13th ICCRTS: C2 for Complex Endeavors

VISHON:
Vertical Integration of Sensors providing Human-on-the-loop Optimal kNowledge

modeling and simulation

Amber Fischer and Nathan Denny
Nathan Denny
21st Century Systems, Inc.
6825 Pine St.
Suite 141
Omaha, NE 68106
(402) 502-8439
nathan.denny@21csi.com

VISHON:

Vertical Integration of Sensors providing Human-on-the-loop Optimal kNowledge

Draft 20080316.33

Amber Fischer, Nathan Denny
21st Century Systems, Inc.

Abstract Today's asymmetric adversaries prefer to operate in urban environments where they attack and then dissipate under the cover of "urban fog." Friendly forces may pursue, but armed pursuit may play to the enemy's benefit. Endangering civilians is often an intended secondary effect of asymmetric and 4GW adversaries. Harassing, wounding, or killing civilians, even if unintended, can create a backlash of public sentiment that erodes indigenous support for friendly forces. Thus, it is essential that friendly forces act with high confidence.

Here we introduce VISHON: Vertical Integration of Sensors providing Human-on-the-loop Optimal kNowledge; a simulator modeling a collaborative system of sensors fused to provide vehicle tracking in urban environments. VISHON applies an innovative design for model compression, thus allowing a substantial reduction in processing requirements for the discrete simulation, enabling instantaneous concurrent feedback. Using VISHON, the warfighter can actively engage in solving the problem of sensor allocation and placement in real-time, maximizing confidence in seeking and tracking targets through the urban fog. VISHON will make a significant contribution to warfighters against an asymmetric adversary while minimizing the risk to innocent civilians.

1 INTRODUCTION

A scenario - Three members of Al Qaeda Iraq attempted to plant a bomb at a gathering in the parade grounds west of the Tigris River in Baghdad. An Iraqi Police agent spots the three before they can fully carry out their plans. The terrorists flee as he tries to pursue them. They reach their vehicle, an early model, battered white Mercedes. As the engine coughs then roars, the tires squeal and the terrorists are well on their way toward escaping. While running in pursuit, the agent calls in the vehicle to his commander. Breathless, he breaks off the chase, shaking his head in frustration as he knows, uncaptured, the bombers will bomb again.

Figure 1: Suspects Fleeing

Hoping for the best, but preparing for the worst, the US military had aerial surveillance assets already on station above the gathering. Upon hearing the report of the fleeing terrorists, aerial assets are immediately assigned to track the vehicle with the aim of trailing it until it is clear of civilians where it can be captured or destroyed with little risk to innocent life.

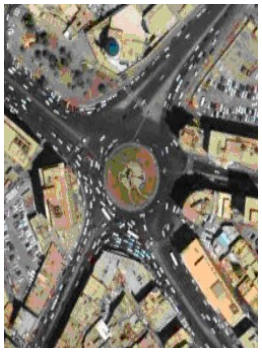
Figure 2: Khulani Square

Aerial reconnaissance tracks the Mercedes as it drives on the Sinak Bridge across the Tigris River (Figure 1). All is going well. As the vehicle turns into the circle at Khulani Square, a communications glitch causes a loss of real time video for several seconds and the white Mercedes harboring the targets, disappears under the "cloak of the urban fog". An operator in charge of the mission scans the re-acquired video stream from the UAV's for the white Mercedes. Which white car has the terrorists (Figure 2)? They must be sure. Apprehending the terrorists may require deadly force. If they begin tracking the wrong vehicle, the occupants of the vehicle may be mistakenly killed by the very people that are trying to protect them. Furthermore, while the UAV operators are tracking the wrong vehicle, the terrorists will get away and almost certainly try to bomb, again.

Assuming the current state-of-the-art in the component technologies were able to meet this challenge, would an operator have enough information in a timely manner to know what resources to request and how to best use those resources to re-acquire and secure the

target?

Once the information on the target has been received, an operator in charge of the mission to re-acquire and secure a target needs to rapidly make a decision on what resources to request for allocation and where, but how does she know? How can she quantify the impact her decisions will have on the probability of success? More importantly, if some resources are unavailable, how will this affect her ability to achieve the goal? Are there alternative resources she can request and will those resources allow her to achieve her goal? She must have confidence for success based on the utilization of a given set of resources to secure the target. In addition, an operator needs to be able to communicate to the C2 that the lack of resource A will affect the probability of success, or be able to stress the importance of being allocated resource B for achieving a minimum confidence in successfully achieving the mission objectives.

The problem space is too complex and dynamic for human operators acting under severe time constraints. What is needed is a decision support system (DSS) that complements the operator. The DSS will provide the fusion of the information and the probability of success – in real time. This capability will substantially improve the odds of capturing fleeing suspects and will save innocent lives.

Following, we describe research in progress towards the development of VISHON: Vertical Integration of Sensors providing Human-on-the-loop Optimal Knowledge.

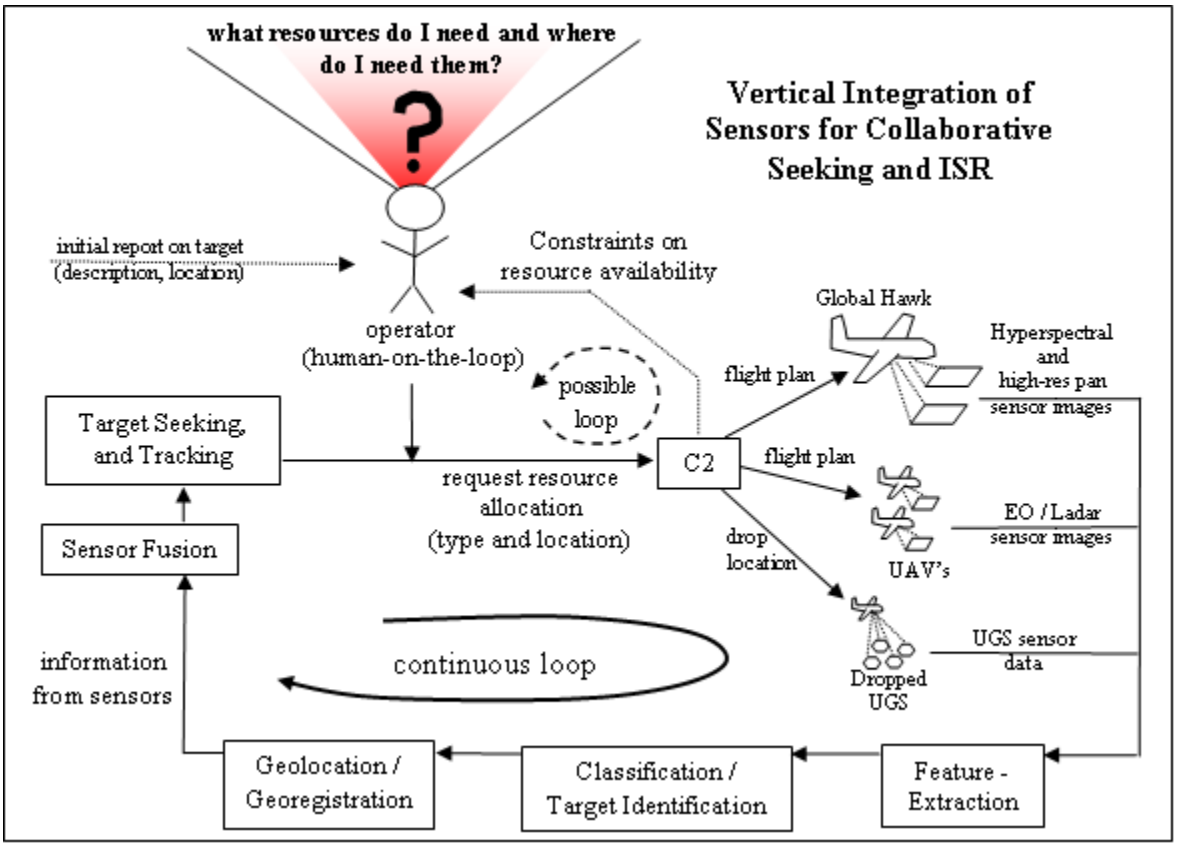

Figure 3: Key technological components with Human-on-the-loop

2 VISHON

Although the scenario above is fictional, it is representative of the problems experienced in current deployments where an asymmetric enemy is able to use the urban environment to their advantage. One solution to the problem narrated above, and many others, is to employ multiple UAVs (or in the future, small or micro-UAVs) as a cooperating pod. In addition to cooperation, anticipated next generation UAVs will be able to process their sensor (typically video and still imagery) data locally and can cooperate to some degree with other UAVs in the same pod. In the context of the UAV roadmap [OSD05], our anticipated UAVs would occupy levels 4, 5, and 6

on the scale given for autonomous capability.

Our VISHON system is situated in the broader context depicted in Figure 3. In the anticipated collaborative seeking and ISR process depicted in Figure 3, a pod of cooperating UAVs is collecting sensor data and performing fusion without the direct involvement of the human operator. Here, the human operator acts in a supervisory role, orchestrating the seeking and ISR process primarily through resource allocation, but not directly involved in the decisions executed by each UAV. Rather than human-in-the-loop, such a configuration is coming to be known as human-on-the-loop [CSN04; D04; HRH04].

The shift from flying UAVs towards supervisory control of UAV pods is expected to produce unique challenges. An earlier Air Force Scientific Advisory Board study [JO03] made the observation: "Mission management is the key technology limitation and enabler." Our VISHON system, which we introduce in this paper, is a decision tool for aiding the supervisory human-on-the-loop operator by providing real-time advice. The real-time nature of VISHON is an absolute necessity as the problem of target tracking and re-acquisition in urban environments is quite difficult. Once a target track is lost, the area of uncertainty grows quickly and every second holds the potential difference between successfully reacquiring the target and failure.

In building the real-time model, which is the core component of our VISHON system, we use a method similar to response surface methodology (RSM). RSM [BW51] is a process which uses experimental design and high-fidelity simulation to build lower-fidelity approximating models (i.e. the response surface) that attempt to isolate the critical input variables. More specifically, we use an updated form of RSM, called the Modified Response Surface Methodology (MSRM) [BS04] which is tailored towards automated knowledge discovery from the simulation results. In MSRM the MSM cycle of design, experiment, and analysis is preceded by a search stage and succeeded by a generalization stage. In the present context, the generalization stage is performed by an operation which we refer to as model compression.

3 HIGH-FIDELITY SIMULATION

Our approach begins with a high-fidelity simulation of the contemporary urban operating environment fusing sensor information extracted across platform modalities, as depicted in Figure 4. Our urban traffic model is currently composed of two major elements: road networks and traversable terrain, and a model of driver intent. Complementary to the urban model, we also model the employed sensors and particular platform allocation, providing a classification accuracy and location accuracy of identified objects with some uncertainty on the confidence of that information.

3.1 Urban Model

The goal is to identify model inputs that we can quickly extract from a real-world location based on a known road network, regional influences (such as cultural driving habits) and current events. These inputs should appropriately influence the urban model in such a manner as occurs in real life. With well defined inputs, our model is capable of generalizing across a range of valid locations. This allows us to use a general model of traffic that does not need to capture all the details explicitly.

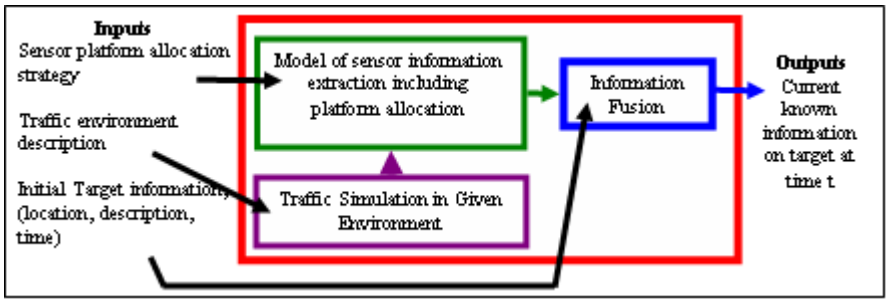

Figure 4: High-Fidelity Simulation

3.1.1 Road networks and traversability

A map encompasses the area of interest, centered at the last identified target location. We use *a priori* knowledge to capture the likelihood of traversable regions of the map. For instance, areas on the map such as buildings and lakes are not traversable, and therefore should have likelihood for traversability of 0. Roads and parking lots have likelihood for traversability of 1. Other regions, such as fields, can be assigned likelihood for traversability between 0 and 1. If

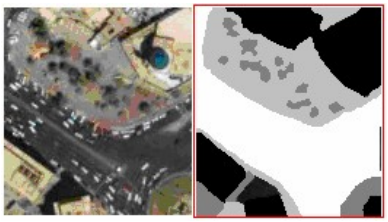
Figure 5: Traversability

a priori information is missing or incomplete, we assign a region's traversability as 1, therefore eliminating the possibility of Type I errors and accepting the added complexity of Type II errors.

Figure 5 shows the upper-left quadrant of the Khulani square (left image) with its associated traversability map (right image). In the traversability map, white regions, corresponding to roads, are fully traversable while black regions, corresponding to buildings and other structures, are not traversable. Dark and light gray denote regions which can be traversed but with difficulty.

The goal is to identify model inputs that we can quickly extract from a real-world location based on a known road network, regional influences (such as cultural driving habits) and current events. These inputs should appropriately influence the traffic model in such a manner as occurs in real life. With well defined inputs, our model is capable of generalizing across a range of valid locations. This allows us to use a general model of traffic that does not need to capture all the details explicitly.

3.1.2 Traffic flow and driver intent

Rather than anonymous, granular traffic flow as is common in cellular automata models (e.g. [SN98;BEF99]), our model must preserve both the identity and intent of each vehicle. Thus, the start location, goal location, and urgency of travel are all factors dependent on driver's intent. A driver's intent is modeled as an agent with beliefs, desires, and intentions (BDI) [ER01;RBB02]. Once we have identified the driver's intent, we use a path planning algorithm (such as A*) to determine optimal travel paths.

3.2 Target, Sensor, and Platform Allocation Model

Our experience with data collected in real-time from aerial platforms as compared to simulated, controlled environment, or laboratory collected test data, has provided for us a good understanding of the challenges that face real-time sensor data collection. These challenges include: sensor noise, environmental effects on sensor performance, geo-registration, feature extraction, and classification accuracy. All of these challenges are sensor and platform dependent in terms of their specific impact on the sensor information. However, generally across all sensor modalities, these challenges can be summed up as an impact on the classification accuracy and location accuracy of an object with some uncertainty on the confidence of that information.

3.2.1 Sensor models

Our model includes a collaboration of different sensor modalities for the tracking task. To effectively track a moving target, both sensors on aerial platforms and stationary ground sensors can be deployed. In general, sensors operate by collection of data, monitoring the visual, acoustic or other domains in a finite location within the environment.

Sensors deployed on aerial platforms generally rely solely on the visual domain, passively or actively capturing snapshots over different spectral frequencies at given time intervals over specific coverage areas of the terrain. High frame rate video can be captured providing high resolution imagery across the 3 wide bands in the visible spectrum. Hyperspectral or multispectral imagery collects information in the visual spectrum in tighter spectral bins providing a higher spectral resolution at the cost of a slower acquisition rate. LIDAR introduces the opportunity to collect information across a spectral channel in combination with a measurement on the distance of objects from the sensor and can therefore provide a 3-D image. SAR and radar are additional sensor modalities that are in operation today that can be utilized to collect information on the environment from aerial platforms.

In our high fidelity simulation, sensors types (classes) are generically modeled given an image capture range based on the sensors current location (defined by the allocation platform motion model, see below). Each sensor type (class) is associated with specific features that through information processing techniques can be extracted from the sensed data, including both spatial and spectral properties of entities, or in the present context, vehicles in the simulation. To model resolution differences between the capabilities of different sensor types to capture spatial or spectral information, the quality (accuracy) of the spatial and spectral information of a given sensor type (class) is captured through an associated standard deviation on the given feature type.

3.2.2 Modeling the target

We assume that objects of interest, including the target, can be described as a set of discrete features. Vehicles are an interesting target class, as there exists a range of different types of vehicles in an traffic environment, but it is not unlikely that similar or same type vehicles will co-exist, as car manufacturer do not make unique vehicles but sell mass quantities of the same vehicle type and color. Due to wear and tear or custom upgrades, individual vehicles may exhibit unique properties, however it is not likely that a fleeing suspect will choose to drive a supped up pink Cadillac as their intent is to blend into the traffic, not stand out. License plates provide another opportunity for a unique fingerprint of a vehicle in a traffic environment, however these may be difficult to image from above due to resolution and angle of the vehicle from the sensor as license plates may be located on the back and/or front of the vehicle. In some cases, vehicles may not even have license plates.

The range of features that describe a given entity is limited in our simulation to the set of sensor types deployed. Within this range, all entities are randomly assigned a specific description of values across the feature space. The target vehicle that we are tracking is also assigned a specific description across the same feature space, thus producing the likelihood of a non-unique target being tracked within the simulation.

3.2.3 Platform allocation

In order to classify collected sensor information, we must have learned to discriminate across a set of vehicle classes. The process of learning to identify which features provide class separability can be achieved through supervised training where we can estimate the density distribution of a vehicle class from some feature type as compared to another vehicle class. For some features, certain classes will be easily distinguished, but may be inseparable in other features. For instance, color may not be very distinguishing but profile geometry may be highly distinguishing. Provided new sensor information, we can extract a given feature and measure the similarity of this feature across the classes.

3.3 Track Fusion

The tracking of a target vehicle is accomplished by fusion of new sensor information from potentially multiple sensors with the area about which we expect the target to be. Track fusion operates by constraining the area we are certain the target could currently be, based on motion physics and the traversable areas of the map. With each passing moment, the area of possible target locations grows increasing our uncertainty on the exact target location. When sensor information can be used to uniquely identify the target, the track fusion is a process of continuously updating the targets current location. However, not all sensors may provide enough resolution across the feature space to uniquely identify a target, and in vehicle tracking, the target will probably not be a unique entity in the environment. In order to effectively track a moving vehicle target, a track fusion model must be employed that operates on constraint of the areas of uncertainty of possible location of the target, through acquisition of sensor information that can be used to reduce the area of uncertainty based on a high confidence that the target is not present at that location.

3.3.1 Area of Uncertainty (AOU)

The area of uncertainty (AOU) is a continuously growing region that with high confidence we believe the target must lie within. The growth of the area of uncertainty is limited by our expectation on the motion physics of a target and the traversable areas of the environment the target exists within. Left unchecked, this area would rapidly grow to a size which would make re-acquisition of the target intractable. In order to constrain the AOU, information must be collected in a timely manner that identifies regions of the AOU where wither we know with 100% confidence the target is, or regions where we have a high confidence the target is not. Because we do not assume the target in our simulation has a unique feature space, we rely on the latter for constraint of the AOU. With enough sensors deployed to cover the expanding AOU and collecting the right information in the right areas, the AOU can be constrained to a single point, thus identifying specifically the exact target location at a given point in time.

3.3.2 Evidential Reasoning Networks and Subjective Logic

21CSI's *Evidential Reasoning Network* or ERN is a semantically tagged belief fusion layer for evidential management and is an ideal mechanism for sensor fusion in the presence of uncertainty. ERN is also

particularly well suited to integrating machine generated facts with human opinions making ERN an excellent technology for human-on-the-loop decision support systems. Our colleagues have used ERN to great effect in previous work and we leave it to the interested reader to find a more complete ERN discussion in those works [OS07;LOZ07;ZOS07].

ERN is a connectionist structure that requires an underlying belief algebra. In the present context, we employ an ERN based on subjective logic to perform sensor fusion in our simulation. Subjective Logic is a way of thinking about uncertainty that builds upon the basic ideas presented by Dempster and Shafer [S76] to incorporate the subjectivity of all observations. Jøsang states the following, "Although including uncertainty in the belief model is a significant step forward, it only goes half the way in realizing the real nature of human beliefs. It is also necessary to take into account that beliefs always are held by individuals and that beliefs for this reason are fundamentally subjective." Jøsang argues that the consensus combination rule generates more intuitively correct results than Dempster's rule and the non-normalized version [J02].

$$k = u_x^A + u_x^B - u_x^A u_x^B$$
$$b_x^{A \oplus B} = (b_x^A u_x^B + b_x^B u_x^A)/k$$
$$d_x^{A \oplus B} = (d_x^A u_x^B + d_x^B u_x^A)/k$$
$$u_x^{A \oplus B} = (u_x^A u_x^B)/k$$
$$a_x^{A \oplus B} = \frac{a_x^B u_x^A + a_x^A u_x^B - (a_x^A + a_x^B) u_x^A u_x^B}{u_x^A + u_x^B - 2 u_x^A u_x^B}$$

Formula 1: Consensus

In Subjective Logic, we operate on opinions as opposed to facts. An opinion ω_x^A of object *x* by a subject (agent) *A* is a 4-tuple of the belief b_x^A, disbelief d_x^A, uncertainty u_x^A, and relative atomicity a_x^A (with respect to all possible states) about some statement, *x*. (Belief mass must sum to unity, thus $b_x^A + d_x^A + u_x^A = 1$.) The object of the opinion defines the semantics of the opinion and as ERN operates on opinions, those semantics are explicitly preserved.

$$b_x^{A \otimes B} = b_B^A b_x^B$$
$$d_x^{A \otimes B} = b_B^A d_x^B$$
$$u_x^{A \otimes B} = d_B^A + u_B^A + b_B^A u_x^B$$
$$a_x^{A \otimes B} = a_x^B$$

Formula 2: Discount

ERNs are formed by linking opinion producers (characterizing software agents or humans) through the operations of consensus and discount. Consensus (Formula 1) fuses two or more opinions of the same semantics while discounting (Formula 2) is used to limit the influence of an opinion on the overall consensus. Opinions of different types can be fused indirectly by an opinion producer which embodies sufficient domain knowledge to derive a new opinion.

4 MODEL COMPRESSION

Stochastic, discrete event simulations can model with very high fidelity. However this fidelity comes with a cost in substantial computing power. Paraphrasing an engineering maxim: A simulation can be high fidelity, require little computing, or require little storage volume – pick any two. In forward deployed conditions, deep computing resources will likely be scarce or non-existent. Consequently, we do not expect a high-fidelity stochastic discrete event simulation to provide the real-time decision support needed by VISHON.

We posit that it is possible to substantially reduce the processing requirements of a discrete event simulation with only a modest trade off in fidelity and storage volume. Using a methodology similar to MSRM, we build a real-time decision module by long running loops of experimentation conducted in the high fidelity simulation and then generalizing over the results of those simulations

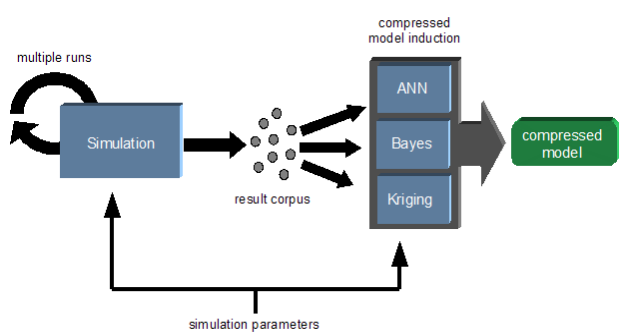

Figure 6: Model compression in VISHON

In Figure 6, we see a general overview of the model compression used to build VISHON. With massive runs of the simulation we generate a result corpus. This result corpus contains variations in all of the model parameters as well as the stochastic variations from each simulation run. After using potentially thousands of hours of off line computing time to generate the result corpus, we randomly partition the corpus into training and testing sets. The training set is used to induce the compressed model while the testing set is used to test the fidelity of the induced model. The compressed model will be integral to an anticipated real-time decision support system that can be used by an operator to quantify the

uncertainty in target sensing and re-acquisition across multiple environments.

5 RESULTS AND DISCUSSION

In our parlance, a macro model is a high fidelity simulation which requires substantial computing resources to execute the model. In the present context, the macro model is a stochastic, discrete simulation of urban traffic.

In contrast to the macro model, we define the micro model as a compression of the macro model. The micro model requires the same input parameters as the macro model and through execution produces the same type of output data as the macro model. However, a micro model can be executed with substantially lower computing power and storage requirements.

5.1 Macro Model

In this early version of VISHON, our macro model is a stochastic discrete event simulation based on three inputs: road congestion, deployed sensors, and simulation end time.

The road congestion input directly determines the number of entities within the simulation, based on a multiple of 10 vehicles per square mile. So an input of 2 would translate to 2 * 10 vehicles per square mile. Also the maximum speed of the entities are constrained by road congestion (y = exp(-x/10)), so as the roads get more congested the average speed of the vehicles will slow down. Because all of the mobile sensors in the simulation travel at speeds much greater than the maximum vehicle speed, this is not a big impact on the model. The start and end location of all vehicles along with intent (motivation which relates to speed) is randomly sampled. In addition, a small percentage of the vehicles are parked cars and thus they do not move. This provides a stochastic process on what types of vehicles the target will pass, how close they are to the target vehicle description, and how many are within AOU (even with a fixed road congestion input).

The simulation currently includes 4 different types of sensors with assigned allocation platforms: coverage area and platform speed, a data acquisition rate, and a set of features that the sensor can collect with a defined variance on that feature upon collection.

The simulation currently includes 3 different features to define an entity: feature 1 (color) is assigned from a 3-tuple of randomly sampled values (uniform distribution) from 0 to 1, feature 2 (size) is assigned from a uniform distribution with equal probability of a value from the discrete set of 4, 5, or 6, and feature 3 (shape) is assigned from a uniform distribution with equal probability of a value from the discrete set of 2, 4, or 8.

The following information was arbitrarily chosen: Sensor 1 only collects feature type 3 with a std of 0.1, a range of 0.3 square miles, a collection rate of 7 seconds, and a platform speed of 110 mph. Sensor 2 only collects feature type 2 with a std of 0.2, a range of 0.2 square miles, a collection rate of 8 seconds, and a platform speed of 140 mph. Sensor 3 only collects feature type 1 with a std of 0.02, a range of 0.25 square miles, a collection rate of 6 seconds, and a platform speed of 210 mph. Sensor 4 collects all 3 features with feature type 1 a std of 0.01, feature type 2 a std of 0.5, and feature type 3 a std of 0.5, a range of 0.6 square miles, a collection rate of 10 seconds, and a platform speed of 400 mph. In follow on research, these parameters will be more representative of sensors currently in the warfighter's inventory.

The allocation platform that is carrying the sensor moves towards a random point in the current AOU. Each time a sensor collects data, the exact location of the sensor and therefore the sensor coverage area is then about the AOU but differs from simulation to simulation.

The sensor data collected from a given entity is based on the actual feature values of the entity but includes noise inherent to the sensor (the sensor feature std), which is samples from a Gaussian distribution with the actual value as the mean.

5.2 Micro Model

Currently, our model compression engine is a perceptron neural network (NN), with 10 neurons in the hidden layer. The micro-model NN was implemented using the netlab toolkit [NET03]. Using the corpus of results generated by simulations of the macro-model, we trained the perceptron neural network using the conjugate gradient descent algorithm with $\alpha = 0.01$ and 1,000 epochs.

In the future, we hope to expand our model compression engine and explore other techniques. Techniques under consideration include standard statistical analysis, function approximation with artificial neural networks (ANNs), case-based reasoning, and multi-dimensional interpolation using the Kriging method [C90].

5.3 Comparison

The Macro simulation model must be run thousands of times to provide a good estimate of the minimum expected track time. This would not be feasible to do in real-time and therefore would be of no value to an operator who would be interested in the confidence in tracking a given target over a given length of time. However, we can run many simulations off-line, and from this data estimate the output confidence. It is not feasible to run across the entire input space however (although ideal) but rather we can run many points across the space and fit a model to this data to approximate the Macro model.

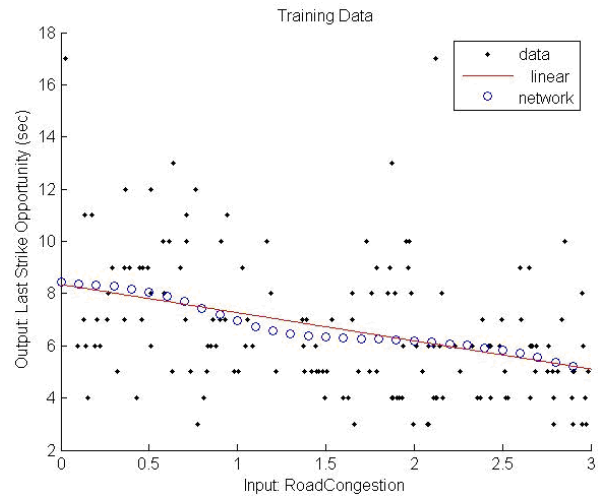

Figure 7: Fit of ANN to Road Congestion Parameter

As a simple example, we evaluated the output over a fixed set of deployed sensors (types [2, 2, 2, 0]), changing only the value for the Road Congestion between 0 and 3. We ran 5 runs, repeating the Road Congestion value and calculated the mean from this set. Figure 7 plots these mean values for the output vs. the input Road Congestion value (black points). We then generated a micro-model of this data using the techniques described in the previous subsection.

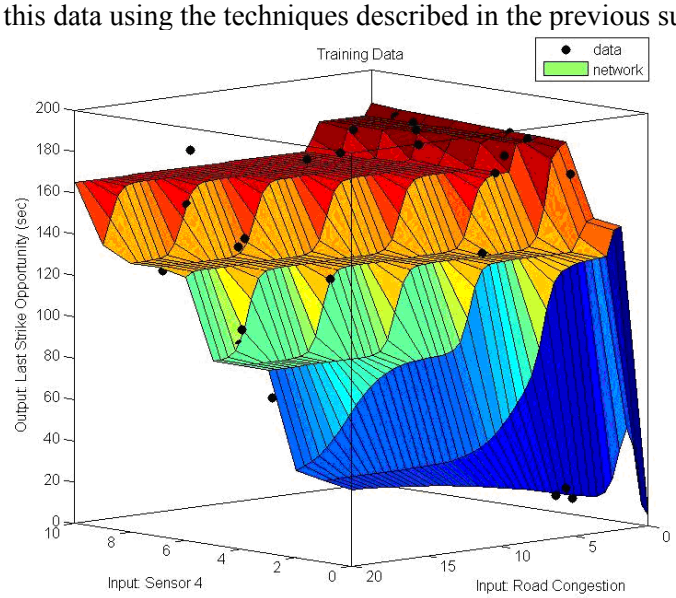

Figure 8: Fit of ANN to multiple parameters

The blue circles in Figure 7 show results from test inputs to our NN. A linear fit has been drawn across the data (red line) to provide reference and demonstrate the trend that as the Road Congestion increases, the length of successful track time decreases. As can be seen, the NN micro-model only slightly over-fit the sparse training data set, but provides the same trend as the linear fit. This simple example provides us opportunity to illustrate our method graphically. Because we only repeated each input parameter value 5 times, the residual error in our model, as observed by the distance of the black points in the figure below to the network result are between 2 and 4 seconds. Had we run more data for training the network, the residual error would have decreased.

As another, slightly more complicated example, we ran our macro simulation across varying two different input variables, road congestion and the number of deployed sensors of type 4. This simulation also deployed one of each of the first 3 sensor types. Across this input parameter space, the output varies from a minimum of 20 to a maximum of 180 seconds (the simulation end time). In this example, we again ran each input set only 5 times, and too the mean of the output results, plotted by the black point sin the figure below. We fit a NN to the training data space, whose surface has been plotted in Figure 8.

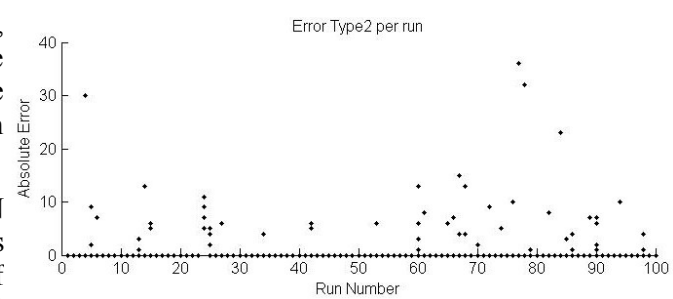

Figure 9: micro model absolute error (seconds)

To compare the Macro Simulation model to the NN micro-model, we randomly generated 100 input points and ran each 5 times through the macro simulation. Of the 500 total runs through the macro simulation 88.8%

produced output whose length of time met or exceeded the NN output. Figure 9 provides the error across all runs. It is clear from these results that the NN model over fit certain points in the input space, producing consistent error across all 5 repeated runs. In general, however, the micro model approximated well the macro model producing errors less that or equal to 10 seconds 98.2% of the time.

Our objective is to embed the micro model into a real-time decision support system. As such, our micro model must not only preserve much of the fidelity of the macro model, but must also possess the ability to execute very quickly. In our final test, we compared the computing speed of the macro and micro models.

We randomly generated 50 sets of input parameters which were fed to both the macro and micro models. In both models, we "tracked" the target for no more than 3 minutes of simulation time. The macro model was able to complete the simulation test set in approximately 92 minutes. With only one run, the stochastics of the simulation may dominate the output. Typically, multiple replications are run for each variation of input parameters in order to reduce variance. With 20 replications, the macro model would need approximately 30 hours to produce good results. In contrast to the macro model, our micro model NN was able to process all 50 input parameter sets in 0.32 seconds. Furthermore, the NN model is quite small and requires little primary and secondary storage. Our micro model is fully realizable on forward deployed laptops and even hand-held (PDA) computers.

6 CONCLUSION

Endangering civilians is often an intended secondary effect of asymmetric and 4GW adversaries. Harassing, wounding, or killing civilians, even if unintended, can create a backlash of public sentiment that erodes indigenous support for friendly forces. Thus, it is essential that friendly forces act with high confidence. The warfighter will use VISHON as a real-time aid for sensor employment that will provide information on the confidence expected in tracking or seeking a vehicle as it travels through an urban environment. Knowing the level of confidence increases the warfighter's awareness and ultimately will lead to fewer incidents of harming civilians and more successful engagements with the enemy.

Our vision for VISHON is to equip the warfighter with a real-time decision support tool that is effective in efficiently allocating scarce mobile sensor platforms. Our solution approach is to use a high fidelity discrete simulation (macro model) that can be used with deep offline computing resources to generate a large results corpus. This corpus is then used to build a micro-model that retains much of the fidelity of the original macro model, but is highly compressed and nearly instantly executable. The very short execution duration allows the decision loop to process faster, leading to fewer escapes by those that wish to perpetrate mayhem.

Our preliminary research results, presented here, show much promise that our solution approach is feasible in terms of both fidelity and real-time computing requirements. Further research is planned.

7 REFERENCES

[BEF99] R. Barlovic, J. Esser, K. Froese, W. Knospe, L. Neubert, M. Schreckenberg, and J. Whale. (1999) Online Traffic Simulation with Cellular Automata.

[BS04] D.E. Brown and J.B. Schamburg (2004) A Modified Response Surface Methodology for Knowledge Discovery with Simulations. Technical Report. http://www.sys.virginia.edu/techreps/2004/sie-040006.pdf

[BW51] G.E.P. Box and K.B. Wilson (1951) On the Experimental Attainment of Optimum Conditions (with discussion). *Journal of the Royal Statistical Society* Series B 13(1) pp. 1-45.

[C90] N. Cressie, (1990) The Origins of Kriging. *Mathematical Geology* vol. 22(3) pp. 239-252.

[CSN04] R.W. Chalmers, D.H. Scheidt, T.M. Neighoff, S.J. Witicki, and R.J. Bamberger (2004) Cooperating Unmanned Vehicles. In *Proceedings of the 1st AIAA Intelligent Systems Technical Conference*.

[D04] M.T. DeGarmo, (2004) "Issues Concerning Integration of Unmanned Aerial Vehicles in Civil Airspace," MITRE report to the FAA.

[ER01] P.A.M. Ehlert, L.J.M. Rothkrantz. (2001) Microscopic Traffic Simulation with Reactive Driving Agents. In *Proceedings of Intelligent Transporation Systems*.

[HRH04] M.L. Hanson, E. Roth, C.H. Hopkins, and V. Mancuso (2004) Designing Human System Interfaces for Supervising Multiple UAV Teams. In *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*.

[J01] A. Jøsang (2001) A Logic for Uncertain Probabilities. In *International Journal of Uncertainty, Fuzziness and*

Knowledge-Based Systems, vol. 9, pp. 279-212.
[J02] A. Jøsang (2002) Subjective Evidential Reasoning. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty (IPMU2002),* 2002.
[JO03] R. Johnson and M. O'Neill (2003) Unmanned Aerial Vehicles in Perspective: Effects, Capabilities, and Technologies. Air Force Scientific Advisory Board Study Outbriefing. 27 June 2003.
[LOZ07] E. Lindahl, S. O'Hara, and Q. Zhu. (2007) A Multi-Agent System of Evidential Reasoning for Intelligence Analyses. In *International Conference on Autonomous Agents and Multi-agent Systems* (*AAMAS) '07*.
[NET03] http://www.ncrg.aston.ac.uk/netlab/ accessed January 22, 2008.
[OS07] S. O'Hara, M. Simon, "Toward distributed ATR using subjective logic combination rules with networked UAVs," SPIE Defense and Security Symposium, Conference on Unmanned System Technology, April 2007.
[OSD05] Office of the Secretary of Defense (2005) Unmanned Aircraft Systems Roadmap 2005-2030.
[RBB02] R.J.F. Rossetti, R.H. Bordini, A.L.C. Bazzan, S. Bampi, R. Liu, D.V. Vliet. (2002) Using BDI Agents to Improve Driver Modeling in a Commuter Scenario. *Transporation Research Part C: Emerging Technologies* vol. 10(5).
[S76] G. Shafer (1976) A Mathematical Theory of Evidence: Princeton University Press Princeton, NJ.
[SF97] T. Schulze and T. Fliess. (1997) Urban Traffic Simulation With Psycho-Physical Vehicle Following Models. In *Proceedings of the 1997 Winter Simulation Conference*.
[SN98] P. Simon and K. Nagel. (1998) Simplified Cellular Automaton Model for City Traffic. *Physical Review E*. vol 58(2) pp. 1286-1295.
[ZOS07] Q. Zhu, S. O'Hara, M. Simon, and E. Lindahl (2007) Collective Agents Interpolative Integral (CAII) for Asymmetric Threat Detection. In *Proceedings of 2007 SPIE Defense and Security Symposium.*