

2013 International Conference on Computational Science

Feature Matching and Adaptive Prediction Models in an Object Tracking DDDAS

Burak UzKent^a, Matthew J. Hoffman^{b*}, Anthony Vodacek^a, John P. Kerekes^a, Bin Chen^a

^a*Chester F. Carlson Center for Imaging Science, 54 Lomb Memorial Dr., Rochester, NY, 14623, USA*

^b*School of Mathematical Sciences, 85 Lomb Memorial Dr., Rochester, NY, 14623, USA*

Abstract

We consider the optical remote sensing tracking problem for vehicles in a complex environment using an adaptive sensor that can take spectral data at a small number of locations. The Dynamic Data-Driven Applications Systems (DDDAS) paradigm is well-suited for dynamically controlling such an adaptive sensor by using the prediction of object movement and its interaction with the environment to guide the location of spectral measurements. The spectral measurements are used for target identification through feature matching. We consider several adaptive sampling strategies for how to assign locations for spectral measurements in order to distinguish between multiple targets. In addition to guiding the measurement process, the tracking system pulls in additional data from OpenStreetMap to identify road networks and intersections. When a vehicle enters a detected intersection, it triggers the use of a multiple model prediction system to sample all possible turning options. The result of this added information is more accurate predictions and analysis from data assimilation using a Gaussian Sum filter (GSF).

© 2013 The Authors. Published by Elsevier B.V.

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science

Keywords: Dynamic Data Driven Application Systems; DDDAS; data assimilation; Target tracking; Feature matching

1. Introduction

Airborne object tracking systems aim to detect and track vehicles in a cluttered urban environment. This is a challenging problem and the DDDAS principles of incorporating additional data as needed and dynamically controlling the location and modality of observations can improve results. It has been accepted that the use of multiple modalities, such as hyperspectral imaging, can better address the tracking problem, but a full hyperspectral sensor results in a dramatic increase in the amount of data that must be transferred, stored, and analyzed. An adaptive hyperspectral sensor has the capability to add the benefits of multiple modalities without dramatically increasing the volume of data. The sensor can be tasked to observe spectrally in targeted areas and

* Corresponding author. Tel.: 585-420-6288; fax: 585-475-6627
E-mail address: mjhsma@rit.edu

the location of the spectral observations can change in each frame to either follow the target or identify background features. Determining where spectral data is needed and dynamically adapting the sampling falls within the framework of a DDDAS.

Spectral data facilitates the identification of targets in a complex environment. Once targets have been identified, their spectral features can be extracted and then used to properly identify tracks at future times. This concept, known as feature tracking, improves the accuracy of a track and makes it possible to reestablish tracks that have been temporarily lost due to obscurations. The downside to hyperspectral imaging is that it results in a dramatic increase in data volume. In this study, we consider an adaptive sensor similar to the RIT Multi-Object Spectrometer, which can perform spectroscopy on selected, individual pixels. The location of the spectral pixels can change each frame, so we use a Gaussian Sum Filter (GSF) data assimilation (DA) system and model predictions to control where spectral measurements are taken. After targets have been identified, the DA system forecasts the movement of the target using a model and uses this prediction—along with the associated uncertainty estimates—to advise the subsequent imaging. We investigate ways to guide the sampling based on the predictions of target location.

More accurate forecast models will improve both prediction and analysis results. Most tracking algorithms employ a constant velocity model, which is a reasonable approximation for normal movement but can be limited when the target is turning. When a target is identified, we bring in additional data—road information from OpenStreetMap—to determine whether the target is traveling along a known road. If so, the airborne image and the OpenStreetMap data are used to identify intersection locations and possible turning paths in those intersections. When the target is predicted to enter an intersection, the constant velocity forecast model is replaced with a multi model system where the individual components of the GSF are assigned different turning models that represent different likely maneuvers. This leads to improved analysis and analysis uncertainty results in the tracking.

2. Tracking System

The basic components of a tracking system are an imaging system, a target detection algorithm for the images, a predictive model for the motion of known targets, and a data assimilation algorithm to combine predictions and observations and evolve the uncertainties.

2.1. DIRSIG Images

To develop and test the system in a controlled environment that allows us a knowable ground truth, we use synthetic imagery generated by the Digital Imaging and Remote Sensing Image Generation (DIRSIG) model. DIRSIG is a first-principles image generation model that computes time and material dependent surface temperature values, incorporates atmospheric contributions using MODTRAN, and predicts bi-directional reflectance functions to render realistic image sets. In addition, the Simulation of Urban Mobility (SUMO) traffic simulator has been integrated with DIRSIG to produce dynamic imagery for tracking scenarios. SUMO has the capability to simulate both vehicular and pedestrian movement, but for this study we consider only vehicular traffic. Different paint models are used for the different vehicles.

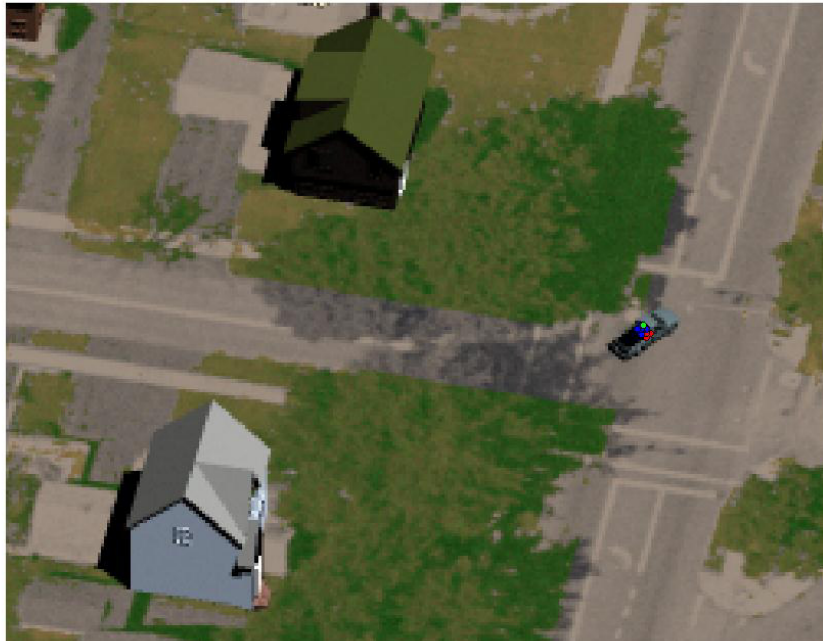


Figure 1: One frame of the DIRSIG simulation showing a truck turning right along with the true center of the truck (green dot) and the analysis results using a constant velocity model (red) and multiple turning models (blue). The multiple turning models gives better results.

The motivation for using synthetic data is that, since we know the true positions and characteristics in a synthetic image, we can accurately compute performance metrics for the tracking system. Furthermore, multiple scenarios and sampling strategies within those scenarios can also be carried out without running multiple experiments. The scenario which is used in this paper comes from DIRSIG Megascene I, which is built to resemble part of Rochester, NY, USA. The simulation uses hyperspectral imaging from a fixed aerial platform assuming a static sensor mount. We focus on a piece of this image that contains two vehicles, which pass close by each other after one has executed a turn (Fig. 1). The spectral range here is 400 to 1000 nm with a spectral resolution of 5 nm.

2.2. Target Detection

Two detection methods are employed in the tracking system: Change and RX detection. Change detection differences multiple frames to isolate anomalies. It can be applied to both panchromatic and hyperspectral images. RX detection, on the other hand, searches for regions that are different than the background in a multi or hyperspectral image. Both algorithms give some spurious results when used individually, but when used in concert provide good target detection performance.

Ultimately, RX detection will not be possible in this application because the adaptive sensor only takes hyperspectral data in a small fraction of the pixels. In this study, however, we are focusing on other parts of the tracking system: the data assimilation algorithm, the forecasting model, and the feature matching. For this reason we use RX detection as a way to limit the possible sources of error and to provide a baseline for the performance of the system with full hyperspectral information in detection.

2.2a. RX-Anomaly Detection

RX-anomaly detection uses the Reed-Xiaoli Detector algorithm that detects the spectral differences between a region and its neighboring area [1]. As a result, spectrally distinct areas are differentiated from the image background [2]. The algorithm outputs a matrix of confidences for every single pixel and a predetermined

threshold is applied to these confidence matrices. If the confidence is higher than the predetermined threshold, the corresponding pixel is labeled as an anomaly pixel. Dimension reduction needs to be performed before applying RX detection so Principal Component Analysis is applied to the hyperspectral images to reduce the number of channels from 121 to 24.

2.2b. Change Detection

The change detection method followed in this paper is the Image Differencing method. This technique measures the spectral changes of two frames at two different time periods. Change detection methods do an outstanding job for hyperspectral images since they are spatially and spectrally rich images but are applicable to panchromatic images as well [3]. In Image Differencing, the difference of the two images at two different times is computed to get the confidence matrices for every single pixel. Then, as in the RX-anomaly detection method, a pre-determined threshold is applied to the result. If the confidence is higher than the predetermined threshold, the corresponding pixels are marked as spectrally different from the background.

2.3. Filtering Process

The Gaussian Sum Filter is a nonlinear filter that represents the state probability density function by a finite mixture of Gaussian density kernels. The mean and covariance of these density kernels are updated using a filtering method such as Extended Kalman filter (EKF) or Unscented Kalman filter. In this study, EKF is used to update the states of density kernels. The weights of the density kernels are updated with the classical weight update method. There are other methods that use Bayes' Rule to update the weights appropriately to increase the accuracy of the tracking which will be implemented in the future [5].

For a process X_{k-1} , the total uncertainty associated with the state vector X_k is determined by the probability density function p . Given the measurement data z_k , one can find the posterior distribution for X_k using the GSF. Gaussian density kernels approximate the conditional pdf as

$$p(t, x(t)_k | z_k) = \sum_{n=1}^M w(t)_k N(x(t)_k; \mu(t)_k^n, P(t)_k^n), \quad (4)$$

where $w(t)_k^n \geq 0$ represents the kernels' weights with $\sum_{n=1}^M w(t)_k^n = 1$, while $\mu(t)_k^n, P(t)_k^n$ are the mean and the covariance of the n^{th} Gaussian kernel regarding the first k measurements. Each Gaussian component is propagated using an EKF update and the mean of the posterior pdf and the corresponding covariance matrix can be estimated as

$$\mu(t)_k = \sum_{n=1}^M w(t)_k^n \mu(t)_k^n, \quad (5)$$

$$P(t)_k = \sum_{n=1}^M w(t)_k^n [P(t)_k^n + (\mu(t)_k^n - \mu(t)_k)(\mu(t)_k^n - \mu(t)_k)^T]. \quad (6)$$

In order to get good performance from the GSF, we need to use an appropriate number of Gaussian kernels. On the other hand, to get a contribution from each component, the covariance belonging to these components need to be small enough so that the components around the mean can approximate the true conditional PDF. For this reason, the Gaussian components should be in the vicinity of the $\pm 3\sigma$ of the mean where σ is the initial standard deviation of the covariance matrix. The implementation of the GSF here has been validated by comparisons with results using a standard Kalman filter.

3. Feature Tracking

In complex environments, tracking will often be lost when vehicles move under trees, behind buildings, or near other moving targets. In these cases, it is important to re-establish the track when a new detection occurs, as opposed to treating the re-detected vehicle as a new object. This is often accomplished through feature matching, where the spectral features of the new target are compared with those of past targets. Here, the comparison is made based on the spectral histogram [6,7]. In each frame a vehicle is detected, the spectral histogram of the vehicle is computed and saved. If a new target appears, the spectral features are extracted and then compared with the spectral histograms of all targets in previous frames. If the spectral histogram of the new target closely matches that of a previous vehicle, the new vehicle is regarded as a re-detection of the old target. If no preexisting target is matched, then a new track is initiated.

In order to compute the spectral histogram, the tracking system must adaptively guide the sensor to take the limited spectral observations in a way that will optimally extract that spectral signal of the target. The system will steer the observations to the area where the target is most likely to be, but there are many possible strategies for sampling that region using only a handful of spectral pixels. This limitation arises from the fact that the RITMOS instrument can only take one pixel of spectral data per row or column per frame. We consider three sampling strategies for observing spectral pixels around the most likely location of the target: sampling pixels along the diagonal of the area, sampling pixels along a vertical line through the area, and a random sampling of one pixel per row in the area.

Whatever measure is used for comparing histograms for feature matching must provide a consistent threshold for distinguishing between targets. This means that comparing histograms of the same target from different frames should be distinct from comparing histograms of two different targets. The distinction is made more difficult by the relatively small amount of pixels with spectral sampling. We consider two metrics for comparing spectral histograms in this paper: the Bhattacharyya measure and the Spectral Angle Mapper (SAM).

The Bhattacharyya distance investigates the similarity of the two distributions. The distance between the two distributions is defined as

$$D(y) = \sqrt{1 - b(y)}, \quad (7)$$

where $b(y)$ is the Bhattacharyya coefficient between the reference vector, r , and the test vector, t . The equation for Bhattacharyya coefficient is defined as

$$b(y) = \sum_{i=1}^N \sqrt{r_i t_i}. \quad (8)$$

The SAM method computes the similarity between two spectra by measuring the angular difference of spectral direction the spectra. It is insensitive to the magnitude of brightness since it takes only the vector direction into account. It is defined as

$$SAM(r, t) = \arccos(r^T t / \|r\| \|t\|), \quad (9)$$

We test both methods here to determine which works better with the limited spectral sample of the targets provided by the adaptive sensor. Interestingly, because of the geometry of the truck bed, there is more variance in the spectral histogram of the truck in different frames than in the histograms of other vehicles we observed. Even with this variance, (which has the effect of raising the Bhattacharyya and SAM measures for the truck compared to itself) we are able to distinguish between the two cars from the entire tracked region.

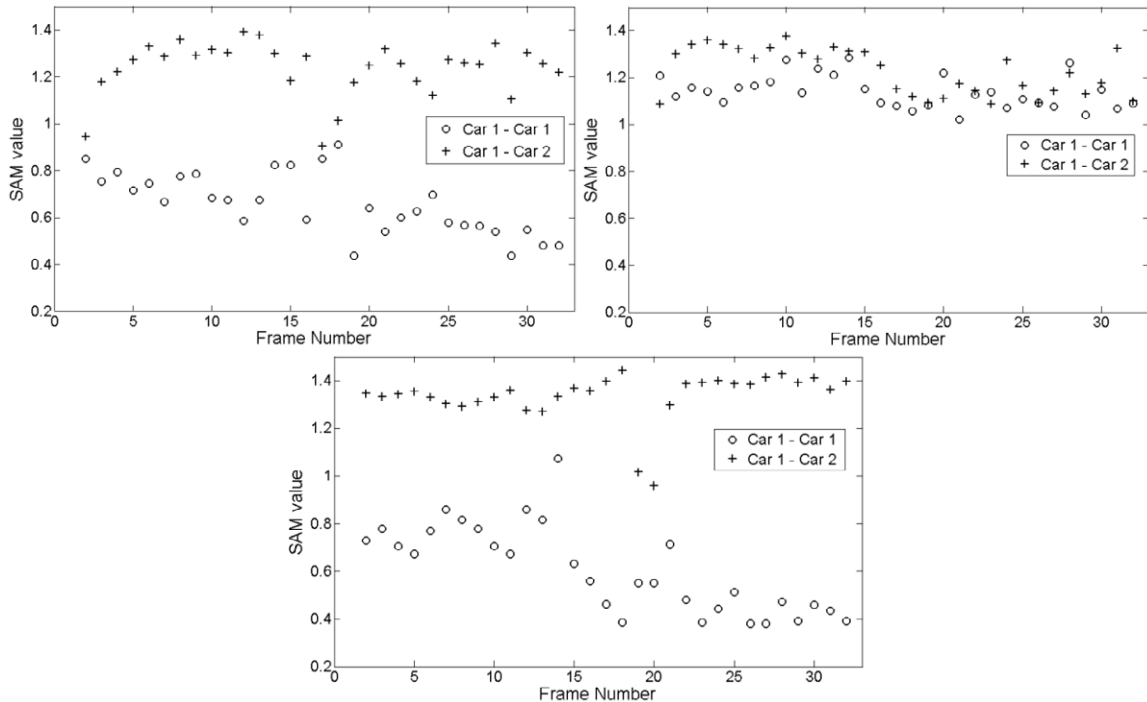


Figure 2: The performance of the SAM measure for the three different sampling methods; (top left) Sampling along the diagonal pixels of the vehicle, (top right) Random Sampling, (bottom) Sampling along the middle column of the vehicle

While the Bhattacharyya measure consistently has lower values when the first vehicle is compared to itself, there are a few frames where the Bhattacharyya measure between car 1 and itself is barely below the measures for car 1 vs. car 2. This is potentially problematic, because it could lead to the misidentification of car 1 if the track was lost. The SAM measure, on the other hand, exhibits larger separation between the values for car 1 compared to itself and car 1 compared to car 2. The difference between the two measures is amplified when one of the adaptive sampling strategies for calculating the spectral histograms is conducted. Based on these results, we use the SAM measure in the tracking code.

After selecting SAM for the feature matching, we next investigate the strategies the system can use for controlling the spectral measurements. Since the RITMOS sensor allows one pixel per row or column of the image to be directed to the spectrometer, we will be not able to obtain spectral information at all of the pixels of the predicted location of the tracked vehicle. The reduction in data—as well as possible inconsistencies in where on the vehicle the spectral information is taken in different frames—makes feature matching more challenging. We test three approaches here, all of which begin with using the predictive model to estimate the most likely location of the target in the next frame. Next, the system will guide the sensor to take spectral measurements either along the diagonal of the predicted vehicle location, at random locations within the predicted location, and along the middle column of the predicted location.



Figure 3: Sample results from the detection of intersections and curvy roads in an image using OpenStreetMap data. Different colors represent different templates that have been matched. The process has identified the major intersections in this image. Notice that the templates try to match the width of the roads as well.

The middle column sampling method outperforms the other two in terms of separating the first car from the second car across most frames (Fig. 2). There is only one place where there is confusion between the two targets: in frame 14. This frame marks the start of the left hand turn that the Car 1 (the truck in Fig. 1) makes. The error is due to the model prediction not accurately capturing the start of the turn. Aside from the start of the turn in frame 14 and the end of the turn in frames 19-20, the middle sampling strategy gives the most consistent similarity scores. Diagonal sampling also does a reasonable job at distinguishing between the two cars, but the SAM values exhibit more variability. One reason that middle column sampling of the predicted vehicle location gives more consistent scores between frames is that the same parts of the vehicle tend to be observed frame to frame. Using random sampling, on the other hand, picks up different parts of the vehicle (and background) in different frames and therefore exhibits significantly worse similarity scores. For many of the frames, the random sampling strategy is unable to distinguish between the two cars (Fig. 2).

It is clear that the similarity scores for the middle (and diagonal) sampling change as the orientation of the vehicle changes. This is reasonable because the middle line will represent a different part of the car when it is traveling east-west to north-south. This hints that the best way to sample in practice will be to adaptively change the sampling axis to try to keep it consistent with the orientation of the car. The tracking system will keep track of the angle of the car, make a prediction of the location and orientation of the car, and then direct the sensor to observe spectral pixels along the line that matches this angle. This adaptive sampling method fits perfectly within the DDDAS framework.

4. Intersection Detection and Multiple Models

One way to improve prediction and analysis performance is to improve the forecast model by adding additional context. Vehicles are more likely to follow road networks, so if we can identify a car and place it on a known road, we can gain better control on the uncertainty for its next location. At intersections, where vehicle

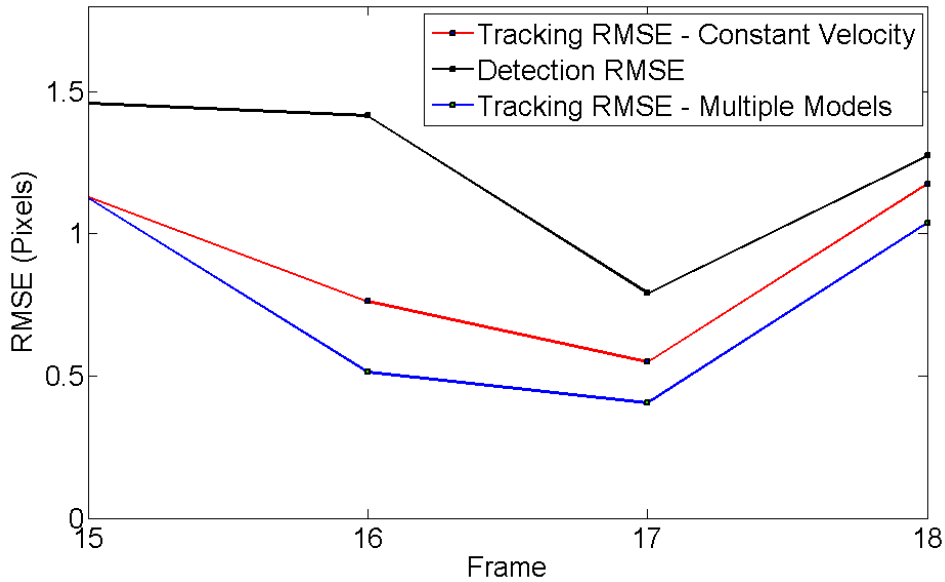


Figure 4: RMSE, during frames where the truck is turning, of the analysis after assimilating the observations (black) using a constant velocity model (red) and multiple turning models (blue).

movement becomes more complex, we can calculate probable paths that the vehicle may take based on the known exits from the intersection.

To identify roads and intersections, we inject OpenStreetMap data into the object tracking system. The OpenStreetMap data are standardized and rasterized, but they are not well registered with the image data due to image distortions, topographical change, inaccurate map survey, and other sources of error. To properly register the image and map data, intersections, end points, and points with high curvature are selected from the map to form templates. Using the map coordinates as a first guess, a search of the extracted image features in the neighborhood of the first guess is used to match the templates. This allows us to find the accurate positions of intersections and curvy roads on the image. To account for different type of roads, different width values are tested during template formation. This process does good job of identifying important intersections in the image that the tracked target might approach (Fig. 3.).

Having identified the location and geometry of the intersections, we then change the prediction model based on whether or not the target is in an intersection. Outside of intersections, we apply a standard constant velocity model for the targets. When a target is determined to enter an intersection, however, the tracking system uses this information to change to a multiple model setup. In the multiple model framework, the geometry of the intersection is used to determine the most probable trajectories that the vehicle might take. For example, a vehicle determined to be entering a T-junction would use variations on models for turning 90° right and 90° left, whereas a vehicle entering a 4-way intersection would have models turning left, right, and going straight. Each component of the GSF is randomly assigned a different turning model while in the intersection and the forecast uncertainty reflects these different possible trajectories. Using more sophisticated weight updates for the GSF components, we should be able to pick out the correct model during the maneuver.

We find that the RMSE is reduced by between 10-20% during the turning frames using the multiple model framework for the truck (seen in Fig. 1) when it enters the intersection (Fig. 4). Moreover, the analysis uncertainty estimate of the target given by the elements of the GSF is more accurate using the multiple models. In many frames, the constant velocity model experiments have the true center of the target (green dot in Fig. 1)

outside the spread of the particles of the GSF (red in Fig. 1). When multiple turning models are used, on the other hand, the analysis uncertainty (blue in Fig. 1) contains the true center. Similar results are seen in other tracking scenarios.

5. Summary

The framework for an object tracking DDDAS has been implemented, consisting of an airborne adaptive sensor capable of taking spectral data in a sparse number of pixels and a Gaussian Sum Filter based data assimilation system. The system brings in additional data, as needed, from OpenStreetMap in order to identify road networks and intersections. The intersection location and geometry is then used change the prediction model to a multiple model framework—where the different models represent different likely turning trajectories—when targets are in intersections

A feature matching system is also implemented and two metrics—Bhattacharyya and SAM measures—are evaluated for distinguishing between new and previously seen vehicles. We determine that the SAM measure is more suitable for this application, considering the small number of spectral pixels available from the adaptive sensor. Using SAM, we compare three different sampling strategies which the tracking system can use when guiding the measurements using the RITMOS instrument. It is found that taking hyperspectral information along the middle column of the predicted target location provided the most consistent and accurate performance in distinguishing between the multiple vehicles. Sampling along a fixed axis can be problematic when the orientation of the tracked vehicle changes, so we have begun implementing an adaptive strategy to match the axis of the sampling with the predicted angle of the vehicle at each frame.

The pixels that take spectral data must be allocated based on the last known location of the vehicle and the model prediction. This results in more background pixels being sampled and, consequently, a reduced performance of the feature matching. Future work will continue to improve the prediction model to improve the selection of which pixels to task as hyperspectral. In addition, we will incorporate knowledge of the background into the decision mechanism. This background information can be collected simultaneously by using hyperspectral information away from targets.

One current strategy to improve prediction and analysis results is using multiple prediction models when the vehicle is in an intersection. From the image and OpenStreetMap data, the geometry of the intersection is found and possible turning arcs are computed. When a vehicle enters the intersection, the different components of the GSF are given different turning models which correspond to our uncertainty in which direction the vehicle will go. In initial testing, the use of multiple models has reduced the analysis RMSE in our tracking and given a better estimate of the analysis uncertainty than when using a single, constant velocity model.

Acknowledgements

This work is supported by AFOSR grants FA9550-11-1-0348 and FA9550-08-1-0028.

References

- [1] - I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 1760–1770, Oct. 1990.
- [2] - Chein I. C., Shao S. C., 2002, "Anomaly Detection and Classification for Hyperspectral Imagery", *IEEE Transactions on Geoscience and Remote Sensing*, Volume:40, No:6.
- [3] - Vongsy K., Karimkhan S., Shaw A. K., Wicker D., 2007, "Change Detection for Hyperspectral Imagery" *Proceedings of the SPIE*, Volume:6565, Pp:656516
- [4] - Li X.R., Jilkov P. V., 2010" Survey of Maneuvering Target Tracking", *IEEE Transactions on Aerospace and Electronic Systems*, Volume:40, No:1.

- [5] - Terejanu G, Singla P, Singh T., Scott P. D., 2011, "Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation", *IEEE Transactions on Automatic Control*, Volume:56, No:9.
- [6] - Nguyen V. H., Banerjee A., Chellappa R., 2010, "Tracking via Object Reflectance using a Hyperspectral Video Camera"; *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp: 44-51.
- [7] – Brown, A., Sullivan, K. J. and Miller, D. J., 2006, "Feature-aided multiple target tracking in the image plane", *Proc. of SPIE*, Vol. 6229.