Chapter 3

Data Fusion Algorithms and Architectures

Sensor and data fusion are used in applications ranging from Earth resource monitoring, weather forecasting, and vehicular traffic control to military target classification and tracking. Data fusion and its objectives based on a model developed for the U.S. Department of Defense are discussed in this chapter. The model divides data fusion into low-level and high-level processes. The low-level processes support preprocessing of data; target detection, classification, and identification; and target tracking. High-level processes support situation and threat assessment and fusion process refinement. Various classes of algorithms have been developed to implement target detection, classification, and track estimation fusion. In addition, several types of data fusion architectures exist for combining sensor data to support the requirements of the data fusion model. The architectures are differentiated by the amount of processing applied to the sensor data before transmission to the fusion process, resolution of the data that are combined, and the location of the data fusion process. The chapter concludes by addressing several concerns associated with the registration of multisensor data. These issues encompass dissimilar sensor footprint sizes, signal generation phenomena, and uncertainty in the location of the sensors.

3.1 Definition of data fusion

In an effort to encourage the use of sensor and data fusion to enhance (1) target detection, classification, identification, and tracking and (2) situation and threat assessment in real time with affordable, survivable, and maintainable systems, the Assistant Secretary of Defense for C^3I (Command, Control, Communications, and Integration) empowered the Joint Directors of Laboratories Data Fusion Subpanel (JDL DFS), now called the Data Fusion Group, to codify data fusion terminology and improve the efficiency of data fusion programs through the exchange of technical information.[1] Acting on this directive, the Office of Naval Technology (ONT) chartered a group, the Data Fusion Development Strategy (DFDS) Panel, to devise a plan for guiding future ONT investment in data fusion.[2] The results of their activity form the basis for the objectives and functional description of data fusion presented here. Their definition of data fusion was enhanced by Waltz and Llinas, who added detection to the functions performed by data fusion and replaced the estimation of position by the estimation of state "to include the broader concept of kinematic state (e.g., higher order derivatives such as velocity) as well as other states of behavior (e.g., electronic state, fuel state)."[3] The resulting definition of data fusion is:

> *A multilevel, multifaceted process dealing with the automatic detection, association, correlation, estimation, and combination of data and information from single and multiple sources to achieve refined position and identity estimates, and complete and timely assessments of situations and threats and their significance.*

The IEEE Geoscience and Remote Sensing Society Data Fusion Technical Committee produced an alternative definition of data fusion as:

> *The process of combining spatially and temporally indexed data provided by different instruments and sources in order to improve the processing and interpretation of these data.*

The goals of data fusion are realized through a five-level hierarchy of processing as shown in Figure 3.1. The processing that occurs at the different data fusion levels is described as follows:

- Level 0 processing: source preprocessing to address process estimation and processor computational and scheduling requirements by normalizing, formatting, ordering, batching, and compressing input data.

- Level 1 processing: achieves refined position and identity estimates by fusing individual sensor position and identity estimates.

- Level 2 processing: assists in complete and timely hostile or friendly military situation assessment. More generally, Level 2 processing involves the relations among the elements being aggregated. The relations may be physical, organizational, informational, or perceptual as appropriate to the need.[4]

- Level 3 processing: a prediction function that assists in complete and timely force-threat assessment using inferences drawn from Level 2 associations. Level 3 fusion estimates the outcome of various plans as they interact with one another and with the environment.

- Level 4 processing: achieves improved results by continuously refining estimates and assessments through planning and control, which includes evaluating the need for additional sources of information, assigning tasks to available resources, or modifying the fusion process itself.

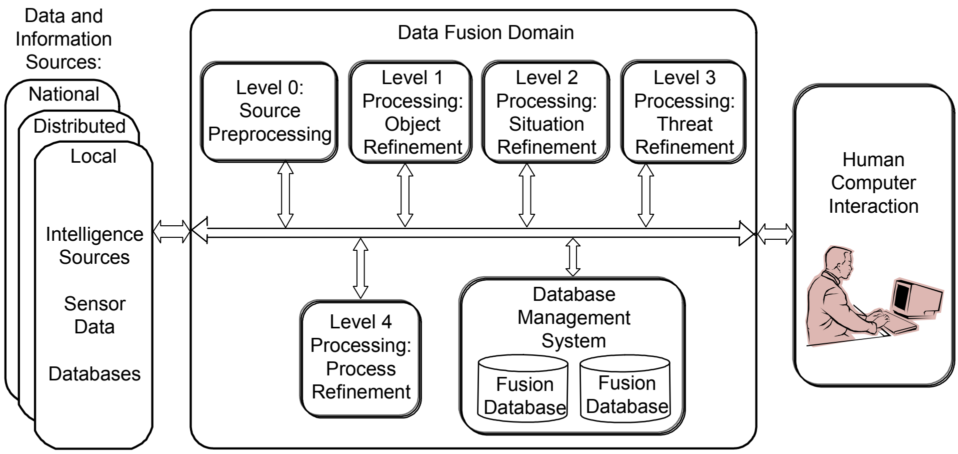


Figure 3.1 Data fusion model showing processing levels 0, 1, 2, 3, and 4.

Data gathered from all appropriate sources, including real-time sensor information, intelligence, maps, weather reports, friendly or hostile status of targets, threat level of targets (e.g., immediate, imminent, and potential), prediction of probable intent and strategies of the threatening targets, and information from other databases, are input to the fusion domain as illustrated on the left of Figure 3.1. The data may be subject to preprocessing or pass directly into one of the other fusion levels. A significant amount of information from external databases is usually needed to support the Level 2 and 3 fusion processes. Interrelationships in Levels 1 through 3 fusion processes are illustrated in Figure 3.2. In some applications such as missile tracking, target detection, classification, and tracking occur simultaneously rather than in separate paths as displayed in Figure 3.2.

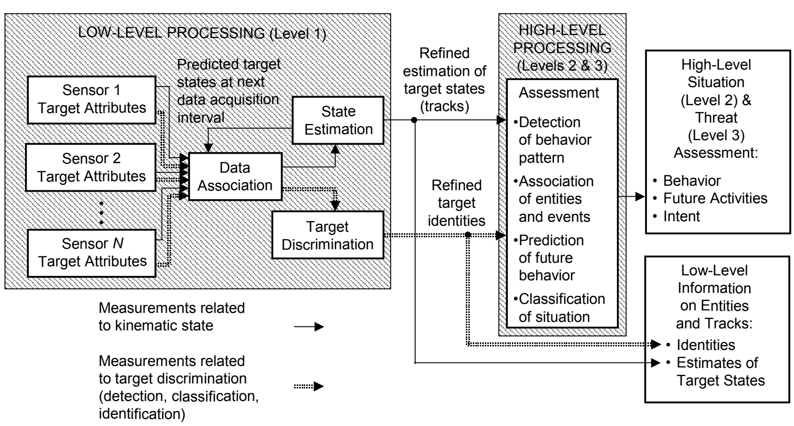


Figure 3.2 Data fusion processing levels 1, 2, and 3. (Adapted from E. Waltz and J. Llinas, *Multisensor Data Fusion*, Artech House, Norwood, MA [1990].)

Level 5 processing, called user refinement, has been proposed to better focus issues related to human processing of fused information, e.g., when automatic target recognition or other computerized analyses are not paramount. Level 5 addresses adaptive determination of (1) who queries and has access to information and (2) which data are retrieved and displayed to support cognitive decision making and action taking. Level 5 processing would appear at the right of the Figure 3.1 architecture as part of human-computer interaction.[5]

3.2 Level 1 processing

Level 1 processing is the low-level processing that results in target track estimation and target discrimination.[6] There is an actual hierarchy of discrimination that, from lowest to highest, encompasses detection, orientation, classification (also called recognition in the older literature), and identification. The interpretation of these terms is shown[7–9] in Table 3.1. The ability to achieve a given level of discrimination depends on the resolution of the sensor and the signal-to-noise ratio at the input to the sensor. These parameters may be traded off against each other to satisfy detection, classification, and identification requirements.[8–10]

Table 3.1 Object discrimination categories.

Category	Interpretation
Detection	Object is present
Orientation	Object is discerned as approximately symmetric or asymmetric and its orientation is determined
Classification	Class to which object belongs is discerned (e.g., building, truck, tank, man, trees, field)
Identification	Object is described to limit of an observer's knowledge (e.g., motel, pickup truck, T-62 tank, M-105 howitzer, soldier)

Sensor outputs are combined through data association processes to produce the desired object or target discrimination level and target position estimate and track. The fusion algorithm used for the detection and classification process need not be the same as that used for track estimation and prediction. For example, a fusion algorithm that accepts highly processed data containing each sensor's best target discrimination estimate can be the optimal one to use for the detection and classification problem when each sensor responds to different signature-generation phenomena. But another fusion algorithm that accepts minimally processed data from more than one sensor and then analyzes and associates these data to form tracks are optimal for obtaining the most accurate track position estimates.

An overview of some 100 articles dealing with applications of information fusion, goals, system architectures, and mathematical tools has been compiled by Valet, Mauris, and Bolon.[11] Their literature survey addresses the selection of data and sensors that provide inputs to fusion systems, mathematical representation of the data and methods to combine them in an optimal way, and choice of output data format to enable easy interpretation of results and their further treatment.

3.2.1 Detection, classification, and identification algorithms for data fusion

A taxonomy for detection, classification, and identification algorithms used in Level 1 processing is shown in Figure 3.3.[2,3,12–14] The major algorithm categories are physical models, feature-based inference techniques, and cognitive-based models. Other mathematical approaches for data fusion, not shown in the figure, have been developed in recent years. These include random set theory, conditional algebra, and relational event algebra.[15] Random set theory deals with random variables that are sets rather than points. Goodman et al. use random set theory to reformulate multisensor, multitarget estimation problems into single-sensor, single-target problems.[15] They also apply the theory to incorporate ambiguous evidence (e.g., natural language reports and rules) into multisensor, multitarget estimation, and to incorporate various expert system methods (e.g., fuzzy logic and rule-based inference) into multisensor, multitarget estimation. Conditional event algebra is a type of probabilistic calculus suited for contingency problems such as knowledge-based rules and contingent decision making. Relational event algebra is a generalization of conditional event algebra that provides a systematic basis for solving problems involving pooling of evidence.

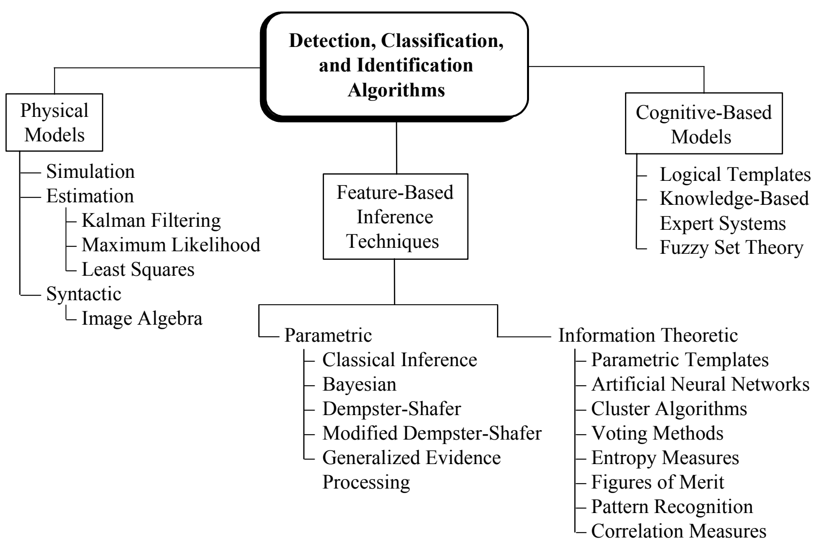


Figure 3.3 Taxonomy of detection, classification, and identification algorithms. [2, 3, 12–14]

Physical models. Physical models replicate object discriminators that are easily and accurately observable or calculable. Examples of discriminators are radar cross section as a function of aspect angle; infrared emissions as a function of vehicle type, engine temperature, or surface characteristics such as roughness, emissivity, and temperature; multispectral signatures; and height profile images. Table 3.2 lists feature categories used in developing physical models, and representative physical features and other attributes of the categories.[14]

Table 3.2 Feature categories and representative features used in developing physical models.

Feature Category	Representative Features	Other Attributes
Geometrical	Edges, lines, line widths, line relationships (e.g., parallel, perpendicular), arcs, circles, conic shapes, size of enclosed area	Represents the geometric size and shape of objects Man-made objects tend to exhibit regular geometric shapes with distinct boundaries
Structural	Surface area; relative orientation; orientation in vertical and horizontal ground plane; juxtaposition of planes, cylinders, cones	Develops a larger scale and contextual view of image segments
Statistical	Number of surfaces, area and perimeter, moments, Fourier descriptors, mean, variance, kurtosis, skewness, entropy	Used at local and global image levels to characterize image data
Spectral	Color coefficients, apparent blackbody temperature, spectral peaks and lines, general spectral signature	Man-made objects tend to possess distinct infrared spectral signatures
Time domain	Pulse characteristics (rise and fall times, amplitude), pulse width, pulse repetition interval, moments, ringing and overshoot, relationship of pulses to ambient noise floor	Selection of time-domain features versus frequency-domain features depends on transmitted waveform and received signal characteristics Less than 100-percent duty cycle signals favor time-domain analysis
Frequency domain	Fourier coefficients, Chebyscheff coefficients, periodic structures in frequency domain, spectral lines and peaks, pulse shape and other characteristics, forced features (e.g., power spectral density of signal raised to N^{th} power)	Information is analogous to that from features in the time domain 100 percent duty cycle signals favor frequency-domain analysis
Hybrid	Wavelets, Wigner-ville distributions, cyclostationary representations	Useful for signals in which both time and frequency are important

Physical models estimate the classification and identity of an object by matching modeled or prestored target signatures to observed data as shown in Figure 3.4. The signature or imagery gathered by a sensor is analyzed for preidentified physical characteristics or attributes, which are input into an identity declaration process. Here the characteristics identified by the analysis are compared with stored physical models or signatures of potential targets and other objects. The stored model or signature having the closest match to the real-time sensor data is declared to be the correct identity of the target or object.

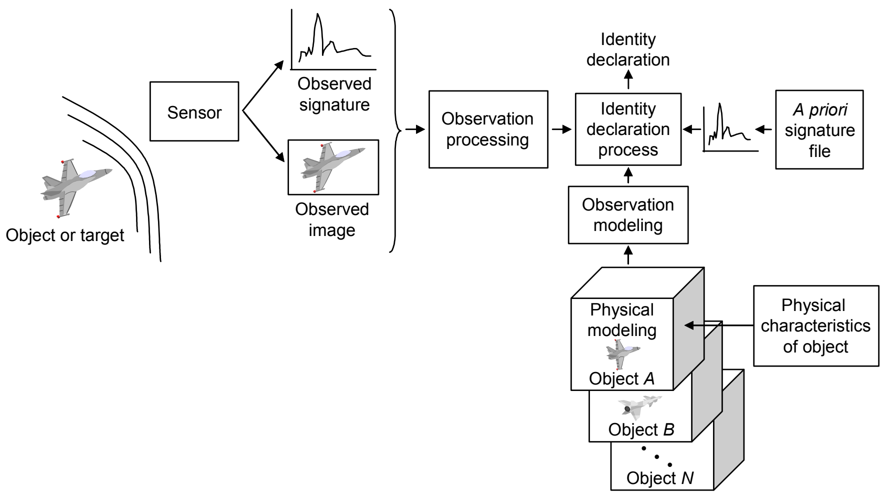


Figure 3.4 Physical model concept.

Physical modeling techniques include simulation, estimation, and syntactic methods. Simulation is used when the physical characteristics to be measured can be accurately and predictably modeled. Estimation processes, such as Kalman filtering, maximum likelihood, and least squares approximation, are similar to the state estimation and tracking algorithms described in Section 3.2.2. The syntactic methods, although listed under physical models, are described later as part of pattern recognition, a subset of information theoretic techniques.

An application of physical modeling based on laser-radar height-profile imagery is illustrated in Figure 3.5. The profile of a shrub and a tank are shown in the left image. The horizontal line passing through the turret of the tank identifies one scan or one profile slice through the image. The plot on the right represents the height of the features detected by the particular scan-line. If the scan-line were lowered to pass through the gun barrel of the tank, a height representing the barrel would be seen in the profile slice data.

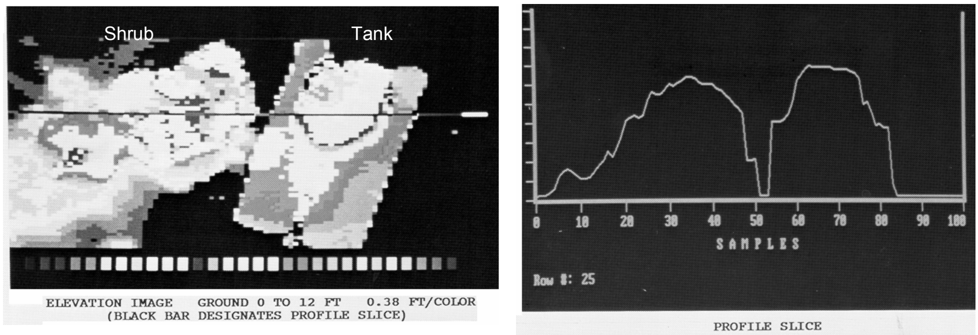


Figure 3.5 Laser radar imagery showing shapes of man-made and natural objects. (Photographs courtesy of Schwartz Electro-Optics, Orlando, FL.)

When many height profiles produced by line scans through different regions of the laser imagery are compared, naturally occurring objects tend to have more random shapes than man-made objects. Thus, an object identification algorithm using shape as a classification criterion can be developed to differentiate between natural objects such as ground clutter (e.g., shrubs, boulder field, and trees) and man-made objects or potential targets having known height profiles.

Feature-based inference techniques. Feature-based inference techniques perform classification or identification by mapping data, such as statistical knowledge about an object or recognition of object features, into a declaration of identity. Feature-based algorithms may be further divided into parametric and information theoretic techniques (i.e., algorithms that have some commonality with information theory) as depicted in Figure 3.3.

Parametric techniques

Parametric classification directly maps parametric data (e.g., features) into a declaration of identity. Physical models are not used. Parametric techniques include classical inference, Bayesian inference, Dempster-Shafer evidential theory, modified Dempster-Shafer methods, and generalized evidence processing.

Classical inference gives the probability that an observation can be attributed to the presence of an object or event, given an assumed hypothesis. Its major disadvantages are: (1) difficulty in obtaining the density function that describes the observable used to classify the object, (2) complexities that arise when multivariate data are encountered, (3) its capability to assess only two hypotheses at a time, and (4) its inability to take direct advantage of *a priori* and likelihood probabilities.

Figure 3.6 illustrates a problem where classical inference is called upon to determine whether the detected radar illumination is from a Class 1 radar with low pulse repetition interval (PRI) or a Class 2 radar with higher PRI. A critical value of the PRI, designated as PRI_c, is selected based on acceptable Type 1 and Type 2 errors (defined in the figure). In this example, the null hypothesis H_0 (the statement being tested) is equated to "The observed PRI is less than PRI_c (i.e., it belongs to a Class 1 radar)" and the alternative hypothesis H_1 (the statement suspected of being true) to "The observed PRI is greater than or equal to PRI_c (i.e., it belongs to a Class 2 radar)." The probability that the observed PRI belongs to a Class 1 radar is calculated using a standardized random variable and the known probability density function that describes the PRI. The probability, computed assuming H_0 is true, that the standardized random variable assumes a value as extreme or more extreme than that actually observed is called the *P*-value of the test. The smaller the *P*-value, the stronger the evidence against H_0 provided by the data. If the *P*-value is as small as or smaller than α, the data are said to be statistically significant at level α. That is, the data give evidence against H_0 such that H_0 occurs no more than α percent of the time.

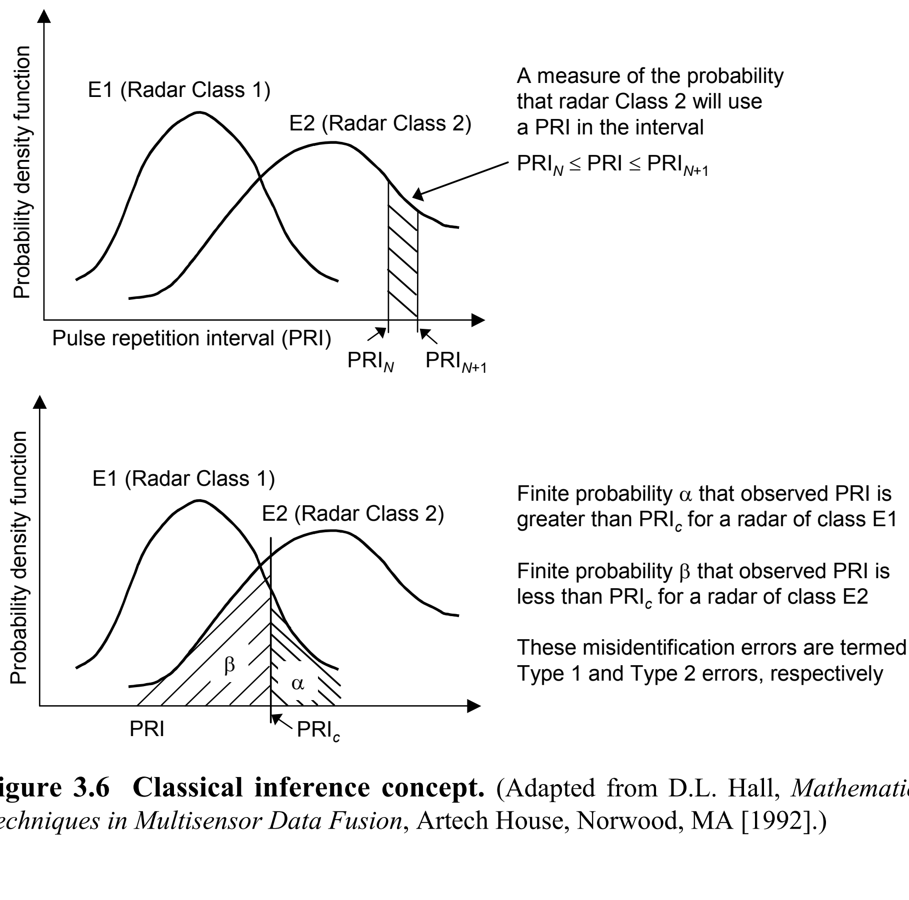


Figure 3.6 Classical inference concept. (Adapted from D.L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Norwood, MA [1992].)

The significance-level α of any fixed level test is equal to the probability of the Type 1 error. Thus, α is the probability that the test will reject hypothesis H_0 when H_0 is in fact true. The probability that a fixed-level α significance test will reject H_0 when a particular alternative value of the parameter is true is called the power of the test against that alternative. Thus, the power is equal to 1 minus the probability of a Type 2 error for that alternative. These concepts are developed further in Chapter 4.

Bayesian inference resolves some of the difficulties with classical inference. It updates the *a priori* probability of a hypothesis given a previous likelihood estimate and additional observations and is applicable when more than two hypotheses are to be assessed.[14,16] The disadvantages of Bayesian inference include: (1) difficulty in defining the prior probabilities and likelihood functions, (2) complexities that arise when multiple potential hypotheses and multiple conditionally dependent events are evaluated, (3) mutual exclusivity required of competing hypotheses, and (4) inability to account for general uncertainty. Bayesian inference is discussed further in Chapter 5.

Dempster-Shafer evidential theory generalizes Bayesian inference to allow for uncertainty by distributing support for a proposition (e.g., that an object is of a particular type) not only to the proposition itself, but also to the union of propositions (disjunctions) that include it and to the negation of a proposition. Any support that cannot be directly assigned to a proposition or its negation is assigned to the set of all propositions in the hypothesis space (i.e., uncertainty). Support provided by multiple sensors for a proposition is combined using Dempster's rule. Bayesian and Dempster-Shafer produce identical results when all singleton propositions are mutually exclusive and there is no support assigned to uncertainty. A requirement of the Dempster-Shafer method is the need to define processes in each sensor that assign the degree of support for a proposition. Disadvantages of the method include the inability to make direct use of prior probabilities when they are known and the counterintuitive output sometimes produced when support for conflicting propositions is large. Several methods have been proposed to modify Dempster's rule through the use of probability transformations that better accommodate conflicting beliefs[17] and, in some cases, through the use of prior knowledge and spatial information.[18–24] Data fusion using Dempster-Shafer evidential theory and examples of its application are developed in more detail in Chapter 6.

Generalized evidence processing (GEP) allows a Bayesian decision process to be extended into a multiple hypothesis space (called the frame of discernment in Dempster-Shafer evidential theory). Evidence that supports nonmutually exclusive propositions can be combined to arrive at a decision by minimizing a Bayesian risk function tying probability masses to likelihood ratios, or equivalently, by maximizing a detection probability for fixed *a priori* miss and false alarm probabilities.[25–28]

In GEP, the evidence collected by the sensors determines the probability mass associated with a decision. The probability mass assignments are conditioned on each postulated hypothesis either through Bayesian reasoning or belief functions as in Dempster-Shafer theory. In the Bayesian approach, the probability mass $m_n^i(d_j)$ assigned by a sensor n to a decision j is equal to the conditional probability of the decision given a hypothesis i. Probability mass assignments are optimal in that they minimize total risk.

As an example, consider two hypotheses H_0 and H_1 that are under test. The probability space is partitioned into two regions according to events $\{\omega = H_0\}$ and $\{\omega = H_1\}$ with probabilities $P_{H_0} \geq 0$ and $P_{H_1} \geq 0$, such that $P_{H_0} + P_{H_1} = 1$. Let the three decisions d_0, d_1, and d_2 (equal to $d_0 \cup d_1$) constitute a frame of discernment, where the decisions correspond to the propositions "H_0 true," "H_1 true," and "H_0 or H_1 true," respectively. Decision d_2 denotes the inability of the decision maker to gather conclusive evidence on the true nature of the hypothesis. The evidence is associated with the set of admissible decisions unconditionally using a likelihood ratio test that minimizes the Bayes risk function. The decision with the minimum Bayes risk is selected. The set of decisions need not be the same as the set of hypotheses as in the above example. Thus evidence combining and decision making in GEP are separate concepts.[26]

If the objective of the fusion process is to minimize a generalized Bayesian risk, evidence combining in GEP theory is performed using likelihood ratios and pair-wise multiplication of probability masses. When the sensor observations are conditionally independent (i.e., conditioned on the hypotheses) and there are two hypotheses, the likelihood ratio for hypothesis H_1 is equal to the pairwise multiplication of the probability mass from each sensor for each decision pair, conditioned on hypothesis H_1, divided by the pairwise multiplied probability mass from each sensor for each decision pair, conditioned on hypothesis H_0. Under each hypothesis, evidence-combining is performed by summing the probabilities whose likelihood ratios fall in specific intervals defined by the optimization criterion that minimizes the Bayes risk. For the three-decision example (i.e., $d = d_0$, d_1, d_2) and two sensors, evidence combining under each hypothesis H_i, $i = 0$, 1 is structured as

$$m_1^i(d_k)\, m_2^i(d_l) \rightarrow \text{decision } d_j \text{ if } \frac{m_1^1(d_k)m_2^1(d_l)}{m_1^0(d_k)m_2^0(d_l)} \in F_j\,, \tag{3-1}$$

where F_j is the decision region that favors decision d_j.

For the binary hypothesis example, the decision regions are defined with simple thresholds. Accordingly Eq. (3-1) simplifies to

$$m_1^i(d_k)\ m_2^i(d_l)\ \rightarrow \text{decision } d_j \text{ if } t_j < \frac{m_1^1(d_k)m_2^1(d_l)}{m_1^0(d_k)m_2^0(d_l)} < t_{j+1} \tag{3-2}$$

for all k, l, and j, where t_j are the thresholds of the likelihood ratios associated with the different decisions that minimize the Bayes risk function.

When more than two hypotheses are postulated, the conditional probability, calculated either through Bayesian reasoning or belief functions, is given by the likelihood ratio Λ as the product of terms formed by the conditional probability of a decision given hypothesis H_i divided by the conditional probability of a decision given hypothesis H_0, where the number of terms equals the number of sensors in the fusion system. The likelihood ratio is thus:[26,29]

$$\Lambda_i(\boldsymbol{d}) = \prod_{j=1}^{N} \frac{P(d_j \mid H_i)}{P(d_j \mid H_0)} \text{ for } i = 1, 2, \ldots, q-1, \tag{3-3}$$

where

N = number of sensors in the fusion system,

d_j = decision of the j^{th} sensor, and

q = number of tested hypotheses.

Sensor evidence is merged by forming the product of the joint probability distribution of the likelihood ratios for each hypothesis as

$$\prod_{j=1}^{N} P(\Lambda_1, \Lambda_2, \ldots, \Lambda_{q-1} | H_i)$$

for i = 1, 2, …, q–1 and j = 1, 2, …, N. When the sensor decisions are conditionally independent, the joint probability distribution of the likelihood ratios becomes

$$\prod_{j=1}^{N} P(\Lambda_1 | H_i)\, P(\Lambda_2 | H_i) \ldots P(\Lambda_{q-1} | H_i)\ .$$

The evidence is then associated with the admissible decisions unconditionally using a likelihood ratio test or another test that optimizes a performance measure. Thus, the combined evidence is compared with a threshold condition or quantization level to determine which decision is selected. Quantization levels, which can be defined at the data fusion processor level or at the individual sensor level, are equal to distinct values of the Bayes risk. In the case of the two hypotheses case, the Bayes risk is equal to the likelihood ratio formed by dividing the probability distribution function for H_1 by the probability distribution function for H_0.[29]

GEP diverges from Dempster-Shafer in two ways:

1. Probability-mass assignments may be based on the Bayesian likelihood function, i.e., the conditional probability of observing evidence given that a particular hypothesis is true, although the probability masses can also correspond to the belief functions used in Dempster-Shafer evidential theory;

2. Decisions are selected in a manner that minimizes a risk function.

Information theoretic techniques

Information theoretic techniques transform or map parametric data into an identity declaration. All these methods share a similar concept, namely, that similarity in identity is reflected in similarity in observable parameters. No attempt is made to directly model the stochastic aspects of the observables. The techniques that can be included under this category are parametric templates, artificial neural networks, cluster algorithms, voting methods, entropy-measuring techniques, figures of merit, pattern recognition, and correlation measures.

In *parametric templating*, multisensor or multispectral data acquired over time and multisource information are matched with preselected conditions to determine if the observations contain evidence to identify an entity. Templating can be applied to event detection, situation assessment, and single object identification.[3,14] Figure 3.7 shows an application of parametric templating to the identification of an emitter, whose pulse repetition frequency and pulse width are measured by a sensor. The measured parameters are overlaid on a template such as the one depicted in the lower right portion of the figure. Identification is made when the parameters lay in a region that corresponds to the characteristics of a known device. In this example, the pulse repetition frequency and pulse width of Emitter 1 are characteristic of those of Emitter Class A. Emitter 2's class is undefined, as it does not fall within the boundaries characterized by either the Class A or Class B emitters.

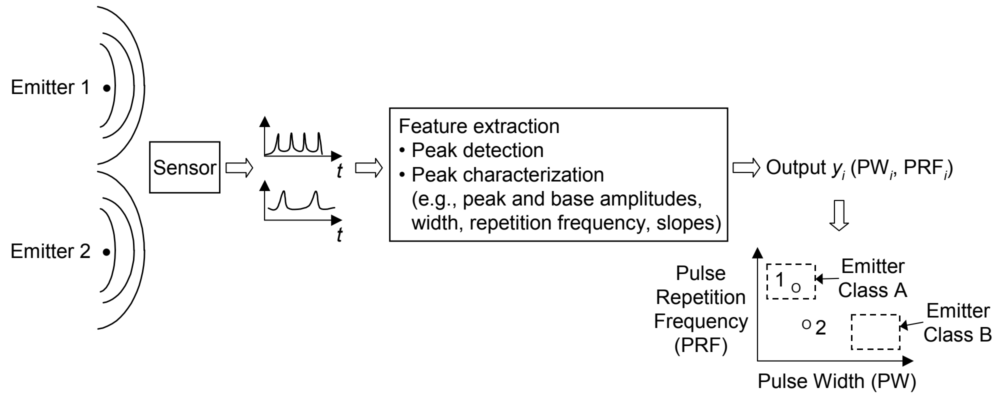


Figure 3.7 Parametric templating concept based on measured emitter signal characteristics. (Adapted from D.L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Norwood, MA [1992].)

An example of parametric templating applied to multispectral or hyperspectral sensor data is given in Figure 3.8. Here the sensors detect the value of the radiance R_i emitted by objects over many spectral bands $\Delta\lambda_i$. The number of bands and spectral bandwidth is dependent on the sensor design. Objects are defined by templates consisting of radiance values for each spectral band in the sensor. The measured radiance values are overlaid on the templates. Identification is made when the measured radiance values over the ensemble of spectral bands correspond to or are best represented by those of a known object.

When an extended object or scene is observed and the sensor is capable of imaging, the radiances in each band are used to identify the particular material or subobject in each sensor pixel or small groups of pixels. After all pixel data are analyzed, an image can be created by adding false color to the particular materials or subobjects of the image.

Artificial neural networks are hardware or software systems that are trained to map input data into selected output categories. The transformation of the input data into output classifications is performed by artificial neurons that attempt to emulate the complex, nonlinear, and massively parallel computing processes that occur in biological nervous systems. Artificial neural networks are discussed in detail in Chapter 7.

Cluster algorithms group data into natural sets or clusters that are interpreted by an analyst to see if they represent a meaningful object category. All cluster algorithms require a similarity metric or association measure that describes the closeness between any two feature vectors, for example, one that represents the input data and one that represents a potential class to which the data belong.

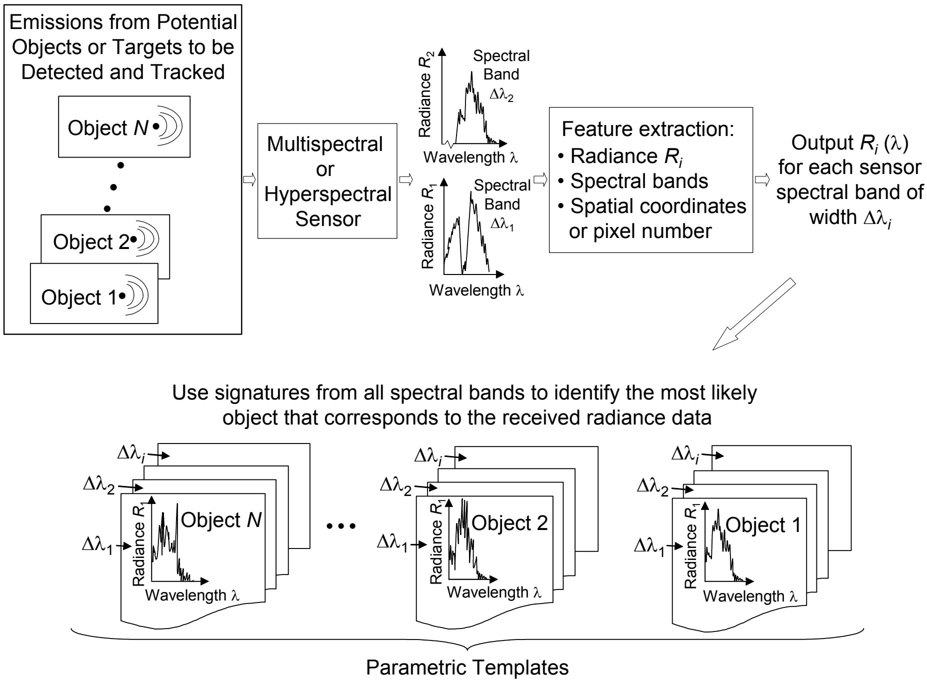


Figure 3.8 Parametric templating using measured multispectral radiance values.

Cluster algorithms operate with five basic steps: (1) selection of sample data, (2) definition of the set of variables or features that characterize the entities in the sample, (3) computation of the similarities among the data, (4) use of a cluster analysis method to create groups of similar entities based on data similarities, and (5) validation of the resulting cluster solution. The application of cluster algorithms may lead to biased results because of the heuristic nature of these algorithms. In general, data scaling, choice of similarity metric and algorithm, and sometimes even the order of the input data may substantially affect the resulting clusters. Hence, application of cluster methods must be judged on their effectiveness and ability to form consistent and meaningful identity clusters.[3,14]

Figure 3.9 depicts one representation of how cluster analysis may be applied. Observations or data acquisition from known objects or targets occur during a training cycle, followed by identification and extraction of features that assist in uniquely classifying the objects or targets of interest. A feature-based classifier operates on the feature vector **Y** and allocates specific regions in the feature space to the objects of interest. When training is complete, unknown objects are observed and the same features are extracted from their signatures. The feature-based classifier then identifies the region in the feature space that best corresponds to the feature vector obtained from the unknown object.

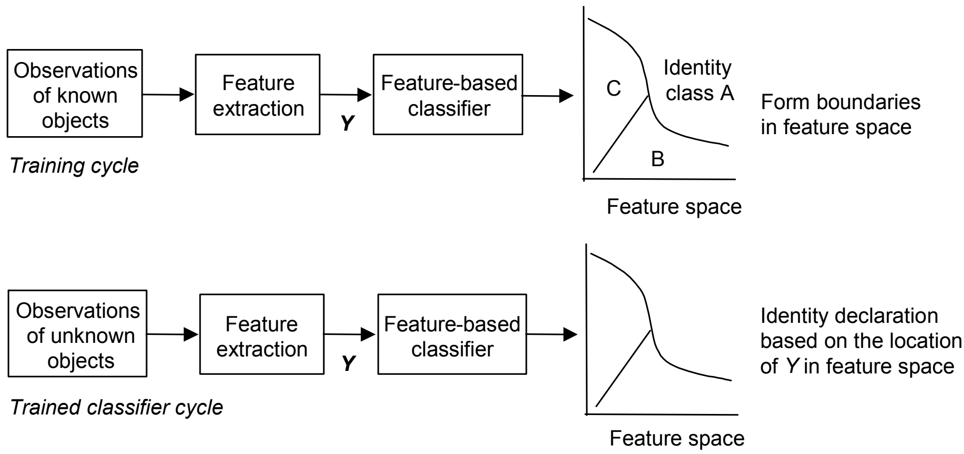


Figure 3.9 Cluster analysis concept. (Adapted from D.L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Norwood, MA [1992].)

Voting methods combine detection and classification declarations from multiple sensors by treating each sensor's declaration as a vote in which majority, plurality, or decision-tree rules are used. Additional discrimination can be introduced via weighting of the sensor's declaration as discussed in Chapter 8 where voting based on Boolean algebra is described.

Entropy measures take their name from communications theory and attempt to measure the importance of the information in a message by its probability of occurrence. Frequently occurring messages or data are of low value, while surprising or rare messages are of higher value. The function that measures the value of information, therefore, has the property that it decreases with increasing probability of receiving the information.

An application of entropy is found in games of Keno, where the player marks some quantity of numbers on a card that contains 80 numbers. An automated and random selection of 20 numbers is made by a machine from among the 80 choices. Payoffs are a function of the number of correct number selections the player has made. Infrequent outcomes are of high value and more frequently occurring events of low or no value. For example, in one game of Keno a $5 bet pays off in 18 ways as shown in Table 3.3. In other Keno games, payoffs are made for correctly picking 1 to 15 numbers.

As an example of applying entropy to multisensor data fusion, consider combining information from two sources that have a numerical measure 1, 4, or 7 assigned to the information value of their data. A larger number denotes more value. Furthermore, suppose the entropy fusion process adds the numbers assigned to the value of the data from each information source. If the sum of the numerical measures is 7 or greater, then the information is considered valuable and is acted upon. Thus, the highest value data from one source or medium value data from each of the sources can initiate an action in this example.

Table 3.3 Keno payoff amounts as a function of number of correct choices.

Play \$5.00	Win Amount	Play \$5.00	Win Amount
0	\$500	11	\$200
1	\$10	12	\$1,200
2	\$5	13	\$5,000
3	\$5	14	\$15,000
4	0	15	\$25,000
5	0	16	\$50,000
6	0	17	\$100,000
7	\$5	18	\$150,000
8	\$10	19	\$200,000
9	\$25	20	\$250,000
10	\$50		

Entropy also finds application in self-organized artificial neural networks, such as the Kohonen model. The parameter to be maximized is the average mutual information between the input vector **X** and the output vector **Y**, in the presence of noise. The average mutual information is equal to the difference between the uncertainty (i.e., entropy) about the system input *before* observing the system output and the uncertainty about the system input *after* observing the system output.[30]

Figures of merit are metrics derived from plausible or heuristic arguments that aid in establishing a degree of association between observations and object identity. They contain flexible sets of algorithms that measure the strength of entity and event relationships. Figure of merit techniques attempt to formulate a relationship among several variables, or as many as possible, in order to improve the association or classification of input data. Sometimes figures of merit are considered a templating approach since they reflect the expected observations, behaviors, logical relationships, and any other basis that profiles an object's identity. Figures of merit also have aspects that are similar to weighted decision formulas.

Pattern recognition concerns the description or classification of data. The three major approaches to pattern recognition are statistical (or decision theoretic), syntactic (or structural), and artificial neural networks. In statistical pattern recognition, a set of characteristic measurements or features are extracted from the input data and used to assign the feature vector to one of c classes. Assuming features are generated by a state of nature, the underlying statistical model represents a state of nature, set of probabilities, or probability density functions that correspond to a particular class.[31] Syntactic pattern recognition is applied when the significant information in a pattern is not merely the presence or absence of numerical values, but rather the interconnections of features that yield structural information. The structural similarity of patterns is assessed by quantifying and extracting structural information using, for example, the syntax of a formally defined language. Typically, syntactic approaches formulate hierarchical descriptions of complex patterns from simpler subpatterns or primitives. Neural computing attempts to mimic the complex, nonlinear, and parallel problem solving processes that occur in biological neural systems.

Pattern recognition is frequently applied to high-resolution, multipixel imagery such as that from a FLIR or high-resolution scanners found on satellites. Features extracted from a FLIR image may consist of temperature gradients, length/width ratios, central moments, and the relative size of subobjects within the boundary of the larger object. Features associated with LANDSAT images are extracted from each pixel of data for each spectral band in the sensor. Frequency domain spectra of MMW signatures also provide features used in statistical pattern recognition algorithms. Features in this case are extracted from the Fourier transformed signal. Schalkoff[31] provides a concise comparison of the attributes of the statistical, syntactic, and neural pattern recognition approaches as shown in Table 3.4.

Correlation measures are derived from weighted combinations of figures of merit. They allow a comparison score or measure of correlation to be calculated for systems that have numerous figures of merit. Thus, the correlation measure represents the total likelihood that two entities are the same.

Cognitive-based models. Cognitive-based models, including logical templates, knowledge-based systems, and fuzzy set theory, attempt to emulate and automate the decision-making processes used by human analysts.

Logical templates

Templating, as the name suggests, is a concept where a predetermined and stored pattern is matched against observed data to infer the identity of the object or to assess a situation. Parametric templates that compare real-time patterns with stored ones can be combined with logical templates derived, for example, from Boolean relationships.[3] Fuzzy logic may also be applied to the pattern matching technique to account for uncertainty in either the observed data or the logical relationships used to define a pattern.

Table 3.4 Comparison of statistical, syntactic, and neural pattern recognition (PR) approaches. (R. Schalkoff, *Pattern Recognition: Statistical, Structural, and Neural Approaches*, John Wiley, NY [1992].)

Attribute	Statistical PR	Syntactic PR	Neural PR
Pattern generation (storing) basis	Probabilistic models	Formal grammars	Stable state or weight array
Pattern classification basis	Estimation/decision theory	Parsing	Based on properties of the neural network
Feature organization	Feature vector	Primitives and observed relations	Neural input or stored states
Typical learning (training) approaches			
Supervised:	Density or distribution estimation	Forming grammars (heuristic or grammatical inference)	Determining neural network system parameters (e.g., weights)
Unsupervised:	Clustering	Clustering	Clustering
Limitations	Difficulty in expressing structural information	Difficulty in learning structural rules	Often little semantic information from the network

Knowledge-based expert systems

Knowledge-based systems incorporate rules and other knowledge from known experts to automate the object identification process. They retain the expert knowledge for use at a time when the human inference source is no longer available. Computer-based expert systems frequently consist of four components: (1) a knowledge base that contains facts, algorithms, and a representation of heuristic rules; (2) a global database that contains dynamic input data or imagery; (3) a control structure or inference engine; and (4) a human-machine interface. The inference engine processes the data by searching the knowledge base and applying the facts, algorithms, and rules to the input data. The output of the process is a set of suggested actions that is presented to the end user.

The knowledge-based system illustrated in Figure 3.10 depicts processed sensor data or imagery as the source of the features that identifies the object or situation.

Three types of rules are listed to assist in correlating information contained in the real-time feature vector with information in the stored knowledge base.

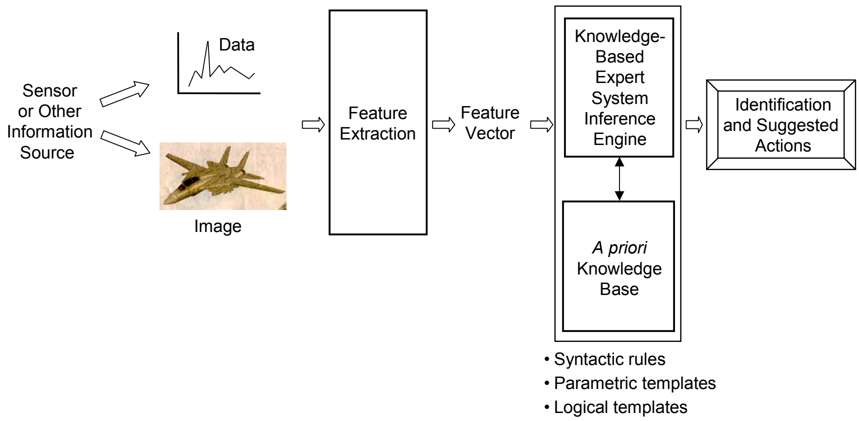


Figure 3.10 Knowledge-based expert system concept.

Syntactic rules are expressed as IF-THEN statements. The IF or antecedent clause states the conditions that must be present for the action specified in the THEN or conditional clause to occur. Expert systems typically rely on binary on-off logic and probability to develop the inferences used in the IF-THEN statements. Parametric templates contain stored data values, images, and other types of information that are associated with known objects or decisions. Logical templates combine the decisions from more than one parametric template using Boolean algebra relationships. The executed object identity or decision is that belonging to the prestored feature vector closest to the vector composed of the real-time feature values.

Fuzzy set theory

Fuzzy set theory opens the world of imprecise knowledge or indistinct boundary definition to mathematical treatment. It facilitates the mapping of system state variable data into control, classification, or other output data. Its use of fuzzy associative memory (also known as production rules) allows a proposition to have a membership value in a given class ranging from 0 (definitely not a member) to 1 (definitely a member). The production rules, which govern the behavior of the system, are in the form of IF-THEN statements. An expert specifies the production rules and fuzzy sets that represent the characteristics of each input and output variable. Fuzzy set theory is intuitively appealing in that it permits uncertainties in knowledge or identity boundaries to be applied to such diverse applications as identification of battlefield threats, target tracking, and control of industrial and automotive processes. Unlike neural networks that sum throughputs, fuzzy systems sum outputs. Chapter 9 contains a detailed discussion of fuzzy set theory, fuzzy logic, and illustrative examples.

3.2.2 State estimation and tracking algorithms for data fusion

A taxonomy for state estimation and tracking algorithms used in Level 1 processing is shown in Figure 3.11.[2,3,12,14] The processes required to perform the tracking function are represented, at the top level, by algorithms that (1) determine the search direction and (2) associate and correlate data and tracks. Association and correlation are further separated into data alignment; data and object correlation; and position, kinematic, and attribute estimation. The majority of this section is concerned with data and object correlation techniques.

Search direction. Direction tracking systems can be sensor (data) driven or target (goal) driven. In sensor-driven systems, target reports (consisting of combinations of range, azimuth, elevation, and range-rate sensor data) initiate a search through the track file for *tracks that can be associated with the reports*. Target-driven systems use a primary sensor for tracking and use the target track to direct other sensors to acquire data or search databases for *reports that can be associated with particular tracks*.

Association and correlation of data and tracks. The proper association and correlation of measurement data and tracks from multisensor inputs ultimately generate optimal central track files. Each file ideally represents a unique physical object or entity. Association and correlation require algorithms that define data alignment, prediction gates, association metrics, data and track association, and position, kinematic, and attribute estimation.

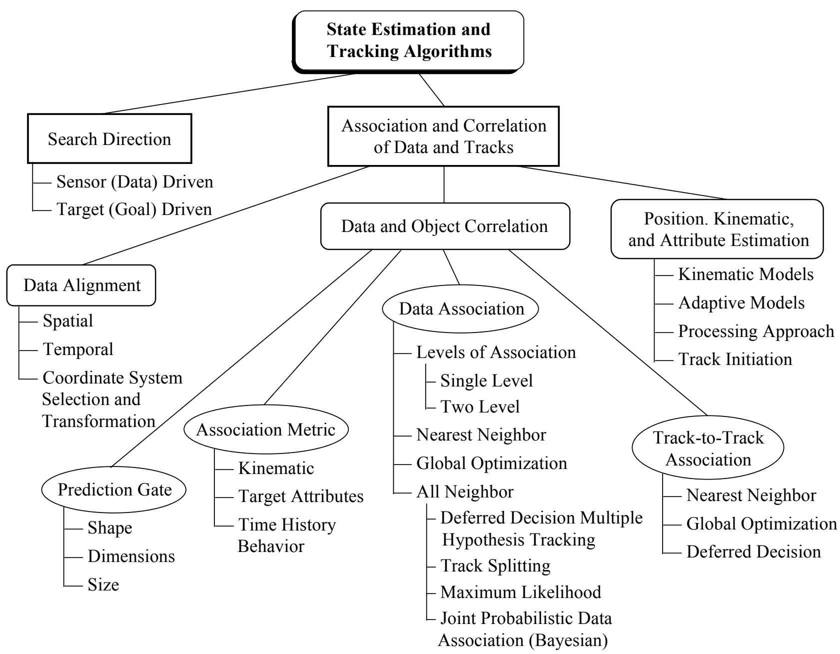


Figure 3.11 Taxonomy of state estimation and tracking algorithms. [2,3,12,14]

Data alignment

Data alignment is performed through spatial and temporal reference adjustments and coordinate system selection and transformations that establish a common space-time reference for fusion processing. Errors introduced by measurement accuracies, coordinate transformations, and unknown target motion are accounted for through the data alignment process.

Data and object correlation

Data and object correlation consist of processes that establish the prediction gate, define the association metric, perform data association, and perform track-to-track association.

Prediction gates

Prediction gates control the association of data sets into one of two categories, namely candidates for track update or initial observations for forming a new tentative track. Data that were originally categorized for track update may later be used to initiate new tracks if they are not ultimately assigned to an existing track. The size of the gates reflects the calculated or otherwise anticipated target position and velocity errors associated with their calculation, sensor measurement errors, and desired probability of correct association.

Association metrics

Association metrics quantify the similarity of the target reports or measurement data. They are also used in track-to-track association to assist in correlating tracks produced by different sensors. Metrics are evaluated using the kinematic parameters (e.g., range, range rate, angle, and position) and target attributes (e.g., temperature, size, shape, and edge structure) that are observed and measured. The metric can be based on spatial distance (e.g., Euclidean distance) or statistical measures of correlation between observations and predictions (e.g., Mahalanobis distance), heuristic functions such as figures-of-merit that use the kinematic and target attribute information, and measures that quantify the realism of an observation or track based on prior assumptions such as track lengths, target densities, or track behavior. Metrics based on spatial distance and statistical measures of correlation are shown in Table 3.5.[14]

Data association

In a multiple target and sensor scenario, data association refers to the statistical decision process that correlates reports (i.e., set of measurement data) from overlapping gates, multiple returns (hits) in a gate, clutter in a gate, and new targets that appear in a gate on successive scans. Thus, data association partitions

Table 3.5 Distance measures.

Metric	Mathematical Expression for One Matrix Element*	Interpretation
Euclidean	$[(y\text{-}z)^2]^{1/2}$	Geometric distance between vectors **Y** and **Z** (square root of vector dot product)
Weighted Euclidean	$[(y\text{-}z)\ w\ (y\text{-}z)^{\mathrm{T}}]^{1/2}$	Euclidean distance weighted by **W**
Minkowski	$[(y\text{-}z)^p]^{1/p}$	Generalized Euclidean distance of order p, where $1 \leq p \leq \infty$
City block	$\|(y\text{-}z)\|$	First order Minkowski distance (also called Manhattan distance)
Mahalanobis	$(y\text{-}z)^{\mathrm{T}} R^{-1} (y\text{-}z)$	Weighted Euclidean distance with weight equal to inverse covariance matrix **R**
Bhattacharyya	$1/8\ (y\text{-}z)^{\mathrm{T}}\{[R_y+R_z]/2\}^{-1}(y\text{-}z) + \frac{1}{2}\ln\{[R_y+R_z]/2\}/\{\|R_y\|^{1/2}\|R_z\|^{1/2}\}$	Generalization of Mahalanobis distance allowing unequal covariance matrices R_y and R_z
Chernoff	$\frac{1}{2}s(1\text{-}s)(y\text{-}z)^{\mathrm{T}}[sR_y+(1\text{-}s)R_z]^{-1}(y\text{-}z) + \frac{1}{2}\ln[\|sR_y+(1\text{-}s)R_z\|]/[\|R_y\|^s\|R_z\|^{1\text{-}s}]$	Generalization of Mahalanobis distance, where $0<s<1$ allows for variation in weighting influence of unequal covariance matrices R_y and R_z; same as Bhattacharyya when $s=\frac{1}{2}$

the measurements into sets that could have originated from the same targets.[32] Association techniques that merge data and tracks from several sources into a single track usually employ either single-level tracking systems or two-level tracking systems.[33] In a single-level tracking system, the data detected by the multiple sensors are transmitted to a single processing node (central-level fusion). Here the data are processed to initiate new tracks and update estimates of existing tracks. Association occurs between each new set of sensor data and the central track file.

Two-level tracking systems have four variants: (1) track-to-track association at the sensors and at a central node; (2) sensor data and track association at a central

* Example: Euclidean distance measure for a data vector of size k is given by $\left[\sum_{i=1}^{k}|y_i - z_i|^2\right]^{1/2}$.

node; (3) sensor data association to form tracks at a central node; and (4) sensor track association at a central node. The first two-level tracking system (known as autonomous or distributed tracking) maintains separate sensor-level and central-level trackers. Each sensor-level tracker independently acquires, initiates, continues, and drops tracks using its own data. In addition, track-to-track association is performed at a single node to form a central track file.

The second two-level system uses either sensor data or sensor tracks to initiate and maintain a central track file. Track-to-track correlation is initiated at a central node by associating the sensor tracks to form a central track file. Sensors send information as data rather than tracks to the central node, where they are associated with the existing tracks or are used to establish new tracks.

The third two-level system uses sensor data to form tracks in a central processor. The predicted gates for the next track update are sent from the central processor back to the sensors in order to cue their search area and velocity for the next update.

A fourth two-level system associates sensor tracks in a central processor using track-to-track correlation to eliminate redundant tracks and to assign future reporting responsibility to the sensor with the best track (based on a covariance calculation or track quality assessment, for example). The data to be fused may originate from other command and control nodes or centers, as well as from sensors under common command and control.[34]

Data association can be performed with nearest-neighbor, global optimization, and all-neighbor techniques. In nearest neighbor, a hard decision is made to pair the input data with the single best track. Several variants of nearest neighbor algorithms are available, including one that uses a Dempster-Shafer formalism to classify the unknown object.[35,36] This approach is of value when the nearest neighbor output provides evidence suggesting the observed object could be a member of a given class, but does not provide 100 percent confidence in that decision. Traditional nearest neighbor rules deteriorate when multiple, closely ranked choices and maneuvering targets are present. One of the methods available to remedy this shortcoming is the Munkres or faster executing JVC (Jonker-Volgenant-Castanon) algorithm, which globally optimize the association of all new data and tracks with any existing tracks.[37,38] An application of the JVC algorithm to the association of direction angle measurements is described in Chapter 10.

All-neighbor association eliminates several of the deficiencies of the nearest-neighbor approach. In one implementation of all-neighbor association, called deferred decision multiple hypothesis tracking, each candidate pairing is considered a viable hypothesis and is retained in the track file until a decision criterion can eliminate or confirm the hypothesis. Final assignment of data is deferred until sufficient information from future scans is available to increase confidence in the hypothesis.[34] When track association is deferred, however, the operator does not see the recommended track until several scans have elapsed. If the tracks are displayed for each scan, then the operator can potentially view multiple tracks, some of which are false, making situation assessment difficult. This deficiency has been overcome with techniques that display only the high-confidence tracks.[39]

A variation of multiple hypothesis tracking, called track splitting, associates each report in the gate with a track, but does not specifically generate "new" tracks, nor compute the probability of correct association. The track splitting technique can be applied when a target maneuver is suspected. In this situation, the expected sensor update data may not be present in the normal gate. Therefore, the gate is enlarged to account for the maximum anticipated target maneuver. If the target is located within the larger gate, then the track is split into two parts, one corresponding to a non-maneuvered target and one to a maneuvered target. The decision to abandon one track or the other is made on the following scan.

Unlike other all-neighbor association techniques, maximum likelihood selects the most likely single set of measurement data for association with a track. Probability density functions are assumed for the target data, target tracks, and the spurious data due to noise, clutter, or decoys. A target is declared present if the likelihood function defined by the product of the probability density functions for the true and false targets is greater than a predetermined threshold.

Another form of all-neighbor association is a Bayesian approach called joint probabilistic data association (JPDA). JPDA is applicable to tracking multiple targets in scenarios with or without clutter. It takes into account scenarios where a measurement may fall inside the intersection of two or more validation gates of several different targets and so could have originated from any of these targets or from clutter.[40] A related technique, probabilistic data association (PDA), is used for tracking single targets.[41] In these methods, each candidate pairing updates the track estimator, which is weighted by a quantitative factor that describes its probability of being correct.[42] All neighboring measurements contribute to the track; hence deferred decision making is not required. A JPDA variation using update times that vary inversely with clutter level has been shown to improve tracking accuracy.[40] Bayesian techniques provide an optimal decision function when the *a priori* probabilities can be confidently established.

Track-to-track association

Track-to-track association is used to merge sensor-level tracks to obtain a central track file. Tracks can be characterized by position, velocity, covariance, and other features. In order to associate the sensor-level tracks, they first are transformed into a common coordinate system and time aligned, as was discussed under data alignment. Gates are then formed and a metric is chosen to evaluate the track association process. Many of the methods discussed for data association can be used to perform track-to-track association. These include nearest neighbor, global optimization, and deferred decision. The latter operates on tracks obtained over several future scans. After the track associations are made, the target position and covariance matrix corresponding to the input tracks are combined to form a new target position and covariance for the fused track. If the states observed by the various sensors are not identical, then only those that are common are used in the association process. The remaining states are augmented to the track and carried along. Subsequent track association can be simplified by storing associated sensor track numbers. As updated tracks arrive from the sensors, the previous track associations are then simply verified before the global track file is updated.

The variation and complexity of the tracking problem, as categorized by single target–single sensor, single target–multiple sensor, multiple target–single sensor, and multiple target–multiple sensor, dictate the data and track association technique as suggested by Table 3.6. The method of association shown is generally appropriate for the given tracking complexity. Of course, the more complicated association techniques can be used for the single target cases as well. Furthermore, in cases where the sensor cannot adequately resolve targets within the gate, groups of targets may be tracked rather than individual targets.

Positional, kinematic, and attribute estimation

These processes optimally combine multiple observations to obtain improved estimates of the position, velocity, and attributes (e.g., size, temperature, and shape) of an object. Estimates of updated target parameters are provided by a tracking filter. The filter uses algorithms that operate on time sequences of associated measurements to develop predictions of target state and attributes. Kinematic and adaptive models of object motion and sequential or batch processing (i.e., where all data are processed simultaneously) techniques are used to support the estimation process. The estimators also include *a priori* models of track dynamics and observations to refine the state estimate and to predict the state at the next observation interval for gating.

Even with *a priori* knowledge, the target may maneuver. Therefore, the state of the tracking filter must be changed to accommodate the maneuver. This can be accomplished in several ways. The first method, used with track splitting, augments the state of the parent track to include the maneuver. The second method, called the multiple-model maneuver, parameterizes the range of the expected maneuver and constructs tracking filters for each set of parameter values. Bloom and Bar-Shalom assume a transition probability for each of the sets of parametric values used to construct the filters.[43] States that are incorporated into filters must correspond to the observables of the tracking sensor. For example, if the state of a tracker is selected as position, velocity, and attitude (pitch, roll, and yaw), but only azimuth, elevation, and range are measured, then the attitude is not observable and the state cannot be updated.

Table 3.6 Suggested data and track association techniques for different levels of tracking complexity.

Tracking Complexity	Association Technique	Number of Scans
Single target–Single sensor	Nearest neighbor	Single
	Multiple hypothesis tracking	Multiple
	Track splitting	Multiple
Single target–Multiple sensor	Nearest neighbor	Single
	Multiple hypothesis tracking	Multiple
	Track splitting	Multiple
Multiple target–Single sensor	Nearest neighbor	Single
	JVC	Single
	Multiple hypothesis tracking	Multiple
	Track splitting	Multiple
	Maximum likelihood	Single or Multiple
	JPDA	Single
Multiple target–Multiple sensor	Nearest neighbor	Single
	JVC	Single
	Multiple hypothesis tracking	Multiple
	Track splitting	Multiple
	Maximum likelihood	Single or Multiple
	JPDA	Single

Track initiation

Several methods of track initiation are available to acquire targets and begin the tracking process. The simplest method uses single scan association to establish a detection gate based on minimum and maximum anticipated target speeds. When a detection not associated with another track is made, a gate is centered about the detection coordinates. Detections made on subsequent scans within the gate are then associated with the first detection. A track is initiated for every possible pairing of the first detection with subsequent ones. Usually detections on two consecutive scans are required to initialize the Kalman state and covariance track estimation filter for position and velocity. By limiting the association of detections to those on two consecutive scans, the gate size is minimized for the second detection and, thus, the creation of false tracks is minimized.

The promotion of the initiated tracks to system tracks is based on rules such as "*n* out of *m*." Here *n* detections out of *m* scans are required to declare the track a system track. Values of *n* and *m* are established from requirements that specify the number of false tracks, probability of target detection, clutter density, and the time allowed to declare a track. Another method of track initiation applies the maximum likelihood algorithm to several scans of stored data to maximize the probability of correctly associating the detections. In this case, processor capabilities usually limit the number of scans that are compared.[44]

3.3 Level 2, 3, and 4 processing

The results of Level 1 or low-level processing, i.e., target identities and tracks, assist in the execution of the situation assessment (Level 2) and threat assessment (Level 3) fusion processes. Refinement of the fusion process itself (Level 4) occurs through process evaluation and control that includes guidance for the acquisition of new data.

According to the Data Fusion Development Strategy Panel, Level 2 processing identifies the probable situation causing the observed data and events. Thus, it develops a description or interpretation of the current relationships among fixed and moving objects and events in the context of the operational environment. The data obtained from Level 1 analysis are now used to gain insights into prescribed event and activity sequences, force structures, and the overall battle environmental factors. Key functions of Level 2 processing, in terms of a military application, include:

- Object aggregation: establishing relationships among objects including temporal, geometrical proximity, communications links, and functional dependence;

- Event and activity aggregation: establishing temporal relationships among diverse entities to identify meaningful events or activities;

- Contextual interpretation and fusion: analyzing data with respect to the context of the evolving situation including weather, terrain, sea state or underwater conditions, enemy doctrine, and socio-political considerations.

Level 3 processing develops a threat-oriented data perspective to estimate enemy capabilities, identify threat opportunities, estimate enemy intent, and determine levels of danger. Threat assessment was originally a process distinct from situation assessment because threat assessment included multiperspective and quantitative enemy force analyses needed to estimate the enemy's course of action and force lethality. The newer definitions of Level 2 and Level 3 fusion define Level 2 fusion more broadly so that Level 3 is actually a subset of Level 2.[4] The critical functions that support threat assessment include:

- Capability estimation: predicting the size, location, and capabilities of enemy forces;
- Prediction of enemy intent: determining enemy intention based on actions, communications, doctrine, culture, history, education, and political structure;
- Identification of threats: identifying potential threat opportunities based on prediction of enemy actions, operational readiness analysis of friendly vulnerabilities, and analysis of environmental conditions;
- Multiperspective assessment: analyzing the data with respect to the friendly, enemy, and neutral perspectives, including effects of time and space on force deployment and preparing estimates of the enemy war plan;
- Offensive and defensive analysis: predicting the results of hypothesized enemy engagements considering rules of engagement, enemy doctrine, and weapon models.

Level 4 processing monitors and evaluates the ongoing fusion process to refine the process itself and to regulate the acquisition of data to achieve optimum results. Fusion process refinement interacts with each of the other levels and with external systems or the system operator. Its key functions include:

- Evaluations: assessing performance and effectiveness of the fusion process to establish real-time control and long-term process improvements;
- Fusion control: identifying changes or adjustments to processing functions within the data fusion domain that may result in improved performance;
- Source requirements processing: determining source-specific data requirements (specific sensors, sensor data, qualified data, reference data, etc.) needed to improve the multilevel fusion products;
- Mission management: recommending allocation and direction of resources (sensors, platforms, communications, etc.) to achieve overall mission goals.

Large databases, including the ability to support fast data insertion and fast data retrieval, are often needed to develop the higher-level fusion processes. The databases are maintained by management systems that provide monitoring, evaluation, addition, updating, retrieval, merging, and purging of data. Time tagging of entries assists in assuring that inferences drawn from these databases are relevant.

Figure 3.12 illustrates a command and control architecture as might be used in a military application to combine sensor data with information from a variety of diverse sources. The operational environment represented by the circle on the left side of the figure contains data entries that aid target identification and tracking, as well as situation and threat assessment found in Level 1, 2, and 3 fusion. The information that typically supports these fusion processes is detection and tracking data from land, air, sea, and space-based sensors including friendly missile guidance data from Global Positioning System satellites; lethality estimates; force and weapon composition; targeting ability; order of battle; and alert status for enemy and friendly forces. Weather sensors, diplomatic messages, analysis of political and economic factors, and other intelligence provide additional information.

The middle of the figure depicts Level 1 data fusion of real-time sensor data and historical database entries in support of target identification and tracking. Data from similar and dissimilar sources have been isolated to indicate that unique processing may be required for each type of information. Additional databases supply information to the Level 2 and 3 situation and threat assessment processes shown on the right. A database management system (DBMS) supports database housekeeping functions. The nodal interconnectivity boxes indicate that processing may occur both within a processing node and across processing nodes. Thus, fusion processes can begin at any level and do not have to progress from Level 1 through Level 4 in a prescribed order. Finally, the term "dynamic, integrated situation representation" represents the changeable nature of military environments and the dependence of the fusion results on the synthesis of information from diverse and multilevel sources.

3.4 Data fusion processor functions

Before discussing data fusion architectures, it is worthwhile to define the processes that usually occur in the data fusion processor. The fusion processor analyzes the inputs from all the sensors and performs the alignment, association, correlation, estimation, classification, and cueing functions defined below:[45]

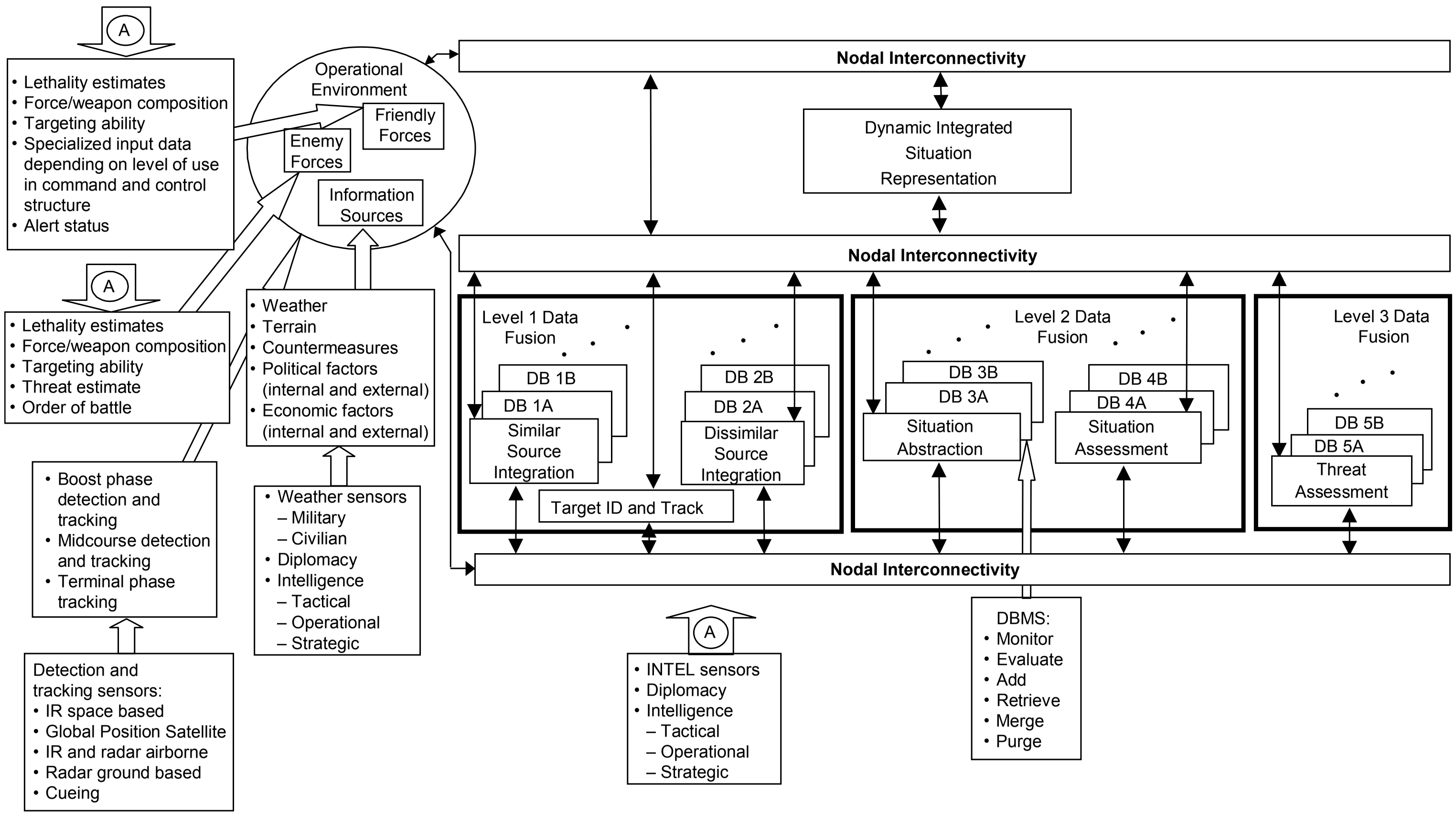


Figure 3.12 Military command and control system architecture showing fusion of information from multiple sources at multiple locations.

- Alignment: referencing of sensor data to a common time and spatial origin;
- Association: using a metric to compare tracks and measurement data (reports) from different sensors to determine candidates for the correlation process;
- Correlation: combining tracks and measurement data that are matched during association to improve detection, classification, and tracking of objects of interest;
- Estimation: predicting an object's future position by updating the state vector and error covariance matrix using the results of the correlation process;
- Classification: assessing the tracks and object discrimination data to determine target type, lethality, and threat priority;
- Cueing: feedback of threshold, integration time, and other signal processing parameters or information about areas over which to conduct a more detailed search, based on the results of the fusion process. For example, if a region of high clutter is found, a command may be sent to the appropriate sensor to increase the threshold setting. Alternatively, when the fusion processing identifies a decoy, a message describing the decoy's location is sent to minimize target search-related signal processing in this region. Another application of cueing is to initiate a search of a small, but high-interest region using a sensor of limited field of regard having high resolution, such as a laser radar.[46]

3.5 Definition of an architecture

An architecture is a system of components whose structure and integration enable it to perform functions that the individual components could not otherwise accomplish. An architecture initially provides conceptual design information to develop cost and operational effectiveness analyses, risk analyses, and technology transition. Design information includes specification of the components and their interconnections, data and information flows, system operating modes, and allocation of functions and subfunctions to particular architecture components and to alternates that assume the functions of failed components. The architecture identifies production, test, and support requirements and determines design constraints for configuration items (i.e., a system element or an aggregation of system elements that performs an end-use function and is designated for configuration control). As the architecture matures, it provides preliminary and detailed design information for system elements and their integration into products and processes.[47,48] As shown in the sections below, the definition of a data fusion architecture fits within the framework laid out in the broader architecture definition.

3.6 Data fusion architectures

There are several ways to classify data fusion architectures. In one approach, the architecture is defined by the extent of the data processing that occurs in each sensor, the data products produced by the individual sensors, and the location of the fusion processes. For example, sensors supplying information to detection, classification, and identification fusion algorithms may use complex processing techniques to provide the object class to a fusion algorithm for further refinement. Alternatively, the sensors may simply provide filtered signals or features to a fusion algorithm, where the signals or features are analyzed in conjunction with those from other sensors to determine the object class. On the other hand, sensors supplying information to state estimation and tracking algorithms may provide either measurement data, i.e., reports that contain the position and velocity of objects, or tracks of the objects. Current values of measurement data may be combined with previously obtained data to generate new tracks or the current data may be used to update pre-existing tracks using Kalman filtering. These processes can occur in the individual sensors or at a central processing node, depending on the architecture. If the sensors supply tracks, the tracks can be correlated with pre-existing tracks residing in individual sensors or at a central processing node.

The terms that describe data fusion architectures based on the extent of the data processing, data product types, and fusion location are sensor-level fusion (also referred to as autonomous fusion, distributed fusion, and post-individual sensor processing fusion), central-level fusion (also referred to as centralized fusion and pre-individual sensor processing fusion), and hybrid fusion, which uses combinations of the sensor-level and central-level approaches.[49–51] The resolution of the data and the extent of the processing by each sensor may also be employed to define another fusion architecture lexicon. The nomenclature used in this case is pixel-level, feature-level, and decision-level fusion.

3.6.1 Sensor-level fusion

With sensor-level fusion, each sensor detects, classifies, identifies, and estimates the tracks of potential targets before data entry into the fusion processor. The fusion processor combines the information from the sensors to improve the classification, identification, or state estimate of the target or object of interest.

The sensor-level fusion architecture, illustrated in Figure 3.13, is optimal for detecting and classifying objects if the sensors use independent signature-generation phenomena to develop information about the identity of objects in the field of regard, i.e., they derive object signatures from different physical processes and generally do not cause a false alarm on the same artifacts.[52] The sensor footprints must also be registered with respect to each other to ensure that the sensor signatures are characteristic of events or objects at the same spatial locations. Registration may be a simple task when the signatures arise from different information channels in the same sensor (e.g., reflectance and range data from a laser radar or multispectral data from a multispectral or hyperspectral infrared or visible wavelength sensor). Registration is more difficult when information from spatially separated sensors is combined.

Phenomena that generate the signatures detected by various types of sensors are listed in Table 3.7. Acoustic sensor signatures are included since they are frequently used in military and transportation applications.

The signatures are not only a function of the objects and background, but also of the sensor type and its design parameters as shown in Table 3.8. The signatures received by active sensors are influenced by the transmitted frequency and polarization, waveform shape, and power. Signatures from passive sensors are not a function of these parameters since no energy is transmitted by a passive sensor. Target shape, size, material, small-scale structure, orientation, and relative motion are other factors that affect the signatures detected by active sensors. Signatures of passive sensors that detect electromagnetic energy are affected by the emissivity, surface temperature, and roughness of the target; incidence angle; and receiver polarization. Passive acoustic and seismic sensors respond to sound and ground motion, respectively. Background and atmospheric effects caused by clutter, weather and other atmospheric obscurants, and countermeasures affect the signatures presented to active and passive sensors by absorbing and scattering energy associated with real targets and by creating false target signatures.

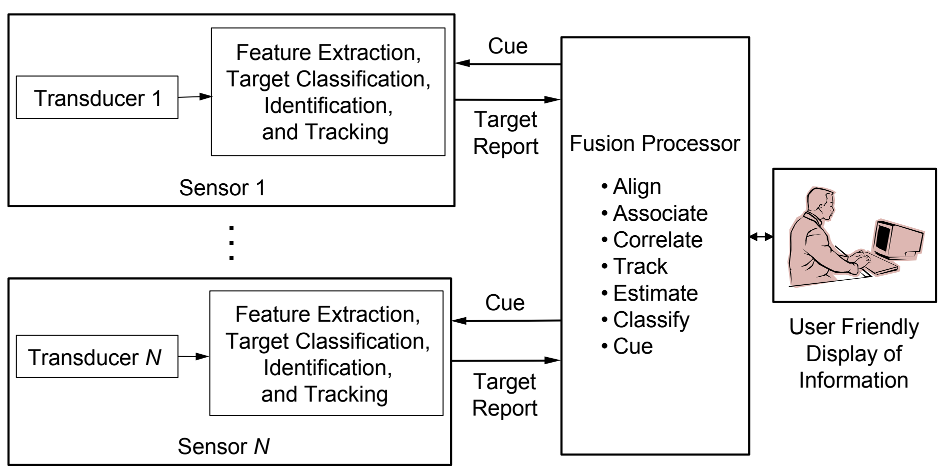


Figure 3.13 Sensor-level fusion.

Table 3.7 Signature-generation phenomena.

Sensor	Detectable Signature	Signature Source
MMW radar	Radar cross section, velocity	Shape, material composition, surface smoothness and regularity, gaps, cavities, receiver polarization, direction of movement with respect to sensor
MMW radiometer	Apparent temperature	Emissivity and temperature of object, receiver polarization and incidence angle, surface roughness, weather, atmospheric conditions
Laser radar	Radar cross section, reflectance, velocity	Shape, material composition, surface smoothness and regularity, gaps, cavities, direction of movement with respect to sensor
Infrared (FLIR or IRST)	Emission and reflectance	Radiance from within the object, such as from engines, and radiance from natural sources, such as direct heating by the sun or from reflected radiation
Visible	Reflection and direct illumination	Weather, atmospheric conditions, contrast with the background, visible emissions from exhausts
Electronic support measures (ESM)	Electronic emissions	Active sensor and transmitter sources such as communications equipment, navigation and guidance systems, fire control systems, electronic countermeasures, and, in general, any other source of electromagnetic radiation
Magnetic	Perturbation in Earth's magnetic field or change in an induced magnetic field	Magnetism associated with ferromagnetic materials (dipoles aligned parallel to their neighbors) and ferrimagnetic materials or ferrites (neighboring dipoles are aligned in an antiparallel arrangement, but different types of dipoles are present and the dipoles do not cancel)[53]
Acoustic	Acoustic energy	Engine noise, noise of object as it moves through air or moves on the ground surface such as produced by an airframe or ground vehicle
Seismic	Vibration or motion of surface	x, y, or z motion of ground surface induced by motion of vehicle upon it, by a hovering helicopter, or by movement of rocks or vegetation

Several types of signature-generation phenomena can be exploited in a multiple sensor system. A passive infrared sensor develops signatures from differences between the absolute temperatures and emissivities of the objects and background in the field of view. The emissivities are dependent on the surface characteristics of the particular object and the wavelength band in which the sensor operates. Laser radar can function as a multiple phenomena-sensing device in its own right. It receives a portion of the transmitted energy scattered from the objects and background that is proportional to their reflectance and scatterer shape and size. It also receives range data from which the distance to the scatterers can be calculated.

Table 3.8 Sensor, target, and background attributes that contribute to object signature characterization.

Sensor	Target	Background
Active or passive operation	Shape	Clutter distribution
Spatial resolution	Overall physical size	Clutter magnitude
Number and width of spectral bands	Small-scale structure	Clutter decorrelation time
Transmit and receive frequencies	Gross and small-scale signature parameters	False targets and sun glint
Frequency stability	Orientation	Jammers
Transmit and receive signal polarizations	Number and relative positions	Rain
Transmit waveform	Velocity and acceleration	Smoke
Transmit power		Dust
Scanning mechanism		Haze
Noise figure		Fog
Receiver sensitivity		Clouds
Receiver bandwidth		
Operating range		
Data registration		

Microwave and MMW radars receive a portion of the transmitted energy scattered from objects and background, which is proportional to the size and orientation of the surfaces that contribute to the scattering cross section of the object. Radars with larger fields of regard are capable of scanning the required search area faster than the infrared wavelength sensors, but with less resolution. However, the microwave and MMW radars operate in rain, fog, haze, clouds, and smoke with less absorption than infrared sensors.

Once the sensor system designer is assured that the sensor selection will provide signatures based on independent phenomena, the sensor outputs can be combined in a sensor-level-fusion architecture. The outputs from the sensors are fed into a fusion processor after each sensor has optimally processed its data. The signal processing can thus be tailored for each sensor according to its spatial, temporal, or frequency resolution; center frequency and bandwidth; field of regard; scan rate; and other attributes. Time-domain processing can be used for one sensor, frequency-domain techniques with another and multipixel image processing algorithms with a third.

In detection, classification, and identification fusion, two pieces of information must be present in each sensor's output to the fusion processor. These are (1) the detection, classification, or identification decision, and (2) how well, or with what confidence, the sensor has been able to detect, classify, or identify the objects in the field of regard. When tracking is of interest, a third piece of information is required, namely the location of the object or its track. With these inputs, it is possible to design a fusion algorithm that can combine the sensor data and improve upon the decision made by any sensor acting alone. In fact, sensor-level fusion can be shown to be as optimal (based on Bayesian decision logic) for detecting, classifying, and identifying targets as central-level fusion, which relies on minimally processed sensor data, when the sensors derive their information from independent signature-generation processes.[52] Three sensor-level fusion approaches, Bayesian inference, Dempster-Shafer evidential theory, and voting fusion based on Boolean algebra, are discussed in detail in later chapters.

3.6.2 Central-level fusion

Figure 3.14 depicts the central-level fusion architecture. In detection, classification, and identification data fusion, each sensor may provide minimally processed data to the fusion processor. Minimal processing includes operations such as filtering and baseline estimation. In state estimation and tracking fusion, the sensors typically supply measurement data, although sensor-generated tracks may also be sent to the fusion processor.

Central-level fusion algorithms are generally more complex and must process data at higher rates than in sensor-level fusion, because the centralized architecture is designed to operate on the minimally analyzed data output by each sensor. The central-level fusion algorithm examines input data for target features

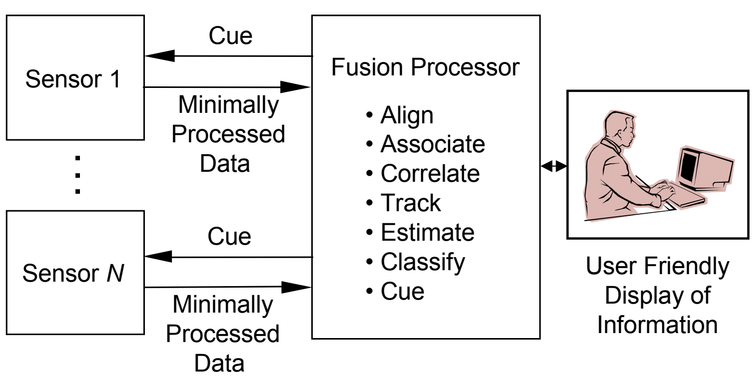


Figure 3.14 Central-level fusion.

or attributes that aid in tracking and discriminating among objects. Central-level fusion is optimal for tracking objects, as it is more effective than sensor-level fusion in estimating or predicting the future position of the object. Blackman observes that the increased tracking accuracy is due to a combination of effects: (1) processing all the data in one place, (2) forming the initial tracks based on observations from more than one sensor, thus eliminating tracks established from partial data received by the individual sensors, (3) processing sensor measurement data directly, eliminating difficulties associated with combining the sensor-level tracks produced by the individual sensors, and (4) facilitating multiple hypothesis tracking by having all data available in a central processor.[49] Deficiencies of the method are reflected in the large amount of data that must be transferred in a timely manner to the central processor(s) and then be processed by them. Central-level fusion target tracking and discrimination algorithms can be written to tolerate lack of particular sensor inputs. The advantages of sensor-level and central-level fusion are compared in Table 3.9. The hybrid fusion algorithm discussed next can be used to combine both target tracks and measurement data from multiple sensors.

Table 3.9 Comparative attributes of sensor-level and central-level fusion.

Sensor-Level Fusion	Central-Level Fusion
Discrimination among potential targets or objects of interest before data entry into the fusion processor reduces the load on the fusion processor	More accurate object discrimination than with sensor-level fusion, if the multisensor data are not generated by independent phenomena
Optimization of each sensor's signal processing to the nuances of the transducer design and kinematics	Optimization of object track and position estimates
Cueing to adjust sensor signal processing or search area parameters based on data from other sensors	Reduced weight, volume, power, and production cost in comparison with sensor-level fusion, if fewer processors are used
Flexibility in the numbers and types of sensors to allow addition, removal, or substitution of sensors without having to alter the fundamental structure of the fusion algorithm	Increased reliability of signal processing hardware, if fewer processors are used overall to support the fusion algorithms; reliability can be increased further, if required, by providing redundant paths for the processing
Cost-effective alternative for adding data fusion into an existing multi-sensor configuration	

3.6.3 Hybrid fusion

In hybrid fusion, shown in Figure 3.15, the central-level fusion process is supplemented by individual sensor signal processing algorithms that may, in turn, provide inputs to a sensor-level fusion algorithm. Hybrid fusion allows the tracking benefits of central-level fusion to be realized utilizing sensor measurement data and, in addition, allows sensor-level fusion of target tracks as computed by the individual sensors. Global track formation that combines the central- and sensor-level fusion tracks occurs in the central-level processor. Hybrid fusion can also be used to support target attribute classification when the signature data are not truly generated by independent phenomena. In this case, minimally processed data are sent to a central processor where they are combined using a fusion algorithm that detects and classifies objects in the field of view of the sensors. The disadvantages of hybrid fusion are the increased processing complexity and possibly increased data transmission rates.

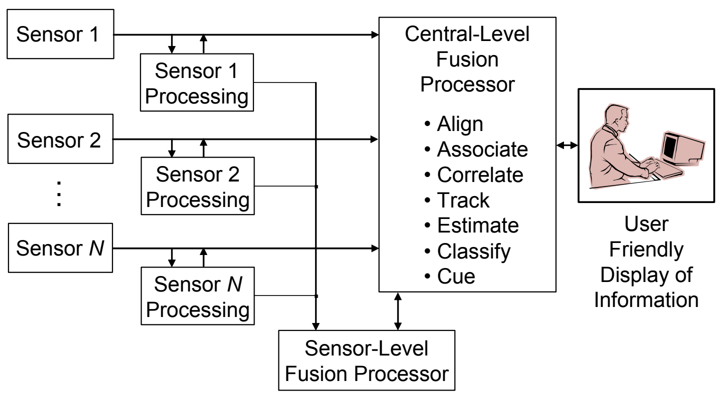


Figure 3.15 Hybrid fusion.

Hybrid fusion can manifest itself in the form of hierarchical architectures, where fusion nodes are arranged in a hierarchy with the lowest level nodes processing sensor data and sending results to higher-level nodes to be combined. One such architecture is shown in Figure 3.16.[32] Neyman-Pearson and Bayesian formulations of the distributed sensor detection problem for parallel, serial, and tree data fusion topologies are discussed by Viswanathan and Varshney.[29]

Fixed superior-subordinate relationships do not exist in a fully distributed architecture. Each node can communicate with other nodes subject to connectivity constraints. The communication can be adaptive and dependent on the information content and requirements of the individual nodes. Significant savings in communication resources are achieved when the higher-level nodes collect processing results periodically. The advantages of a distributed-fusion architecture and the issues raised through its use are summarized in Table 3.10.

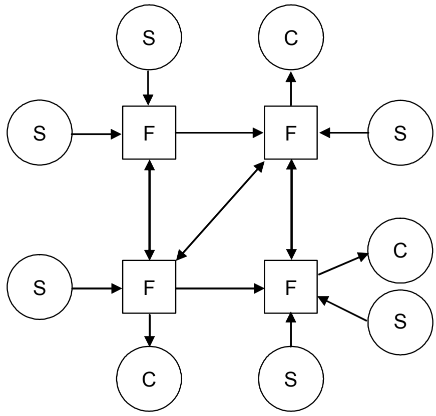


Figure 3.16 Distributed fusion architecture.

Table 3.10 Advantages and issues associated with a distributed fusion architecture.

Advantages	Issues
Lighter processing load at each fusion node because of the distribution of the load over multiple nodes	Architecture: Sharing of fusion responsibility among nodes, e.g., identification of sensors or sources reporting to each node and targets for which each node is responsible
No requirement to maintain a large centralized database since each node has its own database	Communications: Connectivity and bandwidth of the nodal communication network, identification of information sources and sinks, and establishing need for raw data or processing results for each node
Reduced communication load because data are not sent to and from a central processing site	Algorithms: Methods used by nodes to efficiently and effectively fuse data and to select appropriate communication actions (i.e., who, when, what, and how)
Faster user access to fusion results since communication delay is reduced	
Increased survivability due to elimination of single point failure mode (as is associated with a centralized fusion architecture)	

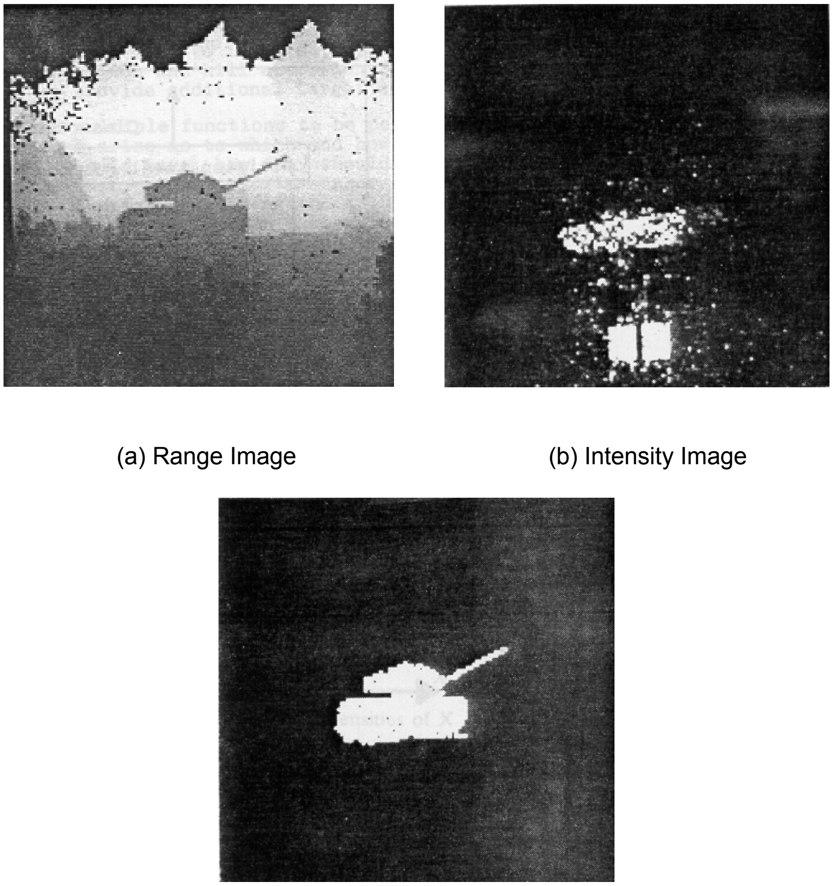
3.6.4 Pixel-level fusion

In pixel-level fusion, minimally processed data from different sensors, or sensor channels within a common sensor, are combined at the pixel or resolution-cell level of the sensors using a central-level-fusion architecture. Little, if any, preprocessing of the data occurs.

Pixel-level fusion is applied to LANDSAT imagery to detect diseased crops or identify a particular crop. Identification is not made using the individual spectral bands of data, but rather the information from all bands is combined in a pixel-level fusion process before the scene is identified.

Figure 3.17 illustrates an example of pixel-level fusion using CO_2 laser radar data. Range histograms derived from target and clutter background imagery [such as in Figure 3.17(a)] are combined with histograms representative of intensity images [shown in Figure 3.17(b)] that correspond to the reflectance of the target and clutter objects. Range histograms may show large numbers of returns from many range cells, making it difficult to isolate the range that corresponds to the target. However, histograms based on intensity images show stronger returns for metallic surfaces than for foliage. Therefore, fusing the range and intensity histogram data to identify the pixels that correspond to targets should assist in segmenting the targets from the background. Accordingly, pixels in the original range image that are not within a range gate near the peak intensity are set to zero, as are pixels in range bins that do not contain more than some predetermined number of pixels.

This technique removes clutter and noise pixels, but also eliminates smaller target features such as gun barrels. These can be restored by exploiting *a priori* knowledge about the expected size of the target at the operating range of the sensor.[52,54] The final fused image in shown in Figure 3.17(c). It is possible to encounter image or data registration problems when fusing data from different sensors. In the laser radar example, however, the pixels in the range and intensity images are perfectly aligned because the same sensor produces them.



(a) Range Image (b) Intensity Image

(c) Fused Image

Figure 3.17 Pixel-level fusion in a laser radar. (A.O. Aboutalib and T.K. Luu, "An efficient target extraction technique for laser radar imagery, *Digital Signal Processing, Association*, *and Tracking of Point Source, Small, and Cluster Targets*, *Proc. SPIE* 1096 [1989].)

3.6.5 Feature-level fusion

Feature-level fusion is applicable to either a central-level- or sensor-level-fusion architecture. Features are extracted from each sensor or sensor channel and combined into a composite feature, representative of the object in the field of view of the sensors. An example of a composite feature is one constructed by stringing individual sensor feature vectors end to end (concatenation) to form a longer vector that serves as the input to a classifier. Another example of feature-level fusion occurs with multilayer artificial neural networks as depicted[55] in Figure 3.18. As shown here, target features are extracted from a millimeter-wave radar, passive infrared sensor, and laser radar. The features are combined to form a composite vector that is input to a neural network. The network, programmed off-line to recognize the targets of interest and differentiate them from false targets or background clutter, assigns observed objects to particular classes with some probability, confidence, or priority. Training is performed using corresponding data from all the sensors. Therefore, if a different sensor type replaces one of the original sensors, sensor data collection and training have to be repeated.

3.6.6 Decision-level fusion

Decision-level fusion is associated with sensor-level fusion shown in Figure 3.13. The results of the initial object detection and classification by the individual sensors are input to a fusion algorithm. Final classification occurs in the fusion processor using an algorithm that combines the detection, classification, and position attributes of the objects located by each sensor. Classification performance is suboptimal compared to that of feature-level fusion unless the sensors respond to independent signature-generation phenomena.[52]

3.7 Sensor footprint registration and size considerations

When sensors are located at different spatial positions or, for that matter, collocated on the same platform, it is desirable to have their footprints overlap in target-detection space. Furthermore, the measurement data or imagery from each sensor must be time and spatially aligned, or registered, with respect to those from the other sensors. Overlapping sensor footprints ensure that time-dependent phenomena (such as clutter decorrelation or target motion) are observed by all sensors at the same time. This footprint configuration supports optimal fusion of the sensor data within the overlapping fields of view. If data from overlapping

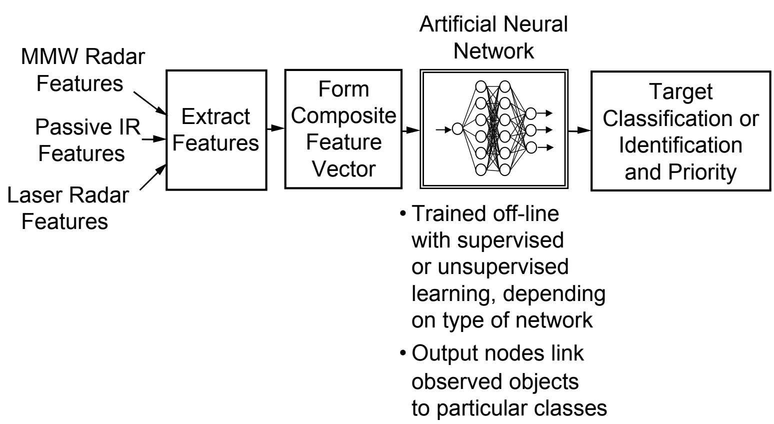


Figure 3.18 Feature-level fusion in an artificial neural network classifier.

sensors are needed by the particular fusion algorithm, the maximum operating range must be limited to that at which all the sensors function.

Usually the selected sensors have different sized footprints. The issue then is over which footprint to compare the multisensor target reports. The obvious choice is to pick the largest footprint. That way data are compared over an area corresponding to the limiting or least resolution sensor (assuming the footprint represents one pixel). The finer resolution sensors, such as a passive infrared sensor or laser radar, must then acquire and process imagery over the larger footprint before sending the results onto the fusion processor as, for example, when sensor-level fusion is used.

When sensors are not collocated, algorithms and their associated parameters are defined to align the multiple sensor data in time and space. These spatial alignment algorithms take into account the coordinate systems that measure the location of the objects and the errors introduced by transforming the measurements into other coordinates. Uncertainties in object location reflected in position or velocity error volumes are typically included in the coordinate transformations. Gates are established to control data association from different sensors and from temporal and spatial measurements. The gate size is selected to obtain a balance between maximizing detection probability (use of large-sized gates) and minimizing misassociation probability (use of small-sized gates).

Several approaches for registering MMW and IR data have been explored.[56–58] Infrared sensors that produce two-dimensional imagery typically provide high-resolution in the elevation and azimuth planes, while two-dimensional MMW sensors provide data in range and azimuth. Scene registration is made easier if, in the design process, the sensors' fields of view are made as equal as design, operating, and cost constraints permit. Scene registration is also affected by operational constraints, such as special topology or potential false targets, and test conditions where sensor mounting, boresighting, and data analysis issues are of concern. In registering MMW and IR sensor data in pixel-level fusion applications, for example, flat versus rolling terrain topology must be accounted for as part of the data analysis task in order to obtain valid results from the data fusion process.

Generation of a site model is another technique used to align multisensor data. A three-dimensional frame of reference is established into which all available relevant structural and contextual information is incorporated. Site models allow the use of prior information about the structure of objects and their immediate environments. This frequently leads to simpler and more robust algorithms.[59]

3.8 Summary

Data fusion consists of low-level and high-level processes. The low-level processes include target detection, classification, identification, and tracking. High-level processes encompass situation and threat assessment. The algorithms generally used to support target detection, classification, and identification are based on physical models, feature-based inference, and cognition. Numerous examples of these techniques were introduced, including Kalman filtering, classical inference, Bayesian inference, Dempster-Shafer evidential theory, generalized evidence processing, artificial neural networks, clustering, voting logic, pattern recognition, knowledge-based expert systems, and fuzzy set theory.

Different algorithms are used for track estimation. The track estimation algorithms are concerned with data alignment, data and object correlation, and position, kinematic, and attribute estimation. Data alignment establishes a common space-time reference for fusion processing. Correlation is performed through prediction gates; kinematic, attribute, and time association metrics; and data and track association techniques. Prediction gates associate data into candidates that are suitable for updating tracks or forming tentative new tracks. Multiple sets of measurement data can arise from overlapping gates, multiple returns in a gate, clutter, new targets in a gate, and returns received over multiple scans. Metrics quantify the similarity of the observations. In the context of a multiple-target and multiple-sensor environment, data association applies the metric to compare tracks and measurement data reports from different sensors to determine candidates for the correlation process. Correlation determines if the tracks and measurement data resulting from association belong to a common object. Track association merges tracks from different sensors to form a central track file. Position, kinematic, and attribute estimation combine information from multiple observations to improve knowledge of the target's position, velocity, and identification.

Evaluation of tracking performance is not limited to assessment of state estimation and prediction errors. Other measures required to characterize the performance of a target tracking system include the number of missed and false tracks, probability of misassociation, and accuracy of the state covariance matrix. A desirable feature of tracking algorithms is to be able to predict their performance as a function of target density, probability of missed and false signals, number of new targets, and other error sources.

Data fusion that supports situation assessment interprets current relationships among objects and events in the context of an operational environment. Important functions included in situation assessment are object, event, and activity aggregation and contextual interpretation and fusion. Fusion in support of threat assessment is designed to estimate enemy capabilities, threat opportunities, enemy intent, and levels of danger. Included in threat assessment are estimation of enemy capability and intent, identification of threats, multiperspective assessment, and analysis of friendly and enemy capabilities.

Data fusion architectures are described in several ways. The first taxonomy is based on the amount of data processing performed by the sensors, data products produced by the sensors, and the location of the fusion processes. In this case, the architectures are referred to as sensor-level fusion (or autonomous fusion, distributed fusion, and post-individual sensor processing fusion), central-level fusion (or centralized fusion and pre-individual sensor processing fusion), and hybrid fusion (using combinations of the sensor-level and central-level architectures). The second fusion lexicon uses the resolution of the data and the extent of the processing performed by a sensor before the data are fused. The nomenclature used in this instance is pixel-level, feature-level, and decision-level fusion. Sensor-level fusion allows signal processing to be optimized for the individual sensors in the architecture, while central-level fusion can be designed to optimally process all the data arriving from the entire suite of sensors. Other considerations arise in selecting an appropriate architecture, such as data processing and communication resources and the application of the fusion products.

References

1. F.E. White, Jr., "Joint directors of laboratories data fusion subpanel report: SIGINT session," *Tech. Proc. Joint Service Data Fusion Symposium*, Vol. I, DFS-90, 469-484 (1990).
2. Data Fusion Development Strategy Panel, *Functional Description of the Data Fusion Process*, Office of Naval Technology (Nov. 1991).
3. E. Waltz and J. Llinas, *Multisensor Data Fusion*, Artech House, Norwood, MA (1990).
4. A.N. Steinberg, C.L. Bowman, and F.E. White, "Revisions to the JDL Data Fusion Model," *Proc. SPIE* 3719, 430-441 (Apr. 1999).
5. E.P. Blasch and S. Plano, "JDL Level 5 Fusion Model 'User Refinement' Issues and Application in Group Tracking," *Proc. SPIE* 4729, *Aerosense*, 270-279 (2002).
6. "1989 data fusion survey," 1990 Data Fusion Symposium, Johns Hopkins University Applied Physics Laboratory, Laurel, MD (May 1990).
7. J. Johnson, Paper AD-220160, Image Intensifier Symposium, Fort Belvoir, VA, 49 (Oct. 1958).
8. F.A. Rosell and R.H. Willson, "Recent psychophysical experiments and the display signal-to-noise ratio concept," Chap. 5 in *Perception of Displayed Information*, Editor: L.M. Biberman, Plenum Press, New York, NY (1973).
9. G.A. Gordon, R.L. Hartman, and P.W. Kruse, "Imaging-mode operation of active NMMW systems," Chapter 7 in *Infrared and Millimeter Waves, Vol. 4: Millimeter Systems*, Editor: K.J. Button and J.C. Wiltse, Academic Press, New York, NY (1981).
10. L.A. Klein, *Millimeter-Wave and Infrared Multisensor Design and Signal Processing*, Artech House, Norwood, MA (1997).
11. L. Valet, G. Mauris, and Ph. Bolon, "A statistical overview of recent literature in information fusion," *IEEE AESS Sys. Mag.*, 7-14 (Mar. 2001).
12. D.L. Hall and R.J. Linn, "Algorithm selection for data fusion systems," 1987 *Tri-Service Data Fusion Symposium Technical Proceedings*, Vol. I, DFS-87, Naval Air Development Center, Warminster, PA (June 1987).
13. D.L. Hall and R.J. Linn, "A taxonomy of algorithms for multi-sensor data fusion," *Tech. Proc. Joint Service Data Fusion Symposium*, Vol. I, DFS-90, 594-610, Naval Air Development Center, Warminster, PA (1990).
14. D.L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Norwood, MA (1992).
15. I.R. Goodman, R.P.S. Mahler, and H.T. Nguyen, *Mathematics of Data Fusion*, Kluwer Academic Publishers, Norwell, MA (1997).

16. J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, San Mateo, CA (1988).
17. J. Dezert, "Foundations for a new theory of plausible and paradoxical reasoning," *Information and Security*, Vol. 9, 1-45 (2002).
18. C.K. Murphy, "Combining belief functions when evidence conflicts," *Decision Support Systems*, Elsevier Science, Vol. 29, 1-9 (2000).
19. P. Smets and R. Kennes, "The transferable belief model," *Artif. Intell.*, Vol. 66, 191-234 (1994).
20. B.R. Cobb and P.P. Shenoy, "A comparison of methods for transforming belief function models to probability models," in T.D. Nielsen and N.L. Zhang (eds.), *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, 255-266 (2003). Also www.business.ku.edu/home/pshenoy/ECSQARU03.pdf, accessed July 2003.
21. A. Jøsang, "The consensus operator for combining beliefs," *AI Journal*, Vol. 14(1-2), 157-170 (2002).
22. R. Haenni and N. Lehmann, "Probabilistic augmentation systems: a new perspective on Dempster-Shafer theory," *Int. J. of Intell. Sys.*, 18(1), 93-106 (2003).
23. D. Fixsen and R.P.S. Mahler, "The modified Dempster-Shafer approach to classification," *IEEE Trans. Sys., Man, and Cybern.–Part A: Systems and Humans*, SMC-27(1), 96-104 (Jan. 1997).
24. A.-S. Capelle, C. Fernandez-Maloigne, and O. Colot, "Introduction of spatial information within the context of evidence theory," *IEEE Int. Conf. on Acoustics, Speech, and Sig. Proc.* (*ICASSP*), 785-788 (2003).
25. S.C.A. Thomopoulos, R. Viswanathan, and D.C. Bougoulias, "Optimal decision fusion in multiple sensor systems," *IEEE Trans. Aerospace and Elect. Sys.*, AES-23(5), 644-653 (Sep. 1987).
26. S.C.A. Thomopoulos, "Theories in distributed decision fusion: comparison and generalization," *Sensor Fusion III: 3-D Perception and Recognition*, *Proc. SPIE* 1383, 623-634 (1990).
27. S.C.A. Thomopoulos, "Sensor integration and data fusion," *J. of Robotic Systems*, 7(3), 337-372 (June 1990).
28. P. Rohan, *Surveillance Radar Performance Prediction*, Peter Peregrinus, Ltd., London, UK (1983).
29. R. Viswanathan and P.K. Varshney, "Distributed detection with multiple sensors: Part I – Fundamentals," *Proc. IEEE*, 85(1), 54-63 (Jan. 1997).
30. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Ed., Prentice Hall PTR, Upper Saddle River, NJ (1998).
31. R. Schalkoff, *Pattern Recognition: Statistical, Structural, and Neural Approaches*, John Wiley, New York, NY (1992).
32. M.E. Liggins II, C.-Y. Chong, I. Kadar, M.G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proc. IEEE*, 85(1), 95-107 (Jan. 1997).

33. O.E. Drummond and S.S. Blackman, “Challenges of developing algorithms for multiple sensor, multiple target tracking,” *Proc. SPIE* 1096, 244-255 (1989).
34. S.S. Blackman, “Theoretical approaches to data association and fusion,” *Sensor Fusion*, *Proc. SPIE* 931, 50-55 (1988).
35. N.R. Pal and S. Ghosh, “Some classification algorithms integrating Dempster-Shafer theory of evidence with the rank nearest neighbor rules,” *IEEE Trans. Sys., Man, and Cybern.–Part A: Systems and Humans*, SMC-31(1), 59-66 (Jan. 2001).
36. J. Francois, Y, Grandvalet, T. Denoeux, and J.-M. Roger, “Resample and combine: An approach to improving uncertainty representation in evidential pattern classification,” *Information Fusion*, (4), 75-85 (2003).
37. J. Munkres, “Algorithms for the assignment and transportation problem,” *J. SIAM*, 5, 32-38 (Mar. 1957).
38. O.E. Drummond, D.A. Castanon, and M.S. Bellovin, “Comparison of 2-D assignment algorithms for sparse, rectangular, floating point, and cost matrices,” *J. SDI Panels Tracking,* Institute for Defense Analyses, Alexandria, VA, Issue No. 4, 4-81 to 4-97 (Dec. 15, 1990).
39. I.P. Bottlik and S.S. Blackman, “Coordinated presentation of multiple hypotheses in multitarget tracking,” *Signal and Data Processing of Small Targets*, *Proc. SPIE* 1096, 152-159 (Apr. 1989).
40. S.S. Ahmeda, M. Keche, I. Harrison, and M.S. Woolfson, “Adaptive joint probabilistic data association algorithm for tracking multiple targets in cluttered environment,” *IEE Proc.-Radar, Sonar Navig.*, 144(6), 309-314 (Dec. 1997).
41. Y. Bar-Shalom and E. Tse, “Tracking in clutter environments with probabilistic data association,” *Automatica*, 11, 451-460 (Sep. 1975).
42. Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, New York, NY (1988).
43. H.A.P. Bloom and Y. Bar-Shalom, “The interacting multiple-model algorithm for systems with switching coefficients,” *IEEE Trans. Automat. Control*, AC-33, 780-783 (Aug. 1988).
44. C.L. Morefield, “Application of 0-1 integer programming to multitarget tracking problem,” *IEEE Trans. Automat. Control*, AC-22, 302-312 (June 1977).
45. Data Fusion Subpanel, *Data Fusion Lexicon*, Joint Directors of Laboratories Technical Panel for C^3 (Oct. 1991).
46. R.C. Luo and M.G. Kay, “Multisensor integration and fusion: issues and approaches,” *Sensor Fusion*, *Proc. SPIE* 931, 42-49 (1988).
47. E. Rechtin, *Systems Architecting: Creating and Building Complex Systems*, Prentice Hall, Englewood Cliffs, NJ (1991).
48. Draft Military Standard, *Systems Engineering*, MIL-STD-499B (May 15, 1991).

49. S.S. Blackman, "Multiple sensor tracking and data fusion," Chapter 7 in S.A. Hovanessian, *Introduction to Sensor Systems*, Artech House, Norwood, MA (1988).
50. V.G. Comparato, "Fusion–the key to tactical mission success," *Sensor Fusion, Proc. SPIE* 931, 2-7 (1988).
51. S.S. Blackman, "Association and fusion of multiple sensor data," Chapter 6 in *Multitarget-Multisensor Tracking: Advanced Applications*, Editor: Y. Bar-Shalom, Artech House, Norwood, MA (1990).
52. G.S. Robinson and A.O. Aboutalib, "Trade-off analysis of multisensor fusion levels," *Proc. of the 2nd National Symp. on Sensors and Sensor Fusion*, Vol. II, GACIAC PR 89-01, 21-34, IIT Research Institute, Chicago, IL, (1990).
53. S. Ramo, J.R. Whinnery, and T. Van Duzer, *Fields and Waves in Communication Electronics*, John Wiley, New York, NY (1967).
54. A.O. Aboutalib and T.K. Luu, "An efficient target extraction technique for laser radar imagery," *Digital Signal Processing, Association*, *and Tracking of Point Source, Small, and Cluster Targets*, *Proc. SPIE* 1096 (1989).
55. P.F. Castalez and J.L. Dana, "Neural networks in data fusion applications," *Proc. of the 2nd National Symp. on Sensors and Sensor Fusion,* Vol. II, GACIAC PR 89-01, 105-114, IIT Research Institute, Chicago, IL (1990).
56. K.S. Strom, J.F. Carter, and J.H. Chockley, "Joint Army/Air Force millimeter wave/infrared seeker development," *Proc. of the 4th National Symp. on Sensor Fusion,* Vol. I, 213400-93-X(I), 63-81, ERIM, Ann Arbor, MI (1991).
57. J.E. Overland et al., "Identifying the density of multi-sensor false alarms," *Proc. of the 4th National Symp. on Sensor Fusion,* Vol. I, 213400-93-X(I), 313-322, ERIM , Ann Arbor, MI (1991).
58. E.D. D'Anna and M.A. Richards, "A radar angular resolution improvement technique for multispectral sensor beam registration," *Proc. of the 4th National Symp. on Sensor Fusion,* Vol. I, 213400-93-X(I), 417-429, ERIM, Ann Arbor, MI (1991).
59. R. Chellappa, Q. Zheng, P. Burlina, C. Shekhar, and K.B. Eom, "On the positioning of multisensor imagery for exploitation and target recognition," *Proc. IEEE*, 85(1), 120-138 (Jan. 1997).