Luís Miguel Gonçalves Miranda

Licenciado em Ciências da Engenharia Electrotécnica e de Computadores

Data Fusion with Computational Intelligence techniques: a case study of Fuzzy Inference for terrain assessment

Dissertação apresentada para obtenção do Grau de Mestre em Engenharia Electrotécnica e de Computadores, pela Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia

Orientador :	José Manuel Fonseca, Professor Auxiliar, FCT/UNL
Co-Orientador :	Rita Almeida Ribeiro, Professora Associada, FCT/UNL

	Júri:
Presidente:	Tiago Oliveira Cardoso, FCT/UNL
Arguente:	Luís Miguel Correira, FCUL
Vogal:	José Manuel Fonseca, FCT/UNL

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março, 2014

Data Fusion with Computational Intelligence techniques: a case study of Fuzzy Inference for terrain assessment

Agradecimentos

Agradeço a todos que directa ou indirectamente contribuiram para a realização deste trabalho, desde os meus companheiros de curso aos meus colegas e orientadores do UNINOVA-CA3.

Resumo

Com a constante progressão tecnológica está inerente o aumento exponencial dos arquivos digitais com todo o tipo de dados. Esses dados, por si só, podem não ter um significado preciso e podem até mesmo ser impossíveis de processar sem auxílio de ferramentas específicas. A fusão de dados contribui para esta problemática através da combinação de dados de forma a gerar informação útil para quem os analisa.

Dentro da fusão de dados existem inúmeras abordagens e metodologias de processamento de dados, sendo aqui dado destaque àquela que em certa medida mais se assemelha ao conhecimento impreciso efectuado por um humano, o raciocínio difuso. Esta metodologia é aplicada nas mais variadas áreas, inclusivé como sistema de inferência em sistemas baseados em regras para escolha de local de aterragem de naves espaciais usando mapas de risco. Para tal é importante o uso de sistemas de inferência difusa, onde o problema é modelado através de um conjunto de regras linguísticas, conjuntos difusos, funções de pertença e demais informação.

Assim, nesta tese foram desenvolvidos um sistema de inferência difuso, para detecção de locais de aterragem seguros utilizando fusão de mapas, e uma ferramenta de visualização de dados. Deste modo ficam facilitadas a classificação e validação da informação que se tem em mãos.

Palavras Chave

Fusão de Dados, Raciocínio Difuso, Prevenção de Riscos, Sistemas de Inferência Difusa, Visualização de Dados

Abstract

With the constant technology progression is inherent storage of all kinds of data. Satellites, mobile phones, cameras and other type of electronic equipment, produce on daily basis an amount of data of gigantic proportions. These data alone may not convey any meaning and may even be impossible to interpret them without specific auxiliary measures. Data fusion contributes in this issue giving use of these data, processing them into proper knowledge for whom analyzes.

Within data fusion there are numerous processing approaches and methodologies, being given here highlight to the one that most resembles to the imprecise human knowledge, the fuzzy reasoning. These method is applied in several areas, inclusively as inference system for hazard detection and avoidance in unmanned space missions. To this is fundamental the use of fuzzy inference systems, where the problem is modeled through a set of linguistic rules, fuzzy sets, membership functions and other information.

In this thesis it was developed a fuzzy inference system, for safe landing sites using fusion of maps, and a data visualization tool. Thus, classification and validation of the information are made easier with such tools.

Keywords

Data Fusion, Fuzzy Reasoning, Hazard Detection and Avoidance, Fuzzy Inference Systems, Data Visualization

Contents

List of Figures

List of Tables

Acronyms

CA3 Computational Intelligence Research Group

CBM Condition-Based Maintenance

DoD Department of Defense

DSET Dempster-Shafer Evidence Theory

EDL Entry, Descent and Landing

ESA European Space Agency

FCL Fuzzy Control Language

FIF Fuzzy Information Fusion

FIS Fuzzy Inference Systems

FRA Fuzzy Reasoning Algorithm

GNC Guidance, Navigation and Control

GUI Graphical User Interface

HDA Hazard, Descent and Avoidance

IPSIS Intelligent Planetary Site Selection

ITS Intelligent Transportation Systems

JDL Joint Directors of Laboratories

KF Kalman Filter

NASA National Aeronautics and Space Administration

OODA Observe, Orient, Decide, and Act

OOSM Out-of-Sequence Measurements

PSO Particle Swarm Optimization

UAV Unmanned Aerial Vehicles

1

Introduction

Contents

Human beings have the biological ability to efficiently combine data, having evolved that capability by using multiple senses which interact with the surrounding environment, and to interpret situations, take decisions and thus act, improving its ability to survive. Integrating sensory data from sight, sound, smell, taste and touch, supported by *a priori* knowledge and a cognitive process, makes the human brain an excellent example of a sophisticated fusion system (Hall and Llinas, 1997, Johansson, 2003, Nilsson, 2008).

This natural fusion system performed by the human brain is remarkably effective in today's contexts. (Ng, 2003) addresses this issue adding "that the ultimate goal is for an intelligent system to be modeled after human intelligence". Yet, the ability to engineer a system that match the functions of the brain in processing and controlling all the sensory systems and analyzing the input information is not a simple task.

The proliferation of available data ("Big Data") and new technologies and their subsequent integration into our lives, creates new challenges and drives the process of decision making to become exponentially more complex. In many cases, the increasing amounts of available data, have to be filtered to create useful information to enable more informed, accurate and successful decision (Nilsson, 2008, Fan and Bifet, 2013). (Biermann et al., 2004) makes clear that thorough functional analysis of both operational and processing aspects are a necessary prerequisite for understanding the deep and sophisticated nature of information and the different processing levels (see Figure 1.1) required for producing intelligence from data.

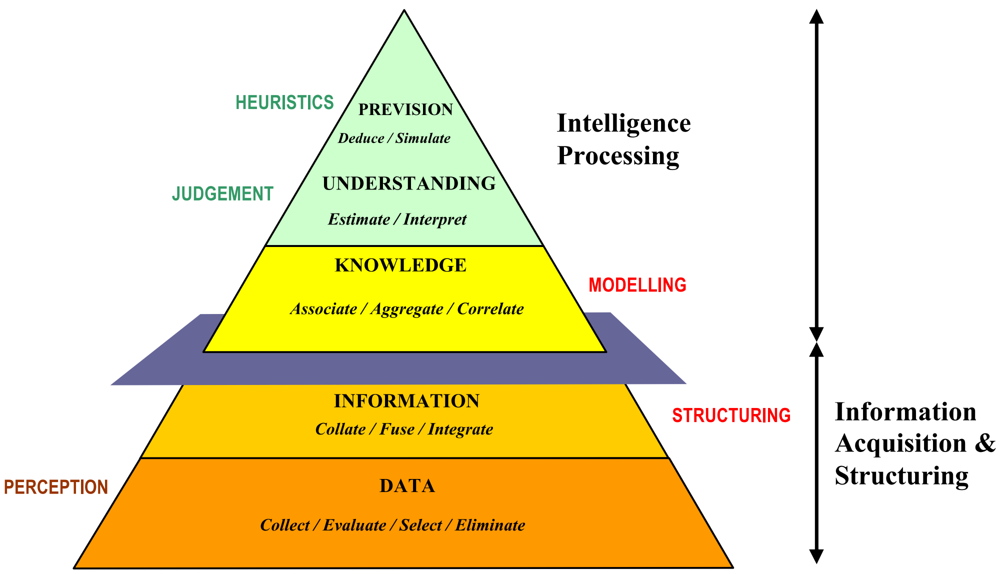


Figure 1.1: The Knowledge Pyramid. Taken from (Biermann et al., 2004).

The bottom of the pyramid is designated by "Information Acquisition & Structuring" and is defined, in most cases, by the set of sensors of a process for data acquisition/measurement (entitled "Perception") of a environment. The information layer (entitled "Structuring") is where the data

fusion/integration is made by a method/algorithm. The Knowledge Pyramid's top encompasses the "Intelligence Processing", i.e. the initial measurements which have been transformed into knowledge with associations and correlations between data that can be understood, be put under judgment and predict future behaviors. However, the decision process in complex systems takes, in the majority of cases, a great effort of our cognitive capabilities which may be the main problem of predicting how the system will react to external manipulation, and understand the multiple interactions between system's variables (Pfister and Bohm, 2008).
An intelligent agent will never get all the relevant information but will be impacted by partial, false, unreliable, irrelevant, redundant pieces of information which he will have to filter (Biermann et al., 2004). The human brain permanently selects and inserts relevant information from its internal, mental understanding and model of the situation as it is perceived so far, and the experience gained from "similar" problems in order to correlate and aggregate it all to a reasonable picture of the situation deciphering the meaning of all pieces of input data (Biermann et al., 2004).
Data fusion with the purpose of improving decision-making requires solutions for matters such, as stated by (Nilsson, 2008), "while information technology can transform a data poor situation into a data rich environment, the fact remains that data needs to be fused and analyzed effectively and efficiently, in order to provide appropriate information for intelligent decision making."
In the 1950s and 1960s, the search for practical methods of merging images from various sensors to provide a single composite image, which could be used to better identify natural and man-made objects, gave data fusion a scientific application perspective (Wang et al., 2005). With the emergence of new sensors, advanced processing techniques and improved processing hardware, real-time[1] fusion of data started to be increasingly viable. Just as the advent of symbolic processing computers (e.g. LISP Machine developed at Massachusetts Institute of Technology (Bawden et al., 1977, Greenblatt et al., 1980)) in the early 1970s, provided an impetus to artificial intelligence (Liggins and Chang, 2009). Its rapid growth, *per se*, started in the late 1980s with the United States Department of Defense (DoD) conducting much of the early research on this technology and exploring its usefulness in ocean surveillance, air-to-air and surface-to-air defense, and battlefield intelligence (Dailey and Lin, 1996, Steinberg et al., 1999, Macii et al., 2008).
Nowadays, several studies and projects are emerging in more diversified areas of application and environment than just the military. Much research and applications are in the Unmanned Vehicles area, including for instance Unmanned Ground Vehicles (e.g. (Blasch et al., 2006)), Unmanned Underwater Vehicles (e.g. (Geder et al., 2009)), Unmanned Surface Vehicles (e.g. (Savitz et al., 2013) and the Portuguese project Ziphius (Azorean, 2014)), Unmanned Aerial Vehicles (UAV) (e.g. (Gonçalves-Coelho et al., 2007, de Fátima Bento, 2008, Morgado and de Sousa, 2009, Hormigo and Araújo, 2013) in a Portuguese context) and Unmanned Space Missions (e.g. (Martin et al., 2007, ESA, 2014a) and recently, the Curiosity Mars Rover (NASA, 2014a)). As

[1] Relating to a system in which input data is processed within milliseconds so that it is available virtually immediately as feedback to the process from which it is coming (Dictionaries, 2014c).

in National Aeronautics and Space Administration (NASA) Curiosity Mars Rover case (NASA, 2014a), spacecraft lander missions are the present and the future of space exploration missions (with European Space Agency (ESA) ExoMars Mars Rover launch due to 2016 (ESA, 2014b)). Unmanned Space Missions represent a very specific topic with a strong component of data fusion on matters such as assessment of the landing site, crucial to ensure safe landing missions, particularly for NASA and ESA. With the advent of space missions and exploration of the universe through unmanned spacecrafts, several data fusion frameworks for terrain safety assessment are emerging, being of particular interest for this dissertation those that handle fuzzy reasoning such as (Howard and Seraji, 2004, Serrano et al., 2006, Seraji and Serrano, 2009). Fuzzy reasoning essentially "mimics" the imprecise knowledge of a human expert (Babuska, 2003).

1.1 Motivation

The work in this dissertation falls under FUSION - Sensor Data Fusion Hazard Mapping and Piloting project (CA3, 2013b) where it was necessary to test different data fusion methods and compare their performance and results. In this context, a fuzzy reasoning algorithm was implemented as a potential candidate method for data fusion in spacecraft landing site assessment.

Being the analysis of the obtained results a complicated task, it was created a data visualization tool to help both decision makers and engineers to analyze and verify those results.

1.2 Objectives and Contributions

In this dissertation we start with an overview of the data fusion domain and all that it entails, from the different main types of data fusion to the applied algorithms and areas of application.

After, we present an implementation of a fuzzy reasoning algorithm for hazard maps fusion within spacecraft landing site assessment, which was based in the works (Howard and Seraji, 2004, Serrano et al., 2006, Seraji and Serrano, 2009). Then its results and performance were compared with those obtained by the Intelligent Planetary Site Selection (IPSIS) algorithm (CA3, 2013d) for the same input dataset.

Last but not the least, was also created an interactive data visualization tool, named Oracle Viewer, for visualizing output's full detail of IPSIS exhaustive search methodology of best landing site, which was also tested in other UAV.

1.3 Structure of the Dissertation

This thesis is organized in the following chapters:

1. Introduction
2. State of the Art
3. Hazard Detection and Avoidance
4. Developed Work
5. Results and Discussion
6. Conclusions

In Chapter 1 is described the introduction of the subject, as well as its objectives and contributions and the present explanation of dissertation organization.

In Chapter 2 is presented an overview of the state of the art of data fusion, including the challenging problems regarding its implementation, a description of its branches, architectures, models, algorithms/methods and applications.

Chapter 3 focus in Hazard Detection and Avoidance (HDA) highlighting the project FUSION, for which the tools presented in this dissertation were developed, and a brief explanation of IPSIS algorithm for safe site selection.

Chapter 4 presents the developed tools, applying Takagi-Sugeno inference scheme and all the knowledge inherent of its application in the Fuzzy Reasoning Algorithm (FRA). In this chapter it is also described a data visualization and analysis tool developed in MATLAB®, called Oracle Viewer. The operation of these tools is analyzed in detail.

In Chapter 5 it is compared and scrutinized the obtained results of the FRA and the IPSIS algorithm. Thus, it is also taken into account the computational requirements of each algorithm.

At last, in Chapter 6 the conclusions of the work performed are presented, as well as some future developments that can be done to improve the work presented in this dissertation.

2

State of the Art

Contents

This chapter contains an overview of the state of the art of data fusion with all that it entails, from its origin and definition, the problems that may result from its implementation and those that it proposes to solve, its branches, architectures, models, algorithms/methods and applications.

2.1 What is Data Fusion?

Data fusion is a wide ranging subject and many definitions are available in the literature. While there is not one commonly referenced definition of data fusion, there is a general consensus of what fusing data means. (Liggins et al., 2008) propose "data fusion techniques combine data from multiple sensors and related information to achieve more specific inferences than could be achieved by using a single, independent sensor." (Mitchell, 2007) suggests that multi-sensor data fusion is "the theory, techniques and tools which are used for combining sensor data, or data derived from sensory data, into a common representational format ... in performing sensor fusion our aim is to improve the quality of the information, so that it is, in some sense, better than would be possible if the data sources were used individually." (Bleiholder and Naumann, 2009) states that "data fusion is the process of fusing multiple records representing the same real-world object into a single, consistent, and clean representation." (Hyder et al., 2002, Ng, 2003, Lee et al., 2010) for instance, give other definitions of data fusion. In the overall, there is a common understanding that data fusion encompasses a wide variety of activities that involve using multiple data sources. In short, data fusion can be defined as any process of aggregating data from multiple sources into a single composite with higher information quality (Ribeiro et al., 2014).

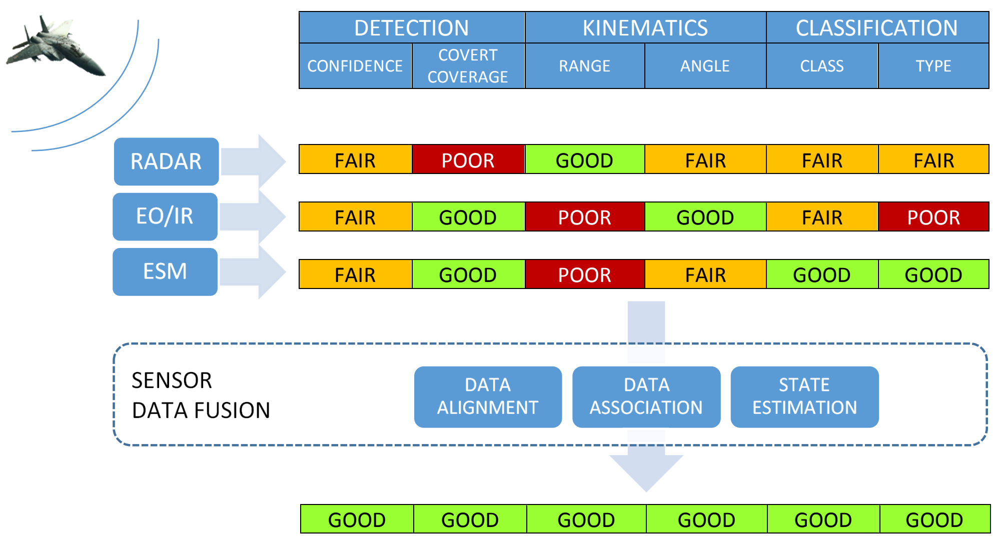


Figure 2.1: Fighter aircraft with multiple sensors uses data fusion for integrating detection, kinematics and classification assessments (EO/IR - Electro-optical/Infra-Red; ESM - Electronic Support Measures). Adapted from (Steinberg et al., 1999).

Data fusion uses overlapping information to determine relationships among data (data association) and synergistic differences in the data to improve the estimate/assessment of a reported environment (state estimation). As such, data fusion can enable improved estimation of situations and, therefore, improved responses to situations, as for instance, the case of a fighter aircraft during flight (see Figure 2.1) (Steinberg et al., 1999).

The results of automated data fusion processes are generally employed to support human decision-making in complicated scenarios by refining and reducing the quantity of information that system operators need to examine to achieve timely, robust, and relevant assessments and projections of the situation (Steinberg et al., 1999, Macii et al., 2008).

2.2 Challenging Problems of Data Fusion

Despite the many recent advances in data fusion, due to the intrinsic imperfections and diversity of sources and contexts, there is still much to be accomplished to obtain a full-proof data fusion method (Ribeiro et al., 2014). (Liggins et al., 2008, Ch. 1) in revision of (Hall and Steinberg, 2000) work stated some important conclusions about the challenges/issues in data fusion such as:

- There is *still* no substitute for a good sensor (and a good human to interpret the results);
- Downstream processing *still* cannot absolve upstream errors (or lack of attention to the data);
- Not only may the fused result be worse than the best sensor result, but failure to address pedigree, information overload, and uncertainty may really fowl up things;
- There are *still* no magic algorithms;
- There will never be enough training data;
- Both research and implementation started at "the wrong end", as we have "started at the data side or sensor inputs" to progress toward the human side and not the opposite.

There are many reasons why we need data and information fusion systems as noted by (Luo and Kay, 1989, Brooks and Iyengar, 1998, Ng, 2003, Mitchell, 2007, Liggins et al., 2008). Yet, to benefit from the advantages of fusing data one must understand and resolve the challenging problems inherent of a fusion system. As stated before, there are a number of issues that make data fusion a challenging task. The majority of these issues arise from the data to be fused, uncertainty and diversity of the sensor technologies, and the nature of the application environment as discussed in (Hall and Garga, 1999, Hall and Steinberg, 2000, Khaleghi et al., 2009, 2013). Essentially, from (Liggins et al., 2008, Khaleghi et al., 2013) it is possible to identify the core of the main concerns and issues inherent to data fusion:

- Data uncertainty: data provided by any type of sensor is always affected by some level of impreciseness and noise in the measurements. Data fusion algorithms should be able to exploit the data redundancy to reduce such effects and help improve accuracy;
- Outliers and spurious data: data uncertainties from sensors are an unavoidable consequence of their use and can also be caused by ambiguities and inconsistencies present in the environment, and from the inability to distinguish between them. Detecting these data discrepancies is an important task since outliers can change the results of a data analysis;
- Conflicting data: different sources may lead to conflicts because of incomplete, erroneous and out-of-date data (Dong and Naumann, 2009). To avoid producing counter-intuitive results, highly conflicting data must be treated with upmost care;
- Data modality: sensor networks may collect the homogeneous or heterogeneous data measurements of a phenomenon. Both cases must be handled by a data fusion system;
- Data alignment/registration: integration of a sensor's data transformation common frame, instead of sensor's local frames, is essential to align data into a common spatial and temporal reference before fusion occurs. Alignment problems are often referred to as sensor registration and deals with the calibration error induced by individual sensor nodes. In practice, data registration is of critical importance to the successful deployment of fusion systems. Several methods of sensor registration are discussed in (Liggins et al., 2008, Ch. 6);
- Data correlation: this issue is particularly important and common in distributed fusion settings, e.g. wireless sensor networks, as for example some sensor nodes are likely to be exposed to the same external noise biasing their measurements. (Board, 1996) considers that sensor measurements should be compared with correlation metrics to score each alternative assignment hypothesis. If such data dependencies are not accounted for, the fusion algorithm may suffer from over/under confidence in results;
- Data association: new problems arise from the major complexity introduced by using multi-target tracking instead of single-target tracking. The difficulties may come in two forms: measurement-to-track and track-to-track association. The former refers to the problem of identifying from which target, if any, each measurement is originated, while the latter deals with distinguishing and combining tracks, which are estimating the state of the same real-world target. A different technique of data association is proposed by (Liggins et al., 2008, Ch. 13);
- Processing framework: data fusion processing can be performed by various fusion system's architectures (see Section 2.4). The architecture chosen must meet problem requirements to enhance the efficiency of the system;

- Operational timing: in a system comprising of several sensors with specific aspects associated, may occur different rates of data transfer in different operation frequencies, as in the case of homogeneous sensors. It is fundamental that a data fusion method can process data in parallel and incorporate multiple time scales in order to deal with such timing variations in data. In distributed fusion architectures (see Section 2.4.3), different parts of the data may traverse different routes before reaching the central fusion node, which may cause out-of-sequence arrival of data. Is then fundamental to prevent this issue to avoid potential performance degradation, which occurs especially in real-time applications;
- Static vs. dynamic phenomena: the phenomenon under analysis may be static (invariant) or dynamic (variant) in terms of time variance. In the latter case, it may be necessary for the data fusion algorithm to incorporate a historical database of recent measurements into the fusion process. Also the frequency of variations must be considered in design or selection of the appropriate fusion approach;
- Data dimensionality: the measurement data could be preprocessed, either locally at each of the sensor nodes (as in Section 2.4.2) or globally at the fusion node (as in Section 2.4.1) to be compressed into lower dimensional data, assuming a certain level of compression loss is allowed. This preprocessing stage is beneficial as it enables saving on the communication bandwidth and power required for transmitting data;
- Cost effectiveness: to build a single sensor that can perform multiple functions is often more expensive than to integrate several simple and less expensive sensors with specific functions. Phenomenon observation requirements should be properly analyzed to conclude which is the best option.

Many problems of fusing data have been identified and solutions have been studied, but as stated previously, there is no magic algorithm capable of addressing all the aforementioned challenges (Bachmann, 2011). Some of the listed issues are addressed with further detail in Section 2.6.1.

2.3 Data Fusion Branches

As theory and applications have evolved over the years, it is important to state that similar underlying problems of data integration and combination occur in a very wide range of engineering, analysis and cognitive situations (Steinberg et al., 1999).

Data fusion is a multidisciplinary research area with techniques and approaches widely applied in areas from many diverse fields as stated in (Dailey and Lin, 1996, Khaleghi et al., 2013, Ribeiro et al., 2014) and (Liggins et al., 2008, Ch. 1). In its current state, the technology can combine multiples sources of information and data, as well as sensor data of many types, including radar, infrared sonar, and visual information. Data fusion technology has rapidly advanced from a loose collection of related techniques to an emerging true engineering discipline with standardized terminology, collections of robust mathematical techniques, and established system design principles (Dailey and Lin, 1996) (Liggins et al., 2008, Ch. 1).

Data fusion can be viewed from different perspectives by different domains, being the most common branches: image fusion, multi-sensor fusion and information fusion. The common denominator for all three domains is that the different sources must address/represent the same subject (e.g. scene, action, local, event, etc.) and that they share available techniques from statistical, probabilistic or computational intelligence methods (Ribeiro et al., 2014).

2.3.1 Image Fusion

The main objective of image fusion is to reduce uncertainty and redundancy while maximizing relevant information, by combining different image representations of the same scene (Mora et al., 2013). (Ardeshir Goshtasby and Nikolov, 2007) identifies image fusion as "the process of combining information from two or more images of a scene into a single composite image that is more informative and is more suitable for visual perception or computer processing". The images are usually provided by different equipments and combine not only position and geometry, but also some semantic interpretation (Mora et al., 2013). This approach has several benefits such as wider spatial and temporal coverage, reduced uncertainty and improved reliability, which may increase the robustness of the system performance (Ardeshir Goshtasby and Nikolov, 2007). Image fusion algorithms can be divided in two categories: pixel based and feature-symbolic based (Ribeiro et al., 2014). Pixel based algorithms are more common and work either in the spatial domain or in the transformation domain (Ribeiro et al., 2014). Although pixel based fusion is a local operation, domain algorithms create the fused images globally, for instance feature-based algorithms (symbolic) typically segment the images into regions and then fuse the regions using the images intrinsic properties (Mora et al., 2013, Ribeiro et al., 2014). There are many suitable algorithms and techniques proposed in the literature to deal with the two levels of image fusion processes, such as: multi-resolution analysis, hierarchical image decomposition, pyramid techniques, wavelet transform, artificial neural networks, biological inspired models, fuzzy reasoning, Principal Component Analysis, known as PCA, and so forth (Ribeiro et al., 2014). Image fusion algorithms have particular relevance in areas such as medical, remote sensing, industrial, and the military (Ardeshir Goshtasby and Nikolov, 2007).

2.3.2 Multi-Sensor Fusion

Multi-sensor fusion is the process of fusing data provided by sensors and consequently, its objective is to integrate data measurements extracted from different sensors combining them into one representation (Ribeiro et al., 2014). In principle, fusion of multi-sensor data provides significant advantages over single source data. Along with statistical advantages gained by combining same-source data (e.g. improved estimate of a physical phenomena via redundant observations), using a set of multiple types of sensors may increase the accuracy with which a measurement can be observed and characterized (Hall and Llinas, 1997).

According to (Mitchell, 2007), performance improvements on multi-sensor data fusion are summarized in four different ways: a greater granularity in the representation of information, greater certainty in data and results, elimination of noise and errors producing greater accuracy, and more complete view on the environment. Nevertheless, often the performance of a multi-sensor system can be worse than one with individual sensors, a situation identified as Catastrophic Fusion (Mitchell, 2007). This can be caused by a set of sensors being designed to operate correctly only under certain conditions (Mitchell, 2007).

Most approaches of multi-sensor data fusion use statistical[1] methods (Kalman filters) and probabilistic[2] techniques (Bayes theory) (Ribeiro et al., 2014). Other strategies include the use of hybrid methods to combine different multi-sensor fusion techniques, taking the advantages of the individual approaches and mitigating their flaws (Ribeiro et al., 2014).

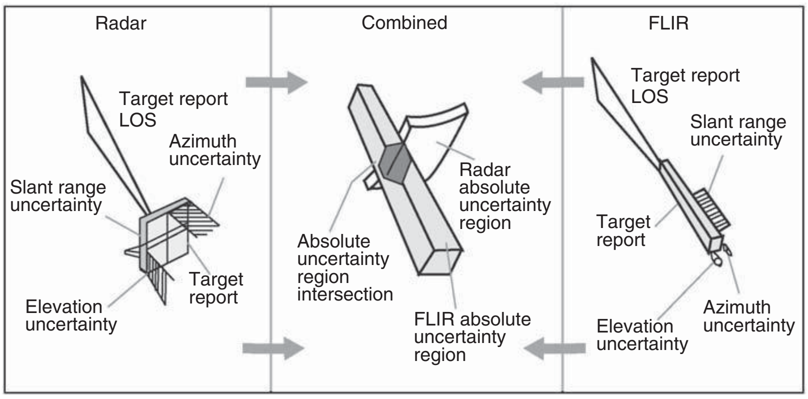


Figure 2.2: Pulsed radar and a Forward-Looking InfraRed (FLIR) imaging sensor with data correlation. Taken from (Liggins et al., 2008, Ch. 1).

Figure 2.2 presents a combined observation system with a pulsed radar and a FLIR. Considering the object under observation a moving object, such as an aircraft, the radar provides the ability to accurately determine the aircraft's range but has a limited ability to determine the angular direction of the aircraft. On the other hand, the FLIR can accurately determine the aircraft's angular

[1] Statistic: a fact or piece of data obtained from a study of a large quantity of numerical data (Dictionaries, 2014e).

[2] Probability: the quality or state of being probable; the extent to which something is likely to happen or be the case (Dictionaries, 2014b).

direction, but is unable to measure range (Hall and Llinas, 1997). If these two observations are correctly associated (as in "Combined"), then the combination of the two sensors data provides an improved determination of location than could be obtained by either of the two independent sensors. This results in a reduced error region as shown in the fused or combined location estimate. A similar effect may be obtained in determining the identity of an object based on observations of an object's attributes. For example, there is evidence that bats identify their prey by a combination of factors that include size, texture (based on acoustic signature), and kinematic behavior (Hall and Llinas, 1997).

2.3.3 Information Fusion

Information fusion is a multi-level process of combining different data to produce fused information (Ribeiro et al., 2014). (Nilsson and Ziemke, 2007) made a reference about how Dasarathy (Dasarathy, 2001) viewed information fusion. To Dasarathy, information fusion refers to " ... exploiting the synergy in the information acquired from multiple sources (sensor, databases, information gathered by humans, etc.) such that the resulting decision or action is in some sense better ... than would be possible if any of these sources were used individually without synergy exploitation" (Nilsson and Ziemke, 2007). Essentially, in information fusion the preprocessed outputs of each single source are combined to create a new interpretation (Ribeiro et al., 2014) in such a way that a better decision could be performed (Nilsson and Ziemke, 2007). Usually, in the realm of artificial intelligence, information fusion has two main goals: to support decision-making and to improve the understanding of an application domain (Ribeiro et al., 2014). As a matter of fact, the traditional research focus within information fusion has been the "process of combining large amounts of information in a more comprehensive and easily manageable form" (Nilsson and Ziemke, 2007). In general, information fusion is used in systems to reduce some type of noise, increase accuracy, summarize information, extract useful information, support decision-making and so forth (Ribeiro et al., 2014). Thus, information fusion should comply with important critical success factors such as the ones pointed in Section 2.2. The most traditional and well-known frameworks for information fusion are based on statistical methods (e.g. Kalman filters, optimal theory, distance methods) and probabilistic techniques (e.g. Bayes theory, and Dempster-Shafer Evidence Theory (DSET)).

In summary, information fusion methods are crucial for obtaining a coherent and uniform view to support decision makers and the correspondent decision-making process. Further, we must ensure that data is comparable and consistent to ensure a suitable outcome from the information fusion (Ribeiro et al., 2014).

2.3.4 Low-Level Fusion versus High-Level Fusion

The distinction between low and high-level fusion processing can be made equivalently, in this context, between multi-sensor fusion and information fusion (Bellenger, 2013). Multi-sensor fusion typically applies to levels 0 and 1 of the JDL model (depicted in Section 2.5.1), whereas the term information fusion is often used to refer to levels 2, 3 and related parts of level 4 (Bellenger, 2013). The main conceptual differences have been summarized by (Waltz and Llinas, 1990), and graphically represented by (Blasch et al., 2010) in Figure 2.3.

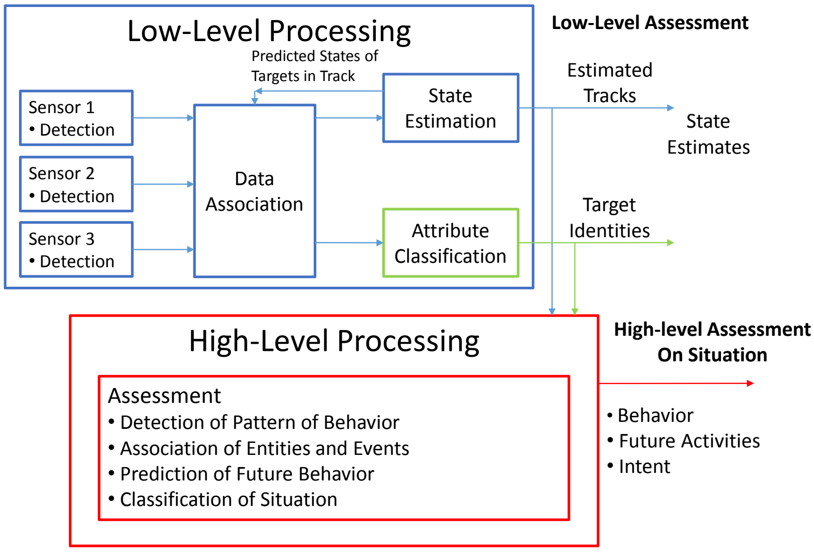


Figure 2.3: Low-level processing versus high-level processing. Adapted from (Blasch et al., 2010).

As described in Section 2.3.2, multi-sensor fusion involves objects having properties, which are usually measurable. As a matter of fact, its goal is largely to infer and predict physical, kinematic properties of physical entities (see Figure 2.1) which makes that mathematical models and algorithms used in this purpose to be closely tied to physics and constrained by the laws of physics (Bellenger, 2013). Over the last years, these numerically based low-level algorithms have been subject of several researches from the scientific community.

On the other hand, while low-level processes support target classification, identification, and tracking, high-level processes support situation, impact, and fusion process refinement, as stated by (Blasch et al., 2010, Bellenger, 2013). Therefore, high-level fusion processes involve several elements and interactions among a wide variety of situation components. According to (Blasch et al., 2010, Bellenger, 2013), high-level processes have the following properties:

- They focus on symbolic reasoning rather than numeric reasoning;
- Reasoning within the context, where data are analyzed with respect to the evolving situation;
- They manipulate both concrete and abstract entities;

- Multiple types of dynamic and static domain knowledge to be processed;
- Numerous constituency-dependency relationships among objects as well as events and activities of interest.

Comparing the two levels of data fusion, information fusion "takes the world to be a world of facts, where facts involve the application of relations between objects". In a more comprehensive manner, higher-levels rely on the realm of cognitive, social and organizational phenomena, which can be combined with the physical notions used by multi-sensor fusion (Bellenger, 2013).

2.4 Fusion System Architectures

Developing a data fusion system is a complex task. For instance, designing a multi-sensor data fusion must consider the possibility of different types of sensors working simultaneously with distinct output formats and periodicity. Such heterogeneous scenario with different levels of integration must have a well-defined communication protocol between the components of the system for proper data flow, in order to create a robust fusion architecture (Veloso et al., 2009). As stated by (Bedworth and O'Brien, 2000), an architecture includes the arrangement of the component parts, their connectivity, and the data flows between them. In most cases, multi-sensor data fusion systems presents a 3 (or even 4) layer architecture: a level to deal with physical issues; a level to fuse data; and a level to present data to final user (Veloso et al., 2009).
(Esteban et al., 2005) conducted a study which presented a synthesis of architectural issues that must be taken into account to develop a multi-sensor data fusion platform. The key issues, organized by (Veloso et al., 2009), encompasses how to process the sensors distribution to form a network (in parallel or serial bus, or even a combination of both); the level of data representation needed in which data can be enriched with different levels of fusion; the possibility of including a feedback mechanism in the system which allows the control of recommendations provided by different levels; how to deal with difficulties and unforeseen situations in data fusion such as sensor failures, corrupted data and even incompatibility with different sensors; and last but not the least, the type of architecture for data fusion to be used (Ng, 2003, Castanedo, 2013). Based on the design criteria, the following types of architectures could be identified:

- Centralized architecture;
- Decentralized architecture;
- Distributed architecture;
- Hierarchical architecture.

The aforementioned architectures are described in detail in the following subsections.

2.4.1 Centralized Fusion Architecture

In a centralized fusion architecture, all fusion processes are executed in a central processor that receives the information (raw data) from the different input sources (Ng, 2003), as shown in Figure 2.4.

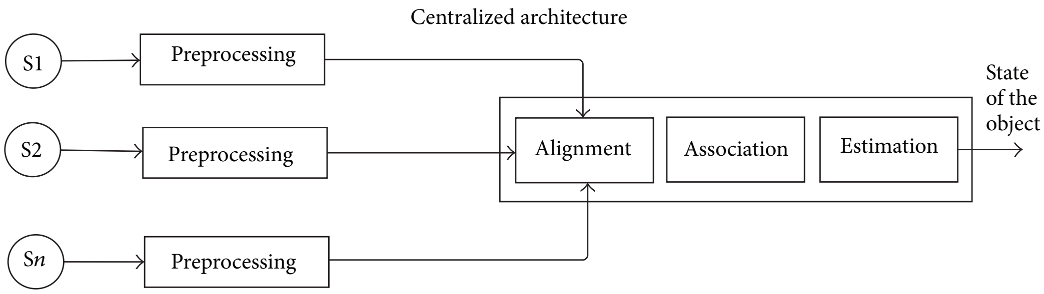


Figure 2.4: Centralized architecture. Taken from (Castanedo, 2013).

In this schema, the sources S obtain only the observations measurements, preprocess those measurements and then transmit them to the central processor where the data fusion process is performed. Issues as data alignment and data association together with time delays, for information transferring between sources and large amount of bandwidth, are some disadvantages that can compromise the results in the centralized scheme (Castanedo, 2013).

2.4.2 Decentralized Fusion Architecture

A decentralized fusion architecture is composed of a network of nodes where each node has its own processing capabilities and there is no single central point of data fusion (Ng, 2003, Castanedo, 2013). Figure 2.5 shows this particular architecture.

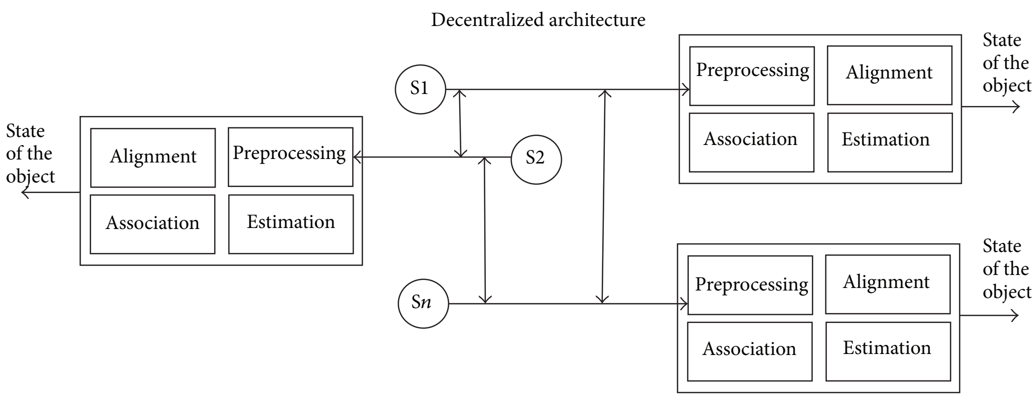


Figure 2.5: Decentralized architecture. Taken from (Castanedo, 2013).

Therefore, data fusion is performed autonomously where each node fuses its local information with the information that is received from its peers (Castanedo, 2013). The main disadvantage of this architecture consists in the communication cost which may cause scalability problems when the number of nodes is increased, with no common communication facility (Liggins et al., 2008, Ch. 14) (Ng, 2003, Castanedo, 2013). Another feature of this architecture is that nodes have no global knowledge of the network topology (Ng, 2003).

2.4.3 Distributed Fusion Architecture

In a distributed fusion architecture, measurements from each source node are processed independently before the information is sent to the fusion node (Ng, 2003). The fusion node accounts for the information that is received from other nodes, i.e. the data association and state estimation are performed in the source node before the information is communicated to the fusion node (Ng, 2003, Castanedo, 2013), as depicted in Figure 2.6.

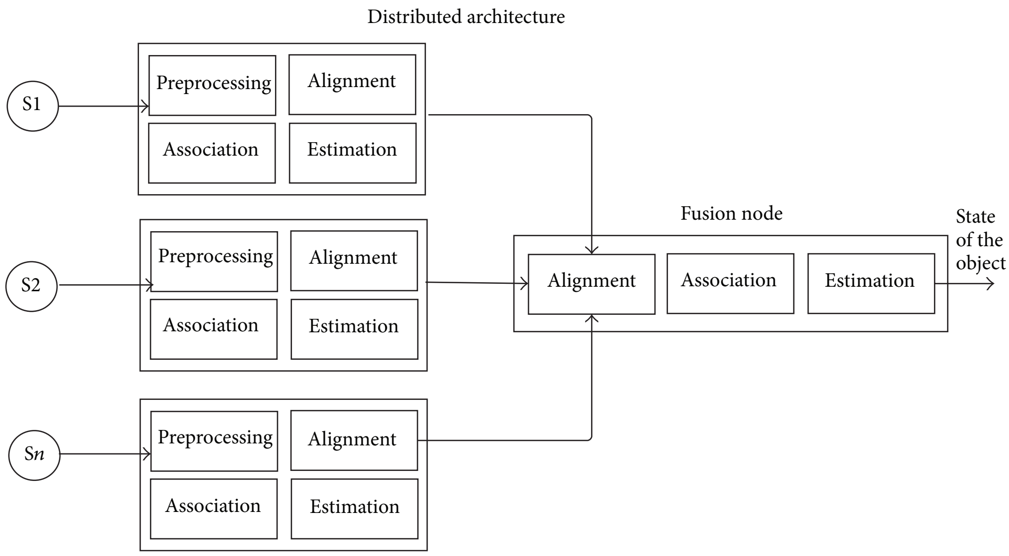


Figure 2.6: Distributed architecture. Taken from (Castanedo, 2013).

Summarizing, as in (Castanedo, 2013) "each node provides an estimation of the object state based on only their local views and this information is the input to the fusion process, which provides a fused global view." Works such as (Liggins et al., 2008, Ch. 8) and (Liggins et al., 2008, Ch. 17) analyze in greater detail the subject of distributed architectures.

2.4.4 Hierarchical Fusion Architecture

Hierarchical fusion architectures are a combination of centralized, decentralized and distributed architectures (Ng, 2003). In this type of architecture, the sensor input measurements are preprocessed at local fusion processing nodes (centralized architecture), and its track results and track estimates are sent in turn to global fusion nodes to generate a common track estimate. The benefits of a hierarchical fusion are identified in (Liggins et al., 2008, Ch. 17) as "the diversity between local nodes and different sets of sensors to view the target, maintaining local storage of assigned sensor reports and track histories at the originating node, while transmitting refined track estimates." Other beneficial characteristic of these architectures, is the use of feedback, which can be used to redistributing the reduced error uncertainty estimates from the global fusion node to the local fusion nodes as in Figure 2.7. Communication problems, such as those identified in Section 2.4.2, may be compensated with these type of architecture in terms of bandwitch for more quantity and quality of data flow (Liggins et al., 2008, Ch. 17).

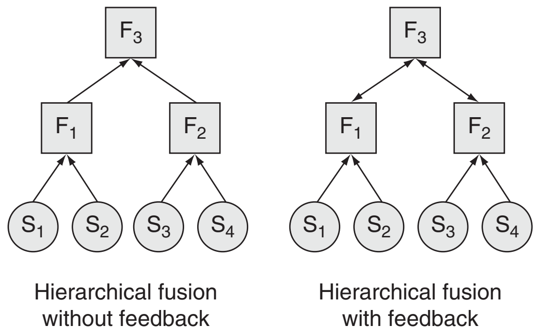


Figure 2.7: Hierachical fusion architecture without and with feedback. Adapted from (Liggins et al., 2008, Ch. 17).

In Figure 2.7, S_i with $i = 1,\ 2,\ 3,\ 4$, represent the set of sensors while F_1 and F_2 are local fusion centers responsible for preprocessing, and F_3 is the global fusion center.

2.4.5 Difference Between Decentralized and Distributed Architecture

In most of the literature, the decentralized and distributed fusions are often used interchangeably, however there is a slight difference between the two terms (Ng, 2003). (Ng, 2003) essentially sets the difference in the absence of a central facility in decentralized fusion case while in the distributed case there still exists the notion of central processing. The difference between both fusion architectures can be analyzed comparing Figure 2.5 with Figure 2.6.

In principle, a decentralized data fusion system is more difficult to implement because of the computation and communication requirements. However, in practice, there is no single best architecture, and the selection of the most appropriate architecture should be made depending on the requirements, demand, existing networks, data availability, node processing capabilities, and organization of the data fusion system (Ng, 2003).

2.5 Fusion Models

(Bedworth and O'Brien, 2000) define a process model to be a description of a set of processes, and several models have been developed to deal with data fusion issues (Veloso et al., 2009). Considering multi-sensor data fusion cases, usually there is a module to deal with the sensors and their output (input module), other to make some preprocessing or full processing of the data and a module to output the data processed on the environment. The so called adjustments/calibrations of the sensors and system itself are normally done in a closed loop interface, common in these cases (Veloso et al., 2009).

2.5.1 JDL Model

Developed in 1985 by the United States Joint Directors of Laboratories (JDL) Data Fusion Group (Steinberg et al., 1999), the JDL data fusion model would become the most common and popular conceptualization model of fusion systems. The group was intended to assist in coordinating DoD activities in data fusion, and improve communication and cooperation between development groups with the purpose of unifying research (Macii et al., 2008, Khaleghi et al., 2013, Hall and Steinberg, 2000). The result of that effort was the creation of a number of activities (Steinberg et al., 1999, White, 1991, Hall and Steinberg, 2000, Hall and Garga, 1999) such as: (1) development of a process model for data fusion (see Figure 2.8); (2) creation of a lexicon for data fusion; (3) development of engineering guidelines for building data fusion systems; (4) organization and sponsorship of the Tri-Service Data Fusion Conference from 1987 to 1992. The JDL Data Fusion Group has continued to support community efforts in data fusion, leading to the annual National Symposium on Sensor Data Fusion and the initiation of a Fusion Information Analysis Center. In the initial JDL data fusion lexicon (dated 1985), the group defined data fusion as "a process dealing with the association, correlation, 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 process is characterized by continuous refinements of its estimates and assessments, and the evaluation of the need for additional sources, or modification of the process itself, to achieve improved results." (Steinberg et al., 1999).

According to this model (see Figure 2.8), the sources of information, Source Preprocessing, used for data fusion can include both local and distributed sensors (those physically linked to other platforms), or environmental data, *a priori* data, and human guidance or inferences (Macii et al., 2008, Khaleghi et al., 2013, Hall and Steinberg, 2000). Using these sources of information, the original JDL data fusion process consists in four increasing levels of abstraction, namely, "object", "situation", "threat" and "process" refinement which were described by (Steinberg et al., 1999,

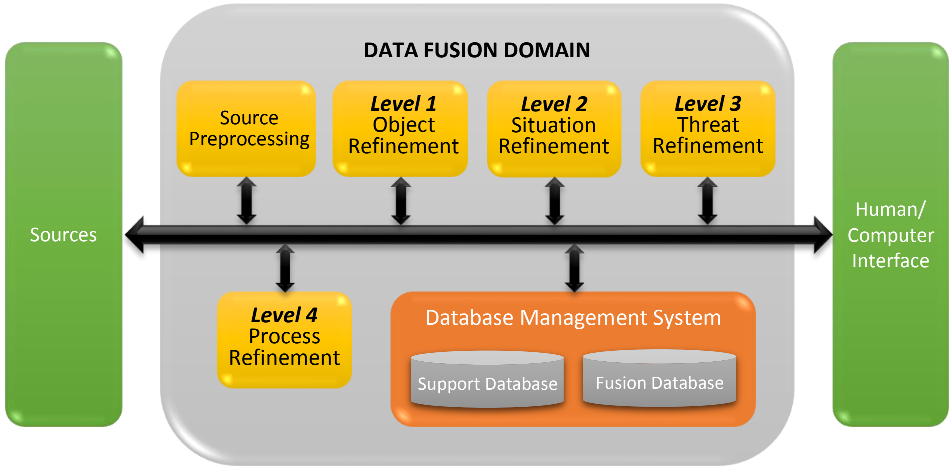


Figure 2.8: The JDL data fusion model. Adapted from (Steinberg et al., 1999).

Hall and Garga, 1999, Hall and Steinberg, 2000) and (Liggins et al., 2008, Ch. 1) as follows:

- Source Preprocessing: estimation and prediction of signal/object observable states on the basis of pixel/signal level data association and characterization;
- Level 1 processing (Object Refinement): estimation and prediction of entity states on the basis of observation-to-track association, continuous state estimation (e.g. kinematics) and discrete state estimation (e.g. object identity);
- Level 2 processing (Situation Refinement): estimation and prediction of relations among entities and events in the context of their environment (e.g. communications and perceptual influences);
- Level 3 processing (Threat Refinement): estimation and prediction of effects on situations of planned or estimated/predicted actions by the participants, to include interactions between action plans of multiple players (e.g. inferences about enemy threats, friendly and enemy vulnerabilities, and opportunities for operations);
- Level 4 processing (Process Refinement): adaptive data acquisition and processing to support mission objectives (e.g. to improve accuracy of inferences, utilization of communication and computer resources).

In order to manage the entire data fusion process through control input commands of information requests, the system also must include a Human-Computer Interaction, an interface to allow a human to interact with the fusion system, as well as a Data Management unit, a lightweight database, providing access to, and management of, dynamic data fusion data (Hall and Steinberg, 2000, Macii et al., 2008).
There are, however, some concerns with the ways in which this model levels have been used in practice. The JDL levels have frequently been interpreted "as a canonical guide for partitioning

functionality within a system" (Steinberg et al., 1999), do level 1 fusion first, then levels 2, 3 and 4. The original JDL titles for these levels appear to be focused on tactical targeting applications (e.g. threat refinement), so that the extension of these concepts to other applications is not obvious (Steinberg et al., 1999). Therefore, despite its popularity, the JDL model has many shortcomings, such as being too restrictive and especially tuned to military applications, which have been the subject of several extension proposals (Hall and Llinas, 1997, Steinberg et al., 1999, Llinas et al., 2004, Steinberg and Bowman, 2004) attempting to alleviate them. Resuming, the JDL formalization is focused on data (input/output) rather than processing (Khaleghi et al., 2013).

2.5.2 Waterfall Model

As mentioned in (Veloso et al., 2009), the Waterfall model is a hierarchical architecture where the information outputted by one module will be inputted to the next module, as depicted on Figure 2.9. The model emphasizes on the processing functions on the lower levels.

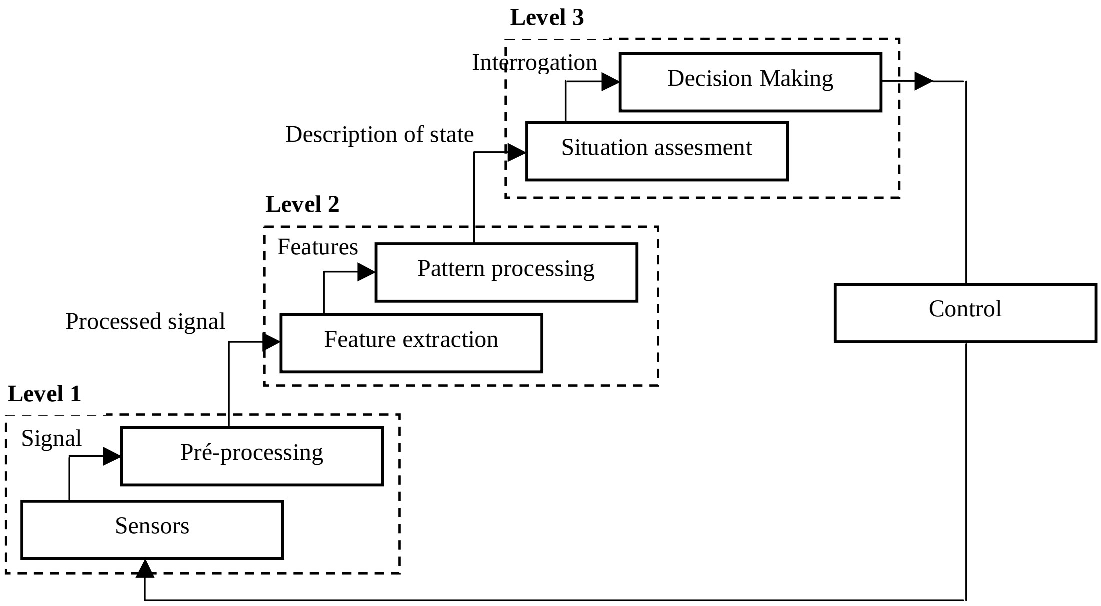


Figure 2.9: The Waterfall data fusion process model. Taken from (Veloso et al., 2009).

The stages relate to the source preprocessing and the levels 1, 2 and 3 of the JDL model as follows: sensing and signal processing correspond to source preprocessing, feature extraction and pattern processing match object refinement (level 1), situation assessment is similar to situation refinement (level 2), and decision making corresponds to threat refinement (level 3) (Veloso et al., 2009). Being this similar to the JDL model, the Waterfall model suffers from the same drawbacks. While being more exact in analyzing the fusion process than other models, the major limitation is the lack of description of the feedback data flow (Bedworth and O'Brien, 2000). However, in (Esteban et al., 2005, Zegras et al., 2008) is proposed that the sensor system is continuously updated with feedback information arriving from the decision-making module, as depicted in Figure 2.9. The main aspects of the feedback element are the re-calibration, re-configuration and data gathering aspects alerts to the multi-sensor system (Esteban et al., 2005).

The Waterfall model does not clearly state that the sources should be parallel or serial (though processing is serial), assumes centralized control, and allows for several levels of representation (Zegras et al., 2008). This model has been widely used in the UK defense data fusion community but has not been significantly adopted elsewhere (Bedworth and O'Brien, 2000).

2.5.3 Luo and Kay Model

(Luo and Kay, 1988) presented a generic data fusion structure based in a hierarchical model, yet different from the Waterfall model. In this system, data from the sensors are incrementally added on different fusion centers in a hierarchical manner, thus increasing the level of representation from the raw data or signal level to more abstract symbolic representations at the symbol level (Esteban et al., 2005, Zegras et al., 2008, Veloso et al., 2009). This model has a parallel input and processing of data sources interface, which may enter the system at different stages and levels of representation, as depicted in Figure 2.10. The model is based on decentralized architecture (see Section 2.4.2) and does not assume a feedback control (Veloso et al., 2009).

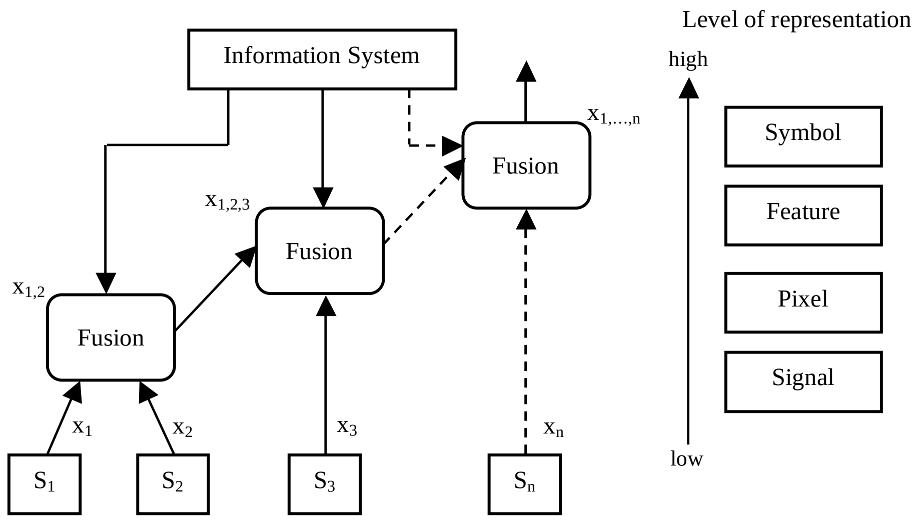


Figure 2.10: Luo and Kay data fusion process model. Taken from (Veloso et al., 2009).

In Figure 2.10, as the information is combined gradually at the different fusion centers, the level of representation increases from signal level (responsible for the raw data) to more abstract symbolic representations of the data at the symbol level (Esteban et al., 2005, Veloso et al., 2009).

Table 2.1 depicts in detail a comparison of the different fusion levels classified by representation, type and model of information (Esteban et al., 2005).

Table 2.1: Characteristics of Luo and Kay Model data fusion levels. Adapted from (Esteban et al., 2005).

Characteristics	Representation level of information	Type of sensory information	Model of sensory information
Signal level	Low	Multi-dimensional signal	Random variable with noise
Pixel level	Low	Multiple images	Random process across the pixel
Feature level	Medium	Features extracted from signals/images	Non-invariant form of features
Symbol level	High	Decision logic from signals/images	Symbol with degree of uncertainty

2.5.4 Thomopoulos Model

In (Esteban et al., 2005, Veloso et al., 2009) works is mentioned that Thomopoulos proposed a three level model, formed by signal level fusion (where data correlation takes place); evidence level fusion (where data is combined at different levels of inference); and dynamics level fusion (where the fusion of data is done with the aid of an existing mathematical model). Depending upon the application, these levels of fusion can be implemented in a sequential manner or interchangeably (Esteban et al., 2005). Figure 2.11 depicts the Thomopoulos Model.

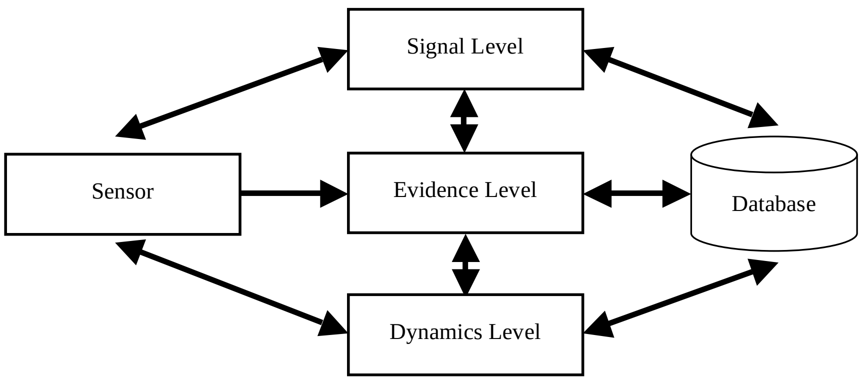


Figure 2.11: Thomopoulos data fusion process model. Taken from (Veloso et al., 2009).

At each level exists data integration preserving a given order which may cause communication problems with data transmission, and factors such as spatial/temporal co-alignment of the data should also be considered (Esteban et al., 2005, Veloso et al., 2009). In Figure 2.11 it is clear a database component, responsible for data gathering which will be used in a learning process (Veloso et al., 2009).

2.5.5 Intelligence Cycle

The intelligence cycle is an approach which falls in a fusion model subgroup with cyclic character (Elmenreich, 2007), being the Boyd control loop (see Section 2.5.6) an integral part.

Intelligence processing involves both information processing and information fusion as observed in (Bedworth and O'Brien, 2000). Although the information is often at a high level (as approached in Section 2.3.4), the processes for handling intelligence products are broadly applicable to data fusion in general (Bedworth and O'Brien, 2000).

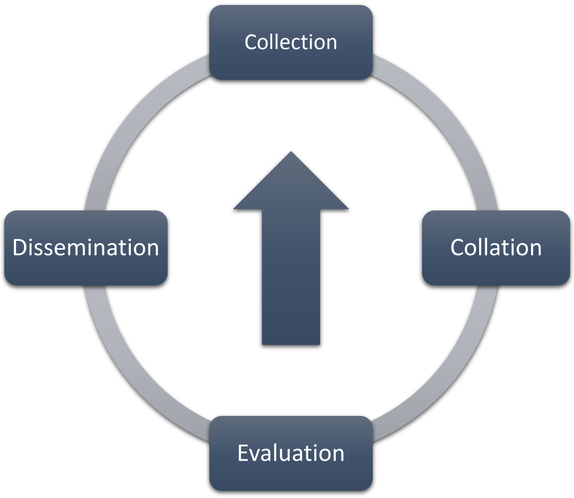


Figure 2.12: The UK intelligence cycle. Adapted from (Bedworth and O'Brien, 2000).

The cycle is depicted in Figure 2.12, and according to (Elmenreich, 2007, Bedworth and O'Brien, 2000) comprises four stages:

- Collection: gathering of appropriate raw intelligence data (intelligence report at a high level of abstraction), e.g. through sensors;
- Collation: associated intelligent reports are correlated and brought together;
- Evaluation: the collated intelligence reports are fused and analyzed;
- Dissemination: the fused intelligence is distributed to the users.

An additional stage is enumerated by (Elmenreich, 2007) regarding to planning and directions, where are determined the intelligence requirements. However, in the particular case of the United Kingdom Intelligence Community model, this stage is subsumed in the dissemination process, unlike the US model (Bedworth and O'Brien, 2000).

2.5.6 Boyd Control Loop Model

In 1987, John Boyd proposed a cycle containing four stages that was first used for modeling the military command process (Bedworth and O'Brien, 2000, Elmenreich, 2007). The Boyd control loop or OODA loop (depicted in Figure 2.13) represents the classic decision-support mechanism in military information operations (Elmenreich, 2007). Since decision-support systems for situational awareness are tightly couple with fusion systems, the Boyd model has also been used for sensor fusion (Bedworth and O'Brien, 2000).

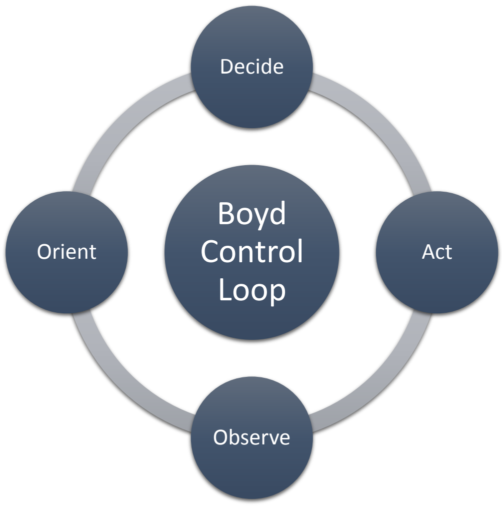


Figure 2.13: The Boyd (or OODA) control loop data fusion process model. Adapted from (Bedworth and O'Brien, 2000).

The Boyd model have some similarities between the Intelligence Cycle and JDL models, which (Bedworth and O'Brien, 2000, Elmenreich, 2007) compared the stages of those models:

- Observe: broadly comparable to the JDL source preprocessing level and part of the collection stage of the intelligence cycle;
- Orient: corresponds to the levels 1, 2 and 3 of the JDL model and includes the structured elements of collection and the collation stages of the intelligence cycle;
- Decide: comparable to level 4 of the JDL model (Process Refinement) and to the dissemination stage activities of the intelligence cycle;
- Act: has no similar in the JDL model and is the only model that explicitly closes the loop by taking account of the effect of decisions in the real world.

Although the Boyd model represents the stages of a closed control loop system and gives an overview on the overall task of a system, the model structure is significantly limited in identifying and separating different sensor fusion tasks (Elmenreich, 2007).

2.5.7 Dasarathy Model

The Dasarathy fusion model is categorized in terms of the types of data/information that are processed and the types that result from the process (Liggins et al., 2008, Ch. 3). Its data flow is characterized by Input/Output (I/O) and processes (Khaleghi et al., 2013), and three main levels of abstraction during the data fusion process are identified (Bedworth and O'Brien, 2000) as:

- Decisions: symbols or belief values;
- Features: or intermediate-level information;
- Data: or more specifically sensor data.

Dasarathy states that fusion may occur both within these levels and as a means of transforming between them (Bedworth and O'Brien, 2000). In the Dasarathy fusion model there are five possible categories of fusion, illustrated in Table 2.2 as types of I/O considered (Liggins et al., 2008, Ch. 3).

Table 2.2: The five levels of fusion in the Dasarathy model. Taken from (Bedworth and O'Brien, 2000).

Input	Output	Notation	Analogues
Data	Data	DAI-DAO	Data-level fusion
Data	Features	DAI-FEO	Feature selection and feature extraction
Features	Features	FEI-FEO	Feature-level fusion
Features	Decisions	FEI-DEO	Pattern recognition and pattern processing
Decisions	Decisions	DEI-DEO	Decision-level fusion

The processes are described by Dasarathy using the notations *DAI-DAO*, *DAI-FEO*, *FEI-FEO*, *FEI-DEO*, and *DEI-DEO* (Bedworth and O'Brien, 2000)(Liggins et al., 2008, Ch. 3).

2.5.8 Omnibus Model

In 1999 (Bedworth and O'Brien, 2000) proposed a framework called Omnibus model, created after analyzing the strengths and weakness of existing models and integrates most of the beneficial features of other approaches. This unified model, shown in Figure 2.14, is a hybrid model based around the cyclic nature of the intelligence cycle and the Boyd control loop but uses the finer definitions of the Waterfall model, each of which can be associated with one of the levels in the JDL and Dasarathy models (Elmenreich, 2007). In the Omnibus model, feedback data flow is explicit and the concept of loops within loops is acknowledged, previously neglected (Bedworth and O'Brien, 2000). The model has a cyclic structure for data fusion, comparable to Boyd control loop, but with a more refined structuring of the processing levels (Bedworth and O'Brien, 2000, Elmenreich, 2007). In addition, the recognized fidelity of representation expressed by the Waterfall model is then easily incorporated into each of the four main process tasks of the Omnibus model (Bedworth and O'Brien, 2000).

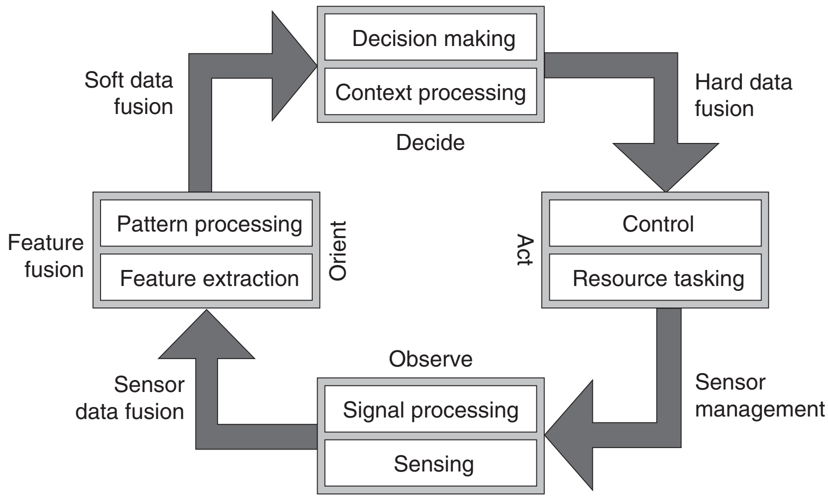


Figure 2.14: The Omnibus model - a unified data fusion process model. Taken from (Liggins et al., 2008, Ch. 22).

The fusion processes are explicitly located, and has the advantage to be used multiple times in the same application recursively at two different levels of abstraction (Bedworth and O'Brien, 2000). Firstly, the model is used to characterize and structure the overall system to provide an ordered list of tasks. Secondly, the same previously subdivided structures are used to organize the functional objectives of each subtasks of the system (Bedworth and O'Brien, 2000, Elmenreich, 2007). (Bedworth and O'Brien, 2000) adds that "using this approach, a data fusion solution is categorized using a dual perspective - both by its system aim and its task objective."
Although the hierarchical separation of the sensor fusion tasks it is a great advantage of the Omnibus model, it does not support a horizontal partitioning into modules that can be separately implemented, with distributed sensing and data processing (Elmenreich, 2007).

2.5.9 Distributed Blackboard Model

(Schoess and Castore, 1988) describe an example of a distributed blackboard data fusion model based on the confidence of the values produced by each sensor (sensors that measure the same phenomenon). An example of this model is shown in Figure 2.15 where two sensors (S_1 and S_2) are connected to a number of transducers (T) (Esteban et al., 2005).
These sensors are monitored by a supervisor which controls how conflicting sensor measurements are handled, based upon confidence levels assigned to each sensors supported by the transducers (Esteban et al., 2005). Essentially, the transducers act as a mechanism of validation to the measures performed by the sensors. Simultaneously with each sensor, the set of transduc-

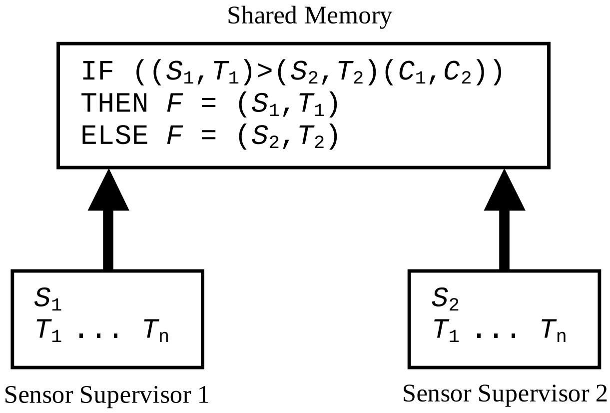


Figure 2.15: Distributed blackboard data fusion process model. Taken from (Veloso et al., 2009).

ers are used to acquire as much information as possible from the physical system under analysis (e.g. temperature, pressure, etc.) (Esteban et al., 2005, Veloso et al., 2009). The fusion algorithm produces a final value, F, which depends of the data gathered by the two sensors (Esteban et al., 2005).

2.6 Data Fusion Algorithms and Methods

An algorithm can be defined as an effective method for solving a problem with a finite sequence of steps and must be precise, unambiguous and give the right solution in all cases. It is used for calculation, data processing, and in a wide range of applications in many fields (see Section 2.8). The necessity of developing efficient algorithms has become a key technological challenge for all kinds of ambitious and innovative data fusion applications, regardless of the fusion architecture (Sidek and Quadri, 2012).

The load on CPU, memory allocation and storage, interprocess communication and finally computation time, are all dependent on algorithmic complexity. Therefore, an algorithm should use simpler communication interfaces and abstraction to enhance the throughput and reduce the delay, and should allow real-time operation without complex global state maintenance (Sidek and Quadri, 2012).

Besides all these performance related characteristics, a data fusion algorithm must cope with the inherent "anomalies" of input data, its processing, and fuse data to produce coherent and precise outputs.

2.6.1 Fusion Algorithm Taxonomy

(Khaleghi et al., 2013) suggests a taxonomy of data fusion methodologies in which different data fusion algorithms can be roughly categorized based on one of four challenging problems of input data: data imperfection, data correlation, data inconsistency, and disparateness of data form.

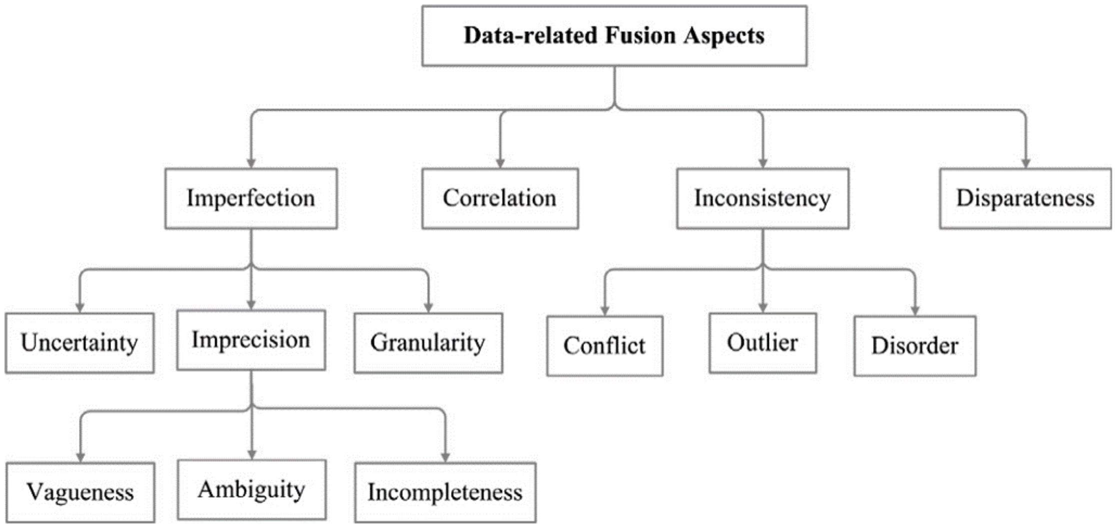


Figure 2.16: Taxonomy of data fusion methodologies. Taken from (Khaleghi et al., 2013).

As discussed in Section 2.2, real-world fusion applications have to deal with several data related challenges and the taxonomy proposed by (Khaleghi et al., 2013) (see Figure 2.16) illustrates an overview of data-related challenges that are typically tackled by data fusion algorithms. The input data to the fusion system may be imperfect, correlated, inconsistent, and/or in disparate forms/modalities and each of these four main categories of challenging problems can be further subcategorized into more specific problems. However, it is important to emphasize that no single data fusion algorithm is capable of addressing all the aforementioned challenges (Khaleghi et al., 2009).

2.6.1.1 Fusion of Imperfect Data

The classification of data as imperfect proposed by (Khaleghi et al., 2013) was essentially based in the works of Smets (Smets, 1997), and Dubois and Prade (Dubois and Prade, 2010). Smets simply stated that "imperfection can be due to imprecision, inconsistency and uncertainty, the major aspects of imperfect data" (Smets, 1997). Several research work and theories have been developed to deal with the inherent imperfection of data, being this the most fundamental challenging problem of data fusion systems (Florea et al., 2002, Khaleghi et al., 2013). There are several mathematical theories available to represent data imperfection, such as probability Bayesian theory, which is known from centuries for dealing with uncertainty (Florea et al., 2002); the theory of fuzzy sets handling vague data; the theory of possibility addressing simultaneously imprecision and uncertainty; rough sets theory which deals with imprecision when uncertainty is involved but not quantified; and DSET able to deal with information containing imprecision and uncertainty

(Khaleghi et al., 2013). These approaches are mostly applied to only one type of imperfection. (Goodman et al., 1997) work proposes potential approaches to cope with all types of imperfection (Florea et al., 2002). From these researches, (Khaleghi et al., 2013) defined three aspects of data imperfection: uncertainty, imprecision, and granularity.
Uncertainty is partial knowledge of the true value of the data, and represents our state of knowledge about a piece of information. Data is considered uncertain when the associated confidence degree of the data is less than 1. On the other hand, imprecision is a property of the information itself, as a form of incompleteness, being imprecise data a reference to several, rather than only one, object(s) (Florea et al., 2002). Finally, the ability to distinguish among objects described by data, being dependent on the provided set of attributes, is defined as data granularity, a term approached by Dubois and Prade (Dubois and Prade, 2010). For instance, if a new proposition is added, it may result in a modification.

2.6.1.2 Fusion of Correlated Data

In order to produce consistent results, a data fusion algorithm often require either independence or prior knowledge of the cross covariance of data. However, due to noise in a phenomena measurements (in centralized architectures) or a problem known as *data incest* (in distributed architectures) arising from the inadvertent (or ignored) multiple use of identical information may potentially cause unknown cross covariances (McLaughlin et al., 2004). A way of preventing data correlations by *data incest* is by avoiding that the same information takes several different paths from the source node to the fusion node (Khaleghi et al., 2013). Instead of removing data correlation, one can use a fusion method named Covariance Intersection developed to avoid the problem of covariance matrix underestimation due to *data incest*. It solves this problem in general form for two data sources (i.e. random variables) by formulating an estimate of the covariance matrix as a convex combination of the means and covariances of the input data. Nevertheless, Covariance Intersection is very demanding computationally and situations of biased estimations, such as high-confidence in a particular value or even divergence, may lead to misleading results and degradation of fusion performance (Khaleghi et al., 2013).

2.6.1.3 Fusion of Inconsistent Data

Data inconsistency is a generic definition that comprehends subjects as spurious[3], disordered and conflicting data. Spurious data can be caused by unexpected failures in sensors and may be potentially harmful if fused with correct input data, leading to inaccurate estimates, which may compromise the performance of the, for instance, Kalman filter. The majority of research on

[3] Not being what it purports to be; false or fake (Dictionaries, 2014d)

treating spurious data has been focused on identification/prediction and subsequent elimination of outliers from the fusion process (Khaleghi et al., 2013).

Regarding disordered data, it is fundamental that the input data of a fusion system to be organized and sequential. The disorder itself occurs if different data sources have varying preprocessing times, operating at multiple rates with different delays in data transmission. Therefore, data arrives out-of-sequence at the fusion system appearing as inconsistent measurements (Khaleghi et al., 2013). These Out-Of-Sequence Measurements (OOSM) pose a problem in usual filtering algorithms, such as the Kalman filter, because the original order of the measurements is usually needed (Westenberger et al., 2012). Due to the increasing popularity of distributed sensing and tracking systems, this subject has been under the focus of several research since it is an inherent problem of sensory systems (Khaleghi et al., 2013).

Another point that encompasses data inconsistency is conflicting data. An observation of a phenomenon may cause different comprehensions in different experts. The challenging task to unify different ideas of a phenomenon has been thoroughly studied either by within the DSET framework, and by other approaches that refute the use of such framework, such as Zadeh in (Zadeh, 1984) (see (Khaleghi et al., 2013) for more details).

Disparate[4] data must also be taken into account since input data of a fusion system may come from a variety of sensors, human, or even archived sensory data (Khaleghi et al., 2013). The different sources of inputs are, as the definition of disparate says, different in kind and therefore propose a major challenge to be compared in order to create a coherent and accurate global view of the observed phenomena. The work of (Hall et al., 2008) focus in the inclusion of a human in the fusion process as a soft data source while electronic sensors would act as hard data sources. However, as stated in (Khaleghi et al., 2013), this topic and its derivatives are still at an almost embryonic stage.

2.6.2 Fuzzy Reasoning

Fuzzy set theory (Zadeh, 1965) is a theoretical reasoning scheme that deals with imperfect data (Babuska, 2003, Khaleghi et al., 2013). It introduces the notion of membership function, which enables imprecise reasoning (Zadeh, 1965, Babuska, 2003, Khaleghi et al., 2013). Fuzzy logic mathematically describes objects or processes classified within classes (e.g. the class of tall man, the class of beautiful women) into membership values ranging (inclusively) between 0 and 1. This normalization process allows efficient fuzzy data fusion when incomplete or vague data is fuzzified (Ribeiro et al., 2014). The principal ability of fuzzy logic is the formalization of humans' capability to reason and support decision making in an environment of uncertainty, imprecision, and incompleteness of information (Zadeh, 1965, Babuska, 2003). Further, in rule-based fuzzy

[4]Essentially different in kind; not able to be compared Dictionaries (2014a).

systems the relationships between variables are represented by If-Then decision rules of the following general form (Babuska, 2003):

If *antecedent proposition* **then** *consequent proposition.*

Depending on the form of the consequent proposition, two main types of rule-based fuzzy models are distinguished (Babuska, 2003):

- Max-Min fuzzy model (Mamdani): both the antecedent and the consequent are fuzzy propositions, i.e. fuzzy inputs and fuzzy outputs that need to be defuzzified;
- Takagi-Sugeno fuzzy model: the antecedent is a fuzzy proposition, the consequent is a crisp function, i.e. fuzzy inputs and crisp output.

The operators for rules' reasoning are usually divided into conjunctive (fuzzy t-norm - AND), and disjunctive (t-conorm - OR) operators (Babuska, 2003, Khaleghi et al., 2013). Conjunctive operators are appropriate when the sources are equally reliable and homogeneous, therefore they need to be all aggregated with the logical AND operator. On the other hand, disjunctive operators are used when (at least) one of the sources is deemed reliable, though which one is not known, or when fusing highly conflictual data (Khaleghi et al., 2013). Besides these operators it is also considered the complement (NOT) operator (Dailey and Lin, 1996, Babuska, 2003). In general, fuzzy reasoning operators have the following logic:

- $OR : A \cup B = MAX(A, B)$;
- $AND : A \cap B = MIN(A, B)$;
- $NOT : \neg X = 1 - X$.

Other operators such as Uninorm aggregation operator, a generalization of t-norm and t-conorm which takes into consideration a neutral element; and Fimica aggregation operator, with full reinforcement behavior similar to Uninorm but with a function to control the operator behavior (Ribeiro et al., 2010), are used later in Section 4.
Fuzzy reasoning systems are composed of four main stages (Ross, 2000, Ch. 4)(Babuska, 2003):

- Fuzzification: mapping from numerical values to the membership functions of the fuzzy variables;
- Rules evaluation: the rules defined are evaluated using the fuzzy logic;
- Aggregation: the results of the rules are aggregated so that they are mapped to the output variables;
- Defuzzification: the final step is the mapping from fuzzy output variables to numerical values.

Unlike probability and evidence theories, fuzzy sets theory is well suited to modeling the fuzzy membership of a target in an ill-defined class, only requiring prior membership functions for different fuzzy sets (Khaleghi et al., 2013).
In the context of this thesis we will focus on the applied fuzzy reasoning model, which is the Takagi-Sugeno method, explained next.

2.6.2.1 Takagi-Sugeno Fuzzy Model

Takagi-Sugeno method is similar to Mamdani method in many aspects (MathWorks, 2014). The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between the two referred fuzzy models is that unlike the Max-Min model, the Takagi-Sugeno model consequent is either crisp or constant. Therefore the rules have the general following form (Babuska, 2003):

$$R_i : \textbf{If } x \; is \; A_i \textbf{ then } y_i = f_i(x), \qquad i = 1, 2, ..., K \tag{2.1}$$

where x is a vector of crisp variables (e.g. hazard map feature values), A_i are the antecedent linguistic terms defined over the domain of x, and y_i is the consequent crisp output variable (e.g. a terrain safety value). In our case study the functions $f_i(x)$ are modeled as constant scalars, being this model know as a zero-order Takagi-Sugeno model, or the singleton model (Babuska, 2003). The scalar $f_i(x)$ is a value between 0 and 1, and corresponds to the output membership function parameterization, which will be weighted with the *degree of fulfillment* of x with respect to A_i. The *degree of fulfillment* β_i represents the *truth value* of the proposition "$x \; is \; A_i$". The β_i is a firing strength of the activated rule and its value is determined by a fuzzy set and an input value x by means of a fuzzy logic operator, which can be conjunction, disjunction and negation (complement). The case study uses the logical min and the max operators, respectively, for the fuzzy AND and OR operators, and a rule defined by equation 2.1, β_i such as

$$\beta_i = \mu_i(x) \tag{2.2}$$

where $\mu_i(x)$ is the membership function associated with the fuzzy set A_i. The membership function is defined as

$$\mu_i(x) : X \; \rightarrow \; [0, 1], \qquad with \; x \; \epsilon \; X \tag{2.3}$$

and covers the domain X where the proposition "$x \; is \; A_i$" holds (Babuska, 2003). The inference mechanism (aggregation step) for Takagi-Sugeno fuzzy model, i.e. the final output of the system, is the weighted average of all rule outputs defined by

$$y = \frac{\sum_{i=1}^{K} f_i(x)\beta_i}{\sum_{i=1}^{K} \beta_i} \tag{2.4}$$

where β_i, from equation 2.2, is the *degree of fulfillment* associated with each rule R_i, defined in equation 2.1 (Babuska, 2003).

2.6.3 Fuzzy Information Fusion

The Fuzzy Information Fusion (FIF) algorithm proposed by (Ribeiro et al., 2014) uses fuzzy reasoning principles aiming to transform measures in such a way that it is possible to combine the data after being transformed by a representation, according to the allowed rules for the chosen framework. This algorithm incorporates the critical success factors for information fusion listed by (Lee et al., 2010) such as robustness; extended spatial-temporal coverage; high confidence; low ambiguity; reliability and validity; and low vulnerability.

The FIF algorithm works over matrices of numbers (e.g. images) representing terrain information such as slope and texture. These matrices, known as hazard maps, are normalized as fuzzy functions to become inputs for a fuzzy multi-criteria model (Ribeiro et al., 2014).

In Figure 2.17 it is depicted the complete architecture of the FIF algorithm. Two hazard maps (slope and texture), as aforementioned, are used to illustrate the steps of the FIF algorithm. Besides these hazard maps, any other data sources that could be normalized with membership functions could be used (Ribeiro et al., 2014).

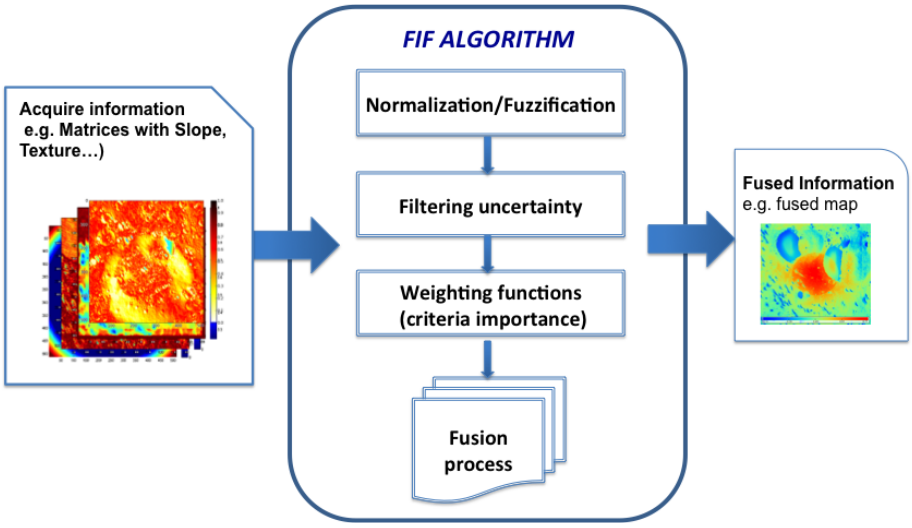


Figure 2.17: FIF algorithm architecture. Taken from (Ribeiro et al., 2014).

In Figure 2.17 are identified the four main steps to achieve a successful fusion of information (Ribeiro et al., 2014). These steps are the following:

- Normalization process, which includes a mathematical transformation (fuzzification) of maps to ensure numerical and comparable data for fusion;

- Filtering uncertainty from data regarding inaccuracies and lack of confidence in input data (e.g. hazard maps matrices have embedded imprecisions);
- Assigning relative importance to each criteria membership value, which depends on the satisfaction/suitability of criteria for a specific alternative;
- An aggregation/fusion method (i.e. aggregation operator) for combining all matrices (criteria) into a single composite (e.g. fused maps for an iteration).

The steps of the FIF algorithm were applied in the case study on this work (see Section 4 for further detail), and the results were compared with the results of the implementation using the Takagi-Sugeno model.

2.7 Supporting Technologies to Data Fusion

There are a number of methods that although not considered data fusion methods, may be used as supporting technologies in data processing. Then a list of the most popular methods is presented.

2.7.1 Probabilistic and Statistical Methods

2.7.1.1 Bayes' Theorem

The Bayesian theorem, from probability theory, was for a long time the only existing theory, used to deal with almost all kinds of imperfect data (Florea et al., 2002). As other probabilistic methods, it relies on probability distribution/density functions to express data uncertainty (Khaleghi et al., 2013). Bayes' theorem enables fusion of prior and observation data, and relates the conditional and marginal probability distributions of random variables (Durrant-Whyte and Henderson, 2008, Ch. 25). The Bayes' theorem provides a relationship (conditional and marginal probabilities) between the probability of an event A conditional on another event B, provided the probability of B conditional on A is defined (Florea et al., 2002)(Durrant-Whyte and Henderson, 2008, Ch. 25) as

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.5}$$

where

- $P(A)$ is the prior or marginal probability of A (is "prior" since it does not take into account the event B);
- $P(B)$ is the prior or marginal probability of B, and acts as a normalizing constant;

- $P(A|B)$ is the conditional probability of A given B;
- $P(B|A)$ is the conditional probability of B given A.

(Florea et al., 2002) notes that this theory was firstly "developed to deal with random experiments involving combinatory logic theory or involving the frequentist limits", and still is very effective for solving uncertainty problems by consolidating and interpreting overlapping data provided by several sensors. Its advantage comes from modeling the unknown system state by using probabilistic functions to determine an appropriate set of actions (Dailey and Lin, 1996).

2.7.1.2 Kalman Filter

In 1960, Rudolf Kalman (Kalman, 1960) proposed a recursive solution to the discrete-data linear filtering problem (Welch and Bishop, 1995), now known as Kalman Filter (KF). In other words, KF is defined by (Maybeck, 1979) as an "optimal recursive data processing algorithm" which "processes all available measurements, regardless of their precision, to estimate the current value of the variables of interest".

The KF is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error (Welch and Bishop, 1995). The filter supports estimations of past, present, and even future states using statistical description of the system noises, measurement errors, and uncertainty in the dynamics models; and it can do so even when the precise nature of the modeled system is unknown (Maybeck, 1979, Welch and Bishop, 1995).

Kalman filtering, which is recognized as an exceptional case of the Bayes filter, focuses in an exact analytical solution due to simplification of the constraints on the system dynamics (Khaleghi et al., 2013). Nonetheless, the KF is one of the most popular fusion methods mainly due to its simplicity, ease of implementation, and optimality in a mean-squared error sense (Maybeck, 1979, Khaleghi et al., 2013). However, similar to other least-square estimators, the KF is very sensitive to data corrupted with outliers and therefore inappropriate for applications whose error characteristics are not readily parameterized (Khaleghi et al., 2013).

Like the Bayesian method, the KF algorithm can demand complex computations. Figure 2.18 shows the recursive calculations involved in a KF operation (Dailey and Lin, 1996).

Kalman filtering has been widely used for a variety of optimization tasks as target location, producing fused data that estimate the smoothed values of position, velocity, and acceleration at a series of points in a trajectory. Although no set of sensors can pinpoint a target with complete accuracy, the tolerance of each sensor's positional fix accuracy can be known and assigned (Dailey and Lin, 1996).

In cases of non-linear system dynamics the KF can be used as one of its variants as the Extended

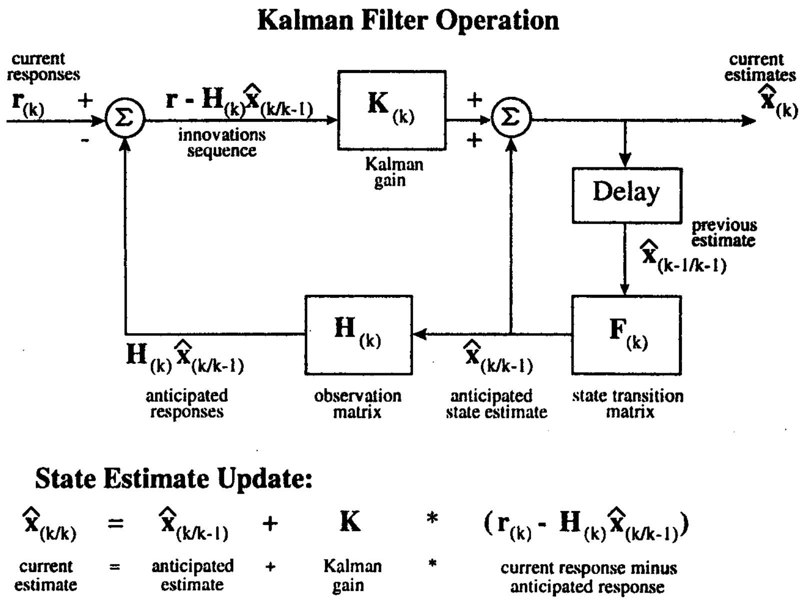


Figure 2.18: Recursive calculations in a KF. Taken from (Dailey and Lin, 1996).

KF and Unscented KF, based on the first-order and second-order approximations as a Taylor series expansion about the current estimate, respectively. However, both of these methods can only handle non-linearities to a limited extent (Khaleghi et al., 2013).

2.7.1.3 Monte Carlo Simulation

The Monte Carlo method provides a numerical solution to a problem that can be described as a temporal evolution of objects interacting with other objects based upon object-object interaction relationships. The rules of interaction in the simulation are processed randomly and repeatedly, until numerical results converge to estimated means, moments and their variances (Seco and Verhaegen, 2013, Ch. 1). Any Monte Carlo method is a random number generator that produces an infinite stream of random variables that are independent and identically distributed according to some probability distribution (Kroese et al., 2013, Ch. 1). This method represents an attempt to model nature through direct simulation of the essential dynamics of the system in question (Seco and Verhaegen, 2013).

Other popular approximating probabilities methods are the Monte Carlo simulation-based techniques such as Sequential Monte Carlo (Kroese et al., 2013, Ch. 14) and Markov Chain Monte Carlo (Kroese et al., 2013, Ch. 6). These techniques are also very flexible as they do not make any assumptions regarding the probability densities to be approximated (Khaleghi et al., 2013).

2.7.2 Dempster-Shafer Evidential Theory

The theory of belief functions initiated from Dempster's work (Dempster, 2008) and then mathematically formalized by Shafer (Shafer, 1976), toward a general theory of reasoning based on evidence, was developed as an alternative to the Bayesian theory that deals with probability mass functions (Khaleghi et al., 2013). As Shafer (Shafer, 1976) defined, "the Dempster-Shafer Evidence Theory, also known as the theory of belief functions, is a generalization of the Bayesian theory of subjective probability. Whereas the Bayesian theory requires probabilities for each question of interest, belief functions allow us to base degrees of belief for one question on probabilities for a related question. These degrees of belief may or may not have the mathematical properties of probabilities; how much they differ from probabilities will depend on how closely the two questions are related." The Bayes and Dempster-Shafer approaches are both based on the concept of weight attribution to the states of the system under measurement (Koks and Challa, 2003). The DSET approach benefits from the ability to handle uncertainty in sensor data (different levels of abstraction) allowing alternative scenarios for the system, treating the sets of alternatives that have a nonzero intersection as a new state corresponding to "unknown", thus being able to distinguish individual classes of entities (Dailey and Lin, 1996, Koks and Challa, 2003, Khaleghi et al., 2013). Nevertheless, the weighting process in DSET not considers *masses* instead of probabilities, and the most important concept of DSET is Dempster's rule of combination that allows the combination of two independent sets of mass assignments. In summary, DSET is based in two concepts (Koks and Challa, 2003):

- Obtain degrees of belief for one question from subjective probabilities for a related question;
- Use Dempster's rule for combining these degrees of belief when they are based on independent items of evidence.

Basically, DSET introduces the notion of assigning beliefs and plausibilities to measure hypotheses along with the required combination rule to fuse them (Khaleghi et al., 2013). At last, (Khaleghi et al., 2013) refers that in order to select between the Bayesian and Dempster-Shafer inference, one has to maintain a trade-off between the higher level of accuracy offered by the former and the more flexible formulation of the latter.

2.7.3 Neural Networks

The concept of artificial neural networks can be explained as a web-like artificial procedure of information processing that emulates the human brain's learning and decision-making processes. Similar to Bayesian and DSET techniques, neural networks "produce interpretive findings that incorporate input from various weighted information sources.", as described in (Dailey and Lin, 1996), and its main advantage over those two techniques is that neural networks can perform data fusion from several simultaneously data streams without *a priori* information. Artificial neural networks consists of several elements called neurons (processing nodes) to collect and correlate information (Dailey and Lin, 1996). A primitive function is located in the nodes and the composition rules are contained implicitly in the interconnection pattern of the nodes, that transmits information synchronously or asynchronously, and in presence or absence of cycle (Rojas, 1996). In other words, the neurons are connected by synapses that assign a weight to each neuron's output (Dailey and Lin, 1996). Figure 2.19 shows the structure of a neuron with n inputs. In the depicted example, each input channel i can transmit a real value x_i and is multiplied by the corresponding associated weight w_i. The transmitted information is integrated at the neuron and the primitive function f, processed in the node, is then evaluated (Rojas, 1996). A neuron may have many inputs, but it has only a single output (Dailey and Lin, 1996).

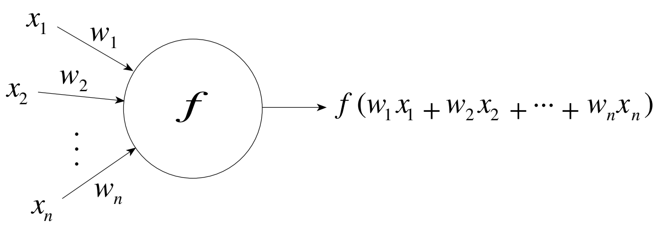


Figure 2.19: An abstract neuron. Taken from (Rojas, 1996).

(Dailey and Lin, 1996) compiles the three defining elements of a neural network as the following:

- The neuron's characteristics (the equations that define the behavior of the neuron);
- The learning rule (weights variance according to neurons stimuli);
- The network topology (connection between neurons).

A particular feature of neural networks is that it always require a learning period to fully establish and test the patterns or rules of the system, which through a multi-layer neural network, is a simple error feedback process. During the learning process each neuron is "taught" with data associations between several input data and its outputs. Throughout these stages, a human "teacher" can extract proper knowledge by adjusting the weights of each neuron until "a know pattern is fully duplicated." (Dailey and Lin, 1996). Figure 2.20 depicts the architecture of the original genre of neural networks systems, that from a biological point of view is called a perceptron (Dailey and Lin, 1996, Rojas, 1996).

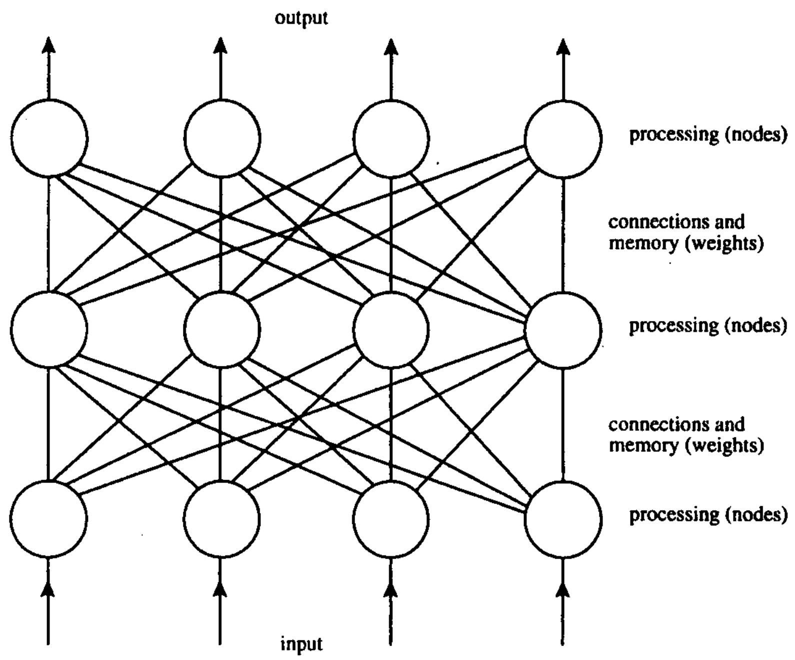


Figure 2.20: Architecture of the original genre of neural network systems. Taken from (Rojas, 1996).

A perceptron consists of four main functions, namely, input/output interfaces; processing (nodes responsible for information-handling tasks); memory (storing information); and connections between the neurons for information flow and control (Dailey and Lin, 1996).

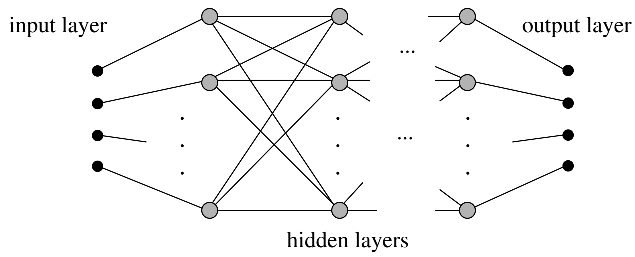


Figure 2.21: A generic layered neural network. Taken from (Rojas, 1996).

In a more generic and simplistic form, a neural network can be defined as group of layers, as depicted in Figure 2.21, where the set of inputs is called the input layer, the set of outputs is the output layer, and the other layers with no direct connections from or to the outside are called hidden layers (Rojas, 1996).

2.7.4 Possibilistic Theory

Possibility theory was originated from Zadeh (Zadeh, 1999) and was later extended by Dubois and Prade (Dubois and Prade, 1988, 2000). It is based in the previous work of Zadeh on fuzzy sets theory (see Section 2.6.2), but is designed to represent incomplete rather than vague data (Borotschnig et al., 1999, Khaleghi et al., 2013). In this theory, imperfect data treatment is similar to probability and DSET theories, diverging in the quantification process (Khaleghi et al., 2013). The possibilistic theory rules resembles those applied for fuzzy reasoning with the distinction of being normalized. The choice of appropriate fusion rules is dependent on the how reliable the data sources are, as in fuzzy reasoning. However, the basic symmetric conjunctive and disjunctive fusion rules of fuzzy set theory are sufficient only for restricted cases (Khaleghi et al., 2013). Possibility theory handles imperfect data in two different ways (Florea et al., 2002):

- When possibility measures are available we are talking about the numerical aspect of the possibility of occurrence of certain faults;
- When the measures of possibility are not available, then the possibility theory deals only with the relations between the possibilities (a minimum information must be known on the order between the possibilities).

There are a number of enhancements of possibilistic fusion methods that allow for handling more difficult fusion scenarios, as presented in (Dubois and Prade, 1988).

2.7.5 Rough Set Theory

Rough set theory, developed by Zdzislaw Pawlak (Pawlak, 1991), deals with imperfect data by representing imprecise data at the lower and upper levels, and ignoring uncertainty at different granularity levels (Khaleghi et al., 2013). (Florea et al., 2002) states that "the basic concept of the rough sets theory is to replace an uncertain imprecise information by two imprecise but certain information: the lower and upper approximations." Applying the theory to the approximations, the imprecise information is then combined. Once approximated as rough sets, data pieces can be fused using conjunctive or disjunctive fusion operators, i.e. intersection or union, respectively (Khaleghi et al., 2013).
A peculiarity of the rough sets theory is that there is no need to quantify the information's uncertainty. This characteristic is viewed as an advantage because normally it is very difficult to quantify the degree of confidence attributed to an information, and as a disadvantage since there is no difference between two pieces of information, that might have different degrees of confidence (Florea et al., 2002). However, (Florea et al., 2002) notes that the lower and upper approximations are combinations between the uncertain information and the prior knowledge about the frame of discernment. The data fusion process relies on the "data granules" type, neither too fine nor too rough (Khaleghi et al., 2013). If the "data granule" is too fine (singleton), the rough set theory simply uses the so called classic set theory conjunctive or disjuntive operators. On the other hand, for very rough "data granules" (large subsets), the lower approximation of data is likely to be empty,

resulting in total ignorance. The major advantage of rough set theory over other alternatives is that it does not require any preliminary or additional information such as data distribution or membership function (Khaleghi et al., 2013).

2.7.6 Methods Summary

(Ribeiro et al., 2014) made a comparative study between techniques of computational intelligence applied for Hazard Detection and Avoidance (HDA).

Table 2.3: Main characteristics of data fusion methods applied to HDA. Adapted from (Ribeiro et al., 2014).

Method	Pros	Cons
Fuzzy Set Theory	High semantic interpretation; heterogeneous data fusion because of intelligent data representation/normalization capability; handles uncertainty and imprecise information	For rule-based paradigms it is context dependent and not easily adapted; needs application domain knowledge for data representation
DSET	Heterogeneous sensor data fusion with missing and noisy data	Difficult to define the belief and plausibility measures; needs application domain knowledge for data representation
Neural Networks	Heterogeneous data fusion; universal function approximator; fuses multiple sensor data with missing and noisy data	Training requires a good and sufficient sampling set; time consuming training; non-adaptable without retraining
Evolutionary Computing and Swarm Intelligence	High computational efficiency and possible parallelization capability; Accepts linear and non-linear parameters; Several algorithms available e.g. PSO, Tabu	Near optimal solution; difficult to parameterize and does not allow representing uncertainty; requires difficult initial parameter tuning

In Table 2.3, a summary of some of the methods previously presented, are dissected in terms of their application in HDA context (subject reviewed in Chapter 3).

2.8 Data Fusion Applications

As seen in Section 2.5.1, the JDL model was intended for military applications under the jurisdiction of the United States DoD for battlefield surveillance and automatic target recognition among others, but data fusion applications do not focus only on this domain. In recent years, the benefits of data fusion have motivated researches in a variety of application areas as remote sensing, automated control of industrial manufacturing systems, medical applications, intelligent transportation systems, and robotics among others (Liggins et al., 2008, Ch. 1).

2.8.1 Military

Military applications of multi-sensor integration and fusion are in the area of intelligence analysis, situation assessment, force command and control, avionics, and electronic warfare. Radar, optical, and sonar sensors with various filtering techniques have been employed for tracking targets such as missiles, aircrafts, and submarines (Luo et al., 2002)(see Figure 2.22).

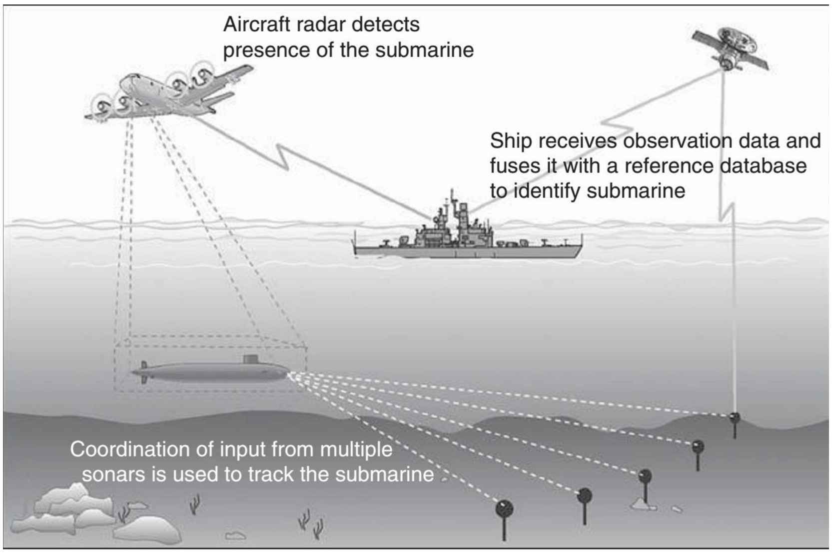


Figure 2.22: Example of an ocean surveillance system. Taken from (Liggins et al., 2008, Ch. 1).

(Liggins et al., 2008, Ch. 1) pointed out some defense systems applications such as ocean surveillance, air-to-air and surface-to-air defense, battlefield intelligence, surveillance, target acquisition, and strategic warning and defense. (Liggins et al., 2008, Ch. 30) proposed a fusion method to combine ground and satellite data via an army battle command system while (Liggins et al., 2008, Ch. 31) have developed information fusion methods for Combat Identification (CI) to help determine with high-confidence the allegiance of an object such as friend, neutral, or hostile.

In (Liggins et al., 2008, Ch. 27), a survey of United States DoD data fusion activities performed in 1999-2000, have revealed the extensive Research and Development (R&D) on multi-sensor data fusion in the United States military, with more than 70 applications. Applications such as land-mine detection, fusion of radar data with Identification Friend or Foe (IFF), and position and attribute fusion of surveillance radar, Electronics Support Measure (ESM), and a tactical data link are point by (Luo et al., 2002).

More recently, in 2013, the Atlantic Council proposed to adopt missile defense sensor data standards to enable cost-effective development of universal data fusion devices for all missile defense systems on the international market (OReilly, 2013).

2.8.2 Remote Sensing

Remote sensing is the process of acquiring data/information about objects/substances from a location that is distant from the data source. Remote sensing is based on image analysis for interpreting specific criteria from a remotely sensed imagery (Morgan and Falkner, 2001). (ESA, 2014c) consider three essential elements in remote sensing: a platform to hold the instrument (e.g. aircraft, satellites); a target object to be observed (e.g. Earth's surface); an instrument or a sensor to observe the target. (Navalgund, 2002) classify sensors as passive or active, in which sensors which sense natural radiations, either emitted or reflected from the Earth, are called passive sensors. On the other hand, sensors which produce electromagnetic radiation in order to interact with the target are called active sensors. One can then conclude that sensors can be either imaging or non-imaging, and are also classified on the basis of range of electromagnetic region in which they operate such as optical or microwave (Navalgund, 2002). In (NASA, 2014b) is presented a very detailed list of sensors currently in use by NASA.

The main purpose of remote sensing systems is to manipulate information (e.g. analogue versus digital) that is obtained from the acquired data, and how it is used and stored (ESA, 2014c). Remote sensing along with it sciences such as photogrammetry, Global Positioning System (GPS) and Geographic Information System (GIS) have gradually progressed during the last years. Applications of remote sensing include assessment and monitoring climate, environment, water sources, soil and agriculture; traffic surveillance, monitoring and management; transportation infrastructure management; and hazards, safety and disaster assessment (Luo et al., 2002).

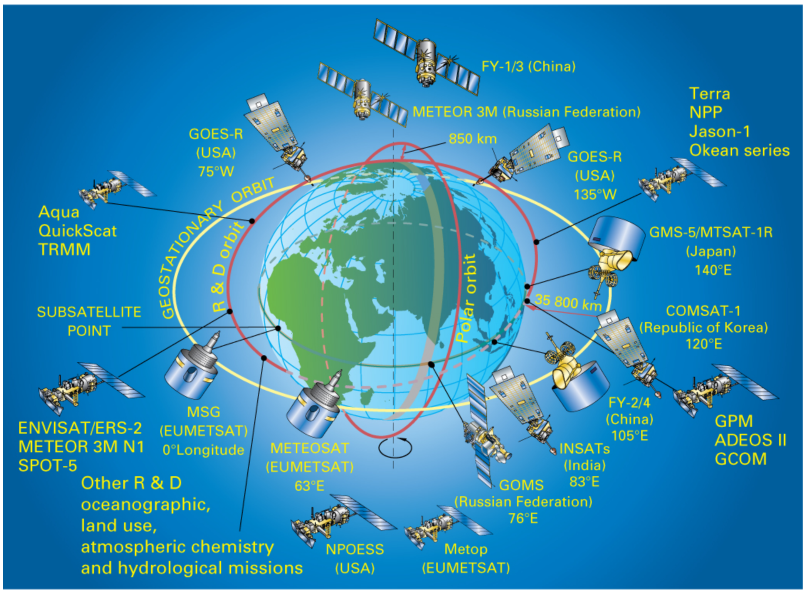


Figure 2.23: The World Meteorological Organization Global Satellite Observing System, 2009. Taken from (WMO, 2010).

In Figure 2.23 it is illustrated a fleet of satellites of the World Meteorological Organization (WMO) that provide data to different user communities, in the field of meteorology, oceanography and climate (WMO, 2010).

2.8.3 Equipment Monitoring and Diagnostics

Conventional approach to machine/equipment monitoring and diagnostics is based on identification of anomalies between the current and nominal values of certain machine characteristics. Condition-Based Maintenance (CBM) is a task of performing maintenance on a machine or system only when there is objective evidence of need or impending failure (Filev et al., 2010). Diagnostics has traditionally been defined as the ability to detect and sometimes isolate a faulted component and/or failure condition. Time-based or use-based maintenance involves performing periodic maintenance after specified periods of time or hours of operation. CBM innovation has the potential to decrease life-cycle maintenance costs (by reducing unnecessary maintenance actions), increase operational readiness, and improve safety (Khaleghi et al., 2013). (Liggins et al., 2008, Ch. 28) work is a good example of equipment monitoring and diagnostics for electromechanical systems. Implementation of CBM depends on multi-sensor data (e.g. vibration levels, temperature, pressure, and oil debris) to perform predictive diagnostics, i.e. diagnosing the current state or health of a machine and predicting time to failure based on an assumed model of anticipated use. While most mechanical defects can be determined through analysis of the vibration data generated from accelerometers, the data gathered from the remaining sensors are necessary to fully assess the state, life expectancy, and potential failure of the machine (Duhaney et al., 2010).

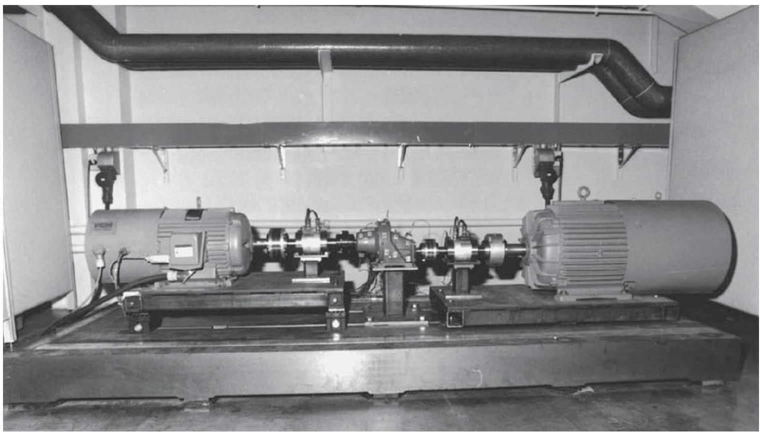


Figure 2.24: Penn State Applied Research Laboratory mechanical diagnostics testbed. Taken from (Liggins et al., 2008, Ch. 28).

Equipment monitoring and diagnostics applications extend to a myriad of areas, from simpler electromechanical systems (see Figure 2.24), to more complex ones such nuclear power plants (Agency, 2007), monitoring of inaccessible ocean machinery (Duhaney et al., 2010), and wind turbines (Márquez et al., 2012).

The diagnosis is commonly performed using feature-based techniques since each sensor of a multi-sensor system may contribute with a unique set of features. Combining these features provide a better estimate of the object's identity. The decision-level fusion is used for the classification stage of the machinery condition, applying techniques such as Bayes inference and DSET (Liggins et al., 2008, Ch. 28). Fuzzy logic is a method that has been successfully applied in many different fault detection and diagnosis technical processes (Ribeiro, 2006).

2.8.4 Biomedical

Multi-sensor data fusion is a powerful solution for solving difficult pattern recognition problems such as the classification of bioelectrical signals or biochemical parameters which allow fast, reliable and accurate information in search of symptom analysis and disease diagnosis. (Liggins et al., 2008, Ch. 29) studied the adaptation of data fusion to several chemical and biological sensors. Biosensing devices offer considerable advantages, such as specificity; small size; faster response; and cost, and therefore are an essential prerequisite for effective healthcare (Malhotra and Chaubey, 2003, Moslem et al., 2012). (Lee et al., 2008) depict a pervasive healthcare monitoring system (Figure 2.25) which enables real-time and continuous healthcare monitoring.

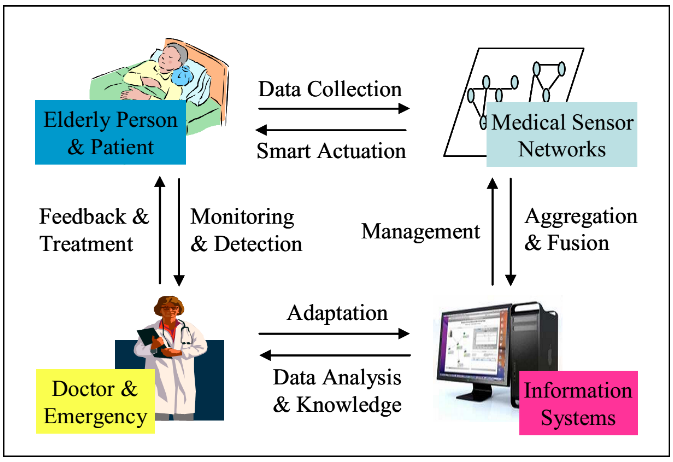


Figure 2.25: A Pervasive Healthcare Approach. Taken from (Lee et al., 2008).

(Moslem et al., 2012) applies data fusion for improving the classification accuracy of uterine EMG signals while (Luo et al., 2002) refer works to enhance automatic cardiac rhythm monitoring by integrating ECG and hemodynamic signals; case-based data fusion methods to improve clinical decision support; and important medical image fusion to, for instance, detecting the esophagus inner wall from ultrasound medical images.

2.8.5 Transportation Systems

Nowadays there are different sources of data to aid the management of transport systems such as for instance, road sensors for traffic data collection; GPS and Radio-Frequency devices for bus location in public transports; communications via Wi-Fi and Bluetooth; and environment sensors for weather, ozone and CO2 assessment (Veloso et al., 2009). A transport system can be considered smart if it is capable of dealing with situations such as safety, traffic congestion, obstacles or modal integration by linking all sources of data to produce valuable information for transport users and operators. Combining data from several and distinct sources in a transportation system can be achieved using the data fusion methods previously addressed. Therefore, implementing Intelligent Transportation Systems (ITS) in a wide variety of applications in the different modes of transport, for both passengers and freight, has a great potential (EU, 2011). (Veloso et al., 2009) survey in transportation data fusion analyzed several federal and nonfederal applications worldwide highlighting functionalities, progresses and future works. Over the last decade, in road transport, applications (Veloso et al., 2009, EU, 2011) such as electronic tolling, dynamic traffic management, real-time information and other advanced driver assistance systems (Darms et al., 2010) like electronic stability control and cooperative adaptive cruise control (Robertson et al., 2010) have been implemented with success. In Figure 2.26 is depicted an example of an advanced driver assistance system.

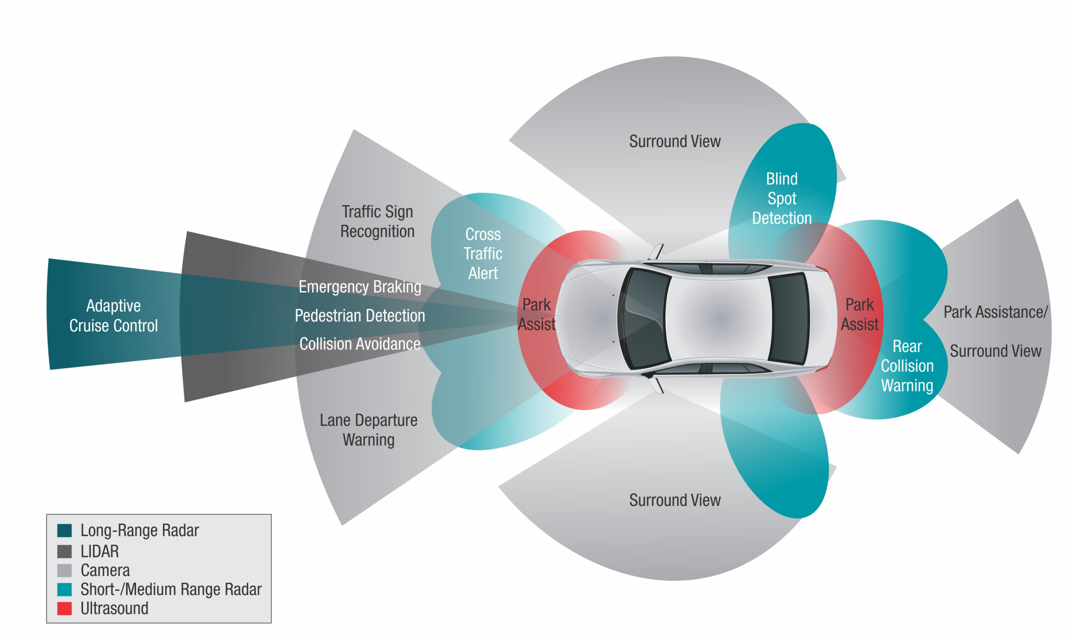


Figure 2.26: Example of an Advanced Driver Assistance System. Taken from (TI, 2013).

The European Commission's ITS action plan EU (2011) has established, in 2011, six principal action areas of intervention for the deployment of ITS in Europe:

- Optimal use of road, traffic and travel data;

- Continuity of traffic and freight management ITS services on European transport corridors and in conurbations;
- Road safety and security;
- Integration of the vehicle into the transport infrastructure;
- Data security and protection, and liability issues;
- European ITS cooperation and coordination.

From (Dailey and Lin, 1996, Luo et al., 2002) works it is clear that the advantages of KF (see Section 2.7.1.2) in positional estimation and error prediction location are widely appreciated in ITS context. (Papadimitratos et al., 2009) work provides an overview of several ITS projects and vehicle communication systems. (Barth and Boriboonsomsin, 2012) study focus on improving environmental performance of ITS applications, and their approach emphasis on Advanced Traveler Information System (ATIS), one of several ITS technologies, that offers users integrated traveler information before and during travel, thereby providing a wider range of choices about how, when, and where to travel based on individual interests and needs.

2.8.6 Robotics

The development of technology allowed robotic systems to have a great variety of sensors to obtain better information from the environment. Robots with sensory capabilities are required in many industrial applications to enhance their flexibility and productivity. Therefore, multi-sensor data fusion has a key role in robotic systems, addressing the problem of data combination from multiple sensors (Luo et al., 2002). In (Luo et al., 2002) is also pinpointed that multi-sensor integration and fusion techniques are suitable for application areas of industrial robots such as material handling, part fabrication, inspection, and assembly. With recent advances in robotics arise multirobot cooperative system (Arai et al., 2002, Farinelli et al., 2004), dexterous hands (Ciocarlie et al., 2007, Ciocarlie and Allen, 2009)(see Figure 2.27), interaction between the robot and the environment (Nehmzow, 2003), teleoperation (Fong and Thorpe, 2001, Cui et al., 2003, Dragan et al., 2013), and so forth.
(Durrant-Whyte and Henderson, 2008, Ch. 25) identifies the two most general areas of multi-sensor fusion in robotics as:

- Dynamic system control: involve real-time feedback with application of uncertainty models and sensors to control the state estimation of a dynamic system (applied in, e.g. industrial robot, mobile robots, autonomous vehicle, etc.);
- Environment modeling: model of some aspects of a physical environment (e.g. the interior of a building).

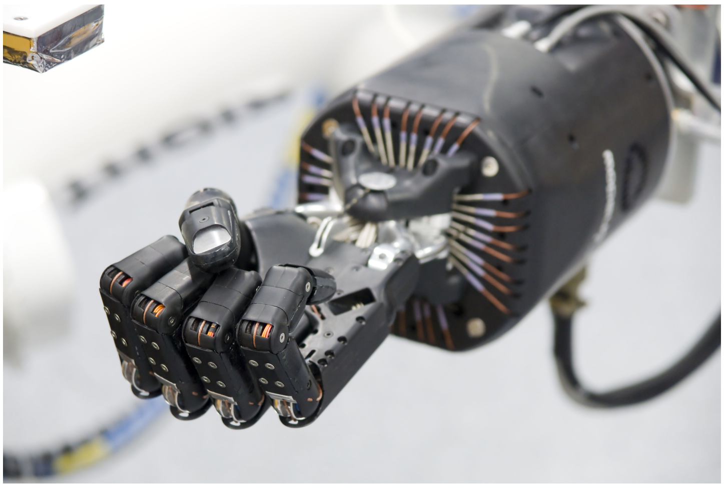


Figure 2.27: An artificial hand for robot arm. Taken from (ESA, 2010).

A great advance in robotics were the autonomous Unmanned Vehicles (see Section 1), applied in a wide variety of environments and purposes. As in (Luo et al., 2002) work, considering the operation in an uncertain or unknown dynamic environment, integrating and fusing data from multiple sensors enable autonomous robots to achieve quick perception for navigation and obstacle avoidance. Perception, position location, obstacle avoidance, vehicle control, path planning, and learning are necessary functions for an autonomous mobile robot. Multi-sensor fusion and integration of vision, tactile, thermal, range, laser, radar, infrared to name a few, play a very important role for these robotics systems (Luo et al., 2002).

3

Hazard Detection and Avoidance

Contents

When the subject is space exploration, safety is of utmost importance. For instance, spacecraft landing site comes across as a factor with a high importance level. The ability to find a landing site free from any type of hazards or at least minimize the dangerousness of the site, is taken into utmost consideration since there will be no scientific return if the spacecraft does not land safely (Howard and Seraji, 2004, Seraji and Serrano, 2009).

In this chapter we present the FUSION project, which topic was safe spacecraft landing with HDA, and then the method to be compared, the IPSIS algorithm. In Chapter 4 it is discussed the fuzzy reasoning method because this was the one implemented in the scope of this thesis work.

3.1 Hazard Detection and Avoidance in Planetary Entry, Descent and Landing

Landing a spacecraft in the Moon, in a planet like Mars or even in a small body such as comet or asteroid, is a delicate task because the terrain available may be quite irregular and can jeopardize the spacecraft landing and thus, the mission (Seraji and Serrano, 2009). It is then necessary to extract and analyze terrain characteristics that satisfy engineering constraints, typically with a set of on-board equipments. The data collected from these equipments must be fused during the entry into the planetary gravity field and atmosphere, descent through that atmosphere towards a region of scientific interest, and landing of the spacecraft preserving the integrity of on-board instrumentation. The Entry, Descent and Landing (EDL) sequence (see Figure 3.1) is critical and must be autonomous (Jones and Howard, 2006).

Figure 3.1: EDL phases of Curiosity Mars Rover. Taken from (NASA, 2014a).

During the EDL, sensor data can be used to analyze terrain features such as craters and boulders, roughness or even the surface slope, to list a few. This analysis enables any trajectory adjustments and alignments to minimize the risk of spacecraft landing (Serrano et al., 2006, Howard and Seraji, 2004). Therefore, landers must be free of human control and be capable of detecting hazards in the region of interest, selecting a safe landing site, and do the proper adjustments and maneuvering to the selected site. Besides the HDA tasks described earlier, real-time communication with Earth is also an obstacle when doing space exploration missions in deep space due to delay and low bit-rate. It is then essential a spacecraft to have, as stated before, some type of autonomous Guidance, Navigation and Control (GNC) behaviour (Zexu et al., 2010).

3.1.1 Safe Site Selection

The most important factor in landed space exploration missions is safety, and therefore it is crucial to find a solution that best contributes to this selection. Whereas site selection assessment is sensor-driven, it is essential to prevent instruments/sensors failure (Howard and Seraji, 2004, Serrano et al., 2006, Seraji and Serrano, 2009). A system with a single terrain sensor is more prone to failure, noisy and misleading information than a multi-sensor system. To overcome unforeseen events and make a robust hazard assessment, multiple data sources can be used to provide a more efficient process for safe site selection. During spacecraft descent phase, a passive sensor like a camera brings benefits in terms of detecting terrain characteristics with its higher resolution but may fail to detect sloped terrain changes under specific lighting conditions. Thus, the inclusion of a high resolution sensor such as a lidar, can reduce this restriction, which is very important at lower altitudes. Lidar sensors can detect surface features like texture, or even illumination levels (camera) or even surface slope and roughness information (lidar) (see Figure 3.2) (Howard and Seraji, 2004, Serrano et al., 2006, Seraji and Serrano, 2009).

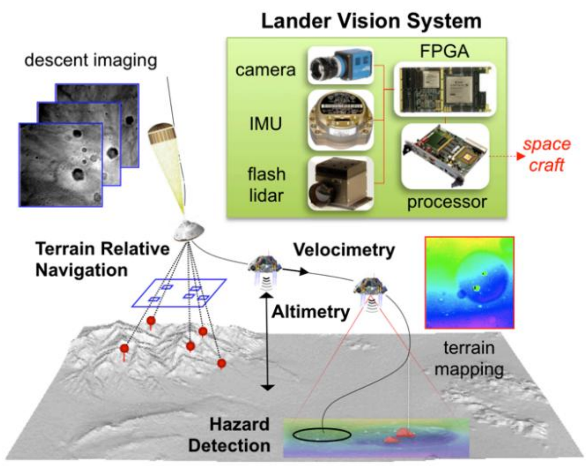


Figure 3.2: Lander vision system. Taken from (Johnson and Golombek, 2012).

Merging the camera and lidar sensors increases the capability of a safe site assessment, by combining possible hazardous features untraceable by a single sensor. However, using multiple heterogeneous sensors with different ranges, resolutions, Fields Of View (FOV) and image sizes, will need a fusion approach that combines the extracted data into a final hazard map (Howard and Seraji, 2004, Serrano et al., 2006, Seraji and Serrano, 2009). The site itself can be a single pixel or a group of pixels (a region), depending on the altitude at which the images were taken. There are numerous multi-sensor data fusion approaches and algorithms with different modules and architectures. These topics were addressed in Chapter 2.

3.2 FUSION - Sensor Data Fusion for Hazard Mapping and Piloting

The FUSION project (CA3, 2013b) included a partnership between the Computational Intelligence Research Group (CA3) of UNINOVA (CA3, 2013a) and Spin.Works (Spin.Works, 2014). The main objectives of the developed project were essentially the following:

- Trade-off terrain sensors fusion solutions for HDA applied to planetary EDL;
- Development of a set of complete hazard mapping data fusion techniques and integration into a simulation environment;
- Perform the benchmarking of the proposed solutions based on their complexity, performance, and robustness;
- Assess the applicability of the methods to terrain relative navigation;
- Provide a roadmap for future developments.

The developed data fusion techniques would have as inputs a set of hazard maps for the consequent data fusion (see Figure 3.3), originating an overall fused hazard map for a final evaluation.

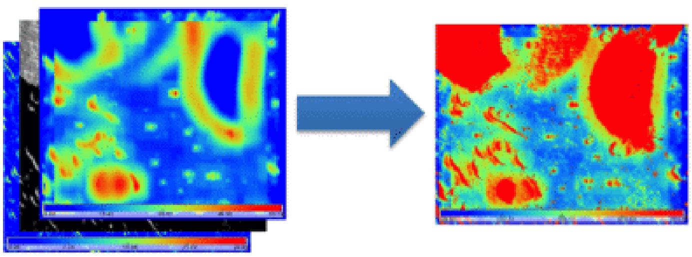


Figure 3.3: Hazard maps fusion. Taken from (CA3, 2013b).

Other aspect of the project was the comparison between three distinct data fusion and piloting approaches namely H2DAS - Hybrid Hazard Detection and Avoidance; IPSIS - Intelligent Planetary SIte Selection; and 3MSF - Tiered multi-sensor fusion. IPSIS is an algorithm property of UNINOVA and an overview is made in Section 3.3.

In this work will be compared two approaches, the IPSIS algorithm and a fuzzy reasoning algorithm, both developed by UNINOVA.

3.3 IPSIS - Intelligent Planetary SIte Selection

IPSIS is an algorithm that uses a dynamic decision-making model with the purpose of selecting safe landing sites in real-time, during a spacecraft's final phase of descent onto a planetary body.

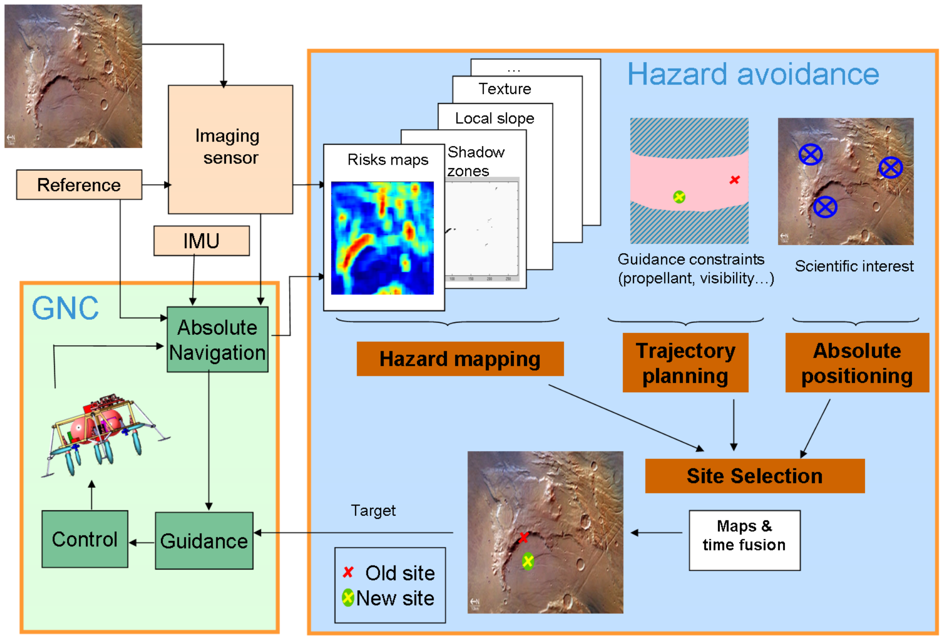


Figure 3.4: HDA architecture and link with GNC. Taken from (Simões et al., 2012).

The IPSIS hazard avoidance function implements the capability to autonomously select and reach a safe landing site. It processes inputs from an imaging sensor and the navigation function provide a target site for guidance. As stated in the beginning this section, selecting, in real-time, safe landing sites which meet mission safety and reachability requirements is of utmost importance for HDA. From (Simões et al., 2012), safety requirements are assessed by a Hazard Mapping (or Hazard Detection) function, which uses imaging sensor(s), such as camera or lidar, to overview and map the terrain in terms of hazards (craters, rocks, high slopes, shadows, etc.). The reachability of potential landing sites is evaluated using a specific guidance function, called Trajectory Planning. If the mission has special constraints, e.g. to land as close as possible to a predefined site of high scientific interest, along with reachability evaluation, are linked with spacecraft navigation. These functions provide the necessary inputs for producing spatial-temporal multi-criteria assessments of candidate landing sites' quality. From these metrics, a piloting function monitors in real-time the suitability of its current target, and provides the autonomous decision-making for

whether to retarget, and where to (Simões et al., 2012). Figure 3.4 illustrates the connections between HDA and GNC modules.
The site selection function (also called piloting) is in charge of selecting in real-time a suitable landing site and providing it to trajectory planning and guidance (Simões et al., 2012). A safe site must meet mission, safety and reachability requirements, such as:

- The site must be reachable with the remaining fuel and the spacecraft propulsive capabilities;
- The site must remain visible by the imaging sensor throughout the descent so that its estimated characteristics can be continuously updated;
- The site must be safe: in a lighted area, with rocks below a certain size and slopes below a certain value;
- The site must be compliant with mission constraints such as scientific interest, visibility from Earth, etc;
- The selected site has to depend on the lander size and respective altitude, therefore it can be one pixel or a region (composed of several pixels).

Criteria for decision are the hazard estimates elaborated by the Hazard Mapping function. These can be estimates of fuel cost to reach a given site (reachability) and mission data (for example, if defined, scientifically interesting sites). Criteria values can be provided in the form of maps the size of the image or evaluated on a case by case basis depending on the retained algorithm.

3.3.1 IPSIS Approaches to Site Selection

In order to select a safe site for landing, IPSIS is programmed to use two distinct site selection approaches:

- Exhaustive site selection approach;
- Non-exhaustive site selection approach.

The exhaustive approach is denominated as Oracle and it analyses every single pixel on the input maps, searching for the optimal landing site. The downside of this approach is that is hardly feasible in a real-time environment with the foreseen CPU platforms used in nowadays spacecrafts (e.g. LEON 3). One of the greatest advantages of IPSIS algorithm is the non-exhaustive search method that performs a trade-off between absolute certainty in finding the best site and a better time efficiency (computational time) than an exhaustive approach, in which the Oracle's approach is inserted. The non-exhaustive approach locate the best site using an intelligent navigation through a minimal sample of the alternatives set the following evolutive programming algorithms(Simões et al., 2012):

- Particle Swarm Optimization (PSO);
- Tabu Search;
- PSO-Tabu (a hybrid between the first two algorithms).

The differences between the classical method (exhaustive evaluation of sites) and the non-exhaustive evaluation are depicted on Figure 3.5 and Figure 3.6 (Bourdarias et al., 2010). Both approaches use historical information with the ranking of evaluated sites acquired in previous iterations.

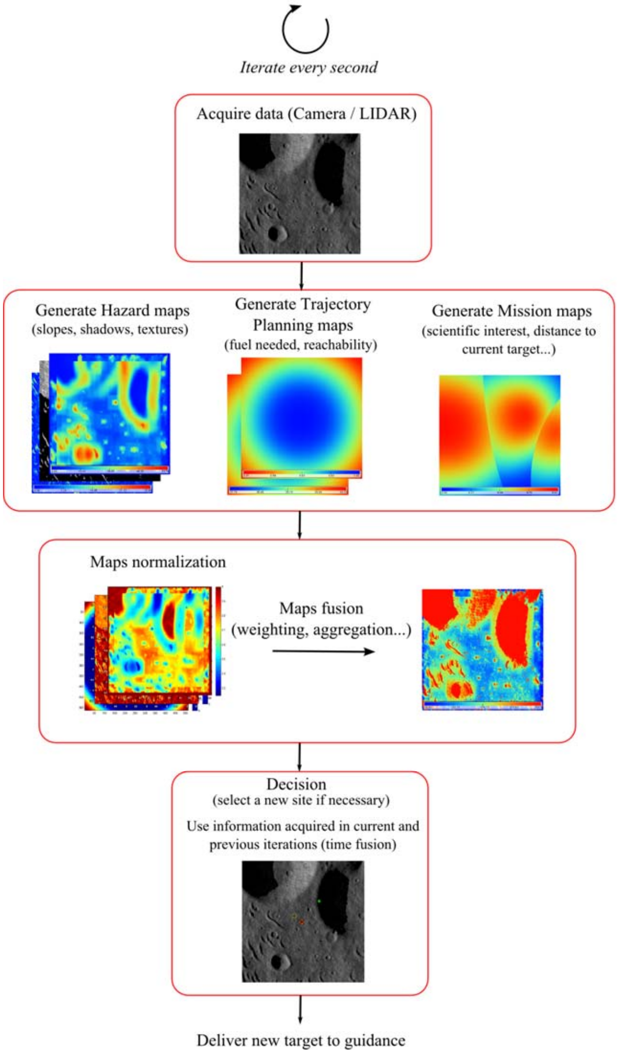


Figure 3.5: Simplified view of classical site selection based on exhaustive evaluation. Taken from (Bourdarias et al., 2010).

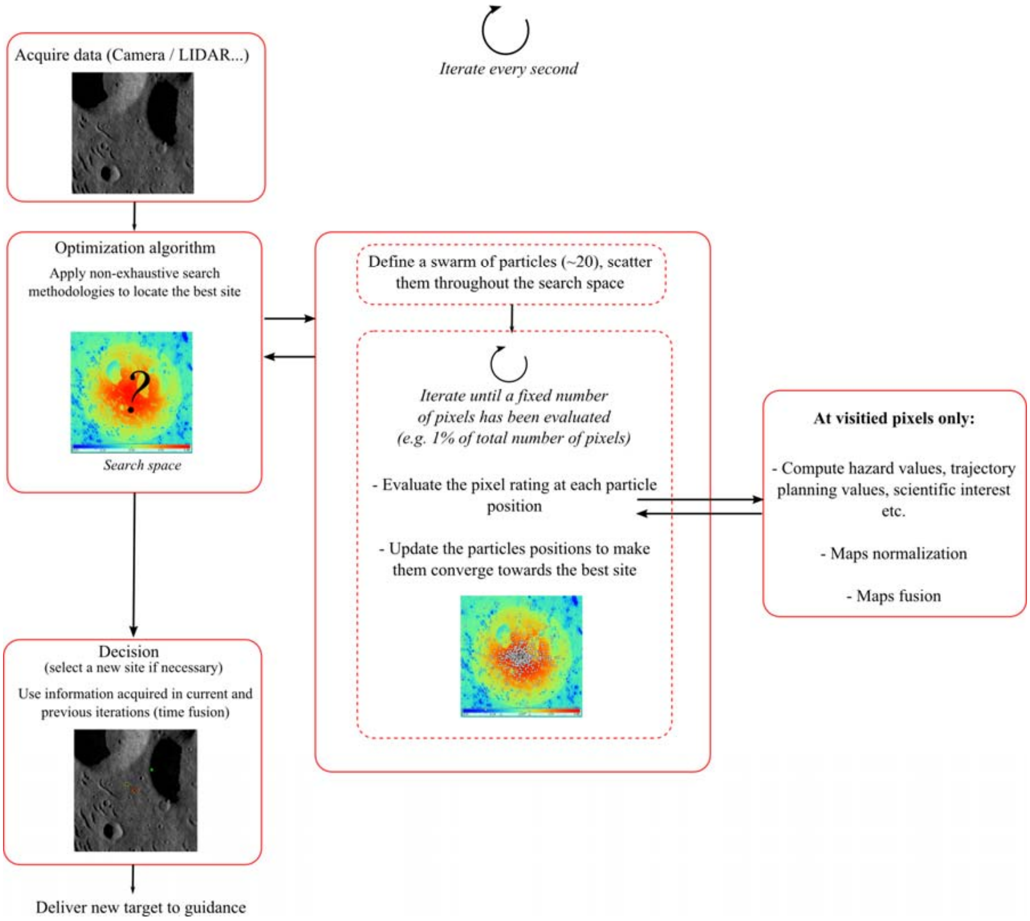


Figure 3.6: Simplified view of site selection based on non-exhaustive search methodologies. Taken from (Bourdarias et al., 2010).

4

Developed Work

Contents

All the software implementations developed for this dissertation are presented and discussed in this chapter, with the steps and considerations taken. Both Fuzzy Reasoning Algorithm (FRA) for landing site assessment, and Oracle Viewer for data visualization and analysis, main functions and functionalities are described in detail.

4.1 Fuzzy Reasoning Algorithm

In this section it is presented a Fuzzy Reasoning Algorithm (FRA) based on the work presented in (Howard and Seraji, 2004, Serrano et al., 2006, Seraji and Serrano, 2009), and its implementation closely follows those work's proposed methods. The algorithm was programmed in C language and the inference engine performs multi-sensor information fusion, combining terrain features derived from heterogeneous sensors into a final terrain safety map. As any other fuzzy logic approach, it includes a model expressed in terms of a fuzzy vocabulary where the underlying relationships, between the related fuzzy sets, are represented by rules (Ribeiro, 2006). These systems are commonly called Fuzzy Inference Systems (FIS) and can be defined as a collection of If-Then rules with fuzzy predicates, where *a-priori* human expert imprecise information is modeled into linguistic rules, as approached in Section 2.6.2, to perform adequate processing into an output safety value (Babuska, 2003, Ribeiro, 2006). In HDA context, the FRA performs hazard mapping data fusion incorporating human expert knowledge, returning a set of safety maps for posterior assessment of best landing's site selection.

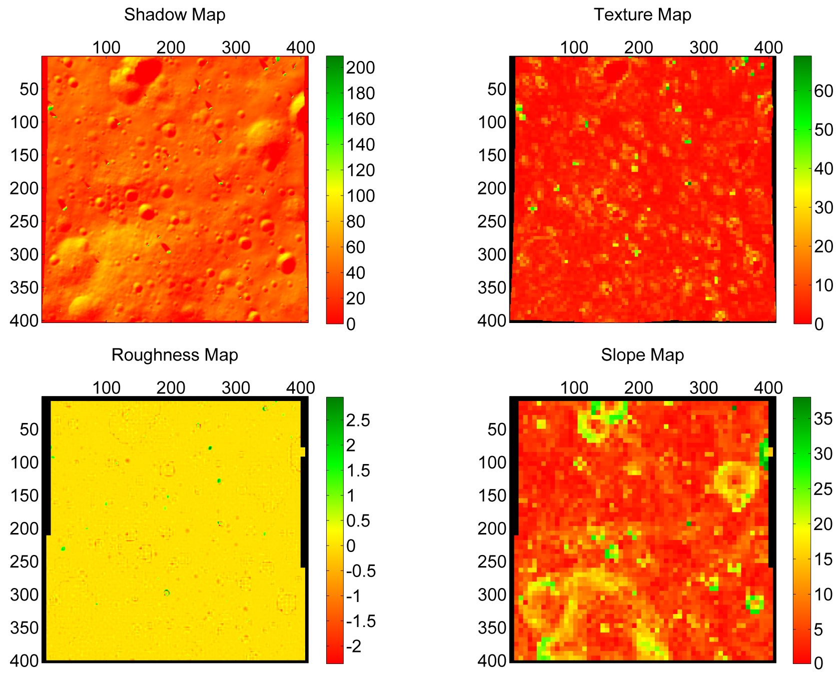


Figure 4.1: Input hazard maps.

The dataset provided by Spin.Works is the input of FRA, and consists in eleven different iterations (observations over-time), with size variable by iteration, of four different types of terrain feature maps; shadow and texture maps from camera sensor, and roughness and slope maps from lidar sensor. We look at these input terrain feature maps as matrices of numbers $(m \times n)$, being each matrix position the true targets of inference. Figure 4.1 shows a data visualization of dataset's first iteration, with matrices of size of (403×411). Each (m, n) position represents an image pixel of the full feature images acquired by the sensors. In a brief overview, the FRA engine has as objective assess the terrain for spacecraft landing and is composed by two independent rule-based FIS, one per sensor, each one modeled as a Multiple-Input Single-Output (MISO) FIS (Ribeiro, 2006), transforming terrain features into safety values, thus combining single-sensor information to produce final terrain safety maps.

4.1.1 Algorithm Description

One of the steps of the implementation was a conversion of each set of terrain feature maps, hereinafter called hazard maps, organized by type of sensor, into a two column input file compatible with MATLAB® (Toolbox, 2010). Therefore, the two FIS created are both Takagi-Sugeno fuzzy models, one for camera sensor (Cam) and another for lidar sensor (Lid), and both have two different sets of hazard maps, as stated previously. On the other hand, the fuzzification/defuzzification procedures are realized using membership functions and rules based in the system depicted in (Serrano et al., 2006). However, the domain and topology of the membership functions developed within this project parameters were made by user experimentation, and the rules were also adjusted according the current system (case study defined by Spin.Works). The description of the membership functions, number of inputs and outputs, and FIS characteristics, such like the set of rules, i.e. the general parameters of the system, are depicted in a parameters file that will work as input of the FRA. The algorithm process is concluded with the output file with FIS safety values per sensor and the final fuzzy-based safety values, i.e. the combination of each evaluated FIS pixel. The overview of this approach is described in Figure 4.2.

At this point it is known that both FIS follow the same type of reasoning, the Takagi-Sugeno fuzzy model. With the defined membership functions for each sensor rule set, the system will perform a mapping (fuzzification process) between the input value read from the input FIS file and the corresponding membership function(s) in which the value falls, processing each FIS value input, row by row, returning its membership level/fuzzified value. The membership level of an input value is then passed to the initial inference process which consists on going through a set of linguistic rules, for reasoning its safety value based on the terrain features, determining the ones that are activated, and obtaining their corresponding firing strength (Howard and Seraji, 2004). Through this procedure a fuzzy operator is applied according to the activated rule, either from

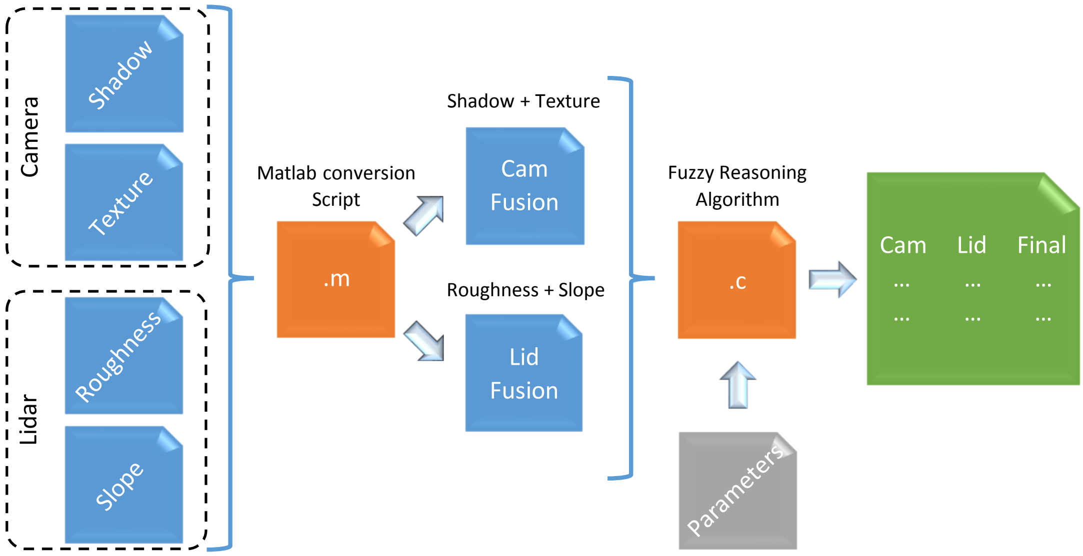


Figure 4.2: Solving approach with Fuzzy Reasoning Algorithm.

camera terrain safety fuzzy rule set or lidar terrain safety fuzzy rule set. The fuzzy operator can be either an "And Method" (intersection operator) or "Or Method" (union operator), depending on the structure of the rule, as in the following example (MathWorks, 2014):

$$\beta_i = AndMethod(MF_1(x), MF_2(y)) \tag{4.1}$$

where β_i is the rule's firing strength, and $MF_{1,2}\,(.)$ are the membership functions for $Input\ 1 = x$ and $Input\ 2 = y$. The rule's firing strength β_i result of the aggregation of the input membership levels with the corresponding operator, for each activated fired rule. A rule can be presented as follows, with an example of the respective evaluation for $Input\ 1 = Shadow = 128$ and $Input\ 2 = Texture = 74$, being then $\beta_i = 0.128$ (see Figure 4.3):

If *Shadow is GOOD and Texture is HIGH* **then** *Terrain is MUNSAFE.*

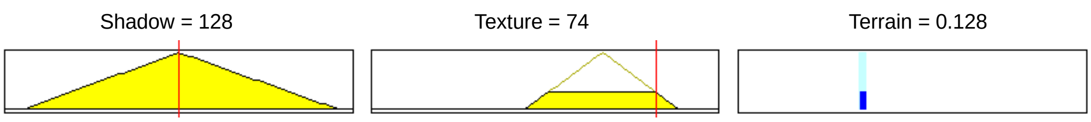


Figure 4.3: Rule evaluation example.

After obtaining all the activated rules firing strengths, the outputs of the two rule-based FIS are calculated using the Takagi-Sugeno inference scheme (suggested by the authors (Howard and Seraji, 2004, Serrano et al., 2006, Seraji and Serrano, 2009)), which is a weighted average of all FIS rule outputs. The steps described are generic and are performed twice, once for camera and another for lidar.

Finally, after obtaining the safety scores for both camera and lidar rule-based systems (Cam and Lid in the output file, see Figure 4.2), the safety scores of each pixel of both sensors are combined, again by using a weighted average. The final fuzzy-based safety map is computed by combining the two inference values with the sensor certainty values as defined below:

$$S^F(m,n) = \frac{\lambda^C S^C(m,n) + \lambda^L S^L(m,n)}{\lambda^C + \lambda^L} \tag{4.2}$$

where $S^C(m,n)$ and $S^L(m,n)$ are the safety values in position (m,n) of each sensor's matrices, and λ^C and λ^L are the certainty values for camera and lidar sensors respectively. The certainty values λ will be equal to 0.5 since it is assumed equal importance to both sensors in the fusion operation.

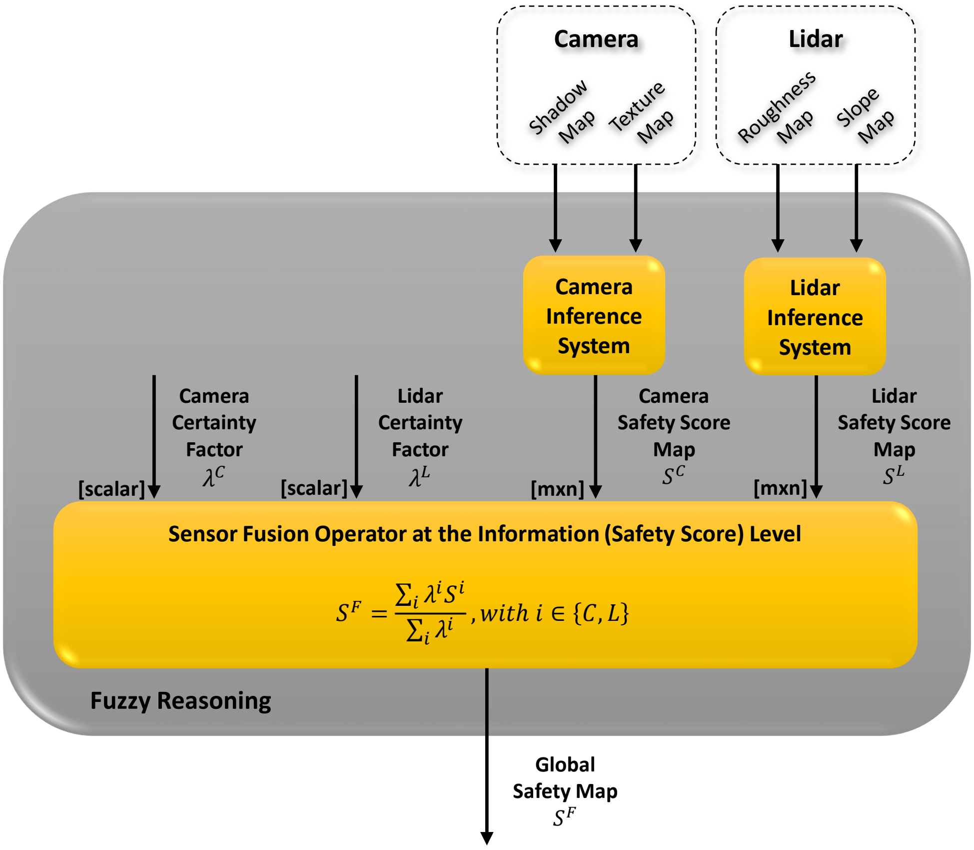


Figure 4.4: Fuzzy Reasoning model decomposition. Adapted from FUSION Internal Report.

Summarizing, the Fuzzy Reasoning model can be decomposed according to Figure 4.4.

4.1.2 Projecting Fuzzy Inference Systems

Fuzzy Inference Systems define the knowledge of the problem, which will contain the set of rules using fuzzy sets usually provided by experts. This way of representing knowledge appears to be well suited for mission control processes since, as enumerated by (Ribeiro, 2006):

- It is almost impossible to define a general mathematical model;

- The knowledge about the parameters is usually imprecise;
- The complexity of parameters behavior is large;
- Sometimes the variables boundaries are soft and not crisp;
- The knowledge is only partially detained by domain experts and includes human judgment;
- Operators need decision support tools to help them make timely decisions to avoid costly and potentially dangerous hazards.

In order to create, test and validate the FIS for this problem context, it was used MATLAB® Fuzzy Logic Toolbox (Toolbox, 2010). An auxiliary open-source software named Fuzzylite (Rada-Vilela, 2014) was also used as alternative for test and validation purposes. It should be noted that the operators shown in the following tables are the fuzzy t-norm (represented by *and*) and t-conorm (represented by *or*) operators, as seen in previous sections. From the t-norm and t-conorm families, here we use, respectively, the logical min and the max operators. The hazard maps are represented using a grid of cells in which the safety values were weighted by fuzzy sets with linguistic labels *{HUNSAFE, MUNSAFE, MSAFE, HSAFE}* which stand for *highly-unsafe*, *moderately-unsafe*, *moderately-safe*, and *highly-safe*, respectively (Serrano et al., 2006), subject discussed further in Section 4.1.2.3. Each cell is associated with a region (set of pixels) physically located on the planetary terrain surface.

4.1.2.1 Camera Terrain Safety Rules

The rule-based Camera FIS has the following set of rules listed in Table 4.1.

Table 4.1: Camera terrain safety fuzzy rule set. Adapted from (Serrano et al., 2006).

Shadow	Operation	Texture	Terrain
BAD			HUNSAFE
GOOD	*and*	VHIGH	HUNSAFE
GOOD	*and*	HIGH	MUNSAFE
GOOD	*and*	LOW	MSAFE
GOOD	*and*	VLOW	HSAFE

The rules set in Table 4.1 and those in (Serrano et al., 2006) for camera terrain safety differ since it was necessary to adapt the rules to the maps provided for camera sensor. However, these features are homogeneous in safety issues to the ones in the literature. The linguistic labels set for *Shadow* provide information about terrain shadow, i.e. illumination levels (if terrain is well illuminated or not), while the labels set for *Texture* define the irregularities on the terrain, being lower, better. To clarify, the fuzzy sets labeled as $\{VLOW, LOW, HIGH, VHIGH\}$ represent respectively, *very-low*, *low*, *high* and *very-high*.

4.1.2.2 Lidar Terrain Safety Rules

Lidar FIS rules, in Table 4.2, used the same criteria and evaluations in (Serrano et al., 2006) since the terrain features considered are the same.

Table 4.2: Lidar terrain safety fuzzy rule set. Adapted from (Serrano et al., 2006).

Roughness	**Operation**	**Slope**	**Terrain**
ROCKY	*or*	STEEP	HUNSAFE
ROUGH	*and*	SLOPED	HUNSAFE
SMOOTH	*and*	SLOPED	MSAFE
ROUGH	*and*	FLAT	MSAFE
SMOOTH	*and*	FLAT	HSAFE

The *Roughness* column refers to either a hazard free terrain (i.e. SMOOTH), or a terrain highly dangerous for landing (i.e. ROCKY), while *Slope* criterion detects if the terrain is appropriate for landing, including three types of considerations: FLAT (better for landing); SLOPED (considerable inclines); and STEEP (hazards and/or ravines present - worst case scenario).

4.1.2.3 Terrain Safety Classification

From Table 4.1 and Table 4.2, some examples of the linguistic rules can be presented as:

If *Shadow is GOOD and Texture is LOW* **then** *Terrain is MSAFE*.

If *Roughness is ROCKY or Slope is STEEP* **then** *Terrain is HUNSAFE*.

These type of rules fulfill the typical fuzzy rule form described in Section 2.6.2. The linguistic labels $\{HSAFE, MSAFE, MUNSAFE, HUNSAFE\}$ have a constant value associated (crisp output) between 0 and 1, that will be weighted with the degree of fulfillment derived from the antecedent proposition. In Table 4.3 are presented the crisp outputs defined for each terrain fuzzy set.

Table 4.3: Crisp output values of the linguistic singleton terrain fuzzy sets.

Terrain fuzzy set	**Crisp output**
HUNSAFE	0
MUNSAFE	0.33
MSAFE	0.66
HSAFE	1

These values were defined by trial and error because there was no information about the method on the base literature.

4.1.2.4 Defining Membership Functions

All the input membership functions are defined with a trapezoidal shape, as the example shown in Figure 4.5, being a triangle a specific case of a trapezoid in which b is equal to c in a set of points $a,\ b,\ c,\ d$.

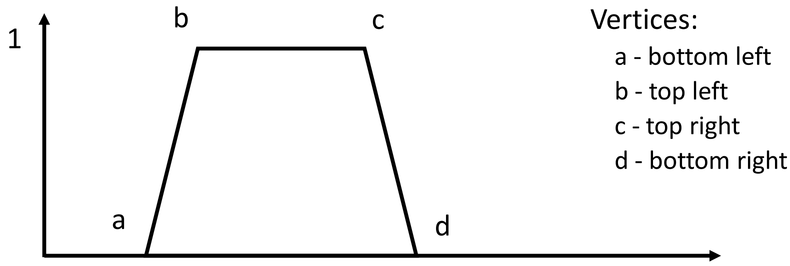


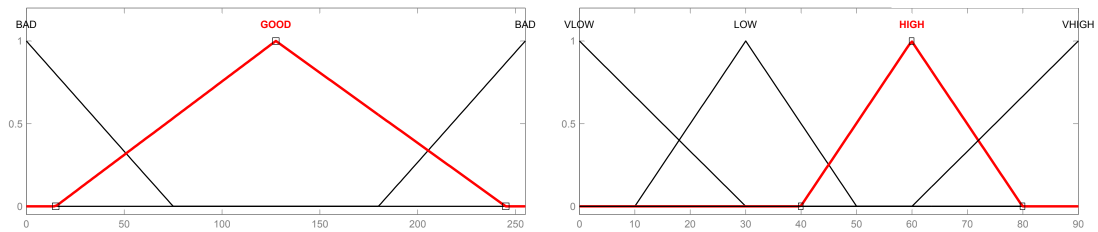
Figure 4.5: Trapezoidal membership function.

In this project, the input variables domain ranges, inclination and bending points were all defined by experimentation, since there were no details about them in the reviewed literature. Each antecedent membership function correspond to a fuzzy set and are based and modeled by the rules listed in Tables 4.1 and 4.2.

Table 4.4: Camera sensor criteria specifications.

Sensor	**Feature**	**Range**	**Fuzzy set**	a	$b = c$	d
Camera	Shadow	$[0 \quad 255]$	BAD	0	0	75
			GOOD	15	127.5	245
			BAD	180	255	255
	Texture	$[0 \quad 90]$	VLOW	0	0	30
			LOW	10	30	50
			HIGH	40	60	80
			VHIGH	60	90	90

The input hazard maps, ranges, membership functions, and membership parameters correspondent to the camera sensor are described in Table 4.4.



(a) Shadow membership functions. **(b)** Texture membership functions.

Figure 4.6: Camera sensor membership functions.

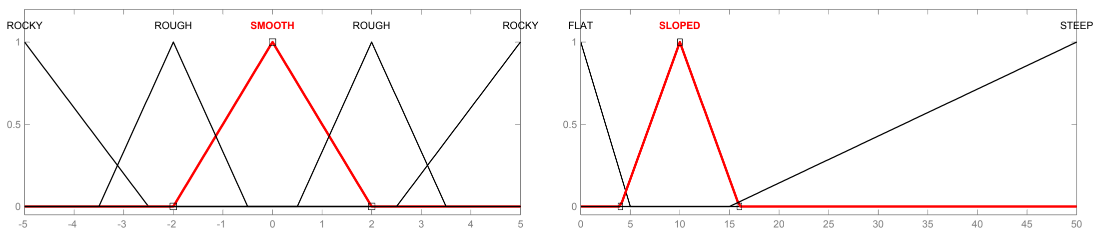
Visually, camera membership's functions are represented in Figure 4.6.

Similar to camera sensor criteria, are presented the specifications for lidar's sensor case.

Table 4.5: Lidar sensor criteria specifications.

Sensor	Feature	Range	Fuzzy set	a	$b = c$	d
Lidar	Roughness	$[-5 \quad 5]$	ROCKY	-5	-5	-2.5
			ROUGH	-3.5	-2	-0.5
			SMOOTH	-2	0	2
			ROUGH	0.5	2	3.5
			ROCKY	2.5	5	5
	Slope	$[0 \quad 50]$	FLAT	0	0	5
			SLOPED	4	10	16
			STEEP	15	50	50

Table 4.5 depicts the input hazard maps, ranges, membership functions, and membership parameters correspondent to lidar sensor. Its membership functions are depicted in Figure 4.7.



(a) Roughness membership functions. **(b)** Slope membership functions.

Figure 4.7: Lidar sensor membership functions.

4.1.3 Functions Description

In this subsection, the main functions of the developed algorithm are described With the aid of Doxygen (van Heesch, 2012), a standard tool for generating documentation for programming languages, it is possible to produce function call graphs to easily understand the operation of the algorithm.

The call graph shown in Figure 4.8 represents the "steps" taken by an execution of the FRA. The function designated by `test_all_iterations` is the actual "path" that the code follows in order to process the dataset under analysis, being the `test_coordinates` function responsible for simple tests validation.

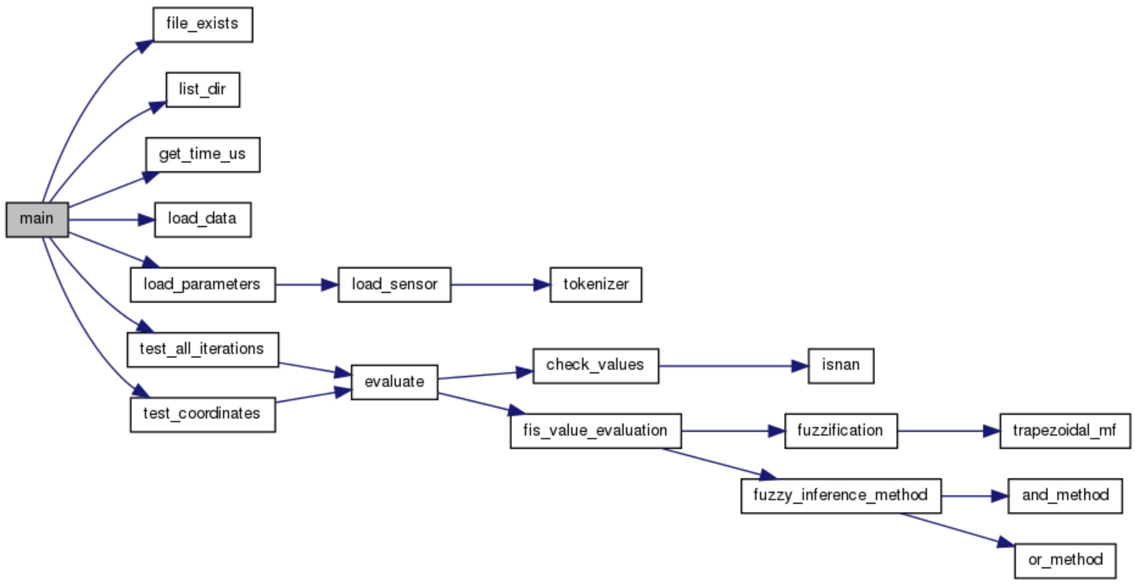


Figure 4.8: FRA `main` call graph.

4.1.3.1 Parameters File Reader Functions

The parameters file is a crucial element of the FRA execution (see Figure 4.2) and its calling is made in `load_parameters` function, in which is verified the existence of a parameters file. If the file exists, two independent callings of `load_sensor` function are made, one per sensor. The `load_sensor` function objective is to read the parameters relative to the sensors. The headers and inputs of both functions are described in the following Tables 4.6 and 4.7.

Table 4.6: FRA `load_parameters` function inputs.

`void load_parameters(FIS **fis_cam, FIS **fis_lid)`	
Input	**Description**
fis_cam	Camera FIS structure
fis_lid	Lidar FIS structure

Table 4.7: FRA `load_sensor` parameters inputs.

`void load_sensor(FILE *param, char *buf, FIS *fis)`	
Input	**Description**
param	File parameters
buf	Input data buffer
fis	FIS structure

The tokenizer function is used to process a sequence of strings into words, phrases, symbols, or other meaningful elements called tokens. In this context, the parameters' file structure is essentially an adaptation of MATLAB® FIS file structure, using Fuzzy Control Language (FCL) (IEC, 1997). The only exception are the rules' form definition, where was used the MATLAB® FIS definition instead of the linguistic rules form of the FCL. This measure was taken considering inherent programming advantages.

4.1.3.2 Evaluate Function

This is the "entry" function of a FIS general process. The `evaluate` function header and its inputs and outputs are described in Table 4.8.

Table 4.8: FRA `evaluate` function inputs and outputs.

`float evaluate(float cam0, float cam1, float lid0, float lid1)`	
Input	**Description**
cam0	Cam file input value from first column (shadow value)
cam1	Cam file input value from second column (texture value)
lid0	Lid file input value from first column (roughness value)
lid1	Lid file input value from second column (slope value)
Output	**Description**
final_fuzzy	Final fuzzy-based value

This function receives as input four values, each one corresponding to pixel values from each hazard map of the dataset. For each sensor (camera and lidar), a structure of the type `EVALUATE` is created, where all values relative to the fuzzification and defuzzification of the inputs will be stored. Inside the `evaluate` function are called the functions `check_values` and `fis_value_evaluation` (if necessary), once for camera's result, another for radar's result. The function `check_values` will return a flag (0 or 1) indicating if sensor's input is a real number or not-a-number. If the flag returned 0 then the `fis_value_evaluation` function will be called, otherwise sensor's output will be 0 for the correspondent pixel. As a return value, the `evaluate` function will give the safety's score final output by combining the inference values from both sensors (weighted average).

4.1.3.3 Fis value evaluation Function

The `fis_value_evaluation` function is the core of the fuzzy inference applied in FRA. The header and correspondent inputs are depicted in Table 4.9.

This function includes the functions `fuzzification` and `fuzzy_inference_method`, responsible for input's fuzzification and rules' activation, respectively. As input it receives two values corresponding to the respective pixels of both shadow and texture, if sensor's values under evaluation are from camera, or roughness and slope, if from lidar. It also receives a pointer to sensor's FIS structure, which contains the sensor parameters, and a pointer to `EVALUATE` structure, where the

Table 4.9: FRA `fis_value_evaluation` function inputs.

void fis_value_evaluation(float input0, float input1, FIS *fis, EVALUATE *eval)	
Input	**Description**
input0	First column value from the sensor's input file
input1	Second column value from the sensor's input file
fis	FIS parameters structure
eval	Evaluate parameters structure

values of fuzzification and defuzzification of a pixel are stored.

4.1.3.4 Fuzzification Function

This particular function is responsible for mapping numerical values in the membership functions. Table 4.10 gathers `fuzzification` function main characteristics.

Table 4.10: FRA `fuzzification` function inputs.

void fuzzification(float x, FIS *fis, EVALUATE *eval, int col)	
Input	**Description**
x	Value to be fuzzified
fis	FIS parameters structure
eval	Evaluate parameters structure
col	Column number of value to be fuzzified

The `fuzzification` function has as inputs the pixel value under evaluation, `x`, the pointer to sensor's structure in which the pixel belongs, the pointer to structure `EVALUATE` in which will be stored the results of pixel fuzzification, and the variable `col` indicating which column (one column per feature) of the to be fused input file is being considered. In this function, the value `x` will be fuzzified using the membership functions (`trapezoidal_mf function`) correspondent to the fuzzy set variables, which returns the degree of fulfillment for each membership function tested. The number of tests will be set by the number of membership functions in the sensor's FIS structure.

4.1.3.5 Fuzzy inference method Function

In the core of the `fuzzy_inference_method` function, the rules defined for each FIS are processed. In Table 4.11 are described both header and function inputs.

The `fuzzy_inference_method` has as inputs two values corresponding to the respective pixels of both shadow and texture, if sensor is camera, or roughness and slope, if sensor is lidar, the pointer of the structure `FIS`, containing all the information of the sensor in consideration, and the pointer to a structure of the type `EVALUATE` which will be used for evaluations' storage purposes. The primary objective of this function is to verify which rules, stored in the respective `FIS` structure,

Table 4.11: FRA `fuzzy_inference_method` function inputs.

`void fuzzy_inference_method(float input0, float input1, FIS *fis, EVALUATE *eval)`	
Input	**Description**
input0	First column value from the sensor's input file
input1	Second column value from the sensor's input file
fis	FIS parameters structure
eval	Evaluate parameters structure

are activated accordingly to the input values. By each rule activated a firing strength is calculated, using one of the two fuzzy logic operator functions, AND or OR, and then weighted with the respective crisp output level. The utilization of the operators and of a certain output level depends on the rule activated. This process is done for both camera and lidar FIS.

4.2 Oracle Viewer

In this section is described a visualization tool that was developed to enable data analysis and validation of safe landing sites results.

In the age of "Big Data" (Lohr, 2012), creating effective methods of data visualization is essential. As said in (McCandless, 2010) "by visualizing information, we turn it into a landscape that you can explore with your eyes, a sort of information map. And when you are lost in information, an information map is kind of useful." The visualization method proposed here follows David McCandless' premise, providing an interactive tool with the ability to reach information's core.

The Oracle Viewer is a Graphical User Interface (GUI) developed in MATLAB® with the aim of facilitating the visualization and analysis of Oracle's, IPSIS exhaustive site selection approach, output files. This visualization tool processes Oracle's outputs from two distinct CA3 projects; FUSION (see Section 3.2) and ILUV - Intelligent Landing of Unmanned aerial Vehicles (CA3, 2013c). These projects are similar in some aspects, but differ on the landing approaches of the spacecraft UAV, input hazard maps and number of iterations among others. Therefore, is mandatory that the Oracle Viewer tool to have different characteristics to each project data visualization.

4.2.1 Functions Description

To fully understand the Oracle Viewer visualization tool it is important to know how its subroutines/functions are related and which task they perform. Since this tool was created in a MATLAB® environment, it was needed an extra tool to automatically generate documentation of our MATLAB® M-files (script files) connections. Using the M2HTML tool (Flandin, 2003), a documentation system for MATLAB® in HTML, along with Graphviz open source graph visualization tool (Ellson et al., 2001), it was created a bundle of well-structured and documented information.

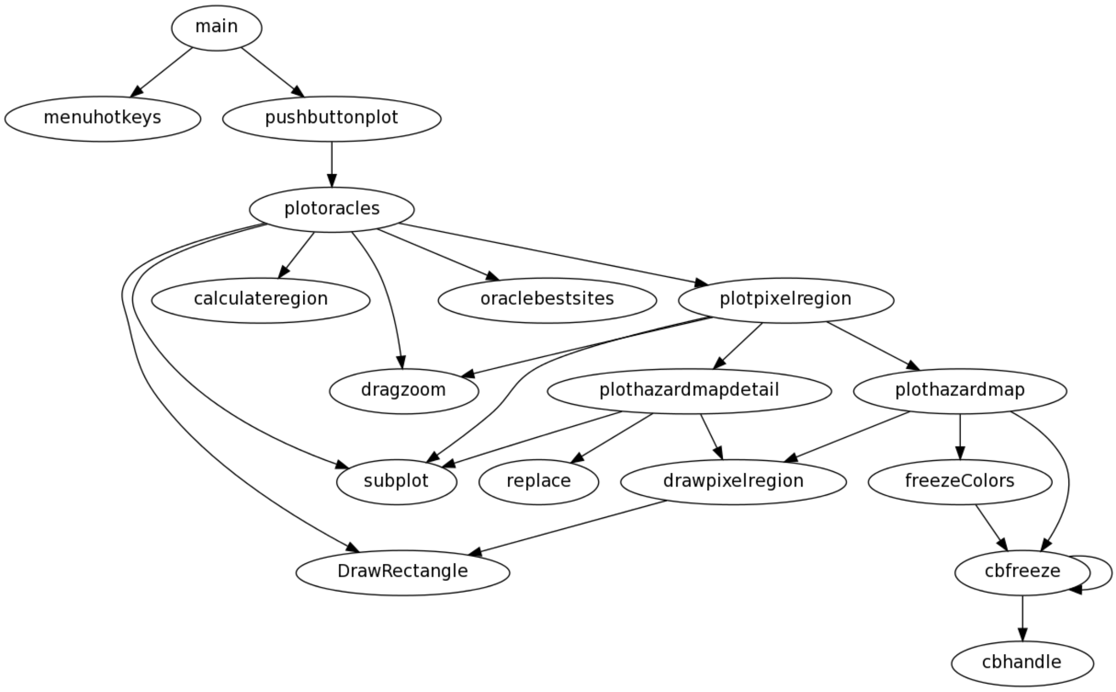


Figure 4.9: Oracle Viewer data visualization tool main call graph.

Figure 4.9 show the call graph arrangement of the M-files that consist the tool. The main M-file is responsible for the initial MATLAB®'s command window based menus (project choosing, desired dataset, landing region size), management of the inputs and workspace, and framework launcher.

Some of the functions used for this tool are available freely in MATLAB® Central File Exchange community, namely:

- "DRAGZOOM" (Pr, 2010) (named `dragzoom` function): allows the user to handy interactively manage the axes in figure;
- "Draw a rectangle" (Mojtahedzadeh, 2011) (named `DrawRectangle` function): draws a rectangle given the center, width and height, and the rotation angle;
- "freezeColors / unfreezeColors" (Iversen, 2005) (named `freezeColors` function): use multiple colormaps per figure;
- "COLORMAP and COLORBAR utilities" (Aguilera, 2009) (named `cbfreeze` and `cbhandle` function): uses MATLAB® color utilities including colormap join and interpolation; freeze and fit colorbar, etc. In this case, `cbfreeze` function freezes the colormap of a colorbar, and `cbhandle` function handle the current colorbar axes.

Outside MATLAB® Central File Exchange community, the `replace` function available in (van der Geest, 2006) was used for a particular case of ILUV's reachability map to, as the name of the function suggests, replace text strings to lookalike wind direction symbols (see Section 4.2.2.5). For further function information, consult the references indicated.

The `subplot` function of MATLAB® was also used as part of the general code but with a minor difference to the original, lower spacing between subplots. The remaining M-files were created from scratch for this tool's purpose and are described in the following subsections.

4.2.1.1 Menuhotkeys Function

This function presents an interface with a list of shortcut/hotkeys combination for GUI interaction. It is accessed via GUI Menu Bar (see Section 4.2.2.1). The `menuhotkeys` function's input is described in Table 4.12.

Table 4.12: Oracle Viewer `menuhotkeys` function inputs.

`function menuhotkeys(varargin)`	
Input	**Description**
varargin	Variable-length input argument list

4.2.1.2 Pushbuttonplot Function

This function action consists in a push button interaction that launches the GUI and plots the desired dataset's iteration. Its input parameter is described in Table 4.13.

Table 4.13: Oracle Viewer `pushbuttonplot` function inputs.

`function pushbuttonplot(~ ,~ ,Params)`	
Input	**Description**
Params	Structure with project parameters

4.2.1.3 Plotoracles Function

As suggested by the function name, the `plotoracles` function plots the scores of each pixel from the three mathematical aggregation methods (i.e. Uninorm, Fimica, and Min) of the Oracle's files. It is also responsible for making the plotted images clickable for user interaction. The processed inputs of this function are described in Table 4.14.

Table 4.14: Oracle Viewer `plotoracles` function inputs.

`function plotoracles(Params)`	
Input	**Description**
Params	Structure with project parameters

4.2.1.4 Oraclebestsites Function

For each plotted iteration, the best sites of the Oracle's files are analyzed, and are presented the top five best sites of each method in a panel with the coordinates and respective score. On the map, the best site are marked with a cross ("x").

Table 4.15: Oracle Viewer `oraclebestsites` function inputs.

`function oraclebestsites(Params,method,string)`	
Input	**Description**
Params	Structure with project parameters
method	Oracle aggregation method
string	String with name of Oracle aggregation method

For this purpose, a set of inputs must be taken into account, as shown in 4.15.

4.2.1.5 Calculateregion Function

This function does the calculations needed to center the region according to the landing region size, previously defined in one of the main menus. As input, the `calculateregion` function processes a structure of region's parameters, returning as output the same region structure with the proper adjustments.

Table 4.16: Oracle Viewer `calculateregion` function inputs and outputs.

`function Region = calculateregion(Region)`	
Input	**Description**
Region	Structure with region of interest size
Output	**Description**
Region	Structure with region of interest size adapted according to central pixel

This scenario is depicted in Table 4.16.

4.2.1.6 Plotpixelregion Function

The `plotpixelregion` function is part of the user-GUI interaction (mouse click in a plotted image) by selecting a landing region around the clicked pixel. A new GUI is then opened, containing the detailed information of dataset's input hazard maps that led to Oracle's pixel evaluation.

The set of inputs under processing in this function are shown in Table 4.17.

Table 4.17: Oracle Viewer `plotpixelregion` function inputs.

`function plotpixelregion(Params,Region,coordinates,rect,pixelorient)`	
Input	**Description**
Params	Structure with project parameters
Region	Structure with region of interest size adapted according to central pixel
coordinates	Vector with mouse click's coordinates position
rect	Pixel's region of interest rectangle
pixelorient	Pixel angle orientation (exclusive of ILUV project)

4.2.1.7 Plothazardmap Function

This function has the responsibility in plotting the input hazard maps linked with the correspondent iteration's dataset. Therefore, it is needed to known the region previously selected and its location to contextualize the hazard maps into the right place.

Table 4.18: Oracle Viewer `plothazardmap` function inputs and outputs.

`function [xc,yc] = plothazardmap(map,gca,Region)`	
Input	**Description**
map	Hazard map
gca	Current axes handle
Region	Structure with region of interest size adapted according to central pixel
Output	**Description**
xc	Region adjustment in x axis
yc	Region adjustment in y axis

Table 4.18 depicts the inputs/outputs of the `plothazardmap` function.

4.2.1.8 Plothazardmapdetail Function

The `plothazardmapdetail` function plots a detailed hazard map with in pixel feature value. This function basically acts as a auxiliary process to user decision-making and analysis of the terrain. Its inputs are shown in Table 4.19.

Table 4.19: Oracle Viewer `plothazardmapdetail` function inputs.

`function plothazardmapdetail( , ,map,Region)`	
Input	**Description**
map	Hazard map
Region	Structure with region of interest size adapted according to central pixel

4.2.1.9 Drawpixelregion Function

This function draws the landing region size (taking the previously clicked pixel as the center) in all plot functions under analysis. Basically, it serves as guide showing the user the predefined region.

Table 4.20: Oracle Viewer `drawpixelregion` function inputs and outputs.

`function [xc,yc] = drawpixelregion(Region)`	
Input	**Description**
Region	Structure with region of interest size adapted according to central pixel
Output	**Description**
xc	Region adjustment in x axis
yc	Region adjustment in y axis

Table 4.20 shows the inputs/outputs of `drawpixelregion` function.

4.2.2 User Guide

Running the main M-file of the project, the user is prompted to choose one of the projects (i.e. FUSION or ILUV) and must choose one of them. The next step is the dataset selection, one within the four available in each project. Last but not the least, is the landing region size selection. The default size is a region of (3×3) pixels for FUSION and a (10×20) pixels for ILUV's case. In the following example, it was chosen a FUSION dataset with the default region size. Once entered all the information needed, the Oracle Viewer's main interface, as shown in Figure 4.10, will be presented.

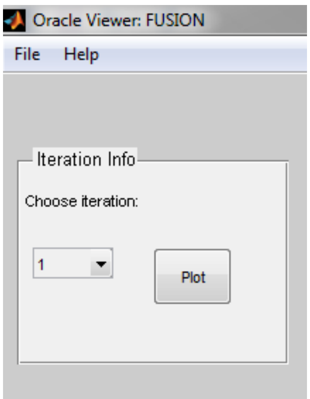


Figure 4.10: Main interface (cropped from the original window size).

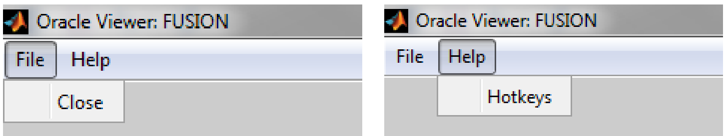
4.2.2.1 Main Interface

Initially, the Main Interface contains the following contents:

- Menu Bar;
- Iteration Info.

Menu Bar

The Menu Bar has two submenus, as shown in Figure 4.11.



(a) File submenu. **(b)** Help submenu.

Figure 4.11: Menu bar.

The "File" submenu has the option "Close" (Figure 4.11(a), for closing the GUI, while "Help" submenu has the "Hotkeys" option (Figure 4.11(b)).

Help submenu: Hotkeys

The "Hotkeys" option in "Help" submenu is a user assistance option, since the built interface offers a level of interaction between the GUI and the user. As shown in Figure 4.12, the hotkeys represent a set of keyboard shortcuts that enables image manipulation.

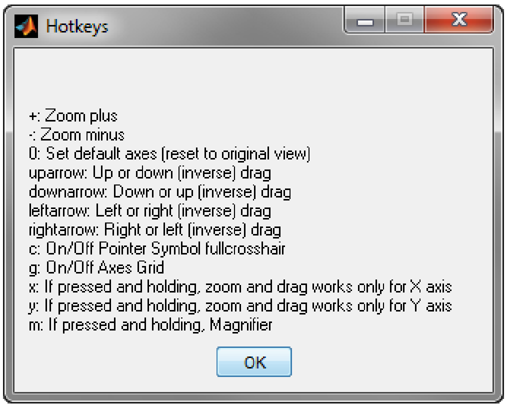


Figure 4.12: Hotkeys list.

Iteration Info

The "Iteration Info" panel is where the user can navigate between the several dataset iterations and command to plot the desired ones. In Figure 4.13, it can be seen a dropdown list which contains the belonging iterations to the previously chosen dataset, and a "Plot" button.

For the following example will be chosen the first iteration of the dataset, and consequently the "Plot" button will be pressed for data visualization.

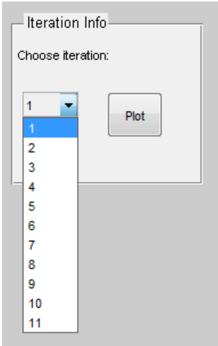


Figure 4.13: Iteration Info panel.

4.2.2.2 Data Visualization and Analysis

Clicking the "Plot" button, the data will be plotted in the main interface, as shown in Figure 4.14.

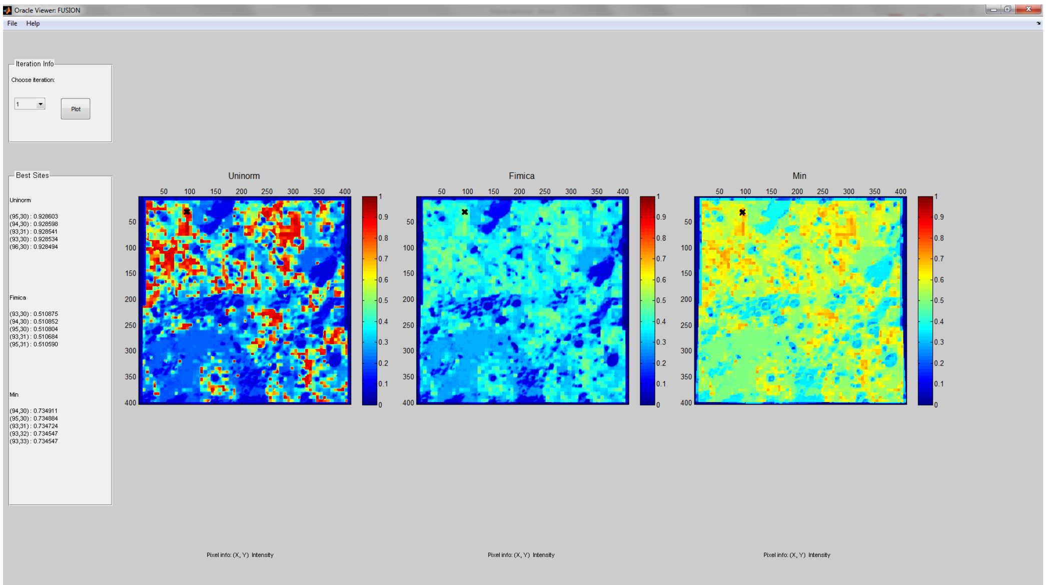


Figure 4.14: Main interface for iteration data visualization and analysis.

Three images have been plotted, one by each aggregation method (Uninorm, Fimica and Min) used in Oracle's site evaluation. It was also created a panel with the top five "Best Sites" for each aggregation method, and a pixel information that retrieves the mouse position in each image with the respective pixel rating. The colormap for each map is a default set of colors, which in this case represent the safety rating of a pixel.

Best Sites

The "Best Sites" panel can be seen in detail in Figure 4.15. The top five best sites (highest rated pixels) of each aggregation method are marked in the correspondent aggregation image with a cross at the designated coordinate of the best site.

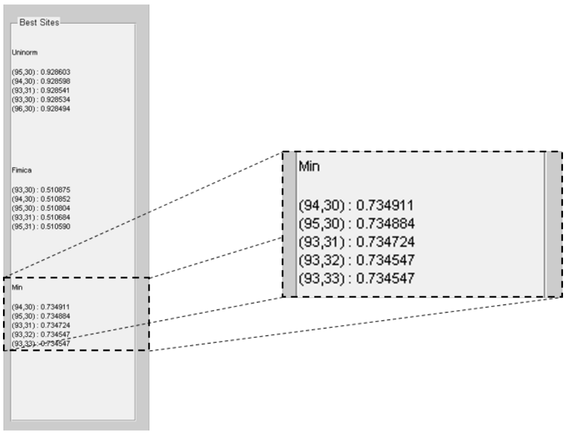


Figure 4.15: Best Sites panel (Min example).

Taking as an example a Min's best site such as $(94, 30)$: 0.734911, the best site's structure of pixel information is established as ($coordinate\ x$, $coordinate\ y$), being x and y the coordinates of the data matrix (Oracle's aggregation map), followed by site's rating, between 0 and 1.

4.2.2.3 Good Landing Site Example

For the next step of Oracle's aggregation method output data analysis, will be considered the previously chosen site, Min's best site $(94, 30)$. In the Oracle Viewer each image displayed has a pixel information panel ("Pixel info"), showing pixel coordinates and rating while the user hovers the mouse over the image. This panel is located bellow the image, as seen in Figure 4.16.

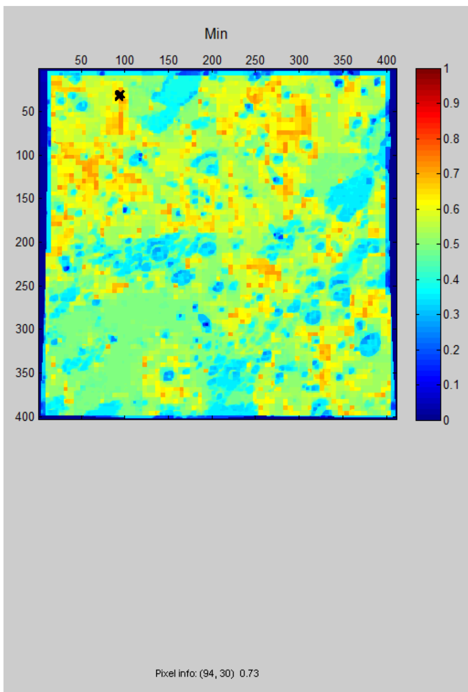


Figure 4.16: Min aggregation method image detail.

The gap between the image and the panel is derived from the significant map size variations per iteration that ILUV's dataset presents. Using the "hotkeys" available in the GUI, the user can zoom in (mouse scroll or "+" key as in Figure 4.12) the image for more detailed data and, in this case, for best site analysis. Then, zooming in the image in the "best sites area", Figure 4.17 is obtained.

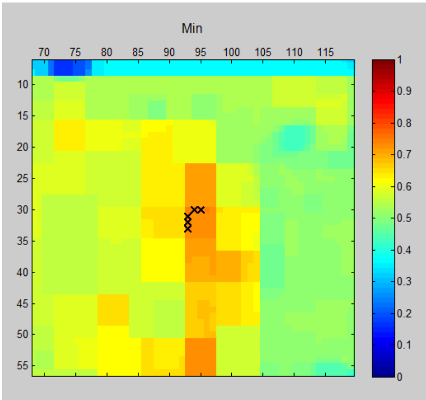
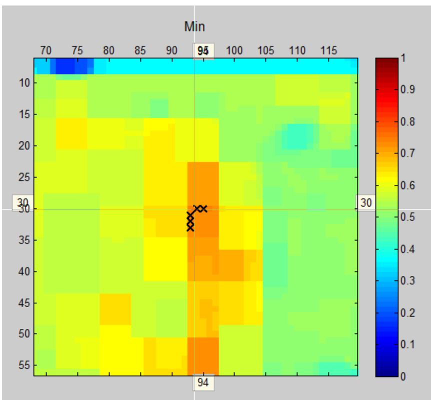
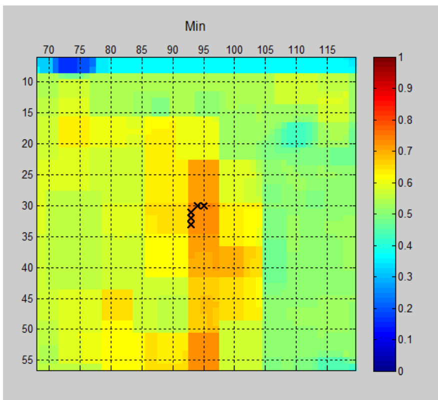


Figure 4.17: Min aggregation method image (zoomed in).

This way the user can achieve a better perception of pixels rating in the best sites' surroundings. Other available "hotkeys" can be used for different purposes, like the crosshair hotkey ("c" key) used to locate the desired best site with further precision, as in Figure 4.18(a), or even use a grid ("g" key), as shown in Figure 4.18(b)



(a) Crosshair functionality.



(b) Grid functionality.

Figure 4.18: Min aggregation method image detail (*crosshair* and *grid on*).

Another available option is the magnifier ("m" key) which can even be used, for instance, with zoom in and grid functionalities. The Figure 4.19 is an example of the visualization that the user can get with this option.

As previously stated, the region of interest chosen was the predefined region with (3×3) pixels. This region is used in case the user wishes to visualize, in a desired region, the correspondent pixel values of the hazard maps that were inputs for the IPSIS algorithm. This enables a better

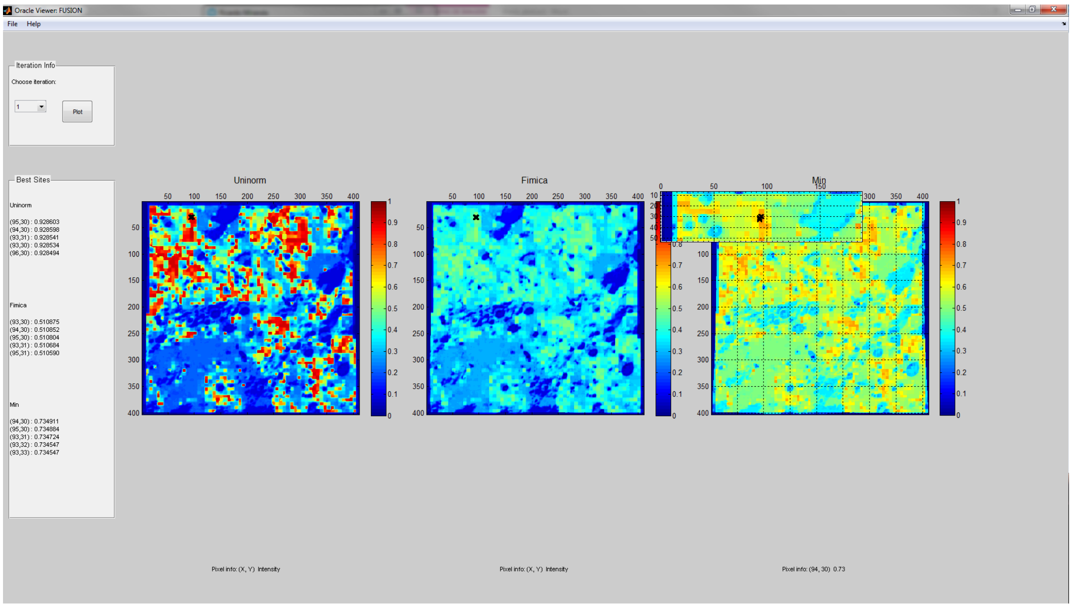


Figure 4.19: Min aggregation method image detail (magnifier with zoom in and grid).

perception why a pixel was rated with a certain value. For this to happen, the user must click in the image desired region to analyze, as in Figure 4.20.

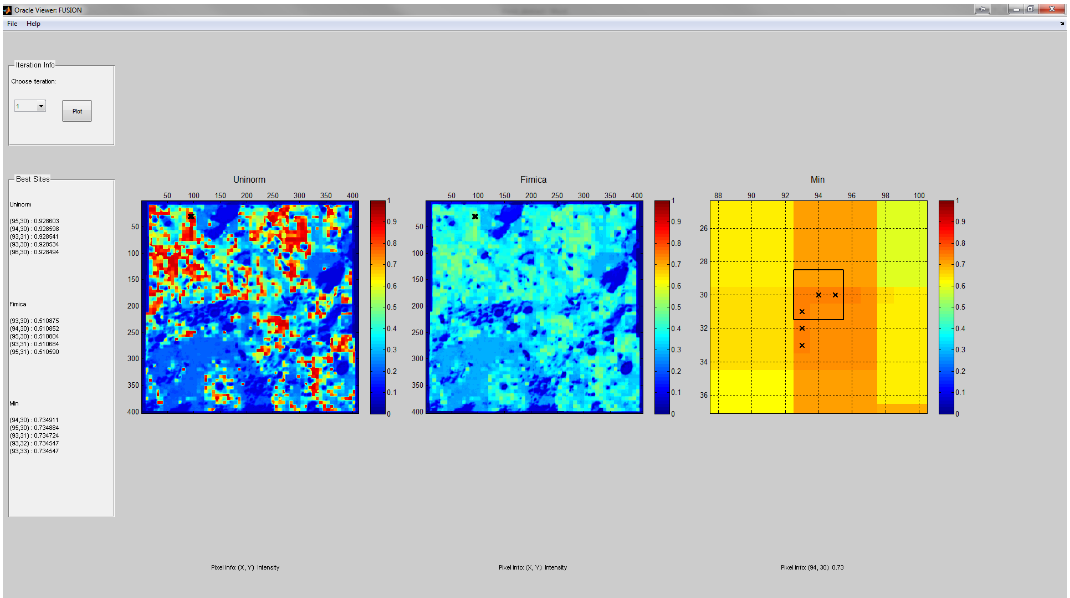


Figure 4.20: Min aggregation method image detail (region selection).

With a region selected, a new interface is opened (see Figure 4.21) showing the hazard maps that are available in the dataset. In this case, a *Shadow*, a *Texture*, a *Roughness* and a *Slope* map are plotted with the respective landing region size, the region's central pixel coordinates are shown on the top left side of the window, and the aggregation method chosen on the top right. The colormaps are dimensioned according to the membership functions of each hazard map.

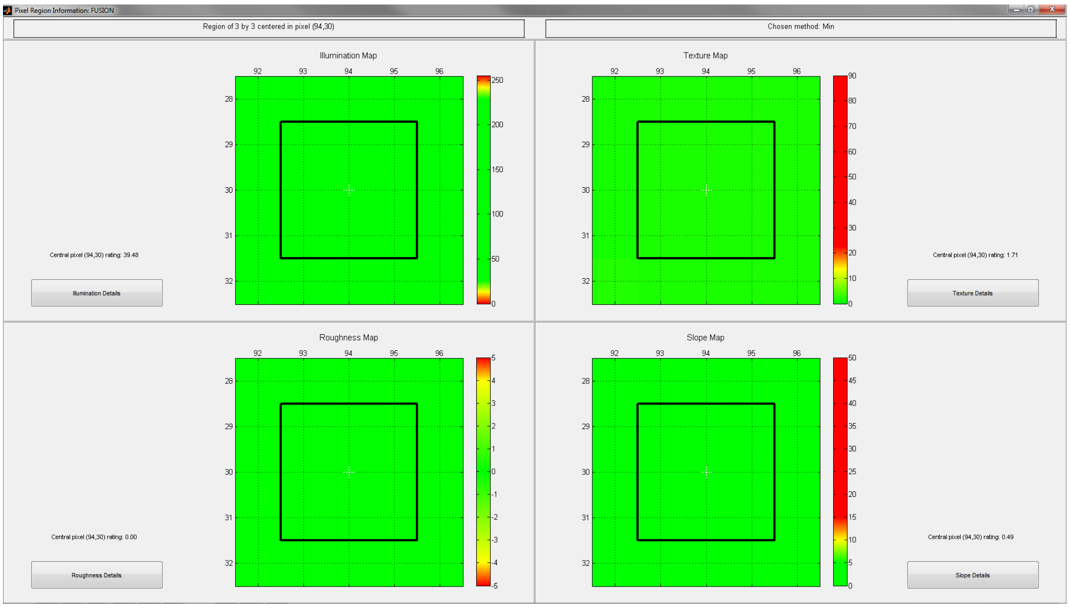


Figure 4.21: Pixel region information of $(94, 30)$ site.

Each hazard map has the region's central pixel value displayed, and a button for further map detailed information. A close up example for *Slope* map is shown in Figure 4.22. A last level of detail can be achieved clicking the designated "*Map* Details" button, in this case, "Slope Details". Here the user is faced with the value of each pixel printed in the hazard map's image, Figure 4.23.

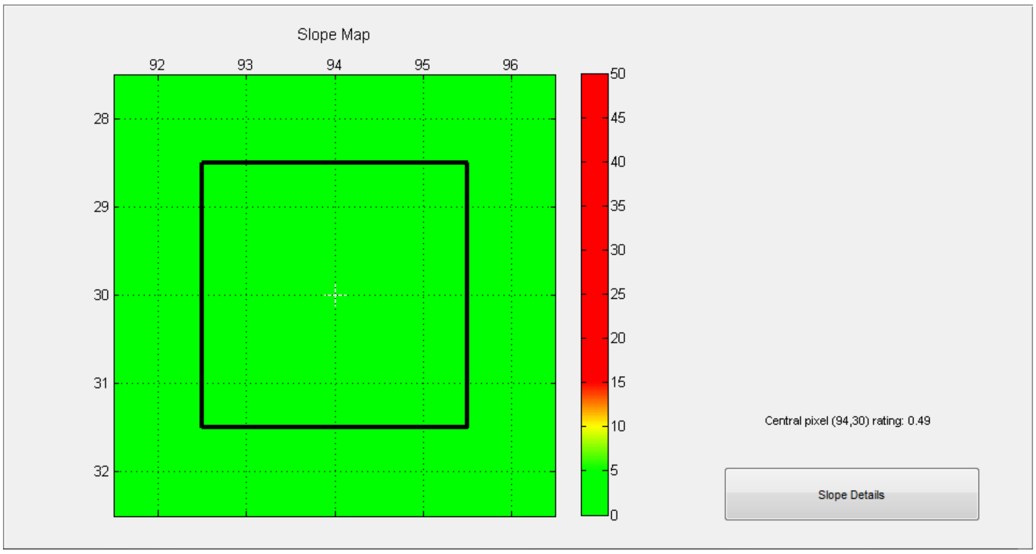


Figure 4.22: Pixel region information of $(94, 30)$ site: Slope map info.

In this case, the slope's pixels have the value of 0.49, which according to Table 4.5 (in Section 4.1.2.4) are in FLAT fuzzy set, a good slope for a spacecraft landing, therefore the color green.

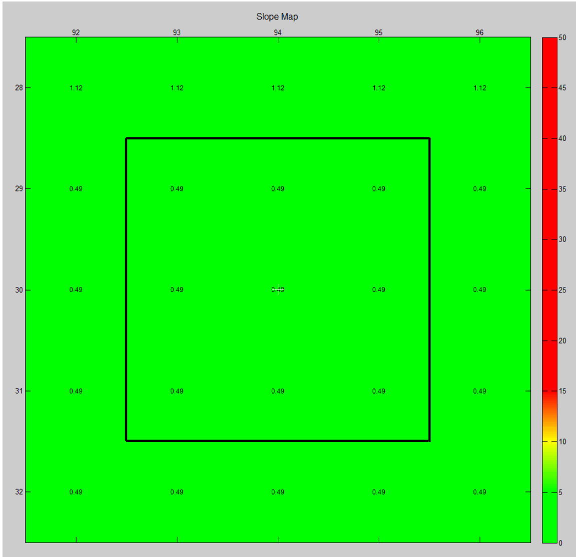


Figure 4.23: Pixel region information of $(94, 30)$ site: Slope details (pixel value).

4.2.2.4 Bad Landing Site Example

As seen in Section 4.2.2.3, the pixel region information (in Figure 4.21) of a good site normally have a homogeneous green color representing the "safeness" of the region, i.e. each specific pixel is between the membership functions' desired range. However, this is not always the case. For instance, selecting the pixel $(118, 100)$, with a rating approximately equal to 0.19, its pixel region information is significantly different than the shown in Figure 4.21 for a good case.

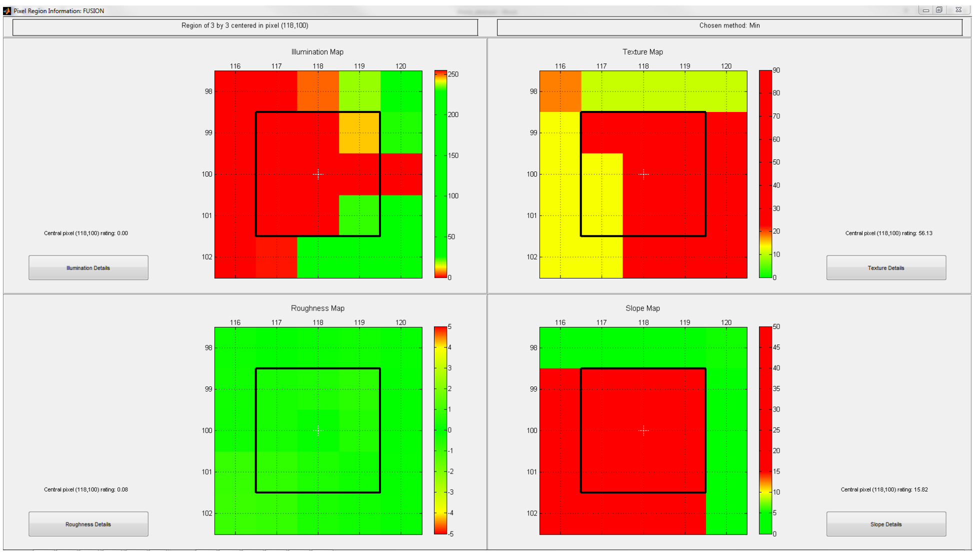


Figure 4.24: Pixel region information of $(118, 100)$ site.

In Figure 4.24 is shown a heterogeneity in the results, alerting the user for a bad landing region, particularly in terms of shadow, texture and slope.

Taking again the case of slope, by Figure 4.25, the selected region has a prohibitive slope of 16

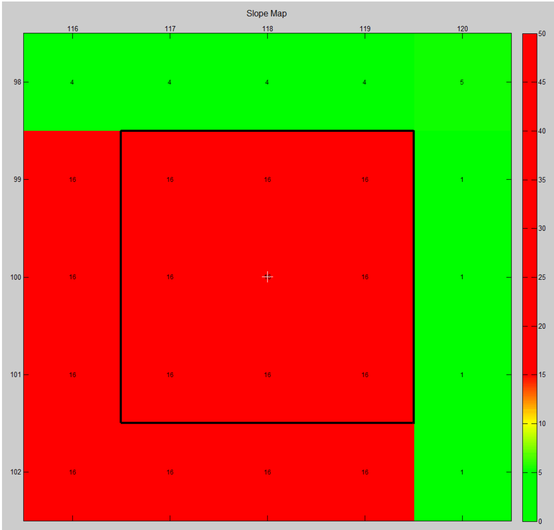


Figure 4.25: Pixel region information of $(118, 100)$ site: Slope details (pixel value).

degrees that could compromise the landing mission.

4.2.2.5 Special Functionality

The description in the preceding examples for FUSION project, also serves for the cases in ILUV's project scope. The exception occurs for a single hazard map in the "*Map* Details" presentation. A "Reachability Map" is taken into account where "Reachability Details" instead of displaying the value of each pixel in and around the selected region on the map, shows the direction relative to each pixel (wind consideration for an UAV landing).

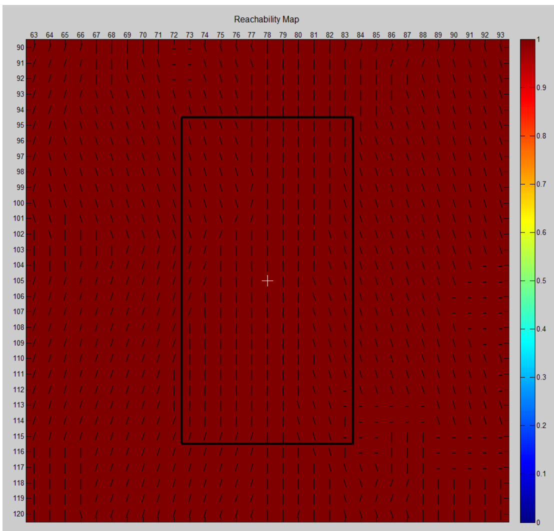


Figure 4.26: Pixel region information: ILUV Reachability details (pixel direction).

Figure 4.26 shows that besides the region being good for landing (dark red color), every pixel have a dash oriented with the best landing direction.

5

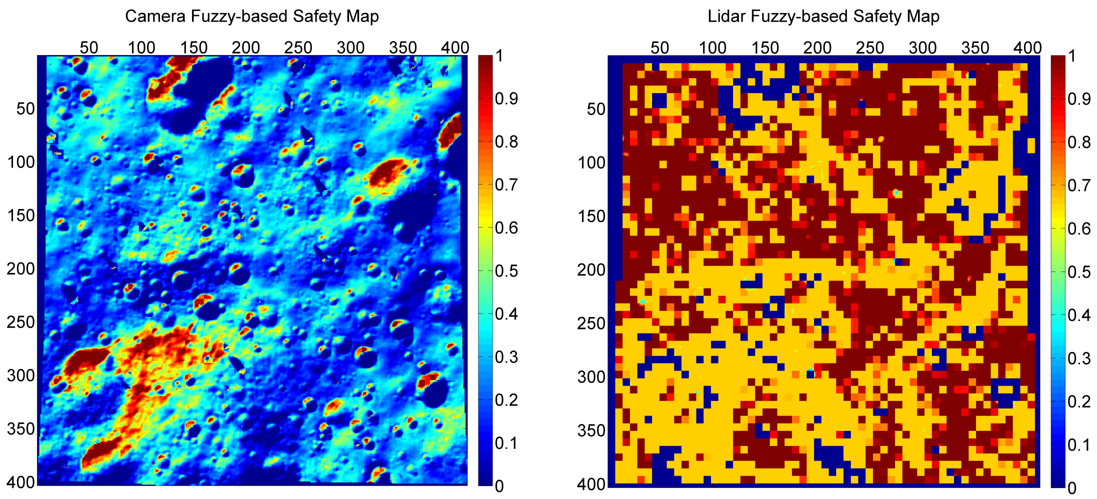
Results and Discussion

Contents

In this chapter are discussed the results obtained with the FRA (unique pixel site), by comparing its best site to the one obtained by IPSIS (considering region aggregation). The computational requirements of both algorithms are also considered, taking into account that EDL phases of a spacecraft occur during brief moments, typically on the order of 1-2 minutes. Therefore, computational requirements and execution times of any algorithm used for terrain analysis is of utmost importance (Serrano et al., 2006).

5.1 Execution Example

The execution example for the FRA was done with the same test dataset used in the IPSIS example. The output generated from the FRA is, as shown in Figure 4.2 of Section 4.1.1 (in green), a file with a camera fuzzy-based result, a lidar fuzzy-based result and a final fuzzy-based output result of the weighted average of both sensor results.



(a) Camera fuzzy-based safety map.

(b) Lidar fuzzy-based safety map.

Figure 5.1: Sensor fuzzy-based safety maps.

Figure 5.1(a) and Figure 5.1(b) are the result of the camera and lidar fuzzy-based safety maps respectively. The x and y axes of the figures define the map's size, while the colormap encompasses the safety values of the pixels. Combining these two sensor fuzzy-based safety maps with the Takagi-Sugeno method proposed in Section 4.1, it is obtained the final fuzzy-based safety map, which is in practice, the desired output of the fuzzy reasoning assessment of the input dataset. Therefore, the output result of both FRA and IPSIS are depicted in Figure 5.2. The best result from the FRA corresponds to the coordinate $(165, 342)$, which was verified analyzing each cell of the output file matrix of results. The IPSIS selected best site, result of Oracle's exhaustive search using the Uninorm operator, is depicted in the top-left area of the map, coordinate $(95, 30)$.

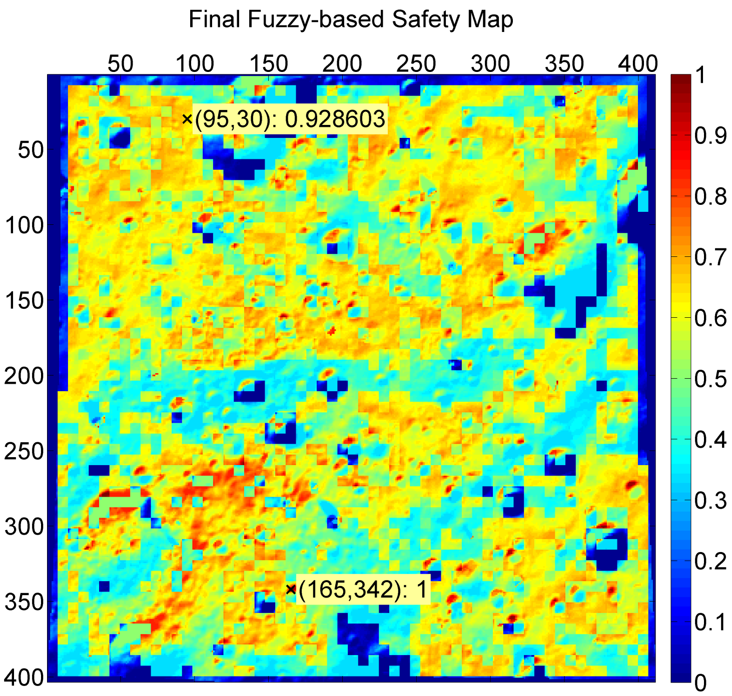


Figure 5.2: Final fuzzy-based safety map output.

Making use of the Oracle Viewer tool zooming option, it was possible to analyze the details of the IPSIS best site, as shown in Figure 5.3. The top five best rated sites are represented with a "x" symbol.

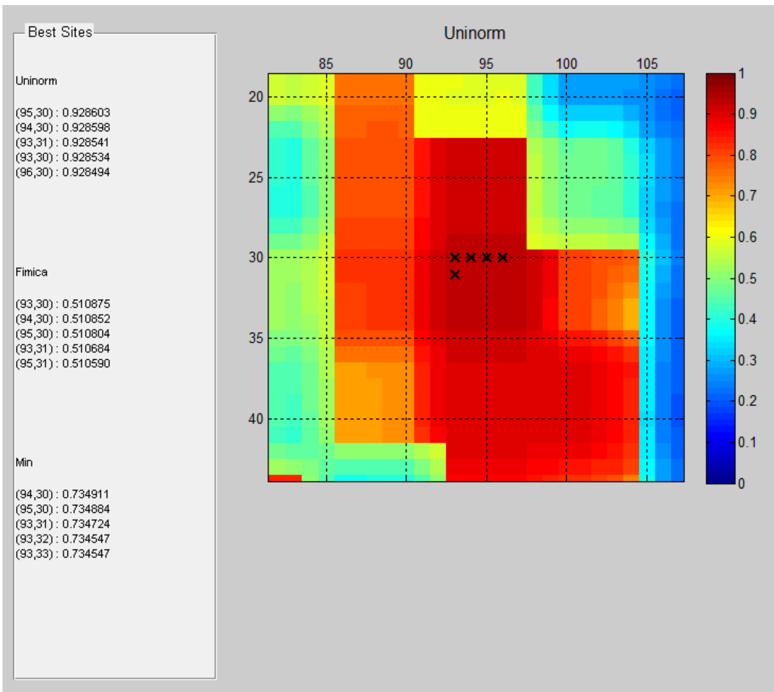


Figure 5.3: Oracle's exhaustive search best site.

As verified, the results of FRA and IPSIS are completely different, standing almost at opposite ends of the map. However, it is necessary to refer that IPSIS Oracle's exhaustive search output maps do not takes into account the fuzzy processing realized in the FRA. The IPSIS best site result corresponds, as stated previously, to Uninorm aggregation method (map detailed in Figure 5.3). Due to the disparity between output results, these will be further explored.

To analyze in detail the best site given by both algorithms, Table 5.1 lists the input values of each hazard map for each selected site. Observing the respective values of the pixels in each hazard map, it is obvious that IPSIS results are, in general, better since they are close to the optimal values.

Table 5.1: Algorithm's best site details.

Algorithm	**Best site coordinate**	**Hazard map values**			
		Shadow	Texture	Roughness	Slope
FRA	$(165,\ 342)$	75.17	8.80	-0.23	2.49
IPSIS	$(95,\ 30)$	39.18	1.71	0.00	0.49

FRA Membership function desired range	GOOD $[15 \quad 245]$	VLOW $[0 \quad 30]$	SMOOTH $[-2 \quad 2]$	FLAT $[0 \quad 5]$
Optimal value	127.5	0	0	0

Further, as seen in Table 5.1, the best sites determined by the algorithms are, differentiating by hazard map, within the desired ranges defined by the FRA membership functions. However, even considering the FRA membership functions, the selected site chosen by IPSIS display three out of four hazard map values near the optimal. Such conclusion can be made since every feature has the same weight in the algorithms. The main reason for this disparity in the results is that the FRA is less precise than IPSIS since the memberships are totally predefined, and as long as the raw values are within the memberships they are acceptable. In the IPSIS algorithm, the memberships are adapted for each iteration following the logic "lower is better" and using a predefined topology, which makes this algorithm much more versatile and robust. Moreover, IPSIS goal is to select the best site for landing while the FRA goal is just to assess the safety of the terrain, which this latter accomplishes but does not prove a "optimum" location.

5.2 Computational Requirements

In this subsection, the previously considered dataset of eleven iterations provided by Spin.Works, with four distinct hazard maps of variable sizes per iteration, serves as the basis of comparison of the computational requirements of each algorithm. It is taken into account for this comparison that IPSIS serves FUSION project purposes as a tool for HDA with safe site selection (using PSO non-exhaustive site selection, which is the method applied for best result/execution time trade-off) and recommending piloting maneuvers, while FRA goal is only to assess the sites and therefore, has a smaller task in HDA process.

The obtained results are the outcome of a series of tests performed on an Intel® Core™ i5 M430 CPU @ 2.27GHz, with 4GB's of RAM DDR3 @ 1066 MHz, on a Linux Mint 15 Cinnamon (32-bit), 3.8.0-26-generic. In Tables 5.2 and 5.3, it is perfectly noticeable the difference in execution times between the two algorithms due the fact IPSIS is using PSO search methodology, which only analyze 1% of all map pixels. Therefore, the execution time of IPSIS is two orders of magnitude lower than the FRA. Theoretically, it is possible to assume that if the FRA only analyzed the same amount of pixels, i.e. 1% of all, the execution times would be quite similar (FRA - $0.01 \times 3851 \approx 39\ ms$; IPSIS - $35\ ms$). It is hard to make a fair comparison because IPSIS is considerably optimized and its objectives are wider in scope, considering that it provides a near-optimal place for landing, but certainly is a more efficient algorithm.

Table 5.2: FRA execution time.

FRA		
Test	**Analyzed pixels**	**Execution time (ms)**
1		3861
2		3834
3	1.737.374	3848
4		3833
5		3881
Average Time		3851

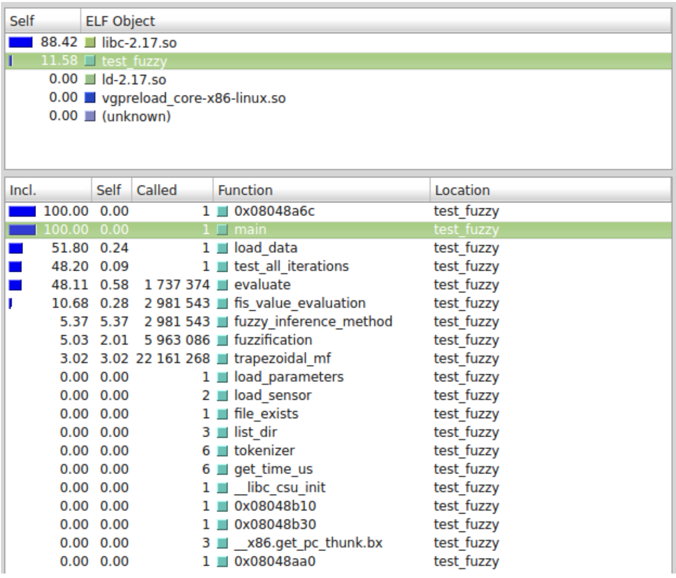
Table 5.3: IPSIS execution time.

IPSIS		
Test	**Analyzed pixels**	**Execution time (ms)**
1		38
2		34
3	$0,01 \times 1.737.374 \approx 17.374$	34
4		32
5		38
Average Time		35

For computational cost calculation strategy was also used a profiling tool called Callgrind (Weidendorfer, 1998) to evaluate the execution times/percentages of the test execution. Considering the total execution time, and the percentage of time taken for a particular function, it is possible to calculate the time spent on each function. Again, note that the following test refers to the execution of all eleven iterations.

According to Figures 5.4 and 5.5, the "Incl." ("Inclusive") column shows a percentage of cost for the function and its successive function calls, while the "Self" column (sometimes referred to as

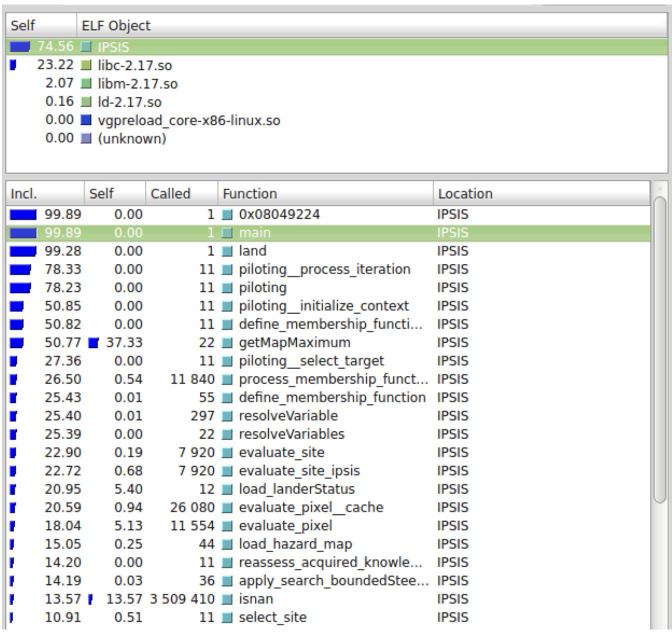
"Exclusive"), is the cost of just the function itself. Therefore, for instance, for `main` function the "Incl." column will present nearly 100%, whereas the "Self" cost is negligible taking into account the other functions where the actual calculations are done. Is easily verified that the degree of complexity of both algorithms is considerably higher in IPSIS by the amount of existing functions.



Self	ELF Object
88.42	libc-2.17.so
11.58	test_fuzzy
0.00	ld-2.17.so
0.00	vgpreload_core-x86-linux.so
0.00	(unknown)

Incl.	Self	Called	Function	Location
100.00	0.00	1	0x08048a6c	test_fuzzy
100.00	0.00	1	main	test_fuzzy
51.80	0.24	1	load_data	test_fuzzy
48.20	0.09	1	test_all_iterations	test_fuzzy
48.11	0.58	1 737 374	evaluate	test_fuzzy
10.68	0.28	2 981 543	fis_value_evaluation	test_fuzzy
5.37	5.37	2 981 543	fuzzy_inference_method	test_fuzzy
5.03	2.01	5 963 086	fuzzification	test_fuzzy
3.02	3.02	22 161 268	trapezoidal_mf	test_fuzzy
0.00	0.00	1	load_parameters	test_fuzzy
0.00	0.00	2	load_sensor	test_fuzzy
0.00	0.00	1	file_exists	test_fuzzy
0.00	0.00	3	list_dir	test_fuzzy
0.00	0.00	6	tokenizer	test_fuzzy
0.00	0.00	6	get_time_us	test_fuzzy
0.00	0.00	1	__libc_csu_init	test_fuzzy
0.00	0.00	1	0x08048b10	test_fuzzy
0.00	0.00	1	0x08048b30	test_fuzzy
0.00	0.00	3	__x86.get_pc_thunk.bx	test_fuzzy
0.00	0.00	1	0x08048aa0	test_fuzzy

Figure 5.4: CallGrind output for FRA (test_fuzzy).



Self	ELF Object
74.56	IPSIS
23.22	libc-2.17.so
2.07	libm-2.17.so
0.16	ld-2.17.so
0.00	vgpreload_core-x86-linux.so
0.00	(unknown)

Incl.	Self	Called	Function	Location
99.89	0.00	1	0x08049224	IPSIS
99.89	0.00	1	main	IPSIS
99.28	0.00	1	land	IPSIS
78.33	0.00	11	piloting__process_iteration	IPSIS
78.23	0.00	11	piloting	IPSIS
50.85	0.00	11	piloting__initialize_context	IPSIS
50.82	0.00	11	define_membership_functi...	IPSIS
50.77	37.33	22	getMapMaximum	IPSIS
27.36	0.00	11	piloting__select_target	IPSIS
26.50	0.54	11 840	process_membership_funct...	IPSIS
25.43	0.01	55	define_membership_function	IPSIS
25.40	0.01	297	resolveVariable	IPSIS
25.39	0.00	22	resolveVariables	IPSIS
22.90	0.19	7 920	evaluate_site	IPSIS
22.72	0.68	7 920	evaluate_site_ipsis	IPSIS
20.95	5.40	12	load_landerStatus	IPSIS
20.59	0.94	26 080	evaluate_pixel__cache	IPSIS
18.04	5.13	11 554	evaluate_pixel	IPSIS
15.05	0.25	44	load_hazard_map	IPSIS
14.20	0.00	11	reassess_acquired_knowle...	IPSIS
14.19	0.03	36	apply_search_boundedStee...	IPSIS
13.57	13.57	3 509 410	isnan	IPSIS
10.91	0.51	11	select_site	IPSIS

Figure 5.5: CallGrind output for IPSIS.

6
Conclusions

Contents

In this chapter are presented the main conclusions regarding the elaboration of this work. Firstly, it was made a brief overview of the state of the art in data fusion area, encompassing the main characteristics of this process. After it, the problem to be solved was contextualized in a HDA perspective, and posteriorly was demonstrated a proposal of a fuzzy reasoning algorithm, presented as FRA, for terrain assessment. Finally, a data visualization and analysis tool, named Oracle Viewer, for expert decision-making was presented.

6.1 Conclusions

The applicability of fuzzy reasoning in a environment of data fusion, applied as a module of terrain assessment for HDA purposes, was successfully verified. The advantages of this implementation turn out to be the same of a fuzzy system: high semantic interpretation, handles uncertainty and imprecise information, and is a good substitute for a mathematical model. Comparatively to the result held by IPSIS algorithm, and although FRA execution time is in theory marginally superior to IPSIS execution (PSO non-exhaustive search analyze 1% of map pixels), it can assess adequately the safety of landing sites. Is also important to note that IPSIS has a wider goal and was optimized to become an extremely efficient solution for safe site selection.

Regarding the Oracle Viewer tool, it proved to be a very powerful tool for visualizing and analyzing IPSIS results, and for detailed information analysis of the input dataset maps. It also facilitates the expert decision-maker task into landing site assessment and validation.

6.2 Future Work

Future work will be adapting the FRA into a more dynamic and scalable algorithm with different FIS approaches, such as Mamdani inference, and more differentiating rules and memberships. With all this changes it is also expected an improvement in the execution time. Moreover, could be interesting to include fuzzy reasoning in IPSIS.

Considering the Oracle Viewer, it can be made the transition from a MATLAB® environment tool to an independent software application. Adapting and extending this tool to be used in distinct scenarios would also be rather interesting for supporting decision makers by providing a sophisticated visualization analytic tool.

Bibliography

Agency, I. A. E. (2007). *Implementation Strategies and Tools for Condition Based Maintenance at Nuclear Power Plants*. IAEA-TECDOC. International Atomic Energy Agency.

Aguilera, C. A. V. (2009). COLORMAP and COLORBAR utilities - File Exchange - MATLAB Central. http://www.mathworks.com/matlabcentral/fileexchange/24371-colormap-and-colorbar-utilities-feb-2014.

Arai, T., Pagello, E., and Parker, L. (2002). Guest editorial: Advances in multirobot systems. *Robotics and Automation, IEEE Transactions on*, 18(5):655–661.

Ardeshir Goshtasby, A. and Nikolov, S. (2007). Guest Editorial: Image Fusion: Advances in the State of the Art. *Inf. Fusion*, 8(2):114–118.

Azorean (2014). Ziphius: Zillions of fun. http://myziphius.com/. Accessed: 2014-01-31.

Babuska, R. (2003). Fuzzy Systems, Modeling and Identification.

Bachmann, C. (2011). Multi-sensor Data Fusion for Traffic Speed and Travel Time Estimation. Master's thesis, University of Toronto.

Barth, M. and Boriboonsomsin, K. (2012). ECO-ITS: Intelligent Transportation System Applications to Improve Environmental Performance. Technical report, University of California.

Bawden, A., Greenblatt, R., Holloway, J., Knight, T., Moon, D., and Weinreb, D. (1977). *LISP Machine Progress Report*.

Bedworth, M. and O'Brien, J. (2000). The Omnibus model: a new model of data fusion? *Aerospace and Electronic Systems Magazine, IEEE*, 15(4):30–36.

Bellenger, A. (2013). *Semantic Decision Support for Information Fusion Applications*. PhD thesis, Institut National des Sciences Appliquées de Rouen.

Biermann, J., de Chantal, L., Korsnes, R., Rohmer, J., and Ündeger, c. (2004). From Unstructured to Structured Information in Military Intelligence - Some Steps to Improve Information Fusion. In *RTO SCI Symposium on Systems, Concepts and Integration (SCI) Methods and Technologies for Defence Against Terrorism*. RTO-MP-SCI-158.

Blasch, E., Llinas, J., Lambert, D., Valin, P., Das, S., Chong, C., Kokar, M., and Shahbazian, E. (2010). High Level Information Fusion developments, issues, and grand challenges: Fusion 2010 panel discussion. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8.

Blasch, E. P., Lakhotia, A., and Seetharaman, G. (2006). Unmanned vehicles come of age: The DARPA grand challenge. *Computer*, 39(12):26–29.

Bleiholder, J. and Naumann, F. (2009). Data Fusion. *ACM Comput. Surv.*, 41(1):1:1–1:41.

Board, U. S. A. (1996). *New World Vistas: Air and Space Power for the 21st Century : Sensors Volume*, chapter 3.0 The Sensor Correlation and Fusion Process. USAF Scientific Advisory Board.

Borotschnig, H., Paletta, L., Prantl, M., and Pinz, A. (1999). A Comparison of Probabilistic, Possibilistic and Evidence Theoretic Fusion Schemes for Active Object Recognition.

Bourdarias, C., Da-Cunha, P., Drai, R., Simões, L. F., and Ribeiro, R. A. (2010). Optimized and flexible multi-criteria decision making for hazard avoidance. In *33rd Annual AAS Rocky Mountain Guidance and Control Conference*. American Astronautical Society.

Brooks, R. and Iyengar, S. (1998). *Multi-Sensor Fusion: Fundamentals and Applications With Software*. National instruments virtual instrumentation series. Prentice Hall PTR.

CA3 (2013a). CA3 - Computational Intelligence Research Group. `http://www.ca3-uninova.org`. Accessed: 2014-01-31.

CA3 (2013b). FUSION - Sensor Data Fusion for Hazard Mapping and Piloting. `http://www.ca3-uninova.org/project_fusion`. Accessed: 2014-01-31.

CA3 (2013c). ILUV - Intelligent Landing of Unmanned aerial Vehicles. `http://www.ca3-uninova.org/project_iluv`. Accessed: 2014-01-31.

CA3 (2013d). IPSIS - Intelligent Planetary Site Selection - Past Research Projects. `http://www.ca3-uninova.org/project_ipsis`. Accessed: 2014-01-31.

Castanedo, F. (2013). A Review of Data Fusion Techniques. *The Scientific World Journal*, 2013.

Ciocarlie, M., Goldfeder, C., and Allen, P. (2007). Dimensionality reduction for hand-independent dexterous robotic grasping. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3270–3275.

Ciocarlie, M. T. and Allen, P. K. (2009). Hand Posture Subspaces for Dexterous Robotic Grasping. *The International Journal of Robotics Research*, 28(7):851–867.

Cui, J., Tosunoglu, S., Roberts, R., Moore, C., and Repperger, D. W. (2003). A review of teleoperation system control. In *Proceedings of the 2006 Florida conference recent advances in robotics (FCRAR), Florida Atlantic University, FL*.

Dailey, Daniel J., H. P. and Lin, P.-J. (1996). ITS Data Fusion. Technical report, University of Washington.

Darms, M., Foelster, F., Schmidt, J., Froehlich, D., and Eckert, A. (2010). Data Fusion Strategies in Advanced Driver Assistance Systems. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 3(2):176–182.

Dasarathy, B. (2001). Information fusion - what, where, why, when, and how? *Information Fusion*, 2(2):75–76.

de Fátima Bento, M. (2008). Unmanned Aerial Vehicles: An Overview. InsideGNSS.

Dempster, A. P. (2008). A Generalization of Bayesian Inference. In Yager, R. and Liu, L., editors, *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 73–104. Springer Berlin Heidelberg.

Dictionaries, O. (2014a). Definition of disparate in english. `http://www.oxforddictionaries.com/definition/english/disparate?q=disparate`.

Dictionaries, O. (2014b). Definition of probability in english. `http://www.oxforddictionaries.com/definition/english/probability?q=probability`.

Dictionaries, O. (2014c). Definition of real time in english. `http://www.oxforddictionaries.com/definition/english/real-time`.

Dictionaries, O. (2014d). Definition of spurious in english. `http://www.oxforddictionaries.com/definition/english/spurious?q=spurious`.

Dictionaries, O. (2014e). Definition of statistic in english. `http://www.oxforddictionaries.com/definition/english/statistic?q=statistic`.

Dong, X. L. and Naumann, F. (2009). Data Fusion: Resolving Data Conflicts for Integration. *Proc. VLDB Endow.*, 2(2):1654–1655.

Dragan, A., Lee, K., and Srinivasa, S. (2013). Teleoperation with Intelligent and Customizable Interfaces. *Journal of Human-Robot Interaction, Special Issue on "Technical and Social Advances in HRI: An Invitational Issue of JHRI"*, 1(3).

Dubois, D. and Prade, H. (1988). *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press.

Dubois, D. and Prade, H. (2000). Possibility theory in information fusion. In *Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on*, volume 1, pages 6–19.

Dubois, D. and Prade, H. (2010). *Formal Representations of Uncertainty*, pages 85–156. ISTE.

Duhaney, J., Khoshgoftaar, T. M., Cardei, I., Alhalabi, B., and Sloan, J. C. (2010). Applications of data fusion in monitoring inaccessible ocean machinery. *Proceedings, 16th International ISSAT on Reliability and Quality in Design*, 5:7.

Durrant-Whyte, H. and Henderson, T. C. (2008). Multisensor Data Fusion. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, pages 585–610. Springer Berlin Heidelberg.

Ellson, J., Gansner, E. R., Koutsofios, E., North, S. C., and Woodhull, G. (2001). Graphviz - Open Source Graph Drawing Tools. *Graph Drawing*, pages 483–484.

Elmenreich, W. (2007). A Review on System Architectures for Sensor Fusion Applications. In *Proceedings of the 5th IFIP WG 10.2 International Conference on Software Technologies for Embedded and Ubiquitous Systems*, SEUS'07, pages 547–559, Berlin, Heidelberg. Springer-Verlag.

ESA (2010). An artificial hand for robot arm. `http://www.esa.int/spaceinimages/Images/2010/06/An_artificial_hand_for_robot_arm`. Accessed: 2014-02-18.

ESA (2014a). ESA Science & Technology: Gaia. `http://sci.esa.int/gaia/`. Accessed: 2014-01-31.

ESA (2014b). ESA Science & Technology: Robotic Exploration. `http://exploration.esa.int/mars/`. Accessed: 2014-01-31.

ESA (2014c). What is remote sensing? `http://www.esa.int/SPECIALS/Eduspace_EN/SEMF9R3Z2OF_0.html`. Accessed: 2014-02-19.

Esteban, J., Starr, A., Willetts, R., Hannah, P., and Bryanston-Cross, P. (2005). A Review of data fusion models and architectures: towards engineering guidelines. *Neural Computing and Applications*, 14(4):273–281.

EU (2011). *Intelligent transport systems in action : action plan and legal framework for the deployment of intelligent transport systems (ITS) in Europe*. Luxemburg : Publ. Off. of the Europ. Union.

Fan, W. and Bifet, A. (2013). Mining Big Data: Current Status, and Forecast to the Future. *SIGKDD Explor. Newsl.*, 14(2):1–5.

Farinelli, A., Iocchi, L., and Nardi, D. (2004). Multirobot systems: a classification focused on coordination. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(5):2015–2028.

Filev, D., Chinnam, R., Tseng, F., and Baruah, P. (2010). An Industrial Strength Novelty Detection Framework for Autonomous Equipment Monitoring and Diagnostics. *Industrial Informatics, IEEE Transactions on*, 6(4):767–779.

Flandin, G. (2003). M2HTML: Documentation System for Matlab in HTML. `http://www.artefact.tk/software/matlab/m2html/`.

Florea, M. C., Jousselme, A.-L., Grenier, D., and Bosse, E. (2002). Unified approach to the fusion of imperfect data? In Dasarathy, B. V., editor, *Sensor Fusion: Architectures, Algorithms, and Applications VI*, volume 4731 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 75–85.

Fong, T. and Thorpe, C. (2001). Vehicle Teleoperation Interfaces. *Auton. Robots*, 11(1):9–18.

Geder, J., Ramamurti, R., Palmisano, J., Pruessner, M., Ratna, B., and Sandberg, W. (2009). Sensor data fusion and submerged test results of a pectoral fin propelled uuv. In *16th International Symposium on Unmanned Untethered Submersible Technology*, pages 23–26.

Gonçalves-Coelho, A., Veloso, L., and Lobo, V. (2007). Tests of a light UAV for naval surveillance. In *OCEANS 2007 - Europe*, pages 1–4.

Goodman, I., Mahler, R., and Nguyen, H. (1997). *Mathematics of Data Fusion*. NATO Asi Series. Springer.

Greenblatt, R. D., Knight, T. F., Holloway, J. T., and Moon, D. A. (1980). A LISP Machine. *SIGIR Forum*, 15(2):137–138.

Hall, D. and Llinas, J. (1997). An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23.

Hall, D., McNeese, M., Llinas, J., and Mullen, T. (2008). A framework for dynamic hard/soft fusion. In *Information Fusion, 2008 11th International Conference on*, pages 1–8.

Hall, D. L. and Garga, A. K. (1999). Pitfalls in Data Fusion (and How to Avoid Them). In *Proceedings of the 2nd International Conference on Information Fusion - FUSION'99*, volume 1, pages 429–436.

Hall, D. L. and Steinberg, A. N. (2000). Dirty Secrets in Multisensor Data Fusion. In *National Symposium on Sensor Data Fusion*, page 14.

Hormigo, T. and Araújo, J. (2013). A Micro-UAV System for Forest Management. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-1/W2:211–216.

Howard, A. and Seraji, H. (2004). Multi-sensor terrain classification for safe spacecraft landing. *Aerospace and Electronic Systems, IEEE Transactions on*, 40(4):1122–1131.

Hyder, A., Shahbazian, E., and Waltz, E. (2002). *Multisensor Fusion*. NATO science series: Mathematics, physics, and chemistry. Springer.

IEC, T. C. N. I. E. C. (1997). IEC 1131 - Programmable Controllers. Technical report.

Iversen, J. (2005). freezeColors / unfreezeColors - File Exchange - MATLAB Central. `http://www.mathworks.com/matlabcentral/fileexchange/7943-freezecolors-unfreezecolors`.

Johansson, R. (2003). Information Acquisition in Data Fusion Systems. Master's thesis, Kungliga Tekniska Högskolan.

Johnson, A. E. and Golombek, M. P. (2012). Lander Vision System for Safe and Precise Entry Descent and Landing. *LPI Contributions*.

Jones, B. and Howard, A. (2006). An imaging technique for safe spacecraft landing and autonomous hazard avoidance. In *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on*, pages 8 pp.–.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, (82 (Series D)):35–45.

Khaleghi, B., Khamis, A., Karray, F. O., and Razavi, S. N. (2013). Multisensor Data Fusion: A Review of the State-of-the-art. *Inf. Fusion*, 14(1):28–44.

Khaleghi, B., Razavi, S., Khamis, A., Karray, F., and Kamel, M. (2009). Multisensor data fusion: Antecedents and directions. In *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*, pages 1–6.

Koks, D. and Challa, S. (2003). An introduction to bayesian and dempster-shafer data fusion. Technical report, Defence Science and Tech Org.

Kroese, D., Taimre, T., and Botev, Z. (2013). *Handbook of Monte Carlo Methods*. Wiley Series in Probability and Statistics. Wiley.

Lee, H., Lee, B., Park, K., and Elmasri, R. (2010). Fusion Techniques for Reliable Information: A Survey. *JDCTA*, 4(2):74–88.

Lee, H., Park, K., Lee, B., Choi, J., and Elmasri, R. (2008). Issues in Data Fusion for Healthcare Monitoring. In *Proceedings of the 1st International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '08, pages 3:1–3:8, New York, NY, USA. ACM.

Liggins, M., Hall, D., and Llinas, J. (2008). *Handbook of Multisensor Data Fusion: Theory and Practice, Second Edition*. Electrical Engineering & Applied Signal Processing Series. Taylor & Francis.

Liggins, M. E. and Chang, K.-C. (2009). Distributed Fusion Architectures, Algorithms, and Performance within a Network Centric Architecture. In *Handbook of multisensor data fusion : theory and practice*. CRC Press, Boca Raton FL, 2nd ed. edition.

Llinas, J., Bowman, C., Rogova, G., Steinberg, A., Waltz, E., and White, F. (2004). Revisiting the JDL Data Fusion Model II. In *In P. Svensson and J. Schubert (Eds.), Proceedings of the Seventh International Conference on Information Fusion (FUSION 2004*, pages 1218–1230.

Lohr, S. (2012). The Age of Big Data. `http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html?pagewanted=all&_r=0#commentsContainer`.

Luo, R. and Kay, M. (1989). Multisensor integration and fusion in intelligent systems. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(5):901–931.

Luo, R., Yih, C.-C., and Su, K.-L. (2002). Multisensor fusion and integration: approaches, applications, and future research directions. *Sensors Journal, IEEE*, 2(2):107–119.

Luo, R. C. and Kay, M. G. (1988). Multisensor integration and fusion: Issues and approaches.

Macii, D., Boni, A., De Cecco, M., and Petri, D. (2008). Tutorial 14: multisensor data fusion. *Instrumentation Measurement Magazine, IEEE*, 11(3):24–33.

Malhotra, B. D. and Chaubey, A. (2003). Biosensors for clinical diagnostics industry. *Sensors and Actuators B: Chemical*, 91(1-3):117 – 127.

Márquez, F. P. G., Tobias, A. M., Pérez, J. M. P., and Papaelias, M. (2012). Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46(0):169 – 178.

Martin, M., Houghton, B., Craig, M., Tooley, R., Richard, M., and Saylor, S. (2007). Mission Design and Operations Considerations for NASA's Lunar Reconnaissance Orbiter.

MathWorks (2014). What is sugeno-type fuzzy inference? `http://www.mathworks.com/help/fuzzy/what-is-sugeno-type-fuzzy-inference.html`. Accessed: 2014-02-23.

Maybeck, P. S. (1979). *Stochastic Models: Estimation and Control*, volume 1 of *Mathematics in Science and Engineering*. Elsevier Science.

McCandless, D. (2010). TED talk: The beauty of data visualization. www.ted.com.

McLaughlin, S., Krishnamurthy, V., and Evans, R. J. (2004). Bayesian network model for data incest in a distributed sensor network. In *In Proceedings of the 7 th International Conference on Information Fusion, volume I*, pages 606–613.

Mitchell, H. B. (2007). *Multi-Sensor Data Fusion: An Introduction*. Springer Publishing Company, Incorporated, 1st edition.

Mojtahedzadeh, R. (2011). Draw a rectangle - File Exchange - MATLAB Central. http://www.mathworks.com/matlabcentral/fileexchange/33767-draw-a-rectangle.

Mora, A., Fonseca, J., and Ribeiro, R. (2013). Real-time image recovery using temporal image fusion. In *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pages 1–5.

Morgado, J. A. and de Sousa, J. T. B. (2009). O Programa de Investigação e Tecnologia em Veículos Aéreos Autónomos Não-Tripulados da Academia da Força Aérea. *Cadernos do IDN*.

Morgan, D. and Falkner, E. (2001). *Aerial Mapping: Methods and Applications, Second Edition*. Mapping Science. Taylor & Francis.

Moslem, B., Diab, M. O., Khalil, M., and Marque, C. (2012). Combining data fusion with multiresolution analysis for improving the classification accuracy of uterine EMG signals. *EURASIP J. Adv. Sig. Proc.*, 2012:167.

NASA (2014a). MSL | NASA. http://www.nasa.gov/mission_pages/msl/index.html. Accessed: 2014-01-31.

NASA (2014b). NASA Earth System Science Remote Sensors. https://earthdata.nasa.gov/data/standards-and-references/nasa-earth-system-science-remote-sensors. Accessed: 2014-02-19.

Navalgund, R. R. (2002). Remote Sensing. 2 - Sensors and Platforms. *Resonance*, 7(1):37–46.

Nehmzow, U. (2003). Quantitative analysis of robot environment interaction - towards scientific mobile robotics. *Robotics and Autonomous Systems*, 44(1):55 – 68. Best Papers of the Eurobot '01 Workshop.

Ng, G. (2003). *Intelligent systems: fusion, tracking and control*. CSI, control and signal/image processing series. Research Studies Press.

Nilsson, M. (2008). Mind the Gap : Human Decision Making and Information Fusion.

Nilsson, M. and Ziemke, T. (2007). Information Fusion: a decision support perspective. In *Information Fusion, 2007 10th International Conference on*, pages 1–8.

OReilly, P. (2013). Universal Data Fusion: Enabling Cost-effective US/Russia/NATO Cooperative Missile Defense.

Papadimitratos, P., La Fortelle, A., Evenssen, K., Brignolo, R., and Cosenza, S. (2009). Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *Communications Magazine, IEEE*, 47(11):84–95.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning About Data*. Mathematics and Its Applications. Soviet Series. Springer Netherlands.

Pfister, H. R. and Bohm, G. (2008). The multiplicity of emotions: a framework of emotional functions in decision-making. *Judgment and decision making*, 3(1):5–17.

Pr, E. (2010). DRAGZOOM - Drag and zoom tool - File Exchange - MATLAB Central. `http://www.mathworks.com/matlabcentral/fileexchange/29276-dragzoom-drag-and-zoom-tool`.

Rada-Vilela, J. (2014). fuzzylite: a fuzzy logic control library. `http://www.fuzzylite.com`.

Ribeiro, R. A. (2006). Fuzzy space monitoring and fault detection applications. *Journal of Decision Systems*, 15(2-3):267–286.

Ribeiro, R. A., Falcão, A., Mora, A., and Fonseca, J. M. (2014). FIF: A fuzzy information fusion algorithm based on multi-criteria decision making. *Knowledge-Based Systems*, 58(0):23 – 32. Intelligent Decision Support Making Tools and Techniques: {IDSMT}.

Ribeiro, R. A., Pais, T. C., and Simões, L. F. (2010). Benefits of Full-Reinforcement Operators for Spacecraft Target Landing. In Greco, S., Pereira, R. A. M., Squillante, M., Yager, R. R., and Kacprzyk, J., editors, *Preferences and Decisions*, volume 257 of *Studies in Fuzziness and Soft Computing*, pages 353–367. Springer.

Robertson, P., Röckl, M., and Angermann, M. (2010). Advances in multi-sensor data fusion for ubiquitous positioning: novel approaches for robust localization and mapping.

Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. Springer-Verlag New York, Inc., New York, NY, USA.

Ross, T. J. (2000). Membership Functions, Fuzzification and Defuzzification. In Szczepaniak, P., Lisboa, P., and Kacprzyk, J., editors, *Fuzzy Systems in Medicine*, volume 41 of *Studies in Fuzziness and Soft Computing*, pages 48–77. Physica-Verlag HD.

Savitz, S., Blickstein, I., Buryk, P., Button, R., and DeLuca, P. (2013). *U.S. Navy Employment Options for Unmanned Surface Vehicles (USVs)*. RAND Corporation.

Schoess, J. and Castore, G. (1988). A Distributed Sensor Architecture For Advanced Aerospace Systems.

Seco, J. and Verhaegen, F. (2013). *Monte Carlo techniques in radiation therapy*. Imaging in medical diagnosis and therapy. CRC Press, Boca Raton, FL.

Seraji, H. and Serrano, N. (2009). A Multisensor Decision Fusion System for Terrain Safety Assessment. *Robotics, IEEE Transactions on*, 25(1):99–108.

Serrano, N., Bajracharya, M., Howard, A., and Seraji, H. (2006). A novel tiered sensor fusion approach for terrain characterization and safe landing assessment. In *Aerospace Conference, 2006 IEEE*, pages 10–.

Shafer, G. (1976). *A mathematical theory of evidence*, volume 1. Princeton University Press Princeton.

Sidek, O. and Quadri, S. (2012). A review of data fusion models and systems. *International Journal of Image and Data Fusion*, 3(1):3–21.

Simões, L. F., Bourdarias, C., and Ribeiro, R. A. (2012). Real-Time Planetary Landing Site Selection - A Non-Exhaustive Approach. *Acta Futura*, 5:39–52.

Smets, P. (1997). Imperfect Information: Imprecision and Uncertainty. In Motro, A. and Smets, P., editors, *Uncertainty Management in Information Systems*, pages 225–254. Springer US.

Spin.Works (2014). Spin.Works - Unmanned Systems, Flight Control & Machine Vision, Defence & Space Systems. `http://www.spinworks.pt/`. Accessed: 2014-01-31.

Steinberg, A. N. and Bowman, C. L. (2004). Rethinking the JDL Data Fusion Levels.

Steinberg, A. N., Bowman, C. L., and White, F. E. (1999). Revisions to the JDL data fusion model. volume 3719, pages 430–441. SPIE.

TI, T. I. (2013). Needed: Crashless cars. `http://e2e.ti.com/blogs_/b/thinkinnovate/archive/2013/10/10/needed-crashless-cars.aspx`. Accessed: 2014-02-19.

Toolbox, M. F. L. (2010). *Release 7.9 (R2009b)*. The MathWorks Inc., Natick, Massachusetts, United States.

van der Geest, J. (2006). Replace.m - psychtoolbox-3-old-gitsvn-basedl. `https://github.com/neurodebian/psychtoolbox-3-old-gitsvn-based/blob/master/Psychtoolbox/PsychOneliners/Replace.m`.

van Heesch, D. (2012). Doxygen. Web Resource.

Veloso, M., Bentos, C., and Pereira, F. C. (2009). *Multi-Sensor Data Fusion on Intelligent Transport Systems*. MIT Portugal Transportation Systems Working Paper Series.

Waltz, E. and Llinas, J. (1990). *Multisensor Data Fusion*. Artech House Radar/Electronic Warfare Library. Artech House, Incorporated.

Wang, Z., Ziou, D., Armenakis, C., Li, D., and Li, Q. (2005). A comparative analysis of image fusion methods. *IEEE T. Geoscience and Remote Sensing*, 43(6):1391–1402.

Weidendorfer, J. (1998). Kcachegrind.

Welch, G. and Bishop, G. (1995). An Introduction to the Kalman Filter. Technical report, Chapel Hill, NC, USA.

Westenberger, A., Muntzinger, M., and Dietmayer, K. (2012). Filtering solution to the out-of-sequence measurement problem with colored and correlated noise. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1911–1918.

White, F. E. (1991). Data Fusion Lexicon. Technical report.

WMO (2010). The Global Satellite Observing System: a success story. `http://www.wmo.int/pages/publications/bulletin_en/archive/59_1_en/59_1_mohr_en.html`. Accessed: 2014-02-19.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338 – 353.

Zadeh, L. A. (1984). Book Review: A Mathematical Theory of Evidence. *AI Magazine*, 5(3):81–83.

Zadeh, L. A. (1999). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 100:9–34.

Zegras, C., Pereira, F., Amey, A., Veloso, M., Liu, L., Bento, C., and Biderman, A. (2008). Data Fusion for Travel Demand Management: State of the Practice and Prospects. In *Travel Demand Management Symposium (TDMó8)*.

Zexu, Z., Weidongm, W., and Pingyuan, C. (2010). A reliable algorithm of rock detection and avoidance for safe spacecraft landing. In *Systems and Control in Aeronautics and Astronautics (ISSCAA), 2010 3rd International Symposium on*, pages 1009–1013.