Context in Robotics and Information Fusion

Domenico D. Bloisi, Daniele Nardi, Francesco Riccio, and Francesco Trapani

Abstract Robotics systems need to be robust and adaptable to multiple operational conditions, in order to be deployable in different application domains. Contextual knowledge can be used for achieving greater flexibility and robustness in tackling the main tasks of a robot, namely mission execution, adaptability to environmental conditions and self-assessment of performance. In this chapter, we review the research work focusing on the acquisition, management, and deployment of contextual information in robotic systems. Our aim is to show that several uses of contextual knowledge (at different representational levels) have been proposed in the literature, regarding many tasks that are typically required for mobile robots. As a result of this survey, we analyze which notions and approaches are applicable to the design and implementation of architectures for Information Fusion. More specifically, we sketch an architectural framework which enables for an effective engineering of systems that use contextual knowledge, by including the acquisition, representation, and use of contextual information into a framework for Information Fusion.

Key words: Context-awareness, Autonomous robotics, Context-dependent information fusion

1 Introduction

The ability of quickly recognizing the context and acting accordingly to it is an highly desirable skill for the development of robotic and intelligent systems. Robotic systems need to be robust and adaptable to multiple operational conditions, in order to be deployable in different application domains. In fact, the use of contextual knowledge can be a key factor for achieving greater flexibility and robustness to

Domenico D. Bloisi, Daniele Nardi, Francesco Riccio, and Francesco Trapani
Department of Computer, Control, and Management Engineering, Sapienza University of Rome, via Ariosto 25, 00185 Rome, Italy. e-mail: ⟨bloisi|nardi|riccio⟩@dis.uniroma1.it

complete the required tasks. In this chapter, we survey several works about context in robotic systems, focusing on the acquisition, management, and deployment of contextual information. While there is a plethora of literature on the topic, we further refine the review to those concepts that can contribute creating a bridge between Information Fusion into robotic architectures.

There are two main ways to use context in robotics design. One is to use context holistically, i.e., by emphasizing its impact on the whole system. The approach we choose, instead, is to use context where needed, i.e., by analysing its influence on the single parts of the system. The explicit representation of knowledge about context in the design phase of a system aims at improving its performance, by dynamically tailoring the functions of the system modules to the specific features of the situation at hand. Indeed, a clear separation of contextual knowledge leads to a design methodology that supports the definition of small specialized system components rather than complex self-contained sub-systems.

Our aim is to analyze which notions and approaches, among the several uses of contextual knowledge (at different representational levels) that have been proposed in the literature, are applicable to the design and implementation of architectures for Information Fusion. More specifically, we sketch an architectural framework which enables for an effective design of systems that use contextual knowledge. As result, we formalize the acquisition, representation, and use of contextual information into a framework for Information Fusion.

The remainder of this chapter is organized as follows. Section 2 provides an overview about the use of context in robotics. In particular, a novel classification of existing methods, based on the context representation, is presented. In Section 3 a context-aware framework for Information Fusion applications is proposed and a context-based architecture for an application example is described in Section 4. Conclusions are drawn in Section 5.

2 Context in Robotics

Contextual knowledge can be defined in general as “the information that surrounds a situation of interest in the world” [1]. With specific reference to robotics, the interest for contextual knowledge is twofold [2]:

- Context is useful in the design and implementation of systems that are focused on cognition;
- The performance of robotic systems, as well as their scope of applicability, can be improved by formalizing different high-level features by means of context representation and contextual reasoning.

This section explores different methods and approaches for managing contextual information. First, we recall the taxonomy defined by Turner [3], then we propose a novel classification that groups existing approaches according to the methodologies used for managing context. Finally, we discuss the advantages of our categorization.

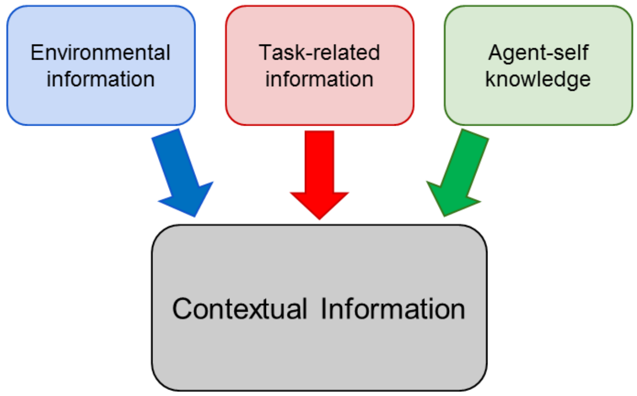

Fig. 1 Turner's context knowledge classification: (i) Environmental information, (ii) Task-related information, and (iii) Agent self-knowledge [3].

2.1 Contextual Knowledge

The identification and exploitation of contextual knowledge plays a key role in robotic systems. Indeed, a robotic system requires High-level Information Fusion capabilities [4], responsiveness, and an appropriate level of awareness about the external environment, its mission, and its own status. In the robotics domain, data fusion techniques have been widely exploited (e.g., [5, 6]), as well as cognitive level fusion (e.g., [7]); however, a common and standard definition of context does not exist, and, in general, the formalization of "context" depends on the actual implementation.

In this work, we adopt the Turner's categories [3] as the main reference for the formalization of context knowledge in robotics applications. Turner defines *context* as an identifiable configuration of features which are meaningful to characterize the world state and useful to influence the decision process of a robotic system. Moreover, he characterizes context information (CI) as a tuple of three elements, namely *Environmental Information* (EI), *Task-related Information* (TI) and *Self Knowledge* (SK). More specifically, Turner refers to contextual information as the sum of these three contributions (see Fig. 1).

Environmental knowledge. This kind of contextual information formalizes data that is environment-dependent and that does not directly depend on the robot actions. The robot perceives the world through its sensors and it infers the context according to the current status of the scenario (e.g., presence of obstacles or people). In a navigation system, the robot can tune its parameters depending on the terrain conditions, or its perception system, the information about the illumination conditions can be used to improve the perception or to discern the saliency of information as related to the task. In the case of a coordinated team of robots, e.g., unmanned aerial vehicles (UAVs) [8], having the task to search for a lost object, the robots may adapt their navigation parameters according to the detected conditions of the environment (e.g., terrain, trafficability, and constraints information).

Task-related knowledge. Task-related information is generally imposed by the mission specifications. Depending on the operating condition and on the task constraints (e.g., time constraints, priorities, and locations), the robot adapts its execution plan in order to increase robustness and efficiency. It is worth noting that the knowledge about a task does not modify the task outcome or requirements, but it is exploited to influence the execution of the task with the aim of improving the performance. Using again the example of multiple robot search, the team of robots can execute the task in different modalities by considering: (i) The current day time; (ii) The location where the robots are searching for the objects; (iii) The information processed by the other teammates; (iv) The known locations where a particular objects uses to be; (v) Additional information gathered during the search (e.g., the robot can receive information about where the object of interest has been detected the last time).

In the above discussed example, the contextual knowledge does not modify the goal of the task (which remains the localization of an object), but it drastically influences the task execution (e.g., ordering) such as sensor and mission management, and thus, the performance of the system (e.g., timeliness, accuracy).

Self knowledge. In this case the robot infers context knowledge by relying on its own status and on the internal representation of the surrounding environment. In the multiple robot search example, for instance, it is possible that one of the teammates recognizes a self malfunctioning or has a low battery level. Then, it can communicate its status to the team. Consequently, the team can consider unreliable the information coming from that particular robot.

In the remainder of this section, we provide an analysis of some context-based systems in robotics. Our aim is not to provide a comprehensive survey. Rather, we reference sample works from the robotic literature, with the purpose of investigating connections with the use of context in Information Fusion. For each of the cited works, we emphasize the type of contextual information and the representation adopted in it. In addition to the Turner's categorization, we use an additional taxonomy based on the representation structures and methodology conceived for exploiting the concept of context.

2.1.1 Environmental Context

Environmental context formalizes the information about the external world that is not necessary for achieving the goals, but provides a more exhaustive and clear modeling of the typical scenarios. This kind of information is useful to recognize situations of interest and to adapt the behavior of the system on the basis of the situation at hand. As an example, Nüchter *et al.* [9] employ environmental knowledge to establish correspondences among objects of the environment by considering geometric information. The proposed system has static knowledge about the geometrical properties of well known items. Whenever these properties are observed, the system makes assumption about the current scenario, and hence tunes its association procedures according to it, which results in a quicker and more reliable completion of the task.

The work by Rottmann *et al.* [10] exploits context-awareness to classify indoor scenarios into semantic classes. After an initial classification phase, based on the recognition of geometrical and visual features, the system makes use of its contextual knowledge to map the observed features to known classes of scenario types. To model this dependency, the system exploits an Hidden Markov Model, that is updated by sensory data and movements of the robot, and outputs the likelihood for the label of the environment.

Hawes *et al.* [11] exploit contextual knowledge about geometric and functional properties of known environments to accomplish recognition of spatial regions. Those properties are basically intended as the types of objects expected to be in a particular region and their location relative to each other. As an example, in the case of a classroom, contextual knowledge would predict the presence of desks, arranged in rows and facing a whiteboard. The context-dependent spatial regions are represented in terms of groups of anchor points, which are symbolic description of these salient objects. Through visual recognition techniques, the agent identifies and estimates the relative positions of the anchor points and hence proceeds in the labeling of the environment.

Triebel *et al.* [12] design a *Multi-Level Surface map* (MLS) to inform the robot about the terrain conditions. The authors divide the environment into different cells and store in each cell the information related to the particular area covered by the current cell. This representation of contextual knowledge is useful in designing navigation and localization systems for outdoor scenarios.

Aboshosha and Zell [13] propose an approach for adapting robot behavior for victim search in disaster scenarios. The authors collect information about unknown indoor scenarios to properly shape the robot behavior. An adaptive controller regulates the robot velocity and gaze orientation depending on the environment of the mission and on the victim distribution within the environment.

Dornhege and Kleiner [14] introduce the concept of *behavior maps*. They represent the environment as a grid and collect for each cell meaningful information related to the current context. They key idea is to directly connect the map of the environment to the behavior of the the robot. By using the information stored in each cell, they shape the behavior of the robot by means of fuzzy rules, in order to make the system *context-sensitive*.

2.1.2 Mission-related Context

Context driven choices are useful in robotic scenarios for adapting the robot behaviors to the different situations. Indeed, systems that use mission related information aim at representing *task-related* features to influence the execution and to improve the system performance. For instance, Simmons and Apfelbaum [15] generate contextual information by characterizing a task at different levels of information. The authors enhance the Task Definition Language (TDL) formalism with a new representation for the robot tasks, called *Task trees*, that relates the information about the tasks and that is a suitable way for reasoning about it.

Saffiotti *et al.* [16] exploit the concept of multivalued logic to define task requirements and specifications. The authors propose an approach for integrating task planning and execution in a unified representation, named *behavior scheme*, which is context-dependent by definition. This approach allows the system to be efficient in characterizing and planning the task and to be as reactive as possible in executing the mission.

Mou *et al.* [17] describe a context-aware robotic system — a robotic walker for Parkinson's disease patients — able to adjusts its behavior according to the context of the task. The robot detects through its sensory system the type of gait and the kind of movement performed by the patient, e.g., "turning" or "going backward". Then, contextual information is represented with a vector of variables, which determines the law of motion of the walker through simple *if-else* structures.

Calisi *et al.* [2] employ an high-level of Petri Net formalism, the so-called *Petri-Net Plans* (PNPs), to represent the task design, execution, and monitoring. The authors deploy a robot in a multi-objective exploration and search scenario. The robot features a strategic level to adapt or modify the task execution according to the mission specifications.

2.1.3 Self-related Context

Self-knowledge is often an underestimated aspect in robotic systems. However, self-related contextual information is crucial to evaluate the status of the robot and the reliability of its decisions while performing a mission. For example, Newman *et al.* [18] exploit introspective, as well as environmental, knowledge by using two different algorithms for incremental mapping and loop closure: An efficient incremental 3D scan matching is used when mapping open loop situations, while a vision based system detects possible loop closures.

Agent-related context directly refers to behavior actions and it can be adopted in behavior specialization routines, in order to optimize the task execution and the system adaptation to the environment. The use of contextual knowledge about the system status for behavior specialization is suggested by Beets *et al.* [19]. The authors exploit introspective knowledge to obtain smooth transitions between behaviors, in particular by applying sampling-based inference methods.

2.2 Context Representation

Environment, task information, and robot self-knowledge are the fundamental concepts for defining the Turner's contextual information taxonomy. Once the system gathers contextual knowledge, a common representation is needed to reason about the collected knowledge. Hence, we focus on the context representation criteria that allows the robot and, more in general, a context-aware system to exploit contextual information at different levels (e.g., at reasoning and sensory level).

A *context representation* has to provide a uniform view of the collected data and a robust reasoning process for state estimations, behaviors specialization, and task execution evaluations. In the rest of this section, we analyze the state-of-the-art by emphasizing the differences between existing context representation methodologies and we present a novel classification that groups representation structures into three classes:

1. *Embedded*;
2. *Logic*;
3. *Probabilistic*.

2.2.1 Embedded Context Representation

Systems using *Embedded Context Representation* represent context as sets of meaningful sensory features that characterize particular situations. Since this kind of representation works at a perceptive level, it is typical of reactive systems. Such systems focus on the recognition and labeling of the current context and adjust their behavior in accordance with the identified scenario, representing it at a sub-symbolic level. However, even if a reactive strategy can be effective for sensory driven recognition of known environments, such a methodology is highly system-dependent and not versatile. In fact, even if the contextual knowledge is formalized explicitly, it is inherently bonded to the perceptual structures, and hence it is specific of the particular system.

Context classification with different sets of features is used for robots relying on visual perception such as scouting mobile robots, and more generally, on systems performing visual recognition. Narayanan *et al.* [20] model reactive behaviors for a mobile robot according to a set of scenarios. Each scenario consists of traces of visual frames with the respective desired movements. During the execution of its tasks, the robot scans the environment and tries to build correlations between the sensed world and the demonstrated scenario frames. Once a correlation is established, the current context is identified and the robot actuators execute the requested motion law. Moreover, the authors describe another approach which substitutes the explicit movement commands with a set of neural networks, previously trained for a specific scenario. Hence, if the scenario has been recognized, then the corresponding network is triggered and commands the system.

When image classification or scene recognition techniques are involved, *a priori* knowledge about the geometrical and visual properties of known classes of objects can be gathered and used to direct the recognition process more efficiently [21]. These features can be encoded explicitly as desired values for functions representing particular visual features, or, implicitly, as collections of frames displaying the desired features. The detection of known features in a target image enables the system to recognize meaningful contextual elements, such as the presence of relevant objects, which are useful cues for the final classification of the image.

Buch *et al.* [22] exploit specific features for evaluating the alignment pose between objects in an image; the problem is addressed by defining descriptors that encode the geometrical features of the objects. In particular, context descriptors are used to represent the relative orientation of feature segments inside an Euclidean neighborhood around the feature of interest. Contextual descriptors are then used to perform alignment estimation with RANSAC.

Costante *et al.* [23] propose a visual classifier that clusters a target image with a *normalized-cut* based approach. In order to increase the efficiency of the system, a measure of similarity with respect to the other previously labeled sets of images is computed before the classification step. Whenever a correlation is found, the system clusters the set of images and exploits the labels of the known images to infer the classification of a new image. Here, contextual information is represented as a set of labeled images, without any further abstraction about the classes they symbolize.

Liu *et al.* [24] present a system for generating abstract graphs for *table-top* scenes from 6D object pose estimates. The system relies on the pose estimations for feature-driven recognitions, which are used to determine spatial objects relations (e.g., points of contact, relative disposition). The obtained relationships are encoded with reactive rules, which contribute to generate the abstract object graph of relations.

2.2.2 Logic-based Representation

The most common choice in modeling contextual information is the use of *declarative knowledge representation languages*. Logic-based representations range from rule-based ontologies to first order logic. The main advantage in using such a representation is that a symbolic framework implicitly provides *inference tools*, which supports planning and reasoning. In Laird *et al.* [25] cognitive architectures integrate sensory data and descriptive knowledge with contextual rules, directly into the *decision making process* of the system. More in detail, Laird's decision procedure aims at modeling the current symbolic knowledge of the system, named *Symbolic Working Memory*. The Symbolic Working Memory communicates to the perception layer and to the *permanent memory* modules, and it provides relational representations of recent sensory data, current goals, and long term memories. Contextual information is structurally defined within the permanent memory modules. More precisely, the context is represented as rules in the *procedural memory* and as scenarios (from past experience) in the *episodic memory*, respectively. The system can query the contextual database by loading the proper memory, which is continuously updated through reinforcement and episodic learning techniques.

The challenging problem for this type of architectures is in developing context modules able to dynamically update and increment their context knowledge. Indeed, turning experience into structured logical assertions needs a high level of abstraction, which is often difficult to achieve. Furthermore, logic-based models require an accurate grounding of semantics into the sensed world. Karapinar *et al.* [26] describe a learning method for expanding and correcting contextual knowledge in robotics systems. The authors represent knowledge by means of *linear temporal logic* formulas, which allow the system to analyze episodes of failures occurred in past experiences and to adjust its internal knowledge. Whenever a failure occurs, the system identifies the related configuration of *risks of failures*, which is context dependent. Therefore, the system learns how to connect possible failures to a *risk of failure* scenario, which can anticipate the failure itself. Inherently, the system learns to avoid potential failure situations, if any, and to handle different routines in performing tasks.

A system based on *formal representation languages* can be easily understood by human operators, which is a main advantage when context information is directly provided by users or obtained through interactions. However, context-aware systems that use a formal representation generally require a high level of abstraction.

Scalmato *et al.* [27] employ *pre-defined logic ontologies* to formalize contexts and situation awareness. Concepts (in form of T-Boxes) are provided by humans, while the contingent knowledge (A-Boxes) is populated by the system. This kind of representation is highly flexible, since a knowledge base based on representation languages does not depend neither on the internal structure of the system nor on its domain. Therefore, the overall context knowledge can be easily shared and adapted to different systems. Turner *et al.* [28] introduce a novel methodology for defining *distributed context-mediated behaviors* for multi-agent systems. In particular, their analysis focuses on the need of a common ontology and of expressing knowledge in a common representation, such as frame-based system or a description logic language. The authors suggest some strategy for the distributed development of contextual knowledge, as a set of comparison, fusion, and integration techniques of the ontologies built out of the experience of the single agents.

2.2.3 Probabilistic-based Representation

A robotic system is affected at any level (i.e., perception, reasoning, and action) by some degree of error, or, more in general, of *uncertainty* in its processes. Therefore, a probabilistic representation of the system is often needed. Several contextual knowledge representations formalize relations between context and desired behaviors through probabilistic structures, e.g., Bayesian Networks. Once the contextual variables are identified, Bayesian networks can model the *degree of belief* of the different scenarios and the most likely behavior quite effectively. A preliminary analysis of the contextual knowledge (both task- and environment-related) needs to be carried out off-line, in order to learn and set the network dependencies.

Witzig *et al.* [29] describe a collaborative human-robot system that provides context knowledge to enable more effective robotic manipulation. Contexts are represented by probability density functions, covering task specifications, known object properties or manipulator poses. Contextual variables are automatically computed by elaborating the perceptual information or they are specified by an external oper-

Table 1 Summary of the surveyed approaches to context-awareness. *Application*: the general field of application. *Task*: the kind of task on which the system has been tested. *T. C.* (Turner Classification): the categories of context formalized by Turner (*Self*, *Environment*, or *Task* related) that are considered. *Repr.*: the type of encoding used for representing the contextual information, i.e., *Embedded*, *Logic*, or *Probabilistic*.

Approach	**Application**	**Task**	**T. C.**			**Repr.**		
			S	E	T	E	L	P
Laird *et al.* [25]	Cognitive Architecture	Navigation	✓	✓	✓	✓	✓	✓
Kurup *et al.* [31]	Cognitive Architecture	Visual Recognition		✓			✓	
Karapinar *et al.* [26]	Planning	Navigation and Simple Object Manipulation	✓	✓	✓		✓	
Scalmato *et al.* [27]	Situation Awareness	Classification		✓	✓		✓	
Turner *et al.* [28]	Distributed Context Assessment for Multiagent Systems	General Decentralized Multiagent Tasks	✓	✓	✓		✓	
Narayanan *et al.* [20]	Mobile Robots Navigation	Navigation		✓	✓	✓		
Buch *et al.* [22]	Visual Recognition	Alignment Estimation		✓		✓	✓	

ator through a software interface. The contextual knowledge is then used to assess the internal Bayesian Network, in order to model the grasp poses of the manipulator.

Probabilistic approaches are also used for object classification. In fact, they allow the system to estimate the likelihood of membership of a particular element with respect to each category present in the learning process. Held *et al.* [30] propose an algorithm for allowing intelligent cars to recognize other cars on the roadway. Vision based object detection techniques are used to perform a preliminary recognition. Then, in order to remove the false positive perceptions, the probability of each candidate object is weighted with a contextual score, and the final likelihood for each item is computed. The contextual score is based on the object size and on the position in the scenario. Size score is high when the dimension of the object is compatible with the one of an actual vehicle. Position score is based on the Global Positioning System (GPS) information: such a score is close to the maximum if the object is positioned on the road consistently with a vehicle position.

2.3 Discussion

From the above sketch of recent developments in the use of context in robotic systems, it results that the contextual information is exploited and involved in many different ways. Here, we focus on a categorization based respectively on the representation of the contextual variables, but other approaches are also possible.

Table 1 shows a summary of the specific classes of representations used in the cited approaches. In addition to the type of encoding used for representing the contextual knowledge, we indicate also the Turner's categories involved, the application scenario, and the main task supported by context information. Fig. 2 shows how the different approaches can fit into our classification. It is worth noting that multiple representations can be exploited within the same system.

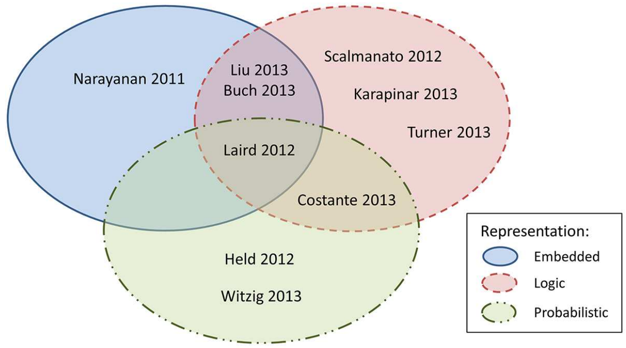

Fig. 2 Some recent approaches using knowledge representation grouped according to our classification.

Since the way of representing contextual knowledge strongly influences the implementation of the system, a representation-based categorization highlights the differences between approaches. As emerges from Table 1, not all the reviewed methods involve all the categories of contextual information. Furthermore, the analysis of the literature shows that there are many representation approaches. Indeed, each approach has its own strengths and weaknesses and multiple approaches can be combined to improve the results.

Logic and probabilistic representations both supply effective structures for describing effects and causes of contextual scenarios, the former focusing on the expressiveness of the language and the latter on the reliability of the estimates. However, logic representations alone fail in modeling inferential processes when they require complex computations. On the other hand, probabilistic encodings lack descriptive power for modeling complex environments.

Embedded representations rely on sub-symbolic structures for an effective mapping between the sensory data in input and the estimates for the contextual variables, but do not produce an easily interpretable knowledge.

The future challenge for context-aware systems will probably be to find a suitable way to combine effectively the different representation strategies, so that they can complement each other. For example, a system following a combined approach may have a layered modeling of the context: (i) A high level layer, where logical structures describe the relationships and the hierarchies among the contextual variables; (ii) A middleware layer, made of probabilistic modules that provide reliable inference processes; (iii) A low level embedded representation, for managing particular context configurations which require quick identification and a fast reactive behavior.

3 A Context-aware Framework for Information Fusion Applications

In this section, we define a context-aware framework for Information Fusion systems. It has been sketched by extending an existing framework, in particular the one proposed by Llinas in [32]. Our aim is to embedded ideas borrowed from the robotics literature into a state-of-the-art framework for Information Fusion.

The key insight is to exploit, beside the use of sensory data, additional information (i.e., context knowledge) to implement a more efficient and adaptable system. The perception system of the robot can be seen as a *Data Fusion system* that builds a representation of the world, which supports the robot operations. It is in charge of reading the sensor data, processing the information, and communicating the inferred knowledge to external entities. In our concept design, contextual knowledge is inherently assessed in the robotic system to influence the agent data-acquisition routines and, eventually, its actions.

3.1 Framework Design

As stated above, the Information Fusion framework proposed by Llinas in [32] is our starting point for developing a context-based architecture. Even if the Llinas' framework includes a component for handling contextual information, the formalization in [32] does not explicitly foresee a feedback data flow that can influence the contextual data base. Our goal is to enhance the Llinas' framework by better defining the role of the contextual system within the Information Fusion one.

In particular, we aim at designing a system able to take advantage of the context representation, containing two components for contextual exploitation: a Context Middleware and a Context Reasoner. The former is in charge of modifying the contextual knowledge base; the latter infers contextual knowledge at a high-level, independently from the robotic system deployed. This formulation allows to influence the robotic system at any layer and to accept feedback from the agent, in order to update the context data.

The use of a middleware is not novel and it has been proposed by the Information Fusion community for exploiting contextual information in high level fusion architectures. For example, Gomez-Romero *et al.* [33] discuss the use of a middleware in "*a priori* frameworks", where contextual information is known at design time and can be incorporated to the fusion procedures (hard-wired). However, in our formulation, we generalize the contribution of the middleware modules, making every layer (i.e., acquisition, detection, and fusion) *context-dependent*.

Indeed, in addition to the use of context for influencing the fusion processes (as in the framework proposed by Llinas), we want to influence also the data acquisition and decision phases. The key insight is that any component of the system can be optimized by means of context. To this end, (i) a proper contextual knowledge, (ii)

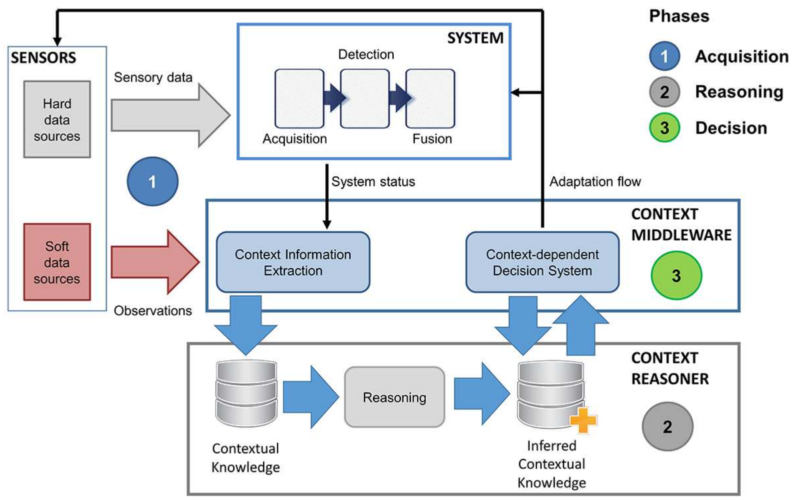

Fig. 3 A context-aware framework for Information Fusion.

a coherent methodology for the reasoning, and (iii) a dedicated adaptation logic for each of the *context-dependent* component have to be defined. By defining a common representation modality, we impose a compact and coherent way for managing every kind of information generated by any of the layer within the framework.

3.2 Framework Scheme

The scheme of our context-aware framework for Information Fusion is shown in Fig. 3. In our concept design, the operation of the context-aware framework is structured into three main phases:

1. *Acquisition*;
2. *Reasoning*;
3. *Decision*.

Acquisition.

In the acquisition phase, hard and soft sensor data are acquired. Hard sensor data refer to the system perceptions directly retrieved from the sensors of the system, while soft sensor data are information provided by human observers, such as reports from humans or context analysis by domain experts [34]. The system acquisition submodule is responsible for managing the hard sensor data. Soft information, instead, are analyzed by the *Context Information Extraction* sub-module, together with the current system status.

Context Middleware.

Context Information Extraction constitutes the first process carried out by the so-called *Context Middleware* module, which is responsible for:

1. Extracting content information from the input data;
2. Adapting the system configuration in accordance to the context.

The Context Middleware constitutes the connection between the Context Reasoner and the system. In particular, the Context Middleware translates the inferred contextual knowledge, generated by the context reasoner, in a suitable format for the underlying system. The Context Middleware allows for creating a clear conceptual separation between the reasoning processes and the state estimation processes. This leads to a less coupled system. Indeed, the chosen representation of the context is totally independent with respect to the particular system implementation.

Reasoning.

The reasoning phase relies on contextual knowledge produces by the Context Middleware. The *Context Reasoner* is responsible for informing contextual knowledge and for making it available back to the Context Middleware.

Decision.

In the decision phase, the Context-dependent Decision System sub-module uses the available contextual information to adapt the system configuration (e.g., the sensor parameters) in accordance with the current context. In such a way, contextual knowledge enhances the effectiveness of the whole system, by influencing its routines of data acquisition and processing. Accordingly, Fig. 3 illustrates the data flow between the context reasoner and context middleware modules. The Context-dependent Decision System generates the action policies for the robotic system, and simultaneously, it allows the *Context Reasoner* to know taken decisions providing feedback information that is used to update contextual knowledge.

It is interesting to notice how this pattern is totally orthogonal to the actual representation of context. Indeed, the internal structure of the context middleware is independent of the structure of the whole system.

In order to give to the reader a clear idea of our concept design and to highlight the features of our proposed framework, we want to describe the application of our framework to a concrete example. Hence, in the following section we illustrate the design of in an intelligent-vehicle system within a context-aware framework.

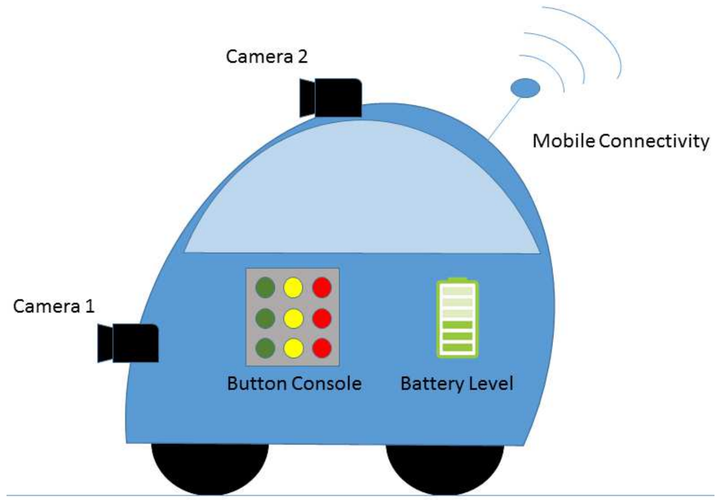

Fig. 4 The intelligent vehicle is equipped with two cameras, a mobile connection, and a button console for interacting with the human driver. It is also possible to obtain the current level of the battery charge.

4 Example of Context-based Architecture with Information Fusion

In this section, an application example is exploited to illustrate how a context-based architecture can be designed by adopting the proposed context-aware framework. In particular, the example concerns the development of a context-aware architecture for an adaptive cruise control system mounted on an intelligent vehicle.

The application scenario is illustrated by providing a description of the available data acquisition devices for the vehicle. The system architecture is designed to allow for a shared acquisition and representation of the contextual knowledge, which can be used to improve the different processes needed for accomplishing the desired tasks.

4.1 Application Scenario: An Adaptive Cruise Control System for an Intelligent Vehicle

Our application scenario focuses on an intelligent vehicle. The goal is to develop an adaptive cruise control system for providing the vehicle with the ability of adjusting its speed according to the conditions of the road (environmental information), the needs of the driver (task-related information), and the vehicle status (agent-self knowledge). Fig. 4 shows the different information sources available for the autonomous vehicle:

- Internet connection, two cameras, battery level indicator (hard data sources);
- Button console (soft data sources).

The hard data sources produce electronic and physics-based data. In this example, hard sensor information comes from two cameras, placed in the front part of the car with different fields of view. Moreover, the vehicle is connected to the Internet and websites can be accessed to extract information about weather forecast and traffic conditions. It is also possible to extract information about the charge level of the batteries that power the vehicle.

Soft data sources acquire data from human observers. In the example, the passengers have a button console that serves to communicate with the vehicle.

Since the adaptive cruise control system has multiple heterogeneous information sources, it is necessary to adopt an architecture conceived for fusing both hard and soft sensor data. Furthermore, the status of the environment, the status of the vehicle, and the goals of the passengers (e.g., the final destination) influence the behavior of the system.

4.2 Problem Formalization

An autonomous vehicle, in its basic formalization, has the task of bringing a passenger from a starting location to a goal destination through a road network. In order to achieve this purpose, the system should plan and execute a sequence of actions (e.g., “turn right at the crossroad”) respecting some constraints (e.g., “stop at the red light”), and possibly maximizing/minimizing some variables (e.g., the safety of the path or the duration of the journey).

In the case of a non context-aware autonomous vehicle, the path is generated according to the information stored in static maps and the plan is executed with the aid of a self-localization module (for instance, using the GPS signal). The topology of the road network and the position of the vehicle are the problem variables, and those variables have to be properly handled for the resolution of the problem.

A context-aware system can be seen as an extension of the above sketched model. Although the tasks, the set of actions, and the constraints are identical, in such a case the system takes advantage of contextual knowledge, thus allowing for developing adaptive solutions. In our example, the described vehicle accesses the data representing the traffic probability distributions over the several roads of the map during the different periods of the day, or information about which roads have an accident rate over some fixed threshold. Moreover, the system has the ability to acquire and take into account observations from the passengers, as the requested driving mode (e.g., economy mode) or the preferred paths (e.g., “avoid toll roads”). Finally, the vehicle can benefit from an Internet connection, supplying streaming data about the weather or the traffic conditions, or it can have a reasoning system, which can infer information about the environment by analysing the images from the cameras.

It is important to notice how all the contextual information does not influence directly the resolution of the task, which is actually solvable independently of it. The context, instead, provides a tool for evaluating the admissible sequences of actions by analysing their characteristics and for selecting among them the ones that best fit the current scenario.

4.3 Taxonomy of Context

In order to organize effectively each piece of information inside the architecture of a system, it is worth to categorize the different nature of contextual data. Indeed, contextual information can be modeled by means of different types of structures.

A first category, called *logical and physical structures*, includes all the static data, usually provided off-line, organized in data sets or probability networks. Examples of information belonging to this group are constant-time data structures, like the rules of the road or the graphs representing the road network, and the knowledge, representable through probabilistic networks, about the relationships among events and contextual variables.

A second set is constituted by the *contextual data* fed to the system during the execution of the task, usually in the form of observations. These assertions might come from human users, as in the case of voice commands from a passenger, or from external systems, like a satellite location system, the web, or the connection to a weather forecast provider.

The *inferred context* represents the third category. The inferred context amounts to all the contextual information that derives from the processing of the system variables and the context data. An example is the estimate of the traffic in a given road, calculated on the basis of the information about weather conditions and past accidents. It is worth to notice, and this third category makes it clear, how several contextual data are reciprocally related and influence each other. For example, the detection of several cars within the same road alters the context variable that represents the intensity of traffic, which in turn can affect the risk of accident of the road.

4.4 Contextual Information Fusion

Some of the above discussed relations between contextual variables follow a layered hierarchy: Contextual information can be obtained as a result of Information Fusion processes and contextual information can in turn influence the fusion processes themselves.

Each contextual variable, independently from its representation, can be used at different fusion layers as a source or as a parameter of the processing function.

In our architecture, we adopt the Joint Directors of Laboratories (JDL) model [4] for information exploitation and consider contextual knowledge to actively influence the underneath system and to improve its performance. In particular, we consider the first levels of the JDL model:

- Level 0: features;
- Level 1: individual entities;
- Level 2: structures;
- Level 3: scenarios and outcomes;
- Level 4: aspects of the system itself.

Each of the above listed levels is particularly suitable for addressing information management and exploitation, depending on the type of knowledge to be represented.

For example, visual features can be used at level 0 to help in detecting the objects of interest on the roadway (e.g., pedestrians or other vehicles). The information about the traffic, acquired through external or internal observations, contributes at level 2 to define a reliable estimation of the current scenario.

At level 4, the analysis of the status of the resources of the vehicle (e.g., battery level, fuel level, possible malfunctioning), can help in controlling the organization of the fusion processes, with the aim of minimizing the consumption of the resources, or reducing possible risks.

The context data, through the Information Fusion process, make the estimations of the state of the environment more complete and trustworthy and, consequently, they influence the fusion processes of the system variables (such as the vehicle localization, the selection of the path, the detection of colliding objects), generating adaptive solutions. It should be kept in mind that the relationship between the fusion processes and the contextual data is not only from the bottom to the top, but it is indeed a two-way relation. For instance, the recognition of several machines increases the probability of being in a congested area; the awareness of being in traffic jam might result in a different *a priori* probability to detect a vehicle, thus influencing the fusion processes at level 0.

4.5 The Information Fusion pipeline following the JDL perspective

Contextual information can affect different JDL levels [35]. Table 2 provides three examples of estimations of the *context variables* for the intelligent cruise control example.

Context variables are calculated by evaluating the available *input variables*, i.e., problem-related variables containing the information that can be useful to infer the context variables. According to the adopted Information Fusion framework, this inferring process includes three stages, namely the processes of *Common Referencing*, *Data Association*, and *Situation Estimation* that are usually indicated as the Fusion Node Functions.

For the application example, we select three problem variables to cover each of the different levels of the JDL model [36] and to reflect the Turner's classification:

Table 2 Example of fusion node functions across the JDL level for the use case.

JDL Level	Context Variables	Input Variables	Fusion Node Functions		
			CR	DA	SE
L1 Object Assessment	presence of cars on the roadway	Features detected by cameras 1 and 2	Correlation of feature points among cameras	Matching of features with car model	Classification
L2 Situation Assessment	safe following distance	Presence of raindrops on cameras, weather forecast	Camera views alignment	Clustering	Thresholding
L3 Impact Assessment	operational mode	User preferences, weather forecast, road type, battery level	Mapping of inputs on mode scores	Calculation of mode scores	Selection of mode with the highest score

1. `presence_of_cars_on_the_roadway` for task-related information, JDL level 1;
2. `safe_following_distance` for environmental information, JDL level 2;
3. `operational_mode` for agent self-knowledge, JDL level 3;

It is important to point out that we consider the levels of JDL and the Turner's categories as two orthogonal concepts and we are not interested in finding a correspondence between them.

Level 1.

The Object Assessment (level 1) example considers the contextual variable representing the presence of another car in the field of views of the cameras. The information used as input for this estimate are the points of interest identified by a feature detector on the two cameras. During the common referencing and data association processes, the features detected by the two cameras are correlated and composed, i.e., the two views are aligned to produce a single view and the detections are grouped by means of an Euclidean clustering. The impact assessment task deals with the comparison of the obtained structure with the models of known cars, and the likelihood of the detected observation being a car is estimated. The output of the whole process is a boolean variable representing the presence of vehicles in the area in front of the intelligent car.

Level 2.

As an example of a Situation Assessment variable (level 2), we consider the safe following distance that the intelligent vehicle has to maintain with respect to other vehicles ahead. Indeed, variables at level 2 models situation comprising relationships among entities with their selves and/or with the environment.

A safe distance from the car ahead with good, dry roads can be calculated by following the so-called "three-second rule" [37]: This time-lapse method uses a mark on the road, such as a power or light pole, to estimate the distance from the vehicle ahead. When weather conditions are not ideal (e.g., in case of rain), the safe following distance increases and it should be doubled to achieve a time interval of six seconds, for added safety.

In our example, the data coming from the two cameras on the vehicle are used to infer meteorological conditions, with specific reference to the detection of rainy weather. A rain drop detection mechanism is used to derive the most appropriate value for the variable `safe_following_distance`, according to the weather conditions.

Input data in such a case are the results of raindrops detections on the two camera lens. For example, raindrops can be detected by using a suitable photometric raindrop model (as in [38, 39]). Since there are two cameras, one on the lower and one on the upper part of the vehicle, it is possible to have different detection results. Thus, it is necessary to perform a projection of the camera detection results in a common reference space.

The CR task involves the alignment of the two camera views, e.g., from different scales to a common one. Information about context can be used to set the focus value of the camera. If a rainfall is expected (e.g., extracting such information from the Internet), then the focus of the cameras can be adjusted in order to better detect raindrops. Indeed a focused raindrop can be more easily detected due to its spherical form. The association task consists mainly in deciding which observations are true positives while discarding false positives. The observations from the two cameras can be grouped according to an Euclidean clustering carried out in the common reference space. The final state estimation is obtained by thresholding the detected number of raindrops onto the camera lens with respect to a predefined threshold. It is worth noting that, in this phase, soft information can be used to validate the estimation.

To set the current state of the context variable `safe_following_distance` requires to fuse the information about the number of raindrops with the weather forecast, because water drops on the camera lens can also be generated by events other than rain.

Level 3.

An example of an Impact Assessment contextual variable (level 3) is the decision about the most convenient operational mode. The operational modes presented to the passengers can be, for example, “economy”, “normal”, and “performance”. Input variables are the observations of the human passengers regarding their favorite driving mode, knowledge about the road type, and other useful data, such as information about the weather conditions. Soft input variables can be transmitted via the button console.

The common reference space is made of the possible operational modes in the internal representation of the system, for example: (1) electric-only with the engine disengaged; (2) hybrid charge-depletion; and (3) hybrid charge-sustaining [40].

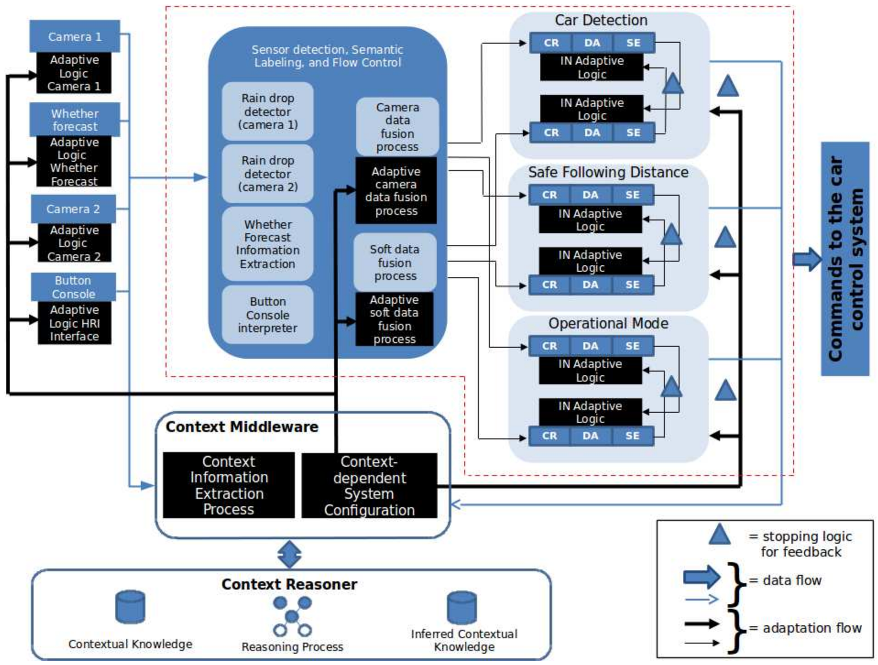

Fig. 5 Context-aware architecture for hard and soft information fusion in an intelligent vehicle system use case.

Then each of the possible values for the input variables is mapped to scores on each of the driving modes.

The association task requires the computation of the final scores for each of the operational modes, among which the one with the highest score is selected. The situation assessment then outputs the most convenient mode.

Given the above description of the application scenario and the context-aware framework for Information Fusion applications discussed in Section 3, it is possible to sketch a context-aware architecture that models the intelligent vehicle use case. Fig. 5 shows the proposed scheme.

The adaptation flow that origins from the Context-dependent system Configuration module is used to modify the parameters of the hard and soft sensor data sources, for influencing the detection, semantic labeling, and flow control, and to direct the fusion nodes functions.

By referring to the previous example, the adaptation flow can be used to change the focus parameter of the cameras, to better detect the raindrops. Moreover, the adaptation flow can be used to select a specific weather forecast website that is considered more reliable (e.g., on the basis of the GPS position).

5 Concluding Remarks

The use of contextual knowledge is a key feature in different application domains, where the acquisition, formalization, and exploitation of context information may substantially improve system performance under different operational conditions.

Throughout this chapter, our aim is to provide a general overview about the use of context information in the robotic domain, in order to discuss the possible connection with Information Fusion. To this end, we analyze the recent approaches in the literature according to the Turner's categories [3] to provide examples of use of contextual knowledge in robotics.

In addition, we highlight the manifold approaches to context representation, proposing, in particular, a novel classification for the considered methods. We categorize context knowledge as: Probabilistic-based, Logic-based, and Embedded.

Furthermore, we sketched a context-aware framework for Information Fusion in a robotic scenario. The proposed framework is an extension of the Information Fusion proposed by Llinas *et al.* in [32] with the introduction of a more detailed implementation of the *contextual module*.

Our context module includes a *Context Middleware* and a *Context Reasoner*. Context information is extracted by the modules inside the in the context middleware, exploiting both soft and hard sensor sources. Then, the context reasoner infers contextual knowledge providing it back to the decision system, which is in charge of adapting the parameters of the system components according to the current contextual configuration.

In order to validate our insights, we provide an application example, concerning the design of a control system for an intelligent vehicle. The considered system may be able to modify its operation and driving policies according to its current contextual knowledge and the available hard and soft sources data.

Acknowledgements This work was supported by ONRG Grant N62909-14-1-N061.

References

1. L. Snidaro, J. Garcia, and J. Llinas. Context-based information fusion: A survey and discussion. *Information Fusion*, 25(0):16–31, 2015.
2. D. Calisi, A. Farinelli, L. Iocchi, and D. Nardi. Multi-objective exploration and search for autonomous rescue robots. *Journal of Field Robotics, Special Issue on Quantitative Performance Evaluation of Robotic and Intelligent Systems*, 24:763–777, 2007.
3. R. M. Turner. Context-mediated behavior for intelligent agents. *International Journal of Human-Computer Studies*, 48(3):307–330, 1998.
4. E. Blasch, J. Llinas, D. Lambert, P. Valin, S. Das, C. Chee, M. Kokar, and E. Shahbazian. High level information fusion developments, issues, and grand challenges: Fusion 2010 panel discussion. In *Information Fusion (FUSION), 2010 13th Conference on*, pages 1–8, 2010.
5. R.C. Luo and M.G. Kay, editors. *Multisensor Integration and Fusion for Intelligent Machines and Systems*. Ablex Publishing Corp., 1995.

6. M. Kam, X. Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proceedings of the IEEE*, 85(1):108–119, 1997.
7. E. Blasch. Assembling a distributed fused information-based human-computer cognitive decision making tool. *Aerospace and Electronic Systems Magazine, IEEE*, 15(5):11–17, 2000.
8. D. Shen, G. Chen, J.B. Cruz, and E. Blasch. A game theoretic data fusion aided path planning approach for cooperative uav isr. In *Aerospace Conference, 2008 IEEE*, pages 1–9, 2008.
9. A. Nüchter, O. Wulf, K. Lingemann, J. Hertzberg, B. Wagner, and H. Surmann. 3D Mapping with Semantic Knowledge. In *RoboCup 2005: Robot Soccer World Cup IX*, pages 335–346, 2005.
10. A. Rottmann, Ó. Martínez, M. Cyrill, and S. W. Burgard. Place classification of indoor environments with mobile robots using boosting. In *in Proceedings of the National Conference on Artificial Intelligence (AAAI*, pages 1306–1311, 2005.
11. N. Hawes, M. Klenk, K. Lockwood, G. S. Horn, and J. D. Kelleher. Towards a cognitive system that can recognize spatial regions based on context. In *AAAI*, 2012.
12. R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.
13. A. Aboshosha and A. Zell. Adaptation of rescue robot behaviour in unknown terrains based on stochastic and fuzzy logic approaches. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS'03)*, volume 3, pages 2859–2864, 2003.
14. C. Dornhege and A. Kleiner. Behavior maps for online planning of obstacle negotiation and climbing on rough terrain. Technical Report 233, University of Freiburg, 2007.
15. R. Simmons and D. Apfelbaum. A task description language for robot control. In *IROS, International Conference on Intelligent Robots and Systems*, volume 3, pages 1931–1937, 1998.
16. A. Saffiotti, K. Konolige, and E. Ruspini. A multivalued logic approach to integrating planning and control. *Artificial Intelligence*, 76:481–526, 1995.
17. W. H. Mou, M. F. Chang, C. K. Liao, Y. H. Hsu, S. H. Tseng, and L. C. Fu. Context-aware assisted interactive robotic walker for parkinson's disease patients. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 329–334, 2012.
18. P. Newman, D. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1180–1187, 2006.
19. M. Beetz, T. Arbuckle, M. Bennewitz, W. Burgard, A. Cremers, D. Fox, H. Grosskreutz, D. Hahnel, and D. Schulz. Integrated plan-based control of autonomous service robots in human environments. *IEEE Intelligent Systems*, 16(5):56–65, 2001.
20. K. K. Narayanan, L. F. Posada, F. Hoffmann, and T. Bertram. Scenario and context specific visual robot behavior learning. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1180–1185, 2011.
21. Z. Liu, E. Blasch, Z. Xue, J. Zhao, R. Laganiere, and W. Wu. Objective assessment of multiresolution image fusion algorithms for context enhancement in night vision: A comparative study. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(1):94–109, 2012.
22. A. G. Buch, D. Kraft, J. K. Kamarainen, H. G. Petersen, and N. Kruger. Pose estimation using local structure-specific shape and appearance context. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2080–2087, 2013.
23. G. Costante, T. A. Ciarfuglia, P. Valigi, and E. Ricci. A transfer learning approach for multi-cue semantic place recognition. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 2122–2129, 2013.
24. Z. Liu, D. Chen, K. M. Wurm, and G. Von Wichert. Using rule-based context knowledge to model table-top scenes. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 2646–2651, 2014.
25. J. E Laird, K. R. Kinkade, S. Mohan, and J. Z. Xu. Cognitive robotics using the soar cognitive architecture. *Cognitive Robotics AAAI Technical Report WS-12-06. Accessed July*, 27:46–54, 2012.

26. S. Karapinar, S. Sariel-Talay, P. Yildiz, and M. Ersen. Learning guided planning for robust task execution in cognitive robotics. In *Proc. of the AAAI-13 Workshop on Intelligent Robotic Systems*, pages 26–31, 2013.
27. A. Scalmato, A. Sgorbissa, and R. Zaccaria. Describing and classifying spatial and temporal contexts with owl dl in ubiquitous robotics. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 237–244, 2012.
28. R. M. Turner, S. Rode, and D. Gagne. Toward distributed context-mediated behavior for multiagent systems. In *Modeling and Using Context*, pages 222–234. Springer, 2013.
29. T. Witzig, J. M. Zollner, D. Pangercic, S. Osentoski, R. Jakel, and R. Dillmann. Context aware shared autonomy for robotic manipulation tasks. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5686–5693, 2013.
30. D. Held, J. Levinson, and S. Thrun. A probabilistic framework for car detection in images using context and scale. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1628–1634, 2012.
31. U. Kurup, C. Lebiere, A. Stentz, and M. Hebert. Using expectations to drive cognitive behavior. In *AAAI*, 2012.
32. J. Llinas. A survey and analysis of frameworks and framework issues for information fusion applications. In *Hybrid Artificial Intelligence Systems*, volume 6076 of *Lecture Notes in Computer Science*, pages 14–23. 2010.
33. J. Gomez-Romero, J. Garcia, M. Kandefer, J. Llinas, J.M. Molina, M.A. Patricio, M. Prentice, and S.C. Shapiro. Strategies and techniques for use and exploitation of contextual information in high-level fusion architectures. In *13th Conference on Information Fusion*, pages 1–8, 2010.
34. S. Acharya and M. Kam. Evidence combination for hard and soft sensor data fusion. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8, 2011.
35. L. Snidaro, I. Visentini, J. Llinas, and G.L. Foresti. Context in fusion: Some considerations in a jdl perspective. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 115–120, 2013.
36. J. Llinas, C. Bowman, G. Rogova, A. Steinberg, and F. White. 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, 2004.
37. CEDRs TG Road Safety. Safe distance between vehicles. Technical report, Conference of European Directors of Roads (CEDR), 2010.
38. M. Roser and A. Geiger. Video-based raindrop detection for improved image registration. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 570–577, 2009.
39. S. You, R.T. Tan, R. Kawakami, and K. Ikeuchi. Adherent raindrop detection and removal in video. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1035–1042, 2013.
40. M.S. Duvall. Battery evaluation for plug-in hybrid electric vehicles. In *Vehicle Power and Propulsion, 2005 IEEE Conference*, pages 6 pp.–, 2005.