Probabilistic Contour Extraction with Model-Switching for Vehicle Localization

Thommen Korah Christopher Rasmussen

Department of Computer and Information Sciences
University of Delaware
Newark, DE 19711

korah,cer@cis.udel.edu

***Abstract*— Over the past few years, Global Positioning Systems (GPS) have been increasingly used in passenger and commercial vehicles for navigation and vehicle tracking purposes. In practice, GPS systems are prone to systematic errors and intermittent drop-outs that degrade the accuracy of the sensor. In this work, we describe an approach to localizing vehicles with respect to the road given erroneous sensor measurements using only aerial images. Our method works on both urban and rural areas, while being robust to a number of occlusions and shadows. The spatial tracker incorporates multiple measurement models with varying constraints, automatically detecting and switching to the appropriate model. We demonstrate our technique by correcting in real-time highly inaccurate GPS readings collected while driving in diverse areas.**

I. INTRODUCTION

A primary task involved in equipping intelligent vehicles with autonomous capabilities is that of robot localization, which is the problem of estimating a robot's position relative to a map of its environment. Localization includes both the ability to home in on the position without any prior information of initial state, as well as tracking the position when the earlier state is known. The need for a highly accurate localization process is crucial for tasks such as map-building, path-planning and autonomous navigation [3], [9], [10].

Sensors like Global Positioning Systems (GPS) and odometry [11] have been widely used for this purpose. In practice however, GPS accuracy is heavily dependent on several factors such as the satellite configuration and multi-path errors. Line-of-sight (LOS) issues make GPS less effective in urban canyons and densely forested regions. GPS errors can routinely range from 2- to 15 meters depending on the sophistication of the unit [11].

To correct such noise in the GPS position, several map-matching approaches [1], [17] have been employed. These techniques use a digital road network and a combination of geometric and topological constraints to "snap" onto the correct road. Digital road-maps can be problematic in dense urban environments as there may be several candidate roads close to a particular location. Localization in off-road and desert terrain is particularly challenging as these maps may not be available.

Vision has recently been investigated as an effective tool to correct for such erroneous sensor data. Much research has been done in robot localization [3], [12] to complement GPS or sonar readings with another on-board sensor such as a camera or laser range finder. This usually entails searching for artificial/natural landmarks in the vicinity of the GPS-estimated position for increased accuracy. Information from on-board sensors are compared to a world model to determine the absolute pose of the robot. These state estimation problems are effectively solved by probabilistic approaches like Bayesian inference, which recursively estimates the posterior probability density over the state space, conditioned on the data collected so far. Implementations of the Bayes filter differ in the manner by which this density is represented. Kalman filters [13] are the most widely used variant due to their efficiency, but have restrictive assumptions such as unimodal Gaussian uncertainty and linear system dynamics.

A powerful means of representing the belief state is *particle filtering* [15], also known as the CONDENSATION [14] algorithm and Monte Carlo Localization (MCL) [9]. Particle filters fall under the general class of Monte Carlo methods which are based on representing a probability distribution function by a set of random weighted samples. The 'particles' represent the distribution of the state vector in state space, and are iteratively updated after an observation. The observation model describes the likelihood of an observation given the current state. Advantages of particle filters include the ability to represent arbitrary probability densities, and applicability to converge in non-Gaussian, non-linear dynamic systems.

This work describes methods to localize a vehicle traveling along a road, given noisy GPS way-points, and an aerial photograph of the surrounding region. We frame vehicle localization as a Bayesian inference problem to integrate and arbitrate between these multiple sources of information. After describing how particle filters are used to localize a vehicle on an aerial map of the region, we detail our measurement likelihood function as used in the prediction and update phases. Finally, we describe our results on varied environments.

Our road extraction method is most similar to JetStream [2], a particle filtering approach to spatially track edge contours including roads. In our case, GPS information places constraints on the tracking and eliminates the high level of user interaction required by JetStream. Monte Carlo methods have been used to localize robots in constrained environments [15], combining measurements from multiple sensors such as GPS, dead reckoning systems, and cameras [12]. Frueh and Zakhor [8] have used particle filters to register laser scan data with Digital Surface Maps, to build 3D textured models of cities. We propose to localize the vehicle using only 2D aerial photographs, which provide higher spatial resolution and important color information.

All the above methods employ only a single cue to measure the strength of their belief. Common cues employed in tracking such as color, edges, or feature templates generally do not work well alone in a wide range of environments. A good measurement likelihood function must be able to determine the most appropriate model at any given time and adaptively switch to the dominant one. Isard and Blake [6] have described a mixed-state CONDENSATION tracker that can handle multiple motion models. While we have a well-defined motion model, we adapt their technique to handle variable modes of perception.

II. Particle Filtering and Vehicle Localization

Particle filtering has proven to be adept at tracking in the presence of complicated likelihood functions and non-linear dynamics. Tracking here refers to following *the state* of a set of variables $\mathbf{x}$ as they evolve over time. We wish to estimate $\mathbf{x}^t$ at time-step t, given knowledge about all the sensor measurements $\mathbf{Z}^t = \{\mathbf{z}^1, \mathbf{z}^2, .., \mathbf{z}^t\}$ up to t. If we construct the posterior density $p(\mathbf{x}^t|\mathbf{Z}^t)$ to represent our belief of the current state, we can infer $\mathbf{x}^t$ by taking either the MAP (maximum a posteriori) or mean estimate.

In particle filters, this belief or posterior density is approximated by a set S^t of N particles $\mathbf{s}_i^t = \{< \mathbf{x}_i^t, w_i^t > |i = 1, .., N\}$, where $\mathbf{x}_i$ is a state (position on the map) and w_i is the importance weight. The importance weights give a measure of how reliable the corresponding state estimate is. The set of samples thus define a discrete approximation of the continuous probability density function. In every iteration, N new particles are sampled from S^t with the probability of survival of a particle $\mathbf{s}_i^t$ being proportional to its weight w_i^t. Each particle is then modified according to the *dynamics* and the weight is updated according to the *measurement model*.

Initially, a set of equally weighted particles are uniformly distributed around the starting position estimated by the GPS reading (u_1, v_1). The state vector $\mathbf{x} = [x, y, \theta, m]$ includes a variable for the road width m, where (x, y) is the mid-point of the road oriented at an angle of θ degrees. At every iteration of the particle filter algorithm, a new set of GPS coordinates (u_t, v_t) is obtained and the relative motion (R_t, Θ_t) is computed. We now describe two schemes to integrate GPS sensor information into the particle dynamics.

A. GPS-driven dynamics

The motion (R_t, Θ_t) is applied to all the particles along with white Gaussian random noise to predict the new position of each particle. The particles are thus subjected to a drift and diffusion process with a relative movement of $(R_t^{'}, \Theta_t^{'}) = (R_t + N(\sigma_r), \Theta_t + N(\sigma_\Theta))$. On applying the motion model, we predict state $\mathbf{x}_i^t = [x', y', \theta', m']$ from $\mathbf{x}_i^{t-1} = [x, y, \theta, m]$ by applying the dynamics:

$$\mathbf{x}_i^t = \begin{pmatrix} x' \\ y' \\ \theta' \\ m' \end{pmatrix} = \begin{pmatrix} x + R_t^{'} \sin(\Theta_t^{'}) \\ y + R_t^{'} \cos(\Theta_t^{'}) \\ \Theta_t^{'} \\ m + N(\sigma_m) \end{pmatrix} \tag{1}$$

where $N(\sigma)$ denotes Gaussian noise with variance σ^2. The standard deviations are a function of the step size with σ_Θ varying from $0.05\sqrt{R}$ to $0.1\sqrt{R}$ and σ_r is set to $0.2R$. When the GPS estimates are fairly reliable, this scheme forces particles to follow the GPS point and look for a possible road within the error ellipsoid.

B. Measurement Likelihood-driven dynamics

If the GPS positions are not reliable, the above method overly restricts the particles from following the road. Setting a larger variance with the previous method does not eliminate the problem, as a majority of particles would still follow the incorrect GPS track. This is especially true in cluttered urban regions since there might possibly be many road-like features such as house tops or shadows of buildings. A more flexible approach in such situations is based on the strength of the measurement likelihood - i.e. allow particles to follow all possible roads and then use GPS to narrow down on the most plausible one in the update phase. Details of the likelihood function are elucidated in the next section.

The formula for $\mathbf{x}_i^t$ is the same as (1) except that the orientation $\Theta_t^{'} = (\Theta_{t-1}^{'} + N(\sigma_\Theta))$ is distributed around the direction the particle was traveling in the previous time step. This allows a much wider angular distribution as well as giving particles a certain momentum to get past erroneous GPS measurements. The formula for $R_t^{'}$ remains the same as before, in order to retain information about the velocity of the vehicle.

C. Update phase

A preliminary weight w_i is computed for each new particle based on the measurement model. This weight could be a measure of our confidence that the estimated position of the particle on the aerial photo lies on the road or how well on-board camera images correlate with the aerial view of the corresponding region. The weighting serves to concentrate the histogram over state space of all the particles around the most likely position that the vehicle could be in. Only the “fittest” particles survive from one iteration to the next, resulting in an evolutionary process. Given this representation of the density function, the current estimate for the position of the vehicle is chosen to be the weighted mean of all the particles.

III. Measurement Likelihood

We employ vision-based techniques to assign relative weights for each particle. These weights reflect the strength of our belief that a particle lies on the road based on the aerial photo. Most particle filters, including JetStream, are flexible and have very few restrictions on the measurement model used. However, a single measurement model alone might not be sufficient to characterize the observation density. Multiple observation models are important for robustness and applicability, with automatic switching between them. We first define the different road models used and in the next section describe a technique to arbitrate between them.

A. Measurement Models

Fig. 1. Example aerial images of urban and desert regions from which we wish to spatially track the road traveled by the vehicle. Roads might appear bright or dark and may have very little contrast in off-road environments. The road is marked in blue for the lower desert image.

Figure 1 shows example aerial images of off-road and urban environments on which we run our technique. JetStream [2] uses the norm of the luminance gradient as a cue to track high contrast contours. The aerial images that we employ are low contrast, noisy and characterized by excessive clutter in urban regions, making simple edge- or color-based methods impractical. We have therefore chosen Gabor filters [5], widely used in texture analysis, to give an initial confidence estimate for each pixel being road or non-road. The general functional for the two-dimensional Gabor filter family can be represented as a Gaussian function modulated by an oriented complex sinusoidal signal. In polar form it is written as

$$G_n(x, y, \lambda) = e^{-\pi[x^2/a^2 + y^2/b^2]} e^{j2\pi[r\cos(\theta-\phi)/\lambda]}. \qquad (2)$$

The real part of the Gabor filter (RG_n) has even symmetry and is a proven blob detector while the imaginary part (IG_n) can be used to detect step edges. Since roads appear as banded segments oriented at some angle, we attenuate RG_n along its width, to give an elliptical pattern that retains only the central 40 percent of the estimated width. While this particular pattern effectively detects bright roads in a darker region, a negated filter (RG_n^-) can detect darker roads. The width of the roads in the image dictate the choice of various scales(λ), and for each scale 10 equally separated orientations are selected from $[0..\pi]$. We then use normalized cross-correlation rather than convolution to pre-compute the response for each of these filters. Doing so compensates for intensity changes, while enabling seamless fusion of multiple scale filter responses.

The measurement models are a function of the features detected by one or more filters in the pre-processing stage. Other features such as color or shape can also be used when available. We currently use only $N_o = 2$ different models defined analytically as:

- M_1= $$RG_n(x^+, y^+, \lambda_1) + RG_n(x^-, y^-, \lambda_1) + RG_n^-(x, y, \lambda_2)$$
- M_2= $$RG_n(x^+, y^+, \lambda_1) + RG_n(x^-, y^-, \lambda_1) + RG_n(x, y, \lambda_2)$$

The superscripts over x and y indicate the road edges on both sides of the particle computed from m and θ. Model M_1 looks for a dark (with respect to the surrounding region) road with thin parallel lines running along the road. This model is very effective for urban roads as well as certain shadowed or occluded regions. Model M_2 is used to detect brighter roads, typical of rural or country roads. A weighting scheme may also be used in the above models to emphasize certain features more than others. For example, the sidewalk is very prominent in suburban areas and so a higher weight factor can be multiplied with the road edge response. Adding more models is straightforward and the method to switch between them is described in the next section. The weight of a particle $\mathbf{X} = [x, y, \theta, m]$ using model o is the output of M_o at orientation closest to θ for the point (x, y).

B. Fusing GPS Information

While JetStream follows contours along the largest gradient, we wish to constrain the dynamics of the tracking to follow the road in the vicinity of the GPS reading at that instant. Although the dynamics is governed completely by the GPS in the GPS-driven scheme, this alone could still cause particles to drift away and track the wrong road. Depending on the accuracy of the GPS and the amount of ambiguity in the aerial maps, we can use an α parameter that weights a particle based on its proximity to the corresponding GPS coordinate. However this will “pull” particles off the road in places where GPS estimations are inaccurate. We therefore define a radial distance of d_{max}

from the GPS position, outside of which all particles are given a weight of 0.

When particles are governed by the measurement likelihood-driven dynamics, we must use an additional angular constraint as well. Given the GPS segment $(u_t, v_t) - (u_{t+1}, v_{t+1})$, only particles within a distance of d_{max} and an angular disparity within δ survive through to the next iteration. In our case, only particles within 60 degrees to either side of the GPS segment were retained.

IV. Integrating Multiple Models

Having defined various models that determine the road likelihood, there is the issue of choosing the appropriate model. We adapt mixed-state tracking techniques [6] to probabilistically detect and switch to the most dominant measurement model at any given time. We define an extended state for each particle to be

$$\mathbf{X}_i = (\mathbf{x}_i, O_i) \tag{3}$$

where $\mathbf{x}_i$ is as defined earlier and $O \in \{1..N_o\}$ is a discrete variable labeling one of N_o observation models. Thus O_i determines which observation model to use in the measurement phase for particle s_i. We also define a state transition probability matrix T which is an adjacency matrix representation of the possible state transitions i.e. T_{pq} is the probability for a particle to change state from p to q. In order to integrate mixed-state models into the particle filtering framework, it is sufficient to split the sampling process of every iteration into two separate phases. In the first phase, the state transition probabilities are sampled from to generate a new observation model density for the particles. The subsequent sampling phase is the same as described previously where particles with higher weights survive while the others are eliminated. In the update phase, the observation model used to measure the reliability of particle s_i^t depends on the value of O_i^t.

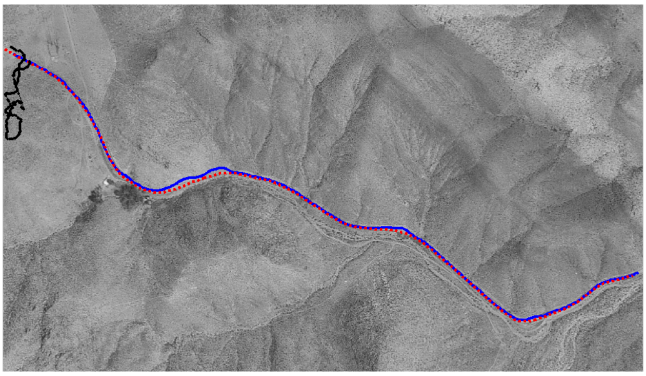

Fig. 2. GPS plot (dotted red) and tracked path (solid blue) through the desert. The darker curve shows the output of conventional JetStream using the luminance gradient.

The formal steps used in the particle filter are described below. We begin with S^{t-1} of N particles $s_i^{t-1} = \{< \mathbf{x}_i^{t-1}, O_i^{t-1}, w_i^{t-1} > | i = 1, ...N\}$ in every iteration followed by:

1) *Sampling:* Construct the n^{th} of N new samples according to the following two steps:
   - *Sample transition probabilities:* Sample from $P(O_i^t = q | O_i^{t-1} = p) = T_{pq}$ to find O_i^t for each s_i.
   - *Sample process density:* Sample from S^{t-1} based on w^{t-1} by generating a random number j with probability proportional to w_j^{t-1} and setting $s_n'^{(t-1)} = s_j^{t-1}$.
2) *Prediction:* We apply our dynamics to each sampled particle as governed by equation (1).
3) *Update:* The likelihood for this particle is computed according to the measurement model specified by O_i. Weights are updated in terms of the latest image data Z^t.

In order to estimate a single most probable position that the vehicle could be in after every time-step, a two-pronged strategy is adopted of first computing the dominant model $\widehat{O}^t$ in force, and then calculating the weighted mean of only those particles in that observation state. $\widehat{O}^t$ is computed according to

$$\widehat{O}^t = \arg\max_j \sum_{i \in \Upsilon_j} w_i^t \text{ where}$$

$$\Upsilon_j = \{i | \mathbf{X}_i^t = (\mathbf{x}_i^t, j)\} \tag{4}$$

V. Results

Fig. 3. Comparison of single-mode and mixed-mode tracking using models M_1 (dark roads) and M_2 (bright roads). Solid yellow indicates that the tracker is in mode M_1 and dashed blue denotes M_2. Notice how tracking only with M_2 causes mistracking on the darker roads.

We show the result of localization on both urban and off-road environments, currently using only aerial images and GPS data. The particle filter was initialized to use 1000 particles - though it is very robust with less than half that number - and first distributed around the starting point. The choice of Gabor filter scales depend on the resolution of the aerial imagery, and for the publicly available 1-meter resolution photos that we used, $\lambda_2 = 10$ was a fair approximation for the width of most roads. To detect the road edges and sidewalks, we used $\lambda_1 = 1$.

To compare our spatial tracker with the pure JetStream approach, we ran the particle filter on a 1.2 mile track obtained during a drive through the southern California desert. After verifying that the Navcom SF-2050G DGPS receiver was accurate to a couple of meters, the particle filter was run similar to JetStream without integrating any of the sensor data. This also enabled us to quantify deviation from ground truth. Shown in figure 2 is a plot of the GPS track (dotted red) on one segment and the estimated path (blue) of our tracker. The dark wayward curve shows the output of the conventional edge tracking method as used by JetStream. Due to the absence of strong edge cues, JetStream does not track correctly beyond a few meters. By extracting the local texture information using Gabor filters, our algorithm does significantly better. The mean distance error with the GPS curve on this run was 3.24 meters and standard deviation was 3.1m. On another similarly curvy segment, the mean was 1.4m and standard deviation was 1m. Estimating the road width to be about 10m from the aerial image and the GPS data itself to be accurate only to a couple of meters, we claim that our image processing alone is robust to handle significant drop-outs on the GPS due to LOS issues.

Figure 3 shows a suburban neighborhood comparing single- and mixed-mode tracking. The roads are either dark or bright, with abrupt transitions between them. To track all possible roads without GPS is not practical as there is a lot of clutter. The comparatively low cost Garmin GPS 16 used for this example was especially unreliable in such environments with accuracy as bad as 25 meters in some places. In addition to correcting the GPS path, our tracker automatically switches to the appropriate model. The yellow sections indicate when the tracker is in model M_1 looking for dark roads, while blue denotes that M_2 is dominant at that point. The transition matrix used was $T = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$ with a slightly higher probability enforced for each particle to remain in the current mode. The value of d_{max} was set to 20. Single-mode tracking using only M_2 causes mistracking to occur on the darker roads.

Fig. 4. Corrected GPS positions (solid blue) at intersections. GPS estimates (dotted red) were especially inaccurate at intersections.

Figure 4 shows zoomed in regions of corners where the GPS (red) tracks have been corrected by our contour extraction method. Figure 5 shows a short GPS run in dotted red. This is difficult both in terms of highly inaccurate GPS readings at the corners, as well as the presence of shadows and trees that mask the road in some places. While GPS-driven dynamics simply do not work correctly in this scenario, likelihood-driven dynamics (blue) is able to trace the road for the entire length of the segment. The particles have enough momentum to keep following the strongest road likelihood without changing direction at every wiggle in the GPS data.

Fig. 5. Using measurement likelihood-driven dynamics and model-switching (not shown) with highly inaccurate GPS (red) positions. Solid blue curve shows corrected path

Illustrated in figure 6 is a typical scenario of what happens when a GPS receiver loses signal due to over-passes or tunnels. There is a sudden glitch in the GPS outputs, as points veer off to the sides before homing in on the actual position again. Likelihood-driven dynamics can easily handle such situations as shown in the figure. This situation might also call for some image processing hacks that detect features orthogonal to the road, but that was not required in this case as particles were distributed far enough along the road to overcome negative filter responses under the over-pass.

VI. CONCLUSION

In this work, we have demonstrated techniques to correct erroneous GPS information for the purpose of vehicle localization. In contrast to map-matching approaches that use digital road-maps, we demonstrate our algorithm on aerial images of diverse environments. We use a combination of image processing and probabilistic methods to make an inference about the most likely road that the vehicle is traveling on, based on explicitly defined road models. Detecting and switching to the correct model is done automatically by the mixed-state tracker. Results are shown on a range of images by correcting inaccurate GPS position estimates. We claim that this kind of localization maybe a useful precursor to any on-the-road path-planning algorithm for intelligent vehicles.

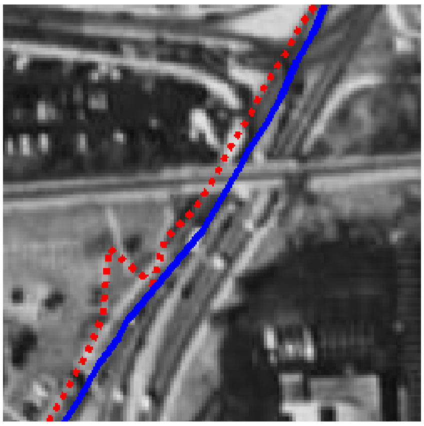

Fig. 6. Using measurement likelihood-driven dynamics to handle characteristic glitches in GPS data caused by over-passes and bridges

One short-coming of the weighted mean estimate of particle locations is that it does not take into account the distribution - which could possibly be clustered over several different roads. This happens most noticeably at intersections where GPS data seemed most unreliable. Clustering algorithms could be used to track multiple peaks in such situations. Using GPS-driven or likelihood-driven dynamics in isolation does not seem robust for long runs. It would be interesting to see the effects of adding the dynamics for each particle as another mode in the mixed-state tracker. This would allow some particles to follow the GPS curve, while other particles would follow the most likely road. Intuitively this seems more robust.

Future work includes integrating information from on-board sensors such as a camera and laser. With aerial images and GPS alone, our algorithm can identify the possible road that the vehicle is on. The use of additional sensors would allow us to correlate the on-board view with the aerial one, giving information about the position and orientation of the vehicle within the road. This can be useful in multiple lane roads or intersections. In off-road environments shown in figure 2, it would be very useful to complement aerial imagery with elevation data sets also.

VII. Acknowledgments

This work was supported by a grant from the National Institute of Standards and Technology. The authors are grateful to D. van Gogh at Caltech for desert road GPS data.

References

[1] M. Najjar and Ph. Bonnifait, "A Roadmap matching method for precise vehicle localization using Belief Theory and Kalman filtering," in *Proc. International Conference on Advanced Robotics*, 2003 .

[2] P. Perez, A. Blake, and M. Gangnet, "JetStream: Probabilistic contour extraction with particles," *International Conference on Computer Vision*, 2001.

[3] C. Frueh and A. Zakhor, "3D Model Generation for Cities Using Aerial Photographs and Ground Level Laser Scans", *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[4] M. Bicego, S. Dalfini and V. Murrino, "Extraction of geographical entities from aerial images," , *International Conference on Image Processing*, 2003.

[5] A. Talukder and D. Casasent, "Multiscale Gabor Wavelet Fusion For Edge Detection in Microscopy Images", *SPIE*, Vol. 3391, pp. 336-347, April 1998.

[6] M. Isard and A. Blake, "A mixed-state CONDENSATION tracker with automatic model-switching," *Proceedings of the* 6^{th} *International Conference on Computer Vision*, 1998.

[7] D. Fox, "Adapting the Sample Size in Particle Filters Through KLD-Sampling," *International Journal of Robotics Research (IJRR)*, 2003.

[8] C. Frueh and A. Zakhor, "Constructing 3D City Models by Merging Ground-Based and Airborne Views", *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[9] F. Dellaert, W. Burgard, D. Fox, and S. Thrun, "Using the Condensation Algorithm for Robust, Vision-based Mobile Robot Localization," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1999.

[10] A. Aboshosha, A. Zell, "Robust Mapping and Path Planning for Indoor Robots based on Sensor Integration of Sonar and a 2D Laser Range Finder," *IEEE 7th International Conference on Intelligent Engineering Systems*, 2003.

[11] J. Borenstein, H.R. Everett, L. Feng, " 'Where am I?' Sensors and Methods for Mobile Robot Positioning," *Technical Report, The University of Michigan*, 1996

[12] G. Atanas and P. Allen, "Vision for Mobile Robot Localization in Urban Environments," *Int. Conf. Intelligent Robots and Systems*, 2002.

[13] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," *Technical Report, Department of Comp. Sc. and Engg., Univ. of North Carolina at Chapel Hill*, 2002.

[14] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *International Journal of Computer Vision*, 29(1), pp. 5–28,1998

[15] S. Thrun, "Particle Filters in robotics," *In Proceedings of the 17th Annual Conference on Uncertainty in AI*, 2002.

[16] M.-F. Auclair-Fortier, D. Ziou, C. Armenakis, and S. Wang, "Survey of work on road extraction in aerial and satellite images," *Technical Report 247, Département de mathématiques et d'informatique, Université de Sherbrooke*, 2000.

[17] S. Syed and M.E. Cannon, "Fuzzy Logic Based-Map Matching Algorithm for Vehicle Navigation System in Urban Canyons," *ION National Technical Meeting, San Diego, CA*, 2004.

[18] M. Spengler and B. Schiele, "Towards robust multi-cue integration for visual tracking," *Machine Vision and Applications*, 14:50158, 2003.