2010 IEEE Intelligent Vehicles Symposium
University of California, San Diego, CA, USA
June 21-24, 2010



# Combining Statistical Hough Transform and Particle Filter for Robust Lane Detection and Tracking

Guoliang Liu, Florentin Wörgötter and Irene Markelić
Bernstein Center for Computational Neuroscience
University of Göttingen, Germany
{liu, worgott, irene}@bccn-goettingen.de

*Abstract*— **Lane detection and tracking is still a challenging task. Here, we combine the recently introduced Statistical Hough transform (SHT) with a Particle Filter (PF) and show its application for robust lane tracking. SHT improves the standard Hough transform (HT) which was shown to work well for lane detection. We use the local descriptors of the SHT as measurement for the PF, and show how a new three kernel density based observation model can be modeled based on the SHT and used with the PF. The application of the former becomes feasible by the reduced computations achieved with the tracking algorithm. We demonstrate the use of the resulting algorithm for lane detection and tracking by applying it to images freed from the perspective effect achieved by applying Inverse Perspective Mapping (IPM). The presented results show the robustness of the presented algorithm.**

## I. Introduction

Vision based street lane detection and tracking is an important factor for Driver Assistance Systems, which can reduce the risk of car accidents. Various lane detection and tracking methods have been proposed [1], [2], [3], [4], [5], but situations like occlusions, strongly differing illuminations, and unmarked or partly marked lanes are still challenging [4], [6].

In this paper, we combine the recently introduced Statistical Hough transform (SHT) [7] with a Particle Filter (PF) [8], [6] and show its application for robust lane tracking. The SHT overcomes shortcomings of the standard Hough transform (HT), which was proved to be a powerful tool for lane detection [5], [3], [6]. Although the standard HT is easy to implement and gives robust results for dashed lane marking, it is usually applied to edge images [5], [3], and imperfections in the edge detectors cause errors in the accumulator histogram of the standard HT. In addition, the limited number of edge pixels causes the histogram to be sparse [7], which makes it non–trivial to find appropriate peaks in it, which indicate lines in an image. SHT is a multi–kernel density based transform, which in contrast to the standard HT, uses all available pixels and their gradients as observation data to give a continuous probability distribution of the HT variables. A similar idea was presented in [9], where an elliptical–Gaussian kernel was used to model the probability distribution of HT variables, however, the edge detector is still required.

This work has been supported by European Commission under project FP6-IST-FET (DRIVSCO).

We propose to combine the SHT with a PF for lane tracking which we find to be mutually beneficial. First, due to the statistical nature of both SHT and PF, the PF can use local descriptors of SHT as measurement, and a new multi–kernel density based observation model can be modeled from SHT. Second, the tracking algorithm reduces the necessary computations. To demonstrate its use for lane detection and tracking, we implement a three kernel density based observation model for both SHT and PF, and apply it to images freed from the perspective effect using a transformation known as Inverse Perspective Mapping (IPM) [1], [10]. The whole algorithm can be seen in Fig. 4.

We test our algorithm on the DRIVSCO dataset [11], which is a freely available collection of real driving images, containing various scenes, including occluded lanes, blurred images and unmarked or partly marked streets. Although we are currently using a simple straight line model, the experimental results show that our tracking algorithm can successfully handle these difficult situations. More complex lane models, such as a parabolic [12] or a circular model [13], can also be used within our suggested framework.

Our algorithm comprises two steps, first an initial detection of the lanes, and second their subsequent tracking. Thus, the paper is organized as follows: Section II introduces the lane detection algorithm including a description of the IPM transform, our lane model, and the SHT algorithm. Section III introduces the lane tracking algorithm using a Particle Filter with our new observation model from the SHT. Results and analysis are given in Section IV and the work is concluded in Section V.

## II. Lane detection using Statistical Hough Transform

As additional constraint to our lane model introduced in Section II-D, we assume lanes are parallel to each other. But this information is lost in the original images due to the perspective effect, e.g. see Fig. 1a. We here recover it using Inverse Perspective Mapping (IPM) [10], [1], which under a flat ground assumption, transforms the image to a top view of the scene. As shown in Fig. 1b, the parallel lane markings appear parallel in the IPM image. Thus, we first apply IPM to the original image, and then run SHT on the resulting IPM image for lane detection.

978-1-4244-7868-2/10/$26.00 ©2010 IEEE

(a) (b)

Fig. 1: (a) An image from the DRIVSCO dataset. Parallel lane markings are not parallel. (b) the transformed image after applying IPM shows a top-view of the scene and the lane markers appear nearly parallel.

### A. IPM transform

IPM is a well known algorithm [10], [1], which can remove the perspective effect, based on the flat ground hypothesis with known extrinsic and intrinsic parameters of the camera. It remaps pixels from the original image to the other image that has a different coordinate system. This remapping procedure can be done by a fast lookup table with a distortion compensation [10]. As the resolution for near and further object is different in the original image, an interpolation process is needed in the IPM algorithm. The resulting image is shown in Fig. 1b, which is a top view image of scene with a cubic interpolation. The lanes looks parallel and the width between lanes becomes near constant.

### B. Lane Model

We assume that the lanes nearby the car appear to be straight, thus we use a simple straight line equation as our lane model, which defined in:

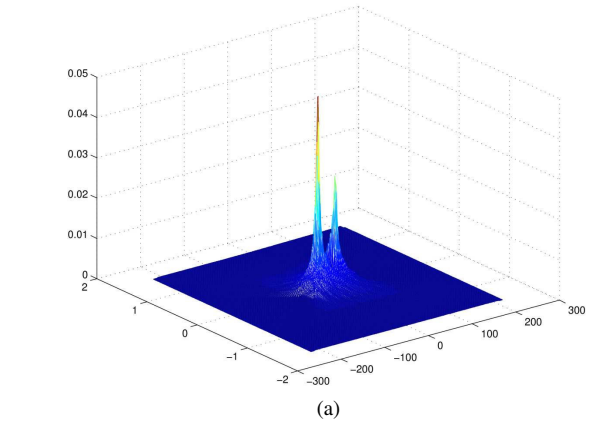
$$\rho = x\cos\theta + y\sin\theta, \tag{1}$$

where $x$ and $y$ correspond to the horizontal and vertical image coordinates, and $\rho$ and $\theta$ are the line parameters.

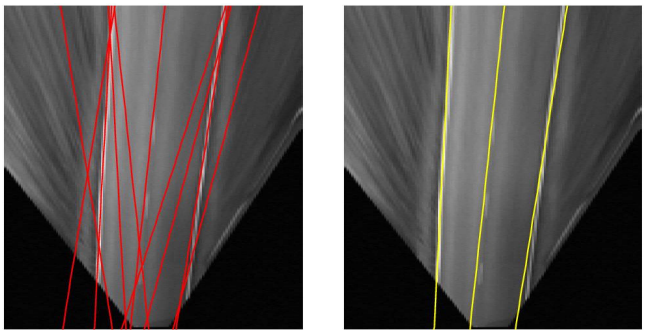
### C. Statistical Hough Transform

The Statistical Hough transform (SHT) introduced in [7], unlike the standard Hough transform, works on intensity images and uses a multiple kernel density to describe the distribution of the Hough variables $(\rho,\theta)$, thus no edge preprocessing is required.

In the SHT algorithm, the distribution of Hough variables is determined by the position of each pixel in the image given by $x_i$ and $y_i$, and the orientation of the pixel $\theta_i$, where $i \in [0,1,\ldots,N]$ and $N$ denotes the total number of pixels in a given image. We refer to the observation space for these three features $x_i$, $y_i$, and $\theta_i$ as $Q_{xy\theta}$. Given the latter, the distribution of Hough variables in SHT is $p(\rho,\theta|Q_{xy\theta})$, and we use Gaussian kernels in our experiment to model this distribution.




(a)



(b) (c)

Fig. 2: (a) Statistical Hough transform histogram, (b) the red color lanes correspond to the top 10 peaks in the SHT histogram, (c) the yellow color lanes are lane detection results using a known lane model.

*1) Gaussian Kernel Density Model:* The probability distribution $p(\rho,\theta,x,y|Q_{xy\theta})$ can be written according to Bayes rule:

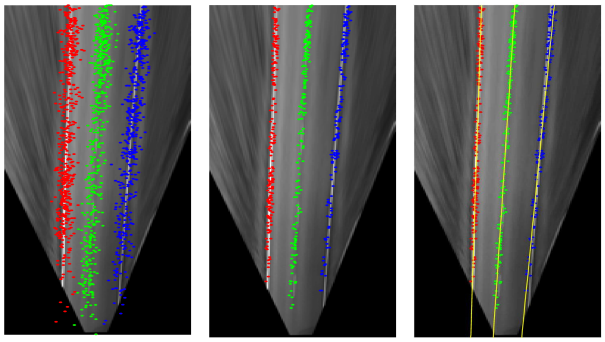
$$p(\rho,\theta,x,y|Q_{xy\theta}) = p(\rho|x,y,\theta,Q_{xy\theta}) \cdot p(x,y,\theta|Q_{xy\theta}) \tag{2}$$

In (2), the first probability $p(\rho|x,y,\theta,Q_{xy\theta})$ is determined by (1), and the second probability $p(x,y,\theta|Q_{xy\theta})$ can be modeled by a Gaussian kernel density function, thus (2) becomes:

$$p(\rho,\theta,x,y|Q_{xy\theta}) = \delta(\rho - x\cos\theta - y\sin\theta)\frac{1}{N}\sum_i K_x K_y K_\theta, \tag{3}$$

where $\delta$ is the Dirac delta function, $K_x = \mathcal{N}(x_i,\sigma_{x_i}^2)$, $K_y = \mathcal{N}(y_i,\sigma_{y_i}^2)$ and $K_\theta = \mathcal{N}(\theta_i,\sigma_{\theta_i}^2)$ are Gaussian kernels, $\sigma_{x_i}^2$, $\sigma_{y_i}^2$ and $\sigma_{\theta_i}^2$ are variances of $x_i$, $y_i$, and $\theta_i$, respectively. The distribution $p(\rho,\theta|Q_{xy\theta})$ can be obtained by integrating (3) over $(x,y)$:

$$p(\rho,\theta|Q_{xy\theta}) = \frac{1}{N}\sum_i K_\theta \cdot G_i(\rho,\theta), \tag{4}$$



(a) (b) (c)

Fig. 3: (a) sampling particles, (b) resampling the weighted particles using Stratified Resampling, (c) update the parameters of the lane model.

where

$$G_i(\rho,\theta) = \frac{1}{\sqrt{2\pi(\sigma_{x_i}^2 cos^2\theta + \sigma_{y_i}^2 sin^2\theta)}} exp\left(\frac{-(\rho - x_i cos\theta - y_i sin\theta)^2}{2(\sigma_{x_i}^2 cos^2\theta + \sigma_{y_i}^2 sin^2\theta))}\right) \quad (5)$$

The detailed description of (5) can be found in paper [7].

*2) Statistical Hough Transform Algorithm:* The algorithm for computing SHT is summarized as follows:

1. Calculate the orientation of the image gradient for each image pixel $\theta_i$ and its magnitude $\Delta I_i$.
2. Compute the variance of the observations: $\sigma_{x_i}^2$, $\sigma_{y_i}^2$ and $\sigma_{\theta_i}^2$. In our experiment, we set $\sigma_{x_i}^2 = 1$, $\sigma_{y_i}^2 = 1$ and $\sigma_{\theta_i}^2 = \sigma^2/\Delta I_i^2$, where $\sigma^2 = 1$. Here we can see that the bandwidth of kernel $K_\theta$ depends on the magnitude of the gradient.
3. Build a 2D histogram based on a fine grid determined by $\theta \in [-\pi/2, \pi/2]$ and $\rho \in [-\rho_p, \rho_p]$, where $\rho_p = \frac{w}{2}cos(atan(\frac{h}{w})) + \frac{h}{2}sin(atan(\frac{h}{w}))$, and $w$ and $h$ are the width and height of the image, respectively. Then compute the probability for every grid entry $(\theta, \rho)$ according to (4).
4. Find the peaks in the histogram by a threshold $T$. We set $T = 0.1$ in the experiment.

The result SHT histogram is shown in Fig. 2a. After thresholding it, we select the ten entries with highest probability which correspond to the ten potential lanes as shown in Fig. 2b.

### D. Lane Detection with SHT

To further specify which of these lines obtained from applying the SHT algorithm indeed corresponds to the correct lanes, we check the following additional constraints:

- Street lanes are nearly parallel to each other, such that $\theta_l \approx \theta_r \approx \theta_m$, where $\theta_l$, $\theta_r$ and $\theta_m$ are the gradient angle of left, right and middle lane respectively.
- The widths between parallel lanes are nearly equal: $|D_{lm}| \approx |D_{rm}|$, where $|D_{lm}|$ is the distance between left and middle lane, and $|D_{rm}|$ is the distance between right and middle lane.

Based on these assumptions, we obtain lane detection results as shown in Fig. 2c.

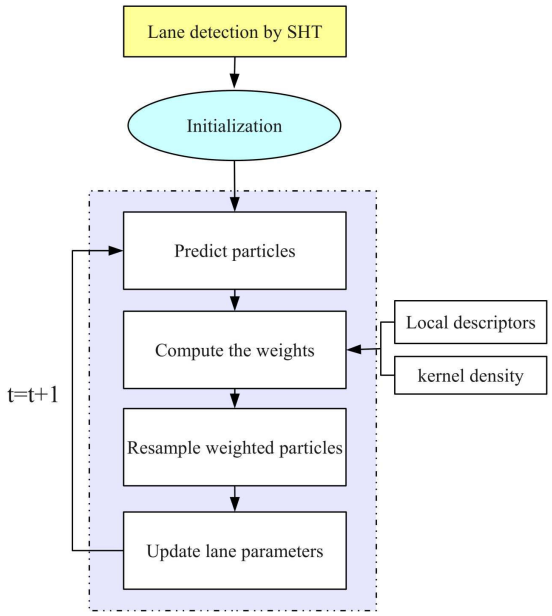



Fig. 4: The flow chart of our lane detection and tracking algorithm.

## III. Lane Tracking using the Particle Filter

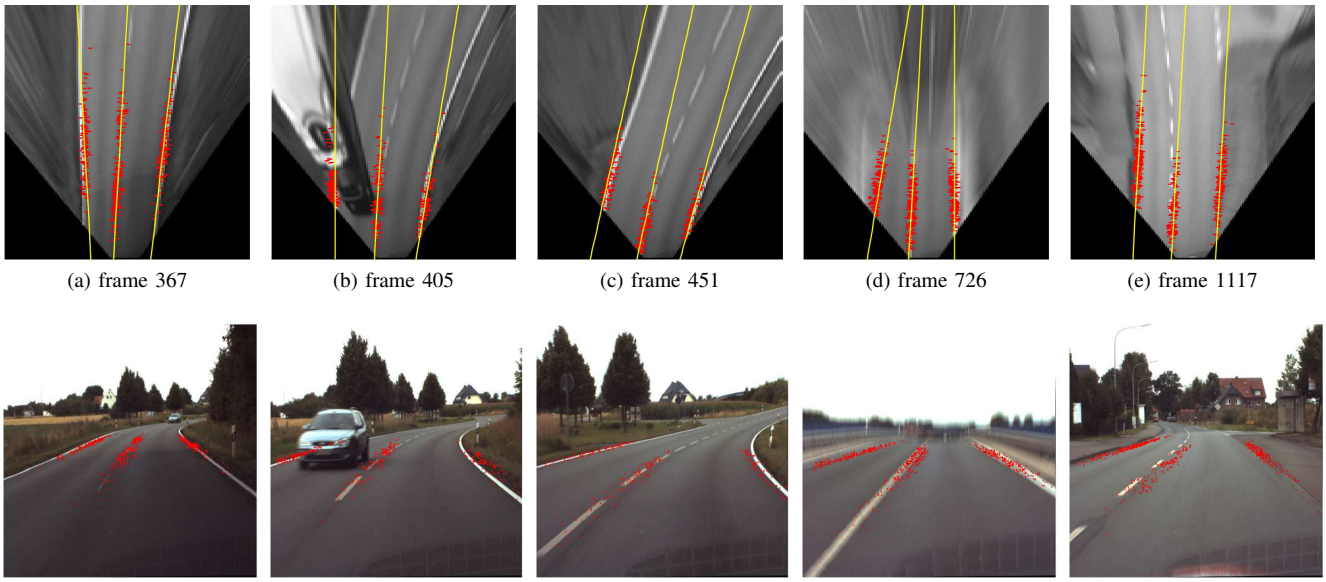
The SHT is computational expensive, as it works on all pixels of an image, thus it is not suitable to run it on every frame. After initial detection based on SHT, we therefore use the Particle Filter to track the detected lanes and to update the parameters of the lane model, which greatly reduces the computational time.

### A. Particle Filter

In paper [6], the author used a Particle Filter to track the control points of cubic spline. A similar idea is used here, however, we track pixels that might belong to a lane, which is represented by the particles of the Particle Filter.

In the Particle Filter algorithm, first we need to initialize the particles set, and then predict the particles in the current time frame. Finally we have to resample the particles based on their weights, which uses the observation model. The algorithm is summarized in Fig. 4. In our experiment, there are multi–lanes on the road. For each lane, the lane model is different, thus we need to initialize, sample and resample particles separately for each lane model.

*1) Initialization of the Particles:* After the SHT ran on the first frame, the parameters of lane models can be obtained, but we do not know where the lane markers are. The particles



(a) frame 367 (b) frame 405 (c) frame 451 (d) frame 726 (e) frame 1117

(f) frame 367 (g) frame 405 (h) frame 451 (i) frame 726 (j) frame 1117

Fig. 5: Tracking results from different scenes in consecutive frames. Top: particles in the IPM image, bottom: particles projected back to the original image. (a) and (f) dashed lines, (b) and (g) occluded by car, (c) and (h) non–flat ground plane, (d) and (i) blurred image, (e) and (j) partly marked lanes.

in the first step are selected on the straight lanes, which are defined by (1), then the state vector can be defined by $X_t = (X_t^1, X_t^2, \cdots, X_t^n)$ where $X_t^i = [x_t^i, y_t^i]^T$ is the 2D coordinate of particle $i$ in the image, $n$ is the number of particles.

*2) State prediction:* Assuming the change of lane boundary positions for two consecutive frames is small, a normal distribution can be used to model the state transition of particles as:

$$p(X_t|X_{t-1}) = \mathcal{N}(AX_{t-1}, \Sigma), \tag{6}$$

where function $\mathcal{N}$ is a normal distribution, the matrix $A$ is the identity matrix as we assume smooth changes of the lane boundaries, and $\Sigma$ is the covariance, which handles the difference of lane boundaries between two consecutive frames.

*3) Observation Model:* The observation space $Q_{xy\theta}$ in the SHT describes the local feature in the image. It also can be used in the Particle Filter. For the $i_{th}$ particle, the measurement is $z_t^i = (x_t^i, y_t^i, \theta_t^i)$. The observation model can be derived from (4):

$$p(z_t^i|X_t^i) = K_{\theta_t} \cdot G_i(\rho_t, \theta_t), \tag{7}$$

where $(\rho_t, \theta_t)$ are lane parameters of the current frame which are unknown, but we assume that the lane boundaries smoothly change between consecutive frames, thus $(\rho_t, \theta_t) \approx (\rho_{t-1}, \theta_{t-1})$. The observation model (7) becomes:

$$p(z_t^i|X_t^i) = K_{\theta_{t-1}} \cdot G_i(\rho_{t-1}, \theta_{t-1}) \tag{8}$$

*4) Resampling:* Stratified Resampling [8] is used to resample particles based on their weights. We use it because Stratified Resampling has a lower sampling variance, and is suitable to track multiple hypotheses [14]. For every particle, we can compute its weight using the observation model (8):

$$w_t^i = \eta\, p(z_t^i|X_t^i), \tag{9}$$

where $\eta$ is a normalization factor, which makes sure that the sum of weights is one. After resampling, those particles that have high weight will be kept [14], and the others that have lower weight will be removed from the particle set $X_t$.

### B. Tracking

The parameters of the lane model are updated frame by frame. We run a simplified version of SHT on the particle sets to update the parameters of the lane model. Due to the smooth change assumption of the lane boundary between two consecutive frames, we define the fine grid in the SHT histogram as $[\rho_{t-1} - \delta\rho, \rho_{t-1} + \delta\rho]$ and $[\theta_{t-1} - \delta\theta, \theta_{t-1} + \delta\theta]$, where $\delta\rho = 10$ and $\delta\theta = 10\pi\backslash 180$ in our experiment (see Section IV). As before, the new lane model parameters can be obtained by detecting the peak in this histogram.

## IV. Results

To demonstrate the performance of our algorithm, we apply it to the DRIVSCO dataset which contains a variety of challenging situations like high-curvature roads, partly marked and occluded lanes, etc. Our algorithm succeeds in tracking 837 images of this dataset, showing its robustness with respect to the difficult situations. Some tracking results are shown in Fig. 5 and a movie of the entire processed sequence is available at [15]. The upper part in Fig. 5 shows the lane tracking results in the IPM images (particles in red and the assigned line in yellow), and the lower part shows that the particles are back projected onto the original image,

mainly for the readers convenience. Note that the particles appear unevenly distributed in the lower part of original image, i.e. it seems like there are few particles close to the image bottom but clusters towards the middle of the image. This is not really the case, as can be seen from the IPM images, but induced by the perspective effect.

In Fig. 5b, the left lane is occluded by the car, however, the lane hypothesis given by the particles correctly indicates the lane. This is possible because the SHT relies on local observations of pixel position and gradient. If one of these is absent, the SHT probabilities for the Hough variables can still be computed (and thus particle weights determined) based on the remaining observations. Concerning Fig. 5b, this means that although no or random gradient information is available on the part where the lane is occluded by the car, the particle position will be considered as main factor on weight. The same reason holds true for the blurred image as shown in Fig. 5d. Similarly if the tracked lane is dashed, which is the case for all presented results but Fig. 5d, the particles that are between two line segments, and thus do not have access to reliable gradient information, can still contribute to the lane hypothesis via the position information. In addition, our tracking algorithm can also handle the unmarked case, as shown in Fig. 5e.

The SHT is computationally expensive, but our idea to combine it with a Particle Filter allows its use for practical applications. Currently it takes 1.1s per frame for tracking, however, we were interested in demonstrating the principle use of the proposed algorithm, and it was implemented in a non-optimized way in Matlab on a conventional INTEL Core 2 Duo ($2.2Hz$) machine. Because all particles can be computed independently from each other, the algorithm can easily be parallelized, and a more efficient implementation, e.g. using the GPU, will lead to a much faster frame rate. During the initial lane detection, the entire image frame must be processed which at the moment takes 700s. This is slow, however, occurs only once for the very first frame and can be improved by simply using the conventional HT for initializing the tracking process.

## V. Conclusion

In this paper we presented our idea to combine the recently introduced Statistical Hough Transform and the Particle Filter, and showed its application for lane detection and tracking on IPM images. The algorithm was tested on the DRIVSCO dataset and 837 frames were successfully tracked. Furthermore, the algorithm was very robust concerning challenging scenes.

We showed that the combination of the SHT with a PF is mutually beneficial in the sense that: a) the SHT descriptors can conveniently be used as measurements for the PF; b) the tracking makes the application of the expensive SHT feasible by considerably reducing computation time. As discussed before, the current implementation works at a frequency of $1Hz$, but is not optimized and implemented in Matlab. Since the algorithm can be parallelized, it is able to work at a much higher frequency. During the initialization, the advantage of the reduced computation due to the tracking is yet not available and using SHT to process an entire image is slow. This can be accelerated by using the original HT once for the very first frame.

To demonstrate our idea, we here used a straight lane model. Note that the algorithmic framework is not limited to this and more sophisticated models can be used equally well. In the future we will optimize the implementation of the algorithm and use it with a parabolic or a circular lane model.

## References


[1] M. Bertozzi and A. Broggi, "Real-time lane and obstacle detection on the gold system," in *Proc. IEEE Intelligent Vehicles Symposium*, Sept. 19–20, 1996, pp. 213–218.

[2] H. Loose, U. Franke, and C. Stiller, "Kalman particle filter for lane recognition on rural roads," in *Proc. IEEE Intelligent Vehicles Symposium*, June 3–5, 2009, pp. 60–65.

[3] K. Macek, B. Williams, S. Kolski, and R. Siegwart, "A lane detection vision module for driver assistance," in *Mechatronics & Robotics*, 2004.

[4] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, Mar. 2006.

[5] M. Aly, "Real time detection of lane markers in urban streets," in *Proc. IEEE Intelligent Vehicles Symposium*, June 4–6, 2008, pp. 7–12.

[6] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16–26, Mar. 2008.

[7] R. Dahyot, "Statistical hough transform," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 8, pp. 1502–1509, Aug. 2009.

[8] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *Proc. 4th International Symposium on Image and Signal Processing and Analysis ISPA 2005*, Sept. 15–17, 2005, pp. 64–69.

[9] A. F. F. Leandro and M. O. Manuel, "Real-time line detection through an improved hough transform voting scheme," *Pattern Recognition*, vol. 41, no. 1, pp. 299 – 314, 2008.

[10] T. Bergener and C. Bruckho, "Compensation of non-linear distortions in inverse-perspective mappings," Institut fr Neuroinformatik, Ruhr-Universit t Bochum, Tech. Rep., April 1999.

[11] http://www.mi.auckland.ac.nz/EISATS.

[12] X. Liu, Q. Song, and P. Li, "A parabolic detection algorithm based on kernel density estimation," in *ICIC (1)*, 2009, pp. 405–412.

[13] M. Nieto, L. Salgado, F. Jaureguizar, and J. Arrospide, "Robust multiple lane road modeling based on perspective analysis," in *ICIP*, 2008, pp. 2396–2399.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.

[15] http://www.youtube.com/watch?v=oXIOHRBvJRc.