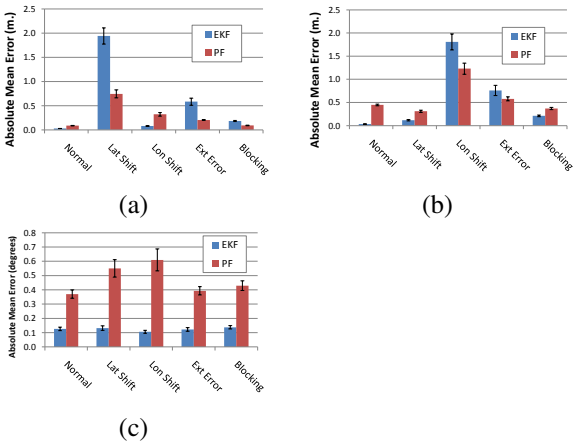


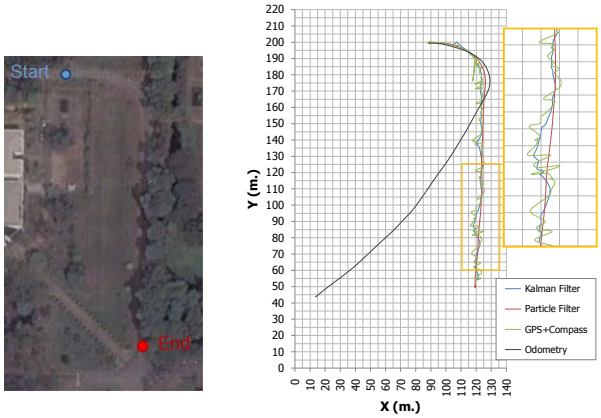
Fig. 11. Evaluation of our proposed method compared to EKF-based localization. (a) Lateral error. (b) Longitudinal error. (c) Heading error.

icantly decrease lateral localization error in comparison to the Kalman filter based localization method. In the case of small GPS error, although our method gives higher error, the difference is only approximately 10 centimeters on average, which is acceptable for vehicle localization.

For longtitudinal error (Fig. 11(b)), our method gives more precise results than the KF in the case of longitudinal shift of GPS and extreme GPS error. For other cases, our method is worse because the variability of road region appearance in longitudinal direction is small, and the filter thus ends up relying more heavily on the noisy odometry measurements.

The chart in Fig. 11(c) shows that our method does not improve orientation estimation. In the simulation, we used a small constant compass error, so the KF is more

Fig. 12. Real vehicle PA-PA-YA used in Experiment II.



(a) (b)

Fig. 13. Experiment II results. (a) Map of the traversed road. (b) Localization results.

precise at predicting vehicle heading. As we previously noted, when sensor errors is small, there may be no sampled particle positioned perfectly on the ground truth, limiting the accuracy of the particle filter.

B. Experiment II: Real vehicle

In Experiment II, we implemented our localization method on PA-PA-YA, an electric golf cart (shown in Fig. 12). It is driven by two DC motors and equipped with a GPS, a compass, three encoders (two for the drive wheels and one for the steering wheel), and an IEEE-1394 camera. The control system runs on a 2.0 GHz Pentium Core 2 Duo with 2 GB of RAM with GNU/Linux (Ubuntu 8.04).

We drove the vehicle at a speed of 5 to 10 km/hr to avoid dependency of control on the localization method used. To create a situation with noisy odometry and GPS, we chose a path covered by trees and containing speed bumps as shown in Fig. 13 (a). We drove the vehicle along the center of the road. The results of the experiment are shown in Fig. 13(b), which clearly shows that our localization method gives results that are more precise than pure odometry and more smooth than the Kalman filter.

V. CONCLUSION

We have shown that our proposed vehicle localization method can increase accuracy in situations where GPS is unreliable. Our machine vision method identifies the road region in front of the vehicle, and our particle filter fuses that result with GPS, compass, and odometry measurements. Although longitudinal error is reduced only moderately by our method, lateral error is substantially reduced. We consider lateral error to be more serious than longitudinal error, because lateral error could cause the vehicle to leave its lane or go off the road.

The proposed localization method can be improved to further reduce longitudinal error. One possible solution is to use additional observation data such as visual odometry. Another solution may be to use other kinds of sensor such as a laser scanner in addition to the camera.

VI. ACKNOWLEDGMENTS

This research was supported by the Thailand National Electronics and Computer Technology Center (NECTEC) and the Royal Thai Government. We are grateful for the hardware support provided by the AIT Intelligent Vehicle team. Methee Sripundit provided valuable comments and support on this work.

REFERENCES

[1] S. Cooper and H. Durrant-Whyte, "A Kalman filter model for GPS navigation of land vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1994, pp. 157–163.

[2] J. Z. Sasiadek, Q. Wang, and M. B. Zeremba, "Fuzzy adaptive Kalman filtering for INS/GPS data fusion," in *Proceedings of the 2000 IEEE International Symposium on Intelligent Control*, 2000, pp. 181–186.

[3] R. Thrapp, C. Westbrook, and D. Subramanian, "Robust localization algorithms for an autonomous campus tour guide," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 2065–2071.

[4] P. Bonnifait, P. Bouron, P. Crubille, and D. Meizel, "Data fusion of four ABS sensors and GPS for an enhanced localization of car-like vehicles," in *Proceedings of the 2001 IEEE International Conference on Robotics and Automation (ICRA)*, 2001, pp. 1597–1602.

[5] S. Panzieri, F. Pascucci, and G. Ulivi, "An outdoor navigation system using GPS and inertial platform," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 2, pp. 134–142, 2002.

[6] A. Georgiev, "Localization methods for a mobile robot in urban environments," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 851–864, 2004.

[7] M. Agrawal and K. Konolige, "Real-time localization in outdoor environments using stereo vision and inexpensive GPS," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, 2006, pp. 1063–1068.

[8] S. Arulampalam and B. Ristic, "Comparison of the particle filter with range parameterized and modified polar EKFs for angle-only tracking," *Signal and Data Processing of Small Targets*, vol. 4048, no. 1, pp. 288–299, 2000.

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

[10] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, 1999, pp. 1322–1328.

[11] ——, "Using the condensation algorithm for robust, vision-based mobile robot localization," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999, pp. 588–594.

[12] R. Gonzalez and R. E. Woods, *Digital Image Processing, 2nd Edition*. Prentice-Hall, Inc, 2002.

[13] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2005.

[14] S. Limsoonthrakul, M. Dailey, M. Srisupundit, S. Tongphu, and M. Parnichkun, "A modular system architecture for autonomous robots based on blackboard and publish-subscribe mechanisms," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2008, pp. 633–638.