IDA

INSTITUTE FOR DEFENSE ANALYSES

Review of Methods and Algorithms for Dynamic Management of CBRNE Collection Assets

A. G. Wilson, Project Leader

July 2013

Approved for public release;
distribution unlimited.

IDA Paper P-4995

Log: H 13-000325

Copy

INSTITUTE FOR DEFENSE ANALYSES
4850 Mark Center Drive
Alexandria, Virginia 22311-1882

The Institute for Defense Analyses is a non-profit corporation that operates three federally funded research and development centers to provide objective analyses of national security issues, particularly those requiring scientific and technical expertise, and conduct related research on other national challenges.

About this Publication

This work was conducted by the Institute for Defense Analyses (IDA) under contract DASW01-04-C-0003, Task DC.1-2607.02, "Dynamic Analysis and Management of CBRNE Collection Assets" for the Defense Threat Reduction Agency (DTRA). The views, opinions, and findings should not be construed as representing the official position of either the Department of Defense or the sponsoring organization.

Acknowledgments

The authors would like to thank the IDA committee, Dr. Steve Warner (Chair), Dr. Arthur Fries, Dr. Ivo K. Dimitrov, Dr. Nathan Platt, Dr. Jeffry T. Urban, and Dr. Gary F. Willmes for providing technical review of this effort.

INSTITUTE FOR DEFENSE ANALYSES

IDA Paper P-4995

Review of Methods and Algorithms for Dynamic Management of CBRNE Collection Assets

A. G. Wilson, Project Leader
S. M. Cazares
K. M. Papadantonakis
M. R. Avery
J. F. Cartier
P. A. Dolph
J. M. Fregeau
J. R. Holzer
K. A. Morrison
S. M. Nunes
S. R. Renn
J. A. Snyder
K. M. Spencer

Executive Summary

The Defense Threat Reduction Agency (DTRA) provides reach-back support to warfighters combating threats from chemical, biological, radiological, nuclear, and enhanced conventional (CBRNE) weapons. The assets accessible through DTRA's reach-back capability include both technology and subject matter expertise. As tactical data infrastructures are used by a greater variety of CBRNE detection systems, the opportunity exists to improve stand-off detection performance through remote processing and fusion of sensor data and modeling of the operational environment. DTRA is actively developing technologies to enable and support a future mode of operations where tactical communication infrastructures transmit data in real time from heterogeneous networks of CBRN (and non-CBRN) sensors positioned in and around an area of operations for remote processing. The same networks could also be used to deliver responses such as real-time sensor tasking and analytical products from the reach-back capability to warfighters.

The integrated support approach being pursued by DTRA must be applicable to a wide range of operational missions including direct force protection, targeted search, long-term threat behavior monitoring, and wide area search. The missions vary in spatial and temporal dimensions; they also vary with regard to the extent of ownership/control of the operational area and the deployed sensors, expertise of users, knowledge of threat signatures, and other factors.

Advances in the development of a number of technologies may enable the realization of such an interconnected mode of operations. The enabling technologies being addressed by DTRA include CBRNE sensor development and improvement, communications and bandwidth improvement, and the development of interfaces and protocols that allow the necessary connections between system components. In addition, DTRA leverages physics-based models for phenomena such as background radiation, neutron transport, and atmospheric transport and dispersion that are relevant to the CBRNE mission. The sensor data received through the tactical infrastructure will need to be integrated with the appropriate physics models in order to provide reach-back support. This integration of sensors and models will require the development of automated methods (algorithms) for combining sensor information with physics models to perform critical functions such as anomaly detection, classification, and sensor management. Development of algorithms to perform supporting functions such as data triage, data fusion, and dimension reduction will also be required. Algorithm development is required to ensure that the analytic tools available to the DTRA reach-back capability keep pace with developments in network, sensor, and computation capabilities.

This report provides a broad literature review of the methods available to support real-time data fusion of sensor measurements with physics-based simulation models. At DTRA's request, this effort is not focused on any particular sensing technologies, physics models, or technology development time horizons. This document provides instead an overview, organization, and high-level assessment of recent methods for sensor fusion, anomaly detection, data assimilation, classification, sensor management, and supporting functions. The purpose of this review is to provide background material useful to an algorithm development program and to support the identification of promising directions for DTRA's ongoing field demonstration program.

The review starts by discussing several taxonomies that provide frameworks for organizing classes of algorithms. Next follows a discussion of data fusion for raw sensor data. Following this, the review considers data triage and compressive sensing, which are methods that selectively reduce the amount of data that needs to be processed. Discussion follows on data assimilation, anomaly detection, dimension reduction, classification, sensor management, and decision fusion.

Following the algorithm overview and discussion, we provide several case studies that focus on diverse applications. At DTRA's request, these are non-CBRNE case studies that illustrate one or more of the algorithms discussed in the first part of the report and point to challenges and lessons learned in the implementation of sensor fusion in real problems. The case studies include examples such as implantable cardioverter defibrillators, autonomous ground vehicles, and hurricane forecasting. The following table details the algorithms discussed, the illustrative examples for each, and potential CBRNE applications.

The report also includes a description of two other DTRA-sponsored programs that involve algorithm development for CBRNE applications. The Algorithms for Threat Detection (ATD) program is jointly sponsored by DTRA, the National Science Foundation, and National Geospatial Intelligence Agency and seeks to build a research community around the development of algorithms for CBRNE threat detection. We provide a detailed review of the research sponsored by this program, as some of the work has direct relevance to the reach-back mission. In addition, the ATD program provides a structure that could be used to drive future reach-back relevant research.

Sensor data-fusion algorithm development has also been supported by DTRA and the Joint Science and Technology Office for Chem-Bio Defense as part of the Joint Effects Model (JEM) development effort. The JEM-related efforts have been significant and have produced data and tools that may be leveraged by a program for real-time sensor and model data fusion algorithm development. Taken together, the ATD and JEM programs provide some guidance for proceeding with such a development program.

The report concludes by suggesting research topics for algorithm development and by suggesting methods for providing analytical support to a research and development program for real-time sensor and model data fusion algorithms. Research areas of

potential interest for ongoing field demonstrations are anomaly detection, data assimilation, and sensor management. Experimental data sets will be necessary for algorithm development and evaluation of results. Carefully designed field experiments can be used in conjunction with models designed to produce realistic synthetic data to generate the needed data sets. Model validation efforts will be critical to the success of an analytical toolset that integrates data from sensor networks with physics-based models.

Algorithms, Example Applications, Possible CBRNE Uses

Algorithms	Example Application Discussed	Possible CBRNE Application
Data Fusion	Tracking, UAV sense and avoid	Combine data from multiple sensors
Data Triage	Experimental particle physics data collection, routine Internet operations	Improve data collection given limited computing power on future low-cost sensor
Compressive Sensing	Data collection along a perimeter	Reduce data acquisition time or number of samples
Data Assimilation	Numerical weather prediction, CO_2 flux models	Combine observed data with modeling and simulation to improve estimation
Dimension Reduction	Hyperspectral imaging, gene sequencing	Extract features to improve CBRNE threat classification
Anomaly Detection	Detection of cheating in high school test taking, biosurveillance for disease outbreaks, detection of abnormal and dangerous fast heart rhythm	Earlier initial threat detection/alert, detection in the presence of complex background
Classification	Required operating characteristic curve for radar, decision about whether heart rhythm is lethal or non-lethal	Reduce CBRNE sensor false alarms, mission/situation-dependent tradeoffs between probability of detection and false alarm rate
Sensor Management	Improving near-real-time storm forecasting	Determine where best to point sensor for next observation
Decision Fusion	Control logic, CEC air defense tactical network	Combine information from sensors with minimal computational power

(This page is intentionally blank.)

Contents

1. Background

The Defense Threat Reduction Agency (DTRA) provides reach-back support to warfighters combating threats from chemical, biological, radiological, nuclear, and enhanced conventional (CBRNE) weapons. The assets accessible through DTRA's reach-back capability include both technology and subject matter expertise. As tactical data infrastructures are used by a greater variety of CBRNE detection systems, the opportunity exists to improve stand-off detection performance through remote processing and fusion of sensor data and modeling of the operational environment.

Advances in the development of a number of technologies may enable the realization of such an interconnected mode of operations. The enabling technologies being addressed by DTRA include CBRNE sensor development and improvement, communications and bandwidth improvement, and the development of interfaces and protocols that allow the necessary connections between system components. In addition, DTRA leverages physics-based models for phenomena such as background radiation, neutron transport, and atmospheric transport and dispersion that are relevant to the CBRNE mission. The sensor data received through the tactical infrastructure will need to be integrated with the appropriate physics models in order to provide reach-back support. This integration of sensors and models will require the development of automated methods (algorithms) for combining sensor information with physics models.

This report has three primary objectives. The first is to provide a literature review of methods available to support real-time data fusion of sensor measurements with physics-based simulation models. At DTRA's request, this review is not focused on any particular sensing technologies, physics models, or technology development time horizons. We first consider classes of algorithms typically used with sensor data, and then expand the discussion to algorithms used with both sensor measurements and simulation models. We give special focus to the role of physics-based models, the feasibility of developing real-time algorithms, and the utility of methods for developing adaptive strategies for sensor management.

The second purpose is to describe several case studies that demonstrate the integrated use of data and models outside CBRNE. These case studies both illustrate the methods discussed in the literature review and point out successes and challenges in integrating sensor measurements and simulations in a broad set of applications.

The third purpose is to summarize two DTRA-sponsored research programs that have supported algorithm development. The Algorithms for Threat Detection (ATD) program is jointly sponsored by the National Science Foundation (NSF), DTRA, and the National Geospatial Intelligence Agency (NGA). ATD was started in 2009 "to develop the next generation of mathematical and statistical algorithms for the detection of

chemical agents, biological threats, and threats inferred from geospatial information." The research performed by ATD is itself of interest in developing CBRNE reach-back capabilities, and the program offers opportunities to leverage its existing structure to drive additional reach-back relevant research. The Joint Effects Model (JEM) is supported by DTRA and the Joint Science and Technology Office for Chem-Bio Defense (JSTO-CBD). It is designed to provide a single Department of Defense (DoD)-approved tool to predict hazard areas and effects resulting from the use of CBRNE weapons and releases of toxic industrial materials. Requirements for JEM have driven science and technology development efforts to incorporate sensor data-fusion algorithms into CBRNE hazard prediction models.

For this report, we consider four DTRA-defined operational missions: direct force protection, targeted search, long-term threat-behavior monitoring, and wide-area search. These missions differ in two aspects: scale (both spatial and temporal) and operational purpose. Direct force protection and targeted search focus on local, immediate threats. Long-term threat-behavior monitoring and wide-area search occur over a relatively wider geographical area and over a longer period of time. Direct force protection and long-term threat-behavior monitoring are watchful activities that may produce a cue requiring a response; targeted and wide area searches are response activities.

In direct force protection and targeted search, expert users operate sensors in relatively controlled settings in order to test a known hypothesis. The sensors are deployed over geographical areas that are relatively small. In direct force protection, these small settings might be an embassy compound, a port, or a forward operating base. These settings can be fully controlled, since they are owned by friendly forces; users can deploy the sensors non-covertly and access the sensors directly for calibrating and downloading data. In targeted search, the setting might be a ship or a truck. These settings may only be partially controlled since they may not be owned by friendly forces; users may have slightly less freedom in the operation of the sensors, compared to direct force protection. In these missions though, the hypothesis to be tested is well known, since it relates directly to the physical phenomena upon which the sensors are based (e.g., the hypothesis, "Gamma radiation is not present," requires the use of gamma radiation sensors). Thus direct force protection is the most straightforward of DTRA's operational missions, followed by targeted search.

In contrast, long-term threat-behavior monitoring and wide-area search are characterized by users operating sensors in large, uncontrolled settings. The sensors may be used to test a known hypothesis or, more often, to support generation of new hypotheses. The individuals who operate the sensors may not be fully trained in their use for DTRA-specific missions. The sensors may collect data over very large areas (e.g., a city) and very long periods of time (e.g., weeks), leading to settings that cannot be controlled due to their very large size. Technologies for long-term threat-behavior monitoring and wide-area search are less mature than those for direct force protection and targeted search.

2. Taxonomies

Three formal taxonomies/models are useful in discussing the algorithms covered in this report. These are the Joint Directors of Laboratories (JDL) Data Fusion Model (Steinberg, Bowman, and White 1999; Steinberg and Bowman 2009), the Dasarathy Model (Dasarathy 1997), and the Omnibus Model (Bedworth and O'Brien 2000). None of these taxonomies provide exactly what is needed to organize the algorithms we consider, but each illustrates important facets of the problem.

The JDL model is primarily of historical interest for this report. An overview is given in Figure 2-1. It was proposed first in 1985 and is the de facto standard for U.S. defense data-fusion applications. The JDL model takes a process view of data fusion and does not have a completely natural mapping to algorithm functionality, but we include definitions of the five key JDL sub-processes for reference:

- Level 0 (sub-object data assessment) – Combine pixel or signal-level data to obtain initial information about an observed target's characteristics. Includes sensing and signal processing.
- Level 1 (object assessment) – Combine sensor data to obtain more reliable and accurate estimates of an entity's position, velocity, attributes, and identity (perhaps to support prediction). Includes feature extraction and pattern processing.
- Level 2 (situation assessment) – Dynamically develop a description of current relationships among entities and events in the context of the environment. Includes object clustering, relational analysis, and situation assessment.
- Level 3 (impact assessment) – Project the current situation into the future to make inferences. Includes decision making, consequence prediction, and vulnerability assessment.
- Level 4 (process refinement) – Using a meta-process, monitor overall data fusion and improve real-time system performance.

An additional level is included in some versions of the model:

- Level 5 (cognitive refinement) – Improve the interaction between the fusion system and one or more users/analysts. Includes visualization, cognitive assistance, bias remediation, and team-based decision-making.

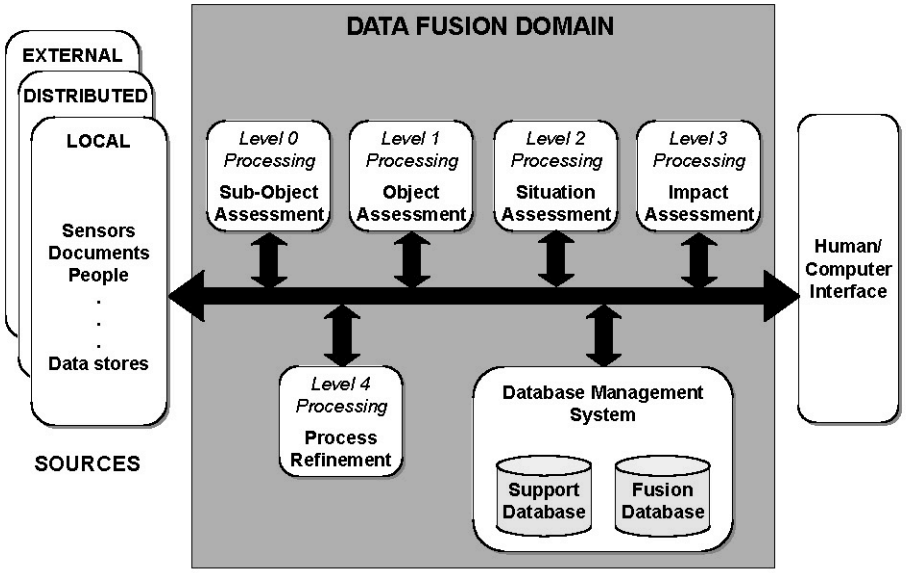


Figure from Steinberg and Bowman (2009).

Figure 2-1. JDL Data Fusion Model

The second taxonomy that we discuss is the Dasarathy Model (Dasarathy 1997), which focuses on the level of abstraction (specifically of inputs and outputs) during fusion processes. In particular, processing can occur at the level of raw sensor data, features, or decisions. Dasarathy (1997) discusses the five combinations that are shaded in Table 2-1; other references [for example, Steinberg and Bowman (2009)] discuss algorithm classes in additional cells. The Dasarathy model provides a flexible framework for categorizing algorithms.

Table 2-1. Dasarathy Model with Examples

Input	Output		
	Data (DAO = DAta Output)	**Features (FEO)**	**Decisions (DEO)**
Data (DAI = DAta Input)	Fusion of multi-spectral data (DAI-DAO)	Feature selection and feature extraction (DAI-FEO)	Anomaly detection, Sensor management (DAI-DEO)
Features (FEI)		Fusion of image and non-image data (FEI-FEO)	Classification (FEI-DEO)
Decisions (DEI)			Decision-level fusion (DEI-DEO)

The Omnibus Model (Figure 2-2) is a hybrid of several other multi-sensor data fusion models, with an emphasis on the feedback cycles within fusion. Within this model, we see the Boyd (or OODA) loop (Boyd 1995) and the Waterfall Model (Bedworth and O'Brien 2000). The OODA loop ("observe, orient, decide, act") was

originally proposed as a model for the military command process but is now widely used to describe data fusion. The Waterfall Model (Figure 2-3) is a more fine-grained description of the lower levels of the JDL model, with sensing and signal processing corresponding to JDL level 0, feature extraction and pattern processing to JDL level 1, situation assessment to JDL level 2, and decision making to JDL level 3. The Waterfall Model is widely used in the UK defense data fusion community but has not been widely adopted elsewhere (Bedworth and O'Brien 2000).

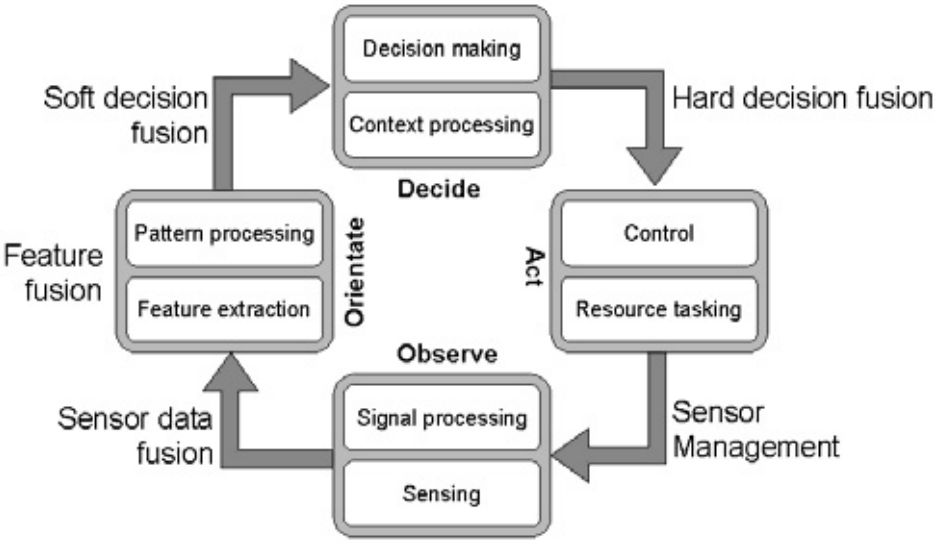


Figure from Bedworth and O'Brien (2000).

Figure 2-2. Omnibus Data Fusion Model

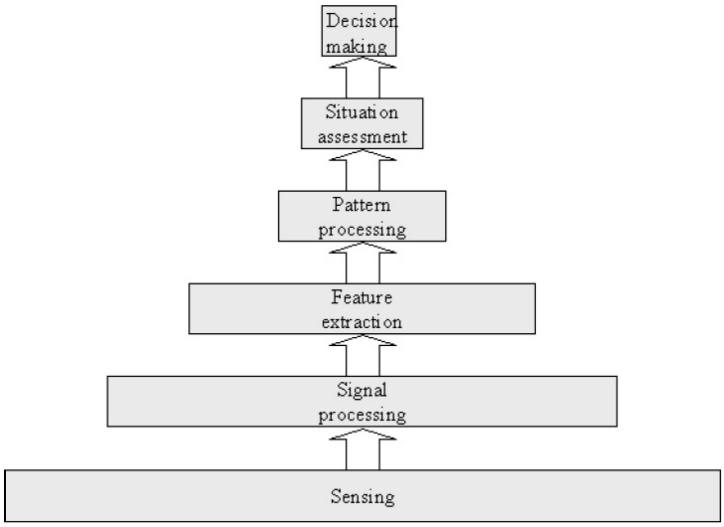


Figure from Bedworth and O'Brien (2000).

Figure 2-3. Waterfall Model

There is a reason for the multiple fusion taxonomies in use today – none of them maps naturally onto *all* possible fusion algorithms. We can organize our discussion of

algorithms (Chapter 3) using both the Omnibus Model and the Dasarathy Model. In addition, we have also adopted the categories of data fusion, feature fusion, and decision fusion (Elmenreich 2002).

Consider first organizing the algorithms using the Omnibus Model. Within the Observe box, we discuss *data triage* and *compressive sensing*, which are algorithms that reduce the amount of raw data that needs to be processed. Data triage is a principled post-acquisition decision about what data to process. Compressive sensing makes a prior decision about what data to acquire. From Donoho (2006):

> As our modern technology-driven civilization acquires and exploits ever-increasing amounts of data, "everyone" now knows that most of the data we acquire "can be thrown away" with almost no perceptual loss – witness the broad success of lossy compression formats for sounds, images, and specialized technical data. The phenomenon of ubiquitous compressibility raises very natural questions: why go to so much effort to acquire all the data when most of what we get will be thrown away? Can we not just directly measure the part that will not end up being thrown away? (p. 1289)

We also consider *data assimilation* algorithms, which are built to use all available sensor data *in combination with a physical model* to reconstruct, as accurately as possible, the properties of the environment of interest. In data assimilation, a forecast step is usually implied, in which observations and the physical model are used to predict the future evolution of the environment of interest.

Within the Orient box, we consider dimension-reduction algorithms, focusing on *feature selection* and *feature extraction*. We also discuss *anomaly detection*, or the detection of patterns in a given data set that do not conform to established "normal" behavior (Chandola, Banerjee, and Kumar 2009). Within the Decide box, we discuss algorithms for *classification* and *tracking*, and within the Act box, we focus on *resource tasking* for sensor planning and real-time sensor allocation.

Moving to the Dasarathy Model and Elmenreich (2002) categories, we think more specifically about the three fundamental types of fusion: data fusion, feature fusion, and decision fusion (Elmenreich 2002). *Data fusion* combines measurements from multiple detectors. Typically, the objective of the fusion is to combine multiple independent measurements of the same object or area to improve detection. When similar detectors are spatially separate, the measurements can be used to localize the source (more accurate position) or estimate the magnitude of the source.

The fusion of measurements from detectors of different modalities can increase the amount of information available and improve the ability of a system of detectors to identify the source. Of course, one does not simply add raw measurements together if they are measuring different things. However, if the target of interest can be distinguished by a characteristic set of measurements, then the occurrence of that set of measurements from multiple detectors may be used a discriminant. This is called *feature*

fusion. A simple architecture for this would employ multiple detectors of different modalities at the same location. Each set of detectors provides a multidimensional measurement of the source, and the right set of detectors may provide a much more reliable identification of the source than would be achievable from a single modality network.

Another way to fuse the information from multiple detectors of different modalities is to allow each instrument to independently form a decision about the source and then combine the decisions of all instruments to reach a final assessment. This is called *decision fusion*. Decision fusion depends critically on understanding the uncertainties contributing to the decision, and these are often difficult to quantify. Unlike a detector measurement, where the physical process and the sensor characteristics define the measurement accuracy, decisions may arise from estimates based on human experience, and each decision may be very different in its uncertainties. On the other hand, the advantage of decision fusion is it greatly reduces the amount of data that has to be exchanged on the network and may be the only way to take advantage of information sources such as human observations.

The type of fusion employed in a tactical network depends on the purpose of the network and the challenges imposed by constructing the network. For example, if raw measurements from the detectors are too voluminous to disseminate to the fusion processor, then reducing the data to a feature or to a decision may be required. In general, the challenges to be considered include accommodating the data exchange and dissemination requirements of the architecture, and the latencies in classification, decision, and response components of the architecture.

In Chapter 3, we describe methods for implementing these three types of fusion. We provide an overview section on data fusion, and then we also discuss the Data-Data (DAI-DAO) algorithms for data triage, compressive sensing, and data assimilation. We consider feature selection and extraction, which cross Data-Feature (DAI-FEO) and Feature-Feature (FEI-FEO) methods.

As a note, DAI-FEO methods are often problem specific. From Dasarathy (1997):

> Fusion in this mode, depending on one's viewpoint, input-fusion *of data* or output-fusion resulting *in features*, has been looked upon either as data fusion or feature fusion. The manner in which depth perception is achieved in humans, by combining the visual information acquired from the two eyes, can be looked upon as a classical paradigm of this feature or information fusion. The traditional approach to the computation of object surface temperatures using the intensities from two infrared (IR) bands of a multispectral scanner is another good example of data in-feature out mode of fusion processing. (p. 29)

Examples of Data-Feature methods can be found in Sections 4.B (Implantable Cardioverter Defibrillators), 4.C (Autonomous Ground Vehicles), and 4.E (Syndromic Surveillance).

We also discuss anomaly detection and sensor management [both Data-Decision (DAI-DEO) methods], classification [a Feature-Decision (FEI-DEO) method], and provide an overview of decision fusion methods [Decision-Decision (DEI-DEO) methods].

3. Algorithms

A. Data Fusion

Data fusion is the direct combination of raw sensor data. This is a broad area of research in which the methods and algorithms typically address one or more of the following concerns (Khaleghi et al. 2013):

- *Data imperfection* – Sensor data will typically have bias, imprecision, and uncertainty.
- *Outliers and spurious data* – Algorithms should be robust to ambiguous and inconsistent measurements.
- *Conflicting data* – Algorithms should produce sensible answers even when the data conflict.
- *Data modality* – Sensor networks may collect qualitatively similar (homogeneous) or different (heterogeneous) data.
- *Data correlation* – In distributed networks, some sensor nodes are likely to be exposed to the same external noise sources.
- *Data alignment/registration* – Sensor data must be transformed from each sensor's local frame into a common frame before fusion. Measurements from each sensor (potentially measured at different times from different viewpoints with different calibration errors) must be transformed into a common coordinate system.
- *Data association* – Data association can either refer to associating particular measurements to a track (identifying from which target, if any, each measurement originated) or track-to-track association (distinguishing and combining tracks).
- *Processing framework* – Is the fusion processing performed in a centralized or distributed manner?
- *Operational timing* – The sensors may collect data at different rates or frequencies, and fusion methods should incorporate multiple time scales and be able to address the issue of out-of-sequence arrival of data.
- *Static vs. dynamic phenomena* –The phenomenon under observation may be time-varying (requiring attention to data freshness) or time-invariant.

- *Data dimensionality* – High-dimensional data may be processed into lower-dimensional data or features. This may save communication bandwidth and computational load at a central fusion load.

Specific algorithms for data fusion are typically sensor-dependent. However, there has recently been some novel work reframing many of the data fusion issues, including data imperfection, outliers, conflicting data, data modality, and data correlation, into a common statistical framework (Rodriguez 2012). This research was funded through the NSF/DTRA/NGA Algorithms for Threat Detection Program[1] (Chapter 5).

Rodriguez (2012) proposes the following framework. Suppose that we observe a vector of sensor data $y_t = (y_{t,1}, y_{t,2}, \ldots, y_{t,q})$. We are interested in estimating the unobserved (fused) signal x_t. The current state of the practice is to apply some linear filter or logical gates to the observed sensor data to estimate the unobserved signal: for example, $\hat{x}_t = \bar{y}_t = \frac{1}{q}\sum_{j=1}^{q} y_{t,j}$ or $\tilde{x}_t = \min\{y_{t,1}, y_{t,2}, \cdots, y_{t,q}\}$). These choices of fusion functions are arbitrary. A more statistical perspective can provide a unifying framework for different approaches that makes assumptions explicit, suggests generalizations, supports metrics for assessing optimality, and provides uncertainty measures.

Consider the "common source" paradigm where x_t is the unobserved signal, and we observe $Y_{t,j} = f_j(x_t) + \varepsilon_{t,j}$ (Figure 3-1). Note that this allows a very general formulation, where what we observe are noisy transformations of a common latent signal.

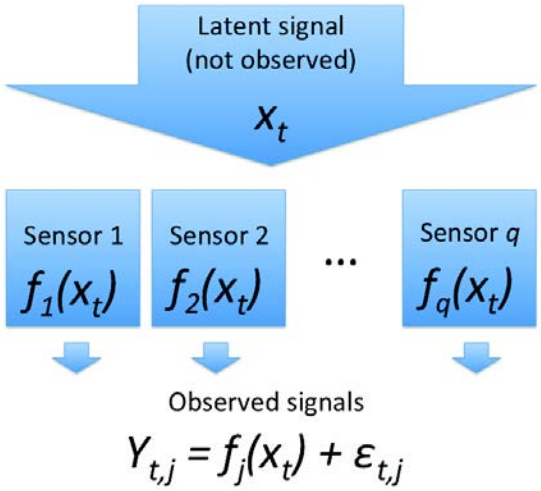


Figure from Rodriguez (2012).

Figure 3-1. "Common Source" Paradigm for Data Fusion

A simple example might have $y_{t,j} = \alpha + \beta x_t + \varepsilon_{t,j}$, with the elements of ε_t normally distributed with variances ψ_j^2. The current standard approach is to fit the parameters (α, β, ψ) based on training data and then to hardwire these estimates into the

[1] The program announcement can be found at http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503427&org=DMS.

filter. This assumes linearity and an additive Gaussian error structure, ignores the uncertainty in the parameter estimates (which is important for small training sets and complex models), and assumes that lab conditions are the same as field conditions. However, by noticing that this is a *one-factor linear factor model*, one can estimate (α, β, ψ) and x_1, …, x_q without training data. If training data are available, they can be incorporated and used to perform model assessment.

A slight generalization, assuming structure on the underlying signal so that $x_t = x_{t-1} + \omega_t$, with ω_t normally distributed, results in a Kalman-filter-based approach with strong links to *Bayesian dynamic linear models* (Prado and West 2010). Further generalizations consider non-linear functions $f_j(x_t)$ with additive noise, which are typically estimated using basis representations (Vidakovic 1999) or Gaussian processes (Rasmussen and Williams 2006). Rodriguez (2012) points to upcoming research, considering $y_{t,j} \sim p_j(\cdot\,|x_t)$, that will allow the x_t to evolve temporally or spatially. Reframing the data fusion problem as a statistical estimation problem allows the application of considerable estimation and computational machinery, as well as explicitly acknowledging the effect of uncertainty and assumptions on the fused output.

B. Data Triage

At its most basic level, *data triage* is a method of reducing the data rate coming from a sensor (or set of sensors) that would otherwise swamp the system that is expected to process and disseminate those data. This could be due to limited bandwidth, limited processing capability, limited storage, or any other limitation on the capability of the processing and dissemination system. The intent of data triage is to allow the system to continue to function at a reduced data rate in situations where the system would otherwise fail completely. In some cases, this is a graceful degradation in performance; in other cases, it is possible to maintain full performance well beyond the point where data triage is necessary.

1. When Is Data Triage Needed?

In general, *data triage* is necessary when the data rate exceeds the speed at which it can be processed. For continuous systems, the rate does not vary significantly, so the mean rate is all that matters. For bursty systems, this consideration is nominally applied at peak data rate. However, appropriately located and sized buffers can be designed into the system to smooth the effective data rate. If these buffers are effective, then the system can be treated as continuous. Part-time systems can be treated as bursty if the latency generated by spreading out the data in time is not problematic. Otherwise, one must pay attention to the peak data rate.

Let us consider an example from the realm of experimental particle physics. There are numerous colliding-beam experiments where a large accelerator generates two counter-rotating beams of particles and steers them into collision at specific locations. Detectors are placed at these locations to observe interesting particle reactions that occur

in these collisions. For many years the Tevatron[2] accelerator at the Fermi National Accelerator Laboratory[3] (FermiLab) was the highest energy particle accelerator in the world. The rate of beam crossing was 7.6 million times per second (7.6 MHz, or 132 ns between bunches).[4] As is typical in colliding-beam experiments, there are particle interactions in each beam crossing, generating data that could be read out. However, the interesting physics[5] is rare and occurs only every millionth (or billionth or trillionth) beam crossing.

The Collider Detector at FermiLab (CDF)[6] was one of the detectors at the Tevatron. The other, D0,[7] is substantially the same, as are most large detectors at other accelerators. In this section, we will concentrate on CDF. The CDF average event size (potentially interesting physics) is 150 kB, and the data had to be assembled from 16 different readout locations (Anikeev et al. 2000). This would have led to a data rate of ~1.1 TB/s when, in reality, their tape system could only handle about 30 events per second (4.5 MB/s).[8] To do this data triage, CDF chose to set up a multi-level trigger system, which is a common solution in experimental particle physics.

The Level 1 trigger is implemented entirely in hardware. It receives data on each bunch crossing (up to 7.6 MHz) and downselects to events of interest whose rate is capped at 50 kHz. It can do this in about 4 μs and is sized to repeat this on each bunch crossing (132 ns). All the readout electronics are pipelined for 42 bunch crossings (5.5 μs) to allow the Level 1 trigger to complete. If an event fails the Level 1 trigger, then data from the readout electronics is discarded and can never be recovered. For events that pass the Level 1 trigger, the Level 2 trigger collects data from the readout electronics into one of four asynchronous local buffers. Using these data it further downselects the events of interest to 300 Hz, taking about 20 μs to make its decision. The Level 3 trigger does a full readout of the detector and sends the data to one node in a processor farm to further reduce the accept rate to 10 Hz.

For this discussion, there is no need to get into the detailed physics of what the triggers were looking for or how they were balanced against each other in an attempt to satisfy all the scientists who were working on CDF. At Level 1, measurements of quantities, such as total transverse energy and missing transverse energy, point toward physics of interest. CDF was ultimately able to implement a tracking algorithm that

---

2 http://www.fnal.gov/pub/science/accelerator/.

3 http://wwwfnal.gov/.

4 This was the design specification for Run II. The other primary mode used in run II was 396 ns bunch crossing time, or 2.5 MHz (Fermilab 1996).

5 There is some debate over what is truly interesting. Here "interesting" means new, undiscovered, or unexplained physics as distinguished from well-understood and well-measured physics.

6 http://www-cdf.fnal.gov/.

7 http://www-d0.fnal.gov/.

8 This appears small by 2012 standards, but in 1996 this was a very high data rate.

could operate fast enough to be used by the Level 1 trigger. This algorithm, the eXtremely Fast Tracker (XFT), takes information from the 30,000 readout channels of the central outer tracker, segments the data, and processes it in a highly parallel manner. XFT compares the measured data to a set of all possible masks for high-momentum tracks within each segment. Segments with valid tracklets are then linked together to find tracks. By operating in parallel and doing fairly crude tracking, the XFT can find high-momentum tracks within the 4 μs time limit for the Level 1 trigger.

Data triage is required throughout the network fabric of the Internet, although it is often implemented as a non-discriminating cutoff rather than intelligent triage. For example, consider congestion in Internet routers (Welzl 2005). Routers pass packets of data across different circuits to get the packets to their intended destinations. Routers have a finite capacity to do this packet transfer. They also have finite buffers where packets are queued if the inflow temporarily exceeds routing capacity. However, if the buffer fills up, then packets are summarily dropped (deleted). This is permissible because in Internet Protocol (IP), packet delivery is done on a best effort basis; delivery is not guaranteed. The deletion is generally done regardless of packet content, even though IP packet headers include a "Type of Service" field, which describes priority. A standard data triage scheme, called Random Early Detection (RED), prevents the buffer from filling up (Floyd and Jacobson 1993). With RED, a small fraction of packets are dropped randomly once the queue length exceeds a threshold. As the queue length continues to grow, the fraction of dropped packets is increased and, at some higher threshold, all incoming packets are dropped. The expectation is that a higher-level protocol (such as TCP) will detect the packet loss in their stream due to RED and take preventative action such as reducing their data rate or choosing a different path. This can lead to reduced congestion at the router before the buffer becomes full.

2. Taxonomy of Data Triage

The examples above are instructive about what types of data triage exist. We propose a taxonomy of data triage techniques:

1. *Fully central* – All data are sent to a central location and processed concurrently. This is equivalent to no data triage. Full bandwidth and full computing capability are required, but the computing can be maintained at a single site, and all computations have access to all data.
2. *Layered central* – All data are buffered (pipelined) and a portion of it is sent to the central site for processing. Any data not requested by the end of the buffer are discarded and cannot be recovered.
3. *Sectorized* – A set of local processors are used to make calculations that determine if the full data should be sent to the central site. These local processors only receive data from one sector, but they get *all* the data from that sector. Sectors are almost always defined spatially, but they can be defined by angle,

range, "checkerboard," or anything else that would be useful. The number of sectors can be anywhere from a few to hundreds.

4. *Layered sectorized* – Layering can be done within the sector or between the sector and the central site. This is closest to the CDF trigger.
5. *Local* – The logical endpoint of sectorization is to have each detector make their own decision with their own data. This decision is sent to the central site and fused with other detectors' decisions to make an overall decision.
6. *Nearest neighbor* – Local decision making can also be done with each detector sharing raw data with nearby detectors. The decision is still local, but the detectors have additional data to draw on.

3. Tradespace

In data triage, one must trade several quantities, including timeliness, accuracy, bandwidth, compute power, and maintainability. For the experimental particle physics example, timeliness is critical, and the trigger system must be designed first and foremost to fit within the time constraints of the accelerator and the readout system. Some intelligence, surveillance, and reconnaissance (ISR) collections are also time-critical, and data will be lost if they are not requested before the end of the buffer. As algorithms are optimized for speed, it may be necessary to make assumptions and/or approximations that reduce accuracy. In many cases, this is an acceptable trade, but it should be tracked carefully, especially when systems are put into new environments or against new threats where the assumptions/approximations may no longer be acceptable.

For many systems, the available bandwidth (either point-to-point or a system aggregate) will limit performance. Although it is convenient to assume away bandwidth issues, deployed systems often must operate in non-ideal conditions, including obstructed line-of-sight or significant interference. The resulting loss of bandwidth must be planned for ahead of time. In addition, deployed systems often need to minimize their local computing power due to size, weight, power, or cooling considerations. From a maintenance standpoint having all the processing power at one location where it can be serviced together and failover options can be installed once is a great advantage.

As sensors proliferate, data triage algorithms will become increasingly useful in CBRNE reach-back. For example, deploying large numbers of sensors with small processing capability will require the development of local data triage combined with decision fusion (Section 3.H). Even with more processing or higher bandwidth, it is likely that fully central processing will not be feasible and that layered central algorithms will be necessary.

C. Compressive Sensing

Compressive sensing was introduced in Donoho (2006). Since then, it has received a high degree of interest in applied mathematics. This is because it is based on proofs

that can seem counter-intuitive. Furthermore, under certain conditions, it can be used to reduce sensor costs and/or data acquisition time. In this section, we will explain what compressive sensing is and how it differs from traditional sampling. We will also outline its main advantages and disadvantages and describe remaining challenges.

1. Example

Figure 3-2 illustrates a simple example for compressive sensing. Data are desired at N points along a perimeter. These data could be any signal of interest, such as the intensity of gamma rays emanating from a radiation source, the level of vibrations on the ground as a person walks past, etc. The N data points can be acquired in two ways: (1) N sensors can be distributed along the perimeter to collect data simultaneously. However, this method could be cost-prohibitive if each sensor is expensive; (2) One sensor can travel along the perimeter, collecting data at each of the N points, one at a time. That is, the single sensor can *raster* the N points. However, this can be slow if the data acquisition at each point takes a long time.

Figure 3-2. Data Are Desired at N Points (Black Dots) along a Perimeter (Dashed Line)

Samples can be taken at M points (red dots), where M << N. Prior knowledge can be used to reconstruct the N desired data points from the M samples.

Sampling techniques could be used to reduce sensor costs and/or data acquisition time. As shown in Figure 3-3, data could be sampled from only M of the N points along the perimeter, where M is less than N. The M samples could be used to reconstruct the data at all N points. This could reduce sensor costs, as M, rather N, sensors would be needed to simultaneously collect the samples. This could also reduce data acquisition time, as a single sensor would have to raster M points, rather than N.

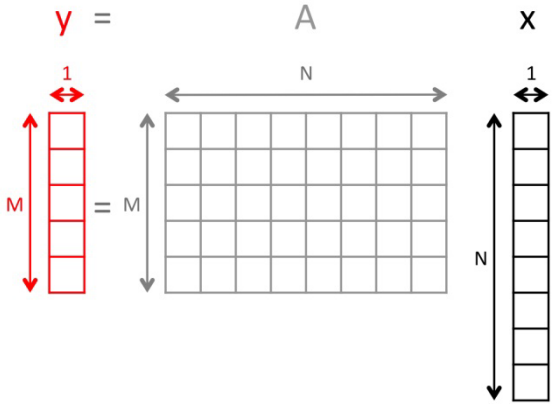


Figure 3-3. Sensing Matrix A Describes Relation between M Samples in y and N Desired Data Points in x

There are infinitely many solutions for x, since M is less than N. Under certain conditions, prior knowledge can be used to select the correct solution.

Matrix algebra can be used to describe how the N desired data points can be reconstructed from the M samples. First, the N desired points can be arranged into **x**, an Nx1 column vector. This is the signal that must be reconstructed from the samples. Similarly, the M samples can be arranged into **y**, an Mx1 column vector. The samples, **y**, can be written in terms of **x**, the desired data points:

$$\boldsymbol{y} = \mathbf{A}\mathbf{x} \tag{1}$$

where **A** is the M×N *sampling matrix*. The rows of **A** describe how, during data acquisition, the data at the N desired points are combined to form each of the M samples. Figure 3-3 illustrates Equation (1) in graphical form.

Equation (1) must be solved for **x**. However, this problem is ill posed: there are fewer samples (M) than there are unknowns (N). As a result, there are infinitely many solutions to Equation (1). Under certain conditions, prior knowledge about **x** can be used to select the correct solution.

2. Traditional Sampling

Traditional sampling theory describes how to select the correct solution to Equation (1) based on prior knowledge of the *bandwidth* of **x** (Karu 1995). The bandwidth of **x** is specified by its highest frequency, B cycles per meter. The Nyquist-Shannon Theorem states that in order to perfectly reconstruct **x** (the N desired data points along the perimeter), the maximum distance between samples must be no larger than $\frac{1}{2B}$ meters. (2B is often referred to as the *Nyquist rate*.) This maximum spacing between samples, along with the total length of the perimeter P, can be used to determine M, the minimum number of required samples:

$$M = {}^{P}\!/_{\left(\frac{1}{2B}\right)} = 2PB \tag{2}$$

For a band-limited signal, if at least M = 2PB samples are collected, spaced no farther than $\frac{1}{2B}$ meters apart, then we can select the correct solution to Equation (1). An M much less than N can significantly lower sensor costs and/or data acquisition time.

Traditional sampling theory requires that **x** is band-limited (Karu 1995). That is, **x** can have no frequencies higher than B cycles per meter, for some value of B. Not all signals meet this criterion, however. Fortunately, one can impose this constraint artificially. That is, one may first apply a low-pass, *anti-aliasing* filter to the signal before it is sampled. A low-pass filter with a cut-off frequency of B cycles per meter will remove all frequencies in the signal higher than B cycles per meter, thereby ensuring that the data do indeed have a bandwidth of B cycles per meter. Of course, this can only be done if the high frequencies filtered out of **x** are not important for further processing. Thus the anti-aliasing filter must be designed with a specific application in mind.

The number of required samples specified by traditional sampling theory, M = 2PB, can still be quite high. A large bandwidth, B, and/or a long perimeter, P, can lead to a large number of required samples, M. As a result, the sensor costs and/or data acquisition time may still be high. Under certain conditions, the number of required samples M may be reduced even further. Compressive sensing theory deals with one particular set of conditions.

3. Compressive Sensing

Compressive sensing can also be used to select the correct solution to Equation (1), sometimes using fewer samples, M, than traditional sampling theory (Candes and Wakin 2008). This is accomplished by using prior knowledge about the sparsity of **x**, rather than its bandwidth. The sparsity of **x** refers to how many values of **x** (how many of the N desired data points along the perimeter) are non-zero. For example, a ground vibration sensor may record non-zero data only if a person walks by; sensors positioned farther away will all record zero.

Provided that we have prior knowledge that **x** is sparse, compressive sensing theory states that, with high probability, the correct solution to Equation (1) is the one with the fewest non-zero elements (the smallest "L_0 norm") (Baraniuk 2007). Therefore, we can reconstruct **x** by minimizing the L_0 norm, subject to Equation (1). Minimizing the L_0 norm is difficult, however, as it is an "NP-hard problem" and can only be solved exactly using computationally intensive, brute-force techniques. Alternatively, compressive sensing theory also states that, with high probability, the correct solution to Equation (1) can also be found by minimizing the L_1 norm, rather than the L_0 norm, of **x**. The L_1 norm is the sum of the absolute values of the elements of **x** (the sum of the absolute values of the N desired data points). At first glance, this result may appear counter-intuitive, since traditional methods are often based on minimizing the L_2 norm, the sum of the *squares* of the elements in **x**. Minimizing the L_1 norm is relatively easy to compute, requiring straightforward methods in linear algebra.

Compressive sensing requires that **x** is sparse (Baraniuk 2007). That is, only S of the N unknown data points can be non-zero. If this condition is not met, then minimizing the L_1 norm will not be an adequate method for choosing the correct solution to Equation (1). Not all signals are sparse. However, this criterion can still be met if there is prior knowledge that the signal is, in fact, sparse in some other domain. In our perimeter example, the signal **x** may not be sparse in the space domain, since it may not be zero at any point along the perimeter. However, the signal may be sparse in the frequency domain, since it may be composed of only one or two pure sine waves, with the energy in all other frequencies equal to zero. This constraint can be imposed artificially through use of an N×N *sparsifying matrix*; the columns of the matrix represent the basis functions of the domain in which the signal is known to be sparse. Thus the sparsifying matrix in compressive sensing performs a similar function to the anti-aliasing filter in traditional sampling. Here, though, the signal is first sampled, then the signal is reconstructed *in the*

domain in which is it known to be sparse, and finally, the reconstructed, sparse signal is transformed back into its original domain, through use of the sparsifying matrix. Of course, this can only be done if it is known in what domain the signal is sparse.

Compressive sensing theory can be used to determine the minimum number of samples required to reconstruct **x** with high probability:

$$M = 2u^2 S\log(N) \tag{3}$$

where S is the sparsity of **x** (the number of desired data points that are non-zero) and u is the *incoherence* between the sampling and sparsifying matrices (the amount of additional information that each new sample provides about the N desired data points). As illustrated by Figure 3-3, the i^{th} row of sensing matrix **A** governs how the N desired data points in **x** are combined to form the i^{th} element of **y**. At first glance, it may appear that sensing matrix **A** must therefore be carefully designed using prior knowledge of **x**; however, randomly choosing the values of **A** has been shown to work just as well (Candes and Wakin 2008). This is because a random matrix **A** will, with high probability, have rows that are different from each other. Therefore, each sample will be based on a different combination of the N desired data points, providing new information about them. For the most part, then, the sampling matrix **A** can be designed with little to no prior knowledge of **x**. This result may seem counter-intuitive – with traditional sampling, one must know the bandwidth of **x** in order to determine the maximum spacing between samples.

4. Advantages and Disadvantages to Compressive Sensing

Compressive sensing has many advantages over traditional sampling:

- Under some conditions, compressive sensing can result in fewer required samples. For example, a signal may be very sparse (requiring few samples with compressive sensing) even though it has a very large bandwidth (requiring many samples with traditional sampling). As a result, compressive sensing can lead to lower sensor costs and/or data acquisition time.
- Designing the sensing matrix **A** is relatively simple – a random matrix can be used regardless of the signal **x** (and can therefore be used for other signals in the future). Thus the sensing, which is performed in the field, is decoupled from the reconstruction, which can be done back at headquarters.
- Compressive sensing can ease encryption. Samples can be collected in the field and then sent back to headquarters for reconstruction. Reconstruction requires knowledge of the sensing matrix **A** or, at the very least, the seed of the random number generator that was used to create **A**. Even if an adversary were to tap into the samples as they were sent back to headquarters, the adversary would not be able to reconstruct the desired signal without knowing **A** (or its seed). This seed is therefore all that needs to be encrypted; the samples themselves do not.

- Compressive sensing can be used for compression. The M samples can be used to represent the N desired data points, just as the JPEG compression scheme is used to compress traditionally acquired images for easier storage. In traditional compression, all N desired data points are acquired; these N points are then processed to extract M representative features. Compressive sensing, on the other hand, acquires only the M representative features (the M samples) without first having to acquire the N desired data points. This is where compressive sensing gets its name: sensing the compressed signal.

Compressive sensing has one large disadvantage: the data *must* be sparse in some domain. Otherwise, minimizing the L_1 norm will not be adequate for choosing the correct solution to Equation (1). In many situations, a sparsifying matrix can be used to transform the signal into another domain in which it is indeed sparse, thereby satisfying this criterion. However, there are other situations in which a sparsifying matrix simply cannot be found. This often occurs when the data have a very low signal-to-noise ratio. In such cases, compressive sensing cannot be used to reduce the number of required samples enough to make a significant dent in either sensor costs or data acquisition time. In these situations, it may be better to perform traditional sampling or no sampling at all.

5. Remaining Challenges in Compressive Sensing

The theory of compressive sensing has been refined over the past several years, and many peer-reviewed journal articles have been written by applied mathematicians. However, few articles describe how compressive sensing has been used in real-life applications. As a result, few lessons learned have been shared about the "art" of compressing sensing. Therefore, this field could benefit greatly from published research in compressive sensing applications, particularly in the following areas:

- What considerations should be taken when designing a sparsifying matrix for a particular application?
- What conditions make the algorithm particularly sensitive? Robust?
- How can one determine M, the minimum number of required samples, if S, the sparsity of **x**, is not explicitly known?
- How can one relate the signal-to-noise ratio of **x** to its sparsity S, in order to predict how successful compressive sensing will be for a particular application?
- How can a particular application exploit the analog processing inherent to compressive sensing? By its nature, compressive sensing performs part of its analysis in the analog domain, as each new sample is a sum of the N desired data points, randomly weighted. This summation could occur in the digital domain; however, performing this summation in the analog domain saves computational power. Magnetic Resonance Imaging and the Rice Single Pixel camera are examples of applications that exploit this analog processing (Baraniuk 2007;

Lustig et al. 2008). Further work should explore what other types of applications could exploit this analog processing as well.

D. Data Assimilation

Data assimilation, in the most general sense, is the use of all available sensor data to reconstruct, as accurately as possible, the properties of the environment of interest. In the most frequently encountered usage, data assimilation is more specifically the use of all available sensor data *in combination with a physical model* to reconstruct, as accurately as possible, the properties of the environment of interest. In addition, a forecast step is usually implied, in which observations and the physical model are used to predict the future evolution of the environment of interest.

Data assimilation techniques are generally applicable to problems in which (1) one wants to determine the properties of the environment of interest, (2) the spatial and temporal variation of those properties is governed by a physical model, and (3) the observational data points are fewer in number than the grid points in the physical model and in different locations. A simple example for which data assimilation is applicable is the following. Imagine an urban environment where one is performing a field experiment for which he needs to know the wind velocity accurately at all spatial locations up to some height above ground level and needs to be able to make forecast predictions of the wind velocity over the same domain. Further imagine a few dozen sensors that measure wind velocity and pressure are placed at several different locations in the environment. In this case, the model governing the properties of interest is fluid dynamics – the physics of fluid flow. In particular, a detailed numerical computational fluid dynamics (CFD) model is used since fluid flow in urban environments is complex. The number of data values (multiple variables at each grid point) needed to run the CFD model forward in time may be 10^6 (a million) or more, yet there are only tens of sensor data points. Data assimilation techniques allow one to combine the sensor data and the physical model to reconstruct as accurately as statistically possible the wind velocity throughout the whole urban environment, even in locations far from sensors.

The primary reason this is possible is because locations in space (and time) are coupled via the physical model. For example, if one knows the wind profile on the windward side of a building, one can reconstruct the wind profile on the leeward side since fluid dynamics tells us how air flows around obstacles. In addition, sensors measuring data that are not of direct interest, such as the pressure measurements in this example, are used in the data assimilation process to increase the accuracy of the reconstructed wind field, since pressure is a key part of fluid dynamics. Data assimilation could further be used in this example to make forecasts for how the wind field will change with time. In such uses, the accuracy typically improves with time. Finally, very recent work has shown that it is possible to localize in space and time the source of the uncertainty of a reconstruction or forecast. In some cases, it is possible to reposition sensors to those locations to improve the accuracy of the reconstruction or forecast. For

our urban wind example, one can imagine small, sensor-equipped unmanned aerial vehicles (UAVs) or soldiers with sensor backpacks that could be repositioned.

Data assimilation is most mature in the field of numerical weather prediction. Many climate centers across the globe perform data assimilation on a continuous basis. They assimilate observations from weather stations and other sensors to reconstruct current weather conditions, track storms, and make weather forecasts. This data assimilation is performed with the aid of fairly detailed weather models, which include the physics of fluid flow, evaporation, radiative transfer, and a host of other physics. Data assimilation is also used extensively in the field of ocean modeling, for which the physics is similar. For example, the Naval Oceanographic Office (NAVOCEANO) uses data assimilation in the production of its many oceanographic data products. Data assimilation techniques have recently been applied to the CO_2 inversion problem, in which CO_2 flux between the earth's surface and the atmosphere is reconstructed. The CO_2 flux physics model is essentially atmospheric transport.

In discussing data assimilation, we have in mind the four operational missions identified as relevant to DTRA's mission, as described in Chapter 1: direct force protection, targeted search, wide-area search, and long-term threat-behavior monitoring. In direct force protection, one might be interested in detecting and tracking the flow of hazardous chemicals in the air in a region encompassing a forward operating base. This flow is governed by the physics of atmospheric transport and dispersion. Data assimilation could be used to obtain the most accurate quantitative reconstruction of the concentration of a hazardous chemical and forecasts of that concentration. These could be used to calculate the locations of hazard areas in which forces should take protective measures or evacuate. Higher-level sensor fusion algorithms could be used to detect a threat condition from the reconstructed concentration field or forecasts, and even issue warnings.

Data assimilation could be used in targeted search, an example of which is the determination of the spatial origin for a radiological attack in a city-sized geographical area. Gamma ray detectors could be employed, in combination with weather and chemical sensors. The physics governing the spread of a radiological threat such as a dirty bomb is atmospheric transport and dispersion, radiative transfer, and light-matter interaction. Data assimilation could be used to combine the observations and the physical model to determine the spatial origin of the threat (which, in the case of a dirty bomb, would be an extended area), and forecast radiation levels throughout the environment. As described below, the 4D-Var (four-dimensional variational) data assimilation technique is particularly well suited to the problem of determining the origin of a threat.

An example of wide-area search that data assimilation could benefit is the search for a moving hazard. Imagine a weapon has been stolen and is being transported in a vehicle, and the weapon emits a radiological signal that can be detected by a network of sensors scattered throughout a large road network. In this case, the most relevant model is a road network model, which constrains the possible locations of the vehicle, followed

by a physical model of radiation propagation. This model can be combined with the sparse sensor observations via data assimilation to obtain a vehicle location estimate that is as accurate as statistically possible. Forecasts of the vehicle's location could also be made.

Finally, data assimilation techniques could also be used for long-term threat-behavior monitoring. Say one is interested in detecting behavior that is out of the ordinary for the purposes of hypothesis generation. One example of this might be traffic patterns in a city. Data assimilation techniques could be used to combine sensor data (e.g., loop detectors and mobile phone signals) with a physical model to determine current traffic flow and forecast future traffic. These products could then be used by higher-level sensor-fusion algorithms to detect changes or generate hypotheses.

Below we describe the landscape of data assimilation techniques. The discussion is more descriptive than formal, although we note that these techniques fit within a formal probabilistic framework. The start of the discussion below follows the development in Kalnay (2003). Note that all the algorithms described are sensor-agnostic.

1. The Successive Corrections Method and Nudging

The successive corrections method (SCM) and nudging are perhaps the most primitive data assimilation techniques. SCM is an empirical technique in which corrections are successively applied iteratively to an initial guess of the background field on a grid. (Here we are using standard terminology. The background field is one component of what we have called the "properties of the environment" above. An example is the temperature at all points in space.) The corrections are composed of the difference between the value of the field at a grid point and the values of the observations nearby, weighted by coefficients that are chosen empirically. SCM is simple and fast and can provide reasonable estimates of the field.

Nudging, or Newtonian relaxation, is a technique that incorporates a physical model. In the equations governing the physical model, a term is added that is a function of the difference between the field estimate and the observations. This term is a forcing term that drives the solution of the model equations toward the observations. This method is empirical and has shown mixed success.

2. Least Squares Methods

Many of the most popular data assimilation techniques are least squares methods, in which some quantity is minimized to provide the optimal estimate of the field of interest. We describe these below, but first we must define a few terms. What we have been calling the optimal estimate of the field, specified as values at grid points, is called the "analysis" and is denoted by the vector $\mathbf{x}_a$. The initial guess of the field is called the "background" and is denoted by the vector $\mathbf{x}_b$. The observations are denoted by the vector $\mathbf{y}_o$. Since the background may be a different physical quantity from what is observed (e.g., pressure vs. displacement of a piston), one needs a method to convert

between model grid values and observation values. This tool is called the "forward observational operator" and is denoted by *H*. The master equation that generally describes all least squares methods is

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{W}[\mathbf{y}_o - H(\mathbf{x}_b)]\,, \tag{4}$$

where **W** is the weight matrix (also called the gain matrix **K**), which optimally adds the differences between the background and observations to the background to obtain an analysis that is as accurate as possible. This equation can be used to understand optimal interpolation, extended and ensemble Kalman filtering, 3D- and 4D-Var, and techniques based on these algorithms.

The analysis is equivalent to what we called "field reconstruction" above. In forecasting, the analysis is input into a forecast model (typically a physics evolution model), which outputs the properties of the field at some specified future time.

In addition to obtaining the best analysis possible, the uncertainties in the analysis are extremely important for most applications. We will describe uncertainty propagation below. Note that uncertainties are typically presented in what are known as "error covariance matrices." They specify the covariance between each pair of variables represented in the matrix and essentially represent all we know about our uncertainties, to first order.

a. Multivariate Optimal Interpolation

In multivariate optimal interpolation (MVOI), the weight matrix **W** is obtained by minimizing the sum of the squared errors (sum of the variances) of the analysis variables at each grid point with respect to the elements of **W**. To be explicit, if we define the analysis error as $\boldsymbol{\varepsilon} = \mathbf{x}_a - \mathbf{x}$, where **x** is the unknown true field, then we obtain w_{ij}, the elements of **W**, by solving

$$\frac{\partial \sum_{k=1}^{n} \epsilon_k^2}{\partial w_{ij}} = 0\,, \tag{5}$$

where the sum is over *n* field grid point variables, *i* refers to those same *n* field variables, and *j* refers to $p<n$ observational grid point variables. In practical implementations of MVOI, the forward observational operator **H** is linearized so that **W** can be expressed in a simple matrix algebraic expression involving **H**, the linearized version of *H*, and the error covariance matrices for the background and the observations.

As of 2005, MVOI was used at the Fleet Numerical Meteorology and Oceanography Center (FNMOC) and NAVOCEANO as their standard data assimilation technique (Cummings 2005). MVOI was used at weather forecast centers as recently as the mid-1990s.

b. 3D-Var

In 3D-Var, a cost function is minimized over the analysis variables (the elements of $\mathbf{x}_a$). If the cost function is chosen appropriately, the value of $\mathbf{x}_a$ at minimum is the optimal analysis. In particular, a cost function like the following is often used:

$$J(\mathbf{x}_a) = \frac{1}{2}(\mathbf{x}_a - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x}_a - \mathbf{x}_b) + \frac{1}{2}[\mathbf{y}_0 - H(\mathbf{x}_a)]^T \mathbf{R}^{-1}[\mathbf{y}_0 - H(\mathbf{x}_a)]\,, \quad (6)$$

where **B** is the error covariance matrix for the background, **R** is the error covariance matrix for the observations, and the minimization is performed with respect to $\mathbf{x}_a$. In practice, this minimization is performed numerically with efficient multi-dimensional minimization algorithms (such as the conjugate gradient method). Since $\mathbf{x}_a$ typically has many fewer elements than **W**, it is usually much less computationally expensive than MVOI and therefore preferred. Note that since the minimization is performed with respect to $\mathbf{x}_a$ the master equation is never used, although for the value of $\mathbf{x}_a$ obtained the solution of **W** would be the same as in MVOI. In fact, it has been proven that MVOI and 3D-Var are solving the same problem (Lorenc 1986). An additional benefit of 3D-Var is its smoothness, since one does not have to choose which observations correspond to which grid points as in MVOI (Kalnay 2003).

c. Extended Kalman Filtering

The Kalman filter is perhaps the most well-known of the techniques we have discussed up to this point. The essence of the Kalman filter is the combination of a physical model with observations to make a better estimate of a system's state than could be made via observations alone. In particular, the standard Kalman filter assumes the system is a discrete, linear, stochastic process, and estimates the system's state in such a way as to minimize the covariance of that estimate. The Kalman filter is typically used in forecast-update cycles in which the best estimate of the current state of the system (the "analysis") is advanced to a future time via a physical model to create the forecast. When that time arrives, another observation of the system is made and combined with the forecast to create a new analysis. That analysis is then propagated forward in time to create a new forecast, etc.

The Kalman filter differs from MVOI in that, in a forecast-update cycle, the error estimate (covariance) of the forecast is updated at each step using the physical model. For linear systems, updating the forecast error covariance is easy, since the physical model is simple and there are typically only a few degrees of freedom. The physical model is then represented by a modestly sized matrix, and the forecast error covariance is given by a simple linear algebraic expression involving that matrix.

The extended Kalman filter allows the physical model to be nonlinear. The forecast step is performed with the full non-linear model. The forecast error covariance, however, is estimated using a linearized version of the physical model, known as the tangent linear model (TLM), and its transpose, known as the adjoint. Physical models such as weather

models or atmospheric transport and dispersion models typically have $N=10^6$ or more degrees of freedom, meaning their TLM is a matrix with $N^2=(10^6)^2=10^{12}$ elements! Matrices of this size are extremely unwieldy and fit in memory only on the largest supercomputers. In addition, generating them requires running the non-linear model roughly $N=10^6$ times, which is impossible without significant computing capabilities. Needless to say, the extended Kalman filter does not find widespread use for systems with complex non-linear models. However, it does find use for smaller, simpler systems – it is the standard for GPS and navigation systems generally. Additional applications of the extended Kalman filter are discussed in the case studies on autonomous ground vehicles (Section 4.C), sense-and-avoid for unmanned aerial vehicles (Section 4.D), and hurricane/storm track forecasting (Section 4.G).

d. Ensemble Kalman Filtering

Ensemble Kalman filtering solves the major problem of extended Kalman filtering by estimating the forecast error covariance in a different way. In ensemble Kalman filtering, an ensemble of models in which random perturbations have been added to the assimilated observations is run forward in time (Evensen 1994). The forecast error covariance is then estimated using the variability in the ensemble. In particular, the forecast error covariance is usually taken as something approximating the average of the forecast error covariances (with respect to the ensemble mean) for each model in the ensemble. The great benefit of the ensemble Kalman filter is that an ensemble may contain only a few tens to hundreds of members, requiring many fewer model integrations than the 10^6 or more that would be required in extended Kalman filtering. Ensemble Kalman filtering is thus one of the most practical data assimilation techniques, and finds widespread use.

e. 4D-Var

4D-Var is an extension of 3D-Var in which the cost function includes forecasts and observations distributed throughout a time interval. In this way it measures differences between the model and observations over several time slices. The cost function is typically minimized with respect to the state of the system at the start of the time interval (since most physical models cannot be run backward), with the analysis being the initial system propagated forward to some future time via the physical model. For certain assumptions (e.g., a perfect physical model, correct initial background error covariance), it can be shown that 4D-Var is equivalent to the extended Kalman filter. As of 2007, several weather forecasting centers, including the European Centre for Medium-Range Weather Forecasts (ECMWF), France, the United Kingdom, Japan, and Canada had switched to 4D-Var.

f. Ensemble Kalman Filter Variants

Because the ensemble Kalman filter is so capable, several variants of it are in use. Due to space constraints, we simply list some of them briefly.

In the local ensemble transform Kalman filter (LETKF), instead of running an ensemble of *global* models, ensembles of *localized* models are run. This allows for much greater computational speed and efficient parallelization.

4D-LETKF, because it operates on multiple time slices, has some of the same smoothing properties as 4D-Var. In particular, it allows for faster spin-up. Spin-up is the period after starting a series of data assimilation cycles in which the forecast error is decreasing before reaching a roughly steady-state value.

"Running in place" (RIP) is a technique used when starting a LETKF to shorten its spin-up time (Kalnay and Yang 2010). Kalnay (2010) demonstrates a LETKF-RIP spinning up in half the number of cycles required for 4D-Var. Quick spin-up data assimilation techniques such as LETKF-RIP are particularly useful in cases where a new phenomenon of interest (e.g., a hurricane or a chemical threat) is detected, and forecast accuracy is desired early in the data assimilation series. LETKF-RIP is also one of the best-performing techniques for sparse observations.

E. Anomaly Detection

*Anomaly (*or *outlier) detection* is the problem of finding observations or patterns in data that exhibit unexpected behavior (Chandola, Banerjee, and Kumar 2009). Detecting anomalies is important because for a variety of applications, "unusual" observations correspond to significant, actionable information. Heuristically, anomaly detection corresponds to finding patterns in data that do not correspond to a well-defined description of "normal" behavior. Consider Figure 3-4, which illustrates this heuristic definition. N_1 and N_2 are considered normal, since most of the data are in these regions. Points o_1 and o_2 and the points in region O_3 are anomalies because they are "sufficiently far away" from the bulk of the data.

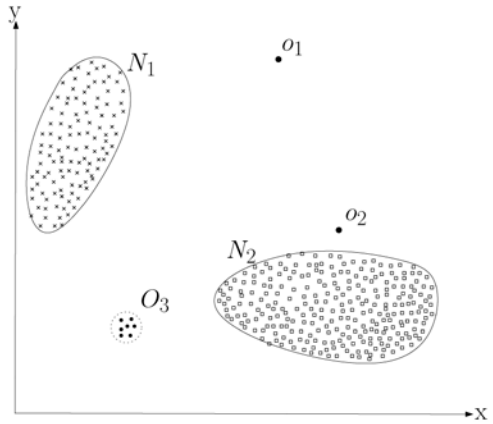


Figure from Chandola, Banerjee, and Kumar (2009).

Figure 3-4. Example of Anomalies in a Two-Dimensional Space

Given the heuristic definition of an anomaly as something that does not exhibit "normal" behavior, a seemingly straightforward approach to anomaly detection is to

define a normal region (in terms of some set of variables) and label anything outside the region as an anomaly. However, a number of challenges face this approach:

- Describing normal behavior may be extremely complex.
- Defining a region that encompasses all possible normal behavior is difficult. Furthermore, the boundary between normal and anomaly may not be precise.
- The definition of normal may evolve over time.
- The definition of anomaly may be different from application to application (e.g., small fluctuations may be important in one domain and negligible in another).
- The availability of labeled training data for developing and validating models is frequently limited.

These challenges have led to a number of approaches to anomaly detection. The approaches can be roughly categorized by the type of anomaly to be detected and the approach to building a model of "normal."

Broadly, anomalies can be categorized into three groups: *point anomalies*, *contextual anomalies*, and *collective anomalies* (Chandola, Banerjee, and Kumar 2009, 2007). A point anomaly is an individual data point that is exhibiting abnormal behavior (e.g., points o_1 and o_2 in Figure 3-4). These are the simplest kinds of anomalies, and much research effort has focused here.

A contextual anomaly is a point anomaly that is anomalous in one context but not another. This is easily illustrated by example. A temperature of 32 °F in Washington, DC, is normal in February but anomalous in July. When contextual anomalies need to be identified in data, identified attributes must define both the context (*contextual attributes*) and the noncontextual characteristics (*behavioral attributes*). In the example, the month and location are contextual attributes, and the temperature is the behavioral attribute. Contextual anomalies have been studied primarily in time series and spatial data.

A collective anomaly is a collection of related data instances that are anomalous with respect to the entire data set. The individual data points may not be anomalous on their own. An example of a collective anomaly is a cyber attack: a specific sequence of commands must occur that, on their own, would not be unusual. Collective anomalies make sense only when the data are related in some way – for example, in time or in space. Consequently, collective anomalies have been most studied in sequence data, spatial data, and graph data (which are useful for modeling very general relationships among entities).

There are three primary approaches to building a model of "normal" behavior. The first is to learn what is normal directly from the data. This leads to the classification and clustering algorithms discussed in Section 3.G. These algorithms can take three different approaches. (1) *Supervised learning* uses a training set to learn the optimal mapping from features to categories. (2) *Unsupervised learning* does not use a training set. Instead, the features are organized into discrete clusters. Then, an expert human is

needed to determine which clusters represent which categories. (3) *Semi-supervised learning* borrows methods from both supervised and unsupervised learning. For example, one could use a clustering algorithm to develop clusters, choose a few points from these clusters and develop ground truth category labels, and then use the labels to map clusters to categories.

As an example of a classification approach applied to detecting a collective anomaly, Lee and Stolfo (1998) discuss anomaly detection in the context of intrusion detection. Particularly, they are interested in distinguishing between "normal" or "legitimate" behavior from "misuse" or anomaly actions, where the misuse is an intrusion on network-based computer systems. Determining when a system is under attack can be crucial for preventing intrusion. To differentiate between these behaviors, the authors use a classification tree-based model. This method relies on supervised learning, so the training data set must contain some examples of the abnormal behavior it is attempting to detect. In many ways, this is a straightforward application of data mining. If the necessary data are available, the challenge lies in finding sensible ways to identify relevant sequences in the data. This method is most applicable for situations where a substantial amount of data, both "normal" and "abnormal," are available.

The second two categories of anomaly detection methods are statistical techniques. In the first, an empirical model of "normal" is built using "current" data. Detecting an anomaly is then framed as changepoint detection. Applications of these methods to syndromic surveillance are discussed in Section 4.E. For example, Hagen et al. (2011) discuss two methods for detecting the outbreak of H1N1 in California in 2009. One approach uses a straightforward moving average for a univariate response (e.g., new mentions of flu-like symptoms in medical records) as a baseline. Comparisons are made using daily records against the baseline, and "signals" are counted when the daily value exceeds the moving average by some threshold. A second approach adapts methodology from the quality/statistical process control (SPC) literature, such as cumulative sum (CUSUM) charts. An important difference between standard SPC methods and biosurveillance is that, while SPC methods sample and consequently work with independent sequences of data, biosurveillance data are always highly autocorrelated (Fricker 2013).

In the second statistical approach, "normal" is explicitly modeled. This may occur through use of a parametric or nonparametric empirical model, or it may be through a combination of physics-based and empirical models. Anomaly detection is then a hypothesis testing problem. For example, Jacob and Levitt (2003) discuss detecting cheating in Chicago public schools. Because of performance incentives, teachers may cheat on behalf of their students in a variety of ways in order to improve students' test scores. The authors hypothesized some methods teachers might use to cheat and, based on those, developed a model for what "cheating" would look like on tests. For example, if a teacher were systematically improving his class's scores through cheating, his students would be expected to show an increase in their year-to-year scores while in his

class, but those gains would disappear the following year. Thus, they examined year-to-year data, estimating a "baseline" for each student and determining when large deviations from those baselines were associated with particular teachers. It was also possible to look for examples of cheating by examining classroom-level data. If the correlation in answers to some questions were much higher than the correlations for other questions on the same exam, this could indicate that the teacher for that class was either feeding students answers to those questions or changing the answers to those particular questions after the test. By establishing norms using data from the entire school district and across many years, it was possible to establish a baseline for the expected variability in question-to-question correlation within a single classroom. Large deviations from this were taken to be instances of cheating.

F. Dimension Reduction

Once an anomaly is detected, it can then be classified. However, an interim step is frequently needed: the data collected for the anomaly must often be preprocessed before they can be input into a classification algorithm. This is because a large amount of data can be collected for each anomaly. For example, an anomaly can be detected in a frame of the video feed from a camera. A 256×256-pixel frame contains 65,536 different values of data. One could arrange these 65,536 values into a 65,536-dimensional vector and then input that vector into a classification algorithm. Training an algorithm to properly classify such high-dimensional vectors is difficult, however. As Duda, Hart, and Stork (2001) explain, "high-dimensional [classification] functions have the potential to be much more complicated than low-dimensional ones, and…those complications are harder to discern" (p. 170). As a result, exponentially more vectors are needed to train a classification algorithm in high-dimensional space. Furthermore, the iterative processes used to optimize the parameters of a high-dimensional classification algorithm can take much longer to converge. This phenomenon is called "*the Curse of Dimensionality*" (Duda, Hart, and Stork 2001; Bishop 1995).

Dimensionality reduction is the pre-processing step that is required between anomaly detection and classification. The data collected for the anomaly are summarized by fewer variables. These summary variables are called *features*. There are three main approaches to dimensionality reduction:

1. *Heuristics* can be used to measure characteristics of the data collected for an anomaly. Each measured characteristic is a feature. The heuristics are formulated based upon prior knowledge of the expected characteristics of the detected anomaly, gained from an understanding of the physical phenomena that is expected to have produced the anomaly in the first place.
2. *Feature selection* techniques can rank each collected datum according to its measured or estimated ability to enable classification. Only those data with high ranks are then retained as features. All other data are discarded.

3. *Feature extraction* techniques can find linear or non-linear combinations of the data that can best enable classification.

1. Heuristics

Heuristics can be used to measure characteristics of the data collected for an anomaly. These heuristics must be based upon prior knowledge of the anomaly's expected characteristics. This prior knowledge comes from an understanding of the physical phenomena explaining how the anomaly was produced.

Consider the video camera example discussed above. If it is known that the detected anomaly is likely to be a person's face, then image-processing techniques can be used to identify the edges surrounding the person's head, as well as the their eyes, nose, and mouth. The cross-sectional area of the person's face, as well as the distance between his or her eyes, nose, and mouth, can then be measured. These measurements serve as the features summarizing the detected anomaly (the person). The 65,536 data values collected for the anomaly have now been reduced to only four features. The four-dimensional feature vector can now be input into an algorithm to classify the person as either a known suspect (threat) or not a known suspect (clutter). Training a classification algorithm in four-dimensional space is a much easier task than in a 65,536-dimensional space. Fewer vectors are needed to train the classifier, and the training process can converge much more quickly.

Similarly, in the medical exemplar discussed in Section 4.B, an implantable cardiac device monitors the signals recorded from a patient's heart. Fast heart rhythms are detected as anomalies. The signals recorded during a fast heart rhythm can be hundreds of samples long – too many samples to classify without pre-processing them first. Therefore, the "onset," "stability," and/or "morphology" of the fast heart rhythm are measured. These features are based on a physiological model of the origin of electrical depolarization in the beating heart. Only then can the features be input into an algorithm to classify the fast heart rhythm as either lethal (threat) or non-lethal (clutter).

Heuristics for dimensionality reduction have both pros and cons. One advantage is that the features measured from the data are easy to interpret since they are based on a physical model of the phenomena that was likely to have produced the anomaly. One disadvantage of this approach is that the heuristics can only be set if one has an understanding of the physical phenomena governing the application.

2. Feature Selection

Feature selection does not require prior knowledge of the physical phenomena governing a particular application. Instead, dimensionality reduction is performed by retaining only a subset of the data collected for the anomaly. Each datum is ranked according to its measured or estimated ability to enable classification. Only those data with the highest ranks are retained as features and input into the classification algorithm. All other data are discarded. Feature selection can be wrapper-based or filter-based.

a. Wrapper-Based Techniques

Wrapper-based techniques rank the data based on how well they enable the classification algorithm to perform correctly on new previously unseen anomalies (Guyon and Elisseeff 2003). First, the classification algorithm is trained on a set of anomalies, using all available data. Then, the trained classifier is applied to new previously unseen anomalies, again using all available data. (Cross-validation, or the "leave-N-out train-and-test" method, can be used here.) The classification performance of the algorithm, when operating on all available data, is assessed and quantified. This is the baseline performance. Next, the classification algorithm is retrained, this time using all but one of the available data. The retrained classifier is then applied to new, previously unseen anomalies and its classification performance is reassessed. The rank of the datum that was *not* used for classification is then estimated based upon the difference between the retrained and baseline classification performances. This process iterates until each datum has been assigned a rank. Only those data with the highest ranks are retained as features for further analysis. All other data are discarded.

Wrapper-based techniques for feature selection have both pros and cons. One advantage is that the data are selected based upon direct measurements of their ability to enable classification. The disadvantage is that wrapper-based techniques are computationally intensive. The classification algorithm must be trained and retrained in order to assign a rank to every datum. This can take a significant amount of time and computational power.

b. Filter-Based Techniques

Filter-based techniques are often used to avoid the time and computational power required by wrapper-based techniques (Guyon and Elisseeff 2003). While wrapper-based techniques rank the data based on the algorithm's measured classification performance, filter-based techniques rank the data based on some surrogate for classification performance. The *Fisher score* is the most well-known technique for filter-based feature selection. The Fisher score is the rank assigned to each datum, one by one, based upon its ability to minimize the difference between anomalies from the same class while maximizing the difference between anomalies from different classes (Gu, Li, and Han 2011). Only those data with the highest ranks are retained as features; all other data are discarded.

Although the Fisher score is quick to compute and easy to interpret, it does not lead to the optimal subset of features. This is due to the fact that the rank of each datum is calculated independently. That is to say, two data may each have a high rank. In such a case, both data will be retained as features. However, these two features may be highly correlated with each other, introducing an unnecessary redundancy into the classification scheme. On the other hand, two data may each have a low rank and, therefore, will both be discarded. Such a data discard would be unfortunate since together, the two data could have provided strong classification ability. To avoid this pitfall, the Fisher score

has been generalized such that a subset of features can be found that, together, maximize the lower bound of the traditional Fisher score. It can be shown that this subset of features is optimal (Gu, Li, and Han 2012).

Note that both the traditional and generalized Fisher scores are examples of *supervised learning*. As will be discussed in Section 3.G for classification, supervised learning requires a training set of data collected from several anomalies. Each anomaly is represented in the training set by two pieces of information: (1) the collected data and (2) the label of the class to which the anomaly truly belongs. These ground truth class labels are needed to differentiate between anomalies in the same class versus anomalies in different classes.

3. Feature Extraction

Feature extraction is another method for dimensionality reduction. Unlike feature selection, feature extraction does not simply find a subset of the original data. Instead, it finds a linear or non-linear combination of data that best summarizes the characteristics of the detected anomaly. Feature extraction can be supervised or unsupervised, linear or non-linear, and based on local or global properties of the collected data.

a. Supervised

The *Fisher criterion* is the most well-known technique for supervised feature extraction. For a problem in which there are N possible classes, the high-dimensional vectors of data collected for each anomaly in the training set are projected down onto an N-1-dimensional subspace. The features retained for further analysis are the coordinates of the high-dimensional data vectors projected down onto the N-1-dimensional subspace. The subspace is found to minimize the covariance between feature vectors in the same class and maximize the covariance between feature vectors in different classes. Hyperplanes can then be found that optimally separate the projected, N-1-dimensional feature vectors into N different classes. This is the basis of Linear Discriminant Analysis, a classification technique discussed later in Section 3.G (Duda, Hart, and Stork 2001).

The Fisher criterion has pros and cons. Its main advantage is its simplicity and computational efficiency. One disadvantage of the Fisher criterion is that it, and the resulting Linear Discriminant Analysis algorithm for classification, works well only when the means of each class are far apart from each other, with respect to the covariance of each class on its own. In cases where the classes significantly overlap, other techniques may work better (Duda, Hart, and Stork 2001). Another disadvantage is that only N-1 features can be retained for each anomaly. If N, the number of classes, is small, then N-1 features may be too few to adequately summarize the characteristics of the anomaly. A third disadvantage to the Fisher criterion is that it is a supervised technique – a training set of class-labeled anomalies is required.

b. Unsupervised

Many techniques for feature extraction are *unsupervised*. A training set of class-labeled anomalies is *not* needed. Instead, these methods find a linear or non-linear combination of data that preserve the "structure" of the unlabeled data. The "structure" of the data can be defined either globally (considering all anomalies at once) or locally (considering each anomaly on its own, along with its most similar anomalies).

1) Global

Principal Component Analysis (PCA) is the most well-known technique for unsupervised, global feature extraction. It finds the linear combination of data that best preserves the variance in the data over all anomalies.[9] To accomplish this, the high-dimensional data collected from all anomalies are projected down onto a low-dimensional subspace. First, the data collected from each anomaly are grouped into a high-dimensional vector. The data vectors for all anomalies are gathered together into a high-dimensional data matrix. The data matrix is decomposed into its eigenvalues and eigenvectors. The K leading eigenvectors (the eigenvectors associated with the K largest eigenvalues) span the low-dimensional subspace onto which the original data are projected. The features extracted from each anomaly are the projected coordinates of the anomaly's data in this low-dimensional subspace.

PCA is attractive for its simplicity and computational efficiency. However, there are some drawbacks:

- PCA is a linear technique. Therefore, it cannot preserve the structure of any non-linear relationships between data (Bishop 1995; Duda, Hart, and Stork 2001). In such a case, PCA may over-estimate the inherent dimensionality of the data (Bishop 1995). Several non-linear techniques for feature extraction have been introduced to address this issue, as will be discussed below.
- PCA is an unsupervised technique. A training set of class-labeled anomalies is *not* used to find the optimal low-dimensional subspace onto which the data can be projected. Therefore, once the data has been projected onto the low-dimensional subspace, it may no longer contain the information needed for classification. This is a deficiency shared by all unsupervised techniques for feature extraction (Duda, Hart, and Stork 2001). However, depending on the application, some unsupervised techniques deal with this issue more effectively than others. For example, *Independent Component Analysis (ICA), a.k.a. Blind Source Separation,* is an unsupervised feature extraction technique that, under certain condi-

[9] Note that the data must first be normalized so that each datum spans a similar range of values. Otherwise, a datum with large values will overpower the eigenanalysis. Prior knowledge can be used to determine how the normalization should take place. For example, if it is known that a datum is Gaussian distributed, then one can normalize the datum for each anomaly by subtracting the mean (calculated over all anomalies) and dividing by the standard deviation (again, calculated over all anomalies.)

tions, can better preserve the classification ability of the data. While PCA maps the high-dimensional data onto its K directions of maximum variance, ICA maps the data to the K directions that are most *independent* of each other. If one can assume that the data were produced by one of K different, independent sources, then mapping the data onto the directions of maximum independence can preserve the ability to classify the data into the correct class (Duda, Hart, and Stork 2001).

- Finally, PCA is a not a generative method – it cannot be used to build a model of the probability distribution of the data. Fortunately, several extensions to PCA have been proposed. Sensible PCA (SPCA), Probabilistic PCA (PPCA), and Combinatorial Probabilistic PCA (CPPCA) are generative models that can be used to accomplish the same effect as PCA (Roweis and Ghahramani 1999). These techniques can be generalized even further, so that the observation noise explicit in the generative model is no longer constrained to be isotropic. Such a method is identical to *Factor Analysis*, a well-known technique for feature extraction in the statistics community (Roweis and Ghahramani 1999).

Non-linear extensions can be made to PCA (Bishop 1995; Duda, Hart, and Stork 2001). *Kernel PCA* uses the "kernel trick," the effect of which is similar to first mapping data onto an even higher-dimensional subspace, one in which linear combinations of the mapped data can be found that preserve the variance in the data. An *Autoencoder Neural Network*, one with at least five layers, can also be used to perform a non-linear version of PCA.

Other non-linear techniques can be used to perform feature extraction by preserving the global structure in the data (van der Maaten, Postma, and van den Herik 2009). As discussed above, PCA, Kernal PCA, and Autoencoder Neural Networks preserve the variance in the data, calculated over all anomalies. In contrast, *Multidimensional Scaling (MDS), a.k.a. Sammon Mapping*, preserves the Euclidean distances between the data collected for all pairs of anomalies. (Under some constraints, MDS is identical to PCA.) While MDS preserves the inter-anomaly Euclidean distances, *Isomap* preserves the inter-anomaly *geodesic* distances. The geodesic distance is measured along the low-dimensional manifold on which it is assumed that the high-dimensional data resides. Therefore Isomap can be useful in applications where it is known that the data resides on a curved or folded manifold. Other non-linear techniques also assume a manifold with specific characteristics (e.g., degree of smoothness, existence of holes, etc). Two examples are *maximum variance unfolding (MVU)* and *diffusion maps*.

2) Local

The feature extraction techniques discussed above attempt to preserve the structure of the data, as measured across all anomalies. Those are "global" techniques. "Local" techniques also attempt to preserve the structure of the data, as measured with respect to each anomaly and the other anomalies most similar to it (its "nearest neighbors").

Local linear embedding (LLE) is one of the most well-known techniques for local, unsupervised feature extraction (Roweis and Saul 2000). First, the data collected from each anomaly are grouped into a high-dimensional vector. Then, a graph is drawn to connect each anomaly to its K most similar anomalies, where "similar" is defined by the Euclidean distance between high-dimensional vectors. Next, each anomaly is represented as a linear combination of its K nearest neighbors. The weights of this linear combination are the features extracted for the anomaly. Although this technique is not linear in a global sense, it is indeed linear in a local sense, hence its name. The local linearity means that the weights can be solved using an optimization process that does not risk getting stuck in a local minimum (a risk shared by most other non-linear techniques). LLE is a very popular local feature extraction technique due to its simplicity and relative ease of computation. However, it has been shown that LLE does not always perform well when there are holes in the low-dimensional manifold in which the original data resides (van der Maaten, Postma, and van den Herik 2009). Other techniques can be used to perform local feature extraction, including *Laplacian Eigenmaps*, *Hessian LLE*, and *Local Tangent Space Analysis* (van der Maaten, Postma, and van den Herik 2009).

4. Visualization

Dimensionality reduction can be useful even after classification, as a post-processing step. This is particularly true for data mining, a technique for unsupervised classification discussed later in this document. In unsupervised classification schemes, the features extracted from anomalies are first organized into discrete clusters in the low-dimensional feature space. An expert human is then needed to determine which clusters are important, as well as which clusters correspond to which classes. Although the space in which the features are clustered has a lower dimensionality than the original data, its dimensionality is still often higher than two or three. Therefore, it can be difficult for a human to examine the clustered features, since it is difficult for humans to perceive information in more than two or three dimensions. Therefore, the dimensionality of the features can be reduced even further, *after* clustering has already been completed, *solely* to make it easier for a human to interpret. This process is referred to as *visualization* (Duda, Hart, and Stork 2001).

Often the best techniques for visualization are those that preserve the Euclidean distance between the clustered features (Duda, Hart, and Stork 2001; Bishop 1995). This is because humans can intuitively understand that features are similar to each other if they are separated by a short Euclidean distance. As discussed above, MDS is a global feature extraction technique that preserves the Euclidean distance across all anomalies. Therefore, MDS is often used for visualization. *Self-organizing maps (SOMs), a.k.a. Kohonen Maps,* are often used for visualization as well. SOMs can be implemented using a two-layer Artificial Neural Network. The neural network maps each feature vector onto squares of a two-dimensional grid. The grayscale value of a grid square is correlated to the number of feature vectors that have been mapped into it. Feature vectors that cluster together in feature space will map onto the same or nearby grid square. The

human expert can look at the grid to determine how many clusters there are, how tightly the clustering occurs, etc.

5. Current Status

Dimensionality reduction, either as a pre- or post-processing step to classification, has been acknowledged as an important part of machine learning for several decades. A flurry of activity has been seen in this area over the most recent decade however. This is due to the fact that in the past 10 years new sensors have been developed that can produce an extremely large amount of data for each sample. Hyperspectral imaging and gene sequencing microarrays are two examples. Data collected from these new sensors suffer from the Curse of Dimensionality. Too many samples would be required to learn the parameters of a classification algorithm operating on the original high-dimensional data. Furthermore, the iterative learning process would take too long to converge. Therefore, dimensionality reduction is more important now than ever. In the last 3 years at the Algorithms for Threat Detection Workshop (Chapter 5), more than 30 submitted papers have focused on dimensionality reduction. Although dimensionality reduction is not a new field, is it definitely back in vogue.

G. Classification

Classification is the process of assigning an observation to a category. In a binary classification problem, there are only two possible categories, such as "threat" vs. "clutter." Other classification problems may have several different possible categories, such as "threat A," "threat B," "threat Z."

Two steps must be accomplished before classification can even begin. First, the observation must be detected and determined to be of interest. In what follows, we will refer to these observations as *anomalies*. Next, representative features must be extracted[10] from the anomaly. Once these two steps are complete, classification algorithms can be constructed to assign the features extracted from the detected anomaly into a particular category. That is to say, the classification algorithm can perform a type of *feature fusion* – fusing the features together in order to output a category label.

Classification can take three different approaches:

1. *Supervised learning* uses a training set to learn the optimal mapping from features to categories.

[10] Throughout this section, we refer to "extracting" features from the detected anomaly. This can mean (1) using heuristics to take measurements of the collected data to use as features, (2) selecting a subset of the collected data to use as features, and/or (3) extracting linear or non-linear combinations of the collected data to use as features. All three of these methods were discussed in Section 3.F on dimensionality reduction.

2. *Unsupervised learning* does not use a training set. Instead, the features are organized into discrete clusters. Then, an expert human is needed to determine which clusters represent which categories. *Data mining* falls into this approach.
3. *Semi-supervised learning* borrows methods from both supervised and unsupervised learning.

This section describes the philosophy behind each of these three approaches. It briefly explains the most common techniques for implementing each approach and discusses the pros and cons of each technique. Finally, this section concludes with an explanation of metrics that can be used to assess the performance of a binary classification algorithm.

1. Supervised Learning

Supervised learning uses a *training set* to learn the optimal mapping from features to categories (Bishop 1995; Duda, Hart, and Stork 2001). The training set consists of a set of anomalies that have already been detected. Each anomaly in the training set comes with two pieces of information: *features* and a *category label*. The features have already been extracted from each anomaly. They can be organized into a column vector **x**, where each element of **x** represents one feature. The category label has also already been set for each anomaly, using ground truth. The category label can be represented by a scalar y, where the value of y represents the category. For example, y=1 can represent a true threat, while y=0 can represent true clutter. In supervised learning, a model is constructed to map the feature vectors **x** to category labels y. There are several approaches to building these models that will be discussed in this section. The two main methods are (1) *generative* and (2) *discriminant*.

a. Generative Methods

Generative methods are used to estimate the *joint probability distribution* P(**x**,y) of the feature vectors **x** and category labels y. Generative methods are often used for two purposes:

1. The joint probability distribution P(**x**,y) can be used to *generate* synthetic data that mimics the data collected from the field – this is how "generative methods" got its name. Synthetic data can be useful for modeling and simulation, as well as for training new users.
2. P(**x**,y) can also be used to classify an anomaly's feature vectors **x** into category *y*, the topic of this section.

Bayes' Rule can be used to formally state how generative methods classify an anomaly into a category:

$$\mathrm{P}(\mathrm{y}|\boldsymbol{x}) = \frac{\mathrm{P}(\mathbf{x},\mathrm{y})}{\mathrm{P}(\mathbf{x})} = \frac{\mathrm{P}(\mathbf{x}|\mathrm{y})\mathrm{P}(\mathrm{y})}{\mathrm{P}(\mathbf{x})} \tag{7}$$

First, the *posterior probability* P(y|**x**) is estimated for each possible category (each possible value of *y*). This can be done by dividing P(**x**,y) by the *evidence* P(**x**). P(**x**) is measured directly from the feature vectors; it does not vary based upon y and is therefore used only to ensure that the posterior probabilities P(y|**x**) range from zero to one. Once the posterior probabilities P(y|**x**) have been calculated for all possible categories, a decision rule must then be used to assign the anomaly into one particular category. Often the *maximum a posteriori (MAP)* rule is used, which assigns the anomaly into the category *y* with the maximum posterior probability P(y|**x**). For example, if P(y=*threat* | **x**) = 0.7 and P(y=*clutter* | **x**) = 0.3, then the anomaly is assigned to the category y=*threat*, since 0.7 > 0.3.

Bayes' Rule describes how the joint probability P(**x**,y), and therefore the posterior probabilities P(y|**x**), can be estimated from two other metrics. P(**x**|y) is the *likelihood* that the feature vector **x** is extracted from the anomaly, given that the anomaly belongs to category y. P(y) is the *prior probability* that the anomaly belongs to category *y*, set even before any feature vectors are extracted and examined. P(y), for each category y, can often be estimated based upon the prevalence of anomalies from each category in the training set, or in similar experiments that took place in the past. The training set can also be used to estimate the likelihoods P(**x**|y), for all possible categories *y*. Both *parametric* and *non-parametric* techniques can be used to estimate the likelihoods.

b. Parametric Techniques

Parametric techniques assume a particular distribution for the likelihoods P(**x**|y). Each distribution can then be fully characterized by a small number of parameters. For example, a Gaussian distribution can be fully characterized by its mean μ and variance σ^2. The training set can be used to learn the appropriate values for these parameters. Both Maximum Likelihood and Bayesian techniques can be used.

Maximum Likelihood techniques assume that each parameter is a deterministic, but unknown, value. This technique searches for the values of the parameters that would have most likely generated the feature vectors extracted from the training set anomalies in a particular category. For example, *linear discriminant analysis* is a Maximum Likelihood technique that builds a model in which the likelihoods P(**x**|y) for the different possible categories are pushed as far apart as possible in feature space. These likelihoods are used to define hyperplanes in feature space that separate, as well as possible, the training set feature vectors according to their ground truth category labels. A new previously unseen feature vector can later be assigned to a category based upon where it falls in feature space with respect to the hyperplanes. Note that linear discriminant analysis is actually a misnomer, as it is actually a generative technique, as opposed to a discriminant technique.

In contrast to Maximum Likelihood techniques, *Bayesian* techniques assume that each parameter is a random variable defined by a probability distribution function, which itself can be characterized by a set of parameters. Prior knowledge can be used to give a

rough estimate of these parameters, before any data are collected. These estimates can then be updated once feature vectors are extracted from the training set anomalies.

Maximum Likelihood and Bayesian techniques produce identical results when the training set is infinitely large. In real-world applications, however, the training set must be of limited size. One must then choose which technique should be used to estimate the likelihoods $P(\mathbf{x}|y)$. Maximum Likelihood techniques are often very popular because they are less computationally intensive and easier to interpret than Bayesian techniques. However, Bayesian techniques can be very attractive when there is sound prior knowledge about the parameters characterizing $P(\mathbf{x}|y)$ and $P(y)$.

Finally, *template matching*, also known as *library matching* or *"fingerprinting"* is an approximate technique for parametric, generative, supervised learning. Here, the training set is treated as a *library* of *templates*. A template is a feature vector extracted from a labeled anomaly. A feature vector extracted from a new previously unseen anomaly is compared to each template in turn. A similarity metric is used to determine how well the new feature vector "matches" each template – often, the Euclidean or Mahalanobis distance is used as the "match" metric. The new feature vector is then assigned to the same category as the template it matches best (i.e., the template that is the shortest distance away).

Regardless of which approach is used, parametric methods often require a smaller training set than non-parametric methods, since only a small number of parameters must be learned. That is one major advantage of parametric methods. The main disadvantage is that one must have prior knowledge about what type of distribution should be used for the likelihoods $P(\mathbf{x}|y)$, and the classification results may be sensitive to misspecifications.

c. Non-Parametric Methods

Non-parametric methods do not assume a particular distribution for the likelihoods $P(\mathbf{x}|y)$. Instead, the entire distribution is estimated from the training data. For example, *Parzen windows* is one technique for accomplishing this task. Gaussian distributions of equal variance are centered upon every feature vector in the training set (a rule is used to set that variance across all training set feature vectors). $P(\mathbf{x}|y)$ is then estimated as the sum of these Gaussians.

Non-parametric methods have both pros and cons. One large advantage is that they can be used in cases where the distribution type for the likelihoods $P(\mathbf{x}|y)$ is not known or is known to be very complex (multi-modal, highly skewed, etc). One disadvantage is that they typically require very large training sets to estimate the distributions.

d. Discriminant Methods

In contrast to generative methods, discriminant methods do not attempt to model the joint probability distribution $P(\mathbf{x},y)$ that generated the features and category labels. Instead, these methods seek to find a function that directly maps the feature vectors $\mathbf{x}$

onto the appropriate category labels y. The mapping functions are defined by parameters; the training set is used to estimate the optimum values of these parameters. The mapping functions can be simple or complex.

K nearest neighbors is one of the simplest mapping functions. A new previously unseen feature vector x is classified based on its K nearest neighbors, the K feature vectors in the training set that are "closest" to the new feature vector **x** in feature space. Often the Euclidean distance is the metric used to define "closest." A "majority vote" is then taken among the K nearest neighbors. That is, the new feature vector x is classified into the category to which a majority of its K nearest neighbors belong. Although K nearest neighbors is a very simple algorithm, it can be fairly robust when the training set is large. Therefore it is often used as a baseline against which all other classification algorithms are compared.

Artificial neural networks (ANN) implement complex (typically non-linear) mappings between feature vectors and category labels (Bishop 1995). ANNs were developed to model the behavior of biological neural networks found inside the body. An ANN is composed of multiple layers of nodes. The feature vector **x** is input into the first layer of the ANN. The output of every node in the i^{th} layer feeds into the input of every node in the $i+1^{st}$ layer. The node multiplies each input by a weight, sums the weighted inputs, applies a non-linear function to the sum, and then returns the output of the non-linear function. The training set can be used to set optimal values for the weights used by each node. For example, the back propagation algorithm uses gradient descent to minimize the mean squared error between the output of the ANN and the ground truth category labels in the training set.

ANNs have a bad reputation. This is fair in some cases and unfair in others. ANNs were introduced in the 1970s with much fanfare. In theory, an infinitely large ANN, trained on an infinitely large training set over an infinitely long time, can learn the mapping between any set of feature vectors x and any set of category labels y (Bishop 1995). In real-world applications, however, the size of ANNs, their training set, and the time over which they are trained must be finite. Therefore, the theory of ANNs did not pan out in real-world applications, and ANNs fell out of favor in the 1980s. ANNs experienced a resurgence in the 1990s, however, once the machine-learning community recognized their limitations and began applying them only to those applications for which they were well-suited (Tarassenko 1998): applications for which large training sets can be compiled and for which dimensionality reduction has already been performed (i.e., feature vectors have already been whittled down to a size that requires only a manageable number of nodes in the input layer of the ANN). One disadvantage to ANNs still remains, however: they are black boxes. That is to say, it is very difficult to understand why an ANN outputs a particular category label. The weights of the nodes give very little insight into how the classification decision was made.

Since the early 2000s, the most popular discriminant technique for supervised learning is the *support vector machine* (SVM) (Meyer, Leisch, and Hornik 2003). Like

linear discriminant analysis, an SVM finds a hyperplane that best separates the feature vectors in the training set according to their ground truth category labels. New previously unseen feature vectors are assigned to a category based on which side of the hyperplane they fall. Unlike linear discriminant analysis, an SVM does not attempt to first learn a generative model as means to learning the optimal hyperplane. Instead, SVMs learn the optimal hyperplane directly. This is done by first identifying the feature vectors in the training set that are on the edges of each category; these are called the support vectors. The hyperplane is then specified by maximizing its distance on one side to the support vectors from one category while maximizing its distance on the other side to the support vectors from the other category. If a hyperplane cannot be found (i.e., if the feature vectors cannot be linearly classified), then SVMs can use the "kernel trick" to perform non-linear classification. The "kernel trick" has the same effect as using a non-linear mapping function to project the feature vectors onto a higher-dimensional space in which a hyperplane can indeed be found (i.e., the feature vectors can indeed be linearly classified in the higher-dimensional space).

SVMs have become popular over the last 10 years because the specification of the optimum hyperplane depends only upon the support vectors – those few feature vectors on the edges of the categories – which allow smaller training sets to be used. Like ANNs, however, SVMs can be difficult to interpret, especially when the data are transformed into a high-dimensional space before linear classification.

2. Unsupervised Learning

Unsupervised learning does not use a training set (Bishop 1995; Duda, Hart, and Stork 2001). Instead, features are extracted from a set of new, previously unseen anomalies, and then any underlying structure in those features is looked for and exploited. Sometimes the underlying structure can be used for dimensionality reduction (i.e., whittling down the features to a smaller number that can be more easily managed in feature space). Other times, the structure can be used for *clustering* (i.e., assigning the features into one of N groups in feature space, even though one does not know in advance what each group represents). A human expert is then needed to determine which group cluster corresponds to which category – this is called *Data Mining*. Unsupervised learning for dimensionality reduction was discussed earlier in Section 3.F. Unsupervised learning for clustering, as a means to classification, is the topic of this section.

Clustering requires the estimation of the distribution of feature vectors in feature space, $P(\mathbf{x})$. Both parametric and non-parametric techniques can be used to estimate this distribution, as was done to estimate the likelihoods $P(\mathbf{x}|y)$ in supervised learning. Also, as in supervised learning, both Maximum Likelihood and Bayesian techniques can be used. Unlike supervised learning, estimating the distribution with unsupervised learning is more computationally intensive and a solution is less likely to converge. The contribution of each feature vector has less effect on learning the distribution, since one

does not know in advance to what category each feature vector belongs. Therefore, approximate solutions are often sought, rather than exact solutions.

K means clustering is one of the most common clustering techniques. It is an approximate solution for a parametric technique. First, it is assumed that there are K clusters of feature vectors in feature space. Each cluster is assumed to be well-modeled by a Gaussian distribution. Then, the means of these K Gaussian distributions are initialized to some values – often, random values are used. Next, each new, previously unseen feature vector is classified into one of the K clusters, based on which cluster's mean is the "closest." Often the Euclidean distance is the metric used to define "closest." Then, the means of the K clusters are updated, based on the feature vectors that were classified into them in the previous step. This process then iterates until a criterion is optimized – often this criterion is the mean squared error between each cluster mean and all feature vectors currently classified into that cluster. One seeks to minimize this mean squared error. Once the minimum mean squared error is found, the cluster means are held constant. All future feature vectors are then assigned to the cluster with the "closest" mean, based on the same metric that was used above. Of course, a human expert is then needed to determine which clusters correspond to which categories.

K means clustering has pros and cons. The main advantage is that it is simple and often works well in practical applications. One disadvantage is that is requires prior knowledge of K, the number of clusters. Another disadvantage is that the iterative process used to set the means of the K clusters can become stuck in local minima. One can mitigate this risk, in part, by making wiser choices about the initial values of the K cluster means. This often requires prior knowledge about the particular application.

Hierarchical clustering is another common clustering technique. With this technique, clusters can have sub-clusters, and sub-clusters can have sub-sub-clusters. *Agglomerative* techniques first assign each feature vector to one of N clusters. Then, clusters are merged together based on whether the feature vectors assigned to them are "similar" – often the Euclidean distance is the metric used to define "similar." The process iterates until a criterion is optimized. *Divisive* techniques are the opposite. All feature vectors are first assigned to one single cluster. Then, the cluster is split based on which feature vectors are "dissimilar." The process iterates until a criterion is optimized.

Hierarchical clustering also has both pros and cons. Its main advantage is that the sub-clusters and sub-sub-clusters give insight into the structure of the data – these different levels of clusters can be visualized as a *dendrogram*. Also, hierarchical clustering does not require prior knowledge about the number of clusters – sub-clusters can be agglomerated or divided until they "look about right" to a human expert. However, one large disadvantage to hierarchical clustering is that it can be computationally intensive to perform. Also, the overall performance of the technique depends heavily on whether the first level of agglomeration/division is "correct"; a mistake in the first level can lead to catastrophic consequences further up/down the hierarchy.

Finally, *anomaly detection* can also be thought of as a type of unsupervised learning. Although some forms of anomaly detection are based on a model of what is considered "normal" (a.k.a. background) clutter, there is no model for threat. Therefore, one simply looks for a departure from "normal clutter," without any further information about whether this anomaly is a threat. A higher form of intelligence – either a human expert or a supervised learning classification algorithm – can then be used to assign the detected anomaly into a labeled category.

Unsupervised learning for clustering/classification has advantages over supervised learning. The main advantage is that a training set does not have to be compiled. This can significantly cut down on time and cost. Resources do not have to be spent collecting data and extracting features from any training set anomalies. What is even more important, resources do not have to be spent acquiring the ground truth needed to assign category labels to any training set anomalies; collecting this ground truth is often one of the most expensive parts of establishing a training set for supervised learning.

Nothing is free however. With unsupervised learning, resources must be spent on displaying to a human expert the features that are extracted from the anomalies and organized into clusters. (This was discussed in Section 3.F for dimensionality reduction for visualization.) Only then can the human expert use his or her training, experience, and intuition to determine which clusters are important, as well as which clusters represent which categories. That is to say, the human expert is the source of ground truth in unsupervised learning. This ground truth comes into play at the end of the process, once the clusters have been formed, rather than at the beginning of the process, as is done in supervised learning when the training set is compiled. Still, unsupervised learning can be helpful in the following situations: (1) when one wants to first organize the data in a meaningful way before presenting it to a human expert, (2) when the category labels change over time more quickly than a training set can be compiled for supervised learning, and (3) when one wants to find out, first and foremost, whether there is in fact any underlying structure to the data (i.e., whether the data even cluster at all). This is all part of *hypothesis generation*.

A disadvantage to unsupervised learning is that there must be an underlying structure in the data. For example, features extracted from a true threat anomaly must occupy a different part of feature space than features extracted from a true clutter anomaly. If the wrong features are extracted, or if the signal-to-noise ratio is low (the threat signal vs. the background clutter noise), then the features may not cluster in feature space, and even the most expert of humans will not be able to understand which area of feature space represents which categories.

3. Semi-Supervised Learning

Semi-supervised learning borrows from both supervised and unsupervised learning (Zhu and Goldberg 2009). As discussed earlier in this section, supervised learning uses a training set of labeled data (feature vectors with ground truth category labels) in order to

learn how to classify new feature vectors into categories. Although supervised learning techniques can be computationally efficient, assembling the training set can be expensive and time-consuming. Unsupervised learning does not use a training set. Instead, unlabeled feature vectors are assigned to clusters. A human expert is then needed to determine which clusters represent which categories. Unsupervised learning can be attractive in situations where resources cannot be spent up-front in assembling a training set. However, techniques for unsupervised learning can be computationally intensive and it can be difficult to converge to a correct solution. Semi-supervised learning seeks the best of both worlds.

Semi-supervised learning operates on both labeled and unlabeled feature vectors. One can approach semi-supervised learning from two different perspectives:

1. One can use a small training set of labeled feature vectors to learn the parameters of a classifier using supervised learning techniques (either generative or discriminative, parametric or non-parametric). The parameters of the classifier are subject to constraints imposed by the unlabeled feature vectors.

2. One can use the unlabeled feature vectors to learn the parameters of a clustering algorithm using unsupervised learning techniques. Then, one can sample a subset of the feature vectors, such as those on the edges of clusters, and collect ground truth category labels for them. These labels can then be used to determine which clusters represent which categories. This second approach is often termed *Active Learning* (Zhang, Liao, and Carin 2004).

There are several different techniques for semi-supervised learning (Zhu and Goldberg 2009). The "correct" technique depends on the particular application. For example, *Expectation-Maximization with Generative Mixture Models* can be an excellent technique when there is prior knowledge that the feature vectors form well-defined clusters in feature space. *Transductive Support Vector Machines* is a discriminant technique that is a natural extension of more traditional SVM techniques. If a related application already uses a traditional SVM, it can make sense to use a transductive SVM for the new application. Transductive SVMs are based on the assumption that the decision boundary – the hyperplane in higher-dimensional space – does not exist in a region of high density. Thus transductive SVMs should only be used in applications for which there is prior knowledge that the categories overlap very little. Regardless of the technique used, one must make sure that the assumptions underlying the chosen technique do indeed hold for the particular application in mind. Otherwise, the unlabeled data are likely to degrade performance, rather than improve it.

4. Metrics for Performance Assessment

Regardless of what method is used for classification, suitable metrics must be chosen to assess the performance of the classification algorithm. The confusion matrix in Figure 3-5 illustrates the binary classification problem. The algorithm classifies each detected anomaly as either "threat" or "clutter" (rows). To score the algorithm's

classifications, ground truth can be used to label each detected anomaly as true threat or true clutter (columns). Then, each detected anomaly can be counted as a True Positive (TP), False Positive (FP), False Negative (FN), or True Negative (TN). These four counts sum to the total number of detected anomalies. The counts are often recast into summary metrics, which are easily interpretable fractions ranging from zero (poor performance) to one (good performance).

		Truth		
		Threat	Clutter	
Algorithm	Threat	*TP*	*FP*	Positive Predictive Value = $\frac{TP}{TP+FP}$
	Clutter	*FN*	*TN*	Negative Predictive Value = $\frac{TN}{TN+FN}$
		Sensitivity = $\frac{TP}{TP+FN}$	Specificity = $\frac{TN}{TN+FP}$	Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Figure 3-5. A Binary Confusion Matrix

Columns refer to ground truth and rows refer to algorithm output. Counts of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) are tallied and summarized in metrics. Black metrics apply to both detection and classification problems, while gray metrics apply to classification only.

Accuracy is the most common summary metric. It measures the number of detected anomalies that were correctly classified. Accuracy does not tell the whole story, however, as it lumps together both threats and clutter. Therefore other metrics must also be used.

Sensitivity and specificity are the next most common metrics. Sensitivity is often referred to as *True Positive Rate*, *Recall*, or *Probability of Detection* (Pd). (Pd is a misnomer because it is *classification* performance, not detection performance, which is being assessed.) All of these terms refer to the same quantity: the fraction of true threats that were correctly classified as threats. Specificity is the inverse: the fraction of true clutter that was correctly classified as clutter. Specificity is also often referred to by different names, such as *True Negative Rate* or *Inverse Recall*. It can also be calculated as one minus the False Positive Rate or one minus the Probability of False Alarm (Pfa). Regardless of the names that are used, sensitivity and specificity are calculated along the ground truth columns of the confusion matrix. They reflect the "scientist's perspective" – how much of the truth was correctly classified?

Positive and Negative Predictive Value (PPV and NPV) are two other metrics. They reflect the "user's perspective" – how many of the classifications were true? PPV measures the fraction of threat classifications that were truly threats; NPV measures the

fraction of clutter classifications that were truly clutter. PPV has many other names, such as *True Discovery Rate* or *Precision*. It can also be calculated as one minus the False Discovery Rate. NPV is sometimes called *Inverse Precision*.

Unlike sensitivity and specificity, PPV and NPV are affected by the *prevalence* of threats in the collected data. Prevalence is defined by ground truth. In classification problems, it is calculated as the fraction of detected anomalies that were true threats:

$$Prevalence = \frac{TP+FN}{TP+FN+FP+FN} \tag{8}$$

Figure 3-6 shows three different confusion matrices, each giving simulated numbers for the performance of the same classification algorithm applied to the same type of data. The only difference is the prevalence of the threat in the data: low, moderate, and high. This difference in prevalence causes differences in PPV and NPV: as prevalence increases, PPV increases and NPV decreases. Sensitivity and specificity are left unchanged. *Thus the "user's perspective" of an algorithm will differ based upon the prevalence of threat.* If the threat is rare (low prevalence), then it will be difficult for the algorithm to fulfill the needs of the user. That is, even though the algorithm may be able to correctly classify a large fraction of both true threat and true clutter (leading to high sensitivity and specificity), it may appear to the user as though the algorithm is performing poorly, since most threat classifications will be incorrect (leading to low PPV). In situations like this, the user may tire of the algorithm and simply turn it off. This is particularly true when false positives bear a high cost.

Four metrics are needed to fully assess the performance of a binary classification algorithm. However, often only sensitivity and specificity are provided to encapsulate the "scientist's perspective." As a result, the "user's perspective" is ignored. PPV and NPV can be calculated from sensitivity and specificity, provided that the prevalence is known as well as the total number of detected anomalies (the sample size). Preferably, though, the four counts of TP, FN, FP, and TN are provided in a confusion matrix, from which all summary metrics are directly calculated.

(a)

		Truth: Threat	Truth: Clutter	
Algorithm	Threat	*TP = 99*	*FP = 50*	Positive Predictive Value = $\frac{99}{99+50} = 0.66$
	Clutter	*FN = 1*	*TN = 950*	Negative Predictive Value = $\frac{950}{950+1} \approx 1.00$
		Sensitivity = $\frac{99}{99+1} = 0.99$	Specificity = $\frac{950}{950+50} = 0.95$	Accuracy = $\frac{99+950}{99+950+50+1} = 0.95$

Low Prevalence = $\frac{99+1}{99+1+50+950} = 0.09$

(b)

		Truth: Threat	Truth: Clutter	
Algorithm	Threat	*TP = 99*	*FP = 5*	Positive Predictive Value = $\frac{99}{99+5} = 0.95$
	Clutter	*FN = 1*	*TN = 95*	Negative Predictive Value = $\frac{95}{95+1} = 0.99$
		Sensitivity = $\frac{99}{99+1} = 0.99$	Specificity = $\frac{95}{95+5} = 0.95$	Accuracy = $\frac{99+95}{99+95+5+1} = 0.97$

Moderate Prevalence = $\frac{99+1}{99+1+5+95} = 0.50$

(c)

		Truth: Threat	Truth: Clutter	
Algorithm	Threat	*TP = 990*	*FP = 5*	Positive Predictive Value = $\frac{990}{990+5} = 0.99$
	Clutter	*FN = 10*	*TN = 95*	Negative Predictive Value = $\frac{95}{95+10} = 0.90$
		Sensitivity = $\frac{990}{990+10} = 0.99$	Specificity = $\frac{95}{95+5} = 0.95$	Accuracy = $\frac{990+95}{990+95+5+10} = 0.99$

High Prevalence = $\frac{990+10}{990+10+5+95} = 0.91$

Figure 3-6. Simulated Performance of the Same Classification Algorithm Applied to the Same Type of Data but with (a) Low, (b) Moderate, and (c) High Prevalence of Threat

As prevalence increases, PPV increases and NPV decreases. Sensitivity and specificity remain unchanged.

It should be noted that only two of these four metrics can be calculated for detection algorithms, such as those that perform anomaly detection. This is due to the fact that it does not make sense to count TNs for a detection problem. The purpose of a classification algorithm is to determine in which class an anomaly belongs. In contrast,

the purpose of a detection algorithm is to identify those anomalies in the first place. Thus, a classification algorithm will provide an output for each detected anomaly, but a detection algorithm will provide an output (sound an alarm, flash a light) *only* when an anomaly is identified. One can certainly count the number of times a detection algorithm sounds an alarm when a true threat is present (TP). One can also count the number of times an alarm is sounded when a true threat is not present (FP), as well as the number of times a true threat is present but an alarm is not sounded (FN). However, it does not make sense to count the number of times a true threat is not present and an alarm is not sounded (TN). As a result, one cannot calculate any metric that is defined in terms of TN, such as specificity, NPV, or accuracy. Instead, only sensitivity and PPV can be calculated, since these metrics are not based on TN. Additional metrics are often calculated for detection algorithms; these metrics involve the passage of time, such as the number of FPs per hour. This metric is often referred to as *False Positive Rate* or *False Alarm Rate*. One must be careful that this metric is not confused with the quantity $\frac{FP}{TN+FP}$, otherwise known as one minus specificity.

The performance of both classification and detection systems can be illustrated in graphical form (See Figure 3-7 for examples.). Receiver-Operating Characteristic (ROC) and Precision-Recall (PR) curves are two of the most common methods (Davis and Goadrich 2006):

- Traditionally, ROC curves plot Pd (i.e., Sensitivity or Recall) versus Pfa (i.e., one minus specificity) for each possible set of parameters on which the classification algorithm is based. In detection problems, where TNs cannot be counted and therefore Pfa cannot be calculated, the horizontal axis is often replaced with the False Alarm Rate (the number of FPs per unit of time). A "perfect" ROC curve will shoot straight up from the origin, touch the upper left corner of the plot, and then shoot straight across to the upper right corner of the plot. This indicates a perfect ability to classify threat vs. clutter. ROC curves are rarely perfect in real-world applications however. The shape of a "good" ROC curve depends on the particular application at hand. The "Area Under the Curve (AUC)" is often used to summarize the shape of a ROC curve, since a "perfect" ROC curve (plotting Pd vs. Pfa) will exhibit an AUC of 1, but a coin flip will exhibit a ROC curve that follows the chance diagonal and has an AUC of 0.5. One must bear in mind that the AUC is useful as a summary metric only when the cost of FNs is similar to the cost of FPs. When FNs and FPs bear very different costs, then the shape of the entire ROC curve must be taken into consideration.

- Precision-Recall (PR) curves plot Precision (a.k.a. Positive Predictive Value) versus Recall (a.k.a. Pd or Sensitivity) for each possible set of algorithm parameters. PR curves can be drawn for both classification and detection problems, as neither Precision nor Recall require a TN count. Unlike ROC curves, a "perfect" PR curve will start in the upper left corner, shoot straight across to the upper right corner, and then shoot straight down to the lower left corner. One must

bear in mind that PR curves are heavily influenced by the prevalence of the threat in the collected data, since Precision is heavily influenced by prevalence.

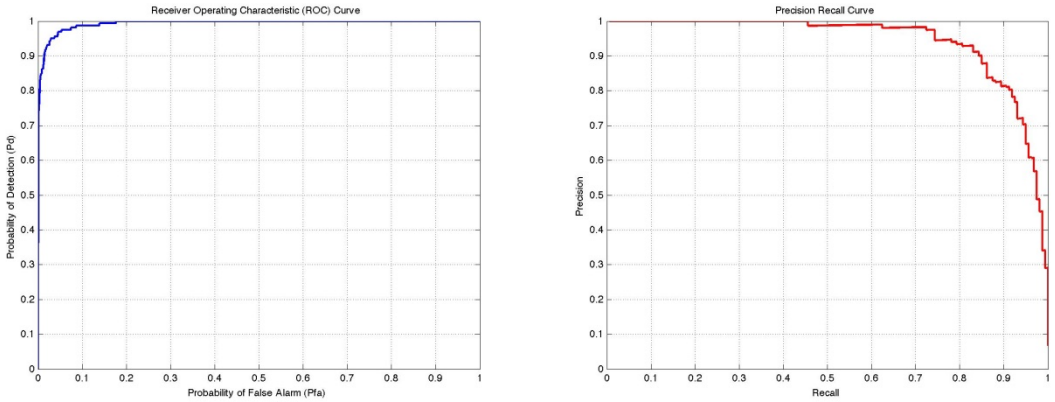


Figure 3-7. Sample ROC (left) and PR Curves (right)

The curves are derived from a real example with a "moderately good" classifier. Approximately 2,300 samples were classified with a prevalence of about 7 percent.

H. Decision Fusion

Decision fusion is the process of combining sensor information at the decision level. In other words, the information being fused is a decision each sensor has made about the presence or absence of a signal. The term "decision" may be slightly confusing, since it often implies an action as the result of a decision. "Decision" in our context might be better expressed as a declaration or an identification. More formally, it is the result of a hypothesis test that each sensor performs individually. We discuss both *hard* and *soft* decision fusion (Waltz and Waltz 2009). *Hard decision fusion* results in a single, optimum choice; *soft decision fusion* accounts for decision uncertainty in each sensor chain by maintaining a composite measure of uncertainty.

There are several reasons why one might fuse sensor data at the decision level. In some cases, the communications links between sensors or between sensors and the central processing facility may not provide sufficient bandwidth to transmit lower-level data. In these cases, each sensor may only transmit decision information, perhaps with additional bits specifying a quantitative measure of the quality of the decision. In other cases, the lower-level information may not be available. This is true for some off-the-shelf "black box" sensors whose internals are not accessible to the customer. This may be the case when decision information is passed between authorities or agencies; it may also happen when there are security classification constraints. One can easily envision cases in which the raw or feature data are at a higher-level of classification than the resulting decision based on these data, and the communications network only supports the lower level of classification.

For our canonical example of decision fusion, let us take the following. Assume one would like to determine if a radiation source is present in some environment of

interest. Further assume that several sensors are trained on this environment and individually make determinations of whether a radiation source is present. The sensors may all be of different designs or made by different manufacturers – what is common among them is that we have access only to their determination of whether a source is present. For the purposes of later discussion, let us label the hypothesis that no source is present as the null hypothesis, H_0, and the hypothesis that a source is present as the alternative hypothesis, H_1.

In practice, decision fusion tends to be done in an ad hoc manner. Typically some measure(s) of performance of the fused decision (e.g., detection rate and false positive rate) are optimized with respect to the manner in which sensor decisions are combined. This approach is amenable to scenarios in which one has access to ground truth data, or can create high-fidelity synthetic ground truth data from a model, either of which can be used to exercise the decision fusion algorithm. A common ad hoc decision fusion algorithm is voting. In our canonical example above, we treat the sensor's decisions as votes and take the majority decision as the fused decision. Refinements of voting are possible, of course. We may learn that some sensors are less accurate than others and weight their votes less heavily than more accurate sensors. We may also tune the number of votes required to reject the null hypothesis.

1. Pure Decision Fusion

As one might expect, the problem of *optimally* fusing hypothesis test decisions from individual sensors has been treated in the research literature. By adopting a likelihood ratio approach, Chair and Varshney (1986) showed that for n sensors, each with false positive probability P_{Fi} and miss probability P_{Mi}, and each sending u_i=-1 if deciding H_0 and u_i=1 if deciding H_1, the optimal decision fusion rule is given by the simple expression

$$f(u_1,\ldots,u_n) = \begin{cases} 1 & \text{if } a_0 + \sum_{i=1}^{n} a_i u_i > 0\,, \\ -1 & \text{otherwise,} \end{cases} \tag{9}$$

where the optimum coefficients are given by

$$\begin{aligned} a_0 &= \log \frac{P_1}{P_0} \\ a_i &= \log \frac{1-P_{M_i}}{P_{F_i}} \quad \text{if } u_i = 1 \\ a_i &= \log \frac{1-P_{F_i}}{P_{M_i}} \quad \text{if } u_i = -1 \end{aligned} \tag{10}$$

with P_0 the prior probability of H_0, and the P_1 the prior probability of H_1. The advantage of this fusion rule is that it is simple, can be performed with minimal computational power, and could therefore be implemented on very small hardware. However, it

requires knowledge of the performance characteristics of each sensor (false positive and miss rates). In addition, it is not clear if this fusion rule represents an improvement in performance over each sensor's individual performance.

Thomopoulos, Viswanathan, and Bougoulias (1987) looked at the decision fusion problem more generally, including the case in which the prior probabilities of H_0 and H_1 are not known. In particular, they considered optimality of the decision fusion algorithm in the Neyman-Pearson[11] sense and looked at the performance of the fusion relative to the performance of each sensor. They showed that a Neyman-Pearson (N-P) optimal decision fusion scheme can perform better than the best performing sensor for a system of three or more sensors. If each sensor additionally sends quality information about its decision, Thomopoulos, Viswanathan, and Bougoulias (1987) showed that the performance of the decision fusion algorithm can be comparable to that of a centralized N-P test – i.e., a single N-P hypothesis test performed with the raw data from *all* sensors. Quality information might be, for example, a simple flag that specifies whether the sensor has confidence in the result of its hypothesis test.

Although the N-P optimal decision fusion algorithm described above is fairly straightforward, this class of algorithm is still used as the basis for some decision fusion techniques today, including, for example, those that fuse human sensor[12] and physical sensor data (Liu, Chu, and Tsai 2012). However, when we go beyond the simple binary hypothesis test, say to consider sensors that sense dependent variables or when we allow sensors to send arbitrary amounts of data in addition to decision information, we enter a vastly more complex sensor fusion landscape. In fact, in this landscape, the line between decision fusion and feature fusion is blurry, in part because real sensor fusion systems confront a reality that is more complex than any simple taxonomy, and in part because many of the algorithms work for both feature and decision fusion.

2. Hybrid Decision Fusion

In a widely cited paper, Tsitsiklis (1993) treated the decision fusion problem more generally than Thomopoulos, Viswanathan, and Bougoulias (1987), relaxing some of the assumptions above. In particular, he provided a Bayesian formulation of the problem, allowing for *M*-ary hypotheses (as opposed to binary hypotheses in the N-P case). He also allowed the decision rule used by each sensor to be a (dependent or independent) random variable, considered sensor configurations more complex than a simple star topology (central node with all others nodes connected), and allowed sensors to send

---

[11] The Neyman-Pearson lemma prescribes how to construct the most powerful likelihood ratio test between two point hypotheses for a given significance. Here we use the terms power and significance in the formal sense. Power is the probability a test will reject the null hypothesis when the null hypothesis is false. Significance is the probability a test will accept the null hypothesis when the null hypothesis is true. See, e.g., http://en.wikipedia.org/wiki/Neyman-Pearson_lemma.

[12] A human sensor is a person armed with one or more smart mobile devices and social networking services.

messages from a finite alphabet. Although team decision problems (of which the Tsitsiklis (1993) formulation is a special case) are in general very difficult to solve computationally, he showed how to construct the optimal fusion rule under certain simplifying assumptions. He also considered the limit where the number of sensors approaches infinity and showed that parallel configurations (those in which information travels in parallel from each sensor to the fusion center) are in general superior to serial configurations (those in which information travels along a line of sensors to the fusion center). For a comprehensive treatment, see Viswanathan and Varshney (1997).

Arbitrary sensor configurations lead naturally to the use of belief networks, and graphical tools generally, from the field of belief propagation. Most prominent among these tools are Bayesian networks and Markov random fields. Bayesian networks are a means of graphically encoding the independence assumptions among variables in a joint probability distribution. Probability operations such as marginalization and conditioning are then simple graph operations, and inference is computationally tractable.[13] One of the features of Bayes networks is that the graph produced is always a directed acyclic graph (DAG). To represent cyclic dependencies (also known as "loopy" belief propagation), one may resort to Markov random fields, which allow for arbitrary connections between nodes. However, dependencies are non-directional in Markov random fields. Thus, a Markov network can represent certain dependencies that a Bayesian network cannot (such as cyclic dependencies); on the other hand, it cannot represent certain dependencies that a Bayesian network can (such as induced dependencies). A Markov construction might be applied when learning takes place, and this causes the system to change how the detectors are used or how detector data and or features are combined.

It should be noted that the nodes in the graphs used in graphical methods represent *different* random variables, all of which are part of some joint probability distribution. This is in contrast to the decision fusion techniques described above, where all sensors are deciding among a common set of hypotheses. As such, graphical methods might more accurately fit under the heading of "feature fusion" in our taxonomy. However, they are sometimes referred to as decision fusion techniques in the literature.

One of the major benefits of graphical methods is that they typically do not require knowledge of (or assumptions about) underlying distributions. This feature is especially useful for fields like computer vision and artificial intelligence, where underlying distributions are highly non-Gaussian. (Consider the eye, which has a strongly bimodal distribution – open or closed.) In a well-known paper, Sudderth et al. (2002) presents a novel nonparametric belief propagation algorithm and demonstrates its superior performance in estimating occluded features (for example, a prediction of whether a person is smiling based only on imagery of their eyes and nose).

---

[13] See http://en.wikipedia.org/wiki/Bayesian_network for the standard "wet grass" Bayes network example.

Nonparametric fusion methods are strongly data-driven. As such they are sometimes the basis for machine learning techniques.

3. Non-Probabilistic Methods

It is widely held that a probabilistic approach to decision fusion (examples of which we have just described) is the most fruitful one (Carl 2001). However, alternative approaches warrant at least brief mention here. The two most useful are fuzzy logic (or possibility theory) and Dempster-Shafer theory (or belief theory).

Fuzzy logic, as the name implies, extends Boolean set theory and logic, allowing truth to take on values between 0 and 1. It does this through the use of membership functions, which define a set of classes (e.g., "hot," "warm," "cold") as a function of a random variable. This is in contrast to probabilistic methods, which specify quantitative knowledge about the distribution of a random variable (e.g., mean and standard deviation of the temperature). Fuzzy logic differs from probabilistic logic primarily in conceptual interpretation. This difference, although it may seem minor, often allows for a more natural specification of actions to take in response to inputs in the case of vagueness in input data. For further reading, see Stover, Hall, and Gibson (1996), who describe a generalized fuzzy logic decision fusion architecture with an emphasis on DoD applications, and Singpurwalla and Booker (2004), who explicitly draw links between probabilistic and fuzzy specifications.

Dempster-Shafer (D-S) theory is a generalization of Bayesian probability. The most striking feature of D-S theory is that belief in a proposition is represented as an interval, bounded on the low side by the "belief" and on the high side by the "plausibility." The power of D-S theory lies in its "rule of combination" (a generalization of Bayes' rule), which prescribes how to combine belief constraints. Although D-S theory can yield counter-intuitive results (including some that contradict probability theory), the framework is quite natural for belief propagation, and so it sometimes finds use in decision fusion problems. For further reading, see Wu et al. (2002) and Koks and Challa (2003).

I. Sensor Management

Sensor management is the determination of an optimal sensor configuration at each time, within constraints, as a function of information available from prior measurements and possibly other sources (Hero and Cochran 2011; Hero et al. 2008). Many parameters of the sensor configuration may be controllable (for example, location, bandwidth, or sampling rate). The selection of the optimal configuration depends on the selection of an appropriate metric of performance.

Figure 3-8 depicts the basic elements of a sensor management system. Within the figure, the blue text below each box gives examples of each element. A sensor (S1, S2, S3) is selected, and a measurement is made. Given the measurement, information relevant to the sensing objective is distilled from the raw sensor data (typically using

various fusion algorithms). This processing will typically produce both information for the sensing objective (estimates, tracks, decisions) and information about the merit/relative importance of each potential observation in the next time period. Using this information, the sensor manager makes an (optimal) decision about which sensor measurement to acquire in the next time period.

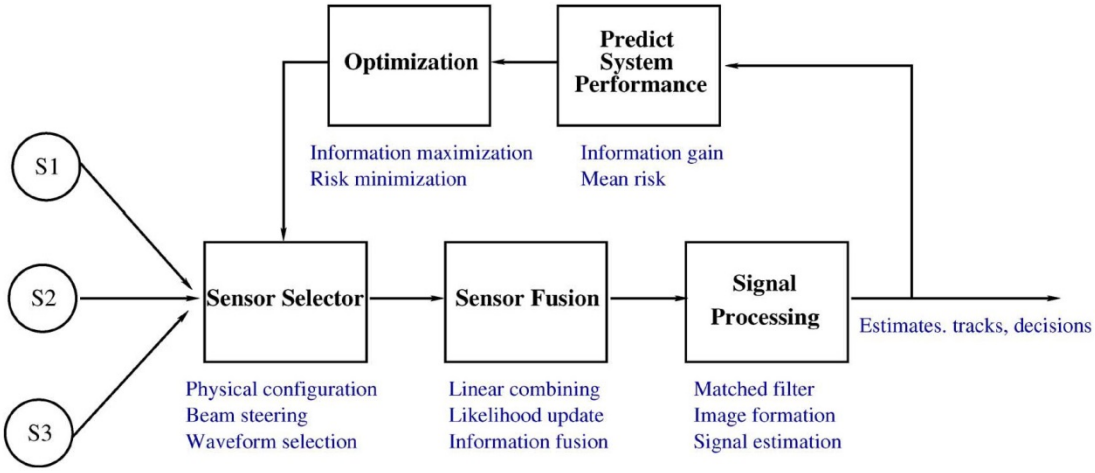


Figure from Hero and Cochran (2011).

Figure 3-8. Conceptual Block Diagram of a Sensor Management System

Given this framework, it is perhaps not surprising that the unifying research framework for sensor management is the theory of *decision processes*. From Hero and Cochran (2011), a decision process is a "time sequence of measurements and control actions in which each action in the sequence is followed by a measurement acquired as a result of the previous action" (p. 3069). From this perspective, designing a sensor manager is equivalent to specifying a decision rule that generates realizations of the decision process that, on average, maximize an expected reward.

Sensor management algorithms fall into three categories. Each makes specific simplifying assumptions about the nature of the decision process. The first category is the theory of Markov decision processes (MDP) and partially observable Markov decision processes (POMDP). These methods make the assumption that the current state of the system given the entire history of observations and actions depends only on the state and action in the previous time period. (The state of the system describes the current environment. The reward function to be maximized depends on the system state and the actions.) MDPs and POMDPs can be solved in general as a backward induction problem using Bellman's equation (Hero et al. 2008). For certain special cases (e.g., the linear Gaussian models used in the Kalman filter), they can be solved quickly, and the solution to the restricted (Markov assumption) problem is equal to the overall optimal solution.

Multi-armed bandit (MAB) problems are another approach to sequential resource allocation. The MAB takes its name from the idea of pulling the "arms" of multiple slot-machines ("bandits") to achieve rewards. From Scott (2010):

> A multi-armed bandit is a sequential experiment with the goal of achieving the largest possible reward from a payoff distribution with unknown parameters. At each stage, the experimenter must decide which arm of the experiment to observe next. The choice involves a fundamental trade-off between the utility gain from exploiting arms that appear to be doing well (based on limited sample information) vs. exploring arms that might potentially be optimal, but which appear to be inferior because of sampling variability. The trade-off has also been referred to as "earn vs. learn." (p. 639)

MABs have been used for search problems for several decades but have only recently been applied to sensor management. In certain special cases, they offer significant computational simplifications over the MDP and POMDP formulations.

Both MDP/POMDP and MAB approaches involve searching over multi-stage, look-ahead policies, which requires evaluating each available action in terms of its effect on the potential rewards for all future actions. Myopic sensor management approaches only look ahead to the next stage. These "greedy" policies are computationally simpler, but lose the guarantee of optimality. However, in some special cases, myopic approaches are close to optimal.

The most obvious way to develop a myopic policy is to consider the effect of the control action only on the immediate reward (the one-step, look-ahead policy). Empirically, one-step, look-ahead approaches that maximize rewards based on information gain, rather than a task-specific reward, often achieve better overall performance. This approach has a long history in sensor management and has been developed extensively as *optimal experimental design* (Lu, Anderson-Cook, and Robinson 2011; Myers, Montgomery, and Anderson-Cook 2009). Recent work has focused on making choices when faced with competing objectives.

As an example, consider the real-time allocation of sensors for numerical weather prediction. These approaches use ensemble Kalman filters with a metric of minimizing prediction error and build on the data assimilation techniques described in Section 3.D. The data assimilation techniques make it possible to determine from where in space and time the uncertainty in an analysis originates. This allows one to determine the optimal allocation of sensors. For example, if there is a critical spatial location where one needs to know the wind speed with more accuracy than the rest of the field, it is possible to determine the optimal placement of sensors to reduce the uncertainty in the analysis at that location. For forecasting, it is possible to determine where to best place sensors so that uncertainty in the forecast is reduced at a specific location at a specific *future* time.

Techniques to optimize sensor allocation are fairly advanced in the field of numerical weather prediction, where they fall under the heading of *targeted observing* or *adaptive observing*. In practice, they may involve the selection of optimal sensor locations before observing begins, the repositioning of sensors during observing, the addition of sensors to specific locations during observing, or the removal of sensors with adverse error properties. Several field trials have demonstrated the effectiveness of targeted observing [e.g., Toth et al. (2011)]. The Fronts and Atlantic Storm-Track Experiments (FASTEX), which studied the life cycle of cyclonic waves, was the first to demonstrate targeted observing in 1997. The North-Pacific Experiment (NORPEX) tested and compared two different targeted observing strategies in 1998. Later in 1998, the California Land-falling Jets Experiment (CALJET) demonstrated targeted observing for the first time on mesoscale events with 12- to 24-hour forecast windows.

In 1999 the Winter Storm Reconnaissance program (WSR99) was the first experiment in which targeted observation locations were chosen *in real time*. WSR99 demonstrated that targeted observing could improve the accuracy of real-time operational forecasts for significant weather events (Szunyogh et al. 2000). In 2003, the Atlantic THORPEX Regional Campaign further demonstrated the use of real-time control of a wide variety of sensor platforms for improving short-range forecasts (Truscott 2004).

Liu and Kalnay (2007) applied targeted observing to the problem of determining when sensors with limited duty cycles should be turned on and where they should be looking. An example sensor is a satellite with a limited energy budget, which may have power to operate, say, only 10 percent of the time. Liu and Kalnay (2007) showed that, for their particular problem, the addition of 10-percent duty cycle Doppler wind lidar (DWL) sensors to the standard weather observing network could reduce the forecast root mean square (RMS) error by a factor of about 3. This is to be compared with the same sensors at a 100-percent duty cycle, which would reduce the RMS error by a factor of about 6.

Kalnay, Ota, et al. (2012) and Kalnay, Yang, et al. (2012) present a simplified framework for what they call forecast sensitivity to observations (FSO). With FSO, one can determine which observations contribute the most uncertainty in the forecast and perform "proactive quality control" by removing those observations from the data assimilation cycle. They demonstrate a slight improvement in the forecast track of typhoon Sinlaku with FSO.

J. Methods, Algorithms, and Operational Missions

In the previous sections, we introduced a number of methods and algorithms. Many of these cut across the operational missions. For example, data fusion methods will be useful whenever multiple sensors are employed. Data triage and compressive sensing methods are helpful whenever the volume of data exceeds the local storage and/or processing capabilities. Other algorithms are relatively more useful for some of the

operational missions. These include anomaly detection, data assimilation, sensor management, and classification.

A *detection algorithm* sounds (or otherwise flags) an alarm if the setting contains a potential threat. Detection is often thought of as a screening tool with low specificity and high sensitivity: for example, a detection algorithm might have high negative predictive value (if it does not alarm, no threat is present), although it may be subject to false positives. However, different operational missions may require working at different locations on the ROC curve. Note that some detector hardware does not actually provide detection as defined here, but instead provides classification.

Data assimilation algorithms are used to reconstruct properties of the environment of interest and to predict the future evolution of the environment of interest. *Resource tasking* and *sensor allocation* algorithms rely on reconstruction of the environment of interest to determine where the next piece of data should be collected.

Classification algorithms are used to estimate the membership of the potential threat in a class (e.g., "threat" vs. "not a threat," or "threat A" vs. "threat B" vs. "threat C"). Clustering can be thought of as a type of classification in which the number and labels of classes are not known in advance. In addition, the dimension-reduction algorithms discussed in Section 3.F can be thought of as data preprocessing before applying classification techniques.

Note that each of these algorithms can be used for prediction or retrodiction. Prediction is the estimation of a variable at a future point in time. For example, a detection algorithm can predict the existence of a potential threat at a future time. Similarly, a classification algorithm can predict the class to which a potential threat will belong at a future time. Conversely, retrodiction is the estimation of a variable at a time point in the past. An algorithm can detect the existence of a potential threat at a time point in the past, as well as determine its class membership. The algorithms may also be used to estimate current conditions in locations where data have not been collected.

While data assimilation and resource allocation depend explicitly on (physics-based) models, these models can be incorporated into the other algorithms. Detection algorithms often calculate a metric based on the collected data and then compare that metric to a threshold; if the metric surpasses threshold, then an alarm is sounded to indicate the existence of a potential threat. Models can be used to set this threshold in the first place. For example, a statistical model can be built of background radiation; any radiation found to be greater than 95 percent of all modeled background can be evidence of a potential threat. Classification algorithms can make use of models in several ways, as well. For example, the parameters of a model can be optimized by fitting them to the collected data; these parameters can then be used as features to input into a classification algorithm such as K-nearest neighbors, a support vector machine, or a neural network.

Finally, each of these three types of algorithms can also be thought of in terms of hypothesis testing and hypothesis generation. In hypothesis testing, a null hypothesis is declared (e.g., "Gamma radiation is not present"). The collected data are then analyzed to determine if there is enough evidence to reject the null hypothesis (e.g., "There is evidence to reject the null hypothesis that radiation is not present"). This is often called *confirmatory data analysis*. However, when a hypothesis is not yet known, *exploratory data analysis* must be used instead. Here, the collected data are analyzed to determine if they cluster or can otherwise be organized in a particular way. The structure of that organization may then suggest a hypothesis. For example, the features extracted from the collected data may cluster in high-dimensional space. The label of the cluster may be unknown. However, the region of space in which it resides may indicate that the label should be "gamma radiation," indicating that the collected data should be classified as gamma radiation – a human in the loop is often required to make this hypothesis. The hypothesis must then be tested statistically using confirmatory data analysis.

A functional taxonomy used to bin algorithms into different types can be applied to DTRA's operational missions. Direct force protection relies heavily on detection algorithms for hypothesis testing. Detection algorithms are used to alert users to the existence of a potential threat, possibly cueing a targeted search. Depending on sensor density, additional resource tasking in support of localization may be needed to cue targeted search. The challenge with direct force protection is fidelity/certainty – detection algorithms must be able to accurately determine the existence of a potential threat even when only a small sample is available. In addition, the algorithms must also be able to function quickly, so that a quick response can be made. This leads us to the second operational mission: targeted search.

Targeted search makes use of localization and classification algorithms for hypothesis testing. Localization algorithms, supported by resource tasking, are needed to home in on the exact location of the potential threat. Classification algorithms are needed to distinguish threat from non-threat objects or materials or to distinguish between different types of threats. Once again, the challenge with targeted search is *fidelity/ certainty* – the algorithms must be able to determine the location and class membership of the potential threat based on a small sample in a short span of time, so that quick decisions can be made about the fate of the area searched.

Long-term threat behavior monitoring relies heavily on detection for hypothesis generation. As in direct force protection, detection algorithms are used to determine the existence of a potential threat. Localization algorithms are needed to specify at least a rough area in which the threat-related activity may be found. Data assimilation algorithms help to develop an understanding of "normal" background. In contrast with direct force protection, however, long-term threat behavior monitoring is often used for hypothesis generation, rather than hypothesis testing. Hypotheses are rarely known in advance because the set of possible threat-related behaviors contains unknown and possible infinitely many members – and because the sensors employed are not specific to

what the system is attempting to monitor or detect (threat behavior vice threat material properties). These features of the long-term threat behavior-monitoring use case lead to a situation where a particular threat behavior (e.g., construction of an improvised nuclear device) may not have a threat signature that is known a priori and that can be looked for in the sensor data. As a result, these systems often require a human in the loop to generate hypotheses.

Once a hypothesis is generated, then responses can be made to acquire the data necessary to test the hypothesis. For example, if the coordinates of the potential threat are fairly well known in both time and space, then the response can include a targeted search. If, however, the spatial location of the potential threat is fairly well known but the time at which it may occur is not, then the response can include the deployment of sensors similar to those used in direct force protection – the choice of sensors is based on the hypothesis that must be tested (e.g., if the hypothesis is "Gamma radiation is not present," then gamma radiation sensors must be deployed). Finally, if the time at which a potential threat may occur is relatively well known but its location is not, then the response can include a wide area search.

A wide area search relies upon localization and classification algorithms for hypothesis testing and/or generation. In contrast to a targeted search, the location of the potential threat is not known. Therefore, a wide area search is heavily dependent on localization algorithms. Classification algorithms are then required to distinguish threat from non-threat objects and materials. A wide area search often results in a clustering exercise (a form of exploratory data analysis), rather than a full classification. Humans are required to determine what the label of a particular cluster might be, based on where it resides in high-dimensional space. A further response (e.g., targeted search or deployment of detectors similar to those used in direct force protection) can then be done to test any generated hypotheses.

The algorithms used by each operational mission are challenged in different ways. Those used in direct force protection and targeted search are challenged by fidelity/ certainty – they must be able to quickly determine the existence of a potential threat, as well as its location and class membership, all with a high degree of certainty, so that appropriate responses can be taken. The algorithms used in long-term threat behavior monitoring and wide area search are challenged by the need for exploratory data analysis for hypothesis generation. Follow-on responses are then needed to test the hypotheses. In all cases, careful consideration must be made for how each type of algorithm can be designed and operated in order to meet the challenge posed by each operational mission.

One can imagine missions where each of these use cases is supported by either a homogeneous or a heterogeneous sensor network. A homogeneous sensor network is a network composed of a set of identical sensors that are operated in a uniform manner (i.e., each sensor in the network responds to the same phenomena and collects and reports data in the same manner). One example of a homogeneous sensor network would be a set of identical cameras distributed throughout an area of interest in order to monitor

activity. A heterogeneous sensor network is composed of sensors that respond to more than one phenomenon and may collect and report the data in different manners, such as different collection frequencies and data formats. One example of a simple heterogeneous sensor network is a direct force protection system that integrates a perimeter laser trip line with a camera. A heterogeneous sensor network can pose the additional challenges to fusion algorithms of properly handling issues such as uneven sampling and missing data.

4. Case Studies

In this chapter, we discuss several case studies. Our goal is to illustrate how the algorithms and levels of fusion that we have discussed come together in real-world applications. We selected these examples to illustrate choices, challenges, and lessons learned that we believe are relevant to the CBRNE mission.

A. Fusion in Tactical Detector Networks

Most operational tactical networks of detectors employ more than one type of fusion, and they may even employ multiple algorithmic methods of fusion within each type. The communication network that enables the data exchange may also be heterogeneous, and various links may operate on different time scales, with some information being reported frequently and other information intermittently. The data may come from different phenomena. Even when data are derived from a single phenomenology, the detectors may be of different manufacture, different sensitivity, and different resolution. For all of these reasons, a tactical data network has to accommodate the complexities of the individual sources and make use of the most appropriate fusion methods to achieve its objectives. To illustrate, we describe the problem space and the solution for a current tactical air defense network.

A network of detectors can operate on various time scales simultaneously: from immediate real-time data exchange, to a slower schedule of periodic exchanges, or even event-driven, unscheduled data exchanges. The data exchange architecture is shaped by the objectives of the system and the practicalities of constructing the communication architecture. The use of detector networks is common and ubiquitous, but that does not mean all networks are easy to implement. The military, for instance, has to address the challenge posed by mobile detectors and hierarchies of communication networks. The military needs communication systems that work reliably, securely, rapidly, and operate in threatening environment anywhere in the world.

A good example of a military detector fusion system is the J3.2 Air Track message set of the Link-16 message system. Link-16 is designed for maritime, land, and space surveillance and engagement coordination. We discuss the J3.2 Air Track message, which is a subset of Link-16 that is designed to provide situational awareness to Air Force aircraft and Navy aircraft and ships. The goal is to provide participants and commanders with a common understanding of the location and classification of all air vehicles in the theater.

A detector platform (such as an aircraft or ship) with Link-16 broadcasts J3.2 air target track reports from its local radar to the other network participants. Each track

report is created by combining multiple individual radar measurements according to a physical model of nominal aircraft movement. For example, although they may be detected, it is not desirable to broadcast reports on the movement of birds. Therefore, the tracker tries to discriminate bird detections based on characteristics of bird flight, although a perfect separation of the class of birds from the class of aircraft is not always possible. Similarly, low-flying aircraft (such as helicopters) have to be modeled to distinguish them from ground vehicles, a distinction that is not always cleanly made.

The formation of a track from the measurements of a single detector is an example of single-detector data fusion. Other network participants compare this track with their own radar tracks, and, if they do not already have a track "like" it, they add it to their track database. On the other hand, if one or more participants find they are tracking the same target, they will negotiate over the network with the original reporter of the track to determine who should have reporting responsibility for that track. Link-16 attempts to provide an adjudication system such that one and only one Link-16 participant has reporting responsibility for each tracked target in the theater. Within the J3.2 message there is a Track Quality field that is scaled from each platform's estimate of error. This field is the key factor in determining which platform has reporting responsibility. The Track Quality field does not report a full covariance matrix for uncertainty, but is instead a combined metric.

In practice, the adjudication system is never perfect. Measurement uncertainty is inherent in the radar, and the translation of a track from one frame of reference to another contributes additional errors. Because the detectors (radars) and the feature processors (trackers) are not necessarily the same for each platform, a slightly different physical model may be used by each platform to form its track. As a consequence, situational awareness provided by Link-16 includes many errors, including unreported targets, the same target reported in multiple locations, and tracks with the same location but confused identifications.

Link-16 falls short of the definition of a feature-fusion system. It does combine the tracks (features) from multiple platforms to create a representation of the air picture for each participant. But it lacks a process for combining individual tracks to achieve a more accurate track. It is better described as a combination of tracks rather than a fusion of tracks. In addition, each participant maintains a different combination of tracks (its own plus the network track). If all of the tracks from all of the participants were fused, then the network could provide each participant with the same fused picture. (It would be difficult for Link-16 to achieve this because it has limited bandwidth.) To perform fusion on this network, two things would have to change: first, all of the tracks from all of the participants would have to be reported, and each participant would have to also transmit the measurement error (full covariance matrix) associated with each track. (As a note, the J3.6 Space Track message set does convey the full uncertainty matrix.) Feature fusion cannot take place without an understanding of the uncertainty associated with each

feature. Without taking the uncertainties into account, the fusion of the tracks would produce a degraded picture, not an improved one.

The air picture as it has been just described is a single phenomenology network (radar only). In fact, the air picture does incorporate other modalities. Most platforms in the network also have detectors to measure IFF (identification friend or foe), and some have detectors that can measure emissions from air vehicles that may be used to classify them. For example, detection of radar or radio emission might be sufficient to classify the type of aircraft, or at least determine a feature of the aircraft, such as what type of radio or weapon it is using.

Another form of evidence that can contribute to the classification of the target is the target's behavior. For example, a target that takes off from an airfield inside enemy territory or one that fails to fly inside designated return-to-force corridors might suggest that the target be identified as hostile. Typically no single decision is regarded as sufficient to declare a target to be hostile. A voting procedure is used to arrive at a fused decision.

The Link-16 operator who has met the rules for level of evidence to classify a target will attach his classification decision to the track that is broadcast to all network participants. Meanwhile, other operators are forming their own decisions. Because of the distributed nature of the decision making and the inherent differences in the local air picture, disputes about the classification of a target on the network are not infrequent.

Thus, within the current air defense tactical data network, we see implementations of all three categories of fusion algorithms: data fusion, feature fusion, and decision fusion. The methods employed have evolved over time, and newer and better techniques are not always economically feasible. The network requires the constant attention of human operators who work to remove errors and negotiate the resolution of conflicting evidence. The result is not as optimal as a design from the ground up might achieve, but it reflects the exigencies of the military situation. The example of the air defense network provides an illustration of how complex it can be when seeking to develop a tactical data network. Different levels of implementation and even selections within the message standard itself lead to different capabilities for data and information fusion.

B. Implantable Cardioverter Defibrillators

Many systems for threat detection involve a three-step process:

1. Anomaly detection, in which algorithms are used to detect a change from the normal state, cueing the system to take further notice.
2. Classification, in which algorithms are used to classify the detected anomaly as "threat" or "clutter."
3. Response, in which algorithms are used to determine what type of action should be taken to respond to a "threat" classification.

Physical models can be used in each of these three steps.

Many medical devices also involve this three-step process. Detection and classification algorithms are used to diagnose disease. Response algorithms are used to determine how to treat disease. All three of these algorithms are often based on physiological models that predict signals recorded from patients. One example of a medical device that uses the three-step process is the *Implantable Cardioverter Defibrillator (ICD).* This section describes the algorithms used in ICDs and discusses their pros and cons.

An ICD is a small electronic device that diagnoses and treats fast, lethal heart rhythms. As shown in Figure 4-1, an ICD is implanted under the skin in the upper chest. Electrodes extend from the ICD, insert into veins leading to the heart, and lodge in the interior walls of the heart. The electrodes continuously record electrical signals from the heart tissue. Algorithms running in the ICD process the signals to detect fast heart rhythms (anomalies) and classify them as lethal (threat) or non-lethal (clutter). Electrodes also deliver electrical energy to the heart in order to treat lethal rhythms (respond to threats) (National Heart Lung and Blood Institute 2013).

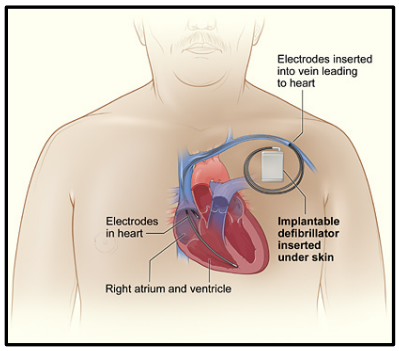


Figure 4-1. Implantable Cardioverter Defibrillators (ICDs)

ICDs are small electronic devices implanted in the upper chest wall to diagnose (detect and classify) and treat (respond to) fast, lethal heart rhythms (threats). The atria are the two upper chambers of the heart and the ventricles are the two lower chambers (National Heart Lung and Blood Institute 2013).

1. Physiological Models

Many threat detection systems use physical models to predict signals that should be recorded during normal conditions (when no threat is present) and emergency conditions (when a threat is present). ICDs do the same. Specifically, ICDs use physiological models to anticipate the signals recorded in patients during three different states: (1) *Normal Sinus Rhythm (NSR),* (2) *Supra Ventricular Tachycardia (SVT),* and (3) *Ventricular Tachycardia (VT).*

NSR refers to a patient's normal heart rhythm at rest – the normal state, when no threat is present. The heart beats due to repeated electrical depolarization of its tissues. In a healthy person, this depolarization originates in the two upper chambers of the heart,

the *atria*, and then conducts down into the two lower chambers of the heart, the *ventricles*. As a result, the heart beats in a coordinated fashion – first the atria, then the ventricles – at a rate sufficient to deliver an appropriate amount of oxygen to the body's tissues. The top diagram in Figure 4-2 shows the direction in which the electrical depolarization conducts during NSR (National Heart Lung and Blood Institute 2013).

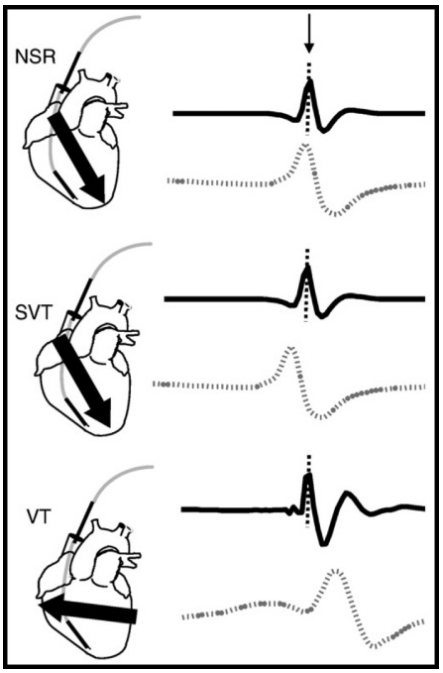


Figure 4-2. ICD during Three States

Top: Normal Sinus Rhythm (NSR) is the normal condition, when no threat is present. The electrical depolarization (black arrow) originates in the atria and conducts down into the ventricles for each heartbeat. The near-field (black) and far-field (grey) signals can be recorded by an Implantable Cardioverter Defibrillator (ICD). *Middle:* Although Supra Ventricular Tachycardia (SVT) is not the normal condition, no threat is present. The electrical depolarization conducts in a similar direction as NSR, but at a faster rate. The near- and far-field signals are similar to NSR. *Bottom:* Ventricular Tachycardia (VT) is the emergency condition, when a threat is present. The electrical depolarization originates in the ventricles and conducts "backwards" through the heart, in a different direction to NSR. As a result, the near- and far-field signals are different to NSR.

SVT is a non-lethal, fast heart rhythm – although this is not the normal condition, no threat is present. When a person exercises or becomes excited, the tissues of the body require more oxygen. Therefore, the heart must pump more blood. One way the heart does this is by increasing the rate at which it beats. The electrical depolarization still originates in the atria and conducts down into the ventricles, however, just like NSR. The middle diagram of Figure 4-2 shows the direction of depolarization conduction during SVT. SVTs are clutter – they are non-lethal and do not require treatment. No response is necessary (Shome et al. 2010).

VT is a lethal, fast heart rhythm – this is the emergency condition, in which the threat is present. For some people with cardiac disease, the electrical depolarization can become disturbed, originating in the ventricles and traveling "backwards" through the

heart, as shown in the bottom diagram of Figure 4-2. This results in fast and erratic beating of the ventricles. Insufficient blood is delivered to the body. If left untreated, the person can die within a few minutes (Shome et al. 2010).

2. Sensors

Some threat detection systems use a single sensor to collect only one type of data, but other systems use multiple sensors to collect multiple types of data. Alternatively, some systems may use a single sensor configured in multiple ways. As a result, the single sensor can provide more than one type of data. ICDs are an example of such a system.

An ICD electrode can be configured in two different, yet complementary, ways: (1) a *near-field* configuration for local measurements and (2) a *far-field* configuration for global measurements. Figure 4-2 shows near- and far-field signals recorded during one heartbeat of NSR, SVT, and VT. A *near-field* electrode records the voltage difference across the small section of heart tissue surrounding the tip of the electrode. As a result, a near-field electrode measures the sharp change in voltage as the electrical depolarization sweeps past its tip; these data are useful for assigning a fiducial timestamp to an individual beat. In contrast, a *far-field* electrode records the voltage difference across the entire heart, from the electrode tip to the ICD. Far-field electrodes measure the superposition of voltage changes as the electrical depolarization conducts throughout the entire heart over the course of an entire beat. As a result, far-field signals exhibit richer texture, useful in determining the origin and direction of the depolarization conduction (Shome et al. 2010).

The use of a single sensor configured in multiple ways has pros and cons. One advantage is that fewer sensors must be used. This can be particularly advantageous if the sensors must be placed in a location that is difficult to reach, such as the inside of the heart. In the case of ICDs, only one electrode must be inserted into the heart; this can reduce implantation time and the risk of infection. One disadvantage is the extra research and development required to create sensors that can be configured in multiple ways. Another disadvantage is the complexity involved with calibrating the sensor in its multiple configurations and keeping track of which data were recorded under which configuration.

3. Algorithms

Threat detection systems must process the data recorded by sensors. As discussed above, ICDs are an example of a system that arranges its algorithms into a three-step process: (1) detection of anomalies (fast heart rhythms), (2) classification of a detected anomaly as a threat (lethal VT) or clutter (non-lethal SVT), and (3) response to a threat classification (lethal VT). This section describes the algorithms used by ICDs to perform each of these three steps and discusses the advantages and disadvantages of the algorithms.

a. Anomaly Detection

ICDs use simple algorithms to detect anomalies, cueing the system to take further notice. An anomaly is any change from the normal condition. In the case of ICDs, this is any change from NSR. Anomaly detection in ICDs is based on the premise that the rate at which the heart beats is a useful indicator of whether the heart is in its normal state; a change in heart rate indicates a change from NSR. First, ICDs use a series of rules to identify the large, sharp deflections in the near-field signal indicating individual heartbeats. Because the near-field signal measures quick, local changes in heart tissue, the timestamps of these deflections can be measured with high certainty and can therefore be used as *fiducial points* for the heartbeats. Next, the heart rate is estimated based on the time intervals between successive fiducial points. This heart rate is then compared to a physician-programmable threshold. If the heart rate surpasses threshold, then an anomaly (a fast heart rhythm) is detected. Thus the cue (detection of a fast heart rhythm) is based on a single feature (heart rate) extracted from a local measurement (the near-field signal) and compared to threshold.

Anomaly detection has pros and cons. One advantage is that a model must be constructed for the normal condition only. One does not have to build a model of the abnormal condition; in fact, there may be so many different ways of being abnormal that it may be impossible to construct models of all of them. Instead, one can simply monitor the signal for a change from normal; this can often be accomplished using rather simple methods. In the case of ICDs, one must only know the range of heart rates expected during NSR; all other heart rates are considered anomalies. The disadvantage of anomaly detection is that one cannot determine why or how the signal is abnormal. The signal may be abnormal because a threat is present or simply because of noise or other background clutter. In the case of ICDs, a heart rhythm may be fast because it is a lethal VT (a threat) or simply a non-lethal SVT (clutter). Thus anomaly detection *on its own* is not sufficient to guide response. An interim step is needed: classification.

b. Classification of Threat vs. Clutter

Once an anomaly has been detected, it must be classified as either threat or clutter. ICDs classify a fast heart rhythm (an anomaly) as either lethal VT (threat) or non-lethal SVT (clutter). Two types of classification algorithms are used in ICDs: (1) interval classification and (2) morphology classification. Some ICDs integrate both types of algorithms.

1) Interval Classification

Interval classification algorithms in ICDs extract features related to the time intervals between heartbeat fiducial points. As discussed above, these fiducial points are local measurements extracted from the near-field signal. Some ICDs extract two new features from these time intervals, *stability* and *onset*, and compare each to a threshold. *Stability* measures the variance in the time intervals; *onset* measures how quickly

successive time intervals shorten (how quickly the heart rate speeds up). A physiological model is created for both lethal VT (threat) and non-lethal SVT (clutter); classification uses these models to determine whether stability and onset features indicate VT or SVT. Specifically, a fast heart rhythm that is stable (stability greater than threshold) with a slow onset (onset less than threshold) is classified as SVT. Other fast rhythms are classified as VT. A physician can program the thresholds (Boston Scientific 2013).

There are pros and cons to interval classification. One large advantage is that it is simple. The simple calculation of features (stability and onset) and comparisons to threshold can be reviewed quickly to understand why a classification decision was made. However, one large disadvantage to interval classification is that physicians must remember what each feature is meant to represent physiologically so that appropriate threshold values can be selected for a particular patient.

2) Morphology Classification

Morphology classification uses more complex algorithms to process the far-field signal. Since the far-field signal is a global measurement, it can be used to determine the origin and direction in which the electrical depolarization conducts throughout the entire heart over the course of an entire beat. This type of classification is based on the physiological model illustrated in Figure 4-2 (Shome et al. 2010): signals recorded during SVT are similar to NSR, since, in both cases, the electrical depolarization originates in the atria and conducts down into the ventricles. In contrast, signals recorded during VT are different from NSR, since depolarization conducts in the opposite direction.

Morphology classification requires an additional calibration step: recording the signal during the normal state. In the case of ICDs, this step must be performed immediately after the ICD is implanted in the patient's chest. Signals are recorded while the patient is in NSR, the normal state. Features are extracted from these signals and stored as the *NSR template* for later use. As soon as a fast heart rhythm is detected, the same features are extracted from the fast rhythm's signals and compared to the NSR template. Fast rhythms that "match" the NSR template are classified as non-lethal SVT and left untreated. All other fast rhythms are classified as lethal VT and treated (Shome et al. 2010; Swerdlow et al. 2002).

Morphology-based algorithms also have both pros and cons. The main advantage is that the classification decision is based on the richer far-field signal representing global measurements. One always wants to use as much information as possible to make a decision. There are two main disadvantages. First, creation of the normal template requires an additional calibration step, which can be burdensome, particularly if the normal state changes frequently, requiring a frequent update to the normal template. Second, the algorithms are complex and can be difficult to understand. Many physicians regard these ICDs as "black boxes," making them more difficult to trust.

3) Combined Classification

Some ICDs use multi-tier decision trees to combine interval and morphology classification. An early tier uses interval techniques to classify a fast rhythm as SVT (clutter) or Unknown. If the rhythm is classified as SVT, the analysis ends, and no treatment is given. If the rhythm is classified as Unknown, the analysis continues. In a later tier, morphology techniques classify the Unknown rhythm as SVT (clutter) or VT (threat). Thus the early tier *screens out* clutter, but the later tier performs a full classification (Medtronic 2012). This is an example of decision fusion, in which decisions output from multiple algorithms are combined to arrive at a final decision.

Multi-tiered, "screen-then-classify" decision trees have pros and cons. One large advantage is that they take into consideration the different perspectives of the constituent algorithms. A large disadvantage is the nuance of combining multiple constituent decisions into one. The early tier's algorithm must rarely output a false negative – an anomaly classified as clutter when it was truly a threat. Otherwise, the anomaly will not have the opportunity to be classified correctly by a later tier. No response will be made for what is truly a threat, leading to potentially tragic consequences. In the case of ICDs, the patient will die. On the other hand, the early tier may often output a false positive – an anomaly classified as Unknown when it was truly clutter. All Unknown rhythms will have the opportunity to be classified correctly by the later tier. Thus one must carefully consider the tested performance of an algorithm *on its own* before determining how it can best complement other algorithms.

c. Response to Threat

Detected anomalies that are classified as threats require a response. In the case of ICDs, lethal VTs must be treated immediately or the patient will die. Preprogrammed instructions dictate how ICDs respond to VTs. Physicians must program these instructions as part of the calibration process immediately after implantation.

Many physicians use an approach similar to the military's "escalation of force" philosophy. The ICD is programmed to first respond to a VT classification by delivering a series of short, low-energy bursts, similar to the output of a pacemaker. In many cases, the low-energy pacing can halt VT and return the patient back to NSR, the normal condition. If the low-energy pacing does not succeed, then the ICD is programmed to respond more aggressively. A different pacing waveform may be used, or the pacing may occur at a faster rate. If unsuccessful, then the ICD escalates to the third tier of therapy, a high-energy shock. These shocks are similar to those that physicians deliver through paddles applied externally to the patient's chest. If the shock does not succeed, then the ICD escalates to its final tier: another shock delivered with the maximum energy possible (National Heart Lung and Blood Institute 2013).

Preprogrammed instructions for threat response have pros and cons. For ICDs, the main advantage is that physicians can specify in advance exactly how the ICD will respond to a VT for a particular patient. Some patients are so unhealthy that no time

should be wasted on lower tiers of response. A high-energy shock is required immediately upon VT classification. Other patients may be healthy enough to tolerate the time spent on lower (and less painful) tiers of therapy; low-energy pacing (which does not cause pain) can sometimes terminate VT without the need for high-energy shocks (which do cause pain). Disadvantages also exist with preprogrammed instructions. In the case of ICDs, physicians must anticipate all possibilities when programming the instructions, because the ICD cannot adapt automatically to unanticipated heart rhythms.

4. Remaining Challenges

ICDs have been in use for 30 years. Most challenges have been addressed. Randomized control trials have shown that ICDs can reduce mortality by 31 percent compared to pharmaceuticals (Moss et al. 2002). Some challenges do remain, however.

One challenge is the cost of false positives. Classification algorithms must always trade off the costs of false positives (true clutter misclassified as threat) and false negatives (true threats misclassified as clutter). The costs associated with false negatives are usually quite high; if a threat is not correctly classified, no response will be made, potentially leading to tragic consequences. In the case of ICDs, a true VT misclassified as SVT would lead to death within minutes. Therefore, the classification algorithms in ICDs are tuned to err on the side of caution; false negatives are minimized, resulting in higher false positives. False positives can come at a cost, too, however. If clutter is misclassified as a threat, then a response will be made unnecessarily. In the case of ICDs, some responses, such as high-energy shocks, can be quite painful. A false positive precipitating a shock comes at a great cost to the patient. Other responses, such as low-energy pacing, are not painful and can sometimes go unnoticed. As a result, a false positive leading to pacing comes at a very low cost. Therefore, further research is currently being done in the ICD industry to (1) lower false positives and (2) reduce the cost of response.

Regulation is another challenge. ICDs are subject to some of the strictest regulations in the health care industry. This is due to the fact that an ICD has only a minute or two to detect, classify, and respond to a VT before the patient dies. There is simply no time for a physician to intervene if the ICD malfunctions. Testing new sensors and algorithms is therefore very risky. Clinical studies must be designed to mitigate those risks. Mitigations can be expensive and time-consuming, however. Therefore, ICD technology evolves very slowly.

C. Autonomous Ground Vehicles

Development of autonomous ground vehicles involves the collection and analysis of heterogeneous data from multiple sources, the fusion of these data, and the use of the fused data in an intelligent, real-time, decision-making process. Autonomous ground vehicles are made possible by the combination of several complicated processes; however, the generic strategy can be described as “sense-think-act.” Data collection and

data analysis from a distributed network of sensors is typically the first step, followed by processing to develop a model of the vehicle's surroundings and determine possible actions. In the final step of the process, the model is converted to a decision-making tool and an action is decided upon and executed.

The Defense Advanced Research Projects Agency (DARPA) has sponsored a series of Grand Challenges that focused on autonomous vehicle development. The first DARPA Grand Challenge was a vehicle race in the Mojave Desert. Vehicles were required to be entirely autonomous and to complete the 143-mile course in less than 10 hours. Successful completion of the task required use of multiple sensors, fusion of collected data, development of an accurate model of the vehicle's surroundings and of vehicle movement, utilization of decision-making algorithms, and ability to execute the decided-upon action (Thrun et al. 2007). When Carnegie Mellon University's vehicle Sandstorm traveled the farthest at 7.3 miles, the competition was held again the following year and the prize was doubled to $2 million. Five vehicles successfully completed the course. Stanford University's vehicle, Stanley, won the race with a time of 6 hours 54 minutes.

DARPA pushed the envelope of autonomous vehicle development again in 2007 with its Urban Challenge. This time the course was 60 miles through an urban environment and was to be completed in 6 hours. The urban environment added a new component to autonomous vehicle development: it was now necessary for the vehicle to obey all traffic laws, detect and avoid other vehicles, and make decisions in real time based on the actions of other vehicles. A collaborative team of Carnegie Mellon University and General Motors Corporation claimed the $2 million prize with their vehicle Boss.

The development of autonomous ground vehicles is characterized by several attributes. The general strategy of DARPA Challenge entries was a sense-think-act framework. Each vehicle had more than a dozen sensors of different modalities, incorporating optical, radar, and GPS data collection. These data were combined and were incorporated into physics-based models. Multiple models were used to develop a coherent picture of a vehicle's environment, and a cost map was developed to evaluate potential actions. Use of terrain evaluation and navigation algorithms enabled a vehicle to decide upon and execute a path.

Because no single sensor could provide the range of wavelengths and field-of-view required to navigate the Challenge courses, vehicles were equipped with a suite of sensors. Sensor data were used to fulfill two primary goals: evaluation of the surrounding terrain and detection and characterization of obstacles. Most vehicles had at least the following sensors: multiple lidar line scanners, a radar scanner, a stereo video camera, and an inertial and a differential GPS sensor. Long-range lidar line scanners were used by many vehicle teams to provide data for terrain topology mapping and obstacle detection and characterization. Lidar line scanners, radar scanners, and stereo video cameras provided information used for obstacle detection (Urmson et al. 2004). As

discussed in Darms et al. (2009), the combination of lidar and radar data was necessary to provide the proper blend of long-range detection with short-range shape or obstacle estimation, and although the performance of both stereo cameras and lidar can be hindered by the presence of dust, radar operates at a wavelength that penetrates through dust. Multiple sensors with overlapping fields-of-view and complementary utilities increased robustness against false readings and sensor failures. In addition to the ability to provide accurate and complementary data, sensors were selected based on reliability, availability, and ability to provide timely information.

The use of a suite of sensors was accompanied by several challenges. Large data sets were produced, and the issue of data processing and transfer had to be resolved. For example, Stanley (the 2005 winning vehicle) received data from sensors at frequencies ranging from 10 to 100 Hz and control of vehicle steering, throttle, and braking occurred at frequencies up to 20 Hz (Thrun et al. 2007). To facilitate the collection, analysis, and transfer of high-frequency data, many successful vehicles employed high-performance file servers and computing power. As described in Urmson et al. (2004), Carnegie Mellon's 2004 vehicle included a 2 terabyte (TB) RAID5 array and a dual Xeon processor-based server. With a well-designed architecture, it was possible to ensure that computational power would not limit vehicle performance.

A common design feature among DARPA Challenge vehicles was the development of a "Sensor Layer" that was kept separate from the other components of the vehicle architecture such as the "Thinking Layer" and the "Decision-Making Layer." Data collection and processing occurred in the Sensor Layer before being sent to other parts of the system. A strategic decision made by many vehicle teams was to timestamp, rather than synchronize, the data (Thrun et al. 2007; Effertz 2008; Darms et al. 2009). This reduced the risk of deadlocks and processing delays (Thrun et al. 2007). Typically, after the timestamp was added, sensor data were analyzed and prepared to be sent to the Thinking Layer. For example, as discussed in Effertz (2008), lidar signals, obtained from illuminating a target with laser light and collecting the backscattered light, were processed and converted into object-oriented data that included obstacle distance, width, and relative velocity.

Successful vehicle entries also used a publish-subscribe paradigm (Thrun et al. 2007) to manage information. Data flowed in one-direction and were analyzed and published in a database. Only necessary data were sent to the appropriate users. By isolating the collection and analysis of sensor data and submitting processed data to the Thinking Layer, the subsequent model, cost map, and decision-making remained sensor agnostic (Urmson et al. 2004). Sensors and other hardware could be added and removed from the vehicle without revamping the entire navigation and decision-making architecture.

During the "think" stage of the process, features and objects extracted from sensor data were interpreted and incorporated into physics-based models. Feature extraction and object interpretation require assumptions such as vehicle size, surface reflectivity, and

color (Darms et al. 2009). One potential pitfall during this stage is misinterpretation of sensor data. This possibility is addressed by combining output of several complementary sensors and weighing the expected accuracy of individual sources. This accuracy may depend upon factors such as amount of ambient light, distance between the sensor and the object, and angle between the sensor and the object.

These processed data were used as inputs for models of the vehicle and its surroundings. Many successful Challenge teams used a variation of the Kalman filter (Kalman and Bucy 1961) for state estimation in their models. Most vehicles developed and managed several separate models. For example, a model was developed for the vehicle itself, as well a model for each obstacle, moving or stationary, encountered by the vehicle. These models were maintained and updated as needed. For example, after a vehicle passed a stationary obstacle, the model of that obstacle became irrelevant. These individual models were combined to develop an overall model of the vehicle's environment. According to Darms et al. (2009), a traditional mechanism for combining multiple-state models for object tracking is the interacting multiple model (IMM) (Mazor et al. 1998). IMM uses two or more Kalman filters in parallel. Each filter uses a different model for target movement. IMM determines an optimal weighted sum of all the outputs of the filters. Advantages of IMM include its recursive nature, which enables it to rapidly adjust to target movement. Another common approach is the use of the switching Kalman filter model (Veeraraghavan, Schrater, and Papanikolopoulos 2005). Like IMM, the switching Kalman filter utilizes multiple models to describe motion. The modeled motion is a weighted combination of multiple models, providing flexibility in describing and predicting vehicle motion. One disadvantage of the switching Kalman filter is the possibility of a high computational requirement, partially due to the requirement of data association. These and similar algorithms are useful in various tracking problems such as radar and GPS tracking.

The final stage in this generic framework is the "act" phase. DARPA Challenge teams used cost functions to transform the vehicle's world model, generated during the "think" stage, into a form that could be used for decision-making. The course was divided into a grid and, based on the developed model, the traversability of each grid cell was determined. The three metrics used by Carnegie Mellon University to determine traversability were slope, roughness, and step height, which is the change in elevation compared to the vehicle's current position. This traversability map was translated into a cost map by calculating cost as the inverse of the traversability score multiplied by the certainty of the data in the grid cell (Urmson, Simmons, and Nesnas 2003). A cost map was the decision-making tool for most Challenge vehicles. A vehicle's next action, stop, turn left, turn right, slow down, etc., was determined by the cost associated with occupying adjacent grid cells and the overall cost of the path. As described in Thrun et al. (2007), the overall path was planned, subjected to updating, as an implementation of a search algorithm that minimizes a linear combination of cost functions.

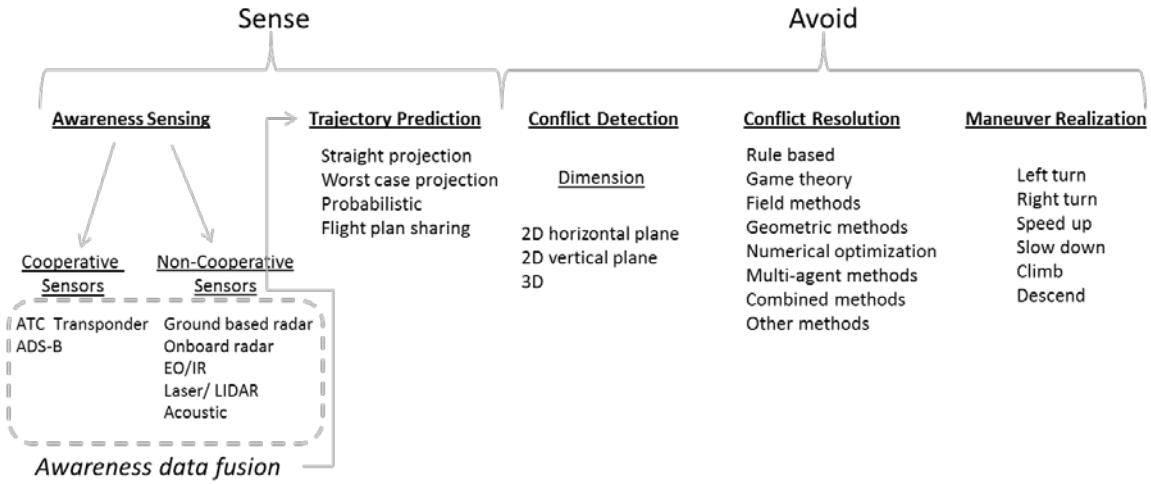
An important attribute of cost models used by successful Challenge vehicles was that only relevant parts of the map were updated as new data became available. Carnegie Mellon University employed the algorithm Dynamic A* (D*) to plan the vehicle's path (Urmson, Simmons, and Nesnas 2003). D* generated an initial cost map of the course from available a priori information. The map was then updated as new sensor data became available. An advantage of D*, and similar algorithms used by other Challenge teams, is that only regions of the course affected by the new information are replanned, not the entire course, avoiding excess memory storage and computational time. This algorithm enabled path planning in unknown, partially known, and changing environments (Stentz 1994) and resulted in computational efficiency and timely course adjustments.

The development of autonomous ground vehicles involves fusion of data from multiple sources and of multiple types, use of physics-based models to predict future events, and real-time analysis, decision-making, and action-taking. Autonomous ground vehicle development has been enabled by high-fidelity vehicle models, availability of high-resolution data of surroundings and objects of interest, and adequate computational power. These DARPA Grand Challenge vehicles were also well-served by a well-defined problem: traverse this course as quickly as possible. Impediments to development include determining how to weigh data from various data sources under different conditions and to develop accurate cost models. In a simplified form, the framework for autonomous ground vehicle development is a "sense-think-act" paradigm. The ability to execute autonomous motion is driven by combining data from multiple, complementary sensors, developing an overall model of the vehicle and its surroundings, and effectively transforming this model into a decision-making tool. In the case of autonomous ground vehicles, traversability converted into cost functions was a driving force in the ability to succeed in the DARPA Grand Challenges.

D. Sense and Avoid for Unmanned Aerial Vehicles

To fulfill mission objectives, UAVs must navigate while avoiding terrain and static obstacles as well as moving obstacles such as other UAVs, airplanes, and balloons, or areas with bad weather conditions. One basic problem still hinders the use of military or civil UAVs in commercially controlled airspace: flight safety in terms of collision risk with respect to other aircraft. In the absence of a human pilot on board the aircraft, a UAV must have a sense and avoid (SAA) capability to detect and resolve potential conflicts. A conflict is defined as the event in which the Euclidean distance between two aircraft is less than the minimum desired separation distance.

A SAA system has five basic functions: sensing, trajectory prediction, conflict detection, conflict resolution, and evasion maneuver generation. Figure 4-3 shows the approaches for accomplishing each of these five functions.



Adapted from Kopriva, Sislak, and Pechoucek (2012)

Figure 4-3. Sense and Avoid Taxonomy

1. Sensing

The SAA system monitors the surrounding environment for static and dynamic obstacles. Determining the types of sensors appropriate for a UAV and its environment is a challenging multidimensional problem (Lacher, Maroney, and Zeitlin 2007). The fundamental information that a sensor or group of sensors needs to acquire is the range, azimuth, and elevation of all targets of interest. Sensors deployed aboard aircraft to perform surveillance for collision avoidance can be divided into two main categories: cooperative and non-cooperative sensors.

Cooperative sensors receive radio signals from another aircraft's onboard equipment. Many aircraft carry a transponder that may be interrogated by other aircraft in a determined range to broadcast range, altitude, and bearing. The Traffic Alert and Collision Avoidance System (TCAS) for manned aircraft relies on this method to discover other aircraft.[1] By 2020, aircraft are required to be equipped with Automatic Dependent Surveillance–Broadcast (ADS-B), which utilizes GPS and broadcasts aircraft position, velocity, and other data without needing to be interrogated. Non-cooperative sensors are required to sense non-transponding targets or stationary obstacles and may do so actively or passively. Available passive sensing technologies include electro-optic cameras, infrared cameras, and acoustic sensors. These sensors tend to be smaller and lighter weight, but a high-resolution field of view might drive a high processing requirement. Active sensors include radar and laser range finding. Since these sensors require more energy to interrogate the target, they tend to be bigger and heavier and can thus be mounted only on larger platforms. For the smallest UAVs without the capacity to carry multiple sensors, receiving information from ground-based sensors might be an attractive option.

[1] http://en.wikipedia.org/wiki/Traffic_collision_avoidance_system.

To detect a potential conflict, the trajectory of a sensed target must be compared to the trajectory of the UAV. Four fundamental methods for trajectory prediction are cited in literature (Kuchar and Yang 2000; Albaker and Rahim 2011). In the nominal method, the trajectory is predicted directly from sensor data without consideration of uncertainties. The worst-case method assumes an aircraft might perform any range of maneuvers physically possible. The probabilistic method weighs each maneuver by a probability of occurring. Finally, the flight plan sharing method requires cooperative sensors to exchange parts of their flight plans giving an exact knowledge of the future trajectory.

2. Avoiding

Once the target trajectory has been predicted, this is compared to the flight plan of the UAV to check if the safety zone has been violated and, if so, a conflict is declared. This conflict is resolved using one of the collision avoidance methods, which include rule-based methods where the UAV avoids the conflict by acting from a prescribed set of rules; game theory methods where the conflict is modeled as a two-player differential game; field methods, which treat each object as a charged particle; geometric approaches, which find an optimal solution based on the geometric relationship between the UAV and the threat; numerical optimization methods, which use a cost metrics and a kinematic model together with a set of constraints to find a solution; and multi-agent methods in which each aircraft is controlled by an agent who communicates to negotiate a solution. The outcome of these methods will be a command for the UAV to perform an evasion maneuver or a combination of maneuvers such as speed up, slow down, maintain speed, turn left, turn right, climb, or descend. For a comprehensive review and categorization of recently published conflict detection and resolution approaches, see Kopriva, Sislak, and Pechoucek (2012).

3. Multi-Sensor Fusion in a SAA Context

Although the sense and avoid functions are independent, they deeply influence each other. For the best possible outcome (maneuver realization), conflict resolution algorithms need the best possible trajectory prediction, which results from the best possible information from the awareness sensors. The remainder of this discussion will focus on the sensing end of SAA, since this can be used to illustrate the utility of multi-sensor fusion to provide the best picture of a UAV's surrounding environment in a timely manner.

According to published performance standards, a UAV must be able to detect and avoid another airborne object within a range of $\pm 15^{o}$ in elevation and $\pm 110^{o}$ in azimuth and be able to respond so that a collision is avoided by at least 500 feet (Table 4-1). The 500-foot safety bubble is derived from the commonly accepted definition of what constitutes a near mid-air collision (Federal Aviation Administration 2012). Additional requirements are derived from these basic requirements and characteristics of a particular

UAV. For example, Geyer, Singh, and Chamberlain (2008) determine the minimum time to perform a basic maneuver, which determines the distance at which the threat must be detected. This time is a function of the maximum banking angle of the UAV. Given a maximum banking angle of 45°, a UAV would have approximately 6 seconds to avert any threat. This also suggests that every target must be detected at least 6 seconds in advance. For sensors whose field of view does not meet the required field of regard, it would then be necessary to scan such that the revisit rate meets the minimum detection time to perform the evasive maneuver. Finally, sensors must be capable of performing to these requirements at all times, day or night, in all weather.

Table 4-1. Field of Regard Requirements for Collision Avoidance

Field of Regard		
Source	**Azimuth**	**Elevation**
International Standards, Rules of the Air, Section 3.2.2.4 (ICAO)	±110°	No guidance
ACC/DR-UAV SMO Sense and Avoid Requirement for Remotely Operated Aircraft (ROA), 25 June 2004	±110°	±15°
American Standards Testing and Materials (ASTM) 2411.04	±110°	±15°
DoD Standardization Program Office	±110°	±15°

Source: McCalmont 2007

A key point to consider in the selection of sensors for SAA is that no single sensor would be capable of fulfilling all requirements discussed in the above paragraph. This is evident in the trade space for SAA sensor types and attributes laid out in Table 4-2 (Lacher, Maroney, and Zeitlin 2007). For example, radar is reliable in all weather and is able to accurately measure range, but the data rate is only about 1 Hz. Electro-optical/infrared (EO/IR) sensors, on the other hand, are best for angular measurement and have a very high measurement rate. Equipping a UAV with multiple fused sensors serves to combine the strengths of each to provide a complete solution. Having non-cooperative sensors as part of the multiple sensor architecture is mandatory since a threat might be intentionally non-cooperative or may not be equipped with cooperative sensors.

Table 4-2. Attributes of SAA Sensors

Sensor	Modality	Range	Bearing (Azimuth)	Bearing (Elevation)	Trajectory
Mode A/C Transponder	Cooperative	Accurate: 10s of miles	Calculated	Calculated based on pressure altitude	Derived
ADS-B	Cooperative	Accurate: 10s of miles	Calculated based on GPS	Calculated based on pressure altitude	Provided
Optical	Non-Cooperative, Passive	Not sensed	Accurate	Accurate	Derived
Thermal	Non-Cooperative, Passive	Not sensed	Accurate	Accurate	Derived
Laser/Lidar	Non-Cooperative, Active	Accurate; 1,000 ft	Narrow	Narrow	Derived
Radar	Non-Cooperative, Active	Accurate; 1 mile	360º	360º (Depends on antenna mounting	Derived
Acoustic	Non-Cooperative, Active	Accurate; 100 ft	360º	360º	Derived

Note: Table adapted from Lacher et al. (2007). Accuracies reflect a sensor's useful detection range for SAA.

4. Example: A Multiple Model Algorithm for Multiple Sensor Tracking

Rousseau, Ratton, and Fournet (2010) and Cornic et al. (2011) highlight why a multiple sensor system with a data fusion algorithm is helpful for situational awareness. Their study considers the UAV to have both radar and EO/IR sensors onboard, and both the UAV and the threat platform are equipped with an IFF transponder and ADS-B. They propose a hybrid hierarchical process for the fusion of the heterogeneous data from this suite of sensors. Here, each sensor does its own tracking by providing a detection-to-track association based on geometrical data as well as the specific signature of the target in the sensor's measurement domain. The sensor provides the fusion with its own tracks and the last data associated with these tracks. Then, data association is from individual tracks coming from each sensor. The multiple sensor fusion is obtained by integrating the associated sensor measurement data in a kinematics estimator. Note also that sensor management to reach the avoid goals may be influenced by the feedback of the fusion output.

The study chooses two models from civil aviation to represent the dynamics behavior of the targets the UAV will encounter: the constant velocity model and the coordinated turn model. For civilian aircraft, the coordinated turn model considers left and right turns and a typical (2°/s) and a maximum (6°/s) value for the turn rate. Since the UAV may also encounter maneuvering targets, such as combat systems, this is modeled using the constant turn model and increasing the rate of turn (9°/s).

The objective of the tracking algorithm is to detect a potential collision between the UAV and any surrounding targets. A target on a collision course will have a close to zero relative angular velocity; thus the tracking algorithm must estimate the target velocity

vector as accurately as possible. The major challenges of target tracking arise from two discrete-valued uncertainties: the measurement origin uncertainty (a measurement may have arrived from an extraneous source) and the target motion uncertainty. In the presence of target motion uncertainty, the IMM is considered to be the state-of-the-art tracking algorithm (Blom and Bar-Shalom 1988; Li and Jilkov 2005). It is considered the best compromise currently available between computational complexity and performance. IMM uses a set of multiple models that represent the possible behavior patterns of the targets the UAV might encounter. A bank of filters is run in parallel, each based on a unique model in the set. Target state estimation is then based on a weighted average of estimates from the different models.

The many different possible maneuvers for a target may not be covered accurately enough by only a small set of models. It has been well established, however, that the IMM performance will not improve, but will get worse as the number of competing models increases, aside from an increase in computational complexity (Li and Bar-Shalom 1996). Information from the sensors will become diffused across the entire set of multiple models instead of being limited to the most appropriate models. The authors thus propose using a Variable Structure IMM (VS-IMM) (Li 2000) for their application, which would be capable managing the set of IMM models and dynamically selecting the most appropriate models and parameters at a given time according to the current situation, i.e., the model set is not fixed over time. The dynamic selection of models is based on target classification (such as that provided by ADS-B) and the current state vector estimate combined with prior knowledge of target dynamics.

The VS-IMM approach used is Kalman filter-based. Specifically, the Extended Kalman Filter (EKF) is applied since the relationships between the target state and most of the sensor measurements is nonlinear. The steps of the EKF are shown in Table 4-3. The structure of their multi-sensor fusion algorithm is diagramed in Figure 4-4. The authors use this algorithm to simulate improvement in tracking using multiple sensors over tracking data from a single sensor.

Table 4-3. Extended Kalman Filter

Initialization	X_o, P_o
Prediction	$\hat{X}_{k+1\|k} = f_k(\hat{X}_{k\|k})$
	$P_{k+1\|k} = F_k P_{k\|k} F_k^T + Q_k$
	$\hat{Y}_{k+1\|k} = h_{k+1}(\hat{X}_{k\|k}, 0)$
Innovation	$I_{k+1} = Y_{k+1} - \hat{Y}_{k+1\|k}$
	$S_{k+1} = H_{k+1} P_{k+1\|k} H_{k+1}^T + R_{k+1}$
	with $H_{k+1} = \frac{\partial h_{k+1}(X)}{\partial X}\Big\|_{X=\hat{X}_{k+1\|k}}$
EKF Gain	$G_{k+1} = P_{k+1\|k} H_{k+1}^T S_{k+1}^{-1}$
Filtering	$\hat{X}_{k+1\|k+1} = X_{k+1\|k} G_{k+1} I_{k+1}$
	$P_{k+1\|k+1} = P_{k+1\|k} - G_{k+1} S_{k+1} G_{k+1}^T$

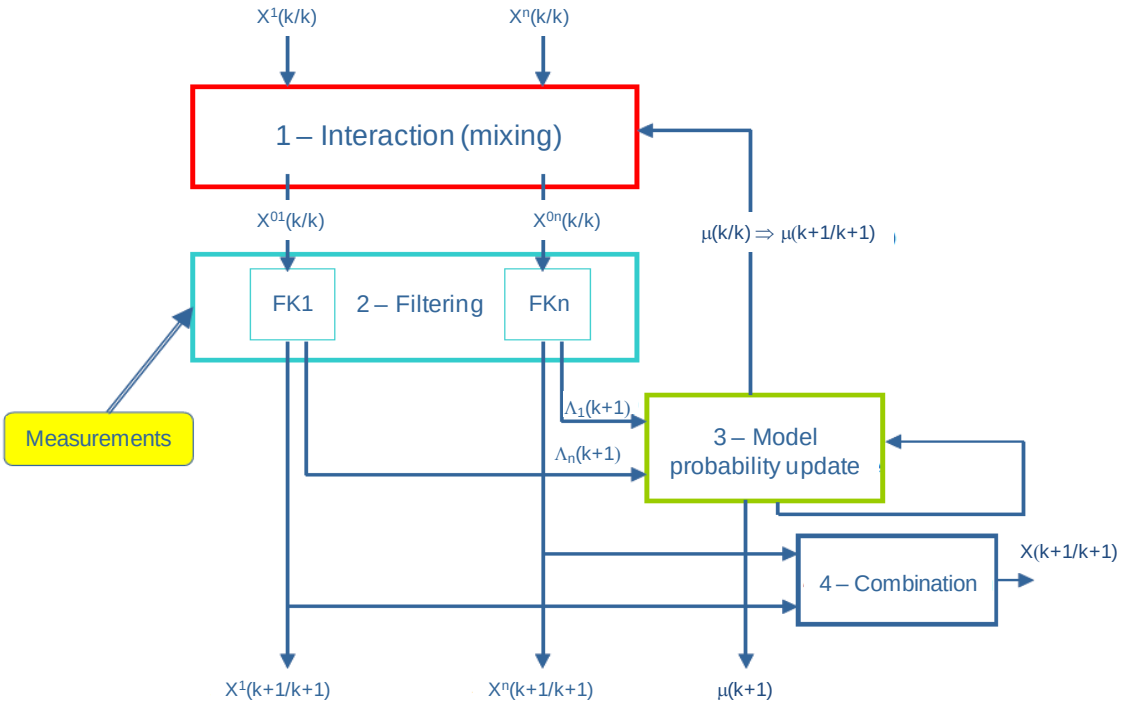


Figure 4-4. VS-IMM Algorithm for Fusion

5. Discussion

The multiple model approach is a state-of-the-art solution to many target tracking problems. The VS-IMM version is most useful in complex situations where a large number of models must be used to describe the target and where the state of the target is highly time variant. An approach to improving solutions is to design a better set of models to describe the target states: The better the motion model, the better the tracker is able to follow the target. This, however, is highly dependent on the specific problem being solved. Another breakthrough would be to produce new filters in a general setting rather than relying on ad hoc designs for each application. For an extensive discussion of VS-IMM and the design and evaluation of algorithms for target tracking applications, see Li (2000).

Note that SAA for UAVs is not a mature technology in the DoD. AFRL's SAA architecture contains an algorithm called MuSICA (Multi-Sensor Integrated Conflict Avoidance), which is in the demonstration phase and claims to have achieved TRL 6 (Graham and Kay 2012; Smith 2012). The Air Force posted a sources-sought notice this past summer in anticipation of releasing an RFP for the Airborne Sense and Avoid program, which seeks to develop a sensor agnostic, platform agnostic sensor fusion and avoidance maneuver product.[2]

[2] https://www.fbo.gov/index?s=opportunity&mode=form&id=db4546a08dd439f93510efccbf6632ff&tab=core&_cview=0.

The example presented in the previous section is somewhat analogous to CBRNE problems: the sensing and tracking problem involves a suite of heterogeneous sensors, and the best possible solution results from the fusion of sensor data in real time with a fusion algorithm based on not just one, but multiple physics-based models. There may also be cases where similar sensors are used for CBRNE as they are for tracking airborne threats, such as radar and EO/IR. Lundberg, Paffenroth, and Yosinski (2010) compare metrics for assessing target-tracking algorithms to CBRNE detection and tracking. For example, track completeness is an estimate of how many targets in a scenario have a track assigned to them and is also a desirable metric for determining how many plumes have been tracked. Spurious track ratio evaluates the ratio of spurious tracks to the target count and would again be important in CBRNE to determine a ratio of spurious clouds. The authors focus most heavily on covariance consistency and its application to CBRNE, which provides a measure how much uncertainty an algorithm thinks it has in its estimate.

E. Syndromic Surveillance

Biosurveillance is the ongoing monitoring of public health trends with the primary goal of informing the response of health authorities to the outbreak or possible outbreak of disease (Fricker 2013; Lawson and Kleinman 2005; Bravata et al. 2004). Biosurveillance objectives include situational awareness (SA) and early event detection (EED). *Clinical surveillance* is focused on situational awareness, which in this context is the monitoring of health indicators and environmental conditions to track the spread through the population of confirmed cases of an identified outbreak or biological attack. *Syndromic surveillance*, on the other hand, is predictive rather than descriptive and seeks early detection of an outbreak by the monitoring of environmental data and "syndromes."

A syndrome is intentionally defined non-specifically as a set of conditions associated with a disease and present prior to a medical diagnosis. The definition is kept broad to capture as many indicators of an outbreak as possible. Historically, designated sentinel physicians have notified the public health authorities of the incidence of highly communicable disease (today all physicians are required to do so), and health officials have used this information to map the outbreak and direct the response. As data collection and analysis have improved, syndromic surveillance has expanded this idea to disease precursors.

The monitoring of influenza-like illness (ILI) provides an example of distinctions between syndromic surveillance and clinical surveillance. Influenza-like illness is broadly defined to include influenza, colds, gastroenteritis, and such potentially lethal conditions as meningitis and the early stages of anthrax (MacIntosh 2004). In the clinical surveillance context, the Centers for Disease Control define ILI symptomatology as consisting of a fever greater than 100 °F, accompanied by cough and/or sore throat, with no known cause other than influenza. Physicians are instructed to report patients meeting the strict ILI case definition, unless another diagnosis is confirmed by laboratory testing.

For example, a patient with no cough or sore throat is not to be reported.[3] On the other hand, a syndromic surveillance system detecting ILI may target a wide range of indicators, any of which will be noted at low levels. Because ILI is seasonal, properly designed syndromic surveillance must account for the cyclical background activity. If an outbreak of ILI is in progress, the elevated syndrome activity becomes the background in monitoring for an attack with a biological agent associated with ILI symptoms.

Biosurveillance is an instance of *anomaly detection*, in which steady-state, historical, or cyclical trends in the background data of interest are monitored for departures from the normal state, which may be signals of concern. In syndromic surveillance, we are interested not simply in outliers in the data, but in anomalous patterns with a profile possibly indicative of an outbreak. Among numerous inputs to syndromic surveillance systems, those most frequently incorporated are: "chief complaints" from hospital emergency room visits, emergency medical service (EMS) calls, sales of over-the-counter (OTC) medicines, and absentee records, primarily from schools. The data are collected automatically under networks operated by local and national health authorities.

1. Methodology

There are a variety of approaches to syndromic surveillance. One is to think of syndromic surveillance as an instance of statistical process control, a quality control methodology initially designed to detect industrial process aberrations in early stages (Shmueli and Fienberg 2006). This analogy only goes so far: in contrast to typical industrial applications, syndromic surveillance data collection typically involves non-specialized sensors, the processing of the data is near real-time, high false-positive rates are tolerated, and extensive human analysis is involved when the system generates a signal.

Following quality control thinking, one method for univariate syndromic surveillance is to set an alarm when an indicator variable crosses a threshold. The sample mean and sample standard deviation have both been used to set alarm levels (Shmueli and Fienberg 2006; Fricker 2013). If the distribution of the indicator is assumed normal, the level for detecting a change in mean might be initially set at three standard deviations from the mean, $\mu \pm 3\sigma$. The level is fine-tuned to optimize timeliness (the speed of identification of a true positive), specificity (the rate of false positive identification), and sensitivity (the probability of successful detection of an outbreak). The frequently used approaches to univariate analyses include time-series methods (e.g., weighted moving averages) that explicitly acknowledge the correlation and seasonality in the data and regression method.

Although syndromic surveillance tracks multiple indicators, there is not a great deal of work on how to analyze the indicators jointly. A straightforward multivariate

[3] http://www.acha.org/ILI_Project/ILI_case_definition_CDC.pdf.

approach is to weigh the counts of different variables, yielding an effectively univariate system for which an appropriate alarm level can be determined (Wong et al. 2005). However, this approach is not particularly well-suited to discovering predictive groupings of factors.

Wong et al. (2005) develop an approach (What's Strange about Recent Events, WSARE) that is applicable to discrete variables. Quoting, "The basic question asked by all detection systems is whether anything strange has occurred in recent events. This question requires defining what it means to be recent and what it means to be strange" (Wong et al. 2005, p. 1964). Using their terminology, define a *rule* as a specification of a particular subset of observations. An example of a rule might be the count of emergency room admissions for a particular gender and a particular age group with a particular complaint. Rules are calculated for a baseline time period and then again for recent (here today's) data. The method systematically searches all one-variable rules to find those that have the most significant differences between baseline and current. Greedy searches identify two-variable through k-variable rules.

Different versions of WSARE have used different methods to characterize the baseline. The first method to build a baseline used records from the same day of the week from 5, 6, 7, and 8 weeks previous. A second method chose all previous records with the same values of environmental conditions as today's data. A third method learned a Bayesian network using environmental variables as predictors. The network was then used to predict what was expected as baseline given today's conditions.

2. Discussion

Current research in syndromic surveillance is directed toward expanding the monitored geographical regions, networking localized surveillance systems, improving the probability of true positive detection, and dealing with the inevitable false positives. One challenge is detection of a biological attack in the midst of an outbreak. For example, influenza contagion is cyclical, and flu outbreaks common. Numerous potential bioweapons, including anthrax and Ebola, exhibit some subset of ILI symptoms at some stage of infection (MacIntosh 2004). Broadly, the research agenda is to discover methods to use multiple heterogeneous data streams to identify patterns that are temporally, spatially, demographically, or symptomatically anomalous. Note that the Algorithms for Threat Detection Program (Chapter 5) funds a small amount of work in syndromic surveillance that focuses on identifying spatial-temporal anomalies (Zou et al. 2011).

F. Tracking

Tracking is ubiquitous. Our eyes track moving objects constantly, and automated tracking systems are becoming more prevalent. Three reasons are important to consider tracking:

1. **Tracking applications cover the full range of a number of interesting dimensions** – Algorithms exist for tracking objects moving faster than the speed

of sound and at speeds barely noticeable over a century. [For example, see Vicente et al. (2010).] There are tracking algorithms that must incorporate a tremendous amount of data in a fraction of a second, and algorithms that have days to process a handful of bytes. Some algorithms are optimized for easy problems, such as tracking a single, high-contrast object, and other algorithms are optimized for difficult problems where multiple objects need to be tracked concurrently while they fade in and out of detectability.

2. **Tracking is applicable to a very broad range of sensor modalities –** EO (visible light) cameras are analogous to human eyes, so one can easily conceptualize tracking with them. IR and ultraviolet (UV) cameras can be used to do tracking in a very similar way, as can more exotic devices such as X-ray and gamma-ray imagers. But it is also common to do tracking with radar, both the pulse-Doppler and moving target indicator (MTI) types. It is uncommon, but possible, to track with synthetic aperture radar (SAR). Sonar and other acoustic sensors do tracking, as do seismic sensors. A variety of electronic intelligence (ELINT) and signals intelligence (SIGINT) sensors can be used for tracking as well. Many fields have also developed tracking sensors specifically for their application, ranging from bee radar [radar for tracking bees, Osborne et al. (1999)] to wire chambers for particle physics.[4]

3. **Tracking is an excellent testbed for multi-sensor fusion –** Many varieties of fusion can be examined and tested on a well-defined problem; for example, consider tracking vehicles with multiple EO/IR sensors at different view angles. Is it better to fuse at the data level and fit a combined track or to do tracking for each sensor individually and fuse at the track level? How detailed do the models need to be in order for the tracking and fusion to be effective?

Tracking is directly relevant to the CBRNE mission. The proactive hunt for threats involves following suspect individuals, precursor material, suspected transport vehicles, etc. All of this requires tracking, done either by human analysts or by automated systems. The data can be as sparse as border-crossing information or as dense as high-resolution, video-rate motion imagery. The timescale for processing the data and making a decision can range from seconds to years. Once a threat is known to exist, its location must be tracked, although the optimization of the tracking algorithm is likely to be quite different from proactive data gathering. The core of plume modeling is – in essence – tracking, and, in situations when low resolution is acceptable, complicated plume modeling can be replaced with simple trackers. For DTRA's mission of feeding information back to troops in the field, it is essential to know where the troops are located. If reporting latencies are small compared to an individual's motion, then "current" location data are sufficient. However, when latencies are not small, it is

[4] Georges Charpak won the Nobel Prize in 1992 for his invention of the multi-wire proportional chamber. See http://www.nobelprize.org/nobel_prizes/physics/laureates/1992/charpak.html.

necessary to track individuals and predict where they will be when the information reaches them. This requires not only quality tracking, but also a good predictive model. In general, any predictive task will require tracking, unless it can be done on a purely statistical basis.

In the following sections, we consider tracking through the lens of differing requirements. To do this, we develop two contrasting examples: experimental particle physics and observational astronomy. Although these areas have a number of similarities, they also have significant differences and will display some sharp contrasts. It should be noted, however, that these are broad fields and are not always amenable to generalizations.

1. Tracking in Experimental Particle Physics

Modern experimental particle physics experiments use a variety of detectors to try and elucidate all information about all the particles that participate in an event. Each detector provides some information, and the information is combined to determine what actually happened in the event. A critical portion of this information for many experiments is tracking of charged particles in a magnetic field. Not only does the track give the particle's position, but also the curvature in the magnetic field gives the particle's charge and momentum.[5] For large experiments, this is typically done with a multi-wire proportional chamber or a similar drift chamber. In recent years, silicon-based vertex detectors (both pixel and strip based) have become common and are fused with the wire chamber data to do precision tracking.

a. Track Initialization (a.k.a. Detection)

The first stage of tracking is called track initialization. This is analogous to the detection stage in an ISR collection and requires a cluster of data that has a very high probability of being part of a track and a very low probability of being simply noise. In experimental particle physics, this is generally done in a wire chamber [for example, Charpak (1978)], where the ionization created by a charged particle's motion is collected on a set of parallel wires (held at high voltage) and read out. The location of the wire gives two dimensions of the particle's position, and, by reading out both sides of each wire, the third dimension can be calculated. In some types of wire chambers, the wires are grouped together into cells that can be processed together and easily clustered to initiate tracks.

b. Track Fitting Within a Detector (a.k.a. Modeling)

After the initializing cluster is found, more data are added to it in an attempt to assemble a coherent track. In general this is done within a single detector, so the modality of the data is the same and uncertainties in the data are identical (or at least

[5] Technically, it gives the sign of the charge and the product of the momentum and the magnitude of the charge. Since most particles are singly charged, this is usually a moot point.

similar). The process of track fitting requires the use of some kind of model, in other words, an estimate of how a track should behave given a set of initial conditions. Many of the models used for wire chambers are quite simple: a charged particle moving through a vacuum in a constant magnetic field will follow a helical path. Wire chambers are designed – sometimes at great effort – to approach these ideal conditions of negligible material and constant magnetic field. Given an initial cluster from a cell of wires, the helical path can be predicted forward and backward from the cell, assuming a particular charge and transverse momentum. If another cluster is found at the predicted position, it can be added to the track. This entails both the bookkeeping that prevents it from being assigned to multiple tracks and the re-fitting of the helical path to better estimate the charge and transverse momentum of the particle. This process is repeated until the track exits the detector. Then another (unused) cluster initializes a new track and that track is fit. This continues until all the high-quality clusters are assigned to tracks. Low-quality clusters can be called noise and subsequently ignored. The dividing line between high- and low-quality is a compromise between tracking efficiency (finding all the true tracks) and tracking quality (avoiding assigning noise or spurious hits to true tracks or generating false tracks).

c. Track Fitting Across Detectors (a.k.a. Multi-Sensor Data Fusion)

The initial track fitting has an advantage because each hit not only represents the same physical phenomenon, but also the uncertainties with that hit are identical or very similar to other hits that may be assigned to that track. Once other detectors become involved, the physical phenomena may be different and the uncertainties are likely to be different. For example, in the wire chamber discussed above, the position of the wire (r and φ in cylindrical coordinates) usually has a small uncertainty (millimeters) but the distance along the wire (z) usually has a much larger uncertainty (centimeters). A silicon-based vertex detector often has all three dimensions with much smaller uncertainties (10s or 100s of microns). This is often obtained by trading off against higher noise or clutter.

In addition, the model used almost always becomes more complex. For a simple wire chamber, the gas and wires are low enough in density to be ignored and the magnetic field can be treated as constant in the interior of the detector. However, most wire chambers are inside strong metal cans,[6] so both approximations are invalid at the transitions from the wire chamber to another detector. In order to perform multi-sensor data fusion, the model must incorporate what happens to the particles at these transition points. Other than the difference in model and uncertainties, the process of track fitting

---

[6] The outside container needs to be air-tight so that the gas mixture used to generate ionization does not become corrupted by oxygen or other contaminants. Also, the wires are strung at high tension to prevent sagging. This creates a tremendous force on the endplates, and the containment vessel needs to be engineered to support this without significant deformation. Aluminum cylinders are quite effective and are also inexpensive. Carbon-fiber-based cylinders have less interaction with the particles that traverse them, but they are significantly more expensive and require a higher level of engineering expertise.

across detectors is not much different than within one detector. Tracks are projected (using the model) to the next detector. If hits near the expected location are found, they are added to the track and the track is refit. This continues for all tracks and all tracking detectors.

The above description is an example of *data fusion*, where the data from the detectors are fused with minimal pre-processing. Another option would be *decision fusion*, where each detector finds tracks on their own, and then the tracks are fused together without further reference to the hit data.[7] In general this is not done in particle physics detectors, since the silicon vertex detectors cannot make sufficiently high quality tracks on their own, but the wire chamber tracks are sufficiently high quality to project into the vertex detectors and locate the associated hits.

d. Model Corrections (a.k.a. JDL Level 4 Fusion)

As is always the case, models need validation and verification (V&V), which generally leads to corrections. In the JDL fusion model (Chapter 2), this falls under Level 4 fusion (process refinement). For a large particle physics experiment, this includes several different types of tests. One is to have a very detailed model of the entire detector and all relevant physics within a Monte Carlo event generator and simulator (MC). By running millions of events in the MC, it is possible to compare statistical quantities (such as the number of hits per track for a specific type of track) between the MC results and the real results. Agreement between the sets of results validates the understanding of the detector, fusion, and processing. Another method is to use high-energy particles (such as cosmic rays) where the tracking uncertainties due to multiple scattering are known to be small. These can be used to verify estimates – for example, estimates of the z uncertainty for each wire in the wire chamber.

2. Tracking in Observational Astronomy

Observational astronomy studies bodies outside our planet using a variety of telescopes and detectors attached to these telescopes. While the objects can range from our moon and neighboring planets to other stars in our galaxy to the most distant galaxies known, the techniques tend to be similar.

Optical astronomy generally uses silicon-based pixel sensors (normally charge-coupled devices, CCDs) that are similar to those now used in consumer electronics. Ultraviolet astronomy generally also uses CCDs, although the optics are quite different and the CCDs are prepared differently. Infrared astronomy generally uses different solid-state pixel detectors (such as those based on HgCdTe), but the data look quite similar once read. In particle physics, emphasis is on maintaining the detector in a constant state

---

[7] More properly this is a ***hybrid*** system where the underlying detectors in each system (e.g., wires within a wire chamber) are fused together with data-level fusion, and the systems (e.g., wire chamber and vertex detector) are fused together using decision-level fusion.

so that valid statistics can be kept over long periods of time (in some cases, years). Thus individual sensors are not put into different collection modes on demand. Instead, any changes to the configuration are discussed and vetted carefully to understand the long-term effect of those changes. In contrast, astronomy has relatively few large-scale surveys where the configuration must be kept constant. Instead, most data are taken over the period of a few hours (at most), and the next night's configuration may be very different. In part this is because the atmosphere is constantly changing, and the corrections made to compensate for atmospheric changes can generally compensate for detector configuration changes as well.

The detection process is nearly identical to the process used for tracking in particle physics, although typically only two dimensions are used since range is so difficult to measure. Peak finding and clustering algorithms are typical for detection and many are available freely in the community.[8] In astronomy, everything is moving relative to each other at all timescales, so defining a reference frame and calibrating to it is a constant battle. There are a few standard frames, and there are standard tools in the community to do the astrometric calibration once objects are detected.

The models used for tracking in astronomy tend to be simple. Common astronomical tracking models are straight line (free body) and elliptical (orbit around one massive body). There are also models for perturbations around an ellipse (e.g., planetary motion in the vicinity of massive planets) and orbits around multiple bodies of comparable mass.

A typical tracking problem would be searching for asteroids in a series of astronomical images. If the images are taken days apart and the orbital periods are expected to be many years, then the free body approximation is valid. There are two primary methods used in asteroid searches:

1. The objects in each image are detected and bright objects are used to calibrate the location of the image. Objects whose position does not change are ignored, and the "movers" are searched for reasonable velocities and directions. When movers in two images appear to fit the model, the track is projected to another image and that image is checked for the mover. Any mover that fits the model in three distinct images is a strong candidate for follow up. This method can be entirely automated so it is useful for large surveys.

2. Each mover has a relatively small local image (a "chip") created from each original image, and each is aligned and scaled so a person can flip between these chips. Asteroids will tend to move in a straight line while maintaining constant brightness. Asteroid 38628 Huya (2000 EB_{173}) was discovered this way (Ferrin et al. 2001).

---

[8] For example, *daofind* in the IRAF package at http://iraf.noao.edu/

In astronomy, the timescales are much longer and the data rates are much smaller so there is much less emphasis on processing speed and volume. In the asteroid-finding example, the data are taken over several days to weeks and comprise megabytes to a few gigabytes of images. If the processing takes a few more days, it is of little concern. Other astronomical applications can have timescales of centuries (Vicente et al. 2010). As discussed previously, typical particle physics events are spaces fractions of a microsecond apart, and decisions need to be made at those rates. Therefore data triage is critical and processing speed can be the limiting factor.

3. Evaluation Dimensions

a. Decision Speed

For any deployed system, the speed at which decisions need to be made is a critical requirement. In experimental particle physics, it is not unusual that decisions must be made in millionths of a second (μs). For example, the CDF level 1 trigger (Section 3.B) must make decisions with 4 μs after each beam crossing to determine if that event should be processed further and potentially recorded, or if it should be discarded immediately to make room for more interesting events. The decision speed requirement drives other requirements since certain types of data taking and processing simply take too long to be used for these decisions. In the case of CDF, the readout electronics closest to each detector was required to have a built-in pipeline that could handle the 4-μs decision time. If the decision time could be made shorter, these electronics could be smaller and cheaper.

b. Precision

Another critical dimension is the precision or accuracy that the decision must have when it is made.[9] Often speed and precision are related. In experimental particle physics, it is common to do imprecise ("quick and dirty") tracking fast enough that it is available to the trigger system. Later, events that are saved will have the tracking rerun to get precision results when more time is available.

As discussed in the previous section, trigger logic must be done quickly and cannot always wait for a full tracking reconstruction. Hence, there are algorithms that do rapid tracking (generally with a subset of the data) and accept lower precision. These rapid algorithms stand in contrast to precision tracking algorithms, which use all the data to fully determine the track at the highest precision possible and accept the length of time and computing power that it takes to do this. If many tracks are present in the detector, it may be necessary to use a multi-hypothesis tracker and prune the hypothesis after the tracks have been built.

---

[9] Although they are treated the same in common usage, *precision* is a measure of repeatability and *accuracy* is a measure of agreement with a standard.

c. Signal-to-Noise Ratio (SNR)

Another dimension has to do with how strong the target's signal is and how easily that target's signal can be distinguished from everything else. The "everything else" includes pure noise, clutter (real objects, but not targets) and confusers (real targets, but not the target of interest). (Sometimes this is referred to as signal-to-noise even when the clutter or confusers dominate.) Experimental particle physics usually operates at high SNR so targets (particles) are easy to detect above the noise. The clutter is usually low, but there are many confusers (see Figure 4-5). In astronomy, not only are there (nearly) always targets at low SNR, but usually the density of targets increases exponentially as the signal strength decreases. As such, astronomers are constantly fighting to lower their detection threshold and work with fainter targets.

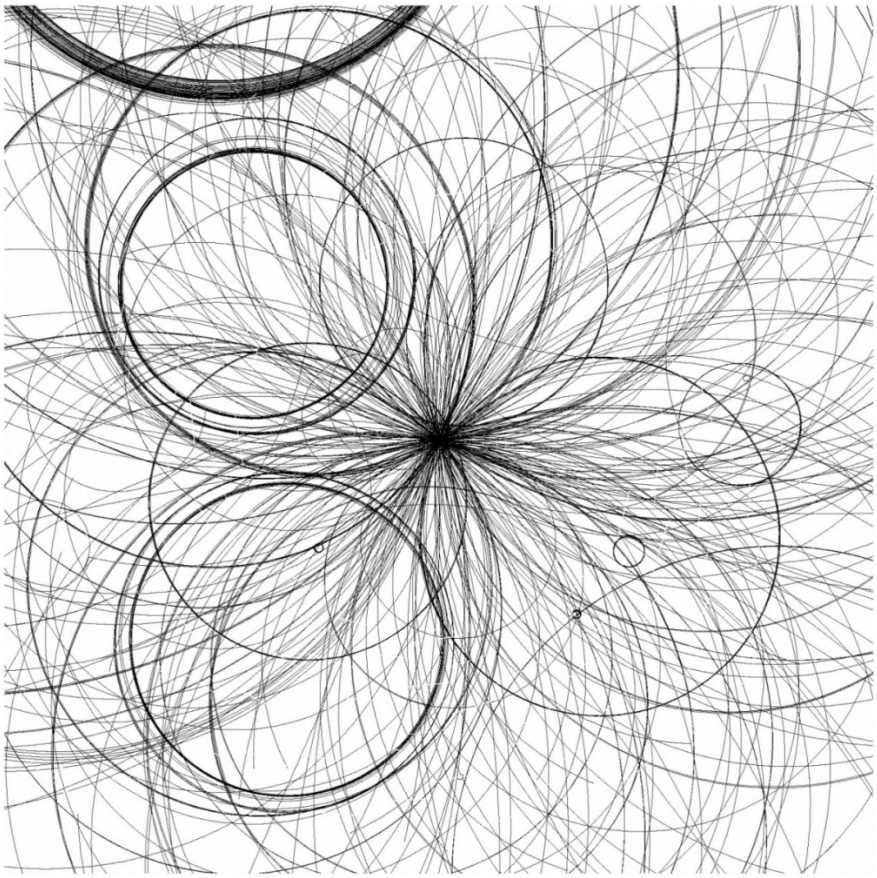


Figure 4-5. Typical Set of Tracks from the CMS Detector at CERN (Simulated)
There are four straight tracks from a Higgs decay. Can you find them?[10]

d. Data Volume

How much data needs to be processed is an important quantity, and it can be related closely to decision speed. In the discussion of data triage (Section 3.B), we pointed out that the data must be processed as fast as they are received or else the system will begin

[10] https://cms-docdb.cern.ch/cgi-bin/PublicDocDB/ShowDocument?docid=6236.

to fail. This is a critical point that we have discussed under the decision speed dimension. Additional considerations only have to do with the volume of data. Despite vast increases in the density and reliability of data storage, it is still possible to take data so quickly that the physical data storage is a problem. In some cases, data are taken and sent directly to an archive with no processing, no assessment of the value of the data, and no attempt to find metadata that would enable automated search. As a result, one winds up with rooms filled to the ceiling with hard drives and tapes – with no idea what is on any of them and no reasonable way to find out. This kind of data overload is not amenable to human filtering either (Shanker and Richtel 2011).

In observational astronomy, it has historically been the case that scientists are data limited, so they can save all of their data forever without causing storage problems. Recently astronomy experiments have entered the realm where saving all the data becomes a serious drain on the budget. Experimental particle physics passed this milestone long ago, and the field resigned itself to discarding uninteresting data and limiting the amount of data saved to disk or tape. The design of new particle physics experiments contains a significant amount of effort to budget the amount of data storage needed and plan on how to appropriately split the measured data in keep and discard streams.

G. Hurricane/Storm Track Forecasting

Hurricane track prediction is used to determine what people and property are at risk from the effects of a hurricane, so property may be secured, supplies may be better positioned, and people may be evacuated before the storm hits.[11] New forecast predictions are issued every 3 hours and are based on the statistics of past storms and physics-based simulation models (most of which take longer than 3 hours to run). The physics-based simulation models generally use the predictions from the last forecast step as the initial conditions for the next forecast step, because they simulate a large area that is difficult to measure directly.

Hurricane track prediction methods use data from multiple sources to constrain the model predictions. The National Oceanic and Atmospheric Administration (NOAA) Hurricane Hunter aircraft fly through hurricanes and collect pressure, wind speed, humidity, temperature and radar data from the inside of the storms. They also use dropsondes and collect environmental data at several altitudes after the dropsondes are deployed. Ground-based radars indicate the amount of water contained by the oncoming storm. Since the data collected are not comprehensive enough to provide the entire set of initial conditions for the track prediction models, the models are run in such a way to infer what set of initial conditions for the model would be most consistent with the few observations that were available.

[11] http://www.nhc.noaa.gov/aboutintro.shtml.

1. Models

The National Hurricane Center (NHC) publishes a table[12] that shows some of the models used for hurricane forecasting. It is interesting to note that some of the models are based on the statistics of past hurricanes' behavior; some use physical parameters inferred from real-time physical modeling to help improve their predictions ("statistical-dynamical"); and some simulate dynamics in only one layer in the atmosphere ("single-layer"). The table also shows which models allow a new forecast within the 3-hour window between hurricane forecasts ("E" for in-time vs. "L" for later) and which predict both the hurricane track and hurricane intensity. "Interpolated" models adjust a forecast that was based on earlier data from a slow, dynamic model so its track and intensity are consistent with the observations of the hurricane at the last forecast time, and "consensus" models combine the predictions from other models (sometimes removing biases in those models) to produce a composite prediction.[13] In addition, much research is also performed with the newer Hurricane Weather Research and Forecasting models (HWRF) and experimental Advanced Research WRF model (ARW-WRF).

Several groups are interested in the forecasting of hurricanes, including NOAA's National Centers for Environmental Prediction (NCEP), Atlantic Oceanographic and Meteorological Laboratory (AOML), Geophysical Fluid Dynamics Laboratory (GFDL),[14] and Environmental Modeling Center[15]; the National Center for Atmospheric Research (NCAR); the Air Force Weather Agency (AFWA); Naval Research Laboratory (NRL) Monterey[16]; the Central Pacific Hurricane Center[17]; and the Fleet Numerical Meteorology and Oceanography Center.[18]

2. Data

Some examples of observations that could be assimilated into forecast predictions are "upper sounding data from radiosondes and dropsondes, surface stations, ships and buoys, aircraft, wind profilers, velocity azimuth display winds, and satellite-based winds" (Kunii, Miyoshi, and Kalnay 2012); "Automated Surface Observing System (ASOS) stations, ships, buoys, and rawinsondes; the Aircraft Communications Addressing and Reporting System (ACARS); and cloud motion vectors" (Weng and Zhang 2012). Radiance measurements of storms by satellites have been assimilated using the technique 4D-Var (Lorenc 2003; Simmons and Hollingsworth 2002). GPS Radio Occultation (Montgomery et al. 2012) and wind speed data from wind farms are newer sources of

---

12 http://www.nhc.noaa.gov/verification/verify6.shtml#TABLE4.

13 http://www.nhc.noaa.gov/modelsummary.shtml.

14 http://www.gfdl.noaa.gov/operational-hurricane-forecasting.

15 http://www.emc.ncep.noaa.gov/.

16 http://www.nrlmry.navy.mil.

17 http://www.prh.noaa.gov/hnl/cphc/.

18 http://www.usno.navy.mil/FNMOC/.

data. An example of the data available from aircraft (in this case, from an NSF/NCAR Gulfstream V) is available in Table 1 of Montgomery et al. (2012).

In addition, we note the GFDL forecast model is a coupled atmosphere-ocean model (that includes the Princeton Ocean Model), and its initialization involves assimilation of ocean temperature vs. depth data from Airborne eXpendable BathyThermographs (AXBTs) (Bender et al. 2007), near-real-time satellite altimetry data, and satellite-derived daily sea surface temperatures (SSTs) (Yablonsky and Ginis 2008).

3. Assimilation Algorithms

Several assimilation methods used for general numerical weather prediction (Bouttier and Courtier 1999; Kalnay 2003; Lahoz, Khattatov, and Menard 2010; Lorenc 1986; Evensen 2009) are also used for hurricane track prediction: variational methods (e.g., 3D-Var), Ensemble Kalman Filter (EnKF) methods, and hybrids of variational methods and Kalman filter methods. Variational methods and ensemble Kalman filter methods can be 3D, where the observations from the field are incorporated using parameters from the time of the last model prediction, or 4D, where the observations are incorporated using parameters that are updated for the time of each observation. There have been several comparisons between the various methods with different weather prediction models (Kalnay et al. 2007; Zhang et al. 2009; Weng, Zhang, and Zhang 2011), with the results being that 4D is usually better than 3D and Kalman filters have gotten better than variational methods. However, most operational weather prediction centers still use variational methods (Raeder et al. 2012).

NOAA, National Weather Service (NWS), and NCAR use a data-assimilation framework called Gridpoint Statistical Interpolation (GSI) for their forecasting,[19] which can assimilate many different kinds of data and be configured to perform 3D-Var, 4D-Var, or a hybrid variational-EnKF scheme.[20] NCEP's Global Data Assimilation System (GDAS) is an operational system that uses a hybrid scheme through GSI. A free community version of GSI is available.[21] Recently, the Hurricane Ensemble Data Assimilation System (HEDAS) using a square root EnKF was developed by AOML's Hurricane Research Division (HWD) for the model HWRF. HEDAS has shown promise (Aksoy et al. 2012) in assimilating simulated airborne Doppler radar observations in the initialization of the modeled hurricane vortex, rather than using an artificial vortex for initialization as most models do (Weng and Zhang 2012). NCAR has a flexible, ensemble-based data assimilation system called the Data Assimilation Research Testbed (DART)[22] (Raeder et al. 2012; Anderson et al. 2009).

---

[19] http://www.emc.ncep.noaa.gov/gmb/gdas/.

[20] http://www.dtcenter.org/com-GSI/users/and http://www.emc.ncep.noaa.gov/gmb/gdas/history.html.

[21] http://www.dtcenter.org/com-GSI/users/.

[22] http://www.image.ucar.edu/DAReS/DART/.

Successive correction, splines, and optimal interpolation with limited data selection are mostly useful in numerical weather prediction situations where the current observations are much more important than consistency with past observations (Lorenc 1986) and are therefore not very useful for hurricane track prediction. It has been reported (Nerger et al. 2012) that the SEIK filter (Nerger, Hiller, and Schroter 2005) family (singular "evolutive" interpolated Kalman filter, a type of ensemble square root Kalman filter) is similarly efficient to the ETKF (Ensemble Transform Kalman Filter), and that particle filter methods (van Leeuwen 2009) might be adapted to geophysical problems to handle strongly nonlinear systems that Kalman filters cannot.

4. Assimilation Algorithm Comparison

The most successful variant of these techniques is the 4D Local Ensemble Transform Kalman Filter (4D-LETKF). It limits the range of Kalman filtering (either by artificially reducing the weights or error terms themselves as a function of distance from each gridpoint), which allows reduced computation time and easier parallelization of the algorithm than similar variational techniques. In fact, in recent testing by Kalnay's group,[23] they ran the main NCEP weather model on a 32-node network with individual LETKF computations running on each node. They found that the 4D-LETKF used approximately two-thirds of the total processor time, and the weather model used one-third of the processor time.

So many details are required to obtain the best results from these general assimilation schemes that we will not be able to catalog them here. It has been observed that, as a result of the similarity of the variational methods and the EnKF methods, several of the computational tricks developed for the variational methods (e.g., 4D, variance inflation) are readily implemented in EnKF methods. EnKF methods, however, must also ensure that the choice of the number and distribution of ensemble members is sufficient to model the problem at hand. LEKF (Local Ensemble Kalman Filter) methods require the choice of how the ensemble Kalman filter will be localized and what parameters to use (e.g., the distance over which the Kalman filter should operate), but LETKF uses only the ensemble data from the local gridpoint and its neighbors.

EnKF methods can also benefit greatly from techniques that discount the initial guess for the weather system state by iteratively applying the ensemble weights from the end of the first forecast step to the initial guess and rerunning the first forecast step. One such procedure is named Running in Place (e.g., LETKF-RIP) and ensures that the prediction at the end of the first forecast step is as consistent as possible with the actual observations during the first forecast step instead of the arbitrary initial guess. For comparison, it can take 2 weeks of simulated time for a global circulation (weather) model and 3 forecast steps for a regional weather model using regular LETKF to start reflecting observations properly, but a regional weather model using LETKF-RIP can

[23] E. Kalnay, personal communication, 2013.

reflect the observations rather well after just the first forecast step. The quick spin-up of the model is very important for cases where there is not much time to wait for the model to start making realistic predictions, such as in the case of hurricane tracking (Lahoz, Khattatov, and Menard 2010). Additional precision can be had by using the same procedure for LETKF-RIP but instead iteratively varying the ensemble mean at the beginning of the forecast step period. This can be combined with the regular LETKF-RIP procedure for a much better performance than 4D-Var (Lahoz, Khattatov, and Menard 2010), which uses a similar technique.

It is important to note that one must match the accuracy of the assimilation technique and the model to the problem that is being solved. Simple test weather codes such as SPEEDY can mimic realistic weather and run quickly on a cluster of regular computers, but they are not as accurate as the more sophisticated codes the National Weather Service and the National Hurricane Center run operationally on large supercomputers/clusters for several hours. Assimilation codes used for weather prediction are not designed to be fast, just flexible and accurate. However, tricks are discussed in the literature for increasing the speed of these algorithms. LETKF can be sped up by computing the weights used to make predictions on a coarser grid than the predictions themselves, without degradation to the predictions (Lahoz, Khattatov, and Menard 2010; Yang et al. 2009). In situations where historical model errors can be compiled (perhaps using test releases of chemicals in the case of a chemical release in a city), a "dressed" ensemble Kalman filter (DrEnKF) scheme can make realistic predictions with a much smaller ensemble than regular LETKF (Wan et al. 2009). Data thinning (Weng and Zhang 2012) of radar and satellite data to reduce computational time and variance inflation (Anderson 2009; Li, Kalnay, and Miyoshi 2009) to account for a variety of uncharacterized errors are also important procedures used in EnKF schemes. Systematic Error Correction and a non-uniform localization (Anderson 2012) may also be useful to reduce the number of necessary ensemble members (or improve the forecast quality).

In Kalnay et al. (2007), a table illustrates several reasons why ensemble Kalman filter methods are preferable to variational methods, the most important of which is that they do not require an adjoint of the weather model to be created. Creating adjoint weather models (basically, an inverse that runs the weather model backwards) is extremely difficult (it is basically an art), requires intimate knowledge of the weather model, and is not possible for weather models that have any irreversible processes in them (such as precipitation or diffusion).[24] In contrast, ensemble Kalman filters are relatively straightforward to implement and work well with any model (linear or nonlinear) where the parameters defining the states being predicted have Gaussian errors (or can be transformed approximately into a parameter with a Gaussian distribution function).

[24] E. Kalnay, personal communication, 2013.

The observations assimilated using these techniques are often isolated (e.g., dropsondes) and none of the comparisons we have seen between assimilation techniques employed radiance data from satellites (which includes a lot of data). Generally speaking, it appears that only a relatively moderate number of observations can be practically assimilated using these computation-heavy techniques.

5. Important Algorithm Characteristics

The assimilation techniques for numerical weather prediction are slow because the forecasted state vector (list of parameters describing the weather system) they try to fit to the observations is huge. To describe an amorphous, extended object like a storm front, hurricane, or plume of hazardous chemicals from a release, the technique for variational methods and EnKF is to use the values of important quantities at every grid point as the parameters vary by the assimilation technique. (For example, Kunii, Miyoshi, and Kalnay (2012) used the LETKF to analyze "all prognostic variables: three-dimensional wind components (u, v, w), temperature (T), pressure (p), geopotential height (gh), humidity (qv), and water/ice microphysics variables.") Assimilating sensor data into a model describing the propagation of a hazardous chemical from a release might require far more grid points in order to give the desired resolution for the courses of action being considered, but may require fewer important quantities at each grid point (local winds, temperature, and pressure could be approximated as fairly uniform, depending on the topography). Depending on how quickly one wants algorithms to predict where the hazardous chemical went and the sensor coverage, it may be important to investigate

- How readily the EnKF algorithm could be applied in a context where the dynamical model uses some pre-computed dynamics or the state tracked by the EnKF is some small subset of the grid modeled in the high-resolution propagation code
- Whether EnKF or a better scheme could be adapted for situations where there is a lot of observational data
- What assimilation scheme is best for a very small amount of observational data for the problem at hand.

5. NSF/DTRA/NGA Algorithms for Threat Detection Program

A. Program Background

The Algorithms for Threat Detection (ATD) Program[1] is a partnership of DTRA, NGA, and NSF's Division of Mathematical Sciences. It was started in 2009 with the mission "to develop the next generation of mathematical and statistical algorithms for the detection of chemical agents, biological threats, and threats inferred from geospatial information." The program seeks to build a research community around this mission by fostering interaction between academic researchers and members of the sensor research and development community.

The ATD program solicits and funds proposals from the mathematical sciences community in two main thrust areas: mathematical and statistical techniques for genomics and mathematical and statistical techniques for the analysis of data from sensor systems. We restrict our discussion to the latter, with focus on algorithms for physical sensors and sensor networks.

The ATD program currently funds 66 awards, with 7 already completed. It also hosts annual workshops that provide a forum for scientific exchange among members of the ATD program. Three workshops have been conducted so far, and the associated briefings, data sets, and other supporting materials are available through the Algorithms for Threat Detection Data Repository.[2]

B. Community Data Sets

One of the primary methods for promoting the growth of the ATD research community has been to make datasets available to algorithmic researchers. Currently, seven of these data sets have been prepared by the sensor research and development community. Progress in the developing algorithms for working with these data sets addresses important CBRNE capability needs such as threat detection, feature extraction,

---

1 The program announcement can be found at http://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503427&org=DMS.

2 The repository is located at Colorado State University, https://grassmann.math.colostate.edu/ATD/home.html. Contact Michael Kirby (kirby@math.coloradostate.edu, 970-491-6850) for the userid and password.

imaging, and multi-sensor fusion. For this reason, ATD program management encourages (but does not mandate) that algorithm researchers use the data.

Six of the seven ATD data sets are listed below. These data are derived from sensors relevant to chemical or radiological threats. The seventh data set is an Illumina gene sequencing data set.

1. Frequency Agile Lidar (FAL) Bio Detection Standoff
2. Ambient Aerosol Background Data Set
3. Fabry-Pèrot Interferometer Sensor Data Set
4. Johns Hopkins Applied Physics Lab Data [long-wave infrared (LWIR) Hyperspectral Data]
5. Swept Wavelength Optical Resonant Raman Detector
6. Ion Mobility Spectrometer Data.

The FAL data set comes from a Research, Development and Engineering Command (RDECOM) testing program (Vanderbeek and Warren 2010). The FAL data was taken at Joint Ambient Breeze Tunnel (Vanderbeek 2013) and involves lidar scattering off of a variety of biological simulants for spores, toxins, and viruses, as well as dust, smoke, and diesel exhaust at a standoff range of 1.2 km. The FAL data has been made available in order to promote the development of algorithms that can extract range-dependent biological (or chemical) concentration profiles and aerosols of interest at standoff range. The algorithms being developed are typically applicable (or readily generalizable) to chemical clouds provided that the chemicals are not fully dissolved into the atmosphere, but have a significant aerosol of suspended chemical droplets.

The second data set is the Ambient Aerosol Background data set (Sivaprakasam, Tucker, and Eversole 2013). This data set provides spectral data taken by a dual wavelength UV fluorescence spectrometer. The system concept is to detect biological agent aerosols in the presence of interference from aromatic hydrocarbons from internal combustion engine exhaust, industrial chemicals, as well as indigenous biological aerosols, such as fungi, pollens, dander, and bacteria that may be normally present in the environment.

The dual wavelength UV fluorescence spectrometer ingests a continuous air stream at the sensor location. The air stream is interrogated by three lasers beams at distinct wavelengths that stimulate UV fluorescence. The scattered and fluoresced radiation intensities are reported in 16 separate wavelength and polarization channels. Currently, the system has difficulty in discriminating between potential threats and non-threats when applied to aerosols that fluoresce significantly. However, this lack of capability might be mitigated by improved data processing algorithms.

The third data set is Fabry-Pèrot Interferometer Sensor Data Set (2013). This is a LWIR (8-11 meters) hyperspectral imaging (HSI) data used in field testing. A simulant

is disbursed by explosive into the troposphere. The explosion partially transforms the simulant vapor phase and condensed aerosols. The goal is to detect and image the resulting cloud against a natural background. During a single scan of the Fabry-Pèrot interferometer, sensor mirrors scan the image scene such that the sensor collects a data cube. Each of these hyperspectral data cubes consists of 256×256 pixel image for each of 20 wavelengths. The Fabry-Pèrot interferometer data cube represents a snapshot of the plume evolution. The system collects successive data cubes in order to record the release event in its entirety. In Figure 5-1(a), we present a fixed-wavelength slice of a data cube. This image nicely illustrates that, in spite of the presence of an aerosol plume, no plume image may be detected by simple inspection of the unprocessed HSI data.

The fourth dataset is the Johns Hopkins Applied Physics Lab (JHAPL) Data Set (2013). The JHAPL data are also hyperspectral imaging data (Carr, Broadwater, and Limsui 2011). In this case, it is LWIR data collected from three identical sensors that have been placed a separate observation points to observe a chemical plume from different viewpoints. The system concept of this network is to enable chemical detection and identification, to determine the release location, to characterize plume transport, and to provide advisories recommending the movement of personnel. Each sensor in the network is a field-portable imaging radiological sensor [Fourier transform infrared (FTIR)]. The sensors capture 129 spectral bands at a frame rate of 0.2 Hz. In the field test, three sensors are located at separate viewpoints at standoff distances which ranged from 2.15-2.82 km from the release point. The released chemicals included three known chemicals in a gaseous or liquid and gaseous state and were to be compared against a library of five signatures.

The fifth data set is the Swept Wavelength Optical Resonant Raman Detector (SWOrRD) data set (Gillis, Grun, and Bowles 2013). These data are produced by a Raman spectrometer that illuminates a *collected* sample by a swept wavelength laser. Each illumination wavelength stimulates the re-emission of a resonant Raman spectrum, which is detected and recorded. As a result, the sample creates a 2-D spectral signature characterized by two wavelengths. The first is the wavelength of the illumination, and the second is the wavelength of the re-emitted photons. The interest in this system results from its demonstrated ability (Grun 2011) to identify biological materials, explosives, and dangerous chemical agents.

The sixth data set is the Ion Mobility Spectrometer (IMS) data (Eiceman 2013): An Ion Mobility Spectrometer is a handheld device useful for chemical and biological agent identification in the field. Currently this device has been deployed by the Army for chemical-agent monitoring. IMS works by ionizing a sample and deriving a distribution of drift times produced by the ionic fragments generated from the sample. The distribution is a signature that can be compared against known agents in order to detect and identify chemical and biological agents.

C. Algorithm Categorization

In Chapter 2, several taxonomies were described. The ATD program does not work broadly across the algorithm landscape. For example, there is no research on resource tasking and sensor management. In Table 5-1, we have categorized the award into several areas, which include fusion, detection, feature extraction, classification, environmental models, and image reconstruction (they use low-dimensional scans or data to build 2D or 3D images). We do not specifically call out image segmentation (dividing the image into regions) as an algorithm category. This omission was intentional. A majority of the hyperspectral imaging research presented at the ATD program workshops could be understood as segmenting (i.e., colorizing) hyperspectral imagery. Moreover, even some lidar research on reconstructing plume surfaces could be viewed as being a type of segmentation. So in this sense, much of the ATD research involves advanced segmentation methods. However, classification or dimensional reduction tends to be the goal of most of these algorithms. Consequently segmentation has significant overlap with classification and dimension reduction. To avoid overlapping categories and to provide more informative categorizations, segmentation was excluded from the typology.

The vast majority of ATD research projects that have been presented at the three workshops attempt to develop algorithms that support a specific ISR capability in an identifiable operational context. A small number of research studies do not allow identification of a capability and an operational context, and we will refer to these as "Theoretical Studies." Our survey of ATD workshop presentations has identified three operational contexts/vignettes: (1) chemical release/attack (2) attempted smuggling of a radiological device into secured area, and (3) targeted search for an emplaced radiological device. The first two of the events can be classified as force protection missions; the third is a targeted search mission. Wide-area search and long-term threat monitoring are not addressed by the current ATD awards.

Each of the three vignettes is associated with a set of capabilities. For instance, an examination of the "chemical release/attack" vignette identifies research supporting plume tracking, release detection, standoff chemical identification, and non-standoff chemical identification. Using algorithm classes and vignettes, one can grid the ATD study projects as shown in Table 5-1. The table provides some insights into overall structure of the ATD program. For instance, research falls into three broad categories: Detector Network algorithms, Detector Algorithms for Plume Tracking, and Detector Algorithms for chemical identification. In contrast, only a few projects support portal detection, targeted search, environmental models, or detector network optimization.

Table 5-1. ATD Portfolio by Vignette and Algorithm Type

The table is based on the oral presentations at the 2010-2012 algorithms for threat reduction workshops, excluding those on genomics. The presentations are coded in the table using the first six characters of the author's last name followed by the two-digit year.

DTRA "Scenario"	ATD Vignette	ISR Capability	Data Fusion	Feature Fusion	Decision Fusion	Compressive Sensing	Anomaly Detection	Feature Extraction	Classifi-cation	Environ-mental Models	Reconstruction
Direct Force Protection	Chemical Release/ Attack	Chemical Release Detection	Rodrig12	Tartak12b Sieg12 Bayrak12	Calder11		Tartak12				
		Plume Imaging/Tracking	Strohm11					Marks12 Ziegel10-12, Favret12 Bertoz12 Kling11-12 Bertozzi10 Esser10 Golow10 Rohrba10b	deVore12 Rohrba10a Chepus11	Xun12	Binev11 DeVore10 Sharpely10 Esedog10 Binev10
		Standoff Chemical Identification				Osher11 Osher10 Zhang11 Chin12		Warren10-12 Sun11 Greer10	Kirby11 BenDav11		
		Non-standoff Chemical Identification	Tsai12	Minor10 Minor11		Sander12	Landon11a Tsai12	Tsai12 Landon11b Yin12 Landon12 Esser11 Xin12 Gillis10	Guhar12 Pladdy12 Tucker10		
	Smuggling of Radiological Device into a Secured area	Radiological Materials Detection at Portal	Xun11								
Targeted Search	Emplaced Radiological Device	Search/Detection in Rad Material			Greylau11 Cheng11		Gordon11				
Theoretical			Casazz11-12			Fickus12	Crapa11 Gordon10 Komend11	Tsai10 Xin10 Mallic10	Osher12 Kirby10 Rohrba10a Singpu11 Arias12 Rodrig11 Marks10		Binev12

D. Overview and a Sampling of Recent ATD Research

We now present a brief tour of the ATD research topics. We begin with plume tracking. The plume-tracking process involves imaging the cloud, tracking its movement, and modeling its evolution using convection diffusion or other hydrodynamic models. Typically most of these activities require data from standoff sensors such as lidar and HSIs.

Hyperspectral sensors require processing to reduce the data cubes to 2-D image frames in a manner that reveals a plume image. This is an active research area in algorithm development (Manolakis 2010). Plume imaging has been a challenging problem, as is the more general problem of automated real-time automated processing of HSI data. There are a variety of approaches to processing HSI data (Plaza et al. 2004), many involving spectral demixing, and end-member extraction concepts. Often these algorithmic concepts are based on convex optimization techniques (Bertozzi 2010). However, alternative approaches to HSI processing and image segmentation have been discussed at ATD workshops and are based on clustering and manifold learning

algorithms. Work in this area by Ziegelmeier, Kirby, and Peterson (2011) and Ziegelmeier, Kirby, and Peterson (2012) has explored clustering and the Local Linear Embedding algorithm. Figure 5-1 displays some of the progress in HSI imaging of plumes that the group has achieved. In this figure, spectral clusters are displayed using distinct color with yellow pixels being the chemical of interest. The group has been investigating approaches to reduce the computational complexity of this problem that are based on algorithms that combine their earlier methods with L1 regularization and split Bregman algorithms.

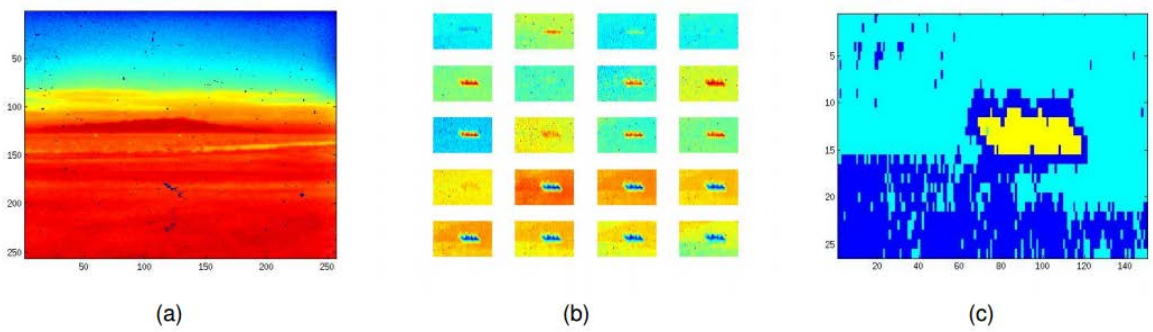


Figure 5-1. Hyperspectral Imaging Plumes

(a) One sheet of a sample 20 dimensional hyperspectral images from the FPIDS data set. (b) After preprocessing and background removal, display of each individual sheet when a chemical plume is present. (c) Image of clusters detected using BLOSSM algorithm.

Regardless of the progress that is being made with HSI imaging of plumes, the data from a single HSI sensor can still only provide a 2-D image. No 3-D depth profile of aerosol concentration levels can be constructed from a single HSI sensor. Typically, to obtain 3-D profiles, one must work with lidar data or fuse HSI data from a network of hyperspectral sensors. An example of the latter, Strohmer (2011) has been attempting to fuse the data streams from a network of LWIR hyperspectral sensors. To do this, he represents the plumes density profile in terms of a dictionary of Gaussians. With this assumption, he has studied the recovery of the density profile can from the line measurements using an "analysis L1 minimization" algorithm.

The alternative to 3-D plume profiling is based on lidar. Significant progress has been made with plume visualization (using reconstructed iso-concentration surfaces) by DeVore and coworkers (DeVore et al. 2013; Binev et al. 2010; Binev 2011). This represents a new area in lidar data processing, as older algorithms have primarily focused on terrain processing and reconstruction. Those older algorithms had flaws that degraded the faithfulness of their terrain reconstructions and prevented plume-imaging applications. Among the cited flaws were the failure to discriminate occlusions from non-returns, sampling limitations, and the failure to take into account geometric and topological structure in the data. New algorithms based on "reliable sets" enable the

imaging plumes obtained from synthetic lidar and terrain data from Air Force Research Laboratory Munitions Group (AFRL/MNG) Vision-Enabled Autonomous Agents (VEAA) Data Set from Eglin AFB (Figure 5-2). These algorithms are intended to provide capability data for both natural and man-made (e.g., urban) environments from one or more mobile lidar sensors. However, it is unknown whether there have been any "third party" validation studies. It is also not known how much further development would be required to turn these methods into an automated real-time capability.

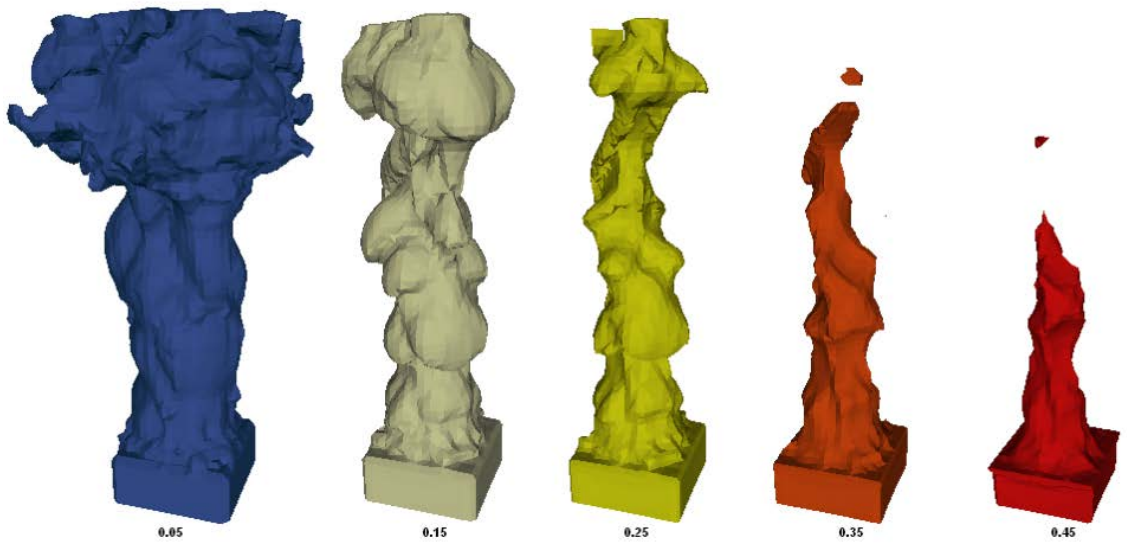


Figure 5-2. Isolevel Reconstruction from Synthetic Lidar Data (Binev 2011)

In addition to visualization processes such as isolevel reconstruction, the plume-tracking process also involves modeling the plume's evolution using convection diffusion or other hydrodynamic models. This requires a determination of wind velocities or diffusion rates and has been a focus of a couple of the ATD research projects. For instance, Xun et al. (2011) and Xun (2011) have been developing a Bayesian method to estimate these environmental parameters from lidar data.

The next set of ISR capabilities addressed by the ATD program is chemical identification. This capability comes in two categories: standoff and non-standoff chemical identification. The sensors involving standoff (e.g., HSI) are rather different than the non-standoff sensors (e.g., Raman, IMS, UV fluorescence), but, in either case, one may wish to match against a signature library of chemical agents. If that is the objective, and if the sample being studied is a mixture, then one must be able de-mix observed spectra.

Demixing (Keshava and Mustard 2002) refers to taking an spectra $\mathbf{x}_b$ where b=1, K index set of spectra bands (or data channels) and solving the Linear Mixing Model (LMM) for the concentrations $\mathbf{c}_i$ of the unknown chemical constituents:

$$x_b = \sum_{i=1}^{N} E_{bi} c_i \qquad b = 1,...K \tag{1}$$

The columns of [$\mathbf{E}_{bi}$] are the spectra of the unknown chemical constituents, called endmembers. The matrix **E** is referred to as the endmember matrix. The concentrations $\mathbf{c}_i$ define linear combinations of the constituent spectra to produce the observed spectra $\mathbf{x}_b$, b=1, …, K.

The simplest case of a demixing problem might specify a small set of chemical agent spectra as endmembers. This is called the non-blind demixing problem. It is a simple problem, and a variety of elementary methods are available such as regression analysis, least squares fitting, or Bayesian methods. An example of the latter is given in Landon (2011). More commonly, the case of a set of constituent spectral signatures is not available. In this case, one has the blind demixing or the blind source separation (BSS) problem (Xin 2010). BSS is a highly under-determined problem. It requires additional assumptions and must be attacked with advanced algorithmic methods such as L1 demixing algorithms (Greer 2010).

Demixing (blind or non-blind) is a central problem in the world of CBRNE detection and identification algorithms. Demixing algorithms can be used to identify relative concentrations of mixtures that include biological or chemical agents. In addition, they have the potential to improve the sensitivity of detection algorithms by avoiding the overlap of multiple signatures. Demixing algorithms are commonly used to process HSI imagery. More complex versions of the demixing problem (either LLM model or NMF model, defined below) can include noise, background terms that vary slowly with wavelength, mixtures where dominant constituents may tend to mask dangerous trace agents of interest, or even multidimensional spectral problem (e.g., Raman spectrum) where the band index is a two-dimensional parameter.

As mentioned above, the demixing problem (particularly blind demixing) is highly under-determined and requires additional assumptions to make it a well-defined problem. For instance, one might assume a modification of the demixing problem in which large number of observations are available and use the same endmembers. When analyzing multiple pixels in a HSI data cube, the demixing problem becomes the non-negative matrix factorization (NMF) problem:

$$X = EC \tag{2}$$

where each observation corresponds to a row of X and a column of C. In this problem, further assumptions are still required. Typically one often assumes that all columns of

the endmember matrix can be found as spectral of individual “pure” pixels, and that [C] is very sparse.

One of the many ATD program examples of research on demixing algorithms is the work of Guo et al. (2011). The study involved the processing of frequency agile lidar data taken at the Joint Ambient Breeze Tunnel Facility at the Dugway Proving Ground. The problem addressed there slightly generalizes the LLM since the concentration is range resolved. The authors approach used the split Bregman algorithm to reconstruct the concentrations given the wavelength-dependent laser backscatter coefficients, then reconstructed the backscatter coefficients using the concentrations, and repeated these two steps until convergence to a self-consistent solution was achieved.

A last example of algorithms for chemical identification, is the work by Esser (2011) on demixing data obtained from Differential Optical Absorption Spectroscopy (DOAS). His investigations deal with issues in DOAS associated with processing the background. His approach uses the convex optimization algorithms developed by Zhang, Burger, and Osher (2011).

In addition to plume-tracking and chemical-identification algorithms, there is a third area of significant ATD research on network algorithms. Most of this research is devoted to detection of chemical or biological releases, although a few algorithms are devoted to the targeted search of radiological devices. For example, Grelaud (2011) has developed an algorithm that supports the detection by a mobile network of inexpensive radiological sensors. The operation context assumed in his research involves a network of radiation sensors mounted on taxi cabs. His approach is based on a sequential Monte Carlo particle filtering approach that uses network sensor data to constantly update the probability distribution describing the likelihood of finding one (or more) radioactive source as a function of location throughout the network coverage area.

Another area of research is the development of change point detection methods. The quickest change point detection algorithms monitor one or more sensor data streams to determine if a change in the statistics of the sensor readings has occurred. These algorithms are important in a number of fields including computer network security monitoring, chemical and biological warfare agent detection, and syndromic surveillance. As part of the ATD program, Tartakovsky (2012) has developed near-optimal change detection procedures that assume an unknown change state. His algorithms do this in a manner that optimizes the tradeoff in detection delay and the frequency of false alarms. In the latest version of his work, he has included adaptive sampling techniques intended to conserve power in a remotely deployed battery-operated sensor network.

There are other examples of change point detection research in the ATD portfolio. For example, Craparo et al. (2011) considered graph-theoretic methods to find change points in general distributions. Siegmund et al. (2012) and Xie and Siegmund (2012)

have developed a mixture procedure for monitoring parallel streams of data for a change-point that affects only a subset of them, without assuming a spatial structure relating the data streams to one another. Arias-Castro (2012) develops methods for identifying anomalies that are identified by unusually large sequential correlations instead of the more typical unusually large amplitudes.

A final example of fusion algorithm research supported by the ATD program is the work by on the detection of low intensity radiation source in the presence of large background noise. Xun et al. (2011) considers how data fusion of the data stream from a network of radiation sensors, say installed at a security portal, would greatly improve detection thresholds. This work is one of a handful of sensor fusion research projects in the ATD program.

6. Joint Effects Model Sensor Data Fusion Science and Technology Efforts

The Joint Effects Model (JEM) is a system designed to provide a single Department of Defense (DoD) approved tool to predict hazard areas and effects resulting from the use of CBRNE weapons and releases of toxic industrial materials. Requirements for the Joint Effects Model have driven science and technology development efforts to incorporate sensor data fusion algorithms into CBRNE hazard prediction models. One requirement in particular, a threshold performance requirement for JEM Block II, directly requires sensor data fusion capabilities:

> JEM shall provide the ability to estimate the locations of sources based on sensor data, and to make refined dispersion calculations by incorporating sensor data with initial dispersion estimates.[1]

DTRA and the Joint Science and Technology Office for Chem-Bio Defense (JSTO-CBD) have been sponsoring science and technology (S&T) efforts and development and integration of sensor data fusion tools into a CBRNE hazard prediction model in order to provide a capability that can meet the JEM requirement.

These efforts have produced an experimental data set designed to be used in the development and evaluation of algorithms that combine sensor data and transport models to estimate the location of a source of an airborne threat. The DTRA/JSTO efforts have also resulted in development of a tool for generating synthetic data useful in the algorithm development and evaluation process and have resulted in some capability in this area.

A. FUSION Field Trial 2007

The Fusing Sensor Information from Observing Networks (FUSION) Field Trial 2007 (FFT-07) was a short-range atmospheric-dispersion field experiment designed to provide an experimental data set to support development of prototype source term estimation (STE) algorithms (Platt and DeRiggi 2012). The STE algorithms must fuse data from meteorological sensors with data from chemical-agent sensors to estimate the location of the source of a past or in-progress release of chemical agent into the atmosphere. The experiment consisted of a series of releases of a chemical tracer gas slightly upwind of a dense grid of concentration samplers. Detailed meteorological data were also collected using a large number of instruments deployed during the experiment.

---

[1] Operational Requirements Document for Joint Effects Model (JEM) ACAT III Prepared for Milestone B, 2004.

A total of 82 trials, involving a mix of instantaneous and continuous releases from up to four sources, were conducted over a period of 2½ weeks at the U.S. Army's Dugway Proving Ground. Releases were executed both during the day and at night. One hundred digital photoionization detectors (digiPIDs) were arranged in a dense regular grid in a square area approximately 450 meters on a side (Storwald 2007). Twenty ultraviolet ion collectors (UVICs) were also deployed in that area. In addition to the chemical sensors, a large number of meteorological sensors, including 40 Portable Weather Instrumentation Data Systems (PWIDS), were used during the experiment.

Although the data set was developed experimentally, the experiment was carefully designed to produce a near-ideal set of data, unlike any that would be obtained in an operational setting. For example, releases were arranged and timed so that the released material would encounter a large number of the deployed sensors. In addition, sensors collected concentration data at high frequency under near-ideal conditions. However, the data from FFT-07 can be processed to simulate more relevant scenarios, for example by using only data from a subset of the samplers and/or by processing the concentration measurements into threshold exceedance (alarm/no alarm) indicators. IDA was tasked by DTRA to compare the prototype STE algorithms and, as a major part of that effort, developed the plans and protocols for the comparison, including processing the FFT-07 experimental data set into sets more likely to be obtainable under operational conditions. Those prior efforts (Platt, Warner, and Nunes 2008) could serve as a starting point for evaluating sensor fusion algorithms (other than just STE) using the FFT-07 data set. The FFT-07 data set is maintained and distributed by Dugway Proving Ground and is readily available by request.

For the STE algorithm comparison, IDA constructed a total of 104 cases of sensor data from a subset of the available digiPID data and made those cases available to STE algorithm developers. The source location, number of sources, and size of the sources was withheld from the developers. A total of 8 different developers participated in the STE algorithm comparison, submitting a total of 14 full and partial sets of predictions to IDA for comparison to truth and to each other. The developers utilized a variety of algorithms, with some developers submitting more than one set of predictions to try more than one approach. The developers also submitted brief descriptions of their algorithms, which can be found in Platt and DeRiggi (2012). The IDA researchers concluded that STE for chemical and biological weapon attacks remains a challenge, noting that the exercise revealed shortcomings with respect to the ability to estimate the location and mass of a release even under highly idealized conditions.

B. VTHREAT

NCAR developed STE algorithms as one of the participating performers in the FFT-07 algorithm comparison exercise. During that effort, NCAR developed and made use of a testbed for chemical and biological releases. That tool, the Virtual Threat Response Emulation and Analysis Testbed (VTHREAT), is a virtual simulation environment that

enables the simulation of physically realistic hazardous-release scenarios. It allows placement of chemical, biological, and meteorological sensors in the simulated environment and allows extraction of the resulting synthetic sensor data streams (University Corporation for Atmospheric Research 2011). VTHREAT was specifically developed for DTRA, and the development effort has been sponsored by the JSTO-CBD. VTHREAT enables testing in a broad range of simulated geographic locations and environmental conditions.

Instead of being a fixed system of software components and models, VTHREAT is a toolbox containing modeling and simulation capabilities. It is a modular framework that allows the use of a variety of atmospheric transport and dispersion (AT&D) models, sensor emulators, data fusion algorithms, and response models selected for the specific needs of the end application.

Typically, the VTHREAT system uses the EUlarian semi-LAGrangian (EULAG) model for geophysical flows to simulate atmospheric turbulence in the planetary boundary layer at micro beta (100 m) and gamma (10 m) scales (Bierberbach et al. 2010). This type of model is also known as a Large Eddy Simulation (LES). A Lagrangian particle dispersion model has been developed to utilize the resolved velocities and variances from the LES to simulate the atmospheric transport and dispersion of passive particles, generating physically realistic single realizations of chemical or biological releases (in contrast to the ensemble averaged plumes typically generated by AT&D models). Those realizations are then converted to observations through models that emulate the characteristics of sensors. The sensor models include detailed information on sensor performance characteristics obtained during developmental or operational testing of the sensors. The sensor output is produced in the same format created by the sensor being modeled and with comparable fidelity.

The sensor models currently available in VTHREAT include both point and standoff sensors. The point sensors include a basic point chemical sensor that emulates the bar levels of the automatic chemical agent detector alarm (ACADA); the Joint Biological Point Detection System (JBPDS); raw point concentration readings that can be used with knowledge of detection limits to emulate sensors such as the digiPIDs and UVICs used during the FFT-07 releases; point meteorological sensors for winds, temperature, and humidity; and several testing assets used by Dugway Proving Ground. The standoff sensors available in VTHREAT include the Joint Lightweight Standoff Chemical Agent Detector (JLSCAD); the Raman-shifted eye-safe aerosol lidar (REAL); and raw path integrated concentration values that can be used as an input to an infrared camera type detector.[2]

[2] Bieringer, P., 2013. Personal Communication (email) with K. Papadantonakis.

In addition to its use in support of STE algorithm development, VTHREAT has been used to assist field experiment design. It has been enhanced to support operational test and evaluation of sensor systems and to provide inputs to toxic load effects models.

7. Looking Forward

In this report, we have described data-fusion algorithms and identified the common approaches for each. In addition to providing a range of illustrative case studies, we have provided an overview of two current DoD programs (Algorithms for Threat Detection and Joint Effects Model) that have been supporting algorithm development.

In this final section of the report, we identify potentially impactful research topics and suggest approaches for providing further analytical support to a research and development program for real-time sensor and model data fusion algorithms.

A. Opportunities for Research and Development

Three areas of algorithm research appear to be potentially useful next steps for the DTRA reach-back capability goals. The first two are closely interrelated and are *anomaly detection* (Section 3.E) and *data assimilation* (Section 3.D). Anomaly detection algorithms are classified in two ways: by anomaly type (point, contextual, or collective) and by the approach to defining "normal" behavior (data-based with classification algorithms, data-based with change-point algorithms, explicitly modeled). Most anomaly detection research has focused on point anomalies, but the CBRNE problems will require methods for detecting contextual and collective anomalies. Currently fielded CBRNE systems use a data-based definition of "normal" and change-point detection to find point anomalies; the systems of the future will use physics-based models and data assimilation methods to explicitly define "normal" behavior and then detect contextual and collective anomalies. These approaches – particularly in the CBRNE context – still require development. In particular, there are challenges in (1) understanding the statistical properties (average behavior and variation) of complex models of "normal," (2) developing appropriate data assimilation methods to integrate observational data into these models of "normal" and update the current state, and (3) developing appropriate statistical hypothesis tests to detect collective and contextual anomalies using the "normal" models as the null hypothesis.

The third key algorithmic area is *sensor management*. For specific problems, real-time algorithms that integrate data and models are becoming available (for example, in numerical weather forecasting). This suggests that, for well-understood CBRNE problems with clearly stated objectives and metrics, progress could be made on developing similar real-time algorithms.

B. Exemplar Data Set Development

There are at least two approaches to developing data sets for algorithm development and testing purposes. In one approach, the data sets are derived directly from data

collected during field experiments. This approach was used for the FFT-07 algorithm development and comparison exercise. In the second approach, systems that integrate environmental, transport, and sensor models can be used to generate synthetic data sets. Each approach has its own set of advantages (and of course, disadvantages).

Experimental data sets have the advantage of being firmly rooted in truth. However, experiments can be expensive to perform and difficult to plan and execute successfully. The "truth" of the experimental data set may reflect a scenario that has been designed to provide an ideal, rather than operationally realistic, data set. For example, consider the case of the FFT-07 experiment. The experiment was designed so that the chemical released could be readily detected in a relatively pristine environment offering no challenge of background or clutter. The releases were sized, located, and timed so that winds were constant at the time of release and would transport the chemical over the dense grid of samplers in a plume or puff of a physical size that was just large enough that an ideal number of simultaneous detections would be measured as the chemical passed through the detector array.

Experiments are (and ought to be) carefully designed to produce useful data sets. Even the most rigorously controlled experiments offer the advantage of possibly producing unexpected results and thereby increasing knowledge of the underlying physics or behavior of systems or components. (Indeed, without rigorous controls, it may be impossible to ever understand an unexpected result.) It may be impossible to gain new scientific knowledge any other way. Unfortunately, it is practically impossible to perform enough experiments and replicates to produce a data set representative of the range of CBRNE missions of interest to DTRA.

Where environmental, transport, and sensor models are available and of reasonable fidelity, they can be used to produce synthetic data sets, which may be an adequate approximation of truth. It is important to note that models used to produce predictions may not be appropriate or ideal choices for components of a system designed for synthetic data generation. For example, a model that produces an averaged result rather than a set of individual realizations may produce synthetic data that lacks the magnitude and effect of important stochastic processes. Synthetic data generation systems offer the advantage of allowing the production of data sets for a practically unlimited range of scenarios. They also offer the opportunity to explore scenarios involving notional components, such as detectors with improved sensitivity or selectivity that might exist in the future.

The FFT-07 data set and the VTHREAT system are existing sources of experimental and synthetic data readily available to DTRA that could be leveraged to provide exemplar data sets for sensor data fusion algorithm developers. These data sources, however, are primarily of use for the chemical detection mission area. Although they may have some limited applicability to biological detection, they have no applicability to the detection of radiological or nuclear materials that have not been dispersed into the atmosphere. We have learned of a DTRA-owned system that may be

capable of generating synthetic data of adequate fidelity for the radiological or nuclear detection missions. If experimental radiological and nuclear detection data sets, of comparable quality to that for FFT-07 exist, they can be leveraged for algorithm development purposes. Thus we suggest that DTRA consider developing sources of both synthetic and experimental data that could be advantageously used for developing and testing sensor data fusion algorithms for the radiological and nuclear detection missions. We also suggest that developers be provided with data from multiple sources to help prevent diversion of their efforts from true exploitable phenomena by quirks or singularities latent in a particular data source.

Upcoming DTRA-sponsored exercises may provide opportunities to collect experimental data sets. We suggest reviewing, and revising where possible, the plans and protocols for those exercises with an eye toward enabling the collection of an experimental radiological or nuclear detection data set suitable for use in sensor data fusion algorithm development.

C. Additional Resources for Algorithm Development

In addition to providing data sets, we also suggest that algorithm developers be provided with a written set of descriptions of the range and type of scenarios to which the algorithms might be applied. The mission descriptions provided earlier in this report might provide a starting point for such material if it has not already been assembled. The descriptions will help developers better understand the problems that they are being asked to help solve.

We also suggest assembling a documented collection of models that could be made available to algorithm developers. The expertise of the developers will not include all of the physics of all of the scenarios and components for which they are developing fusion algorithms, but the physics will play an important role in the development. The goal of providing such a collection would be to give developers a head start on ways to represent the physical science required as a component of their work.

The JEM program has driven some efforts toward development and integration of sensor data fusion capabilities for CBRNE missions involving the atmospheric transport and dispersion of hazardous materials. The JEM science and technology efforts are guided by the program Operational Requirements Document. We suggest developing a documented strategy for a future integrated sensor and modeling system for the nuclear and radiological detection, interdiction, and search type missions as it may prove useful for developers to understand fully the context of their effort.

We have observed substantial efforts to simulate detectors and to integrate real-time sensor locations and measurements with mapping systems. It is, therefore, apparent that a definite foundational effort has been directed toward addressing the needs of these missions. A document fully describing a comprehensive vision of future reach-back capabilities that integrate real-time sensor data and model predictions for the radiological

and nuclear detection and search-type missions could prove useful in communicating with and setting goals for algorithm developers.

We also suggest that, as algorithms are developed, very detailed technical documentation be included. The goal is to reduce any disconnect between model developers and model users. In particular, we suggest documenting "capabilities and limitations" to include the equations employed, the numerical methods that are used, the physical regimes under which approximations are valid, all of the inputs and outputs in different modes of operation, and the conditions under which one sub-model is run versus another (including hidden "switches" between operational modes, etc.).

D. Model Validation

Model validation efforts will be critical to the success of an analytical tool that uses a combination of real-world sensing technologies and physics-based models. It is possible to proceed with algorithm development using synthetic data, and this can be a reasonable path forward, provided the models producing the data have been extensively validated. In the absence of experimental data and thorough model validation, however, the system may be unable to perform its task in the real world. For example, if an algorithm depends upon successful detection of anomalies, and a model has been used to represent the normal signal from the sensor network, errors in the model can cause a complete failure of the detection scheme.

Validation efforts provide a means for understanding the uncertainty and bias in predictions produced by models. This understanding is critical to the proper use and interpretation of model results as they contribute to the reach-back analytical toolset.

Appendix A
References

Aksoy, A., S. Lorsolo, T. Vukicevic, K. Sellwood, S. Aberson, and F. Zhang. 2012. "The HWRF Hurricane Ensemble Data Assimilation System (HEDAS) for High-Resolution Data: The Impact of Airborne Doppler Radar Observations in an OSSE." *Monthly Weather Review* no. 140:1843-1862. *This paper gives an example of the implementation of a square root EnKF algorithm designed to work with the forecast model HWRF and tested on the case of Hurricane Paloma (2008).*

Albaker, B., and N. Rahim. 2011. "A Conceptual Framework for a Review of Conflict Sensing, Detection, Awareness, and Escape Maneuvering Methods for UAVs." In *Aeronautics and Astronautics*, edited by M. Mulder. Rijeka, Croatia: Intech.

Anderson, J. 2009. "Spatially and Temporally Varying Adaptive Covariance Inflation for Ensemble Filters." *Tellus A* no. 61:72-83. *This paper describes adaptive variance inflation, which uses tailoring of the inflation method to different kinds of observations*.

———. 2012. "Localization and Sampling Error Correction in Ensemble Kalman Filter Data Assimilation." *Monthly Weather Review* no. 140:2359-2371. *This paper describes a different approach to localization in EnKF schemes than ignoring observations that are further away than a certain distance.*

Anderson, J., T. Hoar, K. Raeder, H. Liu, N. Collins, R. Torn, and A. Avellano. 2009. "The Data Assimilation Research Testbed: A Community Facility." *Bulletin of the American Meteorological Society*:1283-1296. *This paper gives a general description of the DART facility.*

Anikeev, K., G. Bauer, I. Furic, D. Holmgren, A. Korn, I. Kravchenkol, M. Mulhearn, P. Ngan, C. Paus, A. Rakitine, R. Rechenmache, T. Shah, P. Sphicas, K. Sumorok, S. Tether, and J. Tseng. 2000. "Event-Building and PC Farm based Level-3 Trigger at the CDF Experiment." *IEEE Transactions on Nuclear Science* no. 47 (2):28.

Arias-Castro, E. 2012. *Detecting Correlations*. San Diego, CA: 2012 Algorithms for Threat Detection Workshop.

Baraniuk, R. 2007. "Compressive Sensing." *IEEE Signal Processing Magazine*:118-121. *An introduction to Compressive Sensing and the Rice Single Pixel Camera application.*

Bedworth, M., and J. O'Brien. 2000. "The Omnibus Model: A New Model of Data Fusion." *IEEE Aerospace and Electronic Systems Magazine* no. 15 (4):30-36. *There are several common models for data and information fusion, including the intelligence cycle, the JDL model, the OODA loop, the Dasarathy model, and the*

waterfall model. This paper discusses each, emphasizing advantages and disadvantages, and proposes the omnibus model as a synthesis.

Bender, M., I. Ginis, R. Tuleya, B. Thomas, and T. Marchok. 2007. "The Operational GFDL Coupled Hurricane–Ocean Prediction System and a Summary of Its Performance." *Monthly Weather Review* no. 135:3965-3989.

Bertozzi, A. 2010. *Image Processing of Multi and Hyperspectral Imagery*. Raleigh, NC: 2010 Algorithms for Threat Detection Workshop.

Bierberbach, G., P. Bieringer, A. Wyszogrodzki, J. Weil, R. Cabell, J. Hurst, and J. Hannan. 2010. "Virtual Chemical and Biological (CB) Agent Data Set Generation to Support the Evaluation of CB Contamination Avoidance Systems." In *The Fifth International Symposium on Computational Wind Engineering (CWE2010)*.

Binev, P. 2011. *Point Cloud Processing*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Binev, P., R. Sharpley, R. DeVore, M. Hielsberg, G. Petrova, and S. Schaefer. 2010. *3D Reconstructions of Simulated Frequency Agile Lidar (FAL) Sensed Concentration Fields*. Raleigh, NC: 2010 Algorithms for Threat Detection Workshop.

Bishop, C. 1995. *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press. *The "Bible" for artificial neural networks – a graduate-level textbook covering artificial neural networks and other techniques for statistical pattern recognition*.

Blom, H., and Y. Bar-Shalom. 1988. "The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients." *IEEE Transactions on Automatic Control* no. 33 (8):780-783.

Boston Scientific. *Discover Innovation* 2013 [cited February 15. Available from http://www.bostonscientific.com/templatedata/imports/HTML/intl/discover/spotlight-CRM-discinnov-Intl.html. *A website noting the algorithms used to classify fast heart rhythms as lethal vs. non-lethal in Boston Scientific ICDs*.

Bouttier, F., and P. Courtier. 1999. *Lecture Notes: Data Assimilation Concepts and Methods. These somewhat dated lecture notes from the European Centre for Medium-Range Weather Forecasts are much easier to understand than the journal articles, and give a good practical background on data assimilation for numerical weather prediction*.

Boyd, J. 2013. *The Essence of Winning and Losing* 1995 [cited February 15 2013]. Available from http://www.danford.net/boyd/essence.htm.

Bravata, D., K. McDonald, W. Smith, C. Rydzak, H. Szeto, D. Buckeridge, C. Haberland, and D. Owens. 2004. "Systematic Review: Surveillance Systems for Early Detection of Bioterrorism-related Diseases." *Annals of Internal Medicine* no. 140:910-922. *An overview of syndromic surveillance systems developed from 1985-2002*.

Candes, E., and M. Wakin. 2008. "An Introduction to Compressive Sampling." *IEEE Signal Processing Magazine*:21-30. *An introduction to Compressive Sensing*.

Carl, J. 2001. "Contrasting Approaches to Combine Evidence." In *Handbook of Multisensor Data Fusion*, edited by D. Hall and J. Llinas, 7.1-7.31. Boca Raton, FL: CRC Press.

Carr, A., J. Broadwater, and D. Limsui. 2011. *Challenges in Remote Plume Detection and Identification*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Chair, Z., and P. Varshney. 1986. "Optimal Data Fusion in Multiple Sensor Detection Systems." *IEEE Transactions on Aerospace and Electronic Systems* no. 1:98-101. *Approach for optimally fusing hypothesis test decisions from individual sensors.*

Chandola, V., A. Banerjee, and V. Kumar. 2007. Outlier Detection: A Survey. Minneapolis, MN: University of Minnesota. TR 07-017.

———. 2009. "Anomaly Detection: A Survey." *ACM Computing Surveys* no. 41 (3):15. *A review and categorization of techniques for anomaly detection.*

Charpak, G. 1978. "Multiwire and Drift Proportional Chambers." *Physics Today* no. October:23-30.

Cornic, P., P. Garrec, S. Kemkemian, and L. Ratton. 2011. "Sense and Avoid Radar using Data Fusion with Other Sensors." In 2011 IEEE Aerospace Conference. *This document discusses multiple sensor tracking in a SAA context and presents a fusion algorithm based on the Variable Structure Interacting Multiple Model.*

Craparo, E., R. Koyak, K. Wood, and S. Buttrey. 2011. *Graph-Theoretic Statistical Methods for Signal Detection*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Cummings, J. 2005. "Operational Multivariate Ocean Data Assimilation." *Quarterly Journal of the Royal Meteorological Society* no. 131 (613):3583-3604. doi: doi:10.1256/qj.05.105. *This paper provides an overview of the MVOI ocean data assimilation system in operational use at the U.S. Navy oceanographic production centers.*

Darms, M. S., P. E. Rybski, C. Baker, and C. Urmson. 2009. "Obstacle Detection and Tracking for the Urban Challenge." *IEEE Transactions on Intelligent Transportation Systems* no. 10 (3):475-485. *This paper describes the obstacle detection and tracking algorithms developed by Carnegie Mellon University for their winning entry in the 2007 DARPA Urban Challenge. There is a specific discussion of the need for different types of sensors because no single sensor type would adequately capture the environment.*

Dasarathy, B. 1997. "Sensor Fusion Potential Exploitation–Innovative Architectures and Illustrative Applications." *Proceedings of IEEE* no. 85 (1):24-38. *This paper develops a characterization of the sensor fusion process based on a generalized input-output descriptor pair.*

Davis, J., and M. Goadrich. 2006. "The Relationship between Precision-Recall and ROC Curves." In *Proceedings of the 23rd International Conference on Machine Learning*, edited by W. Cohen and A. Moore, 233-240. New York: Association for Computing Machinery (ACM). *Discussion of the relationship between Precision-Recall and ROC curves.*

DeVore, R., G. Petrova, M. Hielsberg, L. Owens, B. Clack, and A. Sood. 2013. "Processing Terrain Point Cloud Data." *SIAM Journal of Imaging Sciences* no. 6 (1):1-31.

Donoho, D. 2006. "Compressed Sensing." *IEEE Transactions on Information Theory* no. 52:1289-1306. *One of the first journal articles on Compressive Sensing*.

Duda, R., P. Hart, and D. Stork. 2001. *Pattern Classification*. New York: John Wiley and Sons. *The most well-known graduate-level textbook for pattern classification from a machine learning point of view*.

Effertz, J. 2008. "Sensor Architecture and Data Fusion for Robotic Perception in Urban Environments at the 2007 DARPA Urban Challenge." In *Robot Vision: Proceedings of the Second International Workshop, RobVis 2008, Auckland, New Zealand, February 18-20, 2008*, edited by G. Sommer and R. Klette, 275-290. Berlin: Springer. *This paper describes the development of the vehicle assembled by Technische Universität Braunschweig for the 2007 DARPA Urban Challenge. The use of hybrid fusion is described as are sensor signal processing and conversion of these signals into object-oriented data. Obstacle tracking using an Extended Kalman Filter is described.*

Eiceman, G. *Mobility as a Concept for Chemical Measurements* 2013 [cited February 15]. Available from https://grassmann.math.colostate.edu/ATD/IMS.html.

Elmenreich, E. 2002. *Sensor Fusion in Time-Triggered Systems*: Dissertation, Technische Universitat Wien.

Esser, E. 2011. *Variational Models for DOAS Analysis*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Evensen, G. 1994. "Sequential Data Assimilation with a Nonlinear Quasi-Geostrophic Model using Monte Carlo Methods to Forecast Error Statistics." *Journal of Geophysical Research* no. 99 (5):10143-10162. *A description of ensemble Kalman filtering.*

———. 2009. *Data Assimilation: The Ensemble Kalman Filter, 2nd Edition*. Berlin: Springer. *This book has a very theoretical formulation, but could be useful*.

Fabry–Pérot Interferometer Sensor Data Set Algorithm Development Data Set. 2013 [cited February 15]. Available from https://grassmann.math.colostate.edu/ATD/FPISDS.html.

Federal Aviation Adminstration. *Aeronatuical Information Manual* February 9, 2012. Available from http://www.faa.gov/air_traffic/publications/ATpubs/AIM/aim.pdf.

Fermilab. 1996. The CDF II Detector Technical Design Report. Batavia, IL. FERMILAB-Pub-96/390-E.

Ferrin, I., D. Rabinowitz, B. Schaefer, J. Snyder, N. Ellman, B. Vicente, A. Rengstorf, D. Depoy, S. Salim, P. Andrews, C. Bailyn, C. Baltay, C. Briceno, P. Coppi, M. Deng, W. Emmet, A. Oemler, C. Sabbey, J. Shin, S. Sofia, W. van Altena, K. Vivas, C. Abad, A. Bongiovanni, G. Bruzual, F. Della Prugna, D. Herrera, G. Magris, J. Mateu, R. Pacheco, Ge. Sanchez, Gu. Sanchez, H. Schenner, J. Stock, K. Vieira, F.

Fuenmayor, J. Hernandez, O. Naranjo, P. Rosenzweig, C. Secco, G. Spavieri, M. Gebhard, H. Honeycutt, S. Mufson, J. Musser, S. Pravdo, E. Helin, and K. Lawrence. 2001. "Discovery of the Bright Trans-Neptunian Object 2000 EB173." *Astrophysics Journal* no. 548:L243-L248.

Floyd, S., and V. Jacobson. 1993. Random Early Detection Gateways for Congestion Avoidance." *IEEE/ACM Transactions on Networking* no. 1 (4):397-413

Fricker, R. 2013. *Introduction to Statistical Methods for Biosurveillance: With an Emphasis on Syndromic Surveillance*. Cambridge, UK: Cambridge University Press.

Frost, R., and T. Hartl. 1996. "A Cognitive-Behavioral Model of Compulsive Hoarding." *Behaviour Research and Therapy* no. 34 (4):341-350.

Geyer, C., S. Singh, and L. Chamberlain. 2008. Avoiding Collisions Between Aircraft: State of the Art and Requirements for UAVs Operating in Civilian Airspace. Pittsburgh, PA: Carnegie Mellon University. CMU-RI-TR-08-03.

Gillis, D., J. Grun, and J. Bowles. *Swept Wavelength Optical Resonant Raman Detection: An Overview* 2013 [cited February 15]. Available from https://grassmann.math.colostate.edu/ATD/SWOrRD.html.

Graham, S., and J. Kay. 2012. *Multi-Sensor Integrated Conflict Avoidance (MuSICA)*: 2012 International Test and Evaluation Association Technology Review.

Greer, J. 2010. *Sparse Demixing*. Raleigh, NC: 2010 Algorithms for Threat Detection Workshop.

Grelaud, A. 2011. *Nuclear Surveillance with Mobile Sensor Network Using a Dynamic Model Approach*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Grun, J. 2013. *Detecting Dangerous Substances from their 2D Spectra*. SPIE Newsroom 2011 [cited February 15 2013]. Available from http://spie.org/documents/Newsroom/Imported/003600/003600_10.pdf.

Gu, Q., Z. Li, and J. Han. 2011. "Linear Discriminant Dimensionality Reduction." In *ECML PKDD'11: Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases*, edited by D. Gunopulos, T. Hofmann, D. Malerba and M. Vazirgiannis, 549-564. *A discussion of the similarities and differences between the Fisher Score for feature selection versus the Fisher Criterion for feature extraction, as well as a discussion of a generalized technique that links both techniques together*.

Gu, Q., Z. Li, and J. Han. 2013. *Generalized Fisher Score for Feature Selection* 2012 [cited February 15 2013]. Available from http://arxiv.org/abs/1202.3725. *An overview of the advantages and disadvantages of the Fisher Score for feature selection, as well as a discussion of a generalized technique that overcomes many hurdles facing the traditional technique*.

. 2011. *Applications of Splitting Methods*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Guyon, I., and A. Elisseeff. 2003. "An Introduction to Variable and Feature Selection." *Journal of Machine Learning Research* no. 3:1157-1182. *An overview of filter-based and wrapper-based methods for feature selection.*

Hagen, K., R. Fricker, K. Hanni, S. Barnes, and K. Michie. 2011. "Assessing the Early Aberration Reporting System's Ability to Locally Detect the 2009 Influenza Pandemic." *Statistics, Policits, and Policy* no. 2 (1):1.

Hero, A., D. Castañón, D. Cochran, and K. Kastella. 2008. *Foundations and Applications of Sensor Management*. New York: Springer.

Hero, A., and D. Cochran. 2011. "Sensor Management: Past, Present, and Future." *IEEE Sensors Journal* no. 11 (12):3064-3075. *A review and categorization of sensor management algorithms.*

Jacob, B., and S. Levitt. 2003. "Rotten Apples: An Investigation of the Prevalence and Predictors of Teacher Cheating." *Quarterly Journal of Economics* no. 118 (3):843-877.

Johns Hopkins Applied Physics Lab Data. 2013 [cited February 15]. Available from https://grassmann.math.colostate.edu/ATD/APL.html.

Kalman, R. E., and R. S. Bucy. 1961. "New Results in Linear Filtering and Prediction Theory." *Journal of Basic Engineering* no. 83 (3):95-108. *This paper presents the theory behind the Kalman Filter, an algorithm that uses a series of measurements that contain noise and error to produce estimates of unknown variables. These estimates are more precise that those based on a single measurement.*

Kalnay, E. 2003. *Atmospheric Modeling, Data Assimilation, and Predictability*. Cambridge, UK: Cambridge University Press. *This book is a standard text on data assimilation.*

———. 2010. "Ensemble Kalman Filter: Current Status and Potential." In *Data Assimilation: Making Sense of Observations*, edited by W. Lahoz, B. Khattatov and R. Menard, 69-92. Heidelberg: Springer.

Kalnay, E., H. Li, T. Miyoshi, S. Yang, and J. Ballabrera-Poy. 2007. "4-D-Var or Ensemble Kalman Filter?" *Tellus A* no. 59:758-773. *This paper compares 4D-Var performance to EnKF variants' (including LETKF's) performance for Lorenz and SPEEDY global primitive equations models.*

Kalnay, E., Y. Ota, T. Miyoshi, and J. Liu. 2012. "A Simpler Formulation of Forecast Sensitivity to Observations: Application to Ensemble Kalman Filters." *Tellus A* no. 64:18462-18471. doi: http://dx.doi.org/10.3402/tellusa.v64i0.18462.

Kalnay, E., and S. Yang. 2010. "Accelerating the Spin-Up of Ensemble Kalman Filtering." *Quarterly Journal of the Royal Meteorological Society* no. 136 (651):1644-1651. *A heuristic method for accelerating the spin-up time of the ensemble Kalman filter.*

Kalnay, E., S. Yang, J. Kang, T. Miyoshi, S. Penny, and G. Lien. 2012. *Recent Advances in EnKF: Running in Place, Assimilation of Rain, Ensemble Forecast Sensitivity to Observations*. National Taiwan University.

Karu, Z. 1995. *Signals and Systems Made Ridiculously Simple*. Cambridge, MA: ZiZi Press. *An introduction to traditional sampling theory.*

Keshava, N., and J. Mustard. 2002. "Spectral Unmixing." *IEEE Signal Processing Magazine* no. 19 (1):44-57.

Khaleghi, B., A. Khamis, F. Karray, and S. Razavi. 2013. "Multisensor Data Fusion: A Review of the State-of-the-Art." *Information Fusion* no. 14 (1):28-44. *A review of multidisciplinary research on multisensor data fusion.*

Koks, D., and S. Challa. 2003. An Introduction to Bayesian and Dempster-Shafer Data Fusion. Edinburgh, SA, Australia: DSTO System Sciences Laboratory. DSTO-TR-1436. *An introduction to Kalman filtering and Dempster-Shafer information combination.*

Kopriva, S., D. Sislak, and M. Pechoucek. 2012. "Sense and Avoid Concepts: Vehicle-Based SAA Systems." In *Sense and Avoid in UAS: Research and Applications*, edited by P. Angelov. Chichester, UK: John Wiley and Sons.

Kuchar, J., and L. Yang. 2000. "Survey of Conflict Detection and Resolution Modeling Methods." *IEEE Transactions on Intelligent Transportation Systems* no. 1:179-189.

Kunii, M., T. Miyoshi, and E. Kalnay. 2012. "Estimating the Impact of Real Observations in Regional Numerical Weather Predictions Using an Ensemble Kalman Filter." *Monthly Weather Review* no. 140 (6):1975-1987. *This paper gives an example of estimating the impact of individual observations using Liu and Kalnay's ensemble-based sensitivity method for Typhoon Sinlaku (2008), the WRF model, and LETKF.*

Lacher, A., D. Maroney, and A. Zeitlin. 2007. Unmanned Aircraft Collision Avoidance--Technology Assessment and Evaluation Methods. MITRE Corporation, 07-0095. *This paper lays out the tradespace for sensor types and attributes to be considered when choosing a sensor suite for SAA on unmanned aerial systems.*

Lahoz, W., B. Khattatov, and R. Menard. 2010. *Data Assimilation: Making Sense of Observations*. Heidelberg: Springer. *This book has chapters generally describing variational methods and advances in ensemble Kalman Filtering, but is not quite a textbook.*

Landon, J. 2011. *Analysis of Two Data Sets: Chemical Threat Detection and Raman Spectrums*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Lawson, A., and K. Kleinman. 2005. *Spatial and Syndromic Surveillance for Public Health*. Chichester, UK: John Wiley & Sons.

Lee, W., and S. Stolfo. 1998. "Data Mining Approaches for Intrusion Detection." In *Proceedings of the 7th USENIX Security Symposium*, 79–94. Berkeley, CA: Advanced Computing Systems Association.

Li, H., E. Kalnay, and T. Miyoshi. 2009. "Simultaneous Estimation of Covariance Inflation and Observation Errors within an Ensemble Kalman Filter " *Quarterly Journal of the Royal Meteorological Society* no. 135:523–533. *This paper*

describes a technique for estimating the observation errors and best globally uniform inflation factors for the data inside the EnKF scheme.

Li, X. 2000. "An Engineer's Guide to Variable Structure Multiple Model Estimation for Tracking." In *Multitarget-Multisensor Tracking: Applications and Advances*, edited by Y. Bar-Shalom and D. Blair, 499-567. Boston: Artech House.*This book chapter provides an extensive discussion of the variable structure multiple model and the design and evaluation of algorithms for target tracking applications.*

Li, X., and Y. Bar-Shalom. 1996. "Multiple-Model Estimation with Variable Structure." *IEEE Transactions on Automatic Control* no. 41:478-493.

Li, X., and V. Jilkov. 2005. "Survey of Maneuvering Target Tracking Part V: Multiple Model Methods." *IEEE Transactions on Aerospace and Electronic Systems* no. 41 (4):1255-1321. *This is the fifth part of a series of papers that provide a comprehensive survey of techniques for tracking maneuvering targets. Parts I and II deal with target motion models. Part III covers measurement models and associated techniques. Part IV is concerned with tracking techniques that are based on decisions regarding target maneuvers. This part surveys the multiple-model methods–the use of multiple models (and filters) simultaneously–which is the prevailing approach to maneuvering target tracking in recent years.*

Liu, J., E. Chu, and P. Tsai. 2012. "Fusing Human Sensor and Physical Sensor Data." In *2012 5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*, 1-5. *Addresses the problem of symbiotic data fusion and processing, which is how to use human sensor (crowdsourced) data and physical data synergistically to speed up or improve decision processes.*

Liu, J., and E. Kalnay. 2007. "Simple Doppler Wind Lidar Adaptive Observation Experiments with 3D-Var and an Ensemble Kalman Filter in a Global Primitive Equations Model." *Geophysical Research Letters* no. 34 (19):L19808. *This paper gives the original derivation of the ensemble-based sensitivity method of Liu and Kalnay.*

Lorenc, A. 1986. "Analysis Methods for Numerical Weather Prediction." *Quarterly Journal of the Royal Meteorological Society* no. 112 (474):1177-1194. *A review of methodologies for numerical weather prediction prior to 1986.*

———. 2003. "The Potential of the Ensemble Kalman Filter for NWP—A Comparison with 4D-Var." *Quarterly Journal of the Royal Meteorological Society* no. 129:3183-3203.

Lu, L., C. Anderson-Cook, and T. Robinson. 2011. "Optimization of Designed Experients Based on Multiple Criteria Utilizing a Pareto Frontier." *Technometrics* no. 53 (4):353-365.

Lundberg, S., R. Paffenroth, and J. Yosinski. 2010. "Analysis of CBRN Sensor Fusion Methods." In *2010 13th Conference on Information Fusion (FUSION)*. IEEE. *This DTRA-sponsored work demonstrated the applicability of target tracking metrics to CBRN.*

Lustig, M., D. Donoho, J. Santos, and J. Pauly. 2008. "Compressed Sensing MRI." *IEEE Signal Processing Magazine*:72-82. *A discussion of the criteria an application must meet for successful Compressive Sensing and an introduction to the Magnetic Resonance Imaging application*.

MacIntosh, V. 2004. Enhancing Influenza Surveillance Using Electronic Surveillance System for the Early Notification of Community-Based Epidemics (Essence). NATO Paper RTO-MP-HFM-108. A description of *ESSENCE (The Electronic Surveillance System for the Early Notification of Community-based Epidemics), which tracks influence-like illnesses to assist in influenza outbreak detection and response.*

Manolakis, D. 2010. *Is There a Best Hyperspectral Detection Algorithm?* Raleigh, NC: 2010 Algorithms for Threat Detection Workshop.

Mazor, E., A. Averbuch, Y. Bar-Shalom, and J. Dayan. 1998. "Interacting Multiple Model Methods in Target Tracking: A Survey." *IEEE Transactions on Aerospace and Electronic Systems* no. 34 (1):103-123. *This paper describes the Interacting Multiple Model (IMM) estimator. IMM can estimate the state of a dynamic system with several behavior modes that can "switch" from one to another. IMM is both an algorithm for tracking maneuvering targets and a cost-effective hybrid state estimation scheme*.

McCalmont, J. 2007. *Small Sense and Avoid Systems*. Paris: UAV 2007 Conference.

Medtronic. 2013. *Case Study: PR Logic & Wavelet* 2012 [cited February 15 2013]. Available from https://wwwp.medtronic.com/mdtConnectPortal/presentationtools/108/117/1581802200908. *Powerpoint slides discussing two case studies involving the classification of fast heart rhythms as lethal vs. non-lethal in Medtronic ICDs. A multi-tiered "screen-then-classify" decision tree combines both interval and morphology algorithms*.

Meyer, D., F. Leisch, and K. Hornik. 2003. "The Support Vector Machine Under Test." *Neurocomputing* no. 55:169-186. *A comparison of Support Vector Machines vs. 16 other classification methods, applied to artificial and real-world data sets*.

Montgomery, M., C. Davis, T. Dunkerton, Z. Wang, C. Velden, S. Majumdar, F. Zhang, R. Smith, L. Bosart, M. Bell, J. Haase, A. Heymsfield, J. Jensen, T. Campos, and M. Boothe. 2012. "The Pre-Depression Investigation of Cloud-Systems in the Tropics (PREDICT) Experiment." *Bulletin of the American Meteorological Society* no. February:153-172.

Moss, A., W. Zareba, J. Hall, H. Klein, D. Wilber, D. Cannom, J. Daubert, S. Higgins, M. Brown, and M. Andrews. 2002. "Prophylactic Implantation of a Defibrillator in Patients with Myocardial Infarction and Reduced Injection Fraction." *New England Journal of Medicine* no. 346:877-883. *Landmark results of a randomized control trial showing a 31% reduction in mortality for ICDs vs. pharmaceuticals*.

Myers, R., D. Montgomery, and C. Anderson-Cook. 2009. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments, 3rd Ed.* . New York: Wiley.

National Heart Lung and Blood Institute. *What is an Implantable Cardioverter Defibrillator?* 2013 [cited February 15. Available from http://www.nhlbi.nih.gov/health/health-topics/topics/icd/. *An introduction to ICDs*.

Nerger, L., W. Hiller, and J. Schroter. 2005. "A Comparison of Error Subspace Kalman Filters." *Tellus A* no. 57:715-735.

Nerger, L., T. Janjic, J. Schroter, and W. Hiller. 2012. "A Unification of Ensemble Square Root Kalman Filters." *Monthly Weather Review* no. 140:2335-2349.

Osborne, J., S. Clark, R. Morris, I. Williams, J. Riley, A. Smith, D. Reynolds, and A. Edwards. 1999. "A Landscape-Scale Study of Bumble Bee Foraging Range and Constancy, Using Harmonic Radar." *Journal of Applied Ecology* no. 36:519-533.

Platt, N., and D. DeRiggi. 2012. Comparative Investigation of Source Term Estimation Algorithms for Hazardous Material Atmospheric Transport and Dispersion Prediction Tools. Institute for Defense Analyses, D-4048.

Platt, N., S. Warner, and S. Nunes. 2008. Plan for Initial Comparative Investigation of Source Term Estimation Algorithms Using FUSION Field Trial 2007 (FFT 07). Institute for Defense Analyses, D-3488.

Plaza, A., P. Martinez, R. Perez, and Plaza. J. 2004. "A Quantitative and Comparative Analysis of Endmember Extraction Algorithms from Hyperspectral Data." *IEEE Transactions on Geoscience and Remote Sensing* no. 42 (3):650-663.

Prado, R., and M. West. 2010. *Time Series: Modeling, Computation, and Inference*. Boca Raton, FL: Taylor & Francis Group.

. 2012. "DART/CAM: An Ensemble Data Assimilation System for CESM Atmospheric Models." *Journal of Climate* no. 25:6304-6317. *This paper describes the DART system, and shows how it can be used for climate modeling*.

Rasmussen, C., and C. Williams. 2006. *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.

Rodriguez, A. 2012. *Dynamic Sensor Fusion: A Statistician's Perspective*. San Diego, CA: 2012 Algorithms for Threat Detection Workshop.

Rousseau, M., T. Ratton, and T. Fournet. 2010. "Multiple Sensor Tracking in a Sense and Avoid Context." In *27th International Conference of the Aeronautical Sciences*. International Council of the Aeronatucial Sciences. *This document discusses multiple sensor tracking in a SAA context and presents a fusion algorithm based on the Variable Structure Interacting Multiple Model*.

Roweis, S., and Z. Ghahramani. 1999. "A Unifying Review of Linear Gaussian Models." *Neural Computation* no. 11:305-345. *An excellent overview of linear Gaussian models, explaining the links between Principal Component Analysis, Factor Analysis, Kalman Filters, Hidden Markov Models, and Independent Component Analysis*.

Roweis, S., and L. Saul. 2000. "Nonlinear Dimensionality Reduction by Locally Linear Embedding." *Science* no. 290 (5500):2323-2326. *A discussion of the Local Linear Embedding feature extraction technique*.

Scott, S. 2010. "A Modern Bayesian Look at the Multi-Armed Bandit." *Applied Stochastic Models in Business and Industry* no. 26:639-658.

Shanker, T., and M. Richtel. *In New Military, Data Overload Can Be Deadly*. New York Times, January 16, 2011 [cited February 15, 2013. Available from http://www.nytimes.com/2011/01/17/technology/17brain.html?pagewanted=all&_r=0.

Shmueli, G., and S. Fienberg. 2006. "Current and Potential Statistical Methods for Monitoring Multiple Data Streams for Biosurveillance." In *Statistical Methods in Counterterrorism: Game Theory, Modeling, Syndromic Surveillance, and Biometric Authentication*, edited by A. Wilson, G. Wilson and D. Olwell, 109-140. New York: Springer. *A review paper discussing data sources and statistical methodology for syndromic surveillance.*

Shome, S., D. Li, J. Thompson, and A. McCabe. 2010. "Improving Contemporary Algorithms for Implantable Cardioverter-Defibrillator Function." *Journal of Electrocardiology* no. 43:503-508. *An updated description of the algorithm used to classify fast heart rhythms as lethal vs. non-lethal in Boston Scientific ICDs.*

Siegmund, D., N. Zhang, B. Yakir, Y. Xie, and H. Ji. 2012. *Detecting Simultaneous Change-points in Multiple Streams of Data*. San Diego, CA: 2012 Algorithms for Threat Detection Workshop.

Simmons, A., and A. Hollingsworth. 2002. "Some Aspects of the Improvement in Skill of Numerical Weather Prediction." *Quarterly Journal of the Royal Meteorological Society* no. 128:647-677.

Singpurwalla, N., and J. Booker. 2004. "Membership Functions and Probability Measures of Fuzzy Sets (with discussion)." *Journal of the American Statistical Association* no. 99 (467):867-889.

Sivaprakasam, V., J. Tucker, and J. Eversole. *Ambient Aerosol Background Data Set for the Rapid Agent Aerosol Detection (RAAD) Program* 2013 [cited February 15]. Available from https://grassmann.math.colostate.edu/ATD/AABDS.html.

Smith, R. 2012. *AFRL SAA Research*: Defense Acquisition University Acquisition Insight Days.

Steinberg, A., and C. Bowman. 2009. "Revisions to the JDL Data Fusion Model." In *Handbook of Multisensor Data Fusion: Theory and Practice, 2nd Edition*, edited by M. Liggins, D. Hall and J. Llinas. Boca Raton, FL: Taylor and Francis. *The JDL data fusion model and its subsequent revisions is the most widely used system for categorizing data fusion-related functions. The goal of the JDL Data Fusion Model is to facilitate understanding and communication among acquisition managers, theoreticians, designers, evaluators, and users of data fusion techniques.*

Steinberg, A., C. Bowman, and F. White. 1999. "Revisions to the JDL Data Fusion Model." In *AeroSense Conference: Proceedings of SPIE*, 430-441. International Society for Optics and Photonics.

Stentz, A. 1994. "Optimal and Efficient Path Planning for Partially-Known Environments." In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 3310-3317. IEEE.

Storwald, D. 2007. Detailed Test Plan for Fusing Sensor Information from Observing Networks (Fusion) Field Trial (FFT-07). Meteorology Division, West Desert Test Center, U.S. Army Dugway Proving Ground.

Stover, J., D. Hall, and R. Gibson. 1996. "A Fuzzy-Logic Architecture for Autonomous Multisensor Data Fusion." *IEEE Transactions on Industrial Electronics* no. 43 (3):403-410. *Discussion of a general fuzzy-logic architecture for interpretation and fusion of multisensor data.*

Strohmer, T. 2011. *Accurate Chemical Cloud Detection via Sparse Representations and Convex Optimization*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Sudderth, E., A. Ihler, W. Freeman, and A. Willsky. 2002. Nonparametric Belief Propagation and Facial Appearance Estimation. MIT Artificial Intelligence Lab. AIM-2002-020. *Nonparametric belief propagation algorithm applied to estimation of occluded features.*

Swerdlow, C., M. Brown, K. Lurie, J. Zhang, N. Wood, W. Olson, and J. Gillberg. 2002. "Discrimination of Ventricular Tachycardia from Supraventricular Tachycardia by Downloaded Wavelet-Transform Morphology Algorithm: A Paradigm for Development of Implantable Cardioverter Defibrillator Detection Algorithms." *Journal of Cardiovascular Electrophysiology* no. 13:432-441. *A description of the algorithm used to classify fast heart rhythms as lethal vs. non-lethal in Medtronic devices, along with performance results.*

Szunyogh, I., Z. Toth, R. Morss, S. Majumdar, B. Etherton, and C. Bishop. 2000. "The Effect of Targeted Dropsonde Observations During the 1999 Winter Storm Reconnaissance Program." *Monthly Weather Review* no. 128 (10):3520-3537.

Tarassenko, L. 1998. *Guide to Neural Computing Applications*. Oxford: Butterworth-Heineman. *A short book with clear advice on how to avoid the pitfalls of Artificial Neural Networks*.

Tartakovsky, A. 2012. *Advanced Quickest Changepoint Detection Methods for Threat Assessment in Distributed Sensing Systems*. San Diego, CA: 2012 Algorithms for Threat Detection Workshop.

Thomopoulos, S., R. Viswanathan, and D. Bougoulias. 1987. "Optimal decision fusion in multiple sensor systems." *IEEE Transactions on Aerospace and Electronic Systems* no. 5:644-653. *Discussion of optimal decision fusion in the sense of the Neyman-Pearson test in a centralized fusion center.*

Thrun, S., M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. 2007. "Stanley: The Robot That Won the

DARPA Grand Challenge." In *The 2005 DARPA Grand Challenge*, edited by M. Buehler, K. Iagnemma and S. Singh, 1-43. Berlin: Springer. *This paper describes the development of Stanley, Stanford's winning entry to the 2005 DARPA Grand Challenge. A description of the DARPA Challenge is included. The paper discusses sensor selection and implementation, data processing, path planning, and decision-making algorithms*.

Toth, Z., I. Szunyogh, C. Bishop, S. Majumdar, R. Morss, and S. Lord. 2011. "On the Use of Target Observations in Operational Numerical Weather Prediction." In *Fifth American Meteorological Society Symposium on Integrated Observing Systems*, 72-79. Boston: American Meteorological Society.

Truscott, B. 2004. *Overview of the Atlantic-THORPEX Regional Campaign*. Montreal, CA: THORPEX Symposium.

Tsitsiklis, J. 1993. "Decentralized Detection." *Advances in Statistical Signal Processing* no. 2:297-344. *Bayesian formulation of decision fusion.*

University Corporation for Atmospheric Research. *Research Applications Laboratory 2011 Annual Report: Virtual Atmospheric Dispersion Field Testing* 2011 [cited February 15, 2013]. Available from http://nar.ucar.edu/2011/lar/ral/virtual-atmospheric-dispersion-field-testing.

Urmson, C., J. Anhalt, M. Clark, T. Galatali, J. Gonzales, J. Gowdy, A. Gutierrez, S. Harbaugh, M. Johnson-Roberson, H. Kato, P. Koon, K. Peterson, B. Smith, S. Spiker, E. Tryzelaar, and W. R. Whittaker. 2004. High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge 2004. Pittsburgh, PA: Carnegie Mellon University, CMU-RI-TR-04-37. *This paper describes the development of Carnegie Mellon University's entry into the 2004 DARPA Grand Challenge. Discussion topics include development and incorporation of high performance computing and sensing. Isolation of the sensors from the analysis and "thinking layer" of the system to make a sensor-agnostic cost model is described.*

Urmson, C., R. Simmons, and I. Nesnas. 2003. "A Generic Framework for Robotic Navigation." In *Proceedings of the 2003 IEEE Aerospace Conference*, 5_2463-5_2470. *This paper details the generic framework of autonomous vehicle development along sense-think-act lines. Separation of a functional layer and decision layer are described. The paper discusses the combination of Morphin, a local obstacle avoidance algorithm, and D*, a real time path planner. The backbone methodology of traversability analysis that is later converted into cost functions used for decision making is presented*.

van der Maaten, L., E. Postma, and H. van den Herik. 2009. "Dimensionality Reduction: A Comparative Review." *Journal of Machine Learning Research* no. 10:1-22. *A well-written overview comparing the performance of Principal Component Analysis to several non-linear techniques for feature extraction on both synthetic and real-world data sets*.

van Leeuwen, P. 2009. "Particle Filtering in Geophysical Systems." *Monthly Weather Review* no. 137:4089-4114. *This paper is an overview of particle filtering methods for geophysical systems, which are rather new*.

Vanderbeek, R. *Description of the Frequency Agile Lidar (FAL) Bio Standoff Detection Data Obtained Using the Joint Ambient Breeze Tunnel (JABT)* 2013 [cited February 15]. Available from https://grassmann.math.colostate.edu/ATD/Downloads/Datasets/FAL/FALDescription.pdf.

Vanderbeek, R., and R. Warren. 2010. *Lidar for Biodetection*. Raleigh, NC: 2010 Algorithms for Threat Detection Workshop.

Veeraraghavan, H., P. Schrater, and N. Papanikolopoulos. 2005. "Switching Kalman Filter-Based Approach for Tracking and Event Detection at Traffic Intersections." In *Proceedings of the 2005 IEEE International Symposium on Intelligent Control,* 1167-1172. IEEE. *This paper describes the development of a robust tracking algorithm for targets through a combination of multiple cues and multiple motion models. The use of a switching Kalman filter is discussed for use in a simple event detection system.*

Vicente, B., C. Abad, F. Garzón, and T. Girard. 2010. "Astrometry with Carte du Ciel Plates, San Fernando Zone II. CdC-SF: A Precise Proper Motion Catalogue." *Astronomy and Astrophysics* no. 509:A62.

Vidakovic, B. 1999. *Statistical Modeling by Wavelets*. New York: Wiley.

Viswanathan, R., and P. Varshney. 1997. "Distributed Detection with Multiple Sensors I. Fundamentals." *Proceedings of the IEEE* no. 85 (1):54-63. *Review of detection and estimation with distributed sensors.*

Waltz, E., and T. Waltz. 2009. "Principles and Practice of Image and Spatial Data Fusion." In *Handbook of Multisensor Data Fusion: Theory and Practice, 2nd Edition*, edited by M. Liggins, D. Hall and J. Llinas, 89-114. Boca Raton, FL: CRC Press.

Wan, L., J. Zhu, H. Wang, and C. Yan. 2009. "A "Dressed" Ensemble Kalman Filter Using the Hybrid Coordinate Ocean Model in the Pacific." *Advances in Atmospheric Sciences* no. 26 (5):1042-1052.

Welzl, M. 2005. *Network Congestion Control: Managing Internet Traffic*. Chichester, UK: John Wiley & Sons.

Weng, Y., and F. Zhang. 2012. "Assimilating Airborne Doppler Radar Observations with an Ensemble Kalman Filter for Convection-Permitting Hurricane Initialization and Prediction: Katrina (2005)." *Monthly Weather Review* no. 140:841-859. *This paper investigates the performance of EnKF when airborne RADAR observations are available, and discusses the techniques used to prepare the data for assimilation.*

Weng, Y., M. Zhang, and F. Zhang. 2011. "Advanced Data Assimilation for Cloud-Resolving Hurricane Initialization and Prediction." *Computing in Science & Engineering* no. January/February:40-49. *This paper compares the performance of the assimilation schemes 3D-Var, 4D-Var, and EnKF with a version of the WRF model for the case of Hurricane Katrina (2005) making landfall on Florida, with data assimilated from ground-based weather surveillance RADARs.*

Wong, W., A. Moore, G. Cooper, and Wagner. M. 2005. "What's Strange About Recent Events (WSARE): An Algorithm for the Early Detection of Disease Outbreaks." *Journal of Machine Learning Research* no. 6:1961–1998. *A discussion of the WSARE (What's Strange About Recent Events) outbreak detection algorithm that uses a multivariate approach.*

Wu, H., M. Siegel, R. Stiefelhagen, and J. Yang. 2002. "Sensor Fusion using Dempster-Shafer Theory for Context-Aware HCI." In *Proceedings of the 19th IEEE Instrumentation and Measurement Technology Conference*, 7-12. Piscataway, NJ: IEEE. *Dempster-Shafer theory for context-aware computing.*

Xie, Y., and D. Siegmund. 2013. *Sequential Multi-Sensor Change-Point Detection* 2012 [cited February 15 2013]. Available from http://arxiv.org/pdf/1207.2386.pdf.

Xin, J. 2010. *Sparsity Induced Convexity in Blind Source Separation*. Raleigh, NC: 2010 Algorithms for Threat Detection Workshop.

Xun, X. 2011. *Bayesian Approach to Detection of Small Low Emission Sources on a Large Random Background*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Xun, X., B. Mallick, R. Carroll, and P. Kuchment. 2011. "A Bayesian Approach to the Detection of Small Low Emission Sources." *Inverse Problems* no. 27 (11). doi: doi:10.1088/0266-5611/27/11/115009.

Yablonsky, R., and Isaac Ginis. 2008. "Improving the Ocean Initialization of Coupled Hurricane–Ocean Models Using Feature-Based Data Assimilation." *Monthly Weather Review* no. 7:2592–2607. *This paper illustrates the use of "feature-based" data assimilation to initialize the ocean model in the GFDL hurricane model*.

Yang, S, E. Kalnay, B. Hunt, and N. Bowler. 2009. "Weight Interpolation for Efficient Data Assimilation with the Local Ensemble Transform Kalman Filter. "*Quarterly Journal of the Royal Meteorological Society* no. 135:251-262.

YouTube. February 15. *Statistics* 2013February 15]. Available from http://www.youtube.com/yt/press/statistics.html.

Zhang, F., Y. Weng, J. Sippel, Z. Meng, and C. Bishop. 2009. "Cloud-Resolving Hurricane Initialization and Prediction through Assimilation of Doppler Radar Observations with an Ensemble Kalman Filter." *Monthly Weather Review* no. 137:2105-2125. *This paper shows full-color images of the prediction of Hurricane Humberto (2007) using a WFR model and EnKF scheme to assimilate data from ground-based weather surveillance RADARs. This paper shows how much better the system performs than the 3D-Var system in use at the time, and discusses the data-thinning techniques necessary for using RADAR data in an EnKF scheme.*

Zhang, X., M. Burger, and S. Osher. 2011. "A Unified Primal-Dual Algorithm Framework Based on Bregman Iteration." *Journal of Scientific Computing* no. 46 (1):20-46.

Zhang, Y., X. Liao, and L. Carin. 2004. "Detection of Buried Targets Via Active Selection of Labeled Data: Application to Sensing Subsurface UXO." *IEEE*

Transactions on Geoscience and Remote Sensing no. 42:2535-253. *An explanation of how Active Learning is performed in the real-world application of classifying buried metallic objects as unexploded ordnance (threat) vs. clutter*.

Zhu, X., and A. Goldberg. 2009. *Introduction to Semi-Supervised Learning*. San Rafael, CA: Morgan and Claypool Publishers. *An introduction to semi-supervised learning*.

Ziegelmeier, L., M. Kirby, and C. Peterson. 2011. *Locally Linear Embedding Clustering Algorithms for Natural Imagery*

Boston, MA: 2011 Algorithms for Threat Reduction Workshop.

———. 2013. *Locally Linear Embedding Clustering Algorithm for Natural Imagery* 2012 [cited February 15 2013]. Available from http://arxiv.org/pdf/1202.4387v1.pdf.

Zou, F., A. Karr, D. Banks, M. Heaton, G. Datta, J. Lynch, and F. Vera. 2011. *Bayesian Spatio-Temporal Syndromic Surveillance: An Epidemiological Prospective*. Boston, MA: 2011 Algorithms for Threat Detection Workshop.

Appendix B
Acronyms and Abbreviations

3D-Var	3D variational
4D-Var	4D variational
4D-LETKF	4D local ensemble transform Kalman filter
ACADA	automatic chemical agent detector alarm
ACARS	Aircraft Communications Addressing and Reporting System
ADS-B	automatic dependent surveillance–broadcast
AFRL	Air Force Research Laboratory
AFRL/MNG	Air Force Research Laboratory Munitions Group
AFWA	Air Force Weather Agency
ANN	artificial neural networks
AOML	Atlantic Oceanographic and Meteorological Laboratory
ARW-WRF	Advanced Research WRF model
ASOS	Automated Surface Observing System
ASTM	American Standards Testing and Materials
AT&D	atmospheric transport and dispersion
ATD	Algorithms for the Threat Detection
AUC	area under the curve
AXBT	Airborne eXpendable BathyThermograph
BSS	blind source separation
CALJET	California Land-falling Jets Experiment
CBRNE	chemical, biological, radiological, nuclear, and enhanced conventional
CCD	charge-coupled device
CDF	Collider Detector at FermiLab
CEC	Cooperative Engagement Capability
CFD	computational fluid dynamics
CO_2	carbon dioxide
CPPCA	combinatorial probabilistic PCA
CUSUM	cumulative sum
D*	Dynamic A* (algorithm)
DAG	directed acyclic graph
DAI-DAO	data input-data output
DAI-DEO	data input-decision output
DAI-FEO	data input-feature output
DARPA	Defense Advanced Research Projects Agency
DART	Data Assimilation Research Testbed

DEI-DEO	decision input-decision output
digiPID	digital photoionization detector
DOAS	differential optical absorption spectroscopy
DoD	Department of Defense
DrEnKF	"dressed" ensemble Kalman filter
D-S	Dempster-Shafer
DTRA	Defense Threat Reduction Agency
DWL	Doppler wind lidar
ECMWF	European Centre for Medium-Range Weather Forecasts
EED	early event detection
EKF	extended Kalman filter
ELINT	electronic intelligence
EMS	emergency medical service
EnKF	ensemble Kalman filter
EO	electro-optical
ETKF	ensemble transform Kalman filter
EULAG	EUlarian semi-LAGrangian
FAL	frequency agile lidar
FASTEX	Fronts and Atlantic Storm-Track Experiments
FEI-DEO	feature input-decision output
FEI-FEO	feature input-feature output
FermiLab	Fermi National Accelerator Laboratory
FFT-07	FUSION Field Trial 2007
FN	false negative
FNMOC	Fleet Numerical Meteorology and Oceanography Center
FP	false positive
FSO	forecast sensitivity to observations
FTIR	Fourier transform infrared
FUSION	Fusing Sensor Information from Observing Networks
GDAS	Global Data Assimilation System
GFDL	Geophysical Fluid Dynamics Laboratory
GPS	global positioning system
GSI	gridpoint statistical interpolation
HEDAS	Hurricane Ensemble Data Assimilation System
HgCdTe	mercury cadmium telluride
HIS	hyperspectral imaging
HWD	Hurricane Research Division
HWRF	Hurricane Weather Research and Forecasting
ICA	Independent Component Analysis
ICD	implantable cardioverter defibrillator
IFF	identification friend or foe
ILI	influenza-like illness
IMM	interacting multiple model

IMS	ion mobility spectrometer
IP	internet protocol
IR	infrared
ISR	intelligence, surveillance, and reconnaissance
JBPDS	Joint Biological Point Detection System
JDL	Joint Directors of Laboratories
JEM	Joint Effects Model
JHAPL	Johns Hopkins Applied Physics Laboratory
JLSCAD	Joint Lightweight Standoff Chemical Agent Detector
JPEG	Joint Photographic Experts Group
JSTO-CBD	Joint Science and Technology Office for Chem-Bio Defense
LEKF	local ensemble Kalman filter
LES	large eddy simulation
LETKF	local ensemble transform Kalman filter
LLE	local linear embedding
LMM	linear mixing model
LWIR	long-wave infrared
MAB	multi-armed bandit
MAP	maximum a posteriori
MB	megabyte
MC	Monte Carlo event generator and simulator
MDP	Markov decision processes
MDS	multidimensional scaling
MTI	moving target indicator
MuSICA	Multi-Sensor Integrated Conflict Avoidance
MVOI	multivariate optimal interpolation
MVU	maximum variance unfolding
NAVOCEANO	Naval Oceanographic Office
NCAR	National Center for Atmospheric Research
NCEP	National Centers for Environmental Prediction
NGA	National Geospatial Intelligence Agency
NHC	National Hurricane Center
NMF	non-negative matrix factorization
NOAA	National Oceanic and Atmospheric Administration
NORPEX	North-Pacific Experiment
N-P	Neyman-Pearson
NPV	negative predictive value
NRL	Naval Research Laboratory
NSF	National Science Foundation
NSR	normal sinus rhythm
NWS	National Weather Service
OODA	observe, orient, decide, act
OTC	over-the-counter

PCA	principal component analysis
Pd	probability of detection
Pfa	probability of false alarm
POMDP	partially-observable Markov decision processes
PPCA	probabilistic PCA
PPV	positive predictive value
PR	precision-recall
PWIDS	Portable Weather Instrumentation Data System
RDECOM	Research, Development and Engineering Command (U.S. Army)
REAL	Raman-shifted eye-safe aerosol lidar
RED	Random Early Detection
RIP	Running in Place
RMS	root mean square
ROA	remotely operated aircraft
ROC	receiver-operating characteristic
S&T	science and technology
SA	situational awareness
SAA	sense and avoid
SAR	synthetic aperture radar
SCM	successive corrections method
SEIK	singular ‘‘evolutive’’ interpolated Kalman
SIGINT	signals intelligence
SNR	signal-to-noise ratio
SOM	self-organizing map
SPC	statistical process control
SPCA	sensible PCA
SST	sea surface temperature
STE	source term estimation
SVM	support vector machine
SVT	supra ventricular tachycardia
SWOrRD	Swept Wavelength Optical Resonant Raman Detector
TB	terabyte
TCAS	Traffic Alert and Collision Avoidance System
THORPEX	The Observing System Research and Predictability Experiment
TLM	tangent linear model
TN	true negative
TP	true positive
UAV	unmanned aerial vehicle
UV	ultraviolet
UVIC	ultraviolet ion collector
V&V	validation and verification
VEAA	vision-enabled autonomous agents

VS-IMM	Variable Structure IMM
VT	ventricular tachycardia
VTHREAT	Virtual Threat Response Emulation and Analysis Testbed
WRF	Weather Research and Forecasting
WSARE	What's Strange about Recent Events
WSR	Winter Storm Reconnaissance
XFT	eXtremely Fast Tracker

(This page is intentionally blank.)

Appendix C
List of Illustrations

Figures

Tables

(This page is intentionally blank.)

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YY)	2. REPORT TYPE	3. DATES COVERED (FROM – TO)
July 2013	Final	September 2012 - April 2013

4. TITLE AND SUBTITLE

Review of Methods and Algorithms for Dynamic Management of CBRNE Collection Assets

5A. CONTRACT NO.

DASW01 04 0003

5B. GRANT NO.

5C. PROGRAM ELEMENT NO(S).

6. AUTHOR(S)

Wilson, A. G.; Cazares, S. M., Papadantonakis, K. M., Avery, M. R., Cartier, J. F., Dolph, P. A., Fregeau, J. M., Holzer, J. R., Morrison, K. A., Nunes, S. M., Renn, S. R., Snyder, J. A., Spencer, K. M.

5D. PROJECT NO.

5E. TASK NO.

DC-1-2607.02

5F. WORK UNIT NO.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Institute for Defense Analyses
4850 Mark Center Drive
Alexandria, VA 22311-1882

8. PERFORMING ORGANIZATION REPORT NO.

IDA Paper P-4995

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Operations Research, Modeling, & Analysis Office, J9ISM
Information Sciences and Applications Department
Defense Threat Reduction Agency (DTRA)
8725 John J. Kingman Rd. Stop 6201
Ft Belvoir, VA 22060

10. SPONSOR'S / MONITOR'S ACRONYM(S)

DTRA

11. SPONSOR'S / MONITOR'S REPORT NO(S).

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

Alyson G. Wilson, Project Leader

14. ABSTRACT

The Defense Threat Reduction Agency (DTRA) provides reachback support to warfighters combating threats from chemical, biological, radiological, and nuclear weapons. DTRA is actively developing technologies to enable and support real-time integration of data from heterogeneous networks of CBRN (and non-CBRN) sensors positioned in and around an area of operations with physics-based modeling capabilities. The integrated toolset being pursued by DTRA must be applicable to a wide range of operational scenarios including direct force protection, targeted search, long-term threat behavior monitoring and wide area search. These integrated systems of sensors and models will require the development of automated methods (algorithms) for combining sensor information with physics models to perform critical functions such as threat detection, classification, identification, or localization. Development of algorithms that perform supporting functions such as data triage, null hypothesis generation, and reallocation of sensing resources will also be required. Algorithm development is required in order to ensure that the analytic reachback capabilities keep pace with developments in network, sensor, and computation capabilities. This report reviews the current state-of-the-art in algorithms and supporting methodologies and suggests possible directions for future development.

15. SUBJECT TERMS

algorithm, multi-sensor fusion, classification, anomaly detection, sensor management, dimension reduction

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NO. OF PAGES	19A. NAME OF RESPONSIBLE PERSON
A. REPORT	B. ABSTRACT	C. THIS PAGE	Unlimited	150	Dr. Andrew CK Wiedlea
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED			19B. TELEPHONE NUMBER (INCLUDE AREA CODE) (703) 767-3166