Technical Report: COE-ECE-MMK-1/94

Towards a Formal Theory of Sensor/Data Fusion

Mieczyslaw M. Kokar
Department of Electrical and Computer Engineering

Northeastern University
330 Snell Building
360 Huntington Avenue
Boston, Massachusetts 02115
USA

Jerzy Tomasik
Universite D'Auvergne - Clermont 1

Laboratoire de Logique, Algorithmique et Informatique
France

May 1994

1 Introduction

A *sensory data fusion system (SDFS)* (see Figure 1) receives inputs from a number of different sensors, e.g., vision cameras, infrared cameras, radars, accelerometers, ultrasound. Sensors provide measurements of a number of interrelated variables (*n-tuples*). In mathematical sense, sensors output either functions or relations. The sensors are controlled by the SDFS according to a SDFS *goal*. The goal is one of the external inputs to the SDFS, while controls are one kind of SDFS outputs. Other external interfaces of the SDFS include I/O to/from data and knowledge bases, input from and display to humans, and links to other SDFS's.

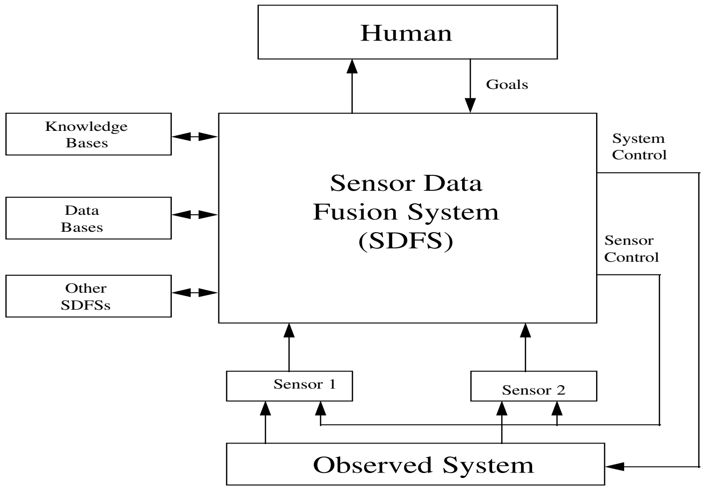

Figure 1: **Sensory Data Fusion System (SDFS)**

In general, the goal of a SDFS is to interpret data received through sensors in a prespecified *target language* understandable to either the user or another system. This language depends on the SDFS's goals. For instance, the goal can be to reconstruct the 3-D scene. In such a case the target language is simply a set of quadruples, where each element of the set represents the XYZ-coordinates of the world and the value of the light intensity at that point. In a tracking application, on the other hand, the goal is to find the state vector of the tracked object, and therefore the target language contains six-tuples representing X, Y, and Z positions, as well as their first derivatives (velocities). If the goal is *object recognition*, then the target language describes types of objects. In a *situation assessment* scenario, the language can be much more complex, and include descriptions of possible courses of events, consequences of particular actions, configurations of objects, features associated with objects and configurations, behaviors, etc. In the sensor fusion literature the target language is often called *world model*.

A natural requirement for a sensor data fusion system is that the interpretation of the data be "correct". Intuitively, this means that the objects identified by the SDFS really exist in the world, that these objects have the features as identified by the SDFS, that the relations recognized by the SDFS really exist in the world, and that the interpretation does not violate the constraints that the world is known to obey, e.g., the laws of physics. In order to maintain the truthfulness of the interpretation, the system (*agent*) must maintain *consistency* of its representation.

To deal with the issues of correctness of interpretations we use the framework of model theory [1]. In particular, we make use of *formal languages* to describe the world and the sensing process and *models* to represent sensor data, operations on data, and relations among the data. Models consist of *domains, operators*, and *relations* among the elements of the domains. We use *theories* to represent symbolic knowledge about the world and about the sensors.

Fusion is then treated as a goal-driven operation of combining two languages, two models, and at the same time, two theories, related to two different sensors, into one combined language, one combined model of the world and one combined theory. This involves combining domains, data, operators, and relations of the two initial models into new domains, data, operators and relations of the *fused model*. For the fusion of theories, this usually involves combining the theorems of the two theories, although one can also think of combining rules of inference. Therefore, fusion is a *formal system operator* that has multiple languages, models and theories for inputs and a single language, model, and theory as the output. The justification for fusion comes from the fact that the fused formal system performs better, in relation to a prespecified performance measure, than any subsystem involving only one sensor. This may be either an added capability of answering queries about the world that cannot be answered using only one sensor, or increased precision of the answer, or an improvement of any other prespecified measure.

This differs from more traditional approaches to SDF [2, 3], where data fusion is treated as extensions to signal processing, detection, estimation, and identification, in the case when multiple sensors are present. In those approaches, such issues like consistency are not dealt with explicitly. Rather, there is an underlying presumption that the operations of fusion are implemented in a consistent way by the human. In our approach, on the other hand, a framework is provided in which the requirement of consistency of representations can be formally and explicitly specified and then can be manipulated by the computer using automatic reasoning techniques.

Our approach is strongly motivated by software engineering. From the software engineering point of view, there exists a need for formal frameworks, in which systems can be represented, checked, and verified. Formal approaches provide more credibility for the system structure, and more predictability of its dynamical behavior. The use of formal methods brings more rigor to the development of complex computer systems[6]. Our goal is to develop formal specification methods oriented specifically towards sensor data fusion. Although specialized architectures have been proposed, for instance [5, 4], approaches to formal specifications of SDF systems, to our best knowledge, are not known in the literature.

From the programming languages point of view we treat our SDF system as a network of objects (in the object oriented paradigm) communicating via messages. Each object is an instance of a class. The goal for fusion is then to find classes from which, through class construction and instantiation, a whole SDF system can be built. We have two major classes of objects: the objects which contain models and the objects which contain theories.

The main goal of our research in sensor data fusion (SDF) is to develop a systematic approach (a formal methodology) to specifying, designing, and implementing software systems for interpreting sensory information and for reasoning about the situation based upon this information and upon available data bases and knowledge bases. Towards this aim, we pursue several subgoals: (1) Development of a theory of sensor/data fusion. This includes formalization of the main concepts of the sensor data fusion theory in terms of *models* and *theories*. (2) Development of a framework for specifying, designing, implementing and testing sensor/data fusion software systems. The basic tools utilized in this approach are computer-aided software engineering (CASE) tools and automatic theorem provers. (3) Development of algorithms for sensor data fusion. (4) Testing sensor data fusion approaches on real tasks and real data.

This paper deals primarily with the first of the issues mentioned above – the theory of sensor data fusion. Although there are several definitions of "fusion" in the subject literature, there does not seem to be an agreement on what is and what is not fusion. Our main objective is to contribute to the understanding of this notion. This understanding is necessary for specifying, designing and implementing SDF systems.

2 Formulation of the Fusion Problem

One of our objectives in this paper is to formulate the problem of fusion in strict mathematical/logical terms. Then, based upon this formulation we can formulate research directions that should lead to the solution of the fusion problem. Fusion involves both quantitative data from sensors, data bases and communication lines, and symbolic data from knowledge bases and intermediate processing layers. The main idea of our approach is to combine logic and mathematics in one software development framework. The mathematical/logical tool that we use for this purpose is *model theory* [1]. Following is our working formulation of this problem.

Notation:

$\mathcal{L}_1$, $\mathcal{L}_2$ – two languages associated with two sensors. These languages include variables, constant symbols, operation and relation symbols.

$\mathcal{T}_1$, $\mathcal{T}_2$ – two formal theories (theorems with inference rules) describing knowledge about the two sensors and about the world.

$\mathcal{M}_1$, $\mathcal{M}_2 \in \mathcal{MM}$ – two models associated with the two languages $\mathcal{L}_1$ and $\mathcal{L}_2$

$\mathcal{G}$ – goals. These are queries about the world that cannot be answered by using only one of the sensors (information sources) but can be answered by using both sensors.

$\mathcal{K}$ – background knowledge. These are constraints on possible interpretation of the received data.

Fusion Problem Formulation

Given the knowledge described above, and inputs from sensors (data sources), construct a *fused theory* $\mathcal{T}$ and a *fused model* $\mathcal{M}$, such that:

1. $\mathcal{M} \models \mathcal{G}$
2. $\mathcal{M} \models \mathcal{T}$
3. $\mathcal{M} \models \mathcal{K}$
4. $\mathcal{M} \in \mathcal{MM}$
5. One of the following constraints may be imposed:
   (a) If $\mathcal{M} \models \mathcal{T}$ then $\mathcal{M}_1 \models \mathcal{T}|_{\mathcal{M}_1}$ and $\mathcal{M}_2 \models \mathcal{T}|_{\mathcal{M}_2}$ (consistency)
   (b) If both $\mathcal{M}_1 \models \mathcal{T}|_{\mathcal{M}_1}$ and $\mathcal{M}_2 \models \mathcal{T}|_{\mathcal{M}_2}$ then $\mathcal{M} \models \mathcal{T}$

The above formulation is declarative in nature and thus is not directly translatable into a computable procedure for searching for models. One of the main research issues is to find procedural characterization of this fusion problem. Currently we are pursuing three kinds of investigations towards this goal: (1) the study of model theory and the search for feasible methods of model construction, (2) development of a software engineering tool for analyzing and developing software specifications of sensor/data systems, and (3) investigation of methods for formal analysis of software specifications using an automatic theorem prover (IMPS).

3 Conclusions

In this report we provided a formal definition of *fusion*. Fusion is treated as a formal operator that is applied to two pairs of objects, two *theories* and their *models* and returns a pair – a *fused theory* and a *fused model*. This definition is just a first step in developing a theory of sensor/data fusion. The definition is declarative in nature, and to be practical, needs to be translated into computationally tractable algorithms that can be applied to real domains.

References

[1] C. C. Chang and H. J. Keisler. *Model Theory*. North Holland, Amsterdam, New York, Oxford, Tokyo, 1992.

[2] D. L. Hall. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Boston - London, 1992.

[3] Ren C. Luo and Michael G. Kay. Multisensor integration and fusion in intelligent systems. *IEEE Transactions on Systems, Man and Cybernetics*, 19-5:901–931, 1989.

[4] J.M. Manyika and H.F. Durrant-White. An information-theoretic approach to management in decentralized data fusion. In *Sensor Fusion V*, volume 1828, pages 202–213. SPIE, 1992.

[5] Stelios C.A. Thomopoulos. Dignet: a self-organizing neural network for automatic pattern recognition, classification, and data fusion. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 478–495. SPIE, 1992.

[6] J. M. Wing. A specifier's introduction to formal methods. *IEEE Computer*, 9:8–24, 1990.