Road Network Identification by means of the Hough Transform with Uncertainty Analysis

ERIC SALERNO
NAGAVENKAT ADURTHI
TARUNRAJ SINGH
PUNEET SINGLA
ADNAN BUBALO
MARIA CORNACCHIA
MARK ALFORD
ERIC JONES

The focus of this paper is on the use of ground target kinematics to estimate the underlying road network on which the vehicles are assumed to be travelling. Assuming that the road network can be represented as an amalgamation of straight line segments, a Hough transform approach is used to identify portion of road which correspond to straight line segments. Since multiple tracks can be associated with one segment of the road and since the track estimates are inherently uncertain, an iterative approach is presented to identify a parametric representation of the line segments of the roads using the total least squares cost function. Cramér-Rao bounds are identified to characterize the bounds on the uncertainty associated with the proposed approach. A complex dataset which include multiple tracks is used to illustrate the ability of the proposed algorithm to identify the underlying road network and characterize the uncertainty associated with the parametric estimate of the road.

Manuscript received December 11, 2013; revised November 4, 2014 and February 24, 2015; released for publication March 1, 2015.

Refereeing of this contribution was handled by Robert Lynch.

Authors' addresses: E. Salerno, N. Adurthi, T. Singh, P. Singla, Dept. of Mech. & Aero. Eng., University of Buffalo, Buffalo, NY 14260 (e-mail: esalerno@buffalo.edu, nagavenk@buffalo.edu, tsingh@buffalo.edu, psingla@buffalo.edu). A. Bubalo, M. Cornacchia, M. Alford, E. Jones, Information Directorate, Air Force Research Laboratory, Rome, NY 13441 (e-mail: Adnan.Bubalo@rl.af.mil).

1557-6418/15/$17.00

1 INTRODUCTION

Automatic cartographic feature extraction has been the goal for road network identification given the tremendous growth in automobiles with navigation systems, and the interest in driverless cars. In post hurricane disaster scenarios when bridges and roads might be washed out, there is a need to rapidly update road network databases for logistics. In regions of conflict or deserts, there is a need to develop road maps to identify safe travel routes. Precise road networks can also be used to enhance the performance of ground vehicle trackers by restricting the motion of the target to the road network.

Synthetic Aperture Radar (SAR) and Ground Moving Target Indicator (GMTI) data is often processed and analyzed to produce such networks. SAR produces images of varying intensity which can be processed to separate buildings, roads, and terrain. However, SAR is only able to detect prominent existing features [1]. For example, SAR will only detect a road if there is a distinct outline of such a path. GMTI on the other hand tracks moving targets and relays the latitude and longitude coordinates as well as the kinematic information. The disadvantage of GMTI, however, is the necessity of a moving target. Should the target stop or be obstructed in any way from the sensors, the tracker will lose the target for the duration of the obstructions [2]. Many of the currently available algorithms rely on information from pre-existing road maps, however, in many scenarios the availability of this a priori information is limited and inaccurate. In some situations there are no existing road maps, such as in times of conflict in desert regions. Therefore the need for an algorithm which can accurately estimate road networks in a timely manner is of great importance. Furthermore, there is a lack of a real quantifiable measure of the accuracy of the extracted road estimates. Several available algorithms use a "completeness" and "correctness" measure, which is a comparison of the extracted road network and the actual network [3, 4, 5, 6]. However, as previously stated in many situations there are no available true networks (i.e., desert regions) so the metrics used to evaluate the performance of algorithms are not relevant.

Hu, Razdan, Femiani, Cui, and Wonka use a spoke and wheel method in order to determine the footprints [5]. These footprints or polygons, are road segments which span in any direction and terminate when the intensity of the spoke or line segment falls below a threshold. Then a toe-finding algorithm is used to determine the number of branches in the footprint. In this portion of the algorithm, if the angle between two branches is less than 45 degrees they are merged, in some cases this will eliminate Y-shaped intersections and parallel roads. Instead of using centerlines to approximate the road network they utilize inscribed lines to define the road structure. Finally the road network

is trimmed to eliminate noisy data which might contribute to false roads, and also possible gaps between roads due to obstructions. This algorithm relies heavily on pre-existing information pertaining to roads such as the possible shapes of intersections, road widths, and angle of roads at these intersections.

Tupin, Maitre, Mangin, Nicolas, and Pechersky first identify linear features in the data and then separate the true segments by using a Markov random field (MRF) [1]. Two different line detectors are used to identify candidate road segments and then the results of these two detectors are fused together. With the identification of the candidates, the MRF-based model fills in large gaps and removes the false detections. This MRF-based model relies on a priori information of the road network being developed. The assumption is made that all roads lead to an initial starting point which can limit the accuracy of the overall algorithm.

Shackelford and Davis use a pixel-based fuzzy classifier and an object based classification approach to identify the road networks [6]. Skeletonization is an approach in which the image is thinned or eroded away until only the essential lines remain. This method produces a large amount of false positives in the road network. The second approach utilized is an iterative approach in which the longest roads are initially identified and then shorter and shorter segments are added throughout the algorithm. The second algorithm proves to be much better than skeletonization, however, the "completeness" measure of the road network has decreased in both the urban and suburban scenarios for the second algorithm when compared with the first.

Sklarz, Novoselsky, and Dorfan focus on the fusion of linear segments and curves based on a unified entity approach rather than a single pixel based approach [7]. The road network can either begin as a blank slate or already contain roads. As new tracks become available the curve is associated with an existing curve if one exists otherwise a new segment of road is added to the existing network. Should a track already exist, the new track is cropped into segments to match with the relative endpoints of the pre-existing road segments. The optimization of the curve fusion's computational complexity increases drastically as the curves are discretized into more finite segments thus limiting the potential of the algorithm.

Koch, Koller, and Ulmke utilize a Multiple Hypothesis Tracking (MHT) algorithm, which consists of target track extraction, prediction, filtering, track maintenance, and retrodiction [2]. It is assumed that the posterior probability density function is Gaussian, thus the Kalman filter is utilized since the algorithm breaks the road up into linear segments. The pruning removes any segments, which have a weight smaller than a threshold, depending on the threshold this could cause some issues with removing actual tracks. The merging depends on segments having similar state vectors and covariances.

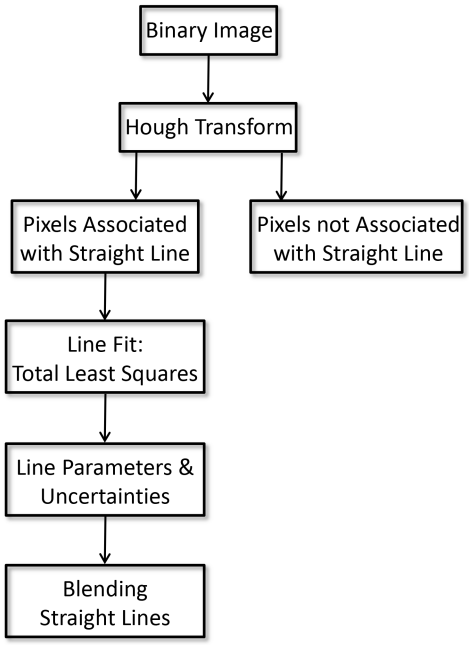


Fig. 1. Algorithm flow chart.

In this paper, a method for developing road networks and characterizing the uncertainty in these estimates is developed. It is assumed that road networks can be broken down into piecewise linear segments. Figure 1 illustrates the sequence of processing of data to generate a parameterization of the road network with the associated uncertainties. The initial processing of the data is done by creating a binary image of the track data and extracting possible line segments using the Hough transform. This is followed by the clustering of the data associated with the identified straight lines. Since the Hough transform does not provide a measure of uncertainty, the Total Least Squares approach is implemented and the Cramér Rao lower bounds is derived from this maximum likelihood estimate. The Total Least Squares solution allows for an iterative estimate, which is updated in time as additional measurements become available. The Least Squares solution will be used as an initial estimate for the recursive Total Least Squares algorithm. Once the data collection has been terminated the individual line segments can be merged, extended or trimmed, and blended to produce a more complete road network.

Section 2 details the derivation of the Hough transform for identifying straight line edges in an image. The recursive Total Least Squares solution is presented in Section 3 along with the corresponding uncertainty analysis and derivations. In Section 4 the results of the algorithms outlined are applied to a data set and we conclude the paper with suggestions for future research in Section 5.

2 HOUGH TRANSFORM

The Hough transform is a well studied feature extraction technique that has been used extensively in

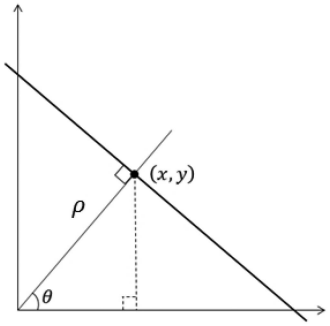


Fig. 2. Parameter identification.

image analysis [9]. This transform requires the image to be binary in nature, where white pixels correspond to ones and black pixels correspond to zeros. The derivation of the Hough transform requires basic trigonometric identities. Suppose we have a line oriented as shown in Figure 2 then by defining the parameters ρ and θ we can derive the Hough transform. The perpendicular distance from the origin to a line is denoted by ρ. The angle that this distance vector makes with the x-axis is θ. We note the definitions of the cosine and sine functions:

$$\cos\theta = \frac{x}{\rho} \qquad \sin\theta = \frac{y}{\rho}. \tag{1}$$

Algebraic substitution of a single cosine and sine term in the Pythagorean trigonometric identity leads to the equation:

$$\frac{x}{\rho}\cos\theta + \frac{y}{\rho}\sin\theta = 1. \tag{2}$$

It is now a simple matter to rearrange Equation (2) to obtain the conventional form of the Hough transform as given by the equation:

$$\rho = x\cos\theta + y\sin\theta. \tag{3}$$

Now if Equation (3) is rearranged into a slope-intercept form

$$y = -\frac{\cos\theta}{\sin\theta}x + \frac{\rho}{\sin\theta} \tag{4}$$

we can infer that when θ approaches zero degrees, corresponding to a vertical line, the slope tends to infinity, resulting in an poor parameterization of the line. However, using Equation (3), we avoid this problem.

The principle concept of the Hough transform in the line identification algorithm can be stated as the following: **if two points are collinear then they share a pair (ρ,θ) of commonality in the Hough space.** However, in order to determine this common pair, a Hough matrix must be constructed, this is done by iterating over a θ range of -90 to 90 degrees for each white pixel, which corresponds to the (x,y) coordinates, in the image. Recall that θ is the angle the ρ vector makes with the x-axis. We can imagine that an arbitrary number of lines (black solid lines) pass through each coordinate shown by the solid black lines in Figure 3, which are associated with a (ρ,θ) given by the normal lines passing through the origin shown by the dashed lines. The solid blue line is the line of interest and Figure 3 illustrates two points which lie on this line. Note that both these points illustrated by the solid circle have a coincident (ρ,θ) pair, which parameterize the dashed blue line.

For each white pixel of the binary image with a coordinate x_i, y_i determine the parameter

$$\rho(\theta) = x_i\cos(\theta) + y_i\sin(\theta) \tag{5}$$

which corresponds to a sinusoid in the (ρ,θ) space. Determine the (ρ,θ) pair for every white pixel of the binary images, round it to the closest discretized value of (ρ,θ), and augment the appropriate indices of an array called the accumulator. Since every point on a line will share a unique (ρ,θ) pair which corresponds to a line normal to the line of interest passing through the origin, the accumulator bin with the highest count will correspond to parameters of a straight line. Figure 4 illustrates the mapping of the Hough accumulator with the white pixels indicating a higher count compared to the black pixels. The point highlighted by the square corresponds to the (ρ,θ) combination associated with a straight line.

Matlab's image processing toolbox is used to determine the Hough transform, identify the peaks which corresponds to the straight lines, identify the points on the images corresponding to the identified line segments which are subsequently used to characterize the uncertainty of the identified lines.

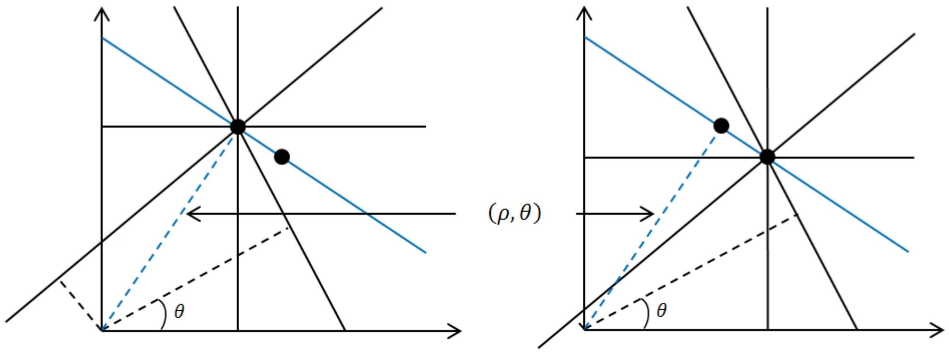


Fig. 3. Collinear points.

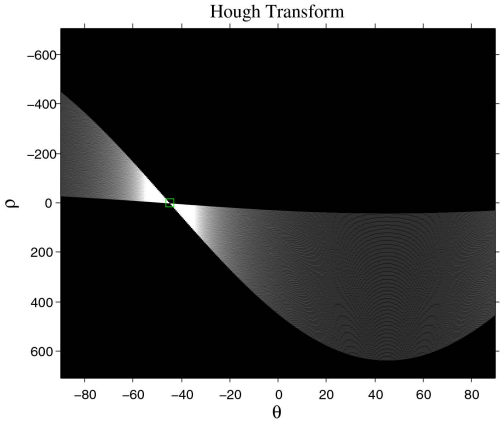


Fig. 4. Hough transform.

3 MAXIMUM LIKELIHOOD ESTIMATORS

In this section we will derive the solutions for the Total Least Squares (TLS) to estimate the parameters of a straight line. In this formulation, the measurement noise covariance matrix is allowed to be populated, i.e., the dependent and independent variables are noisy and can be correlated. The Hough transform presented in Section 2 will be utilized to identify groups of nearly collinear points in an image. A best fit line can be obtained using the TLS solution since there is noise in both the x and y directions. The TLS solution, which will be derived, does not have a closed form solution and therefore requires an initial estimate to initiate the solver which minimizes the normal distance of the measurements from the line. This initial estimate can be given by a transformed version of the Hough coefficients (i.e., transform from ρ and θ to slope, m and intercept b), however if θ is zero degrees then we would obtain an infinite slope, which forces the TLS solution to diverge. Therefore, rather than use the Hough transform coefficients transformed to the appropriate slope and intercept we will simply utilize a Least Squares estimate as the initial guess for the TLS algorithm.

3.1 Total Least Squares

Consider the problem of a straight line fit where the true model is given by the equation:

$$y_i = mx_i + c \tag{6}$$

where y_i and x_i are the true dependent and independent coordinates. m and c correspond to the slope and ordinate intercept of the line. Assuming that the measurement of both y_i and x_i are noisy, the measurement equations and the corresponding pdf of the noise are given by the equations:

$$\tilde{x}_i = x_i + \nu_x \tag{7}$$

$$\tilde{y}_i = y_i + \nu_y \tag{8}$$

$$p(\nu_1,\nu_2) = \mathcal{N}\left(\begin{bmatrix}\nu_1\\ \nu_2\end{bmatrix} : \begin{bmatrix}0\\ 0\end{bmatrix}, \begin{bmatrix}\sigma_{xx}^2 & \sigma_{xy}\\ \sigma_{xy} & \sigma_{yy}^2\end{bmatrix}\right) \tag{9}$$

permitting the measurement noise in x_i and y_i to be correlated. Note that σ_{xx}^2 and σ_{yy}^2 denote the variance of ν_x and ν_y respectively and σ_{xy} represent the cross-covariance of ν_x and ν_y. The notation $\mathcal{N}(\nu : \mu, \Sigma)$ is a Gaussian probability density function(pdf) for the random vector ν with mean μ and covariance Σ. Identifying the parameters of a line when provided with n measurements, the likelihood function for the measurements is given by the equation:

$$p = \prod_{i=1}^{n} p_i \tag{10}$$

$$p_i = p\left(\begin{bmatrix}\tilde{x}_i\\ \tilde{y}_i\end{bmatrix} \mid m, c, x_1, x_2, \ldots, x_n\right)$$

$$= \mathcal{N}\left(\begin{bmatrix}\tilde{x}_i - x_i\\ \tilde{y}_i - y_i\end{bmatrix} : \begin{bmatrix}0\\ 0\end{bmatrix}, \begin{bmatrix}\sigma_{xx}^2 & \sigma_{xy}\\ \sigma_{xy} & \sigma_{yy}^2\end{bmatrix}\right) \tag{11}$$

$$= \mathcal{N}\left(\begin{bmatrix}\tilde{x}_i - x_i\\ \tilde{y}_i - mx_i - c\end{bmatrix} : \begin{bmatrix}0\\ 0\end{bmatrix}, \begin{bmatrix}\sigma_{xx}^2 & \sigma_{xy}\\ \sigma_{xy} & \sigma_{yy}^2\end{bmatrix}\right) \tag{12}$$

$$= \mathcal{N}\left(\begin{bmatrix}\tilde{x}_i\\ \tilde{y}_i\end{bmatrix} : \begin{bmatrix}x_i\\ mx_i + c\end{bmatrix}, \begin{bmatrix}\sigma_{xx}^2 & \sigma_{xy}\\ \sigma_{xy} & \sigma_{yy}^2\end{bmatrix}\right). \tag{13}$$

The Log-Likelihood to be maximized with respect to the free variables m, c and x_i where $i = 1, 2, \ldots, n$ is

$$\ln(p) = -n\ln\left(2\pi\sqrt{\sigma_{xx}^2\sigma_{yy}^2 - \sigma_{xy}^2}\right) - \frac{1}{2(\sigma_{xx}^2\sigma_{yy}^2 - \sigma_{xy}^2)}\sum_{i=1}^{n}(\sigma_{yy}^2(\tilde{x}_i - x_i)^2 - 2\sigma_{xy}(\tilde{x}_i - x_i)(\tilde{y}_i - mx_i - c) + \sigma_{xx}^2(\tilde{y}_i - mx_i - c)^2). \tag{14}$$

There are $n + 2$ variables: m, c, and all the n abscissas x_i. Differentiate the log-likelihood with respect to the variables and solve the $n + 2$ equations:

$$\frac{\partial \ln(p)}{\partial m} = -\frac{1}{2(\sigma_{xx}^2\sigma_{yy}^2 - \sigma_{xy}^2)} \times \sum_{i=1}^{n}(2\sigma_{xy}(\tilde{x}_i - x_i)x_i - 2\sigma_{xx}^2(\tilde{y}_i - mx_i - c)x_i) = 0 \tag{15}$$

$$\frac{\partial \ln(p)}{\partial c} = -\frac{1}{2(\sigma_{xx}^2\sigma_{yy}^2 - \sigma_{xy}^2)} \times \sum_{i=1}^{n}(2\sigma_{xy}(\tilde{x}_i - x_i) - 2\sigma_{xx}^2(\tilde{y}_i - mx_i - c)) = 0 \tag{16}$$

$$\frac{\partial \ln(p)}{\partial x_i} = -\frac{1}{2(\sigma_{xx}^2\sigma_{yy}^2 - \sigma_{xy}^2)} \times (-2\sigma_{yy}^2(\tilde{x}_i - x_i) + 2\sigma_{xy}(\tilde{y}_i - mx_i - c) + 2\sigma_{xy}(\tilde{x}_i - x_i)m - 2\sigma_{xx}^2(\tilde{y}_i - mx_i - c)m) = 0. \tag{17}$$

Solving for the x_i from equation (17) leads to the equation:

$$x_i = \frac{c(-m\sigma_{xx}^2 + \sigma_{xy}) - m\sigma_{xy}\tilde{x}_i + \sigma_{yy}^2\tilde{x}_i + m\sigma_{xx}^2\tilde{y}_i - \sigma_{xy}\tilde{y}_i}{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2\sigma_{xy}m}. \tag{18}$$

Substituting the expression for x_i back into the log likelihood function $\ln(p)$ of equation (14) and on simplification, the final cost function in terms of m and c is:

$$\ln(p) = -n\ln\left(2\pi\sqrt{\sigma_{xx}^2\sigma_{yy}^2 - \sigma_{xy}^2}\right) - \frac{1}{2}\sum_{i=1}^{n}\frac{(\tilde{y}_i - m\tilde{x}_i - c)^2}{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2\sigma_{xy}m} \tag{19}$$

or the problem can be stated as

$$\min_{m,c}\sum_{i=1}^{n}\frac{(\tilde{y}_i - m\tilde{x}_i - c)^2}{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2\sigma_{xy}m}. \tag{20}$$

Additionally, the noise covariance parameters can be specific to each measurement, i.e., σ_{xx}, σ_{yy}, and σ_{xy} can be $\sigma_{xx}^{(i)}, \sigma_{yy}^{(i)}$ and $\sigma_{xy}^{(i)}$ respectively corresponding to the measurement $(\tilde{x}_i, \tilde{y}_i)$. It can be observed that, when the noise in the variables is uncorrelated ($\sigma_{xy} = 0$) and of equal variance($\sigma_{xx} = \sigma_{yy} = \sigma$), the standard Total least square problem is recovered, which can be identified as the perpendicular distance of $(\tilde{x}_i, \tilde{y}_i)$ to the line $y = mx + c$, and

$$\min_{m,c}\sum_{i=1}^{n}\frac{(\tilde{y}_i - m\tilde{x}_i - c)^2}{1 + m^2}$$

is the resulting cost function to be minimized.

3.2 Alternate Equivalent Formulation

The algebraic manipulation involved in arriving at the cost function of (20) can be avoided altogether. Consider the truth model where we represent the truth (x_i, y_i) in terms of the noisy measurement $(\tilde{x}_i, \tilde{y}_i)$ as:

$$y_i = mx_i + c \tag{21}$$

$$x_i = \tilde{x}_i - \nu_x \tag{22}$$

$$y_i = \tilde{y}_i - \nu_y. \tag{23}$$

Substituting (22) and (23) into (21) results in the equation:

$$\tilde{y}_i - m\tilde{x}_i - c = \nu_y - m\nu_x. \tag{24}$$

Define ν' as:

$$\nu' = \frac{\nu_y - m\nu_x}{\sqrt{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}}} \tag{25}$$

whose mean and variance of random variable ν' conditioned on m are given by the equations:

$$E[\nu'] = E\left[\frac{\nu_y - m\nu_x}{\sqrt{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}}}\right] = \frac{E[\nu_y] - mE[\nu_x]}{\sqrt{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}}} = 0 \tag{26}$$

$$E[\nu'^2] = E\left[\frac{(\nu_y - m\nu_x)^2}{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}}\right] = \frac{E[\nu_y^2] + m^2E[\nu_x^2] - 2mE[\nu_x\nu_y]}{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}} = 1. \tag{27}$$

The random variable ν' has a Gaussian distribution given by the equation:

$$p(\nu' \mid m) = \mathcal{N}(\nu' : 0, 1). \tag{28}$$

But from equations (24) and (25), one has

$$\nu' = \frac{\tilde{y}_i - m\tilde{x}_i - c}{\sqrt{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}}}.$$

Hence, for a set of measurements $(\tilde{x}_i, \tilde{y}_i)$, the Maximum Likelihood Estimator (MLE) requires maximizing the function given by Equation (10), which can be rewritten as:

$$p_i = \mathcal{N}\left(\frac{\tilde{y}_i - m\tilde{x}_i - c}{\sqrt{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}}} : 0, 1\right). \tag{29}$$

The negative of the Log-Likelihood to be minimized with respect to the free variables m and c is

$$J = -\ln(p) = \sum_{i=1}^{n}\frac{(\tilde{y}_i - m\tilde{x}_i - c)^2}{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}}. \tag{30}$$

Notice that there are no x_i variables in the cost function. Differentiating the cost function J only with the variables m and c, we arrive at the gradient constraint equations:

$$\frac{\partial J}{\partial m} = \sum_{i=1}^{n}\left(\frac{2(\tilde{y}_i - m\tilde{x}_i - c)(-\tilde{x}_i)}{\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy}} - \frac{(\tilde{y}_i - m\tilde{x}_i - c)^2(2\sigma_{xx}^2m - 2\sigma_{xy})}{(\sigma_{yy}^2 + \sigma_{xx}^2m^2 - 2m\sigma_{xy})^2}\right) = 0 \tag{31}$$

$$\frac{\partial J}{\partial c} = \sum_{i=1}^{n}(-2(\tilde{y}_i - m\tilde{x}_i - c)) = 0 \tag{32}$$

which is nonlinear in m and c and can be solved numerically. One can use the solution of the least squares problem to initialize the nonlinear solver. One can also

estimate the most likely value of the variable x_i using the equation:

$$x_i = \frac{c(-m\sigma_{xx}^2 + \sigma_{xy}) - m\sigma_{xy}\tilde{x}_i + \sigma_{yy}^2\tilde{x}_i + m\sigma_{xx}^2\tilde{y}_i - \sigma_{xy}\tilde{y}_i}{\sigma_{yy}^2 + \sigma_{xx}^2 m^2 - 2\sigma_{xy}m} \tag{33}$$

from the resulting solution for the slope m and intercept c.

3.3 Geometric Interpretation

Consider the case of estimating the parameters of a straight line where the $\tilde{x}_i$ and $\tilde{y}_i$ measurements are contaminated by isotropic noise. Figure 5 illustrates the true x_i and y_i and the corresponding contaminated $\tilde{x}_i$ and $\tilde{y}_i$ data. Assuming the parameters $\sigma_{xx} = \sigma_{yy} = 1$ and σ_{xy}=0, the MLE cost function given by Equation (30) reduces to:

$$J = \sum_{i=1}^{n} \frac{(\tilde{y}_i - m\tilde{x}_i - c)^2}{1 + m^2} \tag{34}$$

which is referred to as geometric distance which is a gradient weighted algebraic distance. This error referred to as the Sampson error is the first order approximation of the geometric distance of the point from the curve.

In Figure 5, one can easily illustrate the line connecting the points $(\tilde{x}_i, \tilde{y}_i)$ along the normal to the true line has a length given by equation $(\tilde{y}_i - m\tilde{x}_i - c)^2/(1 + m^2)$, illustrating that the total least squares cost function corresponds to minimizing the geometric distance. Note that the least squares problem minimizes the distance $(\tilde{y}_i - m\tilde{x}_i - c)^2$ which is referred to as the algebraic distance and corresponds to the distance along an oblique projection.

3.4 Least Squares

The least squares problem is a special case of the total least squares where the independent variable x_i is not random, i.e., $\sigma_{xx} = 0$. This results in the cost function given by Equation (30) reducing to:

$$J = -\ln(p) = \sum_{i=1}^{n} \frac{(\tilde{y}_i - mx_i - c)^2}{\sigma_{yy}^2} \tag{35}$$

which has a closed form solution for m and c:

$$\begin{Bmatrix} m \\ c \end{Bmatrix} = \begin{Bmatrix} \sum_{i=1}^{n} x_i^2 & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & n \end{Bmatrix}^{-1} \begin{Bmatrix} \sum_{i=1}^{n} x_i\tilde{y}_i \\ \sum_{i=1}^{n} \tilde{y}_i \end{Bmatrix}. \tag{36}$$

3.5 Uncertainty Analysis

In this section, we will present the derivation for the Cramér Rao lower bounds which provides a measure of uncertainty in the coefficients of the linear fit. The derivation is based on a fully populated covariance matrix and the bounds are estimated by the inverse of the Fisher Information matrix. The derivations are accompanied by a Monte Carlo simulation to verify the convergence properties of the solutions. For the straight

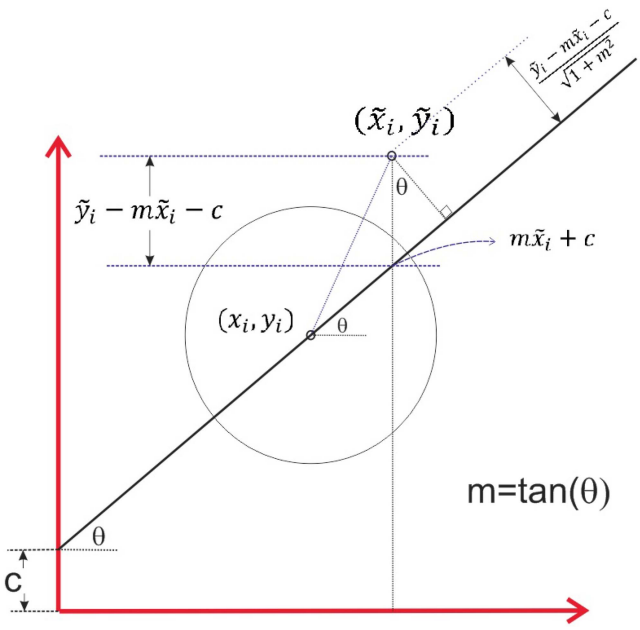


Fig. 5. Total least squares minimization.

line we have a total of 2+n unknown parameters, the slope m, the intercept c, and the true values of $x : x_t$.

After solving for the optimal estimates for the slope $\hat{m}$ and y-intercept $\hat{c}$, the optimal estimate for $\hat{x}_i$ is given by Equation (33). Assuming the parameters $\sigma_{xx} = \sigma_{yy} = 1$ and σ_{xy}=0, Equation (33) reduces to:

$$\hat{x}_i = \frac{-\hat{c}\hat{m} + \tilde{x}_i + \hat{m}\tilde{y}_i}{1 + \hat{m}^2} \tag{37}$$

$$\Rightarrow \hat{x}_i = \frac{-\hat{c}\hat{m} + \tilde{x}_i + \tilde{x}_i\hat{m}^2 - \tilde{x}_i\hat{m}^2 + \hat{m}\tilde{y}_i}{1 + \hat{m}^2} \tag{38}$$

$$\Rightarrow \hat{x}_i = \tilde{x}_i + \frac{\hat{m}e_i}{1 + \hat{m}^2} \tag{39}$$

where

$$e_i = \tilde{y}_i - \hat{m}\tilde{x}_i - \hat{c} \tag{40}$$

is the error associated with the measurement $\tilde{y}_i$ and the estimate $\hat{y}_i$ at the measured x coordinate $\tilde{x}_i$. Figure 6 illustrates the optimal estimates $(\hat{x}_i, \hat{y}_i)$, which results in the equation:

$$\hat{y}_i = \tilde{y}_i - e_i - \hat{m}(\tilde{x}_i - \hat{x}_i) \tag{41}$$

$$\hat{y}_i = \tilde{y}_i - \frac{e_i}{1 + \hat{m}^2}. \tag{42}$$

It can be shown that the generalized solutions for the estimated values for x_{t_i} and y_{t_i} are:

$$\hat{x}_i^2 = \tilde{x}_i + \frac{(\hat{m}\sigma_{x_ix_i}^2 - \sigma_{x_iy_i})e_i}{\hat{m}^2\sigma_{x_ix_i}^2 - 2\hat{m}\sigma_{x_iy_i} + \sigma_{y_iy_i}^2} \tag{43}$$

$$\hat{y}_i^2 = \tilde{y}_i + \frac{(\hat{m}\sigma_{x_iy_i} - \sigma_{y_iy_i}^2)e_i}{\hat{m}^2\sigma_{x_ix_i}^2 - 2\hat{m}\sigma_{x_iy_i} + \sigma_{y_iy_i}^2} \tag{44}$$

based on Equation (33).

Recall that e_i is the error in the measurement with the appropriate estimates $e_i = -\hat{m}\tilde{x}_i - \hat{c} + \tilde{y}_i$. Note that in

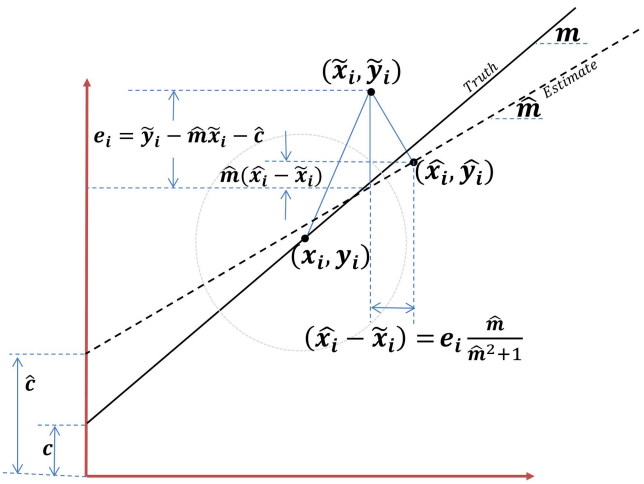


Fig. 6. Optimal estimates of coordinates.

Equations (43) and (44), the estimate for the slope and intercept must be known in order to calculate $\hat{x}_i^2$ and $\hat{y}_i^2$. In our scenario these initial estimates are given by the basic Least Squares solution rather than the solution of the Hough transform. This is to preclude singular estimates for the slope and intercept:

$$\hat{m} = -\frac{\cos\theta}{\sin\theta} \tag{45}$$

$$\hat{c} = \frac{\rho}{\sin\theta} \tag{46}$$

if the Hough transform returns a line estimate with θ exactly zero. Note that the Hough transform is used to identify the measurements which are associated with the line which is being estimated.

With the initial estimates for the slope and intercept, one can update the estimate of the true value $\hat{x}_i$ using Equation (43). In addition to the estimated true value of x, the line's coefficients are updated using the Total Least Squares solution. The next step is to determine the uncertainty in these estimates. The probability density function for (x_i, y_i), given the estimates $[\hat{m}, \hat{c}, \hat{x}_i]$ is:

$$\begin{aligned} &p(\tilde{x}_i, \tilde{y}_i \mid \hat{m}, \hat{c}, \hat{x}_i) \\ &\quad = \frac{1}{2\pi\sqrt{|\mathbf{Q}_i|}} \exp\left[-\frac{1}{2}\left(\begin{bmatrix}\tilde{y}_i \\ \tilde{x}_i\end{bmatrix} - \begin{bmatrix}\hat{m}\hat{x}_i + \hat{c} \\ \hat{x}_i\end{bmatrix}\right)^T \right. \\ &\qquad \left. \times \mathbf{Q}_i^{-1}\left(\begin{bmatrix}\tilde{y}_i \\ \tilde{x}_i\end{bmatrix} - \begin{bmatrix}\hat{m}\hat{x}_i + \hat{c} \\ \hat{x}_i\end{bmatrix}\right)\right]. \end{aligned} \tag{47}$$

The inverse of the covariance matrix $\mathbf{Q}_i$ is easily defined for a 2×2 matrix:

$$\mathbf{Q}_i^{-1} = \frac{1}{\sigma^2_{\tilde{x}_i\tilde{x}_i}\sigma^2_{\tilde{y}_i\tilde{y}_i} - \sigma^2_{\tilde{x}_i\tilde{y}_i}} \begin{bmatrix} \sigma^2_{\tilde{y}_i\tilde{y}_i} & -\sigma_{\tilde{x}_i\tilde{y}_i} \\ -\sigma_{\tilde{x}_i\tilde{y}_i} & \sigma^2_{\tilde{x}_i\tilde{x}_i} \end{bmatrix}. \tag{48}$$

We expand the exponential term in Equation (47) and write it as a product of two variables α_i and K_i which are defined in Equations (49) and (50), respectively:

$$\alpha_i = -\frac{1}{2(\sigma^2_{\tilde{x}_i\tilde{x}_i}\sigma^2_{\tilde{y}_i\tilde{y}_i} - \sigma^2_{\tilde{x}_i\tilde{y}_i})} \tag{49}$$

$$\begin{aligned} K_i = {} & \sigma^2_{\tilde{x}_i\tilde{x}_i}(\tilde{y}_i - \hat{m}\hat{x}_i - \hat{c})^2 - 2\sigma_{\tilde{x}_i\tilde{y}_i}(\tilde{y}_i - \hat{m}\hat{x}_i - \hat{c})(\tilde{x}_i - \hat{x}_i) \\ & + \sigma^2_{\tilde{y}_i\tilde{y}_i}(\tilde{x}_i - \hat{x}_i)^2. \end{aligned} \tag{50}$$

With α_i and K_i defined for each measurement we can then rewrite the likelihood function in a more contracted form as:

$$p(\tilde{x}_i, \tilde{y}_i \mid \hat{m}, \hat{c}, \hat{x}_i) = \frac{1}{2\pi\sqrt{|\mathbf{Q}_i|}} e^{\alpha_i K_i}. \tag{51}$$

For each measurement, there is a corresponding (x_i, y_i), independent of one another. Therefore if we assume there are a total of M measurements, the probability density function for the matrix of measurements, $[\tilde{\mathbf{x}}, \tilde{\mathbf{y}}]$, given the parameter estimates, $[\hat{m}, \hat{c}, \hat{\mathbf{x}}]$, where $\hat{\mathbf{x}}$ is now a vector of estimated x values, is given by the product of each measurement's probability density function:

$$p(\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \mid \hat{m}, \hat{c}, \hat{\mathbf{x}}) = \prod_{i=1}^{M} \frac{1}{2\pi\sqrt{(\sigma^2_{\tilde{x}_i\tilde{x}_i}\sigma^2_{\tilde{y}_i\tilde{y}_i} - \sigma^2_{\tilde{x}_i\tilde{y}_i})}} e^{\alpha_i K_i} \tag{52}$$

The Fisher Information matrix is defined as the negative expected value of the Hessian of the log-likelihood function with respect to the estimated parameters. We define the log-likelihood function as, $f = \ln[p(\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \mid \hat{m}, \hat{c}, \hat{\mathbf{x}})]$, therefore the Fisher Information matrix is defined as:

$$F = -E\begin{bmatrix} \frac{\partial f}{\partial\hat{m}\partial\hat{m}} & \frac{\partial f}{\partial\hat{m}\partial\hat{c}} & \left(\frac{\partial f}{\partial\hat{m}\partial\hat{\mathbf{x}}}\right)^T \\ \frac{\partial f}{\partial\hat{c}\partial\hat{m}} & \frac{\partial f}{\partial\hat{c}\partial\hat{c}} & \left(\frac{\partial f}{\partial\hat{c}\partial\hat{\mathbf{x}}}\right)^T \\ \frac{\partial f}{\partial\hat{\mathbf{x}}\partial\hat{m}} & \frac{\partial f}{\partial\hat{\mathbf{x}}\partial\hat{c}} & \frac{\partial f}{\partial\hat{\mathbf{x}}\partial\hat{\mathbf{x}}} \end{bmatrix} \tag{53}$$

where

$$\frac{\partial f}{\partial\hat{m}\partial\hat{\mathbf{x}}} = \frac{\partial f}{\partial\hat{\mathbf{x}}\partial\hat{m}} = \left[\frac{\partial f}{\partial\hat{m}\partial\hat{x}_1}, \frac{\partial f}{\partial\hat{m}\partial\hat{x}_2}, \ldots, \frac{\partial f}{\partial\hat{m}\partial\hat{x}_M}\right]^T$$

and similarly $\partial f/\partial\hat{c}\partial\hat{\mathbf{x}} = \partial f/\partial\hat{\mathbf{x}}\partial\hat{c}$. The sub-block $\partial f/\partial\hat{\mathbf{x}}\partial\hat{\mathbf{x}}$ of the Fisher Information matrix is a $M \times M$ diagonal matrix with diagonal elements:

$$\frac{\partial f}{\partial\hat{\mathbf{x}}\partial\hat{\mathbf{x}}} = \mathrm{Diag}\left(\left[\frac{\partial f}{\partial\hat{x}_1\partial\hat{x}_1}, \frac{\partial f}{\partial\hat{x}_2\partial\hat{x}_2}, \ldots, \frac{\partial f}{\partial\hat{x}_M\partial\hat{x}_M}\right]\right).$$

First the partial derivatives of f are taken with respect to the estimated parameters. Recall that the probability density function is a product of the individual measurement's and with the properties that the log of

the product becomes a summation over the M measurements. Thus the partial derivatives are:

$$\frac{\partial f}{\partial \hat{m}} = \sum_{i=1}^{M} \alpha_i [-2\sigma_{\tilde{x}_i\tilde{y}_i}(-\tilde{x}_i\hat{x}_i + \hat{x}_i^2) + \sigma^2_{\tilde{x}_i\tilde{x}_i}(-2\tilde{y}_i\hat{x}_i^{+}2\hat{m}\hat{x}_i^2 + 2\hat{x}_i\hat{c})] \quad (54)$$

$$\frac{\partial f}{\partial \hat{c}} = \sum_{i=1}^{M} \alpha_i [-2\sigma_{\tilde{x}_i\tilde{y}_i}(-\tilde{x}_i + \hat{x}_i) + \sigma^2_{\tilde{x}_i\tilde{x}_i}(-2\tilde{y}_i + 2\hat{m}\hat{x}_i + 2\hat{c})] \quad (55)$$

$$\frac{\partial f}{\partial \hat{x}_{t_i}} = \alpha_i [\sigma^2_{\tilde{y}_i\tilde{y}_i}(-2\tilde{x}_i + 2\hat{x}_i) - 2\sigma_{\tilde{x}_i\tilde{y}_i}(-\tilde{y}_i - \tilde{x}_i\hat{m} + 2\hat{m}\hat{x}_i + \hat{c}) + \sigma^2_{\tilde{x}_i\tilde{x}_i}(-2\tilde{y}_i\hat{m} + 2\hat{m}^2\hat{x}_i + 2\hat{m}\hat{c})]. \quad (56)$$

Then the second derivatives, which compose the Fisher Information matrix, can easily be determined using the equations:

$$E\left[\frac{\partial^2 f}{\partial \hat{m}\partial \hat{m}}\right] = \sum_{i=1}^{M} \alpha_i (2\sigma^2_{x_i x_i}\hat{x}_i^2) \quad (57)$$

$$E\left[\frac{\partial^2 f}{\partial \hat{m}\partial \hat{c}}\right] = \sum_{i=1}^{M} \alpha_i (2\sigma^2_{x_i x_i}\hat{x}_i) \quad (58)$$

$$E\left[\frac{\partial^2 f}{\partial \hat{m}\partial \hat{x}_{t_i}}\right] = E\left[\alpha_i(-2\sigma_{x_i y_i}(-x_i + 2\hat{x}_i) + \sigma^2_{x_i x_i}(-2y_i + 4\hat{m}\hat{x}_i + 2\hat{c}))\right] = \alpha_i(-2\sigma_{x_i y_i}\hat{x}_i + 2\sigma^2_{x_i x_i}\hat{m}\hat{x}_i) \quad (59)$$

$$E\left[\frac{\partial^2 f}{\partial \hat{c}\partial \hat{c}}\right] = \sum_{i=1}^{M} 2\alpha_i\sigma_{x_i x_i} \quad (60)$$

$$E\left[\frac{\partial^2 f}{\partial \hat{x}_{t_i}\partial \hat{c}}\right] = \alpha_i(-2\sigma_{x_i y_i} + 2\sigma^2_{x_i x_i}\hat{m}), \quad i = 1,2,\ldots,M \quad (61)$$

$$E\left[\frac{\partial^2 f}{\partial \hat{x}_{t_i}\partial \hat{x}_{t_i}}\right] = \alpha_i(2\sigma^2_{y_i y_i} - 4\sigma_{x_i y_i}\hat{m} + 2\sigma^2_{x_i x_i}\hat{m}^2), \quad i = 1,2,\ldots,M \quad (62)$$

$$E\left[\frac{\partial^2 f}{\partial \hat{x}_{t_i}\partial \hat{x}_{t_j}}\right] = E\left[\frac{\partial^2 f}{\partial \hat{x}_{t_j}\partial \hat{x}_{t_i}}\right] = 0, \quad i \neq j. \quad (63)$$

These equations then give us an estimate of the uncertainty in the estimated parameters. The next step is to perform a Monte Carlo simulation to show the convergence characteristics of this estimate. We begin the simulation by choosing a slope, intercept, and range of x values. These will be the true simulation parameters and are specified as:

$$m = -0.4326 \quad c = -1.6656 \quad x = -3 : 0.1 : 3. \quad (64)$$

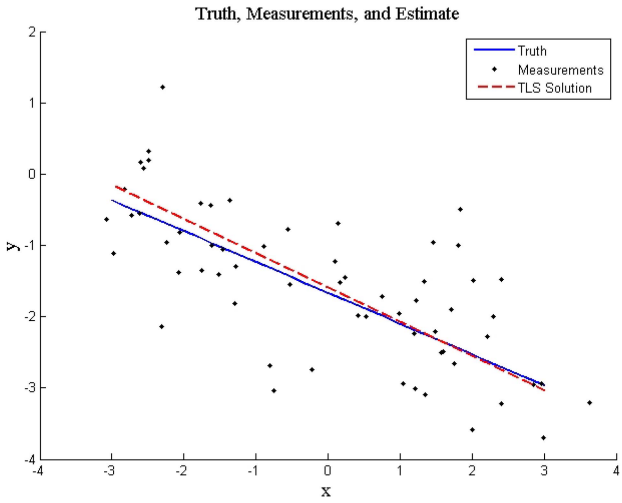


Fig. 7. Truth, measurements, TLS line fit.

Furthermore the covariance matrix, $\mathbf{Q}$ for all measurements is equal and given to be:

$$\mathbf{Q} = \begin{bmatrix} \sigma^2_{yy} & \sigma_{xy} \\ \sigma_{yx} & \sigma^2_{xx} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.0 \\ 0.0 & 0.5 \end{bmatrix}. \quad (65)$$

Since this simulation is to determine the convergence characteristics and not the capabilities of the Hough transform, we will use the Least Squares solution from Equation (36) where the covariance matrix $\mathbf{R}$ is $\sigma^2_{yy}\mathbf{I}_{M\times M}$ and M is the number of measurements, which in this simulation is 61. Since the x truth was already established the y truth can be calculated using the given values for the true slope and intercept. Gaussian white noise is then added to the truth, which was specified in $\mathbf{Q}$ and finally the estimated values of x and y can be obtained via Equations (43) and (44) respectively.

A single simulation's results are shown in Figure 7. Here the truth is shown by the solid blue line, the measurements (truth with added noise) are black dots, and the Total Least Squares line fit is represented by the dashed red line. The estimated values of the slope and intercept from the Total Least Squares algorithm, for this single simulation run are:

$$\hat{m} = -0.3775 \quad \hat{c} = -1.5871. \quad (66)$$

We perform 10,000 simulations to determine the convergence characteristics of the Fisher Information matrix. The measure used for convergence is the determinant of the difference of the Monte Carlo covariance and the inverse of the Fisher Information matrix. The Monte Carlo covariance is calculated as a difference of the truth and the averaged estimates. We will denote this covariance as MC_{cov} and is calculated as:

$$MC_{cov} = \frac{1}{N}\sum_{i=1}^{N}\left(\begin{bmatrix} m \\ c \end{bmatrix} - \begin{bmatrix} \hat{m}_i \\ \hat{c}_i \end{bmatrix}\right)\left(\begin{bmatrix} m \\ c \end{bmatrix} - \begin{bmatrix} \hat{m}_i \\ \hat{c}_i \end{bmatrix}\right)^T \quad (67)$$

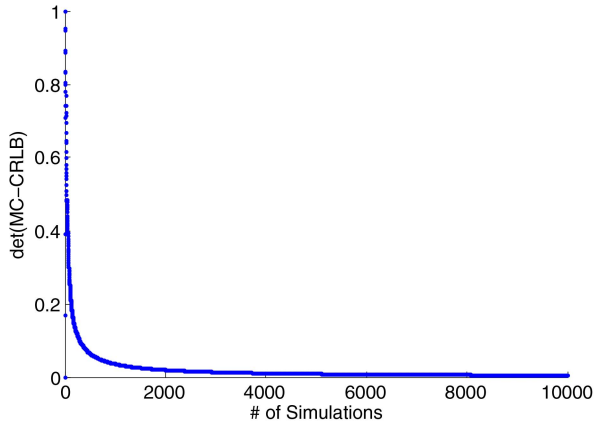


Fig. 8. Monte Carlo simulation.

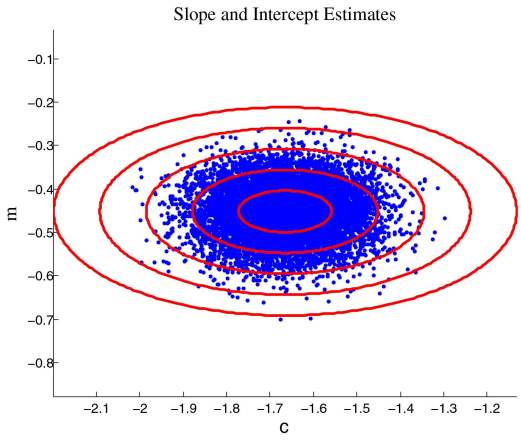


Fig. 9. Sigma ellipses.

where N is the number of Monte Carlo runs and the estimated values m_i, c_i, are used to estimate the covariance. Then the convergence measure is given by the equation:

$$convergence = |MC_{cov} - F^{-1}|. \tag{68}$$

The Fisher Information matrix is also averaged over each simulation. Figure 8 shows the value of this *convergence* measure after each simulation.

Figure 8 shows only a portion of the total number of simulations and in addition the value of the convergence parameter has been normalized. The estimated bounds on the parameters, the inverse of the negative of the Fisher Information matrix after 10,000 simulations is given to be:

$$F^{-1} = 1e^{-4}\begin{bmatrix} 0.14577 & 0.00065 \\ 0.00065 & 0.59065 \end{bmatrix}.$$

Next we examine the estimates and their statistical properties. Each of the estimated values of m and c are plotted in Figure 9 along with the sigma ellipses.

Finally we select all combinations of $\hat{m}$ and $\hat{c}$ which fall within one sigma of the average values of the respective estimates and plot them, where the range is

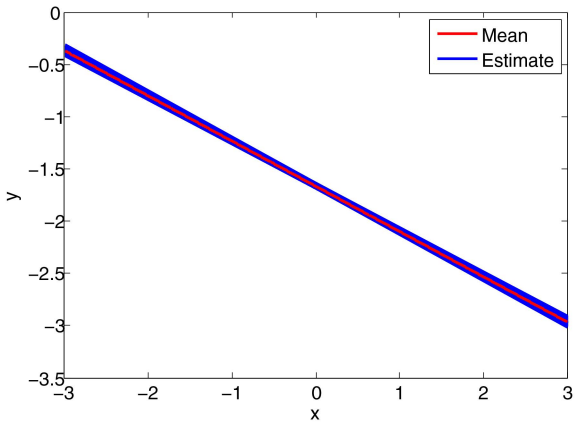


Fig. 10. One sigma slopes/intercepts.

given as:

$$\hat{m} = -0.4326 \pm 0.00382 \tag{69}$$

$$\hat{c} = -1.6648 \pm 0.0077. \tag{70}$$

The lines defined by all such coefficients are plotted, in blue, along with the true fit of the line, in red, in Figure 10 and we can see the uncertainty in the estimates. As we diverge from the relative midpoint of the data range the uncertainty grows. This is to be expected since the variance in the y direction depends on the variance of the slope, intercept, and estimated x value. To prove that the variance in the y direction is dependent on the variance of the slope, intercept, and estimated x value we can transform the Fisher Information matrix from the model parameters into a variance in terms of x and y. This can be done using the Jacobian transformation. In this transformation the Fisher Information matrix is left and right multiplied by the Jacobian of the measurement functions with respect to the estimated parameters. The Jacobian takes the form of:

$$A = \begin{bmatrix} \frac{dy}{d\hat{m}} & \frac{dy}{d\hat{c}} & \frac{dy}{d\hat{x}_i^2} \\ \frac{dx}{d\hat{m}} & \frac{dx}{d\hat{c}} & \frac{dx}{d\hat{x}} \end{bmatrix} = \begin{bmatrix} \hat{x}_i & 1 & \hat{m} \\ 0 & 0 & 1 \end{bmatrix}. \tag{71}$$

Therefore we perform the matrix multiplication to understand the growing variance in the y direction:

$$\begin{aligned} &\begin{bmatrix} \sigma^2_{yy_i} & \sigma_{xy_i} \\ \sigma_{xy_i} & \sigma^2_{xx_i} \end{bmatrix} \\ &\quad = A_i F^{-1} A_i^T \\ &\quad = \begin{bmatrix} \hat{x}_i & 1 & \hat{m} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma^2_{mm} & \sigma_{mc} & \sigma_{mx_t} \\ \sigma_{cm} & \sigma^2_{cc} & \sigma_{cx_t} \\ \sigma_{x_t m} & \sigma_{x_t c} & \sigma^2_{x_t x_t} \end{bmatrix} \begin{bmatrix} \hat{x}_i & 0 \\ 1 & 0 \\ \hat{m} & 0 \end{bmatrix}. \end{aligned} \tag{72}$$

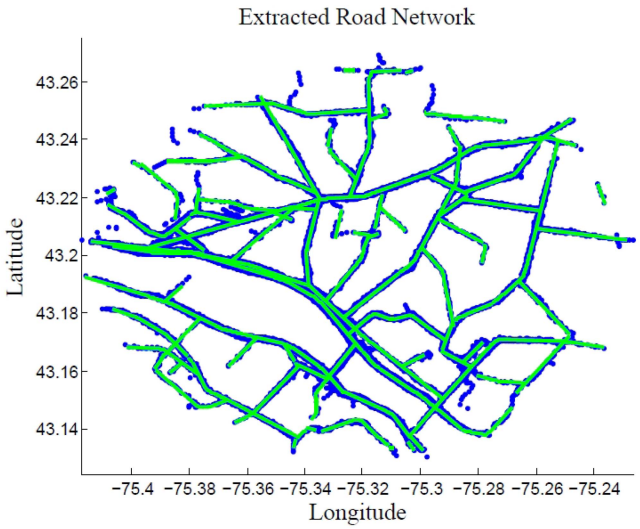


Fig. 11. Data set #2 original measurements and extracted road network.

This results in the covariances taking the form of Equations (73) through (76)

$$\sigma^2_{y y_i} = \hat{x}_i^2 \sigma^2_{mm} + 2\hat{x}_i \sigma_{mc} + 2\hat{m}\hat{x}_i \sigma_{mx} + \sigma^2_{cc} + 2\hat{m}\sigma_{xc} + \hat{m}^2 \sigma^2_{xx} \tag{73}$$

$$\sigma_{x y_i} = \hat{x}_i \sigma_{mx} + \sigma_{cx} + \hat{m}\sigma^2_{xx} \tag{74}$$

$$\sigma_{y x_i} = \sigma_{x y_i} \tag{75}$$

$$\sigma^2_{x x_i} = \sigma^2_{xx}. \tag{76}$$

From the above equations we can see that the variance in the y direction depends on the varying value of $\hat{x}_i$, so therefore as we diverge from $\hat{x}_i = 0$ in either direction the variance in y grows.

4 RESULTS

A Graphical User Interface (GUI) was developed, which allows the user to easily manipulate the processed data. The automated phase of this process consists of the Hough transform and the extraction of the linear features. Several additional operations are allowed by manual interaction, which include merging, trimming, extending, blending, removal, and ellipse identification. The merging option allows the user to select two lines which identify the same road segment. This is possible due to the threshold placed on the distance from a point to the line segment. The trimming and extending features allow a user to smooth out the road network and connect each and every segment with another. A 3rd order polynomial is used as the blending function between two lines which are selected by the user. This updates the road network with curves which are defined by the last xx% of each of the two line segments selected (percentage decided by the user). The final feature, the ellipse identification, is left up to the user due to the computational complexity correlated with automatic identification of an ellipse in an image. The ellipse identification only requires the user to define the number of line segments that are associated with the ellipse and then select each of these segments. This allows for a fast and simple identification of an ellipse.

The data which has been supplied consists of simulated GMTI tracks generated by the Air Force Research Labs in Rome, NY. Each track varies in the number of measurements it contains, however, the structure of each track is consistent. The data structure is broken down as follows:

- *tracks*—main field of the structure.
  - —*tracks(#).loc*—N × 2 matrix of latitude and longitude coordinates.
  - —*tracks(#).cov*—4 × 4 × N covariance matrices corresponding to each measurement of the form.

$$\begin{bmatrix} \sigma^2_{x_i x_i} & \sigma_{x_i y_i} & \sigma_{x_i \dot{x}_i} & \sigma_{x_i \dot{y}_i} \\ \sigma_{y_i x_i} & \sigma^2_{y_i y_i} & \sigma_{y_i \dot{x}_i} & \sigma_{y_i \dot{y}_i} \\ \sigma_{\dot{x}_i x_i} & \sigma_{\dot{x}_i y_i} & \sigma^2_{\dot{x}_i \dot{x}_i} & \sigma_{\dot{x}_i \dot{y}_i} \\ \sigma_{\dot{y}_i x_i} & \sigma_{\dot{y}_i y_i} & \sigma_{\dot{y}_i \dot{x}_i} & \sigma^2_{\dot{y}_i \dot{y}_i} \end{bmatrix}$$

  - —*tracks(#).vel*—N × 2 matrix of component velocities at the same time the measurement is taken.
  - —*tracks(#).update*—unix time representation of measurement time (seconds since January 1, 1970).

The data set that was used to test the proposed algorithm includes 1,675 tracks, where every track had differing numbers of kinematic data.

The data set was completely processed without manual intervention (i.e., merging or ellipse finding). This data set contained several areas of interest which may create issues which include sectors with no track data. Primarily we expect there to be a significant increase in the number of extracted segments.

The line extraction portion of the algorithm took approximately 25 minutes to complete. The extracted line segment data structure was stored before any additional functions were implemented. Prior to any removal or merging of line segments, the data structure contains 150 individual line segments. The merging, blending, and trimming/extending was complete in approximately 3 hours. This data set consisted of 1,675 tracks with a total of 88,685 measurements.

The extracted network was converted back to the latitude longitude coordinate system and the results of the extracted network plotted on top of the original measurements is shown in Figure 11. Although it appears that there is a low association due to the lack of identified segments in certain regions, there are actually very few data points in these areas, 99.8% of the data has been associated with geometric features in the image (88,508/88,685). The final data structure consists of 0 ellipses, 72 third order polynomials, and 131 line segments. We note that some areas are lacking extracted features which is due to either the lack of data or the user removed a feature due to inaccuracy.

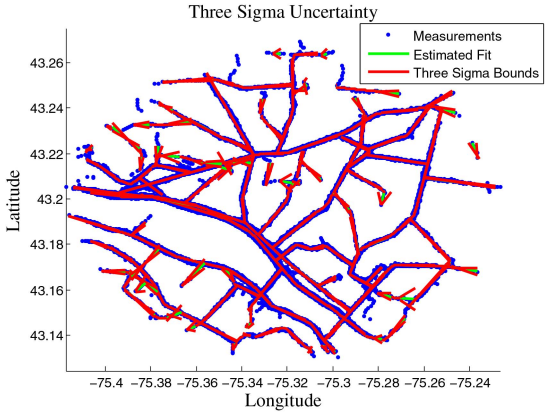


Fig. 12. Data set #2 line CRLB.

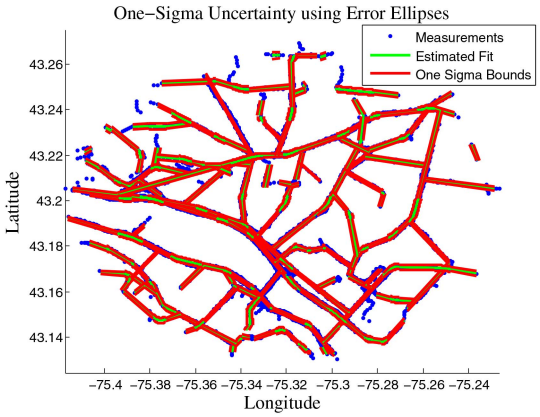


Fig. 13. Data set #2 error ellipses one sigma.

First we present the results for the estimated CRLB matrix corresponding to the line estimate's parameters. Figure 12 shows the three sigma bounded region for the lines, which again due to the number of measurements associated with each of the lines, converges to small values for most lines. For a few straight line segments which have very few associated measurements, the bounds are very lax.

We supplement Figure 12 with an example of a line's CRLB estimate. Equation (77) refers to a line which has 1,034 measurements associated with it:

$$\begin{bmatrix} \sigma^2_{mm} & \sigma_{mc} & \sigma_{mx} \\ \sigma_{cm} & \sigma^2_{cc} & \sigma_{cx} \\ \sigma_{xm} & \sigma_{xc} & \sigma^2_{xx} \end{bmatrix} = \begin{bmatrix} 2.2541e-05 & 3.5129e-09 & -1.7855e-09 \\ 3.5129e-09 & 3.6176e-10 & -1.2573e-10 \\ -1.7855e-09 & -1.2573e-10 & 1.4242e-10 \end{bmatrix} \tag{77}$$

which illustrates a high degree of confidence in the estimated model parameters. We must then use the Jacobian transformation to obtain the covariance in terms of x and y. A single measurement's covariance matrix is given by the equation:

$$\begin{bmatrix} \sigma^2_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma^2_{yy} \end{bmatrix} = 1.0e-06 \times \begin{bmatrix} 0.3850 & -0.4209 \\ -0.4209 & 0.8059 \end{bmatrix}. \tag{78}$$

The Jacobian transformed CRLB is then added to the measurement covariance defined in Equation (78) to produce the final covariance in the measurement including the covariance in the estimate given by Equation (79):

$$\begin{bmatrix} \sigma^2_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma^2_{yy} \end{bmatrix} = 1.0e-06 \times \begin{bmatrix} 0.3852 & -0.4209 \\ -0.4209 & 0.8089 \end{bmatrix}. \tag{79}$$

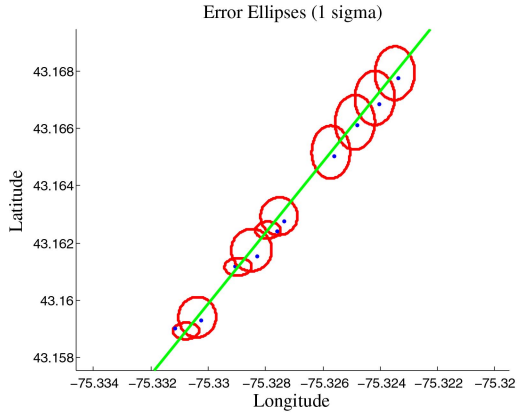


Fig. 14. Data set #2 mean line uncertainty.

Figure 13 shows a few measurement's one sigma ellipses with the line estimate.

Using the error ellipses we once again compute the mean covariance in x and y. This covariance is then used to represent a mean error in the line using one sigma we can see from Figure 14 that this encompasses the majority of the data associated with each of the line estimates.

There are no ellipses identified in this data set but there are a significant number of polynomial blends. A polynomial blend in this data set typical contains from 400 to 1,000 measurements. We present the results for a polynomial, which contains 1,032 associated measurements in Equation (80):

$$\begin{bmatrix} \sigma^2_{AA} & \sigma_{AB} & \sigma_{AC} & \sigma_{AD} \\ \sigma_{BA} & \sigma^2_{BB} & \sigma_{BC} & \sigma_{BD} \\ \sigma_{CA} & \sigma_{CB} & \sigma^2_{CC} & \sigma_{CD} \\ \sigma_{DA} & \sigma_{DB} & \sigma_{DC} & \sigma^2_{DD} \end{bmatrix} = \begin{bmatrix} 3.9162e-12 & 2.9909e-10 & -2.0009e-08 & -1.5304e-06 \\ 2.9909e-10 & 2.2842e-08 & -1.5281e-06 & -0.0001 \\ -2.0009e-08 & -1.5281e-06 & 0.0001 & 0.0078 \\ -1.5304e-06 & -0.0001 & 0.0078 & 0.5980 \end{bmatrix}. \tag{80}$$

5 CONCLUSION

This paper presents a systematic approach for the estimation of the road network from ground moving target data assuming the vehicles to be road based vehicles. The position locations of the multiple ground vehicles are used to generate a binary images, which permits using the Hough transform to identify straight line segments. Having identified the Hough parameters associated with various straight line segments, the position data associated with each of these straight line segments is identified and used by a least squares algorithm to identify the parameters of a straight line, i.e., slope and intercept. Since the two dimensional position data is contaminated with noise in both the dimensions, the total least squares is deemed the appropriate approach for the estimation of the slope and intercept of the straight line segments. A detailed exposition of the total least squares approach for the estimation of the model parameters is followed by the determination of the Cramér-Rao bounds on the estimated parameters. A multi road network with numerous intersections is used as a test case to illustrate the potential of the proposed approach to estimate the road network and characterize the associated uncertainty. Currently the proposed approach is being extended to estimating curved section of roads which are assumed to be representable by arcs of ellipses. This will permit a parsimonious representation of the road-network by eliminating numerous straight line segments which are necessary to represent curved sections of roads. A graphical user interface was also developed to permit a seamless estimation of road-network from ground vehicle kinematic data.

ACKNOWLEDGEMENT

This material is based upon work supported by the AFRL under Cooperative Agreement No. FA 8750-11-2-0082.

REFERENCES

[1] Tupin, Florence, Henri Maitre, Jean-Francois Mangin, Jean-Marie Nicolas, and Eugene Pechersky
Detection of Linear Features in SAR Images: Application to Road Network Extraction.
IEEE Transactions on Geoscience and Remote Sensing 36.2 (1998).

[2] Koch, W., J. Koller, and M. Ulmke
Ground Target Tracking and Road Map Extraction.
ISPRS Journal of Photogrammetry and Remote Sensing 61.3–4 (2006): 197–208.

[3] P. Gamba, F. Dell'Acqua, and G. Lisini
Improving urban road extraction in high-resolution images exploiting directional filtering, perceptual grouping, and simple topological concepts,
IEEE Geosci. Remote Sens. Lett., vol. 3, no. 3, pp. 387–391, Jul. 2006.

[4] V. Amberg, M. Coulon, P. Marthon, and M. Spigai
Structure extraction from high resolution SAR data on urban areas,
in *Proc. IEEE Geosci. Remote Sens. Symp.*, 2004, vol. 3, pp. 1784–1787.

[5] Hu, Jiuxiang, Anshuman Razdan, John C. Femiani, Ming Cui, and Peter Wonka
Road Network Extraction and Intersection Detection From Aerial Images by Tracking Road Footprints.
IEEE Transactions on Geoscience and Remote Sensing 45.12 (2007): 4144–4157.

[6] Aaron, Shackelford K., and Curt H. Davis
Urban Road Network Extraction from High-Resolution Multispectral Data.
2nd GRSS/ISPRS Joint Workshop on Remote Sensing and Data Fusion over Urban Areas, 2003, p. 142–146.

[7] Sklarz, S. E., Novoselsky A., and Dorfan M.
Incremental Fusion of GMTI Tracks for Road Map Estimation.
2008 11th International Conference on Information Fusion, Cologne, Germany, pp. 1–7.

[8] Halir, R. and Flusser, J.
Numerically stable direct least squares fitting of ellipses.
The Sixth International Conference in Central Europe. 1998.

[9] Duda, R. O. and P. E. Hart
Use of the Hough Transformation to Detect Lines and Curves in Pictures,
Comm. ACM, Vol. 15, pp. 11–15 (January, 1972).

[10] Guil, N., and E. L. Zapata
Lower Order Circle and Ellipse Hough Transform.
Pattern Recognition 30.10 (1997): 1729–1744.

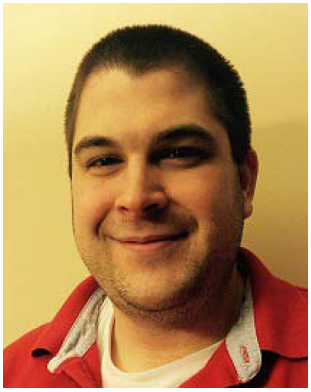
[11] Tsuji, Saburo, and Fumio Matsumoto
Detection of Ellipses by a Modified Hough Transformation.
IEEE Transactions on Computers C-27.8 (1978): 777–781.

[12] Aguado, A. S., and M. S. Nixon
"A New Hough Transform Mapping for Ellipse Detection."
Technical Report, University of Southampton, 1995.

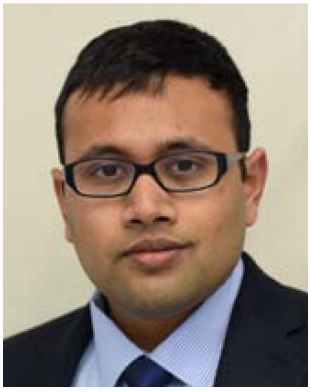
[13] Crassidis, John L., and Yang Cheng
Error-Covariance Analysis of the Total Least Square Problem.
Proceedings of the AIAA Guidance, Navigation and Control Conference, Portland, Oregon, August 8–11, 2011.

[14] Fitzgibbon, A. W. and Fischer, R. B.
A buyer's guide to conic fitting.
Proc. of the British Machine Vision Conference. pp. 265–271. 1995.

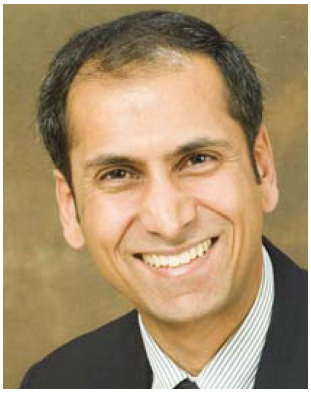
[15] Weisstein, Eric W.
"Ellipse."
From MathWorld—A Wolfram Web Resource.
http://mathworld.wolfram.com/Ellipse.html, Web. 18 Nov. 2014.



Eric Salerno received his Masters of Science in Mechanical Engineering from the State University of New York at Buffalo (Buffalo, New York) in 2013 where he focused his studies on estimation and control systems. He completed two internships with the Air Force Research Laboratory in Rome, New York where he worked on test bedding Natural Language Processing algorithms and developing a Bayesian model for the estimation of remaining useable life for cellular devices in combat zones. He is currently pursuing a second Master's of Science in Computer Science from the State University of New York Polytechnic Institute (Utica, New York) as well as his Certified Information Systems Security Professional certification and currently works at the Department of Defense.



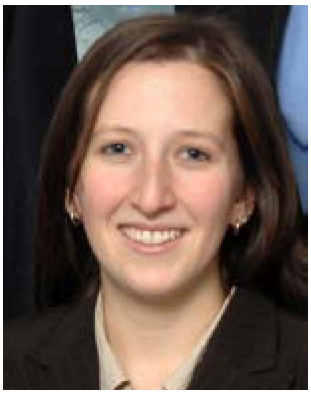
Nagavenkat Adurthi is a graduate student at the University at Buffalo, pursuing a Ph.D. degree. He has received his Bachelors from the Indian Institute of Technology at Guwahati in 2010 and a Master's degree from the University at Buffalo in 2013. He was recognized with the NAGS Distinguished Master's Thesis Award for the work on "The Conjugate Unscented Transform—An Approach to Evaluate Multidimensional Integrals." His research interests include the fields of Uncertainty Quantification, Information Theory and State Estimation.



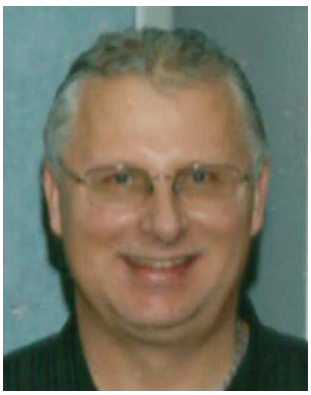
Tarunraj Singh is a fellow of ASME and AAAS. He received his B.E. degree from Bangalore University, Bangalore, India, a M.E. degree from the Indian Institute of Science, Bangalore, and his Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, all in mechanical engineering. He was a Postdoctoral Fellow with the Aerospace Engineering Department, Texas A&M University, College Station. Since 1993, he has been with the University at Buffalo, Buffalo, NY, where he is currently a Professor with the Department of Mechanical and Aerospace Engineering. He was a von Humboldt Fellow and spent his sabbatical at the IBM Almaden Research Center in 2000{2001 and at RWTH Aachen in 2011. He was a National Aeronautics and Space Administration Summer Faculty Fellow with the Goddard Space Flight Center in 2003. His research has been supported by the National Science Foundation, Air Force Office of Scientific Research, National Security Agency, Office of Naval Research, and various industries, including MOOG Inc. Praxair and Delphi Thermal Systems. He has published more than 200 refereed journal and conference papers and has presented over 50 invited seminars at various universities and research laboratories. His research interests are in robust vibration control, estimation, and intelligent transportation.

Puneet Singla (Ph.D., 2006, Texas A&M University) is an Associate Professor and director of graduate studies of Mechanical & Aerospace engineering at the University at Buffalo. He has developed a vigorous research program in the general area of estimation and control with specific focus on uncertainty quantification and characterization at University at Buffalo. He has authored over 100 papers to-date including one book and 27 journal articles covering a wide array of problems. He has received competitive NSF CAREER award for his work on Uncertainty Propagation and Data Assimilation for Toxic Cloud Prediction and the AFOSR Young Investigator Award for his work on Information Collection and Fusion for Space Situational Awareness. He is a senior member of the American Institute of Aeronautics and Astronautics (AIAA), and member of the American Astronautical Society (AAS), the American Society for Mechanical Engineers (ASME), the Institute of Electrical and Electronics Engineers (IEEE) and the Society for Industrial and Applied Mathematics (SIAM).

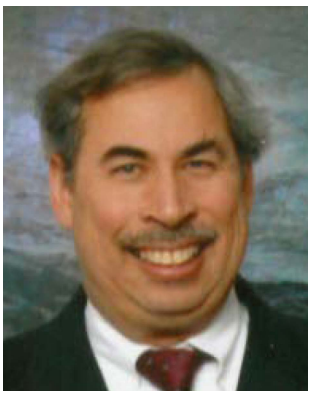
Adnan Bubalo received his M.S. degree in Computer and Information Science from the State University of New York Institute of Technology in 2009. He is presently a Computer Scientist with the Air Force Research Laboratory (AFRL) Information Directorate in Rome, NY. Adnan is the Deputy Lead for the AFRL Processing and Exploitation (PEX) Core Technical Competency (CTC). His research interests include target tracking, information fusion, and clustering.



Maria Cornacchia received her M.S. degree in Computer and Information Science from Syracuse University in 2009. Mrs. Cornacchia is also currently pursuing a Ph.D. in Computer Information Science and Engineering from Syracuse University. She is a Computer Scientist with the Air Force Research Laboratory (AFRL) Information Directorate in Rome, NY. Her research interests include target tracking, information fusion, data mining, and computer vision.



Mark Alford received his M.S. degree in Electrical and Computer Engineering from Syracuse University in 1988. He is presently an Electronics Engineer with the Air Force Research Laboratory (AFRL) Information Directorate in Rome, NY. Mark is the primary in-house program manager for the Activity Based Analysis branch. He has a strong background in fusion and tracking and his research interests include both quantitative and qualitative intelligence analysis and reporting: primarily from chat, video and images.



Eric Jones received his M.S. degree in Electrical Engineering from Syracuse University in 1989. He is presently an Electronics Engineer with the Air Force Research Laboratory (AFRL) Information Directorate in Rome, NY. His research interests include multiple target tracking, nonlinear filtering, and image analysis.