Describing Data Fusion using Category Theory

Steven N. Thorsen∗
Department of Mathematics and Statistics
Air Force Institute of Technology
Wright-Patterson AFB, OH, U.S.A.
Steven.Thorsen@afit.edu

Mark E. Oxley
Department of Mathematics and Statistics
Air Force Institute of Technology
Wright-Patterson AFB, OH, U.S.A.
Mark.Oxley@afit.edu

Abstract – The process of multi-sensor data fusion and its taxonomy is abstracted and described in the language of category theory. Categories are developed for sensors, data sets, processors, feature sets, classifiers, and label sets. Fusion rules are defined and shown to hold a unique role within the various categories. Fusion processes can then be described as the optimization of fusion rules within the appropriate category.

Keywords: Category theory, sensor fusion, sensors, classifiers, data fusion, information fusion.

1 Introduction

Historically, the data (or information) fusion process has been described by an ever-growing collection of mathematical techniques and taxonomies [1]. Improvements have been made by further classifications and better algorithms [2, 3]. This approach has spawned tremendous growth in the field of knowledge, which has led to many useful discoveries, yet lacks a common mathematical description for the data fusion process. While there are sources available which provide a mathematical framework to unite data fusion techniques [1], to our knowledge there is no mathematical explanation of data fusion which poses the data fusion process as an application of mathematical theories. Contrast work done in quantum physics, for example, which uses postulates framed within a state-space (a Hilbert Space) to properly explain the phenomena of the interaction of atomic and molecular particles in the natural universe. This paper is the authors’ first attempts to provide such a description.

1.1 Motivation

Why consider category theory? Our goal is to show regardless of mathematical technique employed, all forms of data fusion have a recognized mathematical structure. This structure is described using category theory. With this recognized structure comes the hope of finding properties, patterns, and theorems regarding the relationships within data fusion. Ultimately, though not in this paper, the authors hope to use this recognized structure to optimize or choose the best fusion rules from among a class of fusion rules. To explain further, the fusion techniques that researchers perform on images may look nothing like fusion techniques performed on, say, speech. Whereas fusion may “look” different on different sensors and classifiers, we conjecture that it is the same when viewed in the abstract sense, and our hope is to be able to optimize the particular fusion rules used in the processes.

1.2 The Multi-Sensor Fusion Process

Consider the simple model of a multi-sensor process using two sensors in Figure 1.

![Figure 1: Simple model of a dual-sensor process](image)

The boxes of the diagram represent sets, while the arrows are mappings between sets. Consider the sets $E_i$, for $i \in \{1, 2\}$, to be event sets. That is, the set of all outcomes of a particular event, such as the event “there is a target present on the ground”. The set $E_i$ is subject to a state at a particular time and in a space of interest. It is useful to think of $E_i$ as the set of all possible outcomes of

∗The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government.
an event occurring within sensor $i$’s field of view. Given $E_i$ thus defined, $s_i$ is then a sensor. A sensor is defined as a mapping from an event set to a data set, $D_i$. A data set could be a radar signature return of an object or a two-dimensional image, for example. In both cases we would like to extract recognizable features. Hence the $p_i$ represent processors which do just that. They are mappings from data sets into feature sets, $F_i$. Finally, from the feature sets we want to determine a label for the event. This is achieved through use of the classifiers $c_i$ which map the feature set into a label set. The label set can be as simple as the two-class set \{target,non-target\} or could have a more complex nature to it, recognizing even the types of targets and non-targets in order to define the battlefield more clearly for the warfighter. Now the diagram in Figure 1 represents a simple sensor process pair involving two sensors, two processors, and two classifiers, but can easily be extended to any finite number. Before we consider what a data fusion diagram looks like, let’s consider the sensor processes of varying event sets. There are two more basic types other than Figure 1 depicted in Figure 2.

![Figure 2: Possible event set combinations for the dual-sensor process](image)

Considering that each event set is a subset of an even larger event set, we see that the first dual-sensor process type in Figure 1, with sufficient modification of the sensor mappings involving restricting the domains, could be replaced by $E = E_1 \cup E_2$, thereby becoming the same as the top diagram of Figure 2. The third type could be replaced by $E = E_1 \times E_2$ again with modifications of the sensor mappings. Thus all types can be replaced by the top diagram of Figure 2, so that our generalized dual-sensor process can be viewed as Figure 3.

![Figure 3: Generalized dual-sensor process](image)

but first we need a notion of what is a fusion rule?

**Definition 1 (Fusion Rule)** A fusion rule is an $n$-ary mapping of $n$ input objects to a single output object of the same class.

Notice that nowhere did we define a fusion rule in terms of producing a better object than the originals! This idea of “betterness” is the goal of the fusion process, not the fusion rule itself. Now we can describe mathematically the fusion of two objects $A$ and $B$ into a third object of the same class, say $C$, using a fusion rule $R$ by the simple fusion diagram

$$ (A, B) \xrightarrow{R} C. \quad (1) $$

### 1.3 Problem Statement

Can the fusion process be described in a unifying mathematical manner? We believe the branch of mathematics known as category theory [4, 5, 6], provides a way to describe the process. Once a process description is attained, the relationships that exist in fusion processes may be explored using the algebraic theorems and properties of category theory. The problem is to determine if it is possible to optimize fusion rules within categories. This is where the idea of “betterness” comes in. Category theory should provide a way of defining what we mean by one fusion rule being “better” than another. If this can be shown, then perhaps there is a way to show that an optimal fusion rule in one category corresponds to an optimal rule in another. We may find that, for example, sensor fusion and classifier fusion are really the same process at an abstract level. Or we may be able to prove that there is a No Free Lunch [7, 8], in as much as there may be no unified rule or analytical approach to finding an optimal fusion rule for any given fusion problem. But for now, we take the first step in describing the categories present in the dual-sensor process described above.
2 Categories

We use notation that is standard in the category literature [4, 5, 6].

Definition 2 (Category) A category \( \mathcal{A} \) is a quadruple

\[
\mathcal{A} = (\mathcal{A}, \text{Hom}_\mathcal{A}, \text{id}_\mathcal{A}, \circ)
\]

where:

- \( \mathcal{A} \) is a collection of objects denoted by \( A, B, \ldots \)
- \( \text{Hom}_\mathcal{A} \) is a collection of homomorphisms between these objects (we will refer to these morphisms as arrows or maps) and \( f \in \text{Hom}_\mathcal{A}(A, B) \) is the arrow from \( A \) to \( B \) and can be written as
  \[
  A \xrightarrow{f} B
  \]
- \( \text{id}_\mathcal{A} \) is the collection of all identity maps of \( \mathcal{A} \)-objects (i.e., if \( A \in \mathcal{A} \) then there exists some \( 1_A \in \text{id}_\mathcal{A} \) with \( \text{Dom}(A) \) and \( \text{Codom}(A) \).)
- \( \circ \) is a binary operation called composition, on which elements of \( \text{Hom}_\mathcal{A} \) can be composed

subject to the following two conditions:

(i) Associativity condition. Given \( A \xrightarrow{f} B, B \xrightarrow{g} C, \) and \( C \xrightarrow{h} D \) then it is true that

\[
h \circ (g \circ f) = (h \circ g) \circ f
\]

(ii) Identity condition. Given any arrow \( A \xrightarrow{f} B \) between two objects \( A, B \in \mathcal{A} \), then identities \( 1_A, 1_B \in \text{id}_\mathcal{A} \) exist such that

\[
f \circ 1_A = f \quad \text{and} \quad 1_B \circ f = f
\]

2.1 The Object-Fusion Category

Since our objective is to provide a framework to examine fusion within category theory, we introduce our first category. In this category, we will not explicitly define the objects of fusion specifically, for reasons which shall become apparent later.

Definition 3 Given that \( \mathcal{O} \) is a class of objects, let \( \text{Cart}(\mathcal{O}) = \{ \mathcal{O}^n \mid \mathcal{O} \in \mathcal{O} \) and \( n \in \mathbb{N} \} \), the set of all finite cartesian products of the same object, for all objects in \( \mathcal{O} \). That is

\[
\mathcal{O}^n = \underbrace{\mathcal{O} \times \mathcal{O} \times \ldots \times \mathcal{O}}_{n \text{ times}}.
\]

Let \( \mathcal{O} = \text{Cart}(\mathcal{O}) \) for simplicity of notation.

Theorem 4 There exists an Object-Fusion category \( \mathcal{O} \).

Proof. Define the quadruple

\[
\mathcal{O} = (\mathcal{O}, \text{Fus}_\mathcal{O}, \text{id}_\mathcal{O}, \circ)
\]

by the following:

- \( \mathcal{O} \) is a class of objects as in Definition 4
- \( \text{Fus}_\mathcal{O} = \{ f \mid f : \mathcal{O}^n \rightarrow \mathcal{O}, \mathcal{O} \in \mathcal{O} \) and \( n \in \mathbb{N} \}. Since \( \text{Fus}_\mathcal{O} \) is a collection of arrows between \( \mathcal{O} \)-objects, it satisfies for the second element of a category quadruple, so it replaces the \( \text{Hom}_\mathcal{O} \) notation. \( \text{Fus}_\mathcal{O} \) comprises a set of fusion rules on \( \mathcal{O} \)
- \( \text{id}_\mathcal{O} = \{ 1_A \mid 1_A(x) = x, \forall x \in A, \forall A \in \mathcal{O} \}. Identities exist, since an identity is a 1-ary fusion rule
- \( \circ \) is composition of functions

Now to show that composition is associative, notice that for any \( f, g, h \in \text{Fus}_\mathcal{O}, \) only one, say \( h, \) can have domain \( \mathcal{O}^n \in \mathcal{O} \) for some \( \mathcal{O} \in \mathcal{O} \) and \( n > 1 \). Therefore, either \( f \) or \( g \) must have \( \mathcal{O} \) as both domain and codomain. Assume \( g : \mathcal{O} \rightarrow \mathcal{O}. \) Thus let \( (f \circ g) \circ h \) be defined for some \( \mathcal{O} \in \mathcal{O}. \) Then we have that for any \( x \in \mathcal{O}^n \)

\[
[(f \circ g) \circ h](x) = [f \circ g](h(x)) = f(g(h(x))) = f([g \circ h](x)) = [f \circ (g \circ h)](x)
\]

Hence we have

\[
(f \circ g) \circ h = f \circ (g \circ h),
\]

showing composition to be associative. Now let \( n \in \mathbb{N} \) and \( \mathcal{O}^n \xrightarrow{f} \mathcal{O} \) be defined. Then there exists \( 1_{\mathcal{O}^n}, 1_\mathcal{O} \in \text{id}_\mathcal{O} \) such that for each \( x \in \mathcal{O}^n \) we have that

\[
[f \circ 1_{\mathcal{O}^n}](x) = f(1_{\mathcal{O}^n}(x)) = f(x).
\]

Thus

\[
f \circ 1_{\mathcal{O}^n} = f.
\]

Furthermore, for each \( y \in \mathcal{O} \) we have that

\[
[1_\mathcal{O} \circ f](y) = 1_\mathcal{O}(f(y)) = f(y)
\]

so that

\[
1_\mathcal{O} \circ f = f.
\]

Therefore \( \mathcal{O} \) is a category. Since it is a category of objects in \( \mathcal{O} \) with fusion rules as arrows, we will denote it as the Object-Fusion category.
3 Categories in the Fusion Process

Now it will become clear as to why we made \( O \) an arbitrary class of objects. We have the foundation necessary to construct six categories of fusion. The first three are immediate, since the objects are readily available from the fusion process as sets of data, features, and labels. The second three require a simple observation to generate similar results.

**Theorem 5** There exists a category \( \mathcal{D} \) of Data-Fusion.

*Proof.* From Theorem 4, let \( O = D \), where \( D \) is a class of data sets. Let \( D = Cart(D) \) then we have that

\[
\mathcal{D} = (\mathcal{D}, Fus_{\mathcal{D}}, \text{id}_{\mathcal{D}}, \circ)
\]

defines a category.

**Theorem 6** There exists a category \( \mathcal{F} \) of Feature-Fusion.

*Proof.* From Theorem 4, let \( O = F \), where \( F \) is a class of feature sets. Let \( F = Cart(F) \) then we have that

\[
\mathcal{F} = (\mathcal{F}, Fus_{\mathcal{F}}, \text{id}_{\mathcal{F}}, \circ)
\]

defines a category.

**Theorem 7** There exists a category \( \mathcal{L} \) of Label-Fusion (Decision-Fusion).

*Proof.* From Theorem 4, let \( O = L \), where \( L \) is the class of label sets. Let \( L = Cart(L) \) then we have that

\[
\mathcal{L} = (\mathcal{L}, Fus_{\mathcal{L}}, \text{id}_{\mathcal{L}}, \circ)
\]

defines a category. Now here are the observations that lead to the next three categories:

**Definition 8** Let \( A \) and \( B \) be two classes of objects. Then define the collection of mappings

\[
B^A = \{ f \mid f : A \rightarrow B, A \in A, B \in B \}.
\]

Let \( E \) be the class of event sets. We define other classes.

**Definition 9** Call \( S = D^E \) the class of sensors.

**Definition 10** Call \( P = F^D \) the class of processors.

**Definition 11** Call \( C = L^F \) the class of classifiers.

**Theorem 12** There exists a category \( S \) of Sensor-Fusion.

*Proof.* From Theorem 4, let \( O = S \), where \( S \) is the class of sensors. Let \( S = Cart(S) \) then we have that

\[
S = (S, Fus_{S}, \text{id}_{S}, \circ)
\]

defines a category.

**Theorem 13** There exists a category \( P \) of Processor-Fusion.

*Proof.* From Theorem 4, let \( O = P \), where \( P \) is the class of processors. Let \( P = Cart(P) \). Then we have that

\[
P = (P, Fus_P, \text{id}_P, \circ)
\]
defines a category.

**Theorem 14** There exists a category \( C \) of Classifier-Fusion.

*Proof.* From Theorem 4, let \( O = C \), where \( C \) is the class of classifiers. Let \( C = Cart(C) \). Then we have that

\[
C = (C, Fus_C, \text{id}_C, \circ)
\]
defines a category.

4 Taxonomies and Fusion Categories

It is interesting to note that the categories constructed above are relevant to the fusion process input/output (I/O) characteristics identified by Desarathy in his book *Decision Fusion* [2, pp. 9-11]. The correspondences are as follows:

<table>
<thead>
<tr>
<th>Desarathy’s I/O Fusion Characteristics</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data In − Data Out</td>
<td>Data − Fusion</td>
</tr>
<tr>
<td>Data In − Feature Out</td>
<td>Processor − Fusion</td>
</tr>
<tr>
<td>Feature In − Feature Out</td>
<td>Feature − Fusion</td>
</tr>
<tr>
<td>Feature In − Decision Out</td>
<td>Classifier − Fusion</td>
</tr>
<tr>
<td>Decision In − Decision Out</td>
<td>Label − Fusion</td>
</tr>
<tr>
<td>N/A</td>
<td>Sensor − Fusion</td>
</tr>
</tbody>
</table>

Interesting enough, there is no correspondence to the Sensor-Fusion category! In our opinion this is due to fusion researchers considering only fusing objects (such as data sets) rather than fusing the mappings themselves. Perhaps this categorical construction of the multi-sensor fusion process will inspire research exploring how to better define the mappings of sensors, processors, and classifiers. At this point, we know of no direct way to fuse these objects; rather, the standard seems to be to define a pointwise rule of combination (fusion rule) and a metric which can be applied to the ranges of the original mappings and the fused results. The fused result is obtained and compared to the same metric applied to the original mappings. If the result is no worse than the originals, then it is declared to be fusion.
5 Conclusion

After defining a simple multi-sensor process, we have shown the process can be described using the mathematical language of category theory. In this language there are categories of Sensor-Fusion, Data-Fusion, Processor-Fusion, Feature-Fusion, Classifier-Fusion, and Label-Fusion all of which share the same algebraic structure, but differ in the objects and arrows. The arrows were defined as fusion rules specifically leaving out any notion of “bitterness”, defining which rules are superior to others. Having constructed the foundation, future work will focus on whether or not there exist methods to optimize fusion rules in order to frame the fusion process as an optimization problem.

6 Acknowledgements

We wish to acknowledge the financial support of the Air Force Research Laboratory, Sensor Directorate, Automatic Target Recognition Branch (AFRL/SNAT) at Wright-Patterson AFB, Ohio and also the Defense Advanced Research Program Administration (DARPA), Mathematical Sciences Division.

References


