Improving Estimation of Vehicle's Trajectory Using the Latest Global Positioning System With Kalman Filtering

Cesar Barrios and Yuichi Motai, *Member, IEEE*

***Abstract*—This paper proposes several extensive methods to predict the future location of an automobile. The goals of this paper are to find a more accurate way to predict the future location of an automobile by 3 s ahead, so that the prediction error can be greatly reduced with the innovative idea of merging global-positioning-system (GPS) data with geographic-information-system (GIS) data. The improvement starts by applying existing techniques to extrapolate the current GPS location. Comprehensive Kalman filters (KFs) are implemented to deal with inaccuracy in the different identified possible states an automobile could be found in, which are identified as constant locations, constant velocity, constant acceleration, and constant jerks. Then, the KFs are set up to be part of a interacting-multiple-model (IMM) system that provides the predicted future location of the automobile. To reduce the prediction error of the IMM setup, this paper imports an iterated geometrical error-detection method based on GIS data. The assumption that the automobile will remain on the road is made; therefore, the predictions of future locations that fall outside are corrected accordingly, making a great reduction to the prediction error. The actual experimental results validate our proposed system by reducing the prediction error to around half of what it would be without the use of GIS data.**

***Index Terms*—Geographic information system (GIS), global positioning system (GPS), Kalman filter (KF), trajectory prediction.**

I. Introduction

ACCURATELY predicting the future location of a vehicle is a very important and relatively difficult topic in intelligent transportation systems. It can be effectively used in obstacle-avoidance systems for vehicles or robots.

Many of the existing obstacle-avoidance systems currently being researched are limited to line-of-sight sensors, such as those described in [1]–[9], which uses sensors around the vehicles to detect nearby objects. For a long-range obstacle-avoidance system, other types of sensors need to be implemented, such as those presented in [10] and [11].

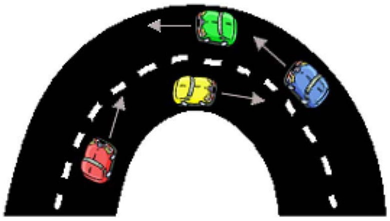

Fig. 1. "C" crossing.

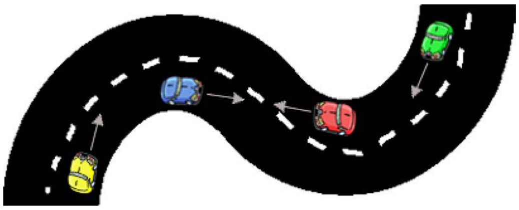

Fig. 2. "S" crossing.

Studies such as [10] or [11] investigate the option of using global-positioning-system (GPS) data collected from the different vehicles to predict the future location of each vehicle. The methods used to make these predictions are somewhat simple and do not give very accurate results in scenarios such as curves (see Figs. 1 and 2), where the estimated future position of the vehicles will not be a straight path.

It is clear from the current research that we need a more accurate way to predict the trajectory of vehicles in all different scenarios. This is where the Kalman filter (KF) comes into play. The KF has a long history of accurately predicting the future states of a moving object and has been applied to many different fields, which is why it has been chosen for this paper [12]–[15].

The contribution of this paper is to investigate the viable idea of using a geographic information system (GIS) to reduce the error in the prediction of the future location of an automobile, particularly during curves. The system implemented in this paper consists of an IMM with different KFs using a GPS to get the spatial information of the vehicle. There are a number of existing studies concerning the best methods to take spatial coordinates that fall outside of a defined road and to estimate where they would fall on an actual road, which is also known as map matching. For example, in [16]–[19], the authors go into a lot of detail to explain the different errors that need to be accounted for when using a GPS sensor (among others) and data for road maps (i.e., in a GIS) and how the GPS bias can be utilized to improve map-matching accuracy. Other studies

Manuscript received August 2, 2010; revised February 3, 2011; accepted March 29, 2011. Date of publication May 19, 2011; date of current version November 9, 2011. This work was supported in part by the United States Department of Transportation through the University of Vermont Transportation Research Center and in part by the National Science Foundation through the Faculty Early Career Development Program under NSF CAREER Award 1054333. The Associate Editor coordinating the review process for this paper was Dr. Antonios Tsourdos.

C. Barrios is with the University of Vermont, Burlington, VT 05405 USA (e-mail: cbarrios@uvm.edu).

Y. Motai is with the Virginia Commonwealth University, Richmond, VA 23284 USA (e-mail: ymotai@vcu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2011.2147670

such as [20] and [21] look into the problem of GPS outages and how the position of a vehicle can be estimated during the outage through the use of KF and map-matching techniques. This paper compares the experimental results of predictions done with and without our GIS-error-correction algorithm and does not consider the problem of GPS outages since other researchers are working solely on this issue.

This paper is based on the use of a GPS receiver to obtain location information and to be able to estimate the projected path for a vehicle. There are many factors that can degrade the GPS signal and thus affect its accuracy, but there are also some innovative ways of correcting these errors. The Holux GR-213 1-Hz GPS receiver used in this paper is wide-area augmentation system (WAAS) enabled.

The WAAS is a system developed for civil aviation by the Federal Aviation Administration in conjunction with the United States Department of Transportation. It is a nationwide differential GPS system where base stations with fixed receivers calculate and transmit the GPS error to the geostationary satellites in its view, which in turn broadcast the corrections that can be used by individual WAAS-capable GPS receivers. Its accuracy is less than 3 m 95% of the time, and our GPS receiver claims to have an accuracy value of less than 2.2 m [22].

Similar systems designed to predict a vehicle's trajectory implement the use of other types of sensors to be able to get an accurate estimation, but this paper looks into the possibility of using a commercially available inexpensive but accurate GPS receiver to do a similar task already implemented in some areas [12], [14], [15], [23]–[25], and it takes advantage of using a location-based system such as knowing where on a road map the vehicle is located.

To be able to predict a vehicle's future location, we used the KF. The KF is a set of mathematical equations that provides an efficient computational (recursive) method to estimate the future state of a process. The filter is very powerful because it supports the estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown [25]–[36].

The multiple-KF-model approach was chosen over one complex model because setting up multiple smaller models for each different scenario would be simpler than defining one complex model that can be accurate in many different scenarios. Each simple model is good for one specific set of conditions; therefore, several models need to be defined to be able to cover most, if not all, possible scenarios in which a vehicle can be found. For our setup, four models have been identified to cover most of the vehicles' behaviors: a vehicle that is not moving, a vehicle traveling at (CV), a vehicle traveling with constant acceleration (CA), and a vehicle traveling with a constant jerk (CJ) (i.e., a constant change in acceleration). These models provide a mathematical set of equations that can be used to predict the vehicle's future location after a set amount of time Δk.

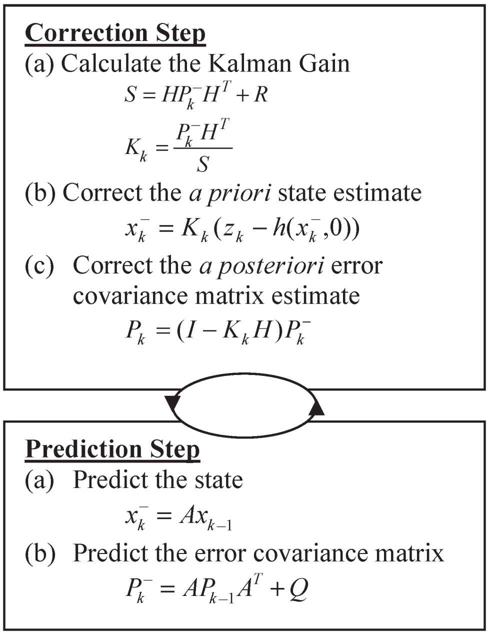
Correction Step
(a) Calculate the Kalman Gain
$S = HP_k^- H^T + R$
$K_k = \frac{P_k^- H^T}{S}$
(b) Correct the *a priori* state estimate
$x_k^- = K_k(z_k - h(x_k^-, 0))$
(c) Correct the *a posteriori* error covariance matrix estimate
$P_k = (I - K_k H)P_k^-$
Prediction Step
(a) Predict the state
$x_k^- = Ax_{k-1}$
(b) Predict the error covariance matrix
$P_k^- = AP_{k-1}A^T + Q$

Fig. 3. Kalman Filter.

This paper presents the trajectory estimation at 3 s ahead of time, which is based on the average human reaction time of 1.5 s to stop a vehicle [37]. We chose to look at 3 s ahead of time as a reference point that is double the reaction time of an average human being. In reality, this number will probably vary in relation to the speed and the type of the vehicle since a faster or heavier vehicle will need more time to slow down. The fastest data rate of the GPS receiver we used is 1 s $(\Delta k = 1)$, so that it is the rate the system will run at, which is set up to estimate the location of the vehicle 1 s later in time. To be able to obtain an estimation for the location of a vehicle 3 s later in time, we need to run the prediction steps of the KF system with Δk set to 3 s and use the IMM to obtain the prediction. This extra step to estimate the 3-s-ahead location adds very little run time to the overall system since it is only used to predict the location and no correct step is needed.

II. KF

The KF estimates a process by using a form of a feedback control loop. The filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements, and then, it repeats (see Fig. 3). In Fig. 3, the following notation holds:

x	state estimate
z	measurement data
A	Jacobian of the system model with respect to state
H	Jacobian of the measurement model
Q	process-noise covariance
R	measurement-noise covariance
K	Kalman gain
P	estimated error covariance
σ_p	prediction noise
σ_m	measurement noise

As such, the equations for the KF fall into two groups: the "prediction step" and the "correction step." The prediction-step equations are responsible for projecting forward (in time) the current state and error-covariance estimates to obtain the *a priori* estimates for the next time step. The correction-step equations are responsible for the feedback, i.e., for incorporating a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate.

For our system, the state vector for this system consists of two parameters obtained from the GPS sensor, with each one decomposed into its x and y components. The general form of the state-estimate matrix is shown as

$$x = \begin{bmatrix} x_v \\ v_v \end{bmatrix} = \begin{bmatrix} \text{Position of vehicle} \\ \text{Velocity of vehicle} \end{bmatrix}. \tag{1}$$

The elements of the state vector in (1) were selected to account for all the measurements available from the GPS sensor, and from them, we derived any other variables needed for the KF models. Keep in mind that each of the components of the state estimate in (1) has x and y components on it. Therefore, for every x_k represented in the equations, there will be x_{kx} and x_{ky}.

The error-covariance matrix is a data set that specifies the correlations in the observation errors between all possible pairs of vertical levels. The error covariance for each KF was approximated by running the filters on their own, but its value gets adjusted every 1 s in our setup.

The estimated error covariance P is used together with the Jacobian matrix H and the measurement-noise covariance R to calculate the Kalman gain K, as shown in Fig. 3. These values are given as follows:

$$P = \begin{bmatrix} x_v x_v & x_v v_v \\ v_v x_v & v_v v_v \end{bmatrix} \tag{2}$$

$$h(x, v) = \begin{bmatrix} x_v + v \\ v_v + v \end{bmatrix} \tag{3}$$

$$H = \left[\frac{\partial}{\partial x} h(x, 0)\right]_{\substack{x=x(k-1) \\ v=0}} \tag{4}$$

$$R = \sigma_m^2 \cdot \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}. \tag{5}$$

Once the Kalman gain K is calculated, the system looks at the measured data Z to correct the predicted position and also the covariance error. Since this system only obtains measurements from a GPS receiver, only locations, speed, and heading angles can be obtained; therefore, the other two parameters need to be calculated from the measured data. The acceleration is calculated from the velocity difference between the current and previous readings, and similarly, the jerk is calculated from the acceleration difference between the current and previous values. For this paper, instead of using the current speed and heading from the GPS sensor, the average speed parameter was also similarly derived from the location difference between the current and previous values.

Another important item to point out is that this paper does not look into solving the GPS measurement errors that are due to many factors. One of these error contributors is the "signal multipath" problem, where the signal reflects offly large objects. In this paper, we are assuming that these errors are minimal since the experiment is done in a very rural area. In addition, signal delays (i.e., ionosphere and troposphere delays) can cause the location readings from the GPS to bounce around and imply movement when the vehicle is not even moving. There are many error contributors to the GPS receivers, but we will assume them negligible in this paper to concentrate on the main objective of this system.

After the correction of the previously predicted values, the system is ready to predict the next position by using the state-vector equations. The filter estimates also the error covariance of the estimated location by using the Jacobian matrix A together with the process-noise covariance Q, where

$$A = \left[\frac{\partial}{\partial x} f(x, w)\right]_{\substack{x=x(k-1) \\ w=0}}. \tag{6}$$

To obtain an accurate prediction of the vehicle's future location, four adaptive prediction algorithms are defined to account for the possible scenarios. The state equations are very different between the models. The following four models account for most, if not all, possible situations in which a vehicle could be found.

1) The constant-location (CL) model is

$$\begin{aligned} x_v(k) &= x_v(k-1) + w \cdot \Delta k \\ v_v(k) &= 0. \end{aligned} \tag{7}$$

2) The CV model is

$$\begin{aligned} x_v(k) &= x_v(k-1) + (v_v(k-1) + w)\,\Delta k \\ v_v(k) &= v_v(k-1) + w. \end{aligned} \tag{8}$$

3) The CA model is

$$\begin{aligned} x_v(k) &= x_v(k-1) + v_v(k-1)\Delta k + \frac{1}{2}\left(a_v(k-1) + w\right)\Delta k^2 \\ v_v(k) &= v_v(k-1) + \left(a_v(k-1) + w\right)\Delta k. \end{aligned} \tag{9}$$

4) The CJ model is

$$\begin{aligned} x_v(k) &= x_v(k-1) + v_v(k-1)\Delta k + \frac{1}{2} a_v(k-1)\Delta k^2 \\ &\quad + \frac{1}{6}\left(j_v(k-1) + w\right)\Delta k^3 \\ v_v(k) &= v_v(k-1) + a_v(k-1)\Delta k \\ &\quad + \frac{1}{2}\left(j(k-1) + w\right)\Delta k^2. \end{aligned} \tag{10}$$

In (7)–(10), Δk represents the period of time passed; therefore, the variables at $k-1$ represent the data from one period of time ago. In our setup, the period of time is driven by the data rate of the GPS (i.e.,1 s). The process-noise covariance for each of models w is based on the constant term only. For example, for the CV model, the process-noise covariance is based on the velocity term only, and it can be derived from the measured data by applying the CV model to it.

Equations (7)–(10) represent the four states in which a vehicle can be found: at rest, moving at CV, moving at CA, or moving at a CJ. Each of these models consists of four state equations used to calculate each component of the state-estimate matrix defined in (1). These models are very important as they are the heart of the prediction system. They need to cover most, if not all, of the possible scenarios, or the predictions will contain more errors.

For more details on how to set-up a KF and a detailed explanation of all required mathematical equations, please refer to publications such as [25], [27], and [38].

III. IMM Estimation

Because the dynamics of automobiles can vary over time, we already defined the state equations (7)–(10) to capture the different states in which a vehicle can be found, but these independent state equations need to be merged to produce only one prediction.

There are several algorithms that exist to modify the stochastic information, and they are well known for their ability to automatically adapt the filter in real time to match any variation of the errors involved.

The interacting-multiple-model (IMM)-estimation algorithm calculates the probability of occurrence for each of the individual filters and uses that information to identify which of the filters will be predominant. This algorithm continues recalculating the probability for each iteration throughout the whole run, weighing the new probability values against the probability values calculated in the previous iteration.

The IMM filter calculates the probability of success of each model at every filter execution, providing a combined solution for the vehicle behavior. These probabilities are calculated according to a Markov model for the transition between maneuver states, as detailed in [28]. To implement the Markov model, it is assumed that, at each execution time, there is probability p_{ij} that the vehicle will make a transition from the model state i to state j. Equation (11) shows the transition matrix for the four defined KF models in Section II. This is given as follows:

$$p_{ij} = \begin{bmatrix} \mathrm{CL}\rightarrow\mathrm{CL} & \mathrm{CL}\rightarrow\mathrm{CV} & \mathrm{CL}\rightarrow\mathrm{CA} & \mathrm{CL}\rightarrow\mathrm{CJ} \\ \mathrm{CV}\rightarrow\mathrm{CL} & \mathrm{CV}\rightarrow\mathrm{CV} & \mathrm{CV}\rightarrow\mathrm{CA} & \mathrm{CV}\rightarrow\mathrm{CJ} \\ \mathrm{CA}\rightarrow\mathrm{CL} & \mathrm{CA}\rightarrow\mathrm{CV} & \mathrm{CA}\rightarrow\mathrm{CA} & \mathrm{CA}\rightarrow\mathrm{CJ} \\ \mathrm{CJ}\rightarrow\mathrm{CL} & \mathrm{CJ}\rightarrow\mathrm{CV} & \mathrm{CJ}\rightarrow\mathrm{CA} & \mathrm{CJ}\rightarrow\mathrm{CJ} \end{bmatrix}. \tag{11}$$

The IMM can be described as a recursive suboptimal algorithm that consists of five core steps:

Step 1) calculation of the mixing probabilities;
Step 2) mixing;
Step 3) mode-matched filtering;
Step 4) mode probability update;
Step 5) estimate and covariance combination.

As in any recursive system, the IMM algorithm needs to be initialized first before it can start its four-step recursion. The number of filters used is four.

Step 1) Calculation of the mixing probabilities. The probability-mixing calculation uses the transition matrix in (11) and the previous iteration model probabilities in (16) to compute the normalized mixing probabilities in (12). The mixing probabilities are recomputed each time the filter iterates before the mixing step. This is given as follows:

$$\lambda_k(i|j) = \frac{p_{ij}\lambda_{k-1}(i)}{\sum\limits_{i=1}^{N} p_{ij}\lambda_{k-1}(i)}. \tag{12}$$

Step 2) Mixing. The mixing probabilities are used to compute new initial conditions for each of the N filters, i.e., four in our case. The initial state vectors are formed as the weighted average of all the filter state vectors from the previous iteration (13). The error covariance corresponding to each of the new state vectors is computed as the weighted average of the previous iteration error covariance conditioned with the spread of the means as follows:

$$x^{0j}_{k-1} = \sum_{i=1}^{N} \lambda_{k-1}(i|j)\hat{x}^{i}_{k-1} \tag{13}$$

$$P^{0j}_{k-1} = \sum_{i=1}^{N} \lambda_{k-1}(i|j) \times \left\{P^{i}_{k-1} + \left[\hat{x}^{i}_{k-1} - \hat{x}^{0j}_{k-1}\right]\left[\hat{x}^{i}_{k-1} - \hat{x}^{0j}_{k-1}\right]^T\right\}. \tag{14}$$

Step 3) Mode-matched filtering. Using the calculated $\hat{x}^{0j}_{k-1}$ and P^{0j}_{k-1} values, the bank of four KFs produce outputs $\hat{x}^{j}_{k}$, the covariance matrix P^{j}_{k}, and the probability density function $f_n(z_k)$ for each filter (n) in (16), according to the equations for the KF. The covariance for each filter is represented by S_k in (15) and (18) and is given as

$$S_k = H \cdot P \cdot H^T \tag{15}$$

$$f_n(z_k) = \frac{1}{\sqrt{(2\pi)^{\frac{4}{2}}|S_k|}} e^{-\left(\frac{1}{2}\right)\cdot\left(V^T\cdot S^{-1}\cdot V\right)}. \tag{16}$$

Step 4) Mode probability update. Once the new initial conditions are computed, the filtering step (step 3) generates a new state vector, an error covariance, and a likelihood function for each of the filter models. The probability update step then computes the individual filter probability as the normalized product of the likelihood function and the corresponding mixing probability normalization factor. This is given as follows:

$$\lambda_k(j) = \frac{f_n(z_k)}{\sum\limits_{i=1}^{N} f_n(z_k)} \sum_{i=1}^{N} p_{ij}\lambda_{k-1}(i). \tag{17}$$

Step 5) Estimate and covariance combination. This step is used for output purposes only; it is not part of the algorithm recursions. This is given as follows:

$$\hat{x}_k = \sum_{j=1}^{N} \lambda_k^j \cdot \hat{x}_k^j \tag{18}$$

$$P_k = \sum_{i=1}^{N} \lambda_k^j \cdot \left\{ P_k^j + \left[\hat{x}_k^j - \hat{x}_k\right] \left[\hat{x}_k^j - \hat{x}_k\right]^T \right\}. \tag{19}$$

IV. GIS

A GIS is a system for capturing, storing, analyzing, and managing data and associated attributes, which are spatially referenced to the Earth. It is a tool that allows users to create interactive queries (i.e., user-created searches), analyze spatial information, edit data and maps, and present the results of all these operations. In this paper, we extracted the road information from the maps being used to display the location of a vehicle. It is not a very accurate map, but it is enough to demonstrate if the implementation of GIS information with the IMM system improves the prediction of the future location of the vehicle or not.

The idea of using GIS data to correct an invalid estimation came about by looking at simulations during curves. When the vehicle enters a turn, the prediction of its future locations is very erroneous, i.e., many times outside of a road. If the system had a way of knowing the direction of the road ahead and whether the estimated future location was on an actual road or not, it would be able to correct its estimation and improve its reliability. This is where the GIS comes into play, with the assumption that the vehicle will always remain on the road. It is also assumed that the driver is handling the vehicle properly and is awake for this GIS correction to be practical. These assumptions, although restrictive, still allow the correction to be useful in scenarios such as road intersections.

When a road is designed, the radius of curvature is known, but this information is not available with the GIS data; therefore, a new method is needed to be able to project the estimation outside of the road back to the road.

Because of the limitation of the mapping software used during this paper (i.e., MapPoint), the only available function to interact with GIS data was to check whether the specific location was on the road or not. A function that provided the distance from the current location to the nearest road would have worked a lot better, but it was not available in MapPoint.

To overcome the limitation aforementioned, a method to map the estimated future location outside of the road to an accurate location inside the road had to be designed. From the current GPS location, distance r and angle β shown in Fig. 4 are calculated. Angle β varies with the direction of the movement and is calculated from the east being 0 °. r is the distance between the current location and the estimated location. This is given in the following:

$$\text{count} = \frac{\text{circumference}}{\text{arc}} \tag{20}$$

$$\alpha = \frac{360 \text{ deg}}{\text{count}}. \tag{21}$$

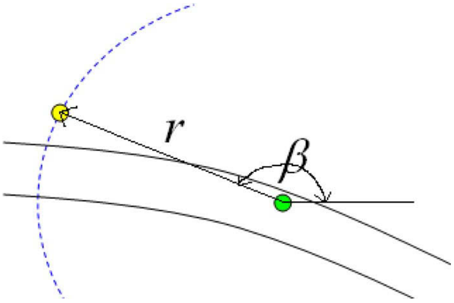

Fig. 4. Displaying parameters used in the method to estimate position on the road.

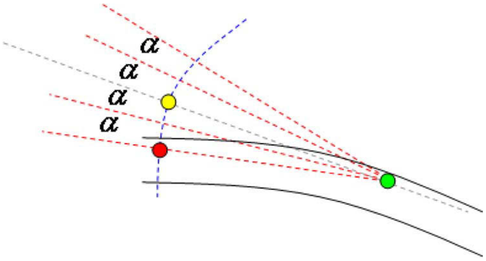

Fig. 5. Geometry used to map the estimated future location outside the road to a location inside the road.

Variable arc used in (20) is the predefined distance between the points in the circumference. The smaller this value is, the smaller the increments between check points in the circumference are and the more accurate the measurement will be. Due to the fact that the arc value is smaller, more points are needed to be checked; thus, it requires more central-processing-unit processing time. Therefore, for this paper, arc has a value of 0.6 m. This value was selected because the smallest road, even if only a one-way lane, cannot be less than 2-m wide. If we used a value bigger than 2 m, we could have the possibility of missing a road between check points; therefore, we chose a significantly smaller value. Angle α calculated in (21) is the actual angle increment needed to match the predefined arc distance on the circumference.

With angle β shown in Fig. 4 and angle α calculated in (21), we can start running through the checkpoints of the circumference. The estimated location is found at angle β, and since this estimated location cannot be very far from the actual road, we start checking from angle β. The system will check both the clockwise and counterclockwise increments of α until a point is found on the road. Fig. 5 provides a graphical view of the GIS error checking implemented. The clockwise and counterclockwise increments will continue to occur until either a road is found and a correction on the estimated future location is made or a maximum number of increments is reached and no correction is made. If a correction is made, the new estimated future location will still be the same distance away r; the only difference is its location coordinates.

In Fig. 6, we can see in MapPoint the current location in a green dot, the predicted future location in a yellow dot, and the GIS-corrected data in a red dot. The smaller red dots are the clockwise and counterclockwise increments aforementioned. Visually, in Fig. 6, the estimated future location is probably incorrect as there is no road in that location. Using GIS data to

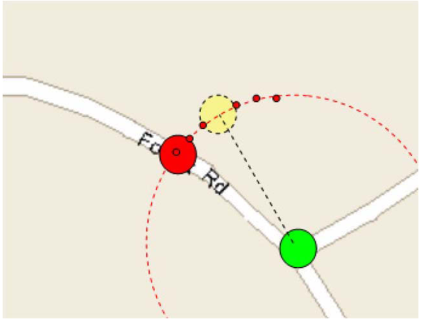

Fig. 6. GIS error correction in MapPoint.

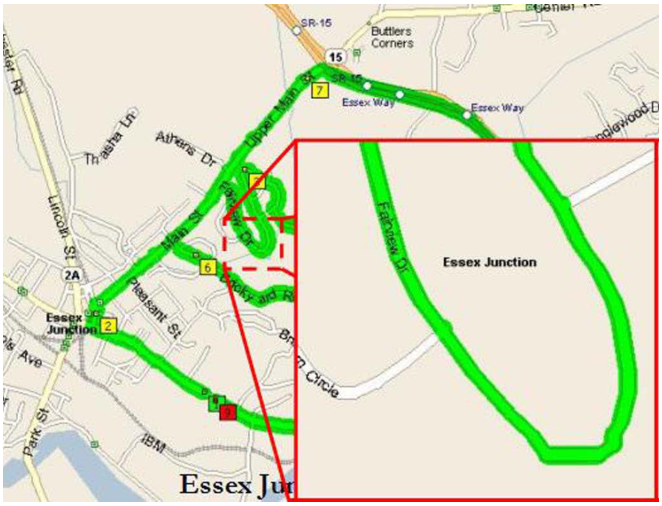

Fig. 7. Trajectory recorded in GPS log file, Essex Jct., Vermont, USA.

locate the road, we can adjust the predicted location to be on the road at the same distance away as the velocity will probably not change significantly under normal circumstances. The result is a more accurately predicted future location. This method seems to work well during curves, but as stated earlier, it requires several restrictive assumptions. Therefore, this system could only be useful as a part of a larger and more robust collision-avoidance system that took into account some of the scenarios not covered by our proposed method.

V. Experimental Results

The experimental setting for testing the models described in Section II needs a log file of GPS data that contains different scenarios, particularly those currently causing problems in the existing systems (see Figs. 1 and 2). Fig. 7 shows the trajectory recorded for this paper. It has many turns and contains various changes in speed and direction. Because we are trying to improve the trajectory estimation during curves, Fig. 7 also shows the curve selected for this paper.

The selected road curve is definitely a nice sharp turn that occurs at medium speeds (i.e., ∼60 km/h). We felt that this turn would be a good scenario to test our improvements on trajectory estimation.

The code was implemented in Microsoft Visual Basic 6 and Microsoft MapPoint 2004, allowing the software to display information in real time on the map as the vehicle moves. Being able to look at the estimated points on an actual map makes it easier to visually inspect and present the system.

TABLE I
Average 3 s Ahead Estimation Error

	CL	CV	CA	CJ
KF	14.9002	9.8786	7.0812	8.9952

Units are in meters. Used 21 data points for the selected curve.

TABLE II
Average Estimation Error

Estimated position	*1 sec ahead*	*3 sec ahead*
IMM	2.9056	8.7880
IMM with GIS	1.7834	4.8244

Units are in meters. Used 21 data points for the selected curve.

A. Implementation of the KFs

To be able to evaluate, the four KF models, i.e., KF-CL, KF-CV, KF-CA, and KF-CJ, had to be coded, tested, and tuned individually to get the most possible accurate estimations. It is given that one of these models will not be very accurate all the time on a real-time GPS log; therefore, in order to calibrate them individually, the GPS log for the full trajectory shown in Fig. 7 was used to calculate the measurement-noise covariance and also each of the process-noise covariance for the four models to exercise only one model at a time. To find the values for the process- and measurement-noise-covariance matrices, we smoothed out data using a moving average window to remove any outlier. The measurement-noise covariance was calculated for each of the filters by calculating the covariance of going frame by frame and calculating the error of the real data to fit into each of the KF models defined in Section II.

Once the filters were tuned, they were individually run through the different scenarios, and only the results for the data points in the selected curve were recorded in Table I.

Running the four filters together showed that, when one was very close to the real value, the other ones were not that accurate. In some instances, more than one filter was accurate, which was probably when speed changes or acceleration changes were very small. In other cases, none of the four filters was accurate at all, which was probably because of an abrupt change in direction or even in speed. The system reads data from the GPS every 1 s; therefore, it is possible, although not common at higher speeds, to have a big change occur during that 1 s, particularly in curves. For the most part, 1 s will not allow the speed and direction to change by a big amount (except in some lower speed scenarios such as at intersections when making a sharp turn), allowing the filters to estimate the next location somewhat accurately. The error calculated in Tables I–III are based from the actual GPS location. It is the distance between the estimated 3-s-ahead location and the actual GPS location 3 s later. Actual GPS errors are not accounted for in this paper; therefore, both the estimated future location and the actual GPS location should be similarly affected by the GPS error.

TABLE III
AVERAGE ESTIMATION ERROR

Estimated position	*1 sec ahead*	*3 sec ahead*
IMM	1.9461	6.5276
IMM with GIS	1.8872	5.1423

Units are in meters. Used 800 data points from trajectory in Fig 8.

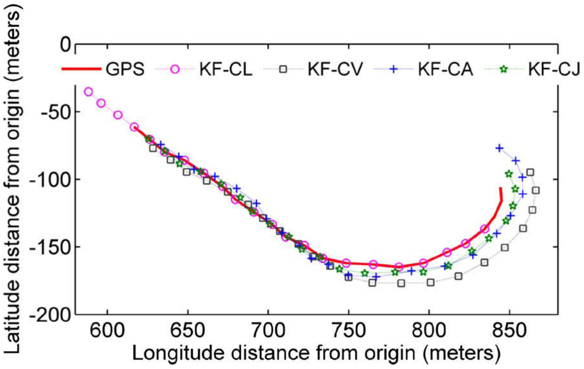

Fig. 8. Comparison of the estimated 3-s-ahead location and actual GPS readings for all four KF models using 21 data points for the above specific turn.

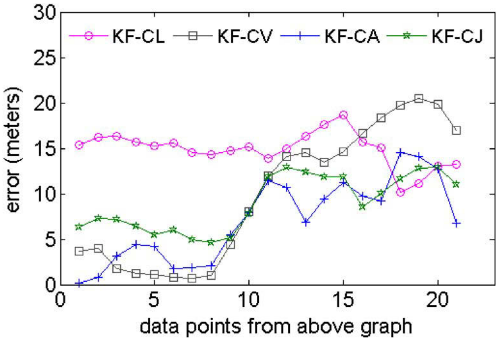

Fig. 9. Calculated error for all KF models between the 3-s-ahead estimated location and the actual location 3 s later using 21 data points for the above specific turn.

In Figs. 8 and 9, we can analyze the results of running the KF by themselves (i.e., each KF is predicting the future location 3 s later in time). Fig. 8 shows the predicted location 3 s ahead of time on the spatial trajectory (i.e., the same curve, as shown in Fig. 7), whereas Fig. 9 shows the error for each of the predictions of the future vehicle's location 3 s later in time compared with that of the actual GPS measurement. Both graphs are needed because KF-CL seems to be accurate in Fig. 8 because it will always be on a real GPS location since it assumes no movement (i.e., CL), but it is the KF-CL that shows a lot of errors in Fig. 9 since the vehicle is always moving through this curve. The estimated future location for this model will be where the current GPS location is (right over the GPS line), but this will not be accurate if the vehicle is moving, and this is where Fig. 9 displays this error.

Because the curve selected in Fig. 7 has been driven at a somewhat constant speed, we can see that both the KF-CV and the KF-CA were the most accurate in this case until the curve started.

B. Evaluation of the IMMs

To set-up the IMM, it was necessary to calculate the transition probability matrix in (11) using the GPS log for the full trajectory shown in Fig. 7. From this full GPS log that contained multiple scenarios, we determined which transition was occurring frame by frame by comparing the actual measurements from the GPS to the smoothed measurements. The smoothing of the data was done with a rolling window using a combination of median smoothing, splitting the sequence, and Hann's sequence, which removed any abrupt changes from the data. Each transition was determined by the type of change, such as no change, a constant change, and so on. Similarly, by calculating the covariance of the differences in the measurements to each other, we obtained the measurement-noise-covariance matrix R. Lastly, by calculating the covariance of the differences in the measurements compared with their respective x and y components, we obtained the process-covariance noise Q for each KF. From this type of information, we obtained the following transition probability matrix:

$$p^{ij} = \begin{bmatrix} 0.154 & 0.154 & 0.385 & 0.308 \\ 0.011 & 0.470 & 0.305 & 0.214 \\ 0.014 & 0.259 & 0.458 & 0.269 \\ 0.002 & 0.243 & 0.508 & 0.247 \end{bmatrix}. \qquad (22)$$

Looking at (22), we can clearly identify some scenarios. For example, when in a CL state (first row), it is more probable for it to change to a CA or CJ state than to a CV state, and this is understandable because, for a vehicle to start moving when it is at a complete stop, it will need to accelerate.

In addition, we are going to look only at the 3-s-away estimation results as this is the most important one for us. Looking at a 1-s-ahead estimation gives us some very accurate results, but this would not be enough warning time for the driver to react; therefore, we will look at a 3-s-away estimation and how accurate we can get that.

The results obtained from the IMM were not good enough to make this system very reliable by itself. In Table II, we can identify a 45% improvement when using the GIS-correction method, but the error is still significant when predicting the vehicle's location 3 s ahead of time. Table III, which is similar with Table II, shows the average errors for the estimated future vehicle's location 1 and 3 s ahead of time, but the whole trajectory, as shown in Fig. 8, is used to test this system. The numbers do not show a great improvement as in Table II because, when the vehicle is traveling in a straight line, the error in the estimated future location is smaller; therefore, adding GIS correction is not as beneficial. Overall, although GIS does show to be very helpful, particularly during curves, it is still not enough to use it by itself as it was set up here. A much needed improvement would be the implementation of more sensors that could run at higher frequencies.

In Fig. 10, we can compare visually the estimated 3-s-ahead positions with the GPS values. It also shows that the IMM had a lot of error at the beginning of the turn and, after a few seconds, converged more with the actual data. Therefore, this method used as part of a collision-avoidance system would produce many false warnings.

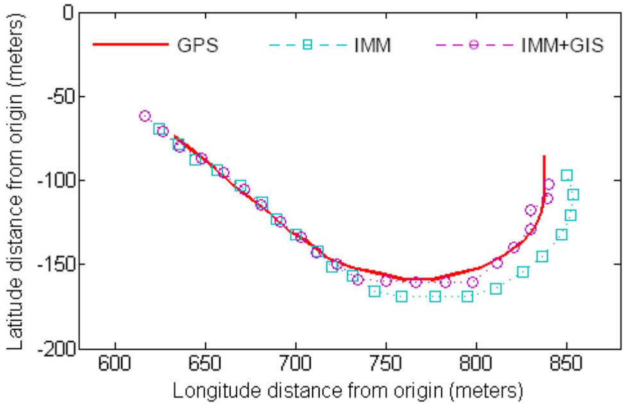

Fig. 10. Comparison of the 3-s-ahead estimated location between IMM, IMM+GIS, and the actual GPS location 3 s later using 21 data points for the selected curve.

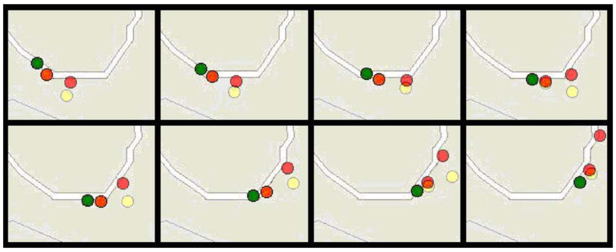

Fig. 11. Frame shots of simulation during the selected curve.

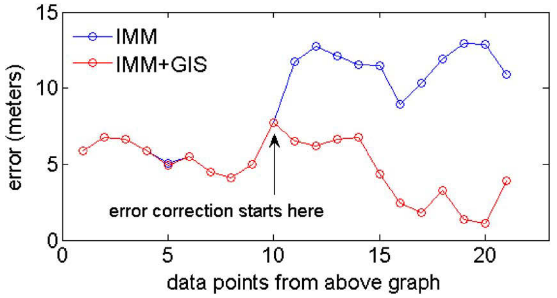

Fig. 12. Error measured between the 3-s-ahead IMM estimation with and without GIS correction and the actual GPS readings 3 s later using 21 data points for the selected curve.

C. GIS

The implementation of GIS data with the IMM-estimation process showed very promising results.

In Fig. 11, we can see the frame shots of the simulation program. It shows in light yellow the three positions corresponding to 1- and 3-s-away estimations. In red, the images show the corrected predicted location for each of the 1- and 3-s-away estimations. It is easy to see how much the GIS correction helps with the actual estimation of the future positions of the vehicle. To look at some numbers, we can use Table II to confirm this visual conclusion. The table shows the average error for the selected turn, and we can see a noticeable difference compared with the method without any GIS correction, particularly when looking at the 3-s-ahead estimation. This method is a lot more reliable and should give a lot less false warnings because its approximated 3 m of error is a little more than a compact-vehicle width and about the same as its length.

Fig. 12 is a further visual aid to be able to compare it to the previous two methods and see how much more accurate this is.

The GIS-error-correction method used in this paper is somewhat simple and straightforward, and it can possibly be improved with other existing methods, but it is only enough to help us determine whether using GIS data with our trajectory estimation models is an improvement.

VI. Conclusion

This paper has implemented four KFs to account for the identified possible states (i.e., CLs, CV, CA, and CJ) an automobile can be found in. These four KFs were set up to be part of an IMM system that provided the predicted future location of the automobile up to 3 s ahead of time. To improve the prediction error of the IMM setup, this paper has added an iterated geometrical error-detection method based on the GIS system. The assumption made was that the automobile would remain on the road; therefore, the predictions of future locations that fall outside of the road were corrected accordingly, making great reduction to prediction error, as shown in the experimental results.

The research-observed estimation values at 3 s ahead of time to allow for enough reaction time if this setup were to be used in some type of driver's aid system. As shown in this paper, a 3-s-ahead estimation has a lot of error, but with the help of GIS data, this error can be reduced drastically, particularly during turns, which is where research seems to have the most problems with [10].

The idea of merging spatial GPS data with GIS road information, which is given some assumptions, has proven to improve the accuracy of predicting a vehicle's future. In addition, in some scenarios, it could be an interesting addition to a collision-avoidance system.

Despite the improved predictions shown in this paper, this system can be further improved. The implemented GIS method in this paper has been straightforward and could be improved by looking into more detailed GIS data and by being able to determine the lane the vehicle is driving in to correct with more accuracy a badly estimated future location. The spatial data used from the GPS can also be complemented by using other types of sensors that are less error prone and can run at a frequency higher than 1 Hz.

References

[1] A. Tascillo and R. Miller, "An in-vehicle virtual driving assistant using neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2003, vol. 3, pp. 2418–2423.

[2] M. Lee and Y. Kim, "An efficient multitarget tracking algorithm for car applications," *IEEE Trans. Ind. Electron.*, vol. 50, no. 2, pp. 397–399, Apr. 2003.

[3] A. Amditis, E. Bertolazzi, M. Bimpas, F. Biral, P. Bosetti, M. Da Lio, L. Danielsson, A. Gallione, H. Lind, A. Saroldi, Sjo, and A. Gren, "A holistic approach to the integration of safety applications: The INSAFES subproject within the European framework programme 6 integrating project PReVENT," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 554–566, Dec. 2009.

[4] Y. Ikemoto, Y. Hasegawa, T. Fukuda, and K. Matsuda, "Zipping, weaving: Control of vehicle group behavior in non-signalized intersection," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2004, vol. 5, pp. 4387–4391.

[5] S. G. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath, "Collision sensing by stereo vision and radar sensor fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 606–614, Dec. 2009.

[6] S. Pietzsch, T. D. Vu, J. Burlet, O. Aycard, T. Hackbarth, N. Appenrodt, J. Dickmann, and B. Radig, "Results of a precrash application based on laser scanner and short-range radars," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 584–593, Dec. 2009.

[7] M. Chowdhary, "Driver assistance applications based on automotive navigation system infrastructure," in *Proc. ICCE*, Jun. 2002, pp. 38–39.

[8] C. Drane and C. Rizos, *Positioning Systems in Intelligent Transportation Systems*. Norwood, MA: Artech House, 1998.

[9] R. Bishop, *Intelligent Vehicle Technology and Trends*. Norwood, MA: Artech House, 2005.

[10] J. Ueki, J. Mori, Y. Nakamura, Y. Horii, and H. Okada, "Development of vehicular-collision avoidance support system by inter-vehicle communications," in *Proc. IEEE 59th Veh. Technol. Conf.*, May 2004, vol. 5, pp. 2940–2945.

[11] R. Miller and Q. Huang, "An adaptive peer-to-peer collision warning system," in *Proc. IEEE 55th Veh. Technol. Conf.*, May 2002, vol. 1, pp. 317–321.

[12] B. P. Zhang, J. Gu, E. Milios, and P. Huynh, "Navigation with IMU/GPS/digital compass with unscented Kalman filter," in *Proc. IEEE Int. Conf. Mechatron. Autom.*, Jul. 2005, pp. 1497–1502.

[13] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 328–342, Jun. 1995.

[14] R. Toledo, M. A. Zamora, B. Ubeda, and A. F. Gomez-Skarmeta, "An integrity navigation system based on GNSS/INS for remote services implementation in terrestrial vehicles," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Washington, DC, 2004, pp. 477–480.

[15] R. Toledo, M. A. Zamora, B. Ubeda, and A. F. Gomez, "High integrity IMM-EKF based road vehicle navigation with low cost GPS/INS," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 3, pp. 491–511, Sep. 2007.

[16] W. Kim, G. Jee, and J. Lee, "Efficient use of digital road map in various positioning for ITS," in *Proc. IEEE Position Localization Navig. Symp.*, San Diego, CA, 2000, pp. 170–176.

[17] X. Zhang, Q. Wang, and D. Wan, "Map matching in road crossings of urban canyons based on road traverses and linear heading-change model," *IEEE Trans. Instrum. Meas.*, vol. 56, no. 6, pp. 2795–2803, Dec. 2007.

[18] C. B. Lee, D. D. Lee, N. S. Chung, I. S. Chang, E. Kawai, and F. Takahashi, "Development of a GPS codeless receiver for ionospheric calibration and time transfer," *IEEE Trans. Instrum. Meas.*, vol. 42, no. 2, pp. 494–497, Apr. 1993.

[19] M. Matosevic, Z. Salcic, and S. Berber, "A comparison of accuracy using a GPS and a low-cost DGPS," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 5, pp. 1677–1683, Oct. 2006.

[20] M. E. Najjar and P. Bonnifait, "A roadmap matching method for precise vehicle localization using belief theory and Kalman filtering," in *Proc. IEEE 11th Int. Conf. Adv. Robot.*, Coimbra, Portugal, 2003, pp. 1677–1682.

[21] Y. Cui and S. S. Ge, "Autonomous vehicle positioning with GPS in urban canyon environments," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 15–25, Feb. 2003.

[22] Holux Technology Inc., *Holux GR-213 GPS Specifications*. [Online]. Available: http://www.holux.com/JCore/en/support/DLF.jsp?DLU=http://www.holux.com/JCore/UploadFile/7011686.pdf

[23] A. Lahrech, C. Boucher, and J. C. Noyer, "Fusion of GPS and odometer measurements for map-based vehicle navigation," in *Proc. IEEE Int. Conf. Ind. Technol.*, Dec. 2004, pp. 944–948.

[24] S. Sukkarieh, "Low cost, high integrity, aided inertial navigation systems for autonomous land vehicles," Ph.D. dissertation, Univ. Sydney, Sydney, Australia, 2000.

[25] C. Hide, T. Moore, and M. Smith, "Multiple model Kalman filtering for GPS and low-cost INS integration," in *Proc. ION GNSS*, 2004, pp. 1096–1103.

[26] J. Bohg, *Real-Time Structure from Motion Using Kalman Filtering*. Dresden, Germany: Technische Universitat Dresden, Mar. 2005.

[27] G. Welch and G. Bishop, "An introduction to the Kalman filter," in *Proc. SIGGRAPH*, 2001, pp. 16–32. Course notes.

[28] J. Bohg, "Real-times structure from motion using Kalman filtering," Ph.D. dissertation, Technische Universität Dresden, Dresden, Germany, 2005.

[29] C. Hu, W. Chen, Y. Chen, and D. Liu, "Adaptive Kalman filtering for vehicle navigation," *J. Global Positioning Syst.*, vol. 2, no. 1, pp. 42–47, 2003.

[30] Y. Zhang, H. Hu, and H. Zhou, "Study on adaptive Kalman filtering and algorithms in human movement tracking," in *Proc. IEEE Int. Conf. Inf. Acquisition*, 2005, pp. 11–15.

[31] L. C. Yang, J. H. Yang, and E. M. Feron, "Multiple model estimation for improving conflict detection algorithms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2004, vol. 1, pp. 242–249.

[32] X. Wang, "Maneuvering target tracking and classification using multiple model estimation theory," Ph.D. dissertation, Univ. Melbourne, Melbourne, Australia, 2001.

[33] C. Hilde, T. Moore, and M. Smith, *Multiple Model Kalman Filtering for GPS and Low-Cost INS Integration*. Nottingham, U.K.: Inst. Eng., Surveying Space Geodesy, Univ. Nottingham, 2004.

[34] E. Derbez and B. Remillard, *The IMM CA CV performance*. unpublished.

[35] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation*. Hoboken, NJ: Wiley, 2001.

[36] L. A. Johnson and V. Krishnamurthy, "An improvement to the interactive multiple model (IMM) algorithm," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 2909–2923, Dec. 2001.

[37] M. Green, "How long does it take to stop? Methodological analysis of driver perception-brake times," *Transp. Hum. Factors*, vol. 2, no. 3, pp. 195–216, Sep. 2000.

[38] H. Himberg and Y. Motai, "Head orientation prediction: Delta quaternions versus quaternions," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1382–1392, Dec. 2009.

[39] Y. Bar-Shalom, Li, and Xiao-Rong, *Estimation and Tracking: Principles, Techniques and Software*. Storrs, CT: YBS Publ., 1998.

Cesar Barrios received the B.S. and M.S. degrees in electrical engineering from the New Jersey Institute of Technology, Newark, in 1999 and 2001, respectively. He is currently working toward the Ph.D. degree in electrical engineering from the University of Vermont, Burlington.

Since he received the B.S. degree, he has been working for IBM. He first started in the information technology field and then moved into the Semiconductor Research and Development Center, Essex Junction, Vermont.

Yuichi Motai (M'01) received the B.Eng. degree in instrumentation engineering from Keio University, Tokyo, Japan, in 1991; the M.Eng. degree in applied systems science from Kyoto University, Kyoto, Japan, in 1993; and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 2002.

He is currently an Assistant Professor of electrical and computer engineering with Virginia Commonwealth University, Richmond. His research interests include the broad area of sensory intelligence, particularly in medical imaging, pattern recognition, computer vision, and sensory-based robotics.