AIIM

Association for Information and Image Management

1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

301/587-8202

Centimeter

Inches

Adaptive Sensor Fusion Using Genetic Algorithms

Deborah S. Fitzgerald

Douglas G. Adams

Sandia National Laboratories
Albuquerque, NM 87185-0780

ABSTRACT

Past attempts at sensor fusion have used some form of Boolean logic to combine the sensor information. As an alternative, an adaptive "fuzzy" sensor fusion technique is described in this paper. This technique exploits the robust capabilities of fuzzy logic in the decision process as well as the optimization features of the genetic algorithm. This paper presents a brief background on fuzzy logic and genetic algorithms and how they are used in an on-line implementation of adaptive sensor fusion.

INTRODUCTION

Currently, most sensor inputs are combined and interpreted manually by the operator at the system console. This task is becoming more complex, however, as site security systems grow. The purpose of the fuzzy fusion technique is to intelligently and adaptively combine sensor information to relieve the operator of this requirement. In turn, this will support the next generation of access control and alarm display systems.

This paper presents a new approach to sensor fusion, which exploits the robust capabilities of fuzzy logic and the optimization features of genetic algorithms. A system application is described first, followed by a brief introduction to fuzzy logic and genetic algorithms. Further details about an adaptive, on-line application of the sensor fusion approach is discussed, and finally, the status of the development is described.

SYSTEM APPLICATION: ADVANCED EXTERIOR SENSOR

Sandia is currently developing an Advanced Exterior Sensor (AES). The AES is an intrusion detection and assessment system designed for wide-area coverage, quick deployment, low false/nuisance alarm operation, and immediate visual assessment. The AES combines three sensor technologies: visible, infrared (IR), and millimeterwave radar collocated on a compact, portable remote sensor module (RSM). The RSM rotates at a rate of 1 revolution per second to detect motion and provide assessment in a continuous 360° field-of-regard. Detection continues during assessment of multiple alarms.

The information from the three sensors is combined to increase the reliability and confidence of a real target (or intrusion) while filtering out targets which are false and reducing nuisance alarms. The genetic adapter, shown in figure 1, is used to adaptively optimize the functions used to correlate the information, thus resulting in a robust sensor. This operation will be described in further detail in the last section of this paper. It is useful first to provide some background information on fuzzy logic and genetic algorithms.

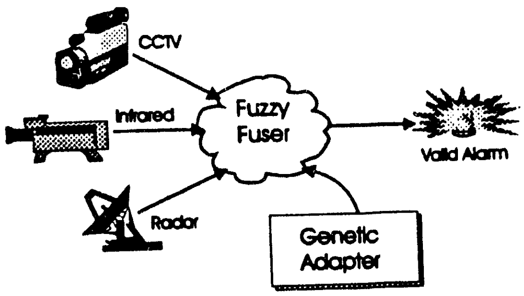

Figure 1 Advanced Exterior Sensor

BACKGROUND: FUZZY LOGIC

In order to explain fuzzy logic, it is helpful to review classical set theory. In classical set theory, an item either is a member of a set or it is not. For example, in the universal set of US cities, Dallas and Houston are members of the set "Cities in Texas", whereas Kansas City, San Francisco, and Naples, Florida, are not (Figure 2). The problem with classical set theory is that it is not capable of describing many real-world problems. For example, what if you wanted to describe the set "Cities NEAR Texas"?

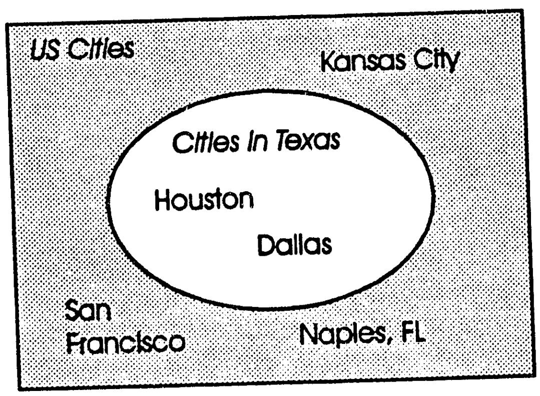

Figure 2 Classical Set Theory

Fuzzy set theory allows this extension, by defining members of a set with a "degree of membership." Continuing with the example, in the set "Cities NEAR Texas", Dallas and Houston have a degree of membership of 1.0 since they are in the state of Texas. Kansas City, the next closest city, has a degree of membership of 0.8. Naples, Florida, a bit farther away, has a degree of membership of 0.5, and San Francisco's degree of membership is 0.3 (Figure 3).

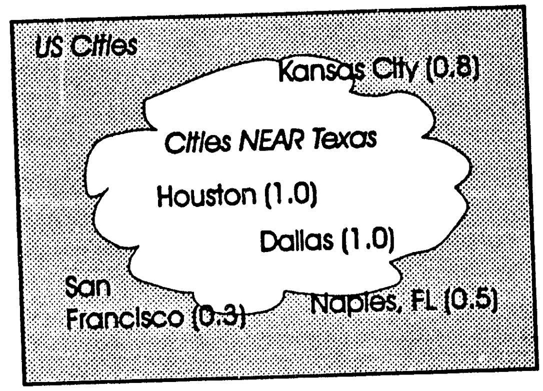

Figure 3 Fuzzy Set Theory

An even better way to graphically describe a fuzzy set is with a fuzzy logic membership function. The function shown in figure 4 is an example of a typical membership function, however, membership functions can be any shape and size.

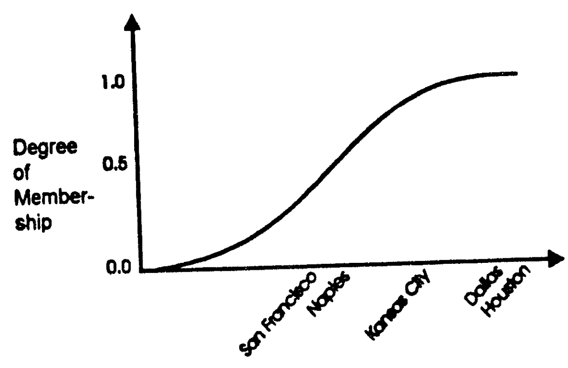

Figure 4 Fuzzy Logic Membership Function

A fuzzy logic inference engine is similar to a rule-based expert system. Figure 5 shows an example inference engine with three rules which describe the performance of a thermal imager. The advantage of an inference engine is it can be used to produce desired results without having to develop a deep mathematical or analytical model. Advantages to a fuzzy logic inference engine include:

- many rules can operate in parallel
- the rules are not absolute: the extent of the antecedent ("If temperature is hot") infers the extent of the consequence ("then IR confidence is bad")
- the rules are easy to express (can use natural language)
- there are generally fewer rules in a fuzzy logic inference engine than in a classical expert system

Rule 1: If temperature is hot, then IR confidence is bad.
Rule 2: If temperature is warm, then IR confidence is fair.
Rule 3: If temperature is cold, then IR confidence is good.

IR Confidence = A combination of Rule 1, Rule 2, and Rule 3.

Figure 5 Fuzzy Logic Inference Engine

Figure 6 illustrates a set of input membership functions that might be used in the example inference engine. The key to developing a successful inference engine is in the definition of the membership functions. For example, who makes the determination as to what is hot and what is not? In figure 6, completely hot begins at 110°F. With just a tweaking of these numbers, the entire inference engine is modified. Figure 7 shows a new set of membership functions by simply modifying the height, width, and centers of the triangle membership functions in figure 6. Now, completely hot begins around 85°F, which is quite a difference. The selection of these defining parameters is the key to optimizing the system. And genetic algorithms are the key to optimizing the parameters.

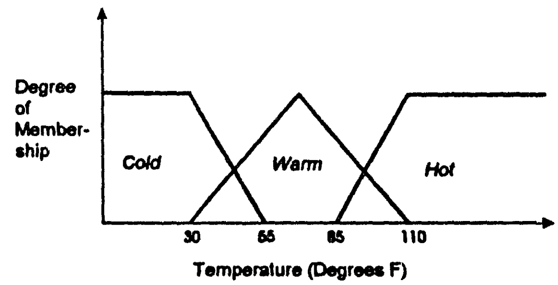

Figure 6 Example Membership Functions

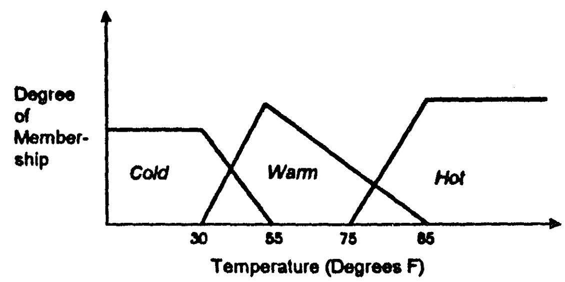

Figure 7 Modified Membership Functions

BACKGROUND: GENETIC ALGORITHMS

A genetic algorithm is a search procedure based on the mechanics of natural selection and natural genetics. Genetic algorithms have several advantages over other search and optimization methods. They are very robust in multimodal problems where there is more than one peak in the search space or surface. This is mainly because genetic algorithms begin their search from a population of points randomly seeded over the entire search space as opposed to beginning at a single point. More importantly, unlike many of tl classical optimization schemes, genetic algorithms do not depend on auxiliary knowledge such as derivatives.

The concept behind genetic algorithms is actually very simple–you start with a population of creatures, each with the same number of genes and chromosomes, usually represented by bit strings.

The genetic algorithm has four basic steps:

Step 1: Each creature in the population is randomly seeded.

Step 2: The fitness of each creature is evaluated. (This fitness is a measure of the creature's performance on the problem to be solved. It plays the same role in genetic algorithms that the environment plays in a natural evolution. Each creature is evaluated and a number or ranking is assigned to the creature as a measure of the creature's fitness or goodness relative to the function to be optimized.)

Step 3: The most fit creatures are selected to mate and produce offspring.

Step 4: The offspring are used to build the next generation population. The process returns to step 2.

The iterations continue until most of the creatures are very similar, at which time reproduction no longer provides improvement. In real applications, the iterations are usually truncated to a certain number in the interest of compute time, and the most fit creature in the last generation represents the optimal solution to the problem to be solved.

When two creatures mate, the reproduction operations are very similar to natural genetics. Figure 8 shows an example of simple crossover and mutation methods. In one-point crossover, a crossover point is randomly selected. All the numbers (or bits in this case) from that point to the end of the string are swapped between the two parents to produce two children. Genetic information from both parents is thus included in the two children.

Mutation is generally applied with a very low probability; some applications have been successful without any mutation at all, others with a mutation probability of 0.008. In the event that a creature is chosen to mutate, as shown in the example in figure 8, a mutation point is randomly selected and the bit corresponding to that random point is flipped.

```
Crossover:
        Parent A:   1 0 1 1 0 | 0 1 1
        Parent B:   0 1 1 0 1 | 0 1 0

  Random Crossover Point  --> |

        Child a:    1 0 1 1 0 | 0 1 0
        Child b:    0 1 1 0 1 | 0 1 1

Mutation:
        Child a:    1 0 1 1 1 0 1 0

  Random Mutation Point
```

Figure 8 Example Crossover and Mutation

As in nature, there are many different methods for mate selection. There are also many other methods for reproduction and crossover in addition those previously described. Recent practical applications of genetic algorithms have shown that these methods, and various combinations of the methods, are application specific–some methods work better than others for particular problems. Additionally, one of the most important aspects of the genetic algorithm is the definition of fitness. If the fitness function is not well-defined, the algorithm may not be successful.

ON-LINE GENETIC APPLICATION

As previously described, the AES combines the information of three sensor technologies. This information is combined in a target correlator, which uses fuzzy logic inference engines to resolve the confidence for a target object. For example, if a target were identified by only one of the three sensors, then the confidence would not be as high as if it were identified by all three sensors. But if the weather conditions were detrimental to the performance of the other two sensors, the confidence may be increased. As shown in figure 9, a genetic algorithm is used to optimize the target correlator. If the target confidence exceeds an adaptive threshold in the target correlator, it is passed on to the operator as an alarm. The operator, who is obligated to assess the alarm, provides feedback to the system through this assessment. This feedback plays an important part in the fitness function of the genetic algorithm.

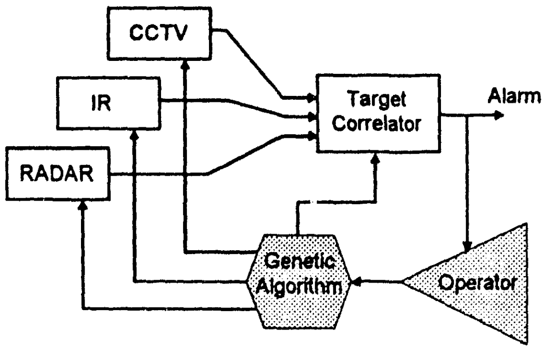

Figure 9 On-line Genetic Application

Also shown in figure 9 are links from the genetic algorithm to each sensor processor. Figure 10 shows this relationship in more detail for one particular sensor. Four separate measures are determined for each potential target:

- physical match–comparing shape and size to keep track of a potential target from frame to frame
- purposeful motion–determining if the target is exhibiting predictable motion
- saliency–evaluates such features as size and brightness of a potential target on a single frame basis
- reliability–based on features such as the number of times the target was seen over a sequence of frames

These four measures are defined with fuzzy logic inference engines, one for each measure. They are all combined into a fifth inference engine which resolves the confidence for a potential target. Again, if this confidence exceeds an adaptive threshold, it is passed on to the target correlator, as illustrated in figure 9. The operator feedback is passed back to each sensor genetic algorithm to update its local membership functions.

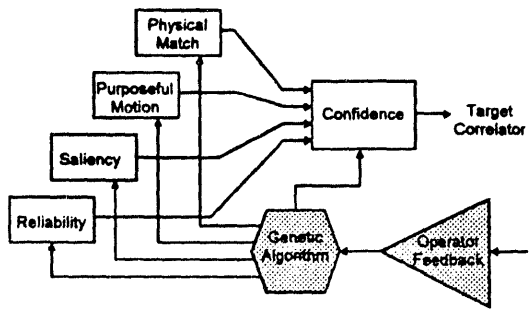

Figure 10 On-line Genetic Application for One Sensor

CURRENT STATUS

The fuzzy logic inference engines have been developed for each of the blocks shown in figure 10. Currently, in the development phase, it takes at least two full days to "hand-tweak" these membership functions. Clearly, this tweaking, or optimization, needs to be automated with genetic algorithms to make it practical for an adaptive on-line application.

A library of genetic algorithms has been developed and are being tested on sample problems. These genetic algorithms will soon be integrated and tested on the fuzzy logic membership functions.

CONCLUSION

A technique for adaptively fusing information from multiple sensors on-line has been presented. The need to perform this adaptation during real-time operation is crucial, as sensor performance constantly changes with the external environment. It is recognized that genetic algorithms do not always converge to the optimal solution. Therefore, the conceptual hardware design includes a separate processor to run multiple genetic algorithms and compare the results with previous settings. This will prevent the genetic algorithm from adapting the system into an unstable state.

The preliminary results are encouraging, and it is expected that the genetic algorithms will provide significant improvement in the adaptability and robustness of the sensor fusion process.

REFERENCES

1. Pritchard, "Advanced Exterior Sensor Data Sheet", Sandia National Laboratories, May, 1994

2. Braude, Eric, "Fuzzy Logic and Expert Systems" Course Notes, April 14, 1994

3. O'Hagen, Michael, "A Fuzzy Decision Maker", Fuzzy Logic '93 Conference Proceedings, *Computer Design* Magazine, 1993

4. Goldberg, David E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Co., 1989

5. Davis, Lawrence, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991

6. Levy, Steven, Artificial Life, Pantheon Books, a division of Random House, Inc., New York, 1992

This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

DATE
FILMED
9 /28/ 94

END