THÈSE

PRÉSENTÉ À

L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN SCIENCES ET TECHNOLOGIES DE L'INFORMATION

PAR

KÉVIN BOUCHARD

M.SC.

UNSUPERVISED SPATIAL DATA MINING FOR HUMAN ACTIVITY RECOGNITION BASED ON OBJECTS MOVEMENT AND EMERGENT BEHAVIORS

JUIN 2014

RÉSUMÉ

Les récents développements technologiques ont créé un nouveau contexte où l'informatique, devenue omniprésente et interconnectée, génère des quantités impressionnantes de données. Ces données, qui demeurent souvent entreposées et sous-exploitées, peuvent renfermer d'importantes informations qui pourraient aider les entreprises à être plus compétitives, offrir de meilleurs services ou même réduire leurs coûts. Ce nouveau contexte ne touche pas seulement les entreprises, mais aussi la recherche. C'est notamment le cas pour la discipline des habitats intelligents. Un habitat intelligent est une résidence standard qui a été augmentée à l'aide de tout genre de capteurs et d'effecteurs afin de fournir des services à un utilisateur. Une des applications les plus prometteuses de ce type de technologie est l'assistance cognitive des personnes avec une autonomie partielle. Le plus grand défi pour atteindre cet objectif est celui de la reconnaissance des activités quotidiennes de la personne qui se fait généralement à l'aide d'un algorithme logique ou probabiliste reposant sur une bibliothèque de plans définie par un expert humain. Cette bibliothèque constitue une limite fondamentale à l'implantation des habitats intelligents. Une nouvelle vision consiste donc à voir ces habitats sous la vision du Big Data et d'exploiter des techniques de forage de données afin d'extraire automatiquement les activités de la vie quotidienne du résident.

Le projet de recherche présenté dans le cadre de cette thèse s'inscrit dans cette vision novatrice de la problématique de la reconnaissance d'activité. En particulier, elle propose une solution d'agrégation de données au problème d'explosion de la taille de l'entrepôt de donnée. Cette solution s'inscrit dans la conception d'un modèle de forage de données entièrement non supervisé se déroulant en trois étapes majeures. Ce nouveau modèle à la particularité d'introduire les notions de raisonnement spatial, ce que la littérature ignore généralement, afin de pouvoir reconnaître les activités avec une meilleure granularité. Ce modèle de forage de données spatiales introduit le positionnement à partir de capteurs RFID passifs afin de créer l'entrepôt. La deuxième phase, celle de la préparation des données, se résume à la transformation des positionnements en information qualitative sur les mouvements d'objets. Enfin, une extension au très connu algorithme du Flocking est proposée afin de faire la segmentation. Plusieurs séries de tests ont été effectuées afin de valider chacune des portions du modèle ainsi que son fonctionnement global.

ABSTRACT

Recent technological developments have created a new context of ubiquitous computing and interconnection, generating massive amounts of data. These data, which remain underutilized and often stored for a short time, may contain significant information, which could help businesses to be more competitive, provide better services or even reduce their costs. This new context affects not only businesses, but also research. This is the case for the discipline of smart homes. A smart home is a standard residence that was improved with all kinds of sensors and effectors in order to provide services to its resident. One of the most promising applications of this type of technology is the cognitive support for people with partial autonomy. The biggest challenge to achieve this goal is the recognition of the daily activities of the person. It is usually accomplished using a logical or probabilistic algorithm, which is based on a library plans defined by a human expert. This library represents a fundamental limit to the installation of smart homes. A new approach is therefore to see these habitats under the vision of Big Data and exploit data mining techniques to automatically extract the activities of daily living of the resident.

The research presented in this thesis is part of this new vision of the problem of activity recognition. In particular, it proposes a data aggregation solution to the problem of the explosion in size of the data warehouse. This solution is part of the design of a fully unsupervised data mining model divided among three major steps. This new model has the particularity to introduce the notions of spatial reasoning, which the literature generally ignore, in order to recognize the activities with greater granularity. This spatial data mining model introduces a passive RFID localization algorithm to create the spatial data warehouse. Then, the positions are transformed into high-level movement information with a gesture recognition algorithm. Finally, an extension to the well-known Flocking algorithm is proposed to perform the clustering. Each part of the new model is thoroughly tested, and the results are discussed.

ACKNOWLEDGMENTS

The research project presented in this thesis is part of the activities of the Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités (LIARA) at the Université du Québec à Chicoutimi (UQAC). I would like to use this opportunity to thank those who helped me throughout this project. First, I would like to express my most sincere gratitude toward my director and co-director, Bruno Bouchard and Abdenour Bouzouane, for their continued support, for the encouragement to surpass myself and for their patience. Their valuable advices were a key element in the success of this project and for this, I am indebted. The years working together as led me to grow friendship bonds toward them and in that regard I wish them the best luck in the continuation of their projects.

I also need to use some words to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) which allowed me to pursue my dreams and realize this work without having to struggle with finance due to the substantial graduate scholarship they granted me. I also need to thank the Fonds de recherche du Québec – Nature et technologies (FRQNT) that also offered to support this project thought scholarship. Finally, I offer special thanks to Jeremy Lapalu, Dany Fortin-Simard, Jean-Sébastien Bilodeau and Sébastien Gaboury which precious work contributed to the completion of this thesis project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

AD:	ALZHEIMER DISEASE
ADL:	ACTIVITY OF DAILY LIVING
AI:	ARTIFICIAL INTELLIGENCE
ALZ:	ACTIVE LEZI
AR:	ACTIVITY RECOGNITION
ATOMGID:	ATOMIC GESTURE IDENTIFIER
BI:	BUSINESS INTELLIGENCE
DBMS:	DATABASE MANAGEMENT SYSTEM
DBSCAN:	DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE
DT:	DECISION TREE
EM:	EXPECTATION MAXIMISATION
FLV:	FUZZY LINGUISTIC VARIABLE
FNR:	FALSE NEGATIVE READING
FPR:	FALSE POSITIVE READING
FRQNT:	FONDS DE RECHERCHE DU QUÉBEC – NATURE ET TECHNOLOGIES
FSM:	FINITE STATE MACHINE
GDBSCAN:	GENERALIZED DENSITY-BASED SPATIAL CLUSTERING OF APPLICATIONS WITH NOISE
GIS:	GEOGRAPHIC INFORMATION SYSTEM
GPS:	GLOBAL POSITIONING SYSTEM
GSP:	GENERALIZED SEQUENTIAL PATTERN
GUI:	GRAPHICAL USER INTERFACE
HCI:	HUMAN-COMPUTER INTERFACE
HMM:	HIDDEN MARKOV MODEL

HHMM:	Hierarchical HMM
ID3:	Iterative Dichotomiser 3
IDE:	Integrated Development Environment
LIARA:	Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités
MAS:	Multi-Agents System
MDT:	Meta Decision Tree
NSERC:	Natural Sciences and Engineering Research Council of Canada
OS:	Operating System
PF:	Particle Filter
QSR:	Qualitative Spatial Reasoning
SH:	Smart Home
RAM:	Random Access Memory
RFID:	Radio-Frequency IDentification
RIMER:	Rule based Inference Methodology using Evidential Reasoning
RSSI:	Received Signal Strength Indication
UQAC:	Université du Québec à Chicoutimi
XML:	eXtended Markup Language

PART I

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 THE ERA OF DATA

Computer science is a recent scientific discipline in constant evolution, which was influenced by five important and continuing trends that led to major changes in our view of the world. These trends are widely recognized as the quest toward: delegation of tasks, intelligence of systems, human orientation, interconnection and ubiquity [1]. With the delegation, scientists have sought to create systems able to execute tasks in the place of humans. The goal was to give more control to software and to replace humans in critical functions (e.g. piloting, missile control, etc.). The topic of intelligence intervenes naturally, as the delegated tasks became more and more complex [2]. Artificial Intelligence (AI) always been in the fiction dreams of human, and thus it was and still is an important trend in creation of computing systems [3]. The venue of Operating System (OS) and Graphical User Interface (GUI) have changed forever the view of computing systems by putting for the first time the human in the center of the design process. Nowadays, this trend is major and software companies spend important sums of money to ensure their systems are intuitive, elegant and easy to use. The interconnection is a trend that appeared with the advent of local network and has widely spread with the apparition of the Internet [4]. Wireless technologies and smart phones are now contributing significantly to push interconnection of devices. Finally, ubiquitous computing is a trend that was first thought by Mark Weiser [5] when he described a vision of the future where computer science would be integrated everywhere around us: in our cars, in our daily life objects (coffee maker, home appliance, etc.), in infrastructure, etc. The trend expressed especially the idea that objects and devices would provide advanced services while keeping the whole computing process invisible to the user [6]. This vision combined the previous trends (the delegation of tasks, intelligence, human centric computing and interconnection) and is increasingly happening every day.

These devices and ubiquitous sensors are now a reality, and they are generating huge amounts of data around the world. In combination with the Internet and social media, these trends have led to a new situation where the data grow exponentially making the current processing and storage methods ineffective. Many researchers are interested in these data that remain, for the moment, underutilized or downright non-persistent [7]. Of this phenomenon, that some already name the era of data, have emerged new and exciting challenges grouped under the so-called Big Data [8]. A better exploitation of big data warehouses could lead to significant changes for business. One can think about being able to accurately target users for promotions or for an advertising campaign. The design of software could also be adapted to groups of users found with data mining techniques, and other patterns could be exploited to improve services offered. In particular, Business Intelligence (BI) would benefit greatly from emergence of Big Data solutions. Both fields are different but have much to say to each other [9]. Indeed, the two fields rely on the principles of frequent patterns extraction and on the exploitation statistics, but from a different angle. In addition, both rely heavily on data mining techniques for extracting trends and prediction [10]. The Figure 1.1 shows the main characteristics defining the particularities of Big Data, data mining and business intelligence.

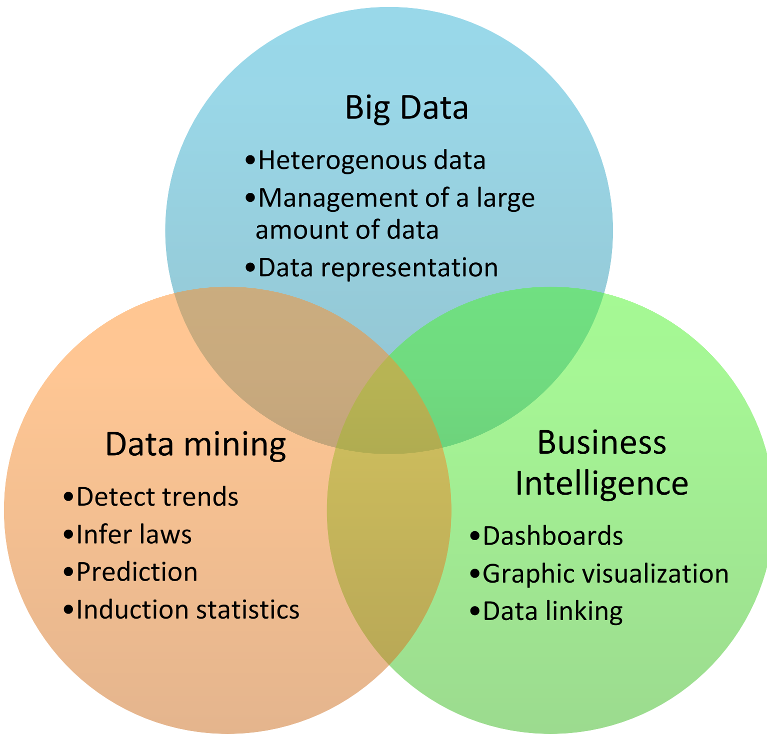


Figure 1.1: This Venn diagram illustrates the differences between Big Data, data mining and BI.

1.1.1 MAINS CHALLENGES RELATED TO BIG DATA

There are many challenges that accompanied the emergence of Big Data warehouses. The first one comes mainly from the sources of data that can be varied ranging from social media to specific hardware using proprietary protocols [11]. That is, the management of hybrid data: textual, numerical, graphical, semi-structured (XML, etc.), video, etc. The question is how to combine these sources and exploit them with software or a single algorithm? Also, how to deal with the different processing and generation speed of each data source? Another very important challenge arises from the storage of the data. The Big Data context makes it difficult to exploit classical Database Management System (DBMS). While important research enabled analytics on large database inside a DBMS [12], these systems cannot compete with new parallel systems, in particular, for the analysis of web-scale text data. The development of MapReduce by Google [13] had influential consequences on the research, attacking the issue of aggregation of the data. Many distributed database systems that were designed for Big Data implemented it such as MongoDB [14], a noSQL (standing for not only SQL) system, or Apache Hadoop [15] an open-source project. As the reader can realize, Big Data offers very interesting problems for the scientists for the next decade. However, this thesis attacks Big Data on the challenges especially related to data mining.

1.1.2 DATA MINING

To understand the context of this thesis, what is meant by data mining must first be defined appropriately. Data mining is the set of methods and algorithms allowing the exploration and analysis of database [16]. It exploits tools from statistics, artificial intelligence and SGBD. Data mining is used to find patterns, association, rules or trends in datasets and usually to infer knowledge on the essential part of the information [17]. It is often seen as a subtopic of machine learning. However, machine learning is typically supervised, since the goal is to simulate the learning of known properties from *experience* (training set) in an intelligent system. Therefore, a human expert usually guides the machine in the learning phase [18]. Within realistic situations, it is often not the case. While the two are similar in many ways, generally, in data mining the goal is to discover previously *unknown* knowledge [9] that can then be exploited in business intelligence to make better decisions.

The complete process of data mining is illustrated on Figure 1.2. Before beginning the cycle, it is important to understand the context and the data related to our situation. For example, what is the goal of the data mining? What are the consequences of errors? Are they insignificant (marketing) or critical (healthcare)? Data consideration is also important but usually for the strategy design. First of all, what types of attributes are interesting? Is there any strong association between two attributes? Those are examples of questions one should try to answer before even beginning the data mining cycle. The first step is to collect and clean the data from potentially more than one source, which can be devices, sensors, software or even websites. The goal of this step is to create the data warehouse that will be exploited for the data mining. The second step consists in the preparation of the data in the format required by the data mining algorithm. Sometime in this step, the numerical values are bounded; other time, two or more attributes can be merged together. It is also at this step that high level knowledge (temporal or spatial relationships, etc.) can be inferred for suitable algorithms. The next step is the data mining itself. It is important to choose or design an algorithm for the context and the data. There are many algorithms to be used and in Chapter 3, the main categories will be reviewed along with an assessment of their advantages and disadvantages. Finally, the data mining step should results in a set of models (decision trees, rules, etc.) that need to be evaluated. In a supervised context, it is usually easily done with statistical methods such as the F-Measure, K-Statistic or the ROC curve [16]. However, in an unsupervised context it is often required to design more complex validation process. If the evaluation is not conclusive enough, the cycle can be repeated one to many times. Indeed, data mining is a method that often does not give expected results the first time. Note that the collection and cleaning step is generally done only once regardless of the results.

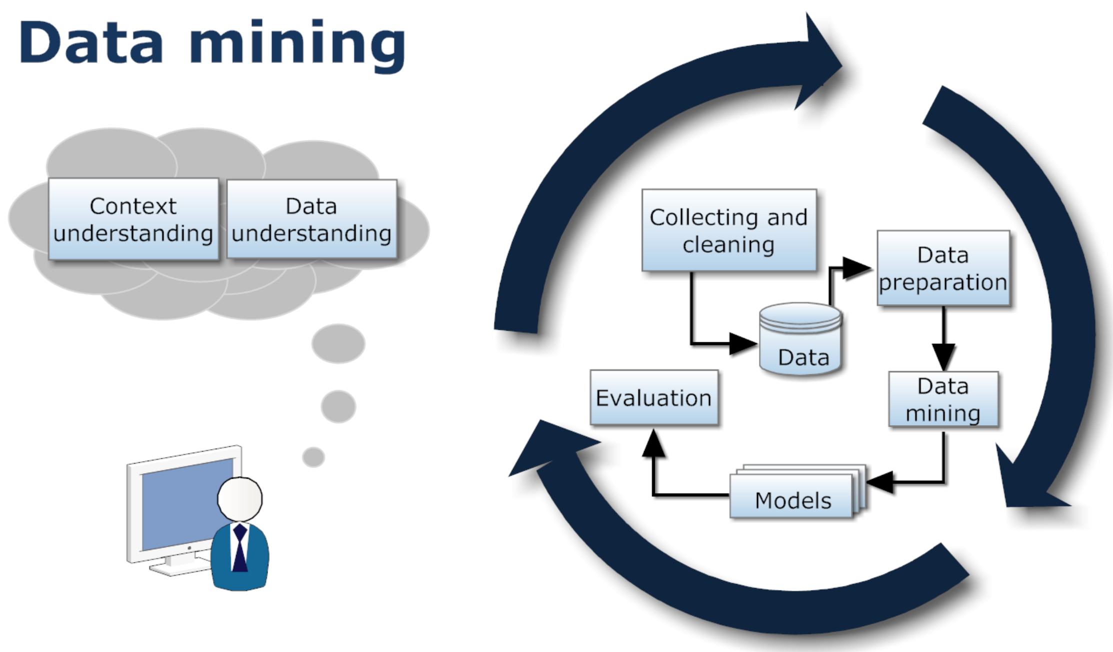


Figure 1.2: The overall data mining process

1.1.2.1 Supervised or unsupervised learning

As we discussed before, whether we talk about data mining method or machine learning in general, the process is usually classified under different categories [19]. The first one is supervised learning. The method is said supervised since it is based on training dataset with labeled examples or classes. The signification is that the algorithm can create a model that describes each class by using the known answers in the training set. In that situation, the idea is to generalize a function that maps the input to the output, and that can be used to generate output for previously unseen situations. The main implication is that somehow a human expert on the subject must label the dataset. On the opposite, unsupervised learning [18] works by using unlabelled examples. The idea is then to find hidden structure or association within the dataset and generalize a model from it. The results are sometime disappointing whether or not hidden knowledge exists in the dataset, but also sometime very surprising as the users do not know necessarily what they look for. The main implication is that there is no reward signal to evaluate the potential solutions. Unsupervised learning is often much harder to implement. Some researchers also use the name semi-supervised learning to describe their models. In that case, it usually means that the training set is partially labeled. However, it is also used to mean that unsupervised learning was applied on a training set divided into several classes by a human or an algorithm [20].

1.1.2.2 Data mining in the context of Big Data

With the emergence of Big Data, data mining needs to evolve in order to become adapted to the new challenges that have arisen. In particular, one of the most interesting and the most difficult issue is due to the incapacity to load all the data into the Random-Access Memory (RAM) of the computer. Because of this, classical data mining algorithms do not work; they must be adapted. There are several branches of the research that try to address this problem in their own way. For example, some are working on the parallelization of the algorithms [21] in order to load all the data in the RAM of a cluster of computers. Others are trying to exploit advanced sampling method to extract representative data set from the big warehouses. However, as shown on Figure 1.1, Big Data is usually a context with low information density, which poses impossible challenges to the sampling. Another possibility is to try to aggregate the low-level data into a smaller set of high level knowledge. This solution is often impossible to implement, but works very well otherwise. Another important question in the context of Big Data is about the evolution of learned models. Currently, data mining process must be repeated every time that one needs to integrate new data. With Big Data warehouse, this process is long and complex, and thus it would be interesting to develop algorithms that dynamically improve learned models from new incoming data [12]. In this thesis, a new model of data mining is proposed, following each step of the Figure 1.2, and it provides a high level knowledge aggregation solution as a first step to the problems related to Big Data.

1.2 APPLICATIVE CONTEXT OF THE LIARA

This thesis project was conducted at the Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités (LIARA). This laboratory implements a new smart home prototype that will be described in details in Chapter 4 which is equipped with a large number of sensors generating multidimensional data. This smart home produces 180 000 binary sensors information per day and as much as few millions of RFID information per day. It thus constitutes a Big Data warehouse.

To understand the applicative context for which this infrastructure was implemented, we have to discuss an important societal transformation that will come with the projected ageing of the population [22]. This transformation is going to have multiple impacts such as an increasing number of persons suffering from a type of dementia such as Alzheimer's disease [23]. These people suffer a progressive deterioration of their cognitive abilities over a period ranging from three to ten years, causing the loss of their autonomy and hence, their ability to take care for themselves. Therefore, at a certain stage in the evolution of the disease, they must be assisted continuously for the rest of their lives [24]. Not only are these people going to need assistance, but an ageing population may cause a shortage of trained health workers, which will have the effect of causing enormous stress on our already fragile health system. Hence, the LIARA and many other teams believe in the necessity of finding technological solutions to address this complex problem [25].

The evolution of information technology and electronics now makes it possible to envisage different approaches to address this societal transformation. Technological assistance inside a home qualified as smart has positioned itself as a significant trend [6] giving a new hope in the effort to postpone the institutionalization of the elderly. A smart home can be seen as a technologically enhanced environment using sensors (e.g. electromagnetic contacts, motion detectors, touch pad, radio-frequency identification tags, etc.), miniature processors integrated into the objects daily living (fridge, coffee maker, clothing, heating, dishes, etc.) and intelligent software agents communicating with each other in a goal of cooperation in the sense of multi-agent systems [1]. These environments must take decisions while taking care to limit their intrusion in order to help the resident to perform his tasks without invading his privacy. For example, if a stove burner is turned on, the device, or rather the artificial agent associated with it, must have a good idea of the behavior of the occupant as well as the context in which it takes place (preparation of a meal) by communicating with other agents. Also, suppose the agent observes that water is boiling for over an hour due to an oversight by the resident related to his cognitive impairment. Then, it could ask the main system to assist him by sending a vocal or video prompt, or using a more discrete media (light, emoticon, beep, etc.) [26]. The message type must be chosen carefully in order to stimulate the brain reactivity of the individual so that he corrects himself his mistake. When continuous support is provided to a patient with Alzheimer, cognitive degeneration of the disease is slowed and the patient can remain independent longer [27]. The first fundamental step of cognitive assistance inside a smart home is to be able to understand the ongoing Activity of Daily Living (ADL) of the inhabitant in order to identify potential problems that may interfere with the accomplishment of that ADL. This difficulty is, in fact, a special form of a well-known problem in artificial intelligence, which is called plan recognition [28]. A plan corresponds to a sequence of elementary step representing a certain ADL. In our application context, the recognition of plans intends to interpret the behavior of a person to provide, timely, appropriate services without being rejected by the individual. It is why a growing community of scientists [29-31], like the UQAC's team at the Laboratoire d’Intelligence Ambiante pour la Reconnaissance d’Activités (LIARA) [32, 33], are currently working on this specific problem of recognizing ADLs inside a smart home.

1.2.1 ACTIVITY OF DAILY LIVING DEFINITION

Through this thesis, we will often refer to the concept of *Activity of Daily Living* (ADL) but one could legitimately ask what meaning this is referring to. The notion of ADL has been first described by the Dr. Katz [34] as the set of activities that an individual performs in his routine to take care of himself. That includes activities such as preparing meals, getting dressed, toileting himself, etc. Healthcare professionals often evaluate the level of autonomy (functional status) of an impaired person with the capacity or incapacity to perform a certain ADLs. This metric is useful in assessing the degree of cognitive degeneration of a patient and to successfully discern the type of support he will need [35]. That is why many cognitive tests are based on ADLs performance such as the *Kitchen Task Assessment* [36] and the *Naturalistic Action Test* [37]. To summarize, the ADLs is a set of common activities that a normal person is supposed to be able to realize to be qualified as autonomous. Today, a consensus from researchers distinguishes two different types of ADLs:

Basic ADLs (BADLs): The basic activities of daily living (BADLs) are the set of activities that are fundamental and mandatory to answer primary needs of a person. Moving around without assistive device (ambulation), going to the bathroom, self-feeding, functional transfers (getting onto or off the bed), etc. These activities are composed of only a few steps and do not require real planning.

Instrumental ADLs (IADLs): This kind of activity needs basic planning to be performed and implies objects manipulations. These activities are needed to live alone and to live in society. For a person, being able to realize all instrumental ADLs translate into being relatively autonomous. That category includes activities such as: preparing a meal, managing money, shopping, using a phone to call someone, etc. IADLs are more complex, are composed of a higher number of steps and require better planning than basic ADLs.

In the scientific literature on assisting technology inside smart homes [38, 39], researchers mostly use ADLs without distinguishing the specific type. However, most of the time assistive systems in smart home focus on recognizing and assisting instrumental ADLs. The main reason is that a person that cannot accomplish successfully basic ADLs will need more comprehensive care that smart home assistance is inappropriate to provide.

1.2.2 DEFINING ACTIVITY RECOGNITION

Human intelligence is amazing in many facets. A good example comes from the fact that we use perceived information from the observation of a pair to deduce the action plan and the intended goal of other humans we come in contact to. That formidable ability allows us to anticipate the needs of others and therefore, promotes collaboration and assistance. From that fact, artificial intelligence has worked long on this problem that was firstly renowned as the *plan recognition* problem [28]. The first definition that we can find in the literature comes from Schmidt [40]. In his work, he defines plan recognition as "…to take as input a sequence of actions performed by an actor and to infer the goal pursued by the actor and also organize the action sequence in terms of a plan structure". In that definition, we can deduce that to perform the *plan recognition*, we suppose the existence of a plan structure (sequence of action organized in time and space) planned by the observed entity (in our case an Alzheimer resident). That structure constitutes the result that the observer tries to recognize (in our case, the smart home's sensors are the senses of the observer agent, and the algorithm is its brain).

That vision of the *plan recognition* problem is inherited from the first expert systems that were created to solve planning problem [41]. The problem of planning an activity is also a well-known challenge of the AI scientific community[3]. We can consider it as the opposite of the activity recognition problem. The difficulty resides in the identification of an actions sequence (a plan of activity) by an agent who will allow it to attain a certain objective at the end of its execution [42]. By opposition, *activity recognition* implies an observed agent that does not know the initial goal of the other agent (the observed entity), and that intends to deduce the objective by inferring from observed actions the possible structure of the ongoing plan.

1.2.2.1 Activity Recognition Inside Smart Environments

Since the original definitions from AI problems, activity recognition has evolved, getting improved by many notable authors such as [32, 43, 44]. Each has tried to adapt it to the very specific context of activity recognition inside a smart home. The trend has been to refine the notion of ambient environment to formally link it with the challenge of the activity recognition problem. For example, Goldman [45] describes it as the process of inferring an agent's plan from the observation of his action. The main distinction from previous definitions is the differentiation between the action of the observed entity, and the observation perceived by the observer. That distinction reflects the fact that actions are not directly observable in smart home context. Patterson [46] has proposed to upgrade the definition by specifying how the observations are made: "…observation made from data from low-level sensors". This new definition adheres to the paradigm of pervasive computing [5] and is much closer to the reality of the problem. It encourages the creation of enhanced environments where common objects will embed multi-modal sensors to remain less intrusive as possible. It also distances the problem of activity recognition from the legacy algorithms that considered we had access to the basic action executed by the observed entity. It is, in fact, not realistic in our context. The definition of Patterson, in contrast, assumes that only the indices triggered by actions are observable (change in the position of objects, change in the state of a sensor, etc.).

1.2.2.2 Limits of AI view on Activity Recognition

The term activity recognition refers to the fact that we presuppose the existence of an activity structure planned at the beginning by the observed entity (in our case the patient). Classically, artificial intelligence community has seen this as a three parts process:

1. Gathering observations perceived through the sensors as a result of interactions (actions) of the person with the environment.
2. Selection of a set of hypotheses (possible activities).
3. Matching method between the observations and the plans from the library describing the activities that are potentially observable [32].

That is, the classical AI view on this problem supposes the existence of a plan library. This library can be encoded in different ways. It can be formally described with first-order logic [47] or description logic [48]. It can be described as an ontology [49], a probabilistic model [50] or even with qualitative constraints of different nature [51, 52]. As we will show in Chapter 2 of this thesis, the AI literature on activity recognition make hard assumptions on the plans' library that pose fundamental limits to the implementation of such algorithms in real deployed smart home [53]. The first limit comes from the fact that it is generally assumed that the library contains all possible activities. The meaning of this is that a human expert should be able to construct formally the plans of everything that a smart home resident can do during his daily life. In fact, even if this was possible, most activity recognition supposes that the basic actions of a plan are known. However, it is equivalent to transferring the difficulty of the problem to another level. Many researchers have been working on sensors fusion [54] and middleware [55] in order to match the raw data to high level actions with a limited success [56].

1.3 Activity Discovery from Big Data Warehouse

The context of smart home assistance is interesting from a data mining point of view. In fact, the smart home can be seen as a Big Data warehouse where high dimensional information is gathered from a multitude of sensing technology [57]. This thesis applies this philosophy and proposes solutions to address the problems related to activity recognition that the classical view from artificial intelligence has yet failed to solve. In particular, the idea was to develop a complete data mining solution that would be able to automatically discover activities of daily living hidden in a Big Data warehouse. The advantages of such a solution are important. First of all, the deployment of an assistive smart home would require less intervention from human experts. Indeed, as we mentioned before, it is near impossible to clearly define how the sensors are bounded to basic actions and to define a complete and correct plans' library. Moreover, we can assume that the deployments of smart home will not always be from new house constructions [58] and thus the arduous configuration process will have to be repeated every time being time-consuming and costly. Another advantage will appear naturally as smart home adoption will spread. Being able to exploit the data warehouses of a smart home network could enable to discover new knowledge about the residents and their activities [49]. Finally, the combination of developed data mining techniques with tools from business intelligence could enable the healthcare professionals to perform a closer monitoring of the state of the residents and the smart home [59].

1.3.1 RELATED WORK ON DATA MINING FOR SMART HOMES

This thesis is not the first effort toward the development of a data mining solution to the activity recognition challenge. Many research teams are trying to exploit data mining techniques in smart homes [60, 61]. Most are, however, supervised in the sense that they require human intervention to label the training datasets. For example, Kasteren et al. [62] exploited a learned Markovian model and conditional random field to perform coarse-grained recognition of activities. Their model achieved a recognition rate of 79.4-95.6%. The labeling of the training dataset is performed by unfolding a voice recognition system to annotate the data during the realization of daily living activities. Models that exploit unsupervised algorithms are still very scarce in the literature and limited to low granularity recognition [63, 64]. This can be explained by two factors. First, there are many challenges to implement such a method: data collection, generalization, etc. Second, most researchers use existing data mining algorithms, which are not adapted to our applicative context. In particular, they only exploit naively the information gathered in the smart home. The Chapter 3 of this thesis will review in details the advances of the literature on this topic.

1.3.2 SPATIAL DATA MINING AND ACTIVITY RECOGNITION

Many researchers have recently begun to claim that one of the major reasons limiting the progression of activity recognition is that some fundamental information hidden in the data is ignored [65-67]. Constraints of different natures (logical [68], probabilistic [69], temporal [70], etc.) can be exploited to improve the recognition. For example, Jakkula & Cook [71] exploited the temporal relationships between events created by the trigger of sensors. Spatial knowledge would naturally fit in the process of data mining for activity recognition. In fact, it should also be understood that an activity can be performed in a valid sequence in time, but still be incorrect due to problems of spatial nature. For example, an activity may seem correct, but due to bad orientation of an object, execution is wrong (e.g. pouring coffee in a cup upside down). The same thing can happen if each step is correctly detected, but at some point, an object is moving away rather than closer to the activity’s zone (the problem of distance). For instance, during the task *Preparing a Coffee*, the system could be waiting to detect a movement of the coffee jar to suppose the step *Putting coffee in the cup* has been fulfilled. If the distance between the coffee and the cup is increasing, it might be because the resident skipped the step and he is storing the coffee jar. Therefore, it is of crucial importance to consider spatial aspects.

Despite this, we note that most of the work focused on the field of smart homes offer rigid recognition models that do not take into account the spatial aspect, or that incorporates it in a very limited way [72] even when they recognize its important role [73]. Moreover, most of the existing works are largely theoretical and not tested or only experimented in a non-realistic context that does not allow determining their actual effectiveness [74]. Most of them only integrated little aspect from spatial reasoning such as the subject position in the smart home [75]. In the recent years, we have worked to develop complete data mining solution addressing partially the spatial aspects in activity recognition [76, 77]. The model focused on the topological relationships [78, 79] that could be observed between objects during the realization of the activities. It was a good introduction to spatial data mining, but it was a supervised method and suffered from many drawbacks.

Since the approach presented in this thesis particularly focus on spatial aspects, we thought it would be necessary to talk a little about the research conducted on spatial data mining for Geographical Information Systems (GIS) where large spatial database are standard. In these conditions, extracting useful spatial patterns is significantly more difficult than traditional mining [80] and thus it is in that broad sense comparable to our applicative context. An example of representative work is the one of Koperski et al. [81] that is based on association rules mining to extract relationships between spatial and non-spatial predicates. Their work is particularly interesting because it is based on the assumption that the user has general knowledge about what he is looking for. In our case, the learning problem is different than those addressed in the most important spatial data mining publications [80-82]. As we will see in Chapter 3, existing approaches such as [83] are simply not adapted to our context. They are built to extract knowledge from large-scale GIS and spatial database.

1.4 CONTRIBUTIONS OF THIS THESIS

The contribution of this thesis follows in the footsteps of data mining and activity recognition approaches that have been developed during the last decades. This thesis tries to make a step forward in the context of Big Data by providing answers to the questions raised that are related to spatial aspects. In particular, this thesis explores the fundamental knowledge related to the movement of objects [84, 85] during the realization of ADLs. As the reader will see through this thesis, the contributions go beyond data mining and Big Data in the context of assistive smart homes. Most of the algorithms developed could be used for other purposes and are general enough to be applied in different applicative contexts.

At the theoretical level, three important contributions are proposed. The first one consists in an extension of the trilateration algorithm [86] which is used for localization of entities. This algorithm, exploited with by the Global Positioning System (GPS) [87] has been rarely implemented for indoor localization using noisy sensing device since the position of an entity is found at the intersection of the circles. The new algorithm is described in detail in Chapter 5. The second theoretical contribution is the creation of a gesture recognition model based on regression analysis [88] and on the spatial framework of Clementini et al.

[85]. This new model is based on solid foundation and is, to the best of our knowledge, the first model practicing segmentation of basic gesture with noisy position set. Finally, we also propose in this thesis an extension to the flocking [89] algorithm in order to be able to use it as a clustering method. While we are not the first to extend the flocking for this purpose [90, 91], our theoretical addition enables the exploitation of high level spatial knowledge extracted from our gesture recognition algorithm.

Second at the practical level, the new proposed model was implemented at the LIARA laboratory. Each part was programmed in Java under the Netbeans Integrated Development Environment (IDE) and was connected to a real smart home infrastructure. As a result of this implementation, passive RFID technology was once again demonstrated as one of the most promising technologies for assistive smart home. In particular, as the reader will discover, the implementation confirmed that in near future, this technology could be effectively used to track gestures performed by a human in real time.

Finally, this work contributes to the experimental knowledge by presenting results of various experiments that were conducted during this project along the years. These rigorous tests were designed from the expertise acquired by the LIARA's team with normal and Alzheimer's subjects [26, 92]. Each chapter of contributions in this thesis presents tests and results that were conducted at the LIARA. In addition, the expertise developed during these last years was applied to create a tangible product that could be sold to consumer or installed in eldercare centers. The product, which is described in details in Chapter 7, is a smart range equipped with a stove and an oven. It is enhanced similarly to a smart home by integrating various types of sensors. With these sensors, an automata and an android tablet are exploited to assist the user in real time by providing hints about the realization of the recipes. The prototype was approved by the university for a provisional patent, and the team obtained subventions to conduct a larger study on it [93].

1.5 RESEARCH METHODOLOGY

The research project presented in this thesis was carried out by following a research methodology divided into four key steps that were not necessarily done fully sequentially.

The first phase of the project aimed to gain knowledge of the targeted area of research by conducting a review of the literature on the problem of activity recognition in general [28, 65, 94]. The first part has allowed having an overview of the field of activity recognition, particularly in an applicative context of technological assistance of people with reduced autonomy. It has helped to identify issues and specific needs of a recognition model designed for this purpose within a smart home. The second part of this phase aimed to achieve a state of the art focused on existing data mining approaches while focusing on the one exploiting the spatial aspects [83, 95, 96]. The classical frameworks of spatial reasoning were also explored in order to develop an accurate solution [97-99]. This part has allowed arriving to the proposed contributions of this thesis.

The second phase consisted of elaborating a complete solution of data mining. To do so, the first thing we did was to select a spatial aspect to focus on: the movement. Then, we decided to focus particularly on the object and see the activity recognition as an observer's task. The model elaborated followed the strict data mining cycle of Figure 1.2. First, the collection and the cleaning of data source to create a data warehouse was done by designing a localization model that enable the tracking of entities indoor. Second, a step of data preparation was designed. That step has for goal the inferring of high level knowledge from the simple positions collected in the data warehouse. To do so, a new model of gesture extraction and recognition was created. The model exploits the well-established qualitative framework of Clementini & al. [85] and was designed to be independent and fully scalable from the localization step. Finally, an unsupervised data mining algorithm was designed to extract representative ADLs from the data warehouse and enable the recognition of activities. The model is an extension of the flocking algorithm [89] and exploit natural spatial attributes to work (acceleration, velocity, direction, etc.).

The third phase consisted into a software implementation of this new formal model of spatial data mining approach in order to validate its performances and to establish a comparison basis for the other approaches, especially those not integrating spatial constraints. To do so, we have chosen to develop it using the Java programming language running on a standard personal computer. The application was directly communicating with a real smart home infrastructure full of sensors and effectors at the LIARA laboratory. It was decided to primarily use RFID technology to deal with the spatial aspect in our model while information from other sensors was also gathered. RFID technology is perfect for the purpose of our model as it enables the tracking in real time of daily life objects at a low-cost and that, invisibly from the user. Further details will be provided in Chapter 4 on the technologies that were exploited during this thesis.

The last phase of this project of research consisted in the validation of the new model created (and implemented). At the same occasion, it has the purpose of verifying the usefulness of spatial property for the process of learning ADLs. Tests were designed at each step of the implementation for each of the three parts of the spatial data mining model. Then, a global experiment was done on the global solution to evaluate it. The Figure 1.3 shows the global solution that is described throughout the chapters of this thesis.

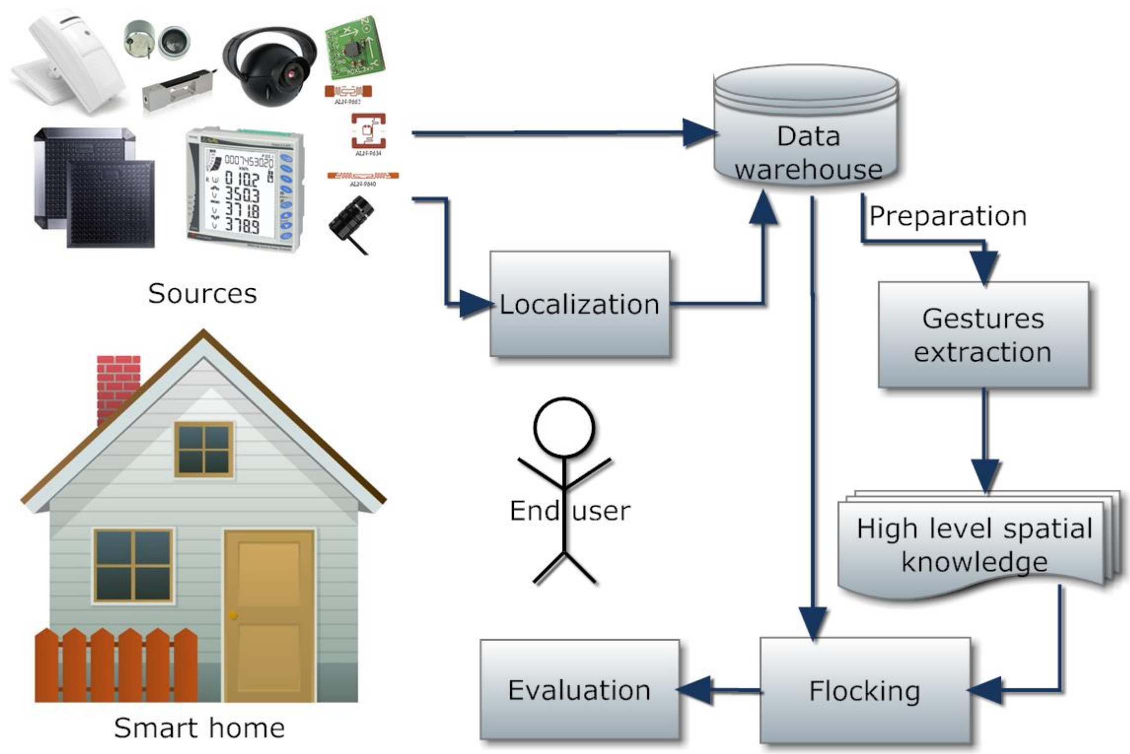


Figure 1.3: The overall spatial data mining model

It might be noted that this thesis was the object of multiple scientific publications. The advances on the localization algorithm were published in the proceedings of the *International Conference on Networked Embedded Systems for Every Application* [100], the *International*

Journal of Distributed Sensor Networks [101], the *International Journal of Wireless Information Networks* [102] and was accepted as a book chapter in *Opportunistic networking, smart home, smart city, smart systems* [103] published by Taylor & Francis. A first draft of the gesture recognition algorithm was published in the proceedings of the *International Conference on PErvasive Technologies Related to Assistive Environments* [104]. Then, the model was improved and published in the proceedings of the *International Conference on Smart Homes, Assistive Technology and Health Telematics* [105] and finally, an improvement beyond this thesis was published in the proceedings of the *AAAI-14 Workshop on artificial intelligence applied to assistive technologies and smart environments* [106]. The third part of the thesis which propose an extension of the flocking algorithm for the clustering is going to be submitted for publication in the following weeks. However, a preliminary version was published at the *International Conference on Ambient Systems, Networks and Technologies* [107]. That encouraging recognition from scientists in the field supports the conclusions of this thesis, the importance of the works realized by our team, and the results obtained.

1.6 THESIS ORGANIZATION

This thesis is divided into four parts divided amongst eight chapters. The first part, which is ending, aimed to set the table and introduce the reader to the concepts that will be discussed throughout the thesis. It has provided a description of the new context named Big Data that is leading researchers to develop new data mining solutions. It also described the applicative context of the LIARA team and linked it with the Big Data. In particular, the limits of artificial intelligence on the problem of activity recognition were described and a new vision of the smart home as a Big Data warehouse was described.

The second part of this thesis reviews the related work within two chapters. The Chapter 2 is discussing the field of activity recognition and the current existing approaches from an artificial intelligence perspective. First, it describes the different families of recognition approaches classified on a constraint type basis. For each family, we will review important works and their advantages and limitations. The first type of approach presented will be the works based on logical mathematical frameworks. These are the oldest and the more mature of all, and they regroup a large variety of approaches. The second type presented is the probabilistic models that are also well established. These models often incorporate form of supervised learning, and the implications are discussed. The third type of models discussed is the hybrid family. The chapter concludes with an assessment of the different works to better situate our contributions.

The Chapter 3 introduces the readers to the fundamental concepts of data mining. It first presents the works on decision trees in general and describes the smart home literature exploiting it. It then describes the association rules approaches, which are closely linked to the decision trees. The theory is given and then an assessment of the smart home approaches exploiting it is provided. In particular, the work of Jakkula & Cook [108] on the discovery on temporal relation is described. Our previous work (master's thesis) is also described since it is part of that family of data mining algorithms. Then, the clustering method is described. Emphasize is on the K-Means [109] and derivate methods, which are well-established and exploited in a multitude of contexts. Next, a special section on the spatial data mining methods discusses the major approaches. In particular, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [110] and its generalized counterpart (GDBSCAN) [83] are presented. The work of Liu & al. [95] on a mobility based clustering is also described. The Chapter 3 concludes with an assessment of the literature and justifies the need for new spatial data mining adapted to the context of smart environments.

The third part of the thesis, which comprises the chapters 4, 5, 6 and 7, describes the contributions of this work. Each chapter is the subject of one part of the data mining process. The Chapter 4 describes the data sources and the smart home infrastructure of the LIARA laboratory. A description of the main types of sensors is provided along with an assessment of their strengths and limitations. The Chapter 5 describes the model developed to track objects in real time with the help of passive RFID technology. Rigorous experiments conducted in the smart home with this model are presented. An assessment of the results is done in comparison with the literature on indoor localization methods. The Chapter 6 presents the new model of gesture recognition. It first describes the problem related to noisy positions data to justify the development of a new algorithm and then the formal model is described. A set of experiments is again presented along with a comparison of the literature on gesture recognition. Finally, the Chapter 7 presents the extension of the flocking algorithm in order to perform clustering on dataset. The sensors' information and the extracted gestures are then exploited in a last set of experiments linking each part as a whole spatial data mining solution.

The four part of the thesis, composed of the Chapter 8 and the appendix, concludes by presenting a detailed account of the research project highlighting the contributions of this work over previous works. This chapter will also address the limitations of the proposed model and future works arising from this research. The chapter concludes with a more personal assessment of this experience of initiation into the world of scientific research. The appendix provides further details on some aspects related to the thesis that were not fitting in the text.

PART II

RELATED WORK

CHAPTER 2

CLASSICAL APPROACHES TO ACTIVITY RECOGNITION

The part two of this thesis aims to explore the main related works relevant to our research and our applicative context. In particular, an overview of the data mining method will be seen in Chapter 3 with an emphasis on the work introducing the spatial aspects. However, before concentrating on the related works for the developed expertise, it is important to explore the classical view of artificial intelligence on the problem of activity recognition. With more precision and a summary of the main models, the reader should acquire the important notions that led us toward a data mining approach. Moreover, as we said in the introduction, the smart home is, to us, a Big Data context and one of the goals of this thesis was to propose a first aggregation solution in this context.

2.1 INTRODUCTION TO ACTIVITY RECOGNITION

Activity recognition is an instance of an old and well-known problem of computer science named the plan recognition paradigm. It has been a very active topic during the past few decades [28] following the large success of expert systems, which were exploited for planning. It has been used in various fields of research such as multi-agent systems [69] and speech recognition [111] (recognition of communicative intentions). Human activity recognition is a specific sub topic of the plan recognition that focuses on the recognition of Activity of Daily Living (ADL) performed by one or many humans in an augmented environment [73]. It has been only since recently with the arrival of ubiquitous computing and the advances of smart environment that it has become a master piece of ambient intelligence [112]. Nowadays, there is still no consensus among the scientists neither to define the problem of recognition nor to classify the various approaches, but most of them usually group the classical one under the labels "logical" or "probabilistic." The first category regroups approaches based upon logical formalisms such as description logic [48], first-order logic [47], lattice theory [32], etc. The second category includes the works that are based on well-establish probabilistic theories such as Bayesian Networks [45] or Hidden Markov Models (HMM) [113].

In this chapter, we will review the representative approaches of the field and describe their evolution through the last few years. As we will see, the classical branches of research on human activity recognition possess fundamental flaws that limit their real-world applicability. It is also very important to keep in mind that many researchers are creating hybrid solution in order to take the advantages from each branch. Before that however, it is important to bring more clarification to the concepts of ADLs recognition inside a smart home.

2.1.1 ACTIVITY RECOGNITION IN SMART HOME

In the introduction chapter, we have tried to bring a consensual definition of the problem of human activity recognition. As we mentioned, through this thesis, we stick to the definition of Patterson [46] which precise the process as recognizing the ADLs from the "…observation made from data from low-level sensors". The first hypothesis that is made for this task is the existence of a plan structure made up by an actor agent toward the realization of one or many high-level goals. It is by acting in his environment that the observing agent will perceive information. That information on the form of raw data must be transformed into high level actions. Then using those perceived actions, the observing agent will construct a set of plans' hypotheses from its own knowledge base. That knowledge base is assumed to contain all possible plans that can be realized by the actor. The Figure 2.1 summarizes the relation that exists between the actor and the observer.

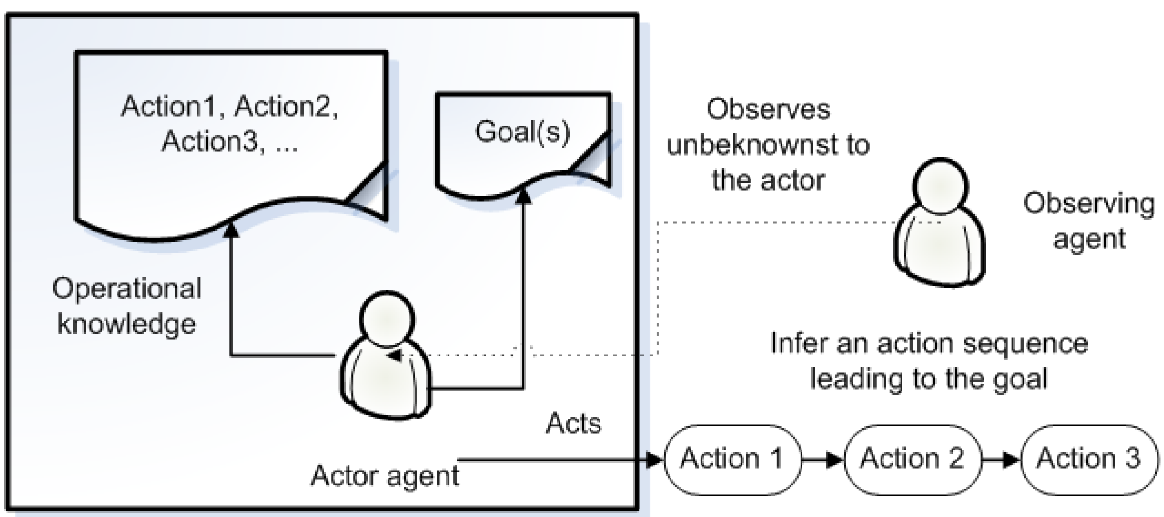


Figure 2.1: Relation between an actor agent and an observing agent trying to recognize ADLs in a smart home.

2.1.2 MAIN STEPS OF ACTIVITY RECOGNITION IN SMART HOME

The classical AI approach to activity recognition in a smart home environment divides the process into four layers as shown in the Figure 2.2. Each layer offers its own challenges and lot of difficulties. Nevertheless, as the reader will discover in this chapter, most models suppose that the first three layers are solved. The first layer is to manage the raw information from the sensors. The interaction of the actor with the smart home might trigger a multitude of sensors, and this layer has for goal to collect the heterogeneous information in order to be usable by an algorithm. There are three important challenges at this level.

- The heterogeneity of the sensors
- The precision of the sensors
- The failure of some sensors

The second layer of the process is the sensors interpretation. This phase is used to eliminate the redundant information and the noise. The main scientific challenge of this layer is the data fusion in order to obtain only the useful information for the activity recognition. The third layer consists in the interpretation of the events to infer high-level actions that could constitute a plan structure. One of the main challenges of this phase is to understand when an action began and ended from the consistent stream of data. Another difficulty is to identify when one or many events are linked to more than one action. Finally, the fourth layer is the activity recognition itself. For this last step, the idea is to use the information on action realized to infer a plan structure and a set of goals. This step was the most extensively explored in the literature.

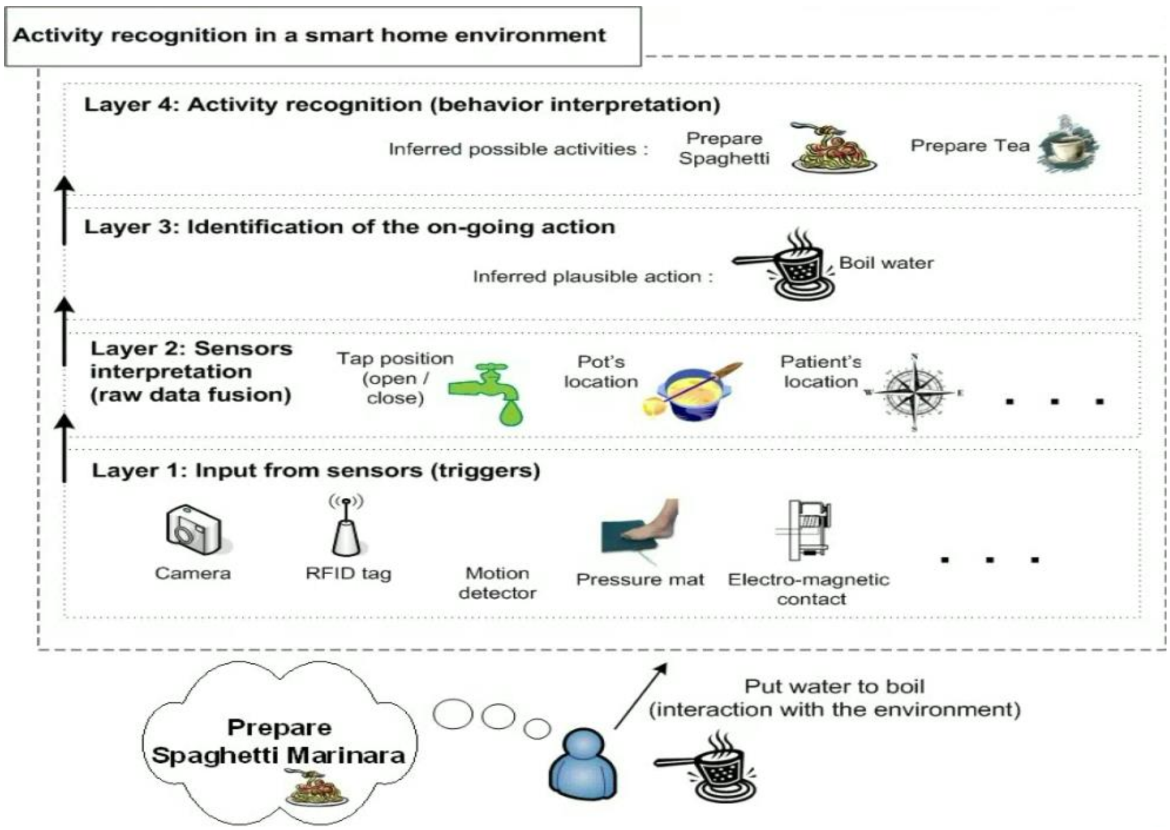


Figure 2.2: The four layers of the activity recognition process.

2.2 LOGICAL ACTIVITY RECOGNITION APPROACHES

The first branch of methods we will explore in this chapter are the one based on the mathematical logic. The logical branch was first explored with the early definition of the plan recognition paradigm for multi-agents systems (MAS) and native languages recognition. The work of Kautz [94] constitutes an important foundation of this branch and is still a reference for every scientist working in the field, whether for human activity recognition or other plan recognition problems. In his work, Kautz exploits predicate logic to infer upon a library of plans (classified in abstraction/decomposition hierarchy) from the observed basic actions. The main drawback of his model is the assumption that the plan's library is complete and correct. Other researchers [68, 114, 115] have created enhanced versions each correcting partially of the drawbacks of Kautz's theory. However, none of them answer all the needs for cognitive assistance in smart home as we will see at the end of the section.

2.2.1 KAUTZ'S FORMAL THEORY FOR PLAN RECOGNITION

Kautz's theory for plan recognition is based on the exploitation of first-order logic to formalize the process of inference (deduction) of the ongoing plan. The theory presumes that there exists a plans' library made of schemas accessible to the observer agent. In this model, plans and actions are indifferently considered as events. An event E can be specialized to form the *abstraction* (ABS) of one or more events. An event can also be decomposed into many steps with the *decomposition* axiom (DEC):

(ABS) $\forall x. E_1(x) \supset E_2(x)$

(DEC) $\forall x. E_0 \supset E_1\big(f_1(x)\big) \wedge E_2\big(f_2(x)\big) \wedge \ldots \wedge E_n(f_n(x)) \wedge \kappa$

The symbol κ describes a conjunction of constraints on E_0. For instance, Kautz used sequential constraints derived from Allen's temporal theory [116] to order the steps of an event. However, the constraints were not used for other means than that; no temporal error detection was implemented, and the goal was to organize steps of a plan more than to exploit temporal aspect.

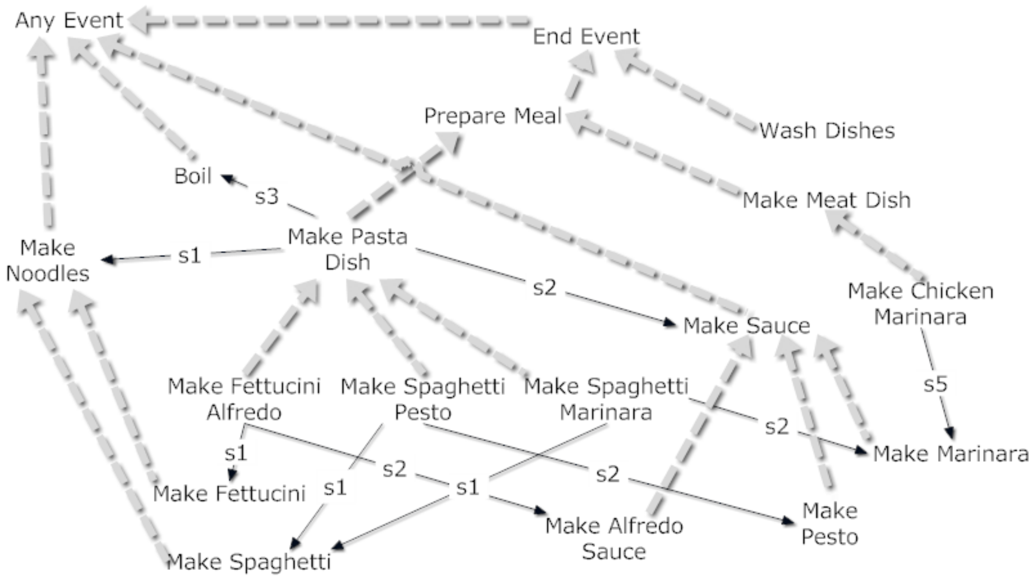


Figure 2.3: Cooking plans' library from [94]

The Figure 2.3 is an example of a library that represents the shared knowledge of a cook and an observer. The grey arrows denote the abstraction relations while the thin black arrows denote the decomposition of an event. The *End Event* is the root of self-motivated events. Such types of events represent intentions of the observed entity (goals). The *Any Event* is the abstraction of all events of the hierarchy. In Kautz's theory, the hierarchy is always presumed correct. That means the observer agent will assume that it does not include any error in the relations of abstraction & decomposition. The second assumption is that the hierarchy is complete, meaning that it includes all the possible events that can be observed. Here are some examples of first-order axioms that are used to encode the library on Figure 2.3:

(2.1) $\forall x\,.\,PrepareMeal(x) \supset EndEvent(x)$

(2.2) $\forall x\,.\,WashDishes(x) \supset EndEvent(x)$

(2.3) $\forall x\,.\,MakePastaDish(x) \supset PrepareMeal(x)$

(2.4) $\forall x\,.\,MakePastaDish(x)$

$$\supset MakeNoodle(s1(x)) \wedge MakeSauce(s2(x)) \wedge Boil(s3(x)) \wedge \kappa$$

The symbols *s1 to s3* map a plan to its steps. Note that they do not reflect any ordering and that their name is only a subjective label. Each event can possess an abstraction. For instance, *MakePastaDish* is the abstraction of these three events: *MakeFettuciniAlfredo*, *MakeSpaghettiPesto*, and *MakeSpaghettiMarinara*. Additionally, there are various types of constraints that could be defined. Kautz gaves four types in his original work [94] that can be seen with an example on *MakePastaDish* in Table 2.1.

Table 2.1: Examples of constraints on *MakePastaDish*

Constraint type	$\forall x\,.\,\boldsymbol{MakePastaDish}(x) \supset \boldsymbol{MakeNoodle}(\boldsymbol{s1}(\boldsymbol{x})) \wedge \boldsymbol{MakeSauce}(\boldsymbol{s2}(\boldsymbol{x})) \wedge \boldsymbol{Boil}(\boldsymbol{s3}(\boldsymbol{x})) \wedge$
Equality	$agent(s1(x)) = agent(x) \wedge$ $result(s1(x)) = input(s3(x)) \wedge$
Temporal	$During\left(time(s1(x)), time(x)\right) \wedge$ $BeforeMeets\left(time(s1(x)), time(s3(x))\right) \wedge$
Preconditions	$InKitchen(agent(x), time(x)) \wedge$
Effects	$PastaDish(result(x))$

The constraints of equality (as seen on Table 2.1) can be used to assert that the agent doing each step is the same that is doing the overall activity. It also can be used to make sure the noodles the agent made (*result* of the step *s1*; *MakeNoodle*) is the thing getting boiled (or the *input* of step *s3*; *Boil*). The temporal constraints, from Allen's theory [70], are used to state relations between a step and the plan or to order the execution of steps of a plan in time. So, it can specify that *MakeNoodles* must take place *during* the time of *MakePastaDish* and that *Boil* must follow the step *MakeNoodles*. As the name explicitly says so, *Preconditions* are used to assert that a condition is respected for an activity. For example, to *MakePastaDish*, the resident must be in the kitchen. Finally, an *effect* of the event is a consequence, such as there is a *PastaDish* that is the *result* of the event.

2.2.1.1 Kautz's assumptions

Considering a certain library respecting the definition provided previously, Kautz describes a recognition process based on four inferences rules. These rules allow extracting a minimal interpretation model from that library (a subset) from the introduction of observations as logic assertions. The result of this inference process is a disjunction of hypotheses (a disjoint set of possible activities) that corresponds to the activities that are included in the minimal covering tree. This process is directly inspired from McCarty's circumscription theory [117]. It uses the fourth following assumptions:

(EXA) $$\forall x. E_0(x) \supset (E_1(x) \vee E_2(x) \vee \ldots \vee E_n(x))$$

(DJA) $$\forall x. \neg E_1(x) \vee \neg E_2(x)$$

(CUA) $$\forall x. E(x) \supset End(x) \vee \big(\exists y. E_{1,0}(y) \wedge f_{1i}(y) = x\big) \vee \ldots$$
$$\vee \big(\exists y. E_{m,0}(y) \wedge f_{mi}(y) = x\big)$$

(MCA_n) $$\forall x_1 \ldots \forall x_n. (End(x_1) \ldots End(x_n) \supset \bigvee_{i,j} (x_i = x_j)) , \quad i,j \in [1,n] \wedge i \neq j$$

These assumptions are exploited every time a new observation is made to selection a minimal set of hypotheses. In the next subsection, we will look through a small applicative example of recognition.

2.2.1.2 Recognition process

Kautz's recognition process works as follow: (1) after each observation, applies the (CUA); (2) uses the (ABS) recursively to obtain an *End* type action; (3) tries to reduce the plans with (DJA)(EXA); (4) merges multiple observations to fewer plans as possible with (MCA). In order to illustrate the algorithm, let's look at an example from the *Cooking World* (see Figure 2.3):

[1]	$MakeNoodles(o_1)$	Observation
[2]	$MakePastaDish(p_1) \wedge step_1(p_1) = o_1$	(CUA)
[3]	$PrepareMeal(p_1)$	(ABS)
[4]	$End(p_1)$	(ABS)

The first thing observed is the action *MakeNoodles*. From the (CUA) assumption we infer that it is the first step of the plan *MakePastaDish* (p_1). Next, we use recursively the (ABS) axiom that allows us to find the root of self-motivated events (*End* node). Now, let us suppose that due to a lack of a certain ingredient, we know that we cannot *MakeAlfredoSauce*:

[5]	$\forall x. \neg MakeAlfredoSauce(x)$	Knowledge
[6]	$MakeSpaghettiMarinara(p_1) \vee MakeSpaghettiPesto(p_1) \vee MakeFettuciniAlfredo(p_1)$	(EXA)
[7]	$MakeFettuciniAlfredo(p_1) \supset MakeAlfredoSauce(step_2(p_2))$	(DEC)
[8]	$\neg MakeFettuciniAlfredo(p_1)$	Modus Tollens

[9]	$MakeSpaghettiMarinara(p_1) \vee MakeSpaghettiPesto(p_1)$	Elimination
[10]	$MakeSpaghettiMarinara(p_1) \supset MakeSpaghetti\big(step_1(p_1)\big)$	(DEC)
[11]	$MakeSpaghettiPesto(p_1) \supset MakeSpaghetti\big(step_1(p_1)\big)$	(DEC)
[12]	$MakeSpaghetti\big(step_1(p_1)\big)$	From 9,10,11

From that knowledge, we were able to reason with the various assumptions, and we concluded that the type of pasta the person was going to cook was spaghetti. Even though the plan is not precisely recognized, this allows predicting some information in order to assist a resident. It is only a small example, but it shows the logic behind Kautz's theory.

2.2.2 Extended work on logical approaches

The work of Kautz is an important theoretical contribution to the field of activity recognition, and none can deny its importance in the development of new models. However, it has never been really implemented and tested in a smart home context due to some major drawbacks. First, Kautz supposes that there exists a method to recognize high level actions, which is, in fact, very hard from multiple sensors and heterogeneous information. There is also no way of identifying errors in the execution of the activities. As a matter of fact, its theory will simply suppose that two plans are realized concurrently. Py et al. [114] tried to address this problem by simply modifying the library of Kautz. The advantage of his approach is that one can simply update the library, and the logic behind Kautz's model remains the same. To do so, Py defines a new type of self-motivated actions named *Error*.

The idea is to define the erroneous version of activities under that new root node so a system could know that the recognized activity is erroneous. The main problem of this particular type of solution is the requirement for an expert to define not only every possible way to perform correctly an ADL, but also enumerate the way it could be wrongly realized. This is unrealistic to assume it is feasible.

Nerzic [115] also tried to address the recognition of erroneous ADLs but from a significantly different angle. The premise is that the notion of error is intimately linked to the process of merging multiple intentions. Nerzic proposes an amendment to the postulate of minimum cardinality (MCA) using second-order logic. The idea is that when the process of merging observations fails; it may simply be the continuation of a plan in an erroneous way. Nerzic process would consist to relax the constraints of the library and retry the merger. If the second merger is a success, we know that it is a mistake to realization. His approach has never been used concretely since the second-order logic is non-tractable. Moreover, concretely, the only type of error recognizable is the error of sequence, which is still limited.

Another problem of the logic approaches is the lack of method to represent that some ADLs are more probable than other (which is addressed by probabilistic approaches). Wobcke [68] has used the possibility theory to represent the plan on a partial order. In his model, the natural plausibility of ADLs is represented qualitatively and allows an algorithm to choose more naturally which plan is ongoing. Still, even if it addresses one of Kautz drawbacks, it requires an even more complex library and does not support errors. Chen L. & al. [49] recently proposed a new system that exploits ontology for explicit activity and context modeling. Their approach is very comprehensive and is one of the first to partially address the real time recognition dilemma. Likewise to other purely logical approaches [32, 52], it is elegant and natural to understand for a human being by the way they model the ADLs and perform the inference.

2.2.3 ASSESSMENT OF LOGICAL APPROACHES

Despite all the recent advances, the purely logical approaches have many limitations restraining their real-world applicability. First, most of these approaches require a complete and exhaustive library of plans. It is not only fastidious to create such a library, but it requires a lot of time and expert resources. It is in general completely unrealistic to suppose it will be possible to do it on real smart home deployment. Secondly, as we mentioned, very often, these approaches do not adequately represent the initial probabilities being fundamentally different from one activity to another. For this reason, recent logic approaches are usually hybrid; they incorporate probabilistic models or exploit machine learning. Finally, the logical models almost always suppose that the three first layers of the activity recognition process (see Figure 2.2) are already implemented and functional. However, as we explained, transforming the raw data from noisy sensing device into high level actions is not an easy task. In fact, this assumption is equivalent to transferring the problem of activity recognition into a problem of sensors fusion and high-level actions recognition.

2.3 PROBABILISTIC ACTIVITY RECOGNITION APPROACHES

The probabilistic branch of works on human activity recognition has first been created to address the problem of equiprobability of ADLs in logical approaches. Nowadays, it is a rich branch where a variety of well performing systems exist. In this section, we will first detail how Charniak [118] and the derivate works exploit the Bayesian inference process to recognize ADLs. We will then talk a little bit about Hidden Markov Model (HMM) which is another probabilistic method that deals with the issue of high complexity of reasoning with Bayesian networks. The section will conclude by assessing the recent advances of the probabilistic branch and will talk about the limits of purely probabilistic approaches.

2.3.1 BAYESIAN APPROACH OF CHARNIAK

Charniak & al. [118] were the first to introduce the Bayesian theory to perform the task of activity recognition. A Bayesian network is a model of knowledge representation, which translates into a directed acyclic graph. Within it, the nodes represent random variables, and the arcs are the causal or conditional influences binding the nodes. In this structure, it is assumed that with the human basic expertise, it is possible to define a unique probability measure on the set of random variables. In plan recognition, the nodes of a Bayesian network correspond to basic action or to plans. The arcs linking them are dependency relationships between plans and actions. The recognition of activities becomes the task of estimating the probability distribution assigned to each node of the Bayesian network, according to the actions observed.

The formal definition of a Bayesian network comprises four important elements: a set of random variables V; a set of relations $r \in R, r \subseteq V \times V$; a table of initial probabilities IPT; and a table of conditional probabilities CPT. The relations in R are directed, and they are represented in the form $r(v_x, v_y)$ meaning that it goes from the node v_x to the node v_y. The IPT table comprises the initial probabilities of the root nodes (or parent nodes) corresponding to plans (or activities). These nodes only possess relation going to other nodes (no relations are going to them). The CPT table comprises the probabilities of non-root nodes. The non-root nodes (child nodes) correspond to basic actions and need to be at least influenced by one other node. For example, to represent the hunting library presented the paper of Kautz [94], it would look like to something similar to Figure 2.4.

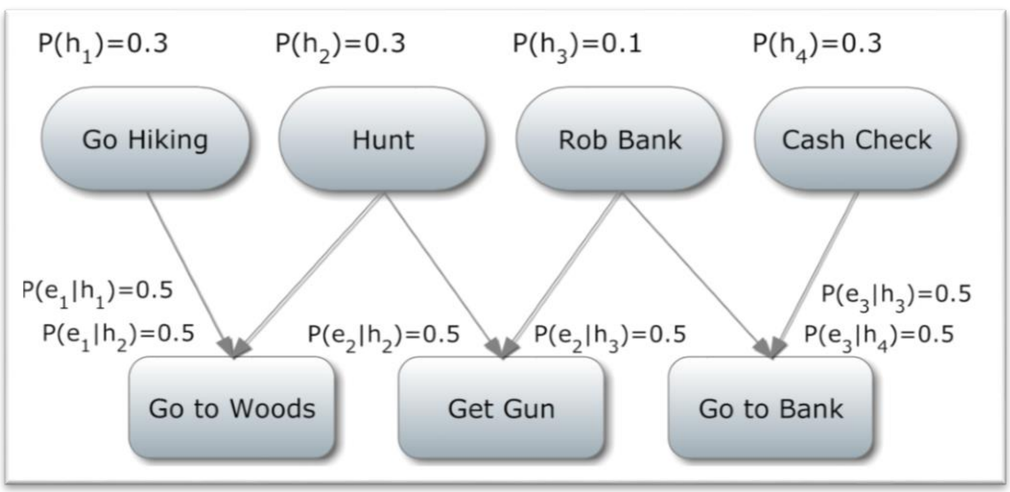


Figure 2.4: A small Bayesian Network representing a library of activities.

As we can see there are four high level nodes. They each have an initial probability of 0.3 except for *Rob Bank*, which start with 0.1. That can be the result of the expert who modeled the data from his knowledge reflecting that the observed agent is most probably honest and consequently, less likely to rob a bank. Note that the sum of the initial probabilities is 1 (and must always be 1). We can also see that the sum of the conditional probabilities going to each action node also equal 1. To perform the recognition, we have to acknowledge three postulates:

1. The hypotheses are mutually independent and disjoint.
2. The hypotheses are exhaustive (the library is complete).
3. There is a conditional independence of observations from the assumptions.

2.3.1.1 Recognition example

To understand the process of recognition with Bayesian inference, we will go through a complete example. Remember that the recognition consists to revise the probability distribution in function of the observations. The general equation for a hypothesis h_i for e_n observations is:

$$P(h_i|e_1 \wedge \dots \wedge e_n) = \frac{P(e_1|h_i) * \dots * P(e_n|h_i) * P(h_i)}{\sum_{j=[1,n]} P(e_1|h_j) * \dots * P(e_n|h_j) * P(h_j)} \tag{2.5}$$

Let's now suppose that we observe *Get Gun*. We have to revise the probabilities of each hypothesis. The results of the inference on that observation would be:

$$P(h_1|e_2) = \frac{0 * 0.3}{(0 * 0.3) + (0.5 * 0.3) + (0.5 * 0.1) + (0 * 0.3)} = 0$$

$$P(h_2|e_2) = \frac{0.5 * 0.3}{(0 * 0.3) + (0.5 * 0.3) + (0.5 * 0.1) + (0 * 0.3)} = \frac{0.15}{0.2} = 0.75$$

$$P(h_3|e_2) = \frac{0.5 * 0.1}{(0 * 0.3) + (0.5 * 0.3) + (0.5 * 0.1) + (0 * 0.3)} = \frac{0.05}{0.2} = 0.25$$

$$P(h_4|e_2) = \frac{0 * 0.3}{(0 * 0.3) + (0.5 * 0.3) + (0.5 * 0.1) + (0 * 0.3)} = 0$$

From these calculations, we conclude that two hypotheses can be eliminated from our current set of observations (*Go Hiking*, *Cash Check*) since their probabilities are equal to zero. We see that *Hunt* is now much more probable than initially (0.75) and that *Rob Bank* has also improved (0.25). The recognition agent could thus conclude at this step the the ongoing plan is probably *Hunt*. Now let's suppose we observe *Go to Bank*. We would have to redo the same calculation but with the new observation. However, we already know that e_2 cannot explain h_1 and h_4 so we do not need to revise their probabilities (it would still equal zero). The calculation for h_2 and h_3 would be:

$$P(h_2|e_2 \land e_3) = \frac{0.5 * 0 * 0.3}{0 + (0.5 * 0 * 0.3) + (0.5 * 0.5 * 0.1) + 0} = 0$$

$$P(h_3|e_2 \land e_3) = \frac{0.5 * 0.5 * 0.1}{0 + (0.5 * 0 * 0.3) + (0.5 * 0.5 * 0.1) + 0} = \frac{0.025}{0.025} = 1$$

After this second observation, the only remaining hypothesis is *Rob Bank*. Therefore, a unique plan has been identified as the current ongoing plan. When two or more hypotheses remain, a recognition system simply selects the one with the highest probability. Notice that the naive plan recognition with Bayesian inference does not support errors. If we had observed *Go to Woods* and *Go to Bank*, no plan could have explained these actions. Moreover, it is also interesting to notice that the observations could have been explained differently. Indeed, the actor could have simply been realizing two consecutives plans: *Hunt* and *Cash Check*.

2.3.1.2 Assessment of the Bayesian approaches

The main advantage introduced by the Bayesian approaches is the representation of the natural fact that some plans are more probable than other even before any action has been performed. Moreover, the probability of each plan evolves through the observation made and thus the recognition can be done *online*, that is, before the plan is completed by the actor agent. As a consequence, these models capture the uncertainty linked to plausible plans in a way that logical approaches cannot.

Nevertheless, these approaches suffer from major drawback that hampers their possible implementation in real-life smart home. First, they are based on the assumption of completeness and exclusivity of all plans of the library which is generally considered unrealistic. Moreover, these approaches suppose that we can estimate the initial probability distribution for all the possible plans and the conditional probability distribution of the linked action. In fact, it is not possible to do so in several application domains. Finally, the main problem of the Bayesian approaches comes from the propagation of the probabilities that must be done through the network during the inference process. In our example, the Bayesian network was comprised of single layer with very few actions and plans, but for larger problem, the computational time required to perform the inference would grow rapidly. Indeed, exact inference is still considered as NP-hard [119].

2.3.2 MARKOVIAN APPROACH

Following a similar path of the Bayesian approaches, many researchers [113, 120] have worked on the Markovian decision process which is a well-established theory [121]. It has been used by many researchers [46] to perform recognition of human activities in smart home. The idea is to build the plan library from a defined set of discrete states. These states are defined to represent every possible configuration of the environment. These states are then governed by a stochastic model specifying the dynamic links between them. This structure is usually described using a Hidden Markov Model (HMM) or an extension such as the Hierarchical HMM (HHMM) [84]. The structure of an HMM is a five-tuple $< S, Obs, A, B, \pi >$. The set S is a set of n hidden states that correspond to the real configurations of the system and that cannot be directly observed. The set Obs specify the input that can be taken by the system. It is comprised of m possible observations. The element A is a probability matrix of state transition. The matrix defines for each action a_{ij} the probability $P(S_{it}|S_{jt+1})$ at a time t for each state $s_i \in S$ to go in each state $s_j \in S$ at time $t + 1$. If the probability equals zero, the transition between the two states is impossible. The element B is also a matrix but defines the probability of perceiving each observation $o_i \in Obs$ for each state $s_i \in S$. Finally, π is the law governing the initial probability for each of the states. The Figure 2.5 below illustrates a very small example of a library represented by an HMM:

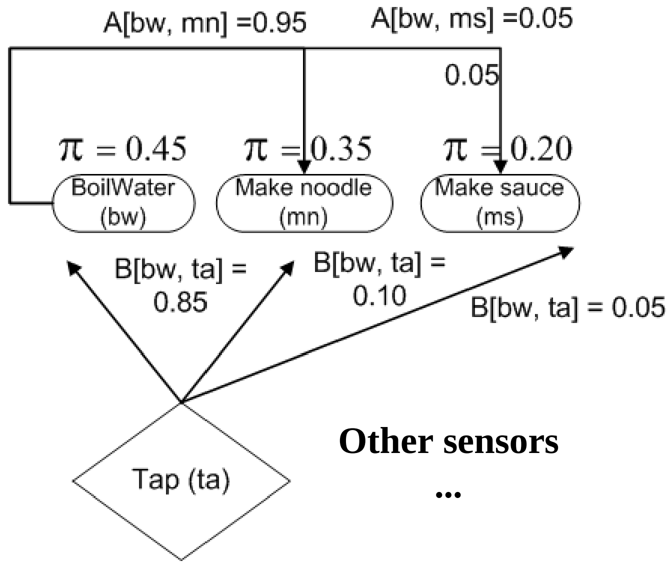


Figure 2.5: A small library represented by an HMM [48].

In the example library, the set of hidden states is defined by $S = \{BoilWater, Make\ noodle, Make\ sauce\}$. As you can see, the probabilities in B represent the probabilities of being in one of those three hidden states considering the observation of the event $Tap(ta)$.

2.3.2.1 Recognition using the Markovian model

The recognition process starts from the initial law of probability π. By knowing the the matrix of states transition A and B, the goal is to estimate the most probable sequence of states capable of explaining the observations that were made in the smart home. For example, let's suppose the set of observations is defined by the ordered list $Obs = (Tap_{Open}, Noodle_{Moving}, Stove_{ON})$. The Figure 2.6 shows all the possible states transitions explaining the three observations from the library represented in Figure 2.5. That is $3^3 = 27$

possibilities. Then, the estimation could result in the highlighted sequence of states. This estimation could be hard and complex to achieve; however a well performing algorithm exists for this task. This algorithm named Viterbi [121] enable more efficiency and the complexity is in order of $O(|Obs| \times |S|^2)$. It is the main advantage of the Markovian approaches over the Bayesian one.

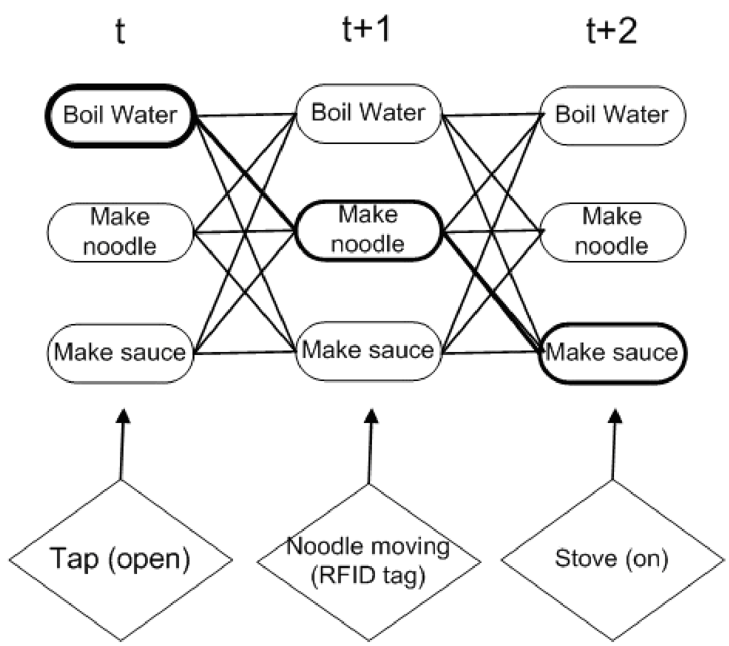


Figure 2.6: Example of a most probable sequence of states transitions from three observations

2.3.3 FINAL WORD ON PROBABILISTIC APPROACHES

We can find many probabilistic approaches in the scientific literature [44, 45, 113] due to the very good accuracy and the easiness of implementation of such models. Most currently working assistive systems in smart home implement a derived form of HMM, and they are certainly the best existing solutions right now [122]. They, however, have some disadvantages. While a Bayesian network is relatively simple to define, it is a whole different story for an HMM. It supposes that we are able to enumerate an exhaustive set of all possible configurations of the environment. Obviously, that limits the implementation of HMM to very small libraries. Moreover, HMM and Bayesian networks are complex to create and hard to scale. When the library is created, adding new ADLs is not a simple task. Finally, the last drawback of these approaches is the supposition that an expert is able to define accurately the initial/conditional probabilities. However, this last issue is solvable by exploiting a learning technique together with the probabilistic model.

2.4 SPATIAL ACTIVITY RECOGNITION APPROACHES

Spatial recognition is only beginning to get the researchers' attention even though it has already been recognized as a fundamental aspect of activity recognition algorithms. In this thesis, our goal was to develop a spatial data mining model in order to introduce the important spatial information embedded in ADLs. In the next chapter, we will look at what the data mining community has to say about it. Before that, in this section we will review the non-data mining models of activity recognition that first addressed this limitation of the literature. In particular, we present three models. The first one is an assistive system that recognizes only one activity. Nevertheless, it is still interesting because it is a concrete working system. The second one is a novel approach that is based on the natural chemotaxis process of bacteria. It integrates the spatial aspect for activity recognition. The third one is the topological model developed during the master thesis project at the LIARA laboratory. This model is still actual despite its drawbacks and being taken over by new students at the laboratory [123].

2.4.1 RIMER SYSTEM

As it has been explained before, recognition approaches based on spatial reasoning is scarce. The work we present in this section comes from Augusto & al. [72] and is named RIMER. This team also works on the problem of technological assistance in smart home. They investigated the integration of spatio-temporal information into smart home algorithms because they believed, as we do, that space is a crucial aspect in monitoring activities. The first thing to know is that RIMER is a Rule based Inference Methodology using Evidential Reasoning that was developed by Yang & al. and published in [124]. Augusto and his team extended RIMER with an active database framework [67] in order to deal with spatio-temporal aspects of human activities monitoring. As you will see through the description of its functioning, it can identify very simple *situations*. To validate their approach, they addressed a particular case study in which the occupant fainted or fell. Therefore, the spatial integration is mostly about the resident position. To follow the resident position, they combined RFID technology with infrared motion sensors. The resident had a tag attached so when passing through door RFID antenna would detect him, and motion sensors would tell in which room he entered. That is a pretty basic system, but it worked fine for their case study.

2.4.1.1 Rule-based design

Active databases are characterized by their Event-Condition-Action (ECA) rules. They are designed to react to incoming information and have the following syntax:

ON <Event>, **IF** <Condition>, **DO** <Action>

The *event* part specifies the signal that triggers the rule whereas the *condition* must be filled in order to react. If the condition is met, the *action* part is executed. However, in smart home, events present uncertainty due to the lack of precision from sensors. There is also such uncertainty in the condition part and in the relation that links both. That is why Augusto & al. used *belief* rule instead of classical one. That kind of rule incorporates a degree of confidence in the statement. In their work, it is merged directly in the rule as follows:

IF at_kitchen_on with *high* confidence **Followed_by** tdRK_on with *medium* confidence **Followed_by** no_movement_detected for 10 units of time
THEN assume with 80% confidence that occupant is compromised

As you can see it is still very straightforward to understand. The events are highlighted in white, and *tdRK_on* is an acronym used to mean *transition* (td) *from room R* (R) *to room K* (K). The *transition* and the *position* state are the tools exploited for spatial reasoning. They also integrated functionalities to deal with time in order to position events in time in their belief rule. They choose to order events using only two temporal relations *earlier than* (<) and *simultaneous* (=). However, events are already ordered using classical logic connectives ($\wedge$, $\vee$, $\neg$) and only from the logic *AND* temporal relations are meaningful (if only one condition *OR* another is fulfilled, there is no need for temporal relations between the two). For that purpose, they introduced two new symbols: $\ddot{\wedge}$, $\overline{\overline{\wedge}}$. So if we have $A \ddot{\wedge} B$, it means A true and later B true too. For $A \overline{\overline{\wedge}} B$, it would means that A and B are simultaneously true.

2.4.1.2 General RIMER operation

The general architecture of RIMER is illustrated on Figure 2.7. The two essential components to a rule-based system are the knowledge base and the inference engine. In their work, the knowledge base is a relational database where the rules are generated entirely by experts. So, in case of rules with confidence degree, experts have to exploit their judgment to approximate the real situation. However, they noticed in their paper that we could use machine learning techniques instead to extract that same knowledge. The inference system is classical one where rules have a *weight* to establish a priority in the case that many can be fired at the same time.

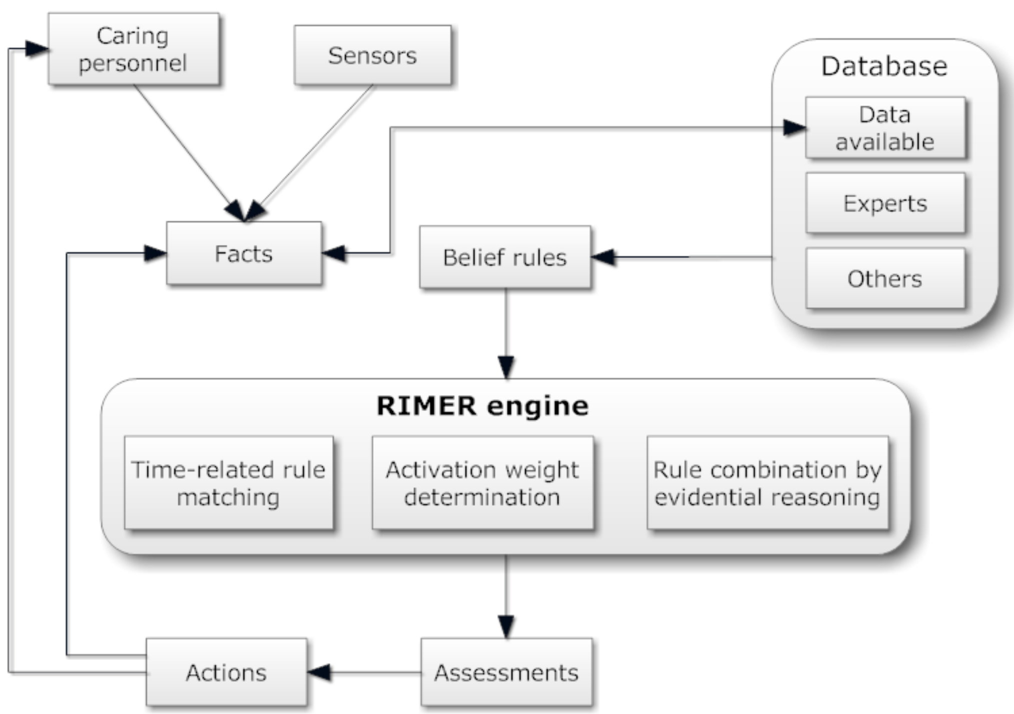


Figure 2.7: General RIMER architecture

2.4.1.3 Example scenario

Consider the problem we described in the earlier sections where an assistive system would try to recognize when the resident has fainted or fallen. An example of rule to detect such a situation in the kitchen would be that:

> **IF** at_kitchen_on Λ̈ tdRK_on Λ̈ no_movement_detected
> **THEN** assume the occupant has fainted

However, that rule must be adjusted with uncertainty. In their work, they define four levels of confidence: High (H), Medium (M), Low (L) and None (N). So the grades for *at_kitchen_on* are:

(2.6) $$A_1^k \in \{H, M, L, N\}, k \in \{1, \dots, nbRules\}$$

Similarly, the same grades are used for *tdRK_on* (A_2^k) and *no_movement_detected* (A_3^k). So the result confidence is represented in a belief distribution representing those four values for each rule. Consequently, if we update our preceding rule, it would give something like this:

> **IF** at_kitchen_on with (H) Λ̈ tdRK_on with (M) Λ̈ no_movement_detected with (H)
> **THEN** the estimation that the occupant has fainted is
> {(H, 0.7), (M, 0.3), (L, 0), (N, 0)}

Here, the belief distribution means the system has a degree of confidence of 70% that the resident has fainted with high possibility and 30% that he has fainted with medium possibility. From these rules, the inference system can decide to assist the resident or not. In that situation, it would report that the resident has fainted to health authorities.

2.4.2 CHEMOTACTIC MODEL

Riedel & al. [125] had the idea of creating a better model for activity recognition using spatial aspect from a chemotactic model. The idea comes from the world of bacteria where a process, named chemotaxis, allows bacteria to directionally swim in response to a chemical or other physical gradient [126]. For motile bacteria such as the *Escherichia coli* (E. Coli), it acts by either attracting the cell to an increasing gradient or repelling from harmful regions [127]. The cellular chemotatic model of Riedel & al. uses an abstraction of that process to represent and recognize ADL in a smart home.

In the model, a cell is composed of receptor type $\{R_i\}_{i=1}^{n}$ that works to match *molecule* from the environment. An activity is a group composed of cells. A molecule is a spatial symbol $u \in U$ where U is the set of all possible symbols. The notation $|R_i|$ denotes the specified number of receptors for each receptor type. Thus, the total number of receptors of a cell is given by $p = \sum_i |R_i|$. The environmental space *E* is a two-dimensional Cartesian plane where the cells possess a pair *x*,*y* positioning it in *E*. The place where molecules are conceptually released is set to the origin point. Cells are, however, positioned to (1.0, 0.0) at the beginning. Cells possess a velocity property *v* determining the movement within the coordinate space *E*. In the model, release of molecules *u* in environmental setting *E* increase the concentration and can be detected by cells with a free receptor of the same receptor type.

Since the highest concentration of molecules is known in the model, cells know exactly the direction to travel and move toward the attractant. A condition subsists: the cell must not be already in a zone with high concentration.

Chemotactic cells possess a memory associated with the irreversible binding of molecules to receptors that grant them the ability to detect increasing environmental concentration. That memory capability is determined by the fixed maximum number of receptors $|R_i|$ of each receptor type R_i that the cell possesses. If a cell does not possess a free matching receptor after the increase of an environmental chemical increase (after the release of a molecule *u*) but still possess receptors of that type, the cell will perform a random walk. When cells move close to the attractant source so the Euclidian distance *d* between the cell and the origin is less than the high concentration threshold, they perform a random walk irrespective of increasing concentrations. Otherwise, it returns to normal behavior. Cells with a higher degree of similitude to tests sequences will get closer to the attractant source. The molecules representing an activity sequence are *released* into the chemotactic environment consisting of β classes with *m* cells per class. Then, the system find the cell ϕ in *Z*, where *Z* is the set of all activity cells, which has a minimum Euclidian distance to the attractant source δ of *E* according to (2.7). The minimum distance cell ϕ is then used in the classification decision.

(2.7) $$\phi = arg\min_{g \in Z} d(g, \delta)$$

2.4.2.1 *Methodology and experiments*

Now that we have reviewed the chemotactic model, let's look at how they implemented it concretely in a smart home infrastructure. To do so, they used the multiple camera tracking system of Nguyen & al. [113] to build a dataset comprising six activities: *getHomeWatchTV*, *haveSnackWatchTV*, *atHomeWatchTV*, *readingNewspaper*, *havingBreakfastToast* and *havingBreakfastEggs*. The Figure 2.8 shows the smart home architecture and the spatial path for each activity's sequence.

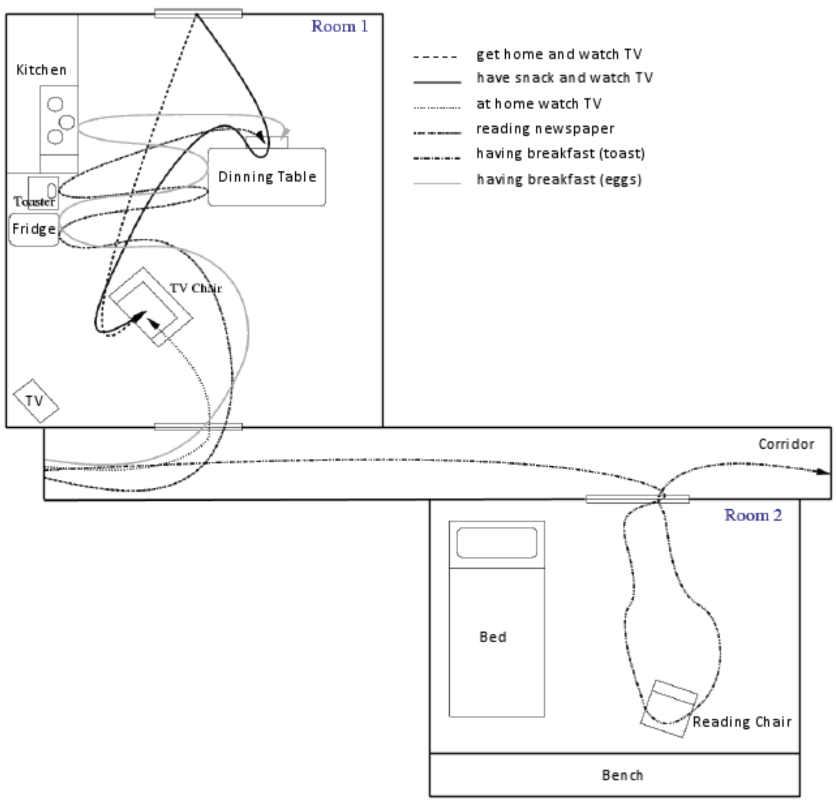


Figure 2.8: Spatial path for each of the six activity sequences [125]

They needed a basis to compare the efficiency of the chemotactic model. To do so, they used Hidden Markov Model (HMM) [121] that allowed them to recognize spatial activity patterns. They divided their smart home into logical spatial zones and created a model that had 156 different spatial states for the HMM construction. A HMM was required for each activity class. The 156 states were corresponding to the set *U* of the chemotactic model. To test the model, it was required to transform two dimensional coordinates (x, y) into a unique integer that would exist in *U*. From that point, they conducted experiments on the six different activities and demonstrated that their model was significantly better than common HMM built for spatial activity recognition. The Table 2.2 shows these results.

Table 2.2: Results for performance comparison of the chemotactic and HMM.

Technique	Precision (%)	Recall(%)
Chemotactic model	99.75%	100%
HMM	90.75%	90.75%

2.4.3 Our previous model using topology

In our previous work [52, 128, 129], we also tried to incorporate spatial information in a new algorithm made to recognize activities of daily living. To do so, we exploited the topological relationships that exist between entities present in the smart environment. It was done by exploiting the framework of Egenhofer & Franzosa [78] which define the relation between two entities e_1 et e_2 with the formal intersection structure between their interior (°) and boundary (∂) points: $< \partial e_1 \cap \partial e_2,\ e_1{}^\circ \cap e_2{}^\circ, \partial e_1 \cap e_2{}^\circ, e_1{}^\circ \cap \partial e_2 >$. By using the simple invariant empty property of sets, there are sixteen possible relation types. However, only eight exist for physical regions without holes as shown on Figure 2.9.

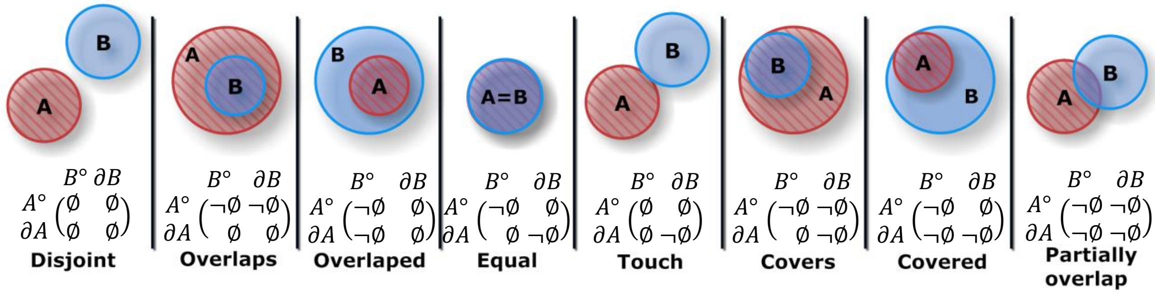


Figure 2.9: Topological relationships between physical entities A and B.

In that model, activities are defined by a set of constraints K such that:

(2.8) $$K = \{T(e_1, e_2) | e_1, e_2 \subseteq O \times O \cup R \times A\}$$

where T is a topological relation, O is a physical object, R is the resident and A is a logical area of the smart home. The recognition process consists then to evaluate the plausibility of each ADLs in the knowledge base from the constraints observations made in the environment. The plausibility is calculated by using the neighborhood graph of the topological relationships which is illustrated by Figure 2.10.

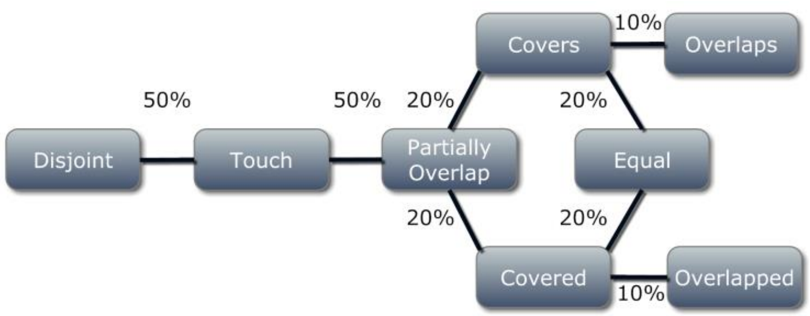


Figure 2.10: Neighboorhood of the topological relation.

For example, if the defined relation in the knowledge base is $PartiallyOverlap(e_1, e_2)$ and the relation observed is $Disjoint(e_1, e_2)$ the number of points gained from the observation is $\varphi * (100\% - 50\% - 50\%) = 0$. On the other hand, if the observed relation is $Covers(e_1, e_2)$, the points equal $\varphi * (100\% - 20\%) = 4\varphi/5$. Obviously, we limit the points from 0 to φ so if the weight exceeds 100% we never get negative scores. Finally, considering that the $Similarity$ function returns the percentage of similarity from 0 to 100%, the scoring of an activity ($a_{\delta,i}$) for the iteration i is:

(2.9)
$$a_{\delta,i} = \sum_{n=a_{t,0}}^{a_{t,i} \in a_T} \sum_{m=l_0}^{l_j \in L} \varphi * Similarity(n, m)$$

where a_T is the set of topological relationships defining the activity a. It is the same calculation for the topological relationships implying the resident and a smart home zone.

The next step of the algorithm is to choose the most plausible activity that is ongoing. In other words, it has to choose which ADL best explains the observations made up until the current iteration. The plausibility of an activity after i_c iterations is not only calculated by the points it earned at the current time. The points previously obtained must also be taken into account. However, we cannot simply sum the points together since the past observations are less important. The plausibility function below (2.10) calculates the total plausibility score of the activity a after i_c iterations.

(2.10)
$$plausibility(a) = \sum_{i=0}^{i_c} a_{\delta,i} * \phi^{i_c - i}$$

That is, the plausibility of a is the sum of all the points gained modulated by an inverted exponential function. The constant parameter $\phi \in (0, 1)$ modulates the speed at which the function tends to 0. Bigger it is, the longer iteration's score has an impact. The last step is to normalize the points gained by the activities with equation 2.11:

(2.11)
$$NormalizedADL = \bigcup_{i=0}^{a_i \in ADL} \frac{score(a_i)}{\sum_{j=0}^{x_j \in ADL} score(x_j)}$$

The ADL with the highest score is the one selected as currently being realized. To test this algorithm, 78 scenarios of five different daily life activities were realized at the LIARA laboratory. The results are difficult to compare with the literature since the algorithm provides a recognition hypothesis every iteration directly while the ADL is being completed. In a matter of offline recognition, the algorithm resulted in a 100% recognition. However, in real-time, the scores are lower. The Figure 2.11 below shows the online recognition rate. That score is calculated by taking the hypotheses of the algorithm from the very beginning of the activity to the end (at one guess per 200ms). The score might seem low, but three out of the five ADLs were very similar and implied the same objects. Also, from these 78 activities, 33 contained spatial errors realized by the actor.

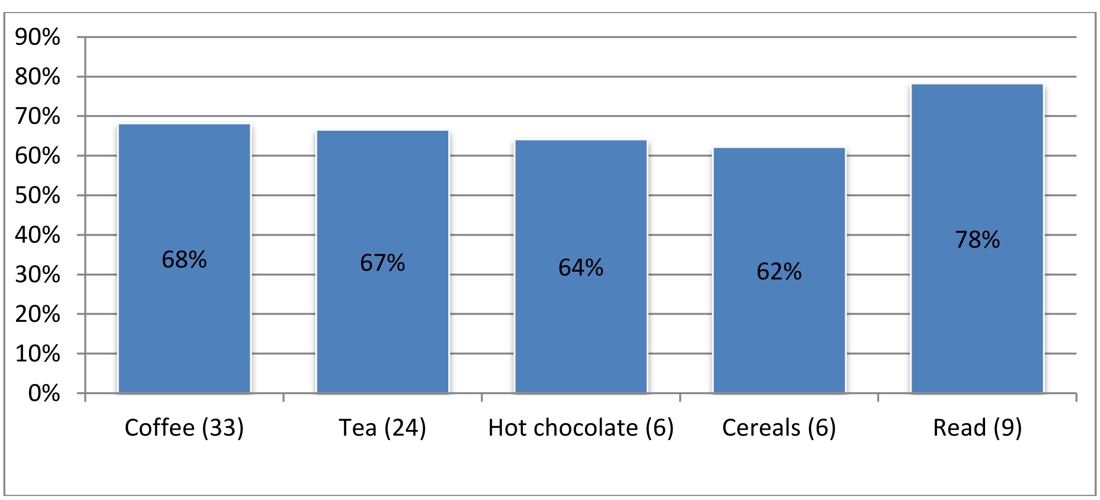


Figure 2.11: Online recognition rate and the number of time each ADL was realized between parentheses.

2.4.4 Assessment of spatial activity recognition

In this section, we have reviewed three very different spatial approaches to activity recognition. Despite their novelty, the three approaches have different drawbacks that limit their use in realistic contexts. First, RIMER and our previous model both require an expert to define the library of plan and thus, they don't provide a solution to the problem of non-data mining approaches. The chemotatic model of Riedel & al. do not suffer from this limitation. However, the model is made only for image processing and was never generalized to be used for other purposes in the best of our knowledge. Camera videos are powerful sensors, but as we will see in the Chapter 4, they may not be adapted to the assistive smart home context. Finally, all three models integrated the spatial aspect in a limited way. Even our previous model that exploited topological relationships ignores other fundamental elements such as orientation, distance, movement, etc. Further research is required on the spatial aspect related to activities.

2.5 Chapter conclusion

In this chapter, we have introduced the reader to the classical viewpoint of the artificial intelligence literature on the long-dated problem of activity recognition. These approaches, that are all non-data mining, proposed significant scientific advances, but still have not solved every problem related to that difficult task. We have reviewed the main approaches based on mathematical formalisms, the one exploiting the well-established probabilistic theory and the one trying to incorporate the spatial aspect in the reasoning process. Despite the difference between those models, commons limitations remain. First, they all suppose that the recognition agent possesses a complete and exhaustive plans' library defined by a human expert. As we have said before, it is an unrealistic assumption and thus machine learning need to be developed in that regard. Second, most of these approaches suppose that the high-level actions are directly observable, or that it is easy to infer them. That is, they suppose that the first three layers of the activity recognition task are solved or easy to perform. It is also very unrealistic, and a lot of research still needs to be conducted for this idea to become reality.

In addition to these limitations, the data collected from a smart home could be exploited for other purposes that are not often discussed. First, with data mining techniques, the models developed could enable an adaptation to the particular profile of the user without prior knowledge by a human expert. This would help in offering better assistive services. Second, the data collected from a network of smart home could be exploited to extract common patterns and new knowledge that even human experts might not possess. In conclusion, for all these reasons, it seems necessary to develop data mining models that could address the limitation of the classical literature on activity recognition. These models should introduce the spatial aspects that are fundamental to the recognition of ADLs.

CHAPTER 3

RELATED WORK ON DATA MINING

In the previous chapter, we introduced the classical approaches to activity recognition. We assessed the advances of artificial intelligence on this topic, and this led us to the conclusion that data mining research would be a good candidate to address some of the remaining challenges. We must stress out that data mining methods are not necessarily in contradiction with the classical models, but sometimes can be seen as complementary. In this chapter, we will review the main data mining techniques, and as you will see more research is needed in this area too. In particular, few spatial data mining models have been developed, and they are mainly built for Geographic Information System (GIS) and thus remain inadequate for our applicative context.

3.1 INTRODUCTION TO DATA MINING

Data mining is the set of methods and algorithms deployed for the exploration and the analysis of possibly huge databases. We introduced the general data mining process in the introduction chapter. As it was described and illustrated by the Figure 1.2, this process is generally divided into four crucial steps:

1. Collection and cleaning of the data
2. Data preparation
3. Construction of the models by mining techniques
4. Evaluation of the model(s) extracted

These four steps can be repeated a few times until the results are meaningful or until the analyst resorts that the data may not hide the knowledge he was expecting. Thus is the art of data mining, human judgment plays a role for each step and for the final decision. One important thing to understand is that the phases one, two and four are highly dependent on the application context. Not that the step three is not, but for the three other phases, *ad hoc* methods that does not generalize are often developed. In this thesis, each of these phases was addressed, and while the complete spatial data mining may not be suitable as a whole for other applicative contexts, each portion of the model is generalized and could be separately used for other purposes.

This chapter will complete the information given in the introduction about data mining and will mostly focus on the data mining phase itself. The chapter has two goals. First, reviewing and describing the three main families of algorithms: decision trees, association rules and clustering. On top of that, the second goal is to discuss the most important data mining model with an emphasis on the one integrating the spatial aspect. An assessment of their advantages and limitations is made at the end, and a conclusion will open to the future challenges of the field.

3.2 DECISIONS TREES

In the field of data mining, we generally classify all the algorithms under three main categories: decision trees, association rules and clustering. The general idea behind decision trees (DTs) is to take a large set of data and find the most discriminative properties to take classifying decisions from. In order to do that, the training set must be labeled (i.e. each entry must have the corresponding class it belongs to). In that sense, decision trees are supervised algorithms as we defined it in the introduction. From that data set, the algorithm will generally go through each attribute and choose, using a heuristic, the one that best divides the instances. It will then divide the data entries using that attribute and repeat the operation for the newly created nodes. However, it is necessary to prevent overtraining. If the DT is too modeled after the data, it might be impossible to classify new instances (unknown). The Figure 3.1 shows an over fitting versus a representative model.

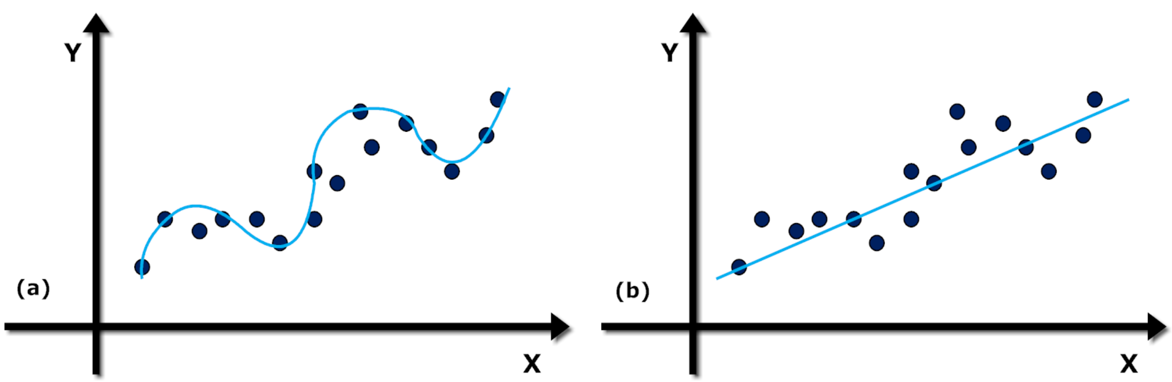


Figure 3.1: (a) Over fitting the data points. (b) A more interesting and simpler model.

To prevent over training, a decision tree classifier needs to have a stop condition. That condition can be: a maximum branching factor, all attributes are used, number of instances per node, etc. The classification of new instances is then simply performed by following the tree until reaching a leaf. In the next subsection, we will review two of the most important algorithms that are exploited to construct a decision tree.

3.2.1 ID3

The first algorithm that we wanted to present is called Iterative Dichotomiser 3 or, more commonly, ID3 [17]. This precursor of the well-known C4.5 is an algorithm used to generate a decision tree from the top to down without backtracking. To select the most useful attribute for classification, a criterion named the information gain based on the information theory is exploited. The information gain of a given attribute X with respect to the class attribute Y is the reduction in uncertainty about the value of Y when we know the value of X. In order to calculate the information gain we need to know the information entropy. If $E(S)$ is the information entropy of the set S and n is the number of different values of the attribute in S, and $f_s(i)$ is the frequency of the value i in the set S, then the information entropy is calculated according to the following formula (3.1):

(3.1)

$$E(S) = -\sum_{i=1}^{i=n} f_s(i) log_2\big(f_s(i)\big)$$

The entropy is always a number comprised between 0 and 1 inclusively. If all the examples are in the same class, the entropy of the population is nil. If there is the same number of positives and negative examples in binary classification, the entropy is maxed. The best attribute is selected based on the information gain factor that is given by the following formula (3.2):

(3.2)

$$G(S, A) = E(S) - \sum_{i=1}^{m} f_s\,(A_i)E\left(S_{A_i}\right)$$

Where $G(S, A)$ is the gain of the set S after a split over the A attribute, m refers to the number of different values of the attribute A in S, $f_s(A_i)$ is the frequency of the items possessing A_i as i^{th} value of A in S and $S_{A_i}$is a subset of S. There are three requirements for the training data of ID3 algorithm. The first one is that all of the training data objects must have common attributes, and these attributes should be previously defined. The second requirement is that the attributes' values should be clearly indicated and a value indicating a special attribute should indicate no more than one state. The third requirement is that there must be enough test cases to distinguish valid patterns from chance occurrences. The Algorithm 3.1 details the ID3 process:

Algorithm 3.1: ID3.

Input:	S learning data set; the set of attributes $A = \{a_j \epsilon \{1, \dots, p\}\}$ where p is the number of attributes remaining

If all elements in S are positive **Then**
  Add $root = positive$
  Return $root$
End
If all elements in S are negative **Then**
  Add $root = negative$
  Return $root$
End
If $A = \emptyset$ **Then**
  Add $root = negative$
  Return $root$
End

Set $a^* = arg \max_{a \epsilon A} gain(S, a)$

Set $root = a^*$
For all values v_i of a^*
    Add a branch to $root$ corresponding to v_i
    Create $S_{a^*=v_i} \subset S$
    If $S_{a^*=v_i} = \emptyset$ **Then**
        Put a leaf with the most common value of the class among S at the extremity of this branch
    Else
        Put ID3($S_{a^*=v_i}$,$A - \{a^*\}$) at the extremity of this branch
    End
End

ID3 possesses the advantage that it is fast, and it builds short trees. Nevertheless, if a small sample is tested, only one attribute at a time is tested for making a decision, and classifying continuous data may be computationally expensive. As any other DT algorithm, data may be over-fitted or over-classified by ID3. The classes created by ID3 are inductive, meaning that, given a small set of training instances, the specific classes created by ID3 are expected to work for all future instances. A limitation of ID3 is that the distribution of the unknown conditions must be the same as the test cases, and the induced classes cannot be proven to work in every case since they may classify an infinite number of instances.

3.2.1.1 Example of construction of a DT

To show the main characteristics of the construction of a decision tree with ID3, we will exploit the example dataset found on Table 3.1.

Table 3.1: Example dataset

Color	*Shape*	*Size*	*Pattern*	*Class*
Cyan	Octagonal	Small	Filled	Edible
Orange	Hexagonal	Small	Filled	Edible
Orange	Octagonal	Small	Striped	Inedible
Magenta	Hexagonal	Big	Striped	Edible
Cyan	Hexagonal	Big	Striped	Edible
Orange	Octagonal	Medium	Filled	Edible
Magenta	Pentagonal	Big	Striped	Inedible
Magenta	Octagonal	Medium	Striped	Inedible

From this dataset S, the overall entropy would be:

$$E(S) = \frac{5}{8} log_2\left(\frac{5}{8}\right) + \frac{3}{8} log_2\left(\frac{3}{8}\right) \approx 0.9544$$

To construct the tree, we would then need to calculate the information gain for each attribute. For example, the calculation for the attribute *Shape* would be:

$$G(S, Shape) = E(S) - \left(\frac{4}{8}\mathrm{E}(2,2) + \frac{3}{8}\mathrm{E}(3,0) + \frac{1}{8}\mathrm{E}(0,1)\right)$$

$$G(S, Shape) = E(S) - \left(\frac{4}{8} * 1 + \frac{3}{8} * 0 + \frac{1}{8} * 0\right)$$

$$G(S, Shape) = E(S) - 0.5 \approx 0.4544$$

Note that there are three entropy calculations made for each possible value of the attribute. For instance, the $\frac{4}{8}\mathrm{E}(2,2)$ is the part for *Octagonal* and 4 out of 8 are octagonal. The 2,2 means that two of the *Octagonal* data entries are positive (Edible) and two are negative (*Inedible*). The gain of the three others attributes would be:

$$G(S, Color) = E(S) - \left(\frac{2}{8}\mathrm{E}(2,0) + \frac{3}{8}\mathrm{E}(2,1) + \frac{3}{8}\mathrm{E}(1,2)\right) \approx 0.2657$$

$$G(S, Size) = E(S) - \left(\frac{3}{8}\mathrm{E}(2{,}1) + \frac{2}{8}\mathrm{E}(1{,}1) + \frac{3}{8}\mathrm{E}(2{,}1)\right) \approx 0.0157$$

$$G(S, Pattern) = E(S) - \left(\frac{3}{8}\mathrm{E}(3{,}0) + \frac{5}{8}\mathrm{E}(2{,}3)\right) \approx 0.3476$$

As it can be seen, the *Shape* would give the highest information gain, thus it is chosen as the root of our DT. The tree would have three branches after this first iteration as shown on Figure 3.2.

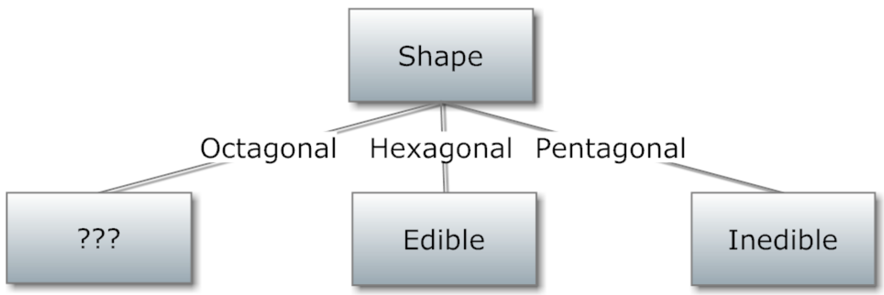


Figure 3.2: The DT after one iteration.

Now, only the *Octagonal* branch does not enable to clearly classify the population of the training set. The entropy of the octagonal subset (S_{oct}) must be calculated and then the information gain for the remaining attributes. In that case, the calculation would be:

$$G(S_{oct}, Color) = E(S_{oct}) - \left(\frac{1}{4}\mathrm{E}(1{,}0) + \frac{2}{4}\mathrm{E}(1{,}1) + \frac{1}{4}\mathrm{E}(0{,}1)\right) = 0.5$$

$$G(S_{oct}, Size) = E(S_{oct}) - \left(\frac{2}{4}\mathrm{E}(1{,}1) + \frac{2}{4}\mathrm{E}(1{,}1)\right) = 0$$

$$G(S_{oct}, Pattern) = E(S_{oct}) - \left(\frac{2}{4}\mathrm{E}(2{,}0) + \frac{2}{4}\mathrm{E}(0{,}2)\right) = 1$$

As we can see, the *Pattern* value gives a maximal information gain for the subset S_{oct} and thus it is chosen to construct the DT. The final decision tree is illustrated by the Figure 3.3.

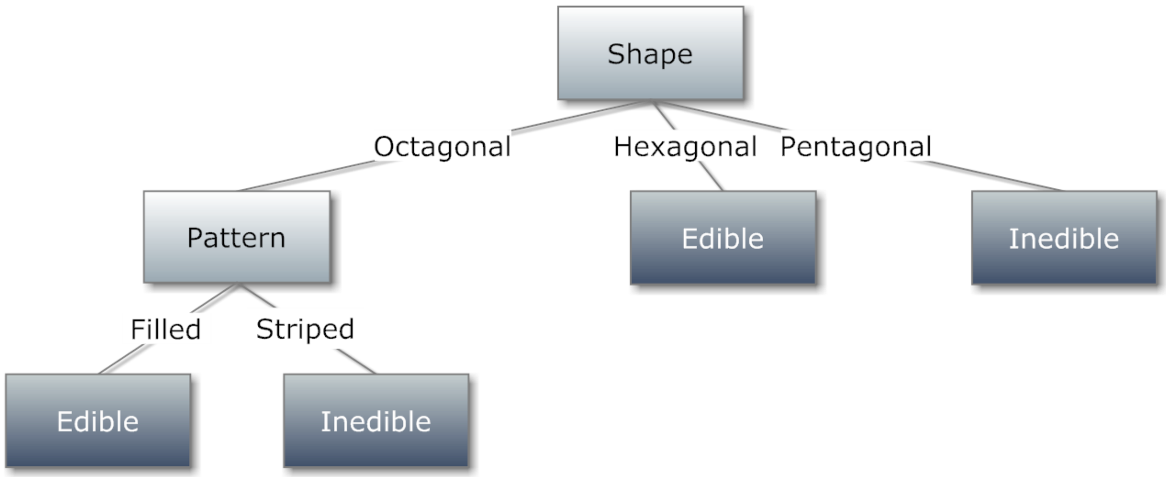


Figure 3.3: The resulting decision tree for the example dataset.

3.2.2 C4.5

The C4.5 algorithm [130] is an extension of the ID3 algorithm. The most important improvement is the capability to handle both continuous and discrete attributes. To do that, a threshold is created and the list of value is split into those which are above the threshold and those that are equal or below. Another improvement comes from its capacity to handle training data with missing attributes. In that case, the values of a missing attribute are simply not used in the gain and entropy calculations. Finally, C4.5 can backtrack on the tree to perform what is called the *pruning*. That important phase reduces the risk of over fitting the data by replacing the branches that do not help for a leaf. Due to that particular step, the C4.5 computational complexity is higher than its predecessor.

3.2.3 STANKOVSKI, DECISION TREE FOR SMART HOME

Stankovski is one of the researchers that has applied the decision trees algorithm in a smart home context [131]. As for any DT based system, a first step consisted of building a supervised dataset. In that case, the dataset contained the whereabouts of a person; interactions with appliances, duration, etc. The DT was created so the usual rules describing the normal setting leading to a particular event in the smart home could be known. The events occurring outside the normal setting were considered as abnormal behaviors, and in that case assistance could be triggered (alarm, message, etc.). To create the training dataset, heavy human expert intervention was required. After that the observations are gathered, the expert needs to specify two more data fields. For each record of observation, he needs to assign an activity and mark which records are normal (usual). The construction of the decision tree is done with ID3. The Figure 3.4 below shows a part of the decision tree built by Stankovski from a dataset of 35 examples.

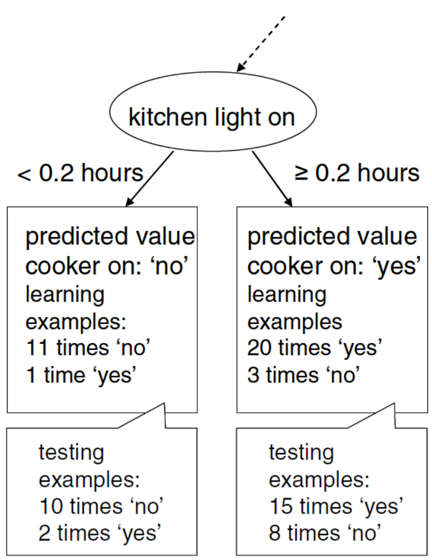


Figure 3.4: A part of a decision tree induced in [131].

3.2.4 ASSESSMENT OF DECISION TREES FOR AR

There are many advantages to use decision trees. First, they create models that are easy to understand and use from a human perspective. They are also very robust to missing data and noise (which there are a lot in smart homes). Furthermore, the classification (not the learning phase) is very fast and therefore made them well suited for online AR. There are many models of decision trees that have been exploited in activity recognition researches such as the famous ID3 and C4.5 [132] we have just seen, or Meta Decision Tree (MDT) [133]. There are two types of application of DTs in the literature. They are often used to perform low granularity AR from a very specific type of information. These works focus on the technological platform rather than on the algorithm and mostly want to demonstrate the feasibility of their idea. For example, Ravi et al. [133] wanted to recognize ADLs from only one simple accelerometer worn by a subject at the belt level. The other type use decision trees in combination with another approach of AR (usually clustering, but it can also be a classical artificial intelligence approach). The DT then acts as a post filtering classifier [134].

The main problem with DTs is that they require a large set of labeled data to perform well. If there is not enough training data, the selected attributes might be misleading and the resulting classification performance poor. Figure 3.5 shows a simple yet stunning example of what can happen if the training set is too small. DTs also do not really support data evolution; that is learning must be redone if the data change too much (new attributes, new type of values, new number range, etc.). Finally, the last but probably the most important limitation for AR is their weakness to distinguish a large number of classes within a dataset.

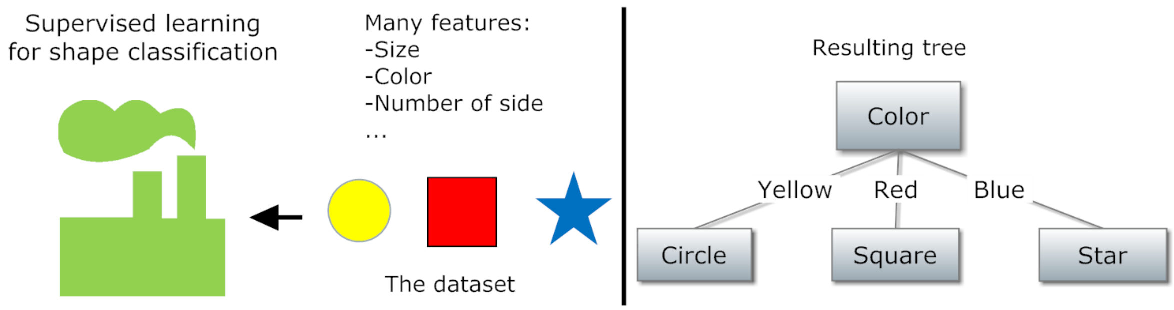


Figure 3.5: A three examples dataset for shape classification resulting in a strange DT based on the color.

3.3 ASSOCIATION RULES MINING

Association rules mining is often confused with decision trees since the latest can always be represented by a set of rules. However, in most situations, rules are different than trees. First, a large tree can often be represented by a smaller equivalent set of rules. Second, DTs try to split all classes while association rules mining considers one class at the time. An association rule is a rule of the form *condition => consequence* that aims to find relations between the data. For example, let's say that we have a dataset comprised of transactions made at Walmart by customers. We could discover a rule such as *if Sunday and Diapers => Beers*. That rule would mean that very often, when it is Sunday and someone buy diapers he will also buy beers. Association rules mining algorithms define the terms *very often* with two attributes named the support and the confidence. The first one defines the minimum frequency of both the left and right part of the rule. For example, supposes we have the item set {{A}, {B}, {AB}, {BA}, {B}, {AB}, {AB}}, the support of AB would be $Support(\{AB\}) = \frac{3}{7} \approx 43\%$. The second, the confidence, is the probability threshold of the right part being true if the left part is validated:

$$Confidence(X => Y) = p(Y|X) = \frac{p(X \cup Y)}{P(X)} = \frac{Support(\{AB\})}{Support(\{A\})} \tag{3.3}$$

Due to their inherent structure, association rules mining algorithms have been used in AR so far to learn models of activities for constraint based approaches. For example, Jakkula & Cook [108] have exploited the popular Apriori algorithm [135], that will be described in the next subsection, to extract temporal relationships characterizing the execution of ADLs. These relationships were taken from Allen's [70] intervals' calculus. They are then used to detect and predict anomalies in ADLs. Similarly, we improved our recently developed logical AR algorithm [76] exploiting topological relationships that was described on the Chapter 2 by trying to learn the models of ADLs from an algorithm named Generalized Sequential Pattern (GSP) [136]. Association rules mining is usually considered as an unsupervised learning method. Therefore, learning of the ADLs models is easier than with a decision tree. However, for AR, the learning typically requires to be performed for each individual activity. That means that there is still a need for a human expert to *label* the data, in the sense that it is impossible to automatically extract the data from the recording of the resident activities.

3.3.1 APRIORI

The first algorithm that is described is arguably the most important association rules mining algorithm. It is named Apriori and was introduced by Agrawal & al. [135]. It relies upon two principles. The first one is the research for frequent k-itemsets whose support is higher than a fixed minimum support. The second phase consists to build the association rules from the found frequent k-itemsets. A rule is retained only if its confidence is higher than a fixed minimum confidence. The Algorithm 3.2 shows the main phase one of the Apriori algorithm.

Algorithm 3.2: Apriori, first phase.

Input:	S learning data set; minimum support (σ) and confidence thresholds
Output:	Set of frequent itemsets

Fetch the item sets that whose $> \sigma \rightarrow L_1$
Set $k = 1$
Repeat
  Increase k
  From L_{k-1} finds C_k the set of frequent itemsets candidates comprising k items
  Set $C_k = L_{k-1} \times L_{k-1}$
  Set $L_k = 0$
  For all $e \in C_k$ **do**
    If $Support(e) > \sigma$ **Then**
      Add e to L_k
    End
  End
Until $L_k \neq \emptyset$

3.3.2 GENERALIZED SEQUENTIAL PATTERN

Another interesting algorithm that was also introduced by the team of Agrawal [136] is Generalized Sequential Pattern (GSP). This algorithm relies on the same foundation than Apriori but was developed to work precisely on data sets of sequence of transactions instead of simple transactional data. The meaning is that the algorithm does not only take into account the presence of items together, but also the sequential ordering. Another particularity of GSP is its capability to exploit a taxonomy by encoding it in the data set. Let's look at an example from the original paper of Agrawal. Suppose we have the sequence <(Foundation, Ringworld) (Second Foundation)> and the taxonomy shown in Figure 3.6. To exploit the said taxonomy, all is required to do is to integrate it directly in the data set: <(Foundation, Ringworld, Asimov, Nirven, Science Fiction) (Second Foundation, Asimov, Science Fiction)>. It is also possible to optimize the encoding in order to avoid the explosion of data [136].

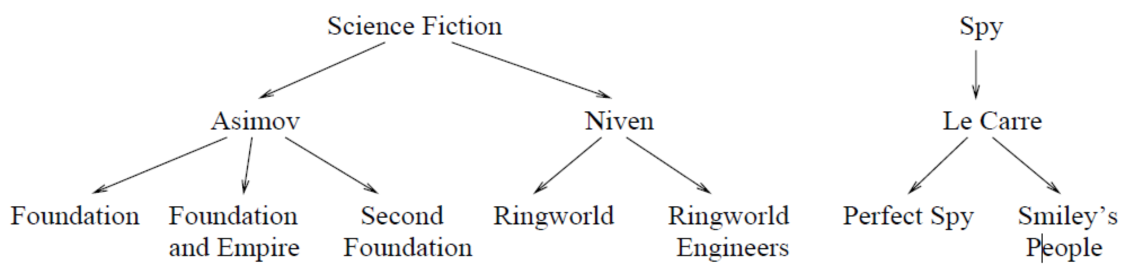


Figure 3.6: Example of taxonomy.

Another interesting part of the GSP algorithm is the pruning which is done directly on the candidate itemsets by introducing the concept of contiguous subsequence. The idea is to suppress the candidates who possess a (k-1)-sequence contiguous with a support smaller than the fixed minimum support. A subsequence contiguous *c* of *s* is a sequence for which one of those three criterions is true:

1. *c* derivates from *s* by rejecting either s_1 or s_k
2. *c* derivates from *s* by rejecting an item from a s_i which possess at least two items
3. *c* is a contiguous subsequence of c' which is a contiguous subsequence of *s*

For example, considers the set *s*=<(1, 2) (3, 4) (5) (6)>. The subsequence <(2) (3, 4) (5)>, <(1, 2) (3) (5) (6)> and <(3) (5)> are all contiguous subsequence of *s*. However, <(1,

2) (3, 4) (6)> and <(1) (5) (6)> are not. Now let's look at an example dataset to demonstrate how the pruning work within GSP algorithm. Considers the seed set consisting of those six frequent 3-sequences:

1. <(1, 2) (3)>
2. <(1, 2) (4)>
3. <(1) (3, 4)>
4. <(1, 3) (5)>
5. <(2) (3, 4)>
6. <(2) (3) (5)>

The junction step of the algorithm would lead to obtain these two frequent 4-sequence considering a support of 100%: <(1,2)(3,4)> and <(1,2)(3)(5)>. The second sequence, <(1,2)(3)(5)>, would be abandoned during the pruning because subsequence <(1)(3)(5)> is not part of L_3 (for GSP, the fourth sequence is not equivalent to <(1)(3)(5)>). In fact, this sequence is contiguous since the criterions number two is true for it. The next subsection will describe a complete smart home solution exploiting rules mining for activity recognition and activity prediction.

3.3.3 Jakkula & Cook, rules mining for SH

Jakkula & Cook developed a renowned approach of activity discovery for smart home which is based on association rules mining. Their system was built in a multi-agents [2] architecture where the agents perceive directly the state of the environment from sensor's output raw data. They collected temporal information constructed from Allen's intervals calculus presented in [70]. Their goal was to process raw data to discover frequent sequential patterns. In that case, it enables the discovery of temporal links existing between frequent events. For example, if recorded data tends to demonstrate that every time *Take Tea* happens the kettle is activated soon after, the recognition system will infer a temporal rule from Allen's thirteen relations (*Boil Water* after *Take Tea*). Supposing that a lot of training data are available, Jakkula & Cook's model works as follows. First, the temporal intervals are found using the timestamp of events and the on/off state of binary sensors. The algorithm that associates these intervals to one of Allens' relations is illustrated below (Algorithm 3.3):

Algorithm 3.3: Temporal Interval Analyzer [20].

Input: E ={set of events}

Output: Set of Allen's relation

```
Repeat
    While (E_i && E_{i+1})
        Find pair ON/OFF events in data to determine temporal range
        Read next event and find temporal range
        Associate Allen's relation between events
        Increment Event pointer
    End
Until end of input
```

The algorithm loops until all the pairs of events are compared. Between each pair, it establishes the Allen's relationship from the beginning and end markers of both events.

The second step in their model is to identify frequent activities or events that occur during a day to establish a reduced set of activities. This step is mandatory because there are too much data from smart home sensors, and many potential anomalies are just noise that should be ignored. They accomplish this task using the Apriori algorithm [108] that was described previously. In their work, Jakkula & Cook not only demonstrated that temporal relationships provide insights on patterns of resident behaviors, but also that it enhances the construction of other smart home assistance algorithms. To do so, they calculated the probability that a certain hypothetic event occurs or not, given the observed occurrence of other events temporally related. It is done from the frequency of the nine relationships out of thirteen they determined that could affect anomaly detection: *before*, *contains*, *overlaps*, *meets*, *starts*, *started-by*, *finishes*, *finished-by* and *equals*. The formula to calculate the evidence of the occurrence of an event X is given by the observation of other events (such as Y) that are temporally related (from previous learning phase). The equation 3.4 below allows such calculus:

(3.4)
$$\begin{aligned} P(X|Y) = {} & |After(Y,X)| + |During(Y,X)| + |OverlappedBy(Y,X)| \\ & + |MetBy(Y,X)| + |Starts(Y,X)| + |StartedBy(Y,X)| \\ & + |Equals(Y,X)| \;/\; |Y| \end{aligned}$$

That equation gives the likelihood of X considering Y. To combine evidence of X from multiple events that are in temporal relationship with X, we have to improve the equation. Consider the events Z, Y that had been observed in this order, the prediction of X is given by the formula $Prediction_x = P(X)$ that is calculated as follows (3.4):

(3.5)
$$\begin{aligned} P(X|Z \cup Y) &= \frac{P\big(X \cap (Z \cup Y)\big)}{P(Z \cup Y)} = P(X \cap Z) \cup \frac{P(X \cap Y)}{P(Z)} + P(Y) - P(Z \cap Y) \\ &= P(X|Z).P(Z) + P(X|Y).\frac{P(Y)}{P(Z)} + P(Y) - P(Z \cap Y) \end{aligned}$$

From the formula, anomalies can be detected and predictions can be made. If an event X as a probability approaching 1, then it is considered as most likely to occur. On the other hand, if its probability is close to 0, it will be considered as an unusual event and will be ignored from further predictions. The final step is to use an enhanced version of the *Active*

LeZi (ALZ) [137] algorithm for the prediction by adding these discovered temporal rules as input data. This predictor is sequential and employs incremental parsing and uses Markov models. It should be noted that *ALZ* improved could be used for anomaly detection. This could be done by using the prediction as input in an anomalies detection algorithm and by comparing prediction sequence with observations. Thus, if the new observation does not correspond to the expected event, an assisting sequence could be triggered. The add-on to the *Active LeZi* is shown below (algorithm 3.4):

Algorithm 3.4: Temporal Rules Enhanced prediction.

Input: Output of ALZ a, Best rules r, Temporal dataset

While $(a! = null)$
    Repeat
        Set r_1 to the first event in the first best rule
        If $(r_1 == a)$ Then
            If $(Relation! = "After")$ Then
                Calculate evidence
                If $(Evidence > (Mean + 2\ Std.Dev.))$ Then
                    Make event in the best rule as next predictor output
                Else
                    *Get next predicted event and look for their temporal relation in the temporal relations database based on the frequency.
                If the relation is after again **Then**
                    Go to * Until no more after relations found then calculate evidence
                    If high **Then** predict;
                    Else Calculate evidence and if high then predict this event based on the relation; Continue.
                End
            End
        End
    Until end of rules
End While

Following the creation of this algorithm, they have conducted experiments that can be seen in Table 3.2 below. It shows the accuracy of the observed prediction performance on real data sets and synthetic. There is a performance improvement of the prediction of activities of the resident of the intelligent environment. The main reason for a significant error rate is the small amount of data used. The search for knowledge-based temporal rules is a new area of research in intelligent habitats. Note that the use of temporal relationships provided a unique new approach for prediction.

Table 3.2: Comparison of ALZ prediction with and without temporal rules

Datasets	Percentage accuracy	Percentage error
Real (without rules)	55	45
Real (with rules)	56	44
Synthetic (without rules)	64	36
Synthetic (with rules)	69	31

3.3.4 Assessment of the association rules mining approaches

As you can see, association rules mining approaches are very interesting and more general than DT. Due to their inherent working, they are perfectly adapted to learn logical rules about activity of daily living and be exploited for AR. In fact, we also extended the spatial AR algorithm presented in Chapter 2 section 2.4.3 in order to exploit association rules mining [101]. Our goal was to exploit GSP in order to learn frequent sequence of topological relationships between objects during the realization of an activity. Those rules would then be exploited to constitute the plans library of the recognition agent. We found out that the recognition performance decreased only by 6% with the learned version. The main advantage of exploiting this method was its simplicity. However, we had to record separately each ADL several times, and in that sense, our approach suffered from the problems of supervised methods. Moreover, association rules mining usually results in an important number of trivial and non interesting rules. That is, a human usually needs to check all the extracted rules in order to find the few that could be exploited. Also, the data set must be adapted to this kind of algorithms. They are not well suited to deal with raw data from sensors and thus an ad hoc method to transform the data is usually required. Finally, the method is not working well for rare items. Due to the high dimensionality of our data, frequent patterns might not be that frequent in real contexts.

3.4 CLUSTERING

To address the issues that exist with DTs and association rule mining, many researchers aim to exploit completely unsupervised learning. Clustering could be a good solution since it can extract similar data automatically from unlabeled data. The idea behind this type of algorithm is simple. The goal is to find *clusters* in the dataset that could separate the records into a number of similar classes. A cluster is, in that context, a set of similar objects, where similarity is defined by some distance measure. The goal of the distance measure is to obtain clusters with a high intraclass similarity and a low interclass similarity. The distance measure should respect these four properties:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \; iff \; x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Among the popular known distances, here are respectively the Euclidian distance, the Manhattan distance and the Minkowski distance:

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad d(x,y) = \sum_{i=1}^{n}|x_i - y_i| \qquad d(x,y) = \sqrt[q]{\sum_{i=1}^{n}|x_i - y_i|^q}$$

Euclidian *Manhattan* *Minkowski*

The clustering problem is a difficult challenge because the attributes (or features) and their values that differentiate one cluster from another are not known. There is no data examples to tell what features differentiate objects that belong to different clusters, and as the size of the dataset increases, the number of clusters, as well as the number and type of differentiating factors might change. Moreover, there is no guide to indicate what constitutes a cluster and the success of the clustering algorithms is influenced by the presence of noise in the data, missing data, and outliers.

3.4.1 K-MEANS

The most important clustering algorithm is without a doubt the *K-Means* [109]. The goal of this algorithm is to split a dataset into *k* clusters where the value of *k* is selected beforehand by the user. The first step of the algorithm is to select *k* random data points as the center of each cluster from the data space *D* which might comprise records that are not part of the training set S. Then, the other data points (or records) are assigned to the nearest center. The third step is to compute the gravity center of each cluster. These k gravity centers are the new centers for the clusters. The algorithm then repeats until it reaches stability. The stability means that none of the data points in S change of cluster from an iteration to another or that the intraclass inertia is now smaller than a certain threshold. The Algorithm 3.5 details the K-Means process:

Algorithm 3.5: K-Means algorithm

Input:	S the dataset, k the number of clusters to create
Output:	Set of k clusters

```
Set the intraclass inertia I_w = ∞
Select k center points c_j ϵ D
Repeat
    For (j ϵ {1, ..., k})
        Set cluster G_j = ∅
    End
    For (i = 1 to |S|)
        Set j* = argmin_{j ϵ {1,...,k}} d(s_i, c_j)
        Set G_{j*} = G_{j*} ∪ s_i
    End
    For (j ϵ {1, ..., k})
        Set c_j = gravity center of G_j
    End
    Calculate I_w
Until I_w < threshold
```

To really understand the algorithm, we have to specify two concepts: the gravity center and the intraclass inertia. The center of gravity of a dataset X described by p features (attributes) is a synthetic data equal to the average a of each attributes in X:

(3.6) $$center\ of\ gravity = (a_1, a_2, \dots, a_p)$$

The inertia of a dataset X of $|X|$ records is defined by equation 3.7:

(3.7)
$$I_X = \sum_{i=1}^{|X|} d^2(x_i, g)$$

where g is the gravity center of X and x_i the i^{th} record of the dataset. The function d^2 represents the Euclidian distance. Finally, the intraclass inertia I_w is given by the following calculation (3.8):

(3.8)
$$I_w = \sum_{i=1}^{k} w_i I_i$$

where w_i is the weight of the i^{th} cluster and I_i its inertia. If the data have all the same weight, this weight is calculated by using the number of elements member of the cluster G_i and using the formula 3.9:

(3.9)
$$w_i = |G_i|/|X|$$

We will now look through a visual example of how K-Means works. Suppose that the dataset is visually represented in a Cartesian plane as shown on Figure 3.7(a). In this example, the goal is to find three clusters, therefore the parameter k is set to three. The Figure 3.7(b) shows a possible initialization for the center points of these three clusters. The records of the dataset are assigned to the nearest center.

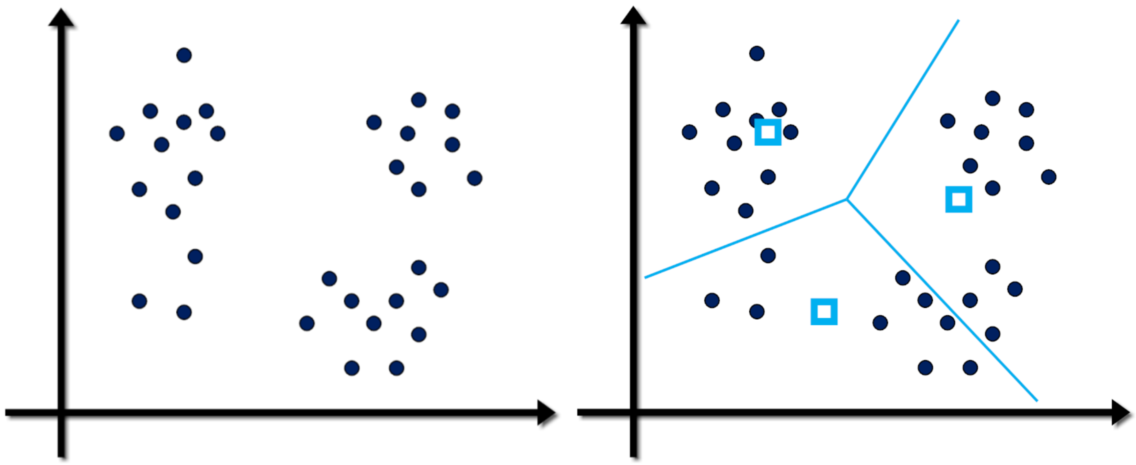


Figure 3.7 : (a) The dataset before the beginning. (b) Example of initialization with three clusters.

Then, as explained, the centers of gravity of each cluster are computed from the instances they contain. The data records are reassigned accordingly from their distance to the new centers (see Figure 3.8(a)). Finally, the process is repeated until stability is reached. The Figure 3.8(b) shows the final clusters in our example.

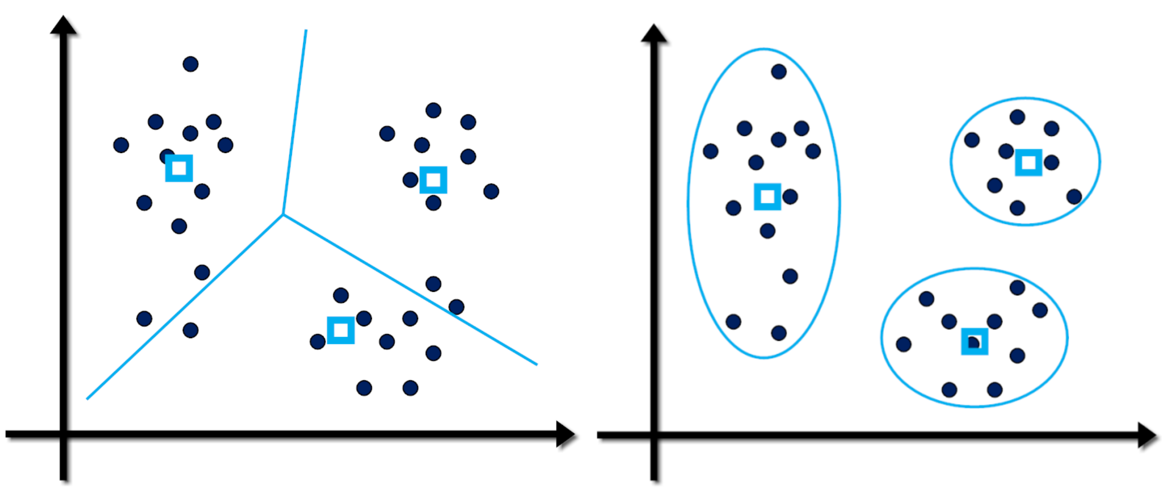


Figure 3.8: (a) New clusters after calculation of the new centers. (b) Final clusters.

The K-Means algorithm is a fast algorithm: it is considered as an algorithm of linear complexity. In fact, it is considered as one of the fastest clustering algorithms, and it usually requires a small number of iterations to find the final clusters [16]. However, there are many drawbacks to the exploitation of this algorithm. First of all, the final clusters are highly dependent on the initial centers that were selected semi randomly. Second, the algorithm converges to local minima. That is, the centers of each cluster move toward a reduction in the distance from their data but there is no guaranty that the global distance will be minimal.

An improved version named K-Means++ was introduced [16] and has for goal to select the center of the first cluster such that it has a uniform probability distribution. Then, the subsequent centers are determined such that their position is proportional to a square of a certain distance value from the first one. That enhancement improves the execution speed and also the precision of the results. However, there is a last problem that remains. To work, the user of K-Means must know the number of clusters beforehand. In the smart home context, it is usually not possible to do since we do not know in advance the number of ADLs that have been realized in the training dataset. Therefore, a clustering algorithm that does not require to specify *k* the number of clusters is required.

3.4.2 Assessment of the clustering approaches for AR

Clustering seems to be a very good opportunity for AR, but only few approaches have successfully exploited it [64]. Moreover, every time it is with a small number of low granularity activities. For example, Palme et al. [63] used completely unsupervised method that extracted the most relevant object to represent an ADL (key object). It is limited by the uniqueness requirement of the key object. In general, however, there are many reasons that explain why very few approaches exist. First of all, the complexity of information gathered from multiple sensors in smart home limit the ability of a standard clustering algorithm to spit correctly the data. In fact, an algorithm such as K-Means is not able to distinguish noise from interesting information. Second, most of the clustering algorithms need the initial number of clusters to work correctly and those that do not are very slow (high complexity). Finally, standard clustering algorithms do not fully exploit the ADL information embedded in the dataset. For example, they ignore many fundamental spatial aspects such as the topological relationships or the movement of entities.

3.5 SPATIAL DATA MINING

To conclude our journey through data mining, we believe that it would be relevant to review two data mining algorithms that have been developed for Geographical Information Systems (GIS). GIS is a field that has considerably made advanced research on spatial reasoning (SR) including spatial data mining because it deals with large spatial databases. Remember that one of the hypotheses we made in the introduction of this thesis is that the spatial aspect is fundamental to achieve the goal of recognizing human activities. We already justified this choice in Chapter 2 with the classical approaches to AR, but this time the data mining algorithms we present will show how it can improve the results of data mining in highly spatially dependent context.

3.5.1 DENSITY BASED CLUSTERING

The first spatial data mining algorithm we present is named Density-Based Spatial Clustering of Applications with Noise or simply DBSCAN [110]. It is a clustering algorithm that supports noise in the dataset. The goal of this algorithm is to address two of the problems of K-Means based algorithms. The first one is the weirdly shaped clusters that cannot be recognized with K-Means. The second is the noise that is necessarily assigned to one of the clusters with K-Means algorithm. The Figure 3.8 shows three sample dataset taken directly from an example of Ester original paper. A human can easily find the clusters just by looking at each dataset, but K-Means will give poor results on the latest two.

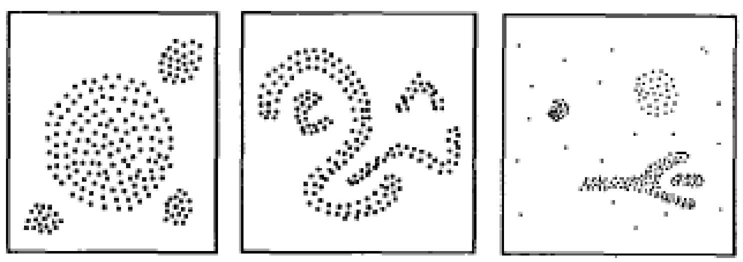


Figure 3.9: Three samples dataset from the original paper of Ester.

DBSCAN is based on four important definitions to establish the notion of dense clusters of points. The first definition is the $\epsilon - neighborhood$ of a point which come from mathematical topology:

(3.10) $$N_\epsilon(p) = \{q \in D | dist(p,q) \leq \epsilon\}$$

This equation describes that q is in the $\epsilon - neighborhood$ of p if the distance between them is smaller than ϵ. An intuitive notion of a dense cluster would be to say that each point has at least *MinPoints* in their $\epsilon - neighborhood$, but it would fail because there are core points and border points in a cluster. The second definition introduced by the team of Ester describes the notion of directly density-reachable point p from a point q:

(3.11) $$p \in N_{\epsilon}(q) \ and \ |N_{\epsilon}(q)| \geq MinPoints$$

That means that p is directly density-reachable from q if it is in its neighborhood and q is a core point (second condition). The relation is symmetric if both points are core type. Third, the point p is density-reachable from a point q if there is a chain of points $P_1, \dots, P_n, P_1 = q, P_n = p$ such that P_{i+1} is directly density-reachable from P_i. Finally, A point p is density-connected to a point q if there is a point o such that both, p and q are density-reachable from o. Using these four definitions, the authors define a dense cluster as a set of density-connected points. A special set is used to comprise the noise. It includes the points that do not belong to any cluster. Figure 3.10 shows visually the concept density reachability and density-connectivity. Algorithm 3.6 gives the general idea of the clustering from the concepts presented.

Overall, DBSCAN possesses two important advantages. First, it can be used for applications with noisy data. Second, the clusters can be of varied shape: circular, rectangular, elongated, concave, etc. There is also a Generalized version (GDBSCAN) [83] which allows to use the algorithm with different distances and with two dimensional-shapes. DBSCAN possesses some limitations for AR. It is not fast enough for online use. Additionally, it is made for static spatial information rather than changing spatial information such as what we get in smart homes. Therefore, it cannot extract the patterns of movement of the various objects in the realization of ADLs.

Algorithm 3.6: DBSCAN algorithm

Input: S the dataset, $mpts$ the minimum number of points, ϵ the neighborhood

Set $clusterID = nextID(NOISE)$
For $(i = 1\ to\ |S|)$
  Set $p = S[i]$
  If $(p.ClID = UNCLASSIFIED)$ Then
    If $(ExpendCluster(S, p, clusterID, \epsilon, mpts))$ Then
      Set $clusterID = nextID(clusterID)$
    End
  End
End

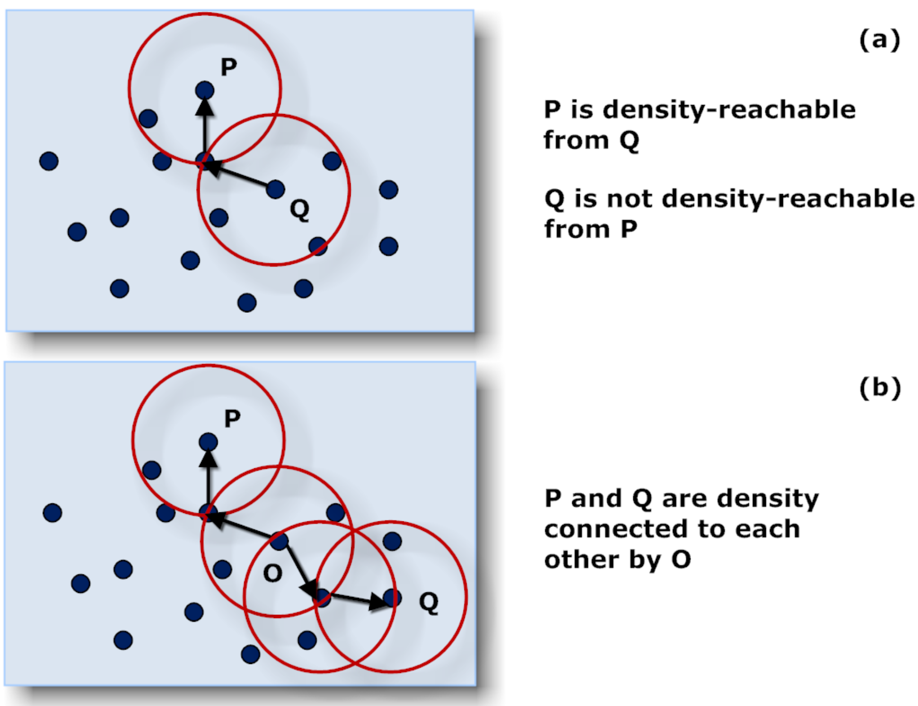


Figure 3.10: (a) Density-reachability. (b) Density-connectivity.

3.5.2 MOVING CLUSTERING

The last algorithm we present is a very different approach to clustering. The algorithm presented in [95] aims to develop a mobility based clustering for the monitoring of vehicles' crowdedness in metropolis. Their idea is to use solely the current speed of vehicles since a high mobility means low crowdedness. The main challenge of their approach is not one of clustering in fact; it is to deal with contextual information (e.g. red light, etc.) and the imprecision of GPS data. Otherwise, most of their work is based on statistical methods. There are many advantages to mobility based clustering. First, it is little sensitive to the size of the sample. Second, it does not require precise position and support errors in positioning. Finally, it naturally incorporates the mobility of different objects such as vehicles. Even through the model is not general enough to be directly applied to our problematic, we found the idea innovating and it inspired us in the quest for a new spatial data mining method. Still, mobility based clustering is new and much work remains to do to obtain interesting accuracy.

3.5.3 ASSESSMENT OF THE SPATIAL DATA MINING

While the literature on spatial data mining is rich and interesting [80], our journey into this area of research led us to grim conclusions for activity recognition. The method developed are aimed at specific context, most notably spatial database from Geographical Information Systems, and are not adapted to our context of research. First of all, the raw data from sensors is noisy and unusable as it is for those algorithms. Secondly, the distance measures remain the same as used in K-Means type of algorithms, and this distance is one of the difficult things to define in our context. Finally, the parameters required by those algorithms are near impossible to determine other than by guessing. More research is required for spatial data mining to blossom.

3.6 CHAPTER CONCLUSION

This chapter aimed to present the literature on data mining methods and to look at the research exploiting them for activity recognition inside smart environments. In particular, we have looked through the three main families of data mining algorithm: decision trees, association rules and clustering. This enabled us to assess the advantages and disadvantages of each model as well as their limitations. In particular, we have explored ID3, C4.5, Apriori, GSP, K-Means, DBSCAN and mobility based clustering. We have seen that decision trees do not scale well to highly dimensional data but remain the most comprehensible solution for a human. Association rules have been exploited a lot for activity recognition, but suffer from similar problems than supervised approaches since the learning of ADLs must usually be performed individually for each activity.

Additionally, we have seen that clustering has the best potential for the smart home context but remains avoided due to the difficulty of exploiting the popular algorithms efficiently and still obtaining good results. Finally, our exploration of the spatial data mining literature as led us to the conclusion that most algorithms, albeit powerful and interesting, are not suitable for the purpose of the goal pursued in this thesis. While they were a great inspiration to this project, particularly for the idea of exploiting movement as the main spatial criterion, researchers need to work toward the development of better adapted algorithms.

This concludes the second part of this thesis. Part three, which comprises the Chapters 4 to 7, will describe the contributions of our new spatial data mining model. But first, the next chapter will overview the experimental context and describe the technological choices that were made for this project. In particular, the smart home infrastructure of the LIARA laboratory will be described in order for the reader to fully understand the constraints and challenges this project needed to take account off.

PART III

CONTRIBUTIONS

CHAPTER 4

SMART HOME PRIMERS

As you have seen through the previous sections, the related work, while quite numerous, let open important issues that prevent the widespread adoption of smart home technologies for assistance and healthcare. Our vision of the smart home, which has been described in the introduction, is to see it as a Big Data warehouse where it is needed to design methods to extract patterns and knowledge on the resident. While we do not completely reject classical solutions, we strongly believe that they should be combined to achieve advanced data mining models. In this third part of the thesis, we will describe the path to the conception/implementation of our new model from the collection of data, passing by the high-level spatial knowledge inference (a method of aggregation) to the spatial data mining from emerging behaviors. But first, Chapter 4 serves the purpose of defining more precisely the concept of smart home. The chapter has two goals: justifying the technological choices and describing the experimental infrastructures of the *Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités* (LIARA).

4.1 WHAT IS A SMART HOME

Originally, a smart home was simply a house with automated environmental systems such as lightning and heating control features. The word smart was wildly used for any technological feature in a house that could automate simple tasks. However, nowadays almost any electrical house components can be included in the system [138], and a wide range of sensors are now within reach of public buildings and residential houses. Smart homes are used for several purposes. They can improve comfort at home, reduce energy consumption and enable automation of household chores. They can provide a better-quality entertainment by adapting their behaviors to the preferences of residents whom they learned the profile [139]. However, many scientists, like the LIARA's team, believe that the field of smart homes will reach its full potential by providing health assistance to impaired or frail persons. This very interesting application would help residents to remain autonomous at their home for an extended period of time. It would reduce the workload of natural caregivers and diminish the anxiety of families when they are not available to monitor activities. Such technology could not only help a resident directly but also produce reports for physician or allow instant monitoring with tools from Business Intelligence (BI) [59].

The LIARA team has chosen to develop assistive smart homes more precisely for persons with cognitive impairment such as Alzheimer's disease (AD). This is a particularly challenging application of smart homes because a resident can act incoherently with respect to his goal and would need to be assisted in the execution of his activities of daily living (ADLs). Moreover, it cannot be assumed that the technologies will be used correctly and therefore fail proof systems and networks are needed. While our precise context of research might be considered as one of the most challenging instances of assistive smart homes, the researchers are linked by the same issues that prevent the practical implementation of these projects in real-world. The smart homes need to be more than an aggregation of sensing technology. They need to become smarter [73] in a broad sense. It can be synthesized by the following questions: How can algorithms become able to learn and recognize the goal and the ongoing resident's activities of daily living [122, 140]? How to identify the appropriate moment for providing help and provide adapted guidance [26]? How can the system automatically adapt to the resident's habits [92, 141]? How can the network and the technology be built to support failure and to be robust [142]? In this thesis, we focus on the learning aspect related to smart home and the vision of it as a Big Data warehouse. For us, a smart home is a standard home enhanced with:

1. Sensing technologies
2. Effectors to interact with the resident
3. Computing system to think and analyze

4.2 HOW TO SELECT HARDWARE

The first step to consider for research with smart home is the kind of technology and hardware to integrate. To assess the most important criterions, we explored the literature of ongoing intelligent houses projects around the world such as MavHome [143], the eHome [144], the DOMUS [145], the gator tech smart house [146], the IATSL [147], the Institute for Infocomm Research [148] and the House_n project [149]. From these projects, we obtained important knowledge that helped to make the right choices for this thesis. One of the most important lessons that we have learned is to take into account that smart homes will not often be deployed via the construction of new houses. The conception for older houses may present challenges that make it harder to implement. However, to spread this technology to residential market, it must be implementable in existing houses, and therefore, it must be considered in our choices.

To choose the hardware, other criterions must be considered. It should be evaluated from both the user's point of view and from the system perspective. On the user side, one certainly prefers to implement a smart home at a reasonable price. Therefore, affordable and low priced technology (*Cost*) should be prioritized. On the other hand, the resident obviously does not want an unreliable system (*Robustness*). Putting bottom-of-the-range residential home automation equipments is, therefore, not an option. Rugged sensors that can withstand daily use are a better conception choice.

On the system side, it is best to have easy-to-install sensors that could be put into every housing without considerable difficulty (*Installation*). This is important to be flexible since real smart homes will be often installed in old buildings as we mentioned. Finally, the precision of the sensors and the complexity of the information which they transmit must be taken into account. It is evident why the first is important, but the justifications for the second are somewhat more obscure. Data complexity should not be ignored for two reasons. First, the objective of a sensor is to get useful information to use in artificial intelligence and data mining algorithms. If data is complex to interpret, it might be difficult to exploit it. Second, it is important for a smart home to act fast when the user needs its assistance. If the system is too slow, it will be more harmful than helpful. If the data is too complex, it is very likely that the algorithms processing it will be calculation hungry. A thumb rule for complexity is that the data of the sensors should be easily usable instantaneously on a human time scale for online services delivery. In other words: in less than a second.

The next subsections delve into two important elements that deserved more than a line or two to talk about. The first is energy efficiency, and the second is perception of the resident about sensors and other technological enhancements.

4.2.1 Energy efficiency

A thing that some researchers fail to recognize is the importance of energy management. There are many reasons why one should choose sensors and devices that minimize energy consumption. First, it matters for the resident. Of course, if researchers aim to spread smart homes adoption in the consumer market, they will need to be proven as an economically viable technology. The resident cares a lot about the cost of his electricity bill and, furthermore, he might want to reduce his environmental footprint. As a consequence, technologies that optimize electricity consumption should be prioritized and those that use disposable batteries should certainly be avoided whenever possible (no user likes to buy and changes batteries). Moreover, the latter is a big issue for assistive smart homes, since it is expected that the smart homes remain completely independent and autonomous. For example, if a smart home needs to exploit RFID technology, passive tags should be preferred over their active counterparts.

4.2.2 PERCEPTION OF THE RESIDENT ABOUT SENSORS

Another point that is often minimized is the perception of the sensors and habitat by the resident. Various researches have shown through time that residents that feel observed and invaded in their private life have a lower quality of life. In addition, if a resident suffers from a cognitive affliction, his state might worsen significantly as a consequence [150]. That is directly in contradiction with the goal we try to achieve by assisting Alzheimer's subjects with smart homes' technologies. Therefore, it is important to carefully choose the sensors and the effectors of a smart home in order to minimize the negative impact of invasiveness. Sensors should also be installed with special care to hide them from the view of the resident in the house whenever possible.

4.2.3 DESCRIPTION OF THE MAIN TYPES OF SENSORS

To select the technology to exploit in this thesis, we first compiled information on the most common types of sensors that are usually deployed in smart homes. Table 1 at the end of this subsection summarizes the main characteristics to allow a fast comparison between them. However, a few of them merit further consideration to properly evaluate their characteristics. The next subsections describe each of them and highlight both their advantages and disadvantages. Figure 4.1 shows an aggregation of images from many types of sensors.

Figure 4.1: (A) IR motion sensor - (B) Ultrasonic sensor - (C) Load cell - (D) Video camera - (E) Accelerometer - (F) Pressure mat - (G) Smart power analyzer - (H) RFID tags - (I) Microphone.

4.2.3.1 Video cameras and microphones

Smart homes are often equipped with cameras in the scientific literature [31, 84]. These offer the advantage of being able to play the role of a large number of different sensors. It is indeed the type of sensor that offers the greatest information expressivity and can enable to extract many spatial features. The cameras are available at a considerable variety of prices and most models are sufficiently robust to withstand continuous employment in a smart home. However, they are highly invasive and the processing of their data is complex. For instance, recognizing simple shapes under a wide range of lighting conditions, orientations and colors requires fairly elaborate AI algorithms [30]. One consequence is the difficulty to build a generalized smart home solution that can be straightforwardly installed in any house.

Microphones share similar characteristics with cameras. Recognizing ADLs from the sound is possible and very interesting [151, 152], but not stable enough to be used alone since a high decibel background sound prevents it from working (e.g. dishwasher). Microphones can also be rejected by the resident and/or the family.

4.2.3.2 Smart power analyzer

Our team recently explored the use of a smart power analyzer which enables the reading of electrical outlets throughout the house [153]. With a low end model, in our experiments, a developed algorithm was able to recognize all the electrical devices of the smart home and thus, many simple ADLs. These devices are available under many industrial models, but the cost is generally not under a thousand dollars. However, only one is required to cover the electrical box of an entire house. Thus, it is an inexpensive price to pay for the quantity of information it gives. The installation is simple and the sensor can withstand daily usage without failing. Its major drawback comes from the fact that each electrical device must be labeled manually (since they have a unique signature). In fact, if the resident buy a new appliance, an expert will have to enter the new signature in the knowledge base even if it is to replace an existing one from another brand.

4.2.3.3 Radio-frequency identification

Another interesting technology is the Radio-Frequency IDentification (RFID). The base cost of an RFID system is generally important (mainly due to the software/firmware on the collection module), but supplementary tags and antennas are cheap. There are two main families: passive RFID and active RFID. Active RFID can be useful to track a resident around the house but require batteries, and tags are more costly/invasive than their counterparts. It is preferable not to rely on battery-powered devices since they require punctual maintenance. Passive tags are much cheaper ranging from one or two dollars to only a few cents. They are less precise but small enough to be hidden on/in objects. The technology is very robust though the initial installation might be difficult depending on the algorithm implemented. The simplest way to use them is through proximity-based localization.

4.2.3.4 Ultrasonic sensors

Ultrasonic sensors are often used to partially replace video cameras or RFID. This technology works by emitting ultrasonic waves that hit objects and rebound back. The distance to an object can be evaluated by calculating the traveling time of the sound wave. They are often exploited for localization [154] especially for robots. These sensors work very well and can give a clear view of an environment in 3D. However, they are usually slow and so their information is unreliable in real time (not up to date). Moreover, they suffer from a line-of-sight problem. They are not very invasive due to their small size and can be hidden easily. Literature over their use in smart homes is scarce, but experiments on them indicate that they prove themselves very useful [146].

4.2.3.5 Other sensors & comparison

There is still a wide variety of types of sensors on the market that can perform tasks more or less specific that were not covered but are described in Table 4.1. For instance, infrared (IR) motion sensors are a cheap solution to track a resident in the house. However, they are very imprecise. Light sensors and others can be combined to improve precision. They are also very useful to check whether a light has been forgotten and can be exploited to optimize the energy consumption. Finally, to be able to adequately weight the information presented on Table 4.1, it is necessary to provide precisions on some criterions [38, 73, 138, 144].

Data complexity: Previously, the importance of this aspect was mentioned. However, one cannot only choose sensors with very low data complexity since it is directly linked with the expressivity of the information. Therefore, researchers must try to balance between having too complex data and lacking of information.

Cost: The cost range varies in function of two things: the products' offering and the quantity required. For example, video cameras can be cheaper than a smart power analyzer but many are required to cover the whole smart home.

Robustness: It does not take into account that some sensors would be out of reach of the resident such as cameras because behaviors are often unpredictable with an AD subject.

System Installation: It covers the initial installation aspects (e.g. calibration). For example, adding RFID tags is very easy and simple, but setting the module and the antennas correctly might be tricky.

Table 4.1: Comparison of the most common sensors. A: Best to E: Worst

	Cost	Robustness	Precision	Invasiveness	System installation	Data complexity
Accelerometers	A	A	B	A-B	B	B-C
Loadcells	B	A	B	A	B-C	B-C
Ultrasonic sensors	B-D	B	B-D	A	C-E	C-D
Temperature sensors	A	A	B	A	C	A
Flow switch	B	B	B	A	B-C	B
Light sensors	B	B	B	A	C	A
Pressure mats	C	B	A	B-C	A	A
IR motion sensors	A	A	C	B	A	A
Electromagnetic contacts	A-B	A	A	A	B	A
RFID	B-C	B	B-D	B	C-E	B-C
Smart power analyzer	B-C	A	B-C	A	A	C
Microphones	B-D	A	B	C	B-C	D
Video cameras	C-E	B-C	B	E	B-D	E

Remember that this table was built from the literature review on smart homes. Researchers usually describe precisely the drawbacks of the technology they used and price of the technology can easily be found via the companies that sell and install automation.

4.2.4 CENTRALIZED OR DECENTRALIZED PROCESSING

When designing a new smart home a choice comes from using classical centralized communication through a server or trying to decentralize the communication as in the vision of ubiquitous computing. In a centralized system, components are dumb; they transmit directly their input to a server. On the other hand, in a decentralized system, components communicate with each other trying to take decisions and collaborate on services. There are many researchers working toward the development decentralized and auto deployment system [56, 155]. These systems would be able to adapt their services to the appearance or disappearance of a new component. Their major drawback is the design complexity. Therefore, in this thesis, we first put our efforts in the creation of a working centralized solution but acknowledge the importance of more research on decentralized implementations.

4.2.5 CHOOSING THE RIGHT EFFECTORS

While it is less significant for our work, it is interesting to discuss the effectors which could be used to assist the resident during his daily life activities. Indeed, collecting information from sensors on the resident activities in the smart home is very important, but it would be for naught without methods to react or to provide assistance. As shown by

Lancioni et al. [156], the improvement in the performance achieved by participants prompted adequately by assistive technology seem to counter the growing failure in the realization of their activities, the frustration and withdrawal. Moreover, in the case of an Alzheimer's afflicted resident, good assistance can slow the progression of the disease. Smart home's literature predominantly uses verbal prompts with little knowledge of their effectiveness [157]. A deeper research revealed that it was generalized in research for assistive technology to persons with Alzheimer disease. To be effective, it is important to use prompts that are optimized with the profile of the resident and the characteristics of the tasks. For instance, a verbal prompt would have little effect on a person with Wernicke's aphasia, a language comprehension disorder. That is why part of our team is investigating the effectors' efficiency. Experience has shown that each type of prompt has a contextual specialty. While we did not exploit effectors for this thesis, we suggest reading the guidelines developed by our team for further information [26, 92].

4.3 IMPORTANT CONSIDERATION FOR SOFTWARE

Building a smart home is more than choosing which technologies it should implement. In between the sensors, the architecture and the effectors, another crucial step remains. Software applications should be included to provide an abstraction layer to work with the infrastructure. This step improves the usability by removing the need to redo the communication with all the heterogeneous components [54]. This layer can also provide students and researchers with useful services to enhance control flexibility over the smart home. Software side of a smart home has the important role of creating uniformity in the various heterogeneous technologies of the house. Traditionally, this problem is addressed by the development of middleware, which processes input from various sources and changes them into uniform output [55].

4.3.1 CALCULATION COMPLEXITY

The hardware section already covered the topic of data complexity. However, from the software point of view, it is rather more important to pay attention to it. In particular, the artificial intelligence of a smart home must be responsive and very fast to be respected by the residents and to be regarded as intelligent. A slow system would result in delayed interaction with the resident. Let just imagine the case where the system prompt an Alzheimer's patient long after he has already committed a mistake due to his impairment. It will certainly not be very helpful, and it might even confuse him more in some occasion.

It is to avoid such situations that the computational load in the design needs to be evaluated. The usage of non-computing hungry techniques must be maximized to reduce it. In fact, a smart home AI should be able to process all the information almost instantaneously on a human time scale for the important services. It is even more important in the context of assisting technology. Of course, one could argue that it is not significant to care about programs' performances since computer power is relatively inexpensive, but if calculation complexity is greater than quadratic, adding more processing power might not be enough. Moreover, it is desirable to minimize the space required for computer systems at home since it might be limited in some existing building. In this thesis, we took good care of evaluating the complexity of the proposed models for those reasons.

4.3.2 Human centric design

We already discussed in Chapter 1 that human centric design of software was a significant and continuing trend in computer science. It is even more important for assistive smart homes since the effects on its resident and the perception it gives can influence his health. In particular, cognitively impaired persons need to be challenged in order to stimulate their brain activity and slow down the degradation of their state. A smart home should encourage its resident to perform tasks by himself rather than automating them (that would be often easier to accomplish). For instance, if a window needs to be closed because it is raining, the house should influence the resident to go nearby the window and to close it.

Another point that should be considered when conceiving smart homes applications is the notion of control. It is important that the resident feels empowered by the smart home in his activities but that the utmost control remains in his hands [158]. He must not feel like the decisions are taken by the smart home, and all that remains for him is to execute them.

We must also consider the characteristics of residents targeted by our habitat in the design process. As pinpointed before, a person's profile may require different types of prompting or may need an adaptation of the effectors (e.g. a higher audio volume). For elderly, control interfaces should always be intuitive and simple. The graphical user interface (GUI) should be conceived with big buttons, a legible typeface and high-contrast colors. A software GUI built for elderly and persons with cognitive degeneration should be carefully evaluated [159] since it might greatly influence the service efficiency.

4.4 THE LIARA'S SMART HOME

Considering the discussion on the design of smart homes in the previous sections, the team of the LIARA laboratory recently conceived and implemented a new cutting-edge smart home infrastructure. It is about 100 square meters and possesses more than a hundred of different sensors and effectors. Among the sensors, there are infrared sensors, pressure mats, electromagnetic contacts, various temperature sensors, load cells, light sensors, a smart power analyzer, ultrasonic sensors and eight RFID antennas. The smart home is also equipped with many effectors, including an Apple iPad, many IP speakers around the apartment, a flat screen HD television, a home cinema theater and many lights and LEDs hidden in strategic positions. Figure 4.2 shows a cluster of images from different parts and orientations of our smart home.

The main image is the kitchen. At the bottom from left to right you can see: a tagged cup (RFID tag), the dining room, an RFID antenna, the HD television. From top right to bottom can be seen: the server, the bathroom and the library. The server is a Dell industrial blade computer, and it is the main one for the processing of the information. The smart home is also equipped with an AMX system to control multimedia hardware such as the DVD player, the television and the IP speaker. As shown on Figure 4.2, the iPad is embedded in the refrigerator. It controls the habitat for the experiments and can be used to test the equipment or to assist the resident with the help of videos when he is located near the kitchen. With regards to assistance, the television can also be remotely controlled from computer (or AMX) for that purpose. Our respective offices and a meeting room are built around the intelligent habitat. In addition, the inside of the apartment can be seen from outside by the mirror windows specially designed for experiments with subjects.

Figure 4.2: The LIARA's smart home

4.4.1 HARDWARE ARCHITECTURE

The LIARA's smart home hardware architecture follows the lessons learned by the other teams in the scientific literature. It has been conceived sturdy enough to support real intensive daily usage. For that purpose, industrial-grade material was installed while trying to keep the cost as low as possible. Hazardous situations need to be avoided as most as possible. In our architecture, the various sensors and RFID antennas are connected to four independent fault-tolerant islands. If a block fails, only sensors of that zone are affected. An APAX-5570 automaton collects the information in real time and sent it to the central computer to a SQLServer database. Thereby, this transfer hides the heterogeneity of the information coming from sensors and resolves potential communication incompatibilities between various standards exploited by the manufacturers of the sensors. By default, the central database offers no persistence on the system. The automaton simply overwrites the existing information each time. Therefore, for this thesis, we had to use a separate computer that implemented persistence to build a data warehouse. We did it on a different computer to avoid the crashing of the main system if any problem occurred during the recording as it might occur with prototype projects. Figure 4.3 shows the hardware architecture of the laboratory.

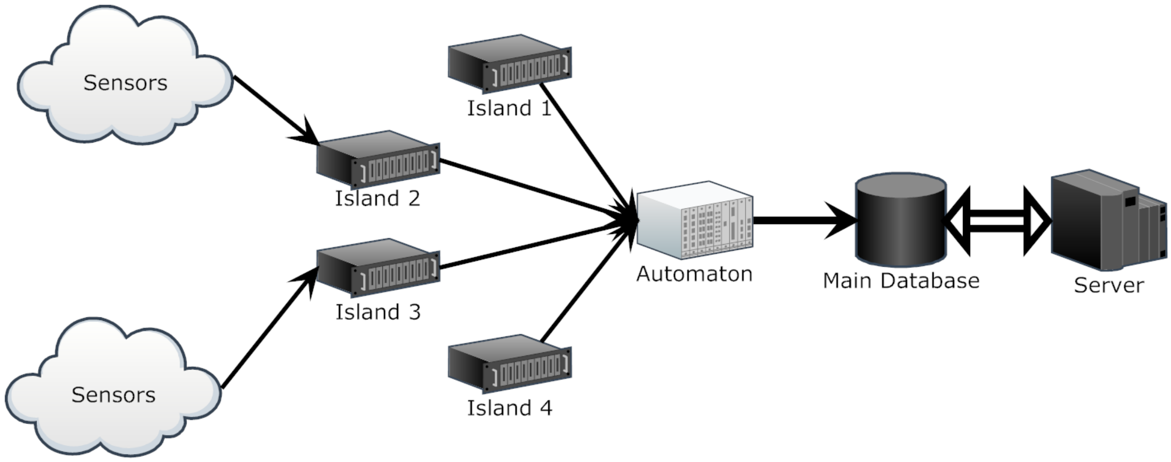


Figure 4.3: Hardware architecture of the LIARA's smart home.

4.4.2 SENSORS AND EFFECTORS

The LIARA integrates a wide range of sensing technology in order to experiment and test different approaches. First, there are classical infrared motion sensors that provide simple to process binary information about the presence of activity in a zone or not. These sensors are not only little invasive (due to their physical appearance) but also cheap to buy. There are also electromagnetic contacts that give binary information about the state of the two part of the sensor (touches, or not). Although they are wired, they can be completely hidden from the user's view due to their very small size. They are used mostly on doors and panels. Moreover, they are connected on an island which is hidden nearby. While the electromagnetic contacts must be wired, the island can be wireless. On few strategic locations, pressure mats were integrated. They give a binary information (pressed, or not).

Among other types of sensors, we use light detection, temperature and RFID technology but no video camera or look alike technology. That is because cameras are too much invasive and almost always rejected by people [160]. That is true even when the residents are told that only the system will ever access the image. Moreover, computer vision is far from the capacity to obtain all the information from a complex video camera output in reasonable computational time [46]. Some researchers have drawn attention toward the exploitation of the Microsoft Kinect to replace cameras [161], but it is still not easy to process the output data. Moreover, to reduce invasiveness it would have to be limited to a usage in the living room (on the television) which is not a highly interesting location for assistive technology. Kinect should be used for specific services, and the smart home should not be dependent on that more than on camera videos.

4.4.2.1 RFID Technology

The most important technology installed at the LIARA that was used for this thesis is the RFID system. We needed to obtain spatial information and since cameras were out of question, we opted for RFID to accomplish this task. It is not an easy task to choose the right set of RFID and configure them for best performance. We chose to use passive RFID tags since we needed to put them on everyday life objects and active one are too big for that. Indeed, as you will see in the remaining of this thesis, the model developed relies on the extraction of movement patterns from the objects during the realization of an ADL. Our need was thus to be able to not only localize the objects in the smart home, but also to infer high level spatial knowledge from the same objects.

The main advantages of passive tags are that they are cheap to buy (often less than a dollar) and require no other power than a radio pulse from a nearby antenna. Of course, when comparing to their active counterparts that use their own power to emit and that are always awake, passive tags have a reduced range and precision. Nevertheless, with proper adjustment, they can give good results, and they are robust (even washables). RFID will be discussed in more details in the Chapter 5 when presenting the new localization model. There are other technologies integrated at the LIARA such as the ultrasonic sensors or the smart power analyzer, but they were not exploited in this project. The ultrasonic sensors are usually used for similar purposes than RFID, but we preferred the latter over the former since we wanted to track object in real time.

4.4.3 SOFTWARE

Following the discussion of the LIARA's smart home hardware, it is only natural to engage a discussion on the software that is implemented. An application was designed to control the smart home to enhance flexibility and robustness. This software reads the database in real time and copies the data in a second identical database (AIDB) for the communication with AI processes. This is important since it protects the real data from being modified from third party users (malfunctioning programs, students doing experiments, etc.). Nevertheless, the reason this was implemented is to allow easy rerouting of the data source. As a consequence, the source could be changed without the third party applications ever noticing. It would also work as the opposite: route the main data to another place. Besides, that multi layered architecture allows to add different autonomous modules that can provide services to transform raw data in high level information. The software architecture can be seen on Figure 4.4.

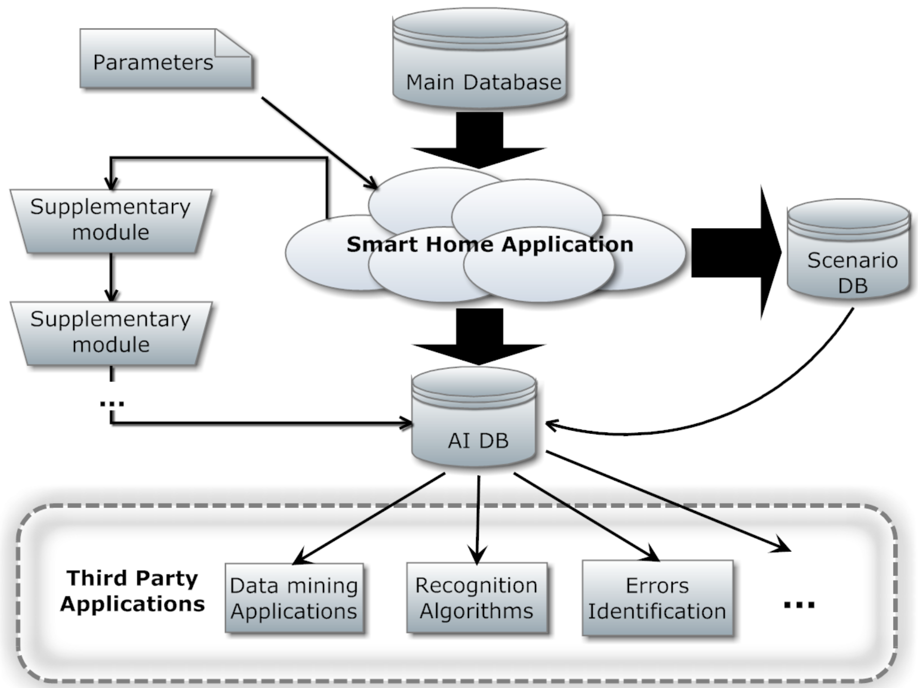


Figure 4.4: Software architecture of the LIARA's smart home.

In order to facilitate testing, a smart home visualization software was developed by the team and was exploited throughout this project. A screen shot of this software showing the overall smart home can be seen on Figure 4.5. The graphical interface of this software allows us to see different parts of the smart home or the overall picture. In each of these interfaces, we can see the state of many sensors such as infrared sensors, light sensors, etc. We also can see an approximate position of the objects in the smart home (rounded rectangle, proximity based algorithm [162]) and the current position of the resident (in front of the kitchen counter on the right part of Figure 4.5). These functionalities are very useful when conducting experiments since they allow analyzing what went wrong by reproducing sensors' activation and double checking if the material works properly. In addition, it allows manual testing of effectors of the smart home, including the television, the oven and the audio system.

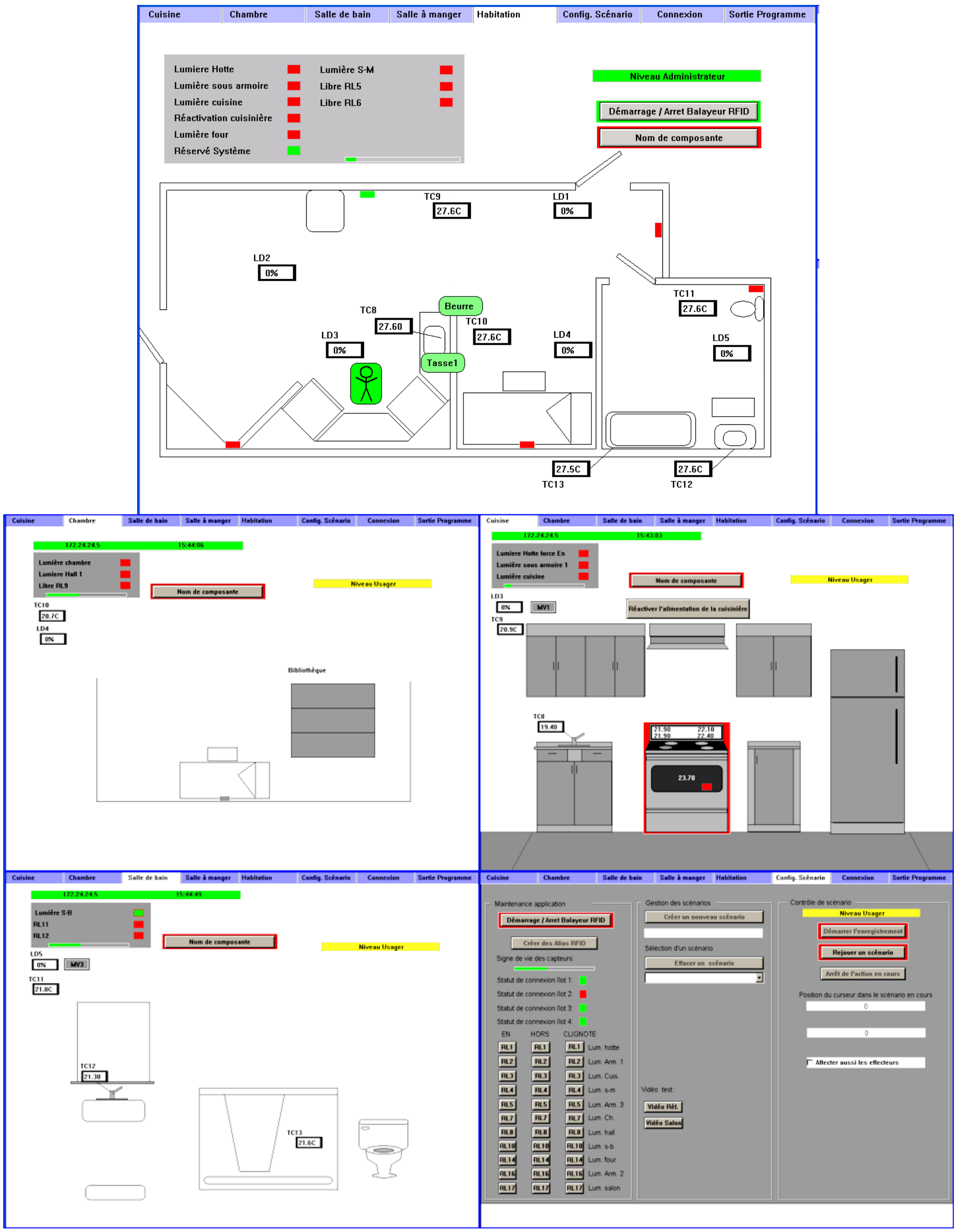


Figure 4.5: The LIARA's demonstration and diagnostic software.

On a more practical side, the application also allows to remotely control the smart home from a tablet pc. It is particularly useful when the experiments are done on a remote computer, and we need to control the smart home in real time during the said experiments.

4.4.3.1 Tracking the resident

One of the most active research area of smart homes is the localization/tracking in real time of the resident in the house. To begin with, many solutions use wearable devices for this purpose [46]. It is unrealistic to expect the resident to always wear them, especially if he is afflicted by a cognitive impairment. The LIARA team created a simple tracking AI agent that is primarily based on motion sensors, which are moderately slow and cannot cover every part of a zone (there are blind spots). The consequence is that we cannot always locate the resident with a hundred percent certainty. However, it is not necessary. We improve the certainty when an activity is detected in the smart home by considering every sensor's activation in the house. If the system loses track of the resident, it considers that he has not moved from the last place he was located. In addition, for most smart homes applications, approximate position is enough (at the scale of large part of room). This is why our system is divided into logical zones for this service. While it is not a contribution of this thesis, it seemed important to give a word about it since some tests exploited the positions of the resident.

4.5 SPATIAL DATA MINING MODEL

The Chapter 4 now arrives to a conclusion, and the remaining three chapters of the part three of this thesis will develop step by step our spatial data mining model. It seems therefore appropriate to take a moment in order to describe briefly that model. The Chapter 5 explains how we exploited the passive RFID technology to localize daily life objects in real-time. It constitutes the first fundamental step to achieve spatial data mining since from this step we can create a data warehouse. The Chapter 6 describes how to prepare the dataset for the final step by aggregating the positions of the objects into few high-level gestures or atomic directions. Finally, the Chapter 7 presents an extension to the flocking algorithm [163] in order to exploit it as a clustering method for the spatial dataset created. All the experiments that are described through these chapters have been conducted using the infrastructure described. Figure 4.6 illustrates the complete spatial data mining model developed.

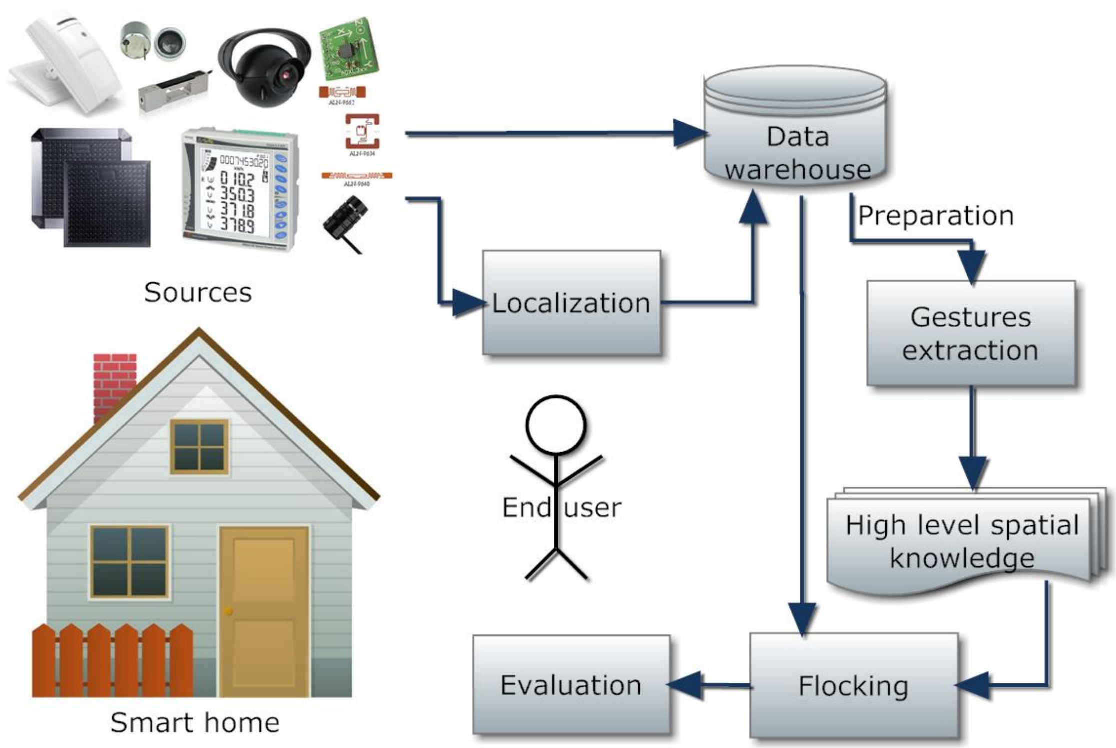


Figure 4.6: The overall spatial data mining model

4.5.1 ILLUSTRATION OF THE COMPLETE SPATIAL DATA MINING

In order for the reader to be able to fully grasp the model which is described through the remaining chapter, we will now look at a complete example of the life cycle of a simple object which is tagged. Let us say that the resident is currently beginning the ADL *PrepareCoffee* and that he is moving the cup from the position $(0,0)$ to the position $(10,0)$. Our cup is equipped with four passive RFID tags and four antennas in the kitchen are currently seeing these tags. Supposing that there are 10 iterations of the system during the movement, the list of real positions could be $L_r = [(0,0), (1,0),...,(10,0)]$. For each of those real positions, we observe sixteen Received Signal Strength Indications (RSSIs). To localize the object from RSSI, we first select the strongest RSSI for each antenna since it is most probably from the tag directly facing the said antenna. From those four RSSIs, we apply filters that improve their value and stability over time. Then, the RSSI values are transformed into distance, which is then used to perform the trilateration. The trilateration always gives more than one potential position and thus a special algorithm helps in choosing the best.

The trilateration allows to obtain a list of observed positions L_o which are within an ε distance of the real positions. We then proceed to the inference of a simple movement information. To do so, we exploit a recursive algorithm that we will explain in detail later, but for now, let suppose that we already have segmented the data and that we know only one direction can be extracted from that set of positions. The first test that is made is if the smallest enclosing circle covering the list L_o is bigger than the average error ε. If it was not, there would be no movement inferred. Then, a linear regression is computed on the list along with the correlation coefficient. If the correlation is high enough, the equation extracted is put in a qualitative spatial reasoning framework to transform it into a single basic direction. In our case, let us suppose the direction is *East*.

At this point, the set of ten positions has been transformed into one single movement. That movement would be put into a training dataset for the next and final step. Let suppose that we have a complete dataset of such basic movement information and proceed to the clustering. To extract the clusters representing the ADLs, our algorithm extends the Flocking by adding two similarity function that enables simple agents moving freely in an environment to group with the agents who are similar and flee those that are too different. The similarity function needs to be adapted for the movement information. The movements are compared with a neighborhood graph which can be automatically generated. Finally, the clustering is done simply by letting the agents evolve in the environment for a time.

4.6 CHAPTER CONCLUSION

The Chapter 4 which is ending had for goal to introduce the reader to the technologies integrated inside smart homes. As it was shown, many criterions must be considered to choose sensors and effectors to exploit. In particular, low cost is necessary but without sacrificing robustness because the house needs a maximal autonomy. The expressivity of the collected information is also an important criterion to improve services and activity recognition, but too complex data might hamper the performance of the system. Finally, above all, it is necessary to integrate technology that will limit the intrusiveness and the modification of the resident's home.

The second goal of this chapter was to describe the experimental infrastructure of the LIARA laboratory. As you have seen, the prototyping smart home is following the lessons learned by other research teams in the world. Nevertheless, since the goal of the smart home is to act as a prototyping infrastructure, most types of sensors discussed were installed except for video cameras. However, for this thesis, we chose to not exploit ultrasonic sensors that give similar information as RFID, and we chose not to exploit the smart power analyzer which is currently under study by other members of the team. In future work, the integration of the smart power analyzer could constitute an interesting enhancement to the developed model.

CHAPTER 5

EXPLOITING PASSIVE RFID TECHNOLOGY

In the previous chapter, we introduced the reader to the smart home infrastructure and technology exploited for this thesis. The first step of the spatial data mining model is to collect the information from the various data sources in order to construct a Big Data warehouse. This chapter covers the particularity of this first step in the data mining process. For the most part of the sensors, there is nothing else to do than collect the raw data and directly store it in a database. However, in order to exploit the spatial aspects, we need to perform a transformation on a part of the data. As we discussed, the goal is to explore the importance of the *movement* of objects in daily life activities. Our hypothesis was that patterns can be found to help identifying uniquely each ADLs. To extract movement information, we first need to be able to obtain the live positions of the objects in the smart home. To be able to do so, we developed a novel elliptical trilateration algorithm that exploits passive RFID technology.

The remainder of this chapter is divided as follows. The next section describes RFID technology. Then, the challenges related to indoor localization are discussed with a brief presentation of the literature on the subject. This presentation has for goal to demonstrate how actual and critical this problem is. Thereafter, the section 5.3 discusses the method developed to improve the raw data collected before performing the trilateration. The section 5.4 presents the new elliptical trilateration model and a set of experiments conducted at the LIARA's smart home. Finally, a small conclusion will discuss the limits and the future work related to this first contribution.

5.1 PASSIVE RFID TECHNOLOGY

The RFID technology is used extensively in some industries such as retail business to track goods in big warehouse or in the shipment business to allow users to follow the delivery of their package in real time. However, in research, it is primarily robotics domain which has served to advance the localization techniques [164, 165]. This technology is also increasingly used in smart homes [166] but most researchers consider the resulting data like any other whether it is to perform recognition of activities or to extract knowledge with data mining techniques. Whatever the field of application, to track people or objects, everyone would benefit from a better exploitation of this technology for localization.

Typically, a RFID system consists of three elements: radio frequency tags, at least one RF antenna and a data collection module. The system works as follows. First, the RF antenna emits a radio wave. Then, if a tag is located within its coverage area, the tag intercepts the signal, and its internal chip retransmits a signal to the antenna containing its unique identification code and sometime other information. The transmitting antenna receives this new signal, and it returns the information to the collection module. Radio frequency tags are subdivided into three families: active, semi-active and passive. Active and semi-active tags are battery powered and often have an internal erasable memory. Therefore, an active system can transmit low-power RF emission, and tags remain able to meet with high-level signals. On the other hand, passive RFID tags are in a dormant state and wake up when receiving power from a RF wave emitted from nearby antennas. They then use that same energy to power their inner chip and send a RF incorporating their unique ID (see Figure 5.1).

Figure 5.1: Passive RFID system. Farther tags receive less power and then are harder to detect.

Because of that internal power, active tags achieve a much higher range and a reliable accuracy. Moreover, the localization systems for these technologies are considerably more advanced. For instance, Hekimian-Williams et al. [167] have obtained very precise results (millimeters range) by using software coupled with accurate clocks to estimate the phase difference from signals received from two separate antennas. Nevertheless, passive RFID seems more promising for smart environments over the long term because of many advantages. First, passive tags have a technically unlimited lifespan and do not need external power supply. Therefore, a system relying on them will require little maintenance and will be more autonomous. Second, they are much smaller than their active counterparts and thus reduce intrusiveness. Consequently, it is possible to embed passive RFID tags in everyday life objects such as cups, plates, etc. Finally, passive tags cost generally only a few pennies while active and semi-active tags are in a much higher bracket of price. The Figure 5.2 shows examples of tagged objects in our smart home.

Figure 5.2: Daily life objects mounted with washable passive RFID tags.

5.2 CHALLENGES RELATED TO LOCALIZATION

The exploitation of RFID technology for localization offers many challenges that we tried to address in this thesis. We already discussed the importance of choosing hardware carefully, but another aspect come to play for localization with passive RFID; the physical configuration of such a system must be done carefully. The antennas position might determine which techniques can be exploited for the localization. Also, one must try to avoid many antennas to broadcast at the same time on the same frequency. Each receiver usually implements a time slice method to avoid interferences between its antennas, but antennas from different receivers could be an issue. In the related work section, a few word of advice is given on the placement of the antennas for each family of localization approaches.

In addition to the physical configuration of antennas, it is important to select the tags carefully for good localization results. In order to minimize the problem related to a variation of receptivity, we prefer to use only one kind of tag for all objects. We suggest testing the various types of tags and retaining those that limit the variation of the signal strength. The type of tag is not the only thing that might decrease the precision/accuracy of localization. Even among tags that are technically identical, sometimes the sensitivity is very different [168]. To address that problematic behavior, we propose taking care of testing every tag before installing them on the objects and eliminate those that are too far from the average sensitivity. This step should not take a lot of time and should greatly increase the overall accuracy. Also, if one wishes to use precise localization (such as trilateration), it might be interesting and profitable to put more than one tag on each object. In fact, the main issue of localization is the bad angles or arrival of radio wave on a tag. Consequently, covering more angles should ensure a better quality of information.

Finally, there are two other very important challenges that we tried to address in the model proposed by this thesis: the false readings and the high variation of Received Signal Strength Indication (RSSI). These two challenges are described in details along with the proposed solution later in this chapter.

5.2.1 RELATED WORK ON LOCALIZATION

Over the years, the question of localizing entities in noisy smart environments has attracted many researchers resulting in hundreds of scientific publications which cover diverse topics and technologies. Despite this fact, indoor localization of objects is still a challenging issue that could find applications in several areas. In particular, techniques related to RFID technology have been blossoming in the last few years [169]. However, due to the inherent imprecision, positioning and tracking with RFID is still very hard to achieve. That explains why many researchers explored hybrid approaches based on ultrasonic sensors, accelerometers, cameras and LEDs [170]. For example, Addlesee et al. [171] developed a successful system that relies solely on ultrasonic sensors. In a controlled environment, it achieves a respectable precision of (≈3 cm) which places it among the most precise indoor localization systems. The Table 5.1 presents a summary of the main localization systems found in the literature for a quick reference.

While non-RFID and hybrid approaches usually give better performances than pure RFID localization they are arguably less appropriate in the smart home context. First, they are more costly than RFID approaches. Second, they rely on technologies that suffer from high intrusiveness (cameras are particularly problematic in many cases). Third, they generally impose line-of-sight constraints that radio-frequency avoids. Fourth, none of these systems offer robustness comparable to passive RFID tags. For example, they cannot be put into a dishwasher, and they often need batteries. They are also slower and harder to install than simple RFID methods. Finally, these technologies are too cumbersome for objects tracking inside building. The remaining of this subsection presents three large families of approaches for indoor localization with RFID technology.

Table 5.1. A summary of few localization systems

Authors	Technology	Technique	Precision (cm)
Addlesee et al. [172]	Ultrasonic	Reference receivers	3
Choi B. et al. [165]	Ultrasonic/RFID		1.6-2.4
Chen et al. [173]	ZigBee	Trilateration/RSSI	119
Jin et al. [174]	Active RFID	Reference tags	72
Fu et al. [175]	Active RFID	Trilateration/RSSI	200-300
Hekimian-Williams et al. [167]	Active RFID	Phase difference	≈1
Hähnel et al. [176]	Passive RFID	Reference tags	100-140
Zhang et al. [177]	Passive RFID	Direction of Arrival	100
Vorst et al. [178]	Passive RFID	Reference tags	20-26
Joho et al. [179]	Passive RFID	Reference tags	35.5
Chawla et al. [168]	Passive RFID	Reference tags	18-35
Lei et al. [180]	Passive RFID	Reference tags	20

5.2.1.1 Proximity based localization

The proximity based method uses an intuitive strategy to track the tags in a smart environment [162]. The idea behind this technique is to deploy a large number of antennas and to calibrate their range in order to reduce as much as possible the overlapping. An object that enters an antenna detection zone is assumed to be at the same position than this antenna. When multiple antennas detect an object, the position is assumed to be at the position of the antenna that receives the most powerful signal. Therefore, the precision of the localization is proportional to the increase of the number of antennas and to the decrease of their reception range. Although apparently very basic, this technique works very well and is currently the most robust method to exploit for objects tracking with passive RFID tags. Many authors have worked on the proximity method to localize entities in a smart environment. In particular, some have used different technologies than RFID to do so (e.g. infrared). In fact, proximity methods exist in a wide range of precision, technology and coverage. Some of them extend the technique by using sophisticated filters or algorithms. One worthy of mention is the WiFi based method of Youssef et al. [181]. The idea behind their efforts was to be able to track a person in an environment equipped only of WiFi access points and stations. To do so, they monitored the RSSI and analyzed the changes in the environment to correlate them with the person moving from room to room. Their concept is very promising and possesses the advantages of using cheap technology that is often already installed in working environment. Moreover, it allows following a person without the needs of a wearable technology. In that view, it is similar to an infrared tracking system, but with a higher precision.

This method was also implemented at the LIARA smart home prior to the research we conducted for this thesis which led us to a more precise localization algorithm. To use it, the antennas were configured to make them cover a circular area of approximately 1 meter of diameter. The Figure 5.3 shows their normal coverage and the rough False Positive Reading (FPR) range on an aerial map of the infrastructure. The shortcoming of this method is the lack of precision of the information which does not enable to extract much knowledge. Moreover, as it can be seen on Figure 5.3, eight antennas only cover less than 10% of our smart home. Consequently, one would need to deploy an enormous number of antennas to cover the whole area of any big smart environment. The cost and the difficulty of configuration would rise accordingly.

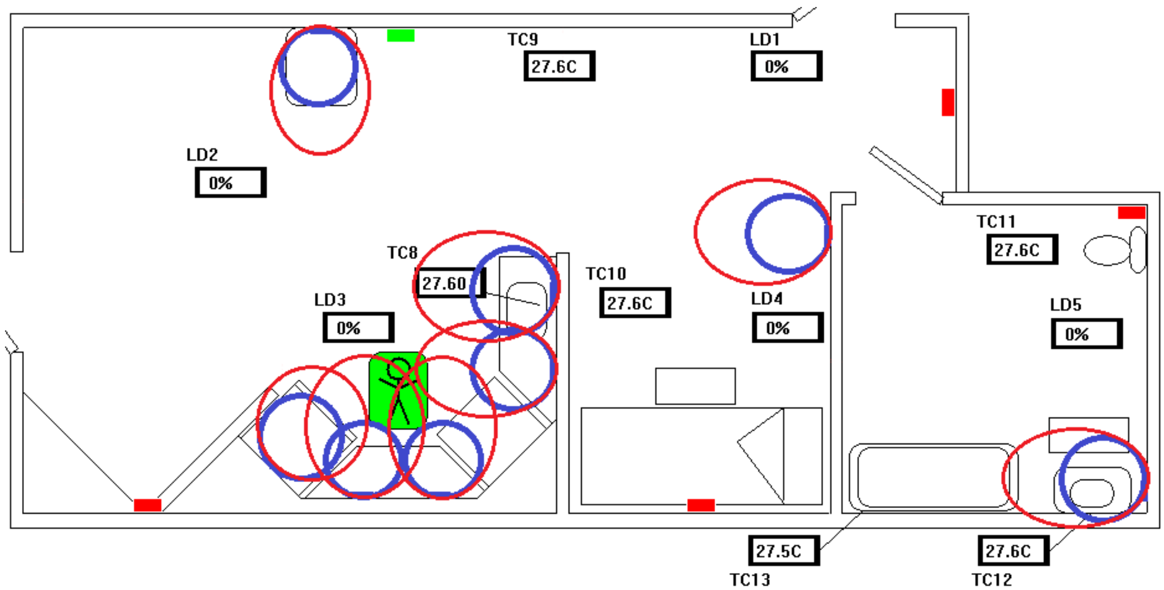


Figure 5.3: The antennas approximate emission/reception bubble in proximity mode (blue). Red →false positive readings area (tags sometime appear but should not).

5.2.1.2 Reference tags based localization

There are a large number of positioning approaches based upon the use of reference tags. They arise directly or indirectly from the well-known LANDMARC system [182]. The idea of reference tags is to fix a matrix of tags on the ground and storing their real position into a knowledge base for the localization algorithm. Using these known positions, the localization algorithm can then correct the RSSI of the tag to localize by comparing it to the reference tags. The Figure 5.4 illustrates the idea behind the technique:

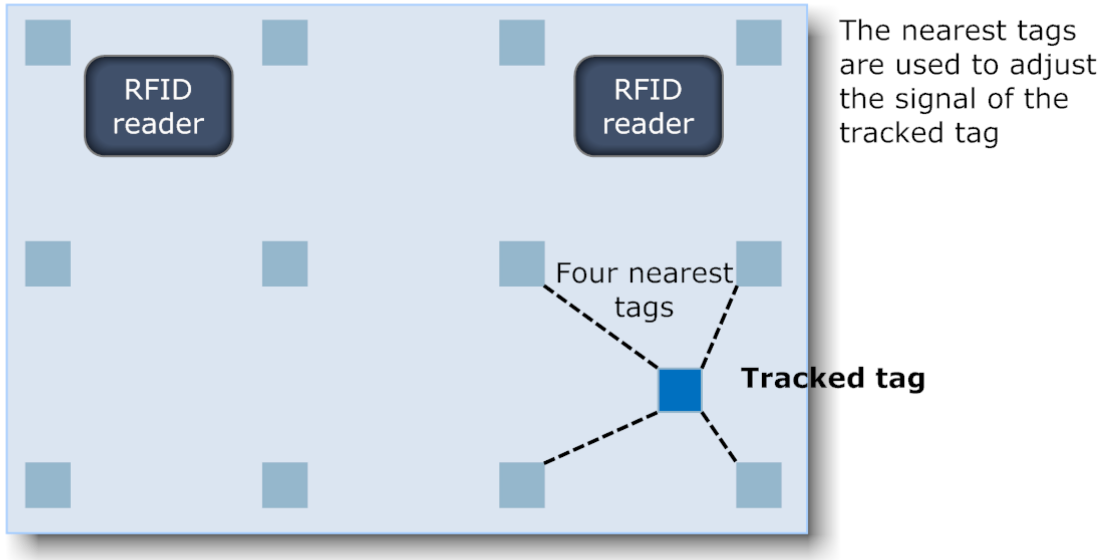


Figure 5.4: The idea behind the reference tags.

Vorst et al. [183] implemented that concept. Their model uses passive RFID tags and an onboard reader to localize mobile objects in an environment. A prerequisite learning step is required to define a probabilistic model. This model is exploited with a particle filter (PF) technique, which estimates the position. It achieves a precision of 20-26 cm. The major drawback is the relatively high computational cost (at least for real-time tracking). Lei et al. [184] addressed this issue by combining PF with weighted centroid localization. They switch between the two methods depending on the estimated velocity of the tracked object. In ideal condition, they localize an antenna with an average error of 20 cm while greatly increasing the speed of the process. Another model, from Joho et al. [179], use reference tags in combination with different metrics. In particular, it is based on both the RSSI and the antennas' orientation to get an average localization error of 35 cm. Chawla & Robins [168] developed a model based on the variation of antenna power to estimate the distance of nearby reference tags. They incrementally adjust the antenna decibel until the tag is in range. Thereafter, they use many tags' distance from the antenna to localize a mobile robot. Their approach yields an accuracy varying from 18 to 35 cm.

Some of these approaches provide very good results; more than enough to exploit them for smart environments. However, they all rely on the large deployment of tags of references. While it is a fairly good solution for robot localization, it is not always feasible to deploy them in smart environments context. Finally, most of the previously techniques localize antennas instead of tags. Antennas are much too big to be bundled on objects. Therefore, it is not an interesting solution for the objects tracking which is needed for the deployment of our spatial data mining model.

5.2.1.3 Trilateration based localization

The trilateration has been largely ignored in the scientific literature despite the simplicity and the potential applications it has. This is mainly because this technique is quite challenging to use with noisy and imprecise information. Despite this, it is the technique that we extended in this thesis. The basic idea behind this technique is to find the distance between the object to localize an at least three antennas. Thereafter, virtual circle can be traced using each distance as the radius and their intersection is the position of the object (more details in section 5.4). A recent instance of an RFID localization system based on this technique is the approach of Kim & Kim [185]. They performed a classical trilateration calculus from active tags by using the time of arrival of the signal to calculate the distance from each antenna.

Their contribution is from the introduction of a circular polarization antenna and a positioning filter that enabled them to achieve a meter range precision. Another worth mentioning is the approach of Chen et al. [186] that perform trilateration with a different radio-frequency technology (ZigBee). They developed a fuzzy inference engine with one variable that correlates the RSSI of an object transmitter to the distance separating it from a receiver. They achieved a precision of 119 cm.

While these two systems are interesting, their precision is far from being sufficient for the purpose of the spatial data mining model developed in this thesis. Indeed, we aim at exploiting the movement patterns of objects in daily life activities. With that precision, it is almost impossible to extract significant movement patterns in most ADLs.

5.3 IMPROVING THE BASIC INFORMATION GATHERING

To develop a new localization algorithm, the first challenge we encountered is situated at the basic step of the collection of the raw signal from the antennas. Due to the nature of the system operation, it is very common to obtain False-Negative Readings (FNRs). A FNR occurs when a tag is in the antenna coverage area but is not detected during a certain period of time. This type of problem happens in all the passive RFID systems. However, it is slightly more frequent on inexpensive systems. Brusey & al. [187] identified three reasons to explain this situation:

- The reader can fail to see all tags for a certain time due to an unknown internal problem

- The radio waves emitted from more than one tags may collide
- An interference might occur due to environmental emissions or due to surrounding metal shielding

There is also the opposite situation. Although this is much less common, it happens to detect tags that are not in the normal area of the antenna. These are called False-Positive Readings (FPRs). In a smart home environment, this translates as an object that might be stored somewhere (in the cabinet, for instance) and the signal that gets stronger because of an uncontrollable event for a period of time. For example, a person may use a metal object (such as a kettle) which enables the signal to rebound or travel farther than usual. This could cause a localization algorithm to interpret that an object has moved if not handled correctly.

5.3.1 ITERATION BASED FILTER

We are not the first to try to address the problem of FNRs. From the review of literature on passive RFID localization systems, we found an interesting solution that inspired [187]. The solution is a time filter based on the general rule that if an object is expected to be in an antenna range, but is not, it is considered as not present only after no detection has occurred for a period of time. That interval of time needs to be carefully tweaked. It must be as high as possible (for bigger impact) but not too much because tags might become too hard to detect. To this end, the authors introduced a function named top-hat. This function excludes all the readings that, from the current time (t_{now}), are separated by more than a certain time interval Δt_{hat}. The function 5.1 $f_{hat}(t)$ returns true if there is a detection or false otherwise. With this method, the tag is considered as detected, as soon as there is more than one detections during the time interval.

(5.1)
$$f_{hat}(t) = \begin{cases} True & |t_{now} - t| < \Delta t_{hat} \\ False & otherwise \end{cases}$$

In our smart home context, FPRs were also an issue and this function did not allow dealing with them. We generalized the $f_{hat}(t)$ function for that purpose. The new function (5.2), denoted by $f_{ite}(i)$, is constructed by using iteration instead of time as a parameter. It is preferable to use fixed time interval since it is easier and more intuitive. We can decide if a (boolean) tag's detection state (O_s) has changed by subtracting the first detection iteration (i_d) of a sequence of the opposite state to the current iteration (i_c) and comparing it with a Δi. The Δi is the minimum number of iterations the object's state needs to be stable before considering that the detected state has changed. The difference with [187] is subtle, but it allows one to deal with both kinds of wrong readings (FNR-FPR). Moreover, it enables to predict the effects of the filter as it will be seen in the next subsection.

(5.2)
$$f_{ite}(i, O_s) = \begin{cases} !\,O_s & |i_c - i_d| \geq \Delta i \\ O_s & otherwise. \end{cases}$$

5.3.1.1 Exploitation of the filter

To illustrate how this filter works, we will look through a simple execution example. Let us suppose that the tag X is undetected at iteration 1. Then, the parameter Δi is set to one for the simplicity of the example. The next iterations could go like this:

Iteration 2: X is *detected,*	$\lvert 2-2 \rvert \ngeq 1,$	no change
Iteration 3: X is *undetected,*	$\lvert 3-3 \rvert \ngeq 1,$	no change
Iteration 4: X is *detected,*	$\lvert 4-4 \rvert \ngeq 1,$	no change

Iteration 5: X is *detected*, $|5 - 4| \geq 1$, X is now detected
Iteration 6: X is *undetected*, $|6 - 6| \ngeq 1$, no change
Iteration 7: X is *undetected*, $|7 - 6| \geq 1$, X is now undetected

The performance of the function relies on the Δi that depends greatly on the RFID configuration. The parameter can be automatically determined if one knows the rate of false readings (fr_{rate}). For example, supposes that $fr_{rate} = 25\%$, that means roughly one of each four readings the tag should not be detected. If one aims to have a $fr_{rate} \approx 99.5\%$ the Δi would be 4 since:

$$ {}^{1}/_{4}{}^{4} = {}^{1}/_{256} \approx 0.39\% $$

The false readings rate will be probably higher since the behavior is not random (the tags often disappear for few iterations), but in case the calculation of Δi is not satisfactory it can be determined experimentally. It is important to notice that Δi value will increase the response time when the tags state really changes ($\Delta i * i_{length}$). Therefore, the value should be decided according to the need of the system.

5.3.1.2 Experiments with the filter

We tested the $f_{ite}(i, O_s)$ function in the LIARA's smart home with a proximity localization method and a trilateration for which only the antennas configurations and the reading speed had a different impact on the false readings rate. We computed the rate in the Figure 5.5 below:

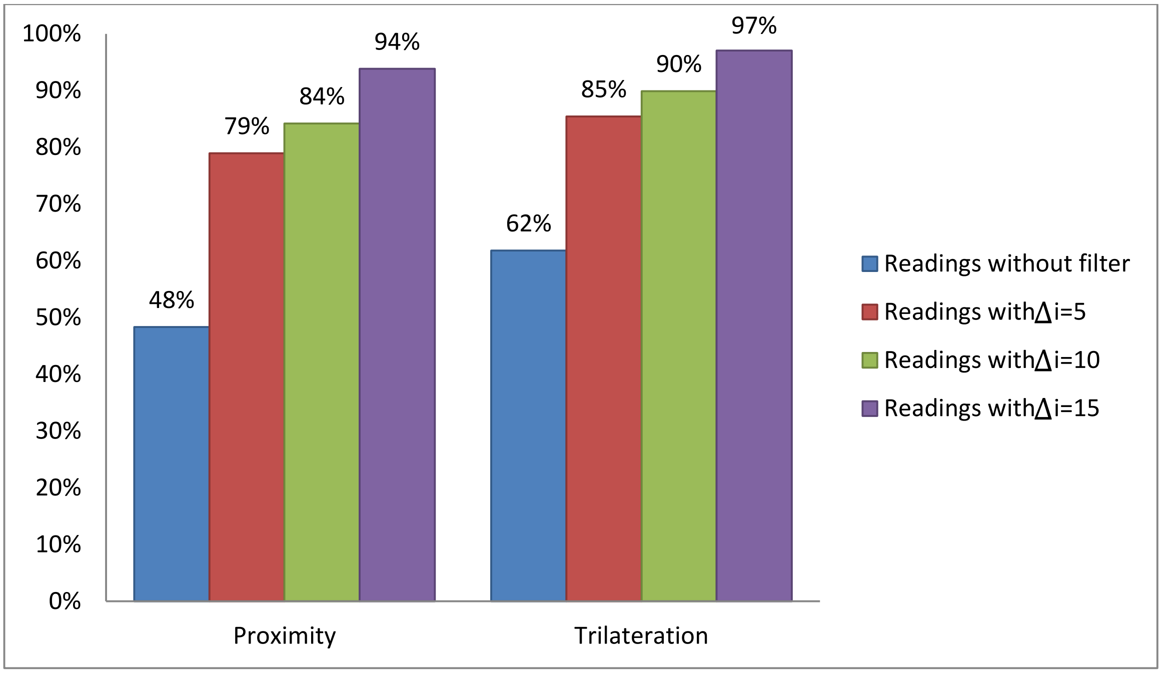


Figure 5.5: The false readings rate with and without $f_{ite}(i, O_s)$ at a reading rate of 5 per second.

One could also argue that the false readings problem could be addressed with a simple average instead of our newly developed filter. However, the results would not be the same and there would be information loss. The example below illustrates the difference between a simple average and the $f_{ite}(i, O_s)$ function with $\Delta i = 2$.

$O_s[\cdot] = \{T, T, T, T, F, T\}$	Average= True	$f_{ite}(i, O_s)$= True
$O_s[\cdot] = \{T, T, T, T, F, T, F, F, F\}$	Average= True	$f_{ite}(i, O_s)$= False

5.3.2 Addressing the variation of RSSI

Another problem with passive technology is that the RSSI usually have high variation from iteration to another even when the tag has not moved. At the step of localization, this high standard deviation results, on the tracked object, in a seemingly perpetual random movement. These variations cannot be completely eradicated because they are caused by an unchangeable fact. The RF signal is indeed greatly influenced by the environmental variables (persons, liquids, metal, etc.). The amount of flickering can easily be reduced with an average; however, it would give the same importance to old and late RSSIs. It would, consequently, delay the real movement of a tracked object. To overcome this issue, a Gaussian mean can be applied to RSSI returned by the antennas. To exploit it, we center the bell-shaped curve of the distribution on the current iteration number i_c as shown on equation 5.3. The parameter i is the iteration number associated with the RSSI record that we are weighting and the constant σ is determined proportionally to the iteration length. Thereafter, the mean weighted RSSI of a tag is computed by making use of the next function (5.4) $f_{strenght}(t[i_c])$:

(5.3)
$$f_{Gaussian}(i) = e^{-\frac{1}{2}\left(\frac{i_c - i}{\sigma}\right)^2}$$

(5.4)
$$f_{strenght}(t[i_c]) = \frac{\sum_{i=i_c-\Delta i}^{i_c} t[i]_{rssi} * f_{Gaussian}(i)}{\sum_{i=i_c-\Delta i}^{i_c} f_{Gaussian}(i)}$$

where $t[i]_{rssi} * f_{Gaussian}(i)$ denotes the weighted RSSI for the i^{th} iteration. This function receives as a parameter the RSSI of a tag to calculate the mean weighted RSSI. That parameter consists in an array ($t[\cdot]$) containing the RSSI readings for each iteration. Then, the sum of the weighted RSSI, for all iterations satisfying $i_c - i \leq \Delta i$, is divided by the total weight of the Δi reads. The constant Δi is the number of iterations considered for the RSSI mean calculation and is necessary only to limit the computation (remember that it is important in our context). Note that the history of RSSI values $t[\cdot]$ is emptied when the object change of state from detected to undetected in combination with the previous filter. If we did not emptied the array, the calculation could be done for very old values with new one. For instance, if an object got undetected during iteration 3 to 99, it would mean that at iteration 101 (for $\Delta i \geq 3$) the computation would be performed on $t[\cdot] = \{i_1, i_2, i_{100}, i_{101}\}$ which obviously does not make sense.

The Δi and σ constants can be determined automatically corresponding to the reading speed (s) and the time one wants to weight. Let us suppose that at least one second of reading needs to be given importance. Then, when s=200ms it results in five iterations, at s=20ms fifty iterations and so on. The rule that was used in our various experiments (with different speed configurations) for sigma was $\sigma = number\ of\ iterations/2$. The Table 5.2 below gives sample weight values for an iteration with few different sigma values with iteration ranked from the latest to the oldest:

Table 5.2: Sample weights of a reading with various sigma

Iteration Rank	σ=2	σ=5	σ=10	σ=25	σ=100
#1	100,00%	100,00%	100,00%	100,00%	100,00%
#2	60,65%	81,87%	90,48%	96,08%	99,00%
#3	36,79%	67,03%	81,87%	92,31%	98,02%
#4	22,31%	54,88%	74,08%	88,69%	97,04%
#5	13,53%	44,93%	67,03%	85,21%	96,08%
…	…	…	…	…	…
#11	0,67%	13,53%	36,79%	67,03%	90,48%
#21	0,00%	1,83%	13,53%	44,93%	81,87%
…	…	…	…	…	…
#51	0,00%	0,00%	0,67%	13,53%	60,65%
…	…	…	…	…	…
#201	0,00%	0,00%	0,01%	1,91%	13,67%

5.3.2.1 *Test of the Gaussian filter*

To validate that the Gaussian filter had an effect upon the accuracy of the position, we conducted a test with the four antennas that are installed in the kitchen of our smart home. To do so, we used a basic trilateration algorithm (see Section 5.4) using the Friis equation [188] without any other pretreatment filters or any post treatment. The experiments demonstrated that the Gaussian Mean Weighting filter greatly improves the accuracy. With this filter, the effect of the fluctuations on the signal strength is considerably reduced. After the application of this filter, the estimated positions are more accurate and grouped. The results can be seen on Figure 5.6.

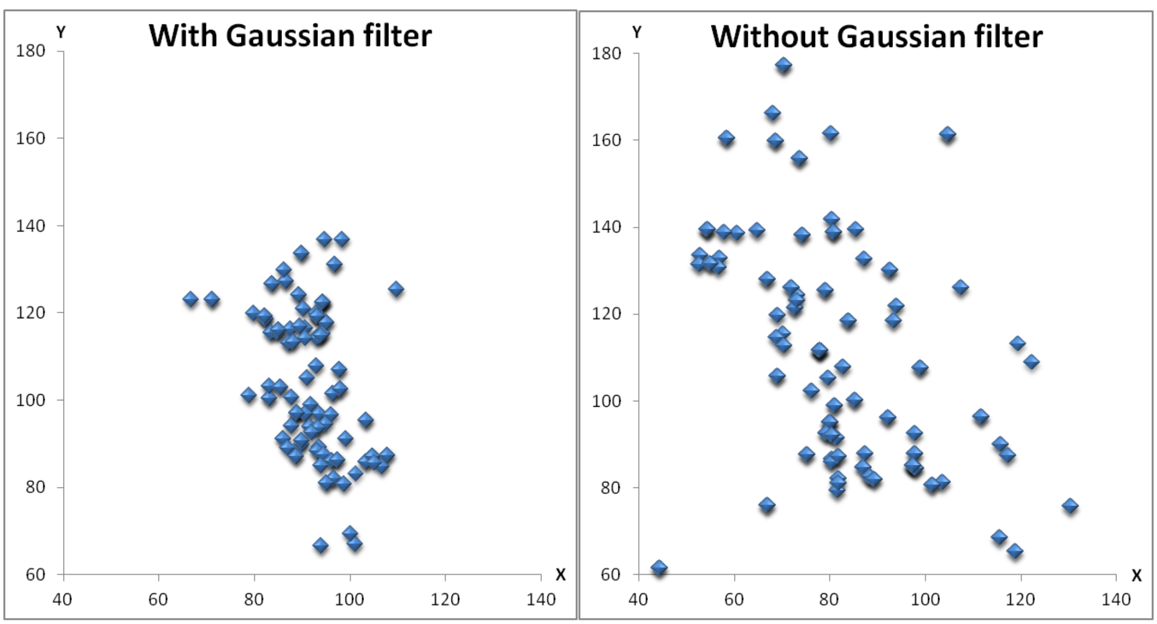


Figure 5.6: Concentration of the approximate positions of an object with and without the filter.

5.4 TRILATERATION WITH RSSI

The method of trilateration is a well-known process to localize an object using geometry of circles, triangles or sphere (in 3D). Often confused with triangulation, trilateration is performed by exploiting distance measurements in contrast to the latter that exploits the angle of arrival of the signal of two receivers separated by a known distance and the properties of triangles. The fundamental step to perform trilateration consists in finding the distance between the object being tracked and each antenna. The easiest way to do it with passive RFID technology is to use the received signal strength indication. The transformation of the signal to a distance with the RSSI can be accomplished with the Friis [188] transmission equation:

(5.5)
$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi D}\right)^2$$

where P_r, P_t, G_t, G_r, λ and D denote respectively the power received by the antenna, the power of the RF emitted by the antenna, the gain of the transmitter antenna, the gain of the receiver antenna, the wavelength of the emission and the distance from the antenna. However, in practice, this equation is often far from properly representing the wave propagation observed in smart environments due to noise and other interferences. To address this situation, our first alternative method was to design a custom equation for the RFID hardware exploited. To do so, we collected data series of a tag at different positions and learned the model [189].

The next and final step of basic trilateration is to solve circle equations to find an intersection point. In the general case, a minimum of three circles drawn from three signals is required to find the position. However, our smart home context, the antennas are put on the walls which make the half the surface of the circles unusable (where the second intersecting point would be). The Figure 5.7 shows an example of trilateration from two antennas placed at two known position P_1 and P_2.

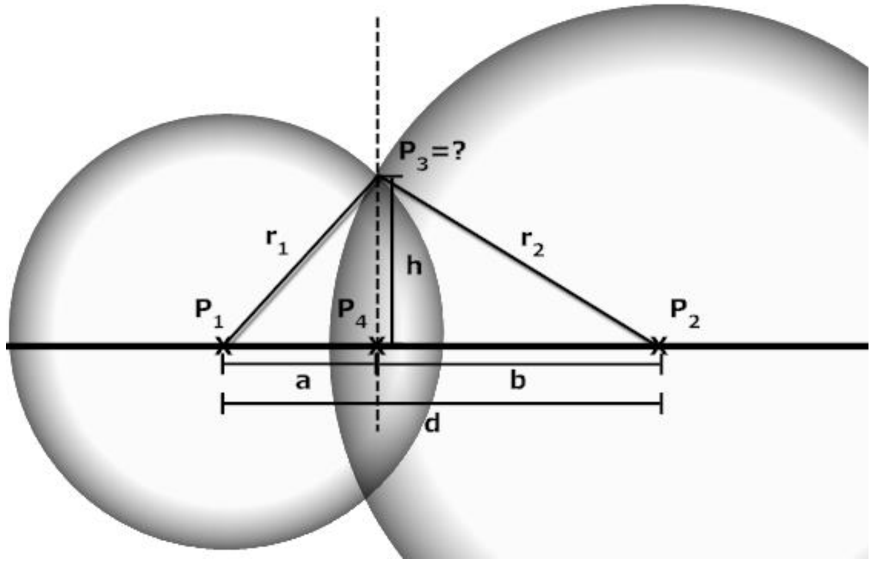


Figure 5.7: Example of trilateration with two antennas.

The points P_1 and P_2 can be written as Cartesian coordinates, that is $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$. The aim is to find P_3 that corresponds to the tracked object. First step consists to calculate the distance between both points: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_2)^2}$. Note that if d is bigger than the sum of both radius $(r_1 + r_2)$ there is no solution, so it might be worth ensuring the distance values are correct before performing the calculation. Then we must calculate a (or b) to find h: $a = ({r_1}^2 - {r_2}^2 + d^2)/2d$. Next, we must solve h to be able to find the center point P_4. Finally, we can compute the object position P3. The three equations below (5.6-5.8) resume the calculations that are required:

(5.6) $$h = \sqrt{(r_1^2 - a^2)}$$

(5.7) $$P_4 = P_1 + a(P_2 - P_1)/d$$

(5.8) $$P_3 = P_4 + h(P_2 - P_1)/d$$

To conclude, when using trilateration, the system should be designed differently than with a proximity based method. First, the antennas should be disposed in a way to ensure that always at least two of them detect the tracked objects at all time. The final configuration and arrangement of the antennas in the smart environment have a high impact upon the performance of the trilateration method and, thus, should be taken seriously into account. The antennas can generally be used at, or near, their full capacity (sensitivity, power). In that way, they can cover a much larger area (in our case, up to 3 meters in front of them). Additionally, trilateration is not very accurate with passive RFID. Therefore, we suggest speeding up the system as quick as possible to have more data which can be used to average the objects trajectories (for real-time tracking). We first tested trilateration at 200ms with reasonable results, but our system now supports reading cycle up to 20 ms which was used in this thesis.

5.4.1 Elliptic trilateration

Our first attempt to localize object with standard circular trilateration [189] was not satisfactory for this thesis. It has many problems and while the test gave good results, the stability of the localization with moving objects in realistic conditions did not enable a good online tracking. In fact, since our antennas are directional, the strength of the signal decreases in function of the angle of emission. This is to be expected since the radiation pattern of such antennas looks like a sausage. In such situation, elliptical propagation model approximates better the real behavior and are, thus, more appropriate. The is why we created a trilateration model that is based on the equation of an ellipse (5.9) [102]:

$$\frac{(x-h)^2}{A^2}+\frac{(y-m)^2}{B^2}=1 \tag{5.9}$$

In this equation, A and B are the values of major and minor axis of the ellipse and the variables h and m are the coordinates of the center of the ellipse. To compute A and B, we first have to establish the equations corresponding to the distance in function of the RSSI when the object moves away perpendicularly (major axis) and when it moves away from the side (minor axis) of the antenna.

To do this, we used the method of regression which enabled to find both equations. The regression was accomplished by saving the RSSI signal of a tag at different known positions in front of the antenna and on its side. We then performed a linear and a polynomial regression to obtain the equations. Since the polynomial regression had higher correlation coefficients than the linear case (respectively R_M^2=0.908 and R_m^2=0.909) and thus we kept the quadratic equations 5.10 and 5.11 shown below:

$$M_a(RSSI) = 0.1833 \times RSSI^2 + 8.5109 \times RSSI + 104.3 \qquad (R_M^2 = 0.974) \tag{5.10}$$
$$m_a(RSSI) = 0.0462 \times RSSI^2 + 0.8155 \times RSSI + 104.3 \qquad (R_m^2 = 0.937) \tag{5.11}$$

The first equation returns the value of the major axis (M_a) of the ellipse and the other returns the value of the minor axis (m_a) depending on RSSI. From these equations and from the respective positions of each antenna, we are now able to establish the different equations of the ellipse of the corresponding antenna simply from the RSSI received. If at least two antennas on the same wall or three on different walls detect the same tag according to the principle of trilateration the object should hypothetically be where those ellipses intersect.

The Figure 5.8 below shows the RSSI values and the corresponding distances values that were used to find the ellipse equations:

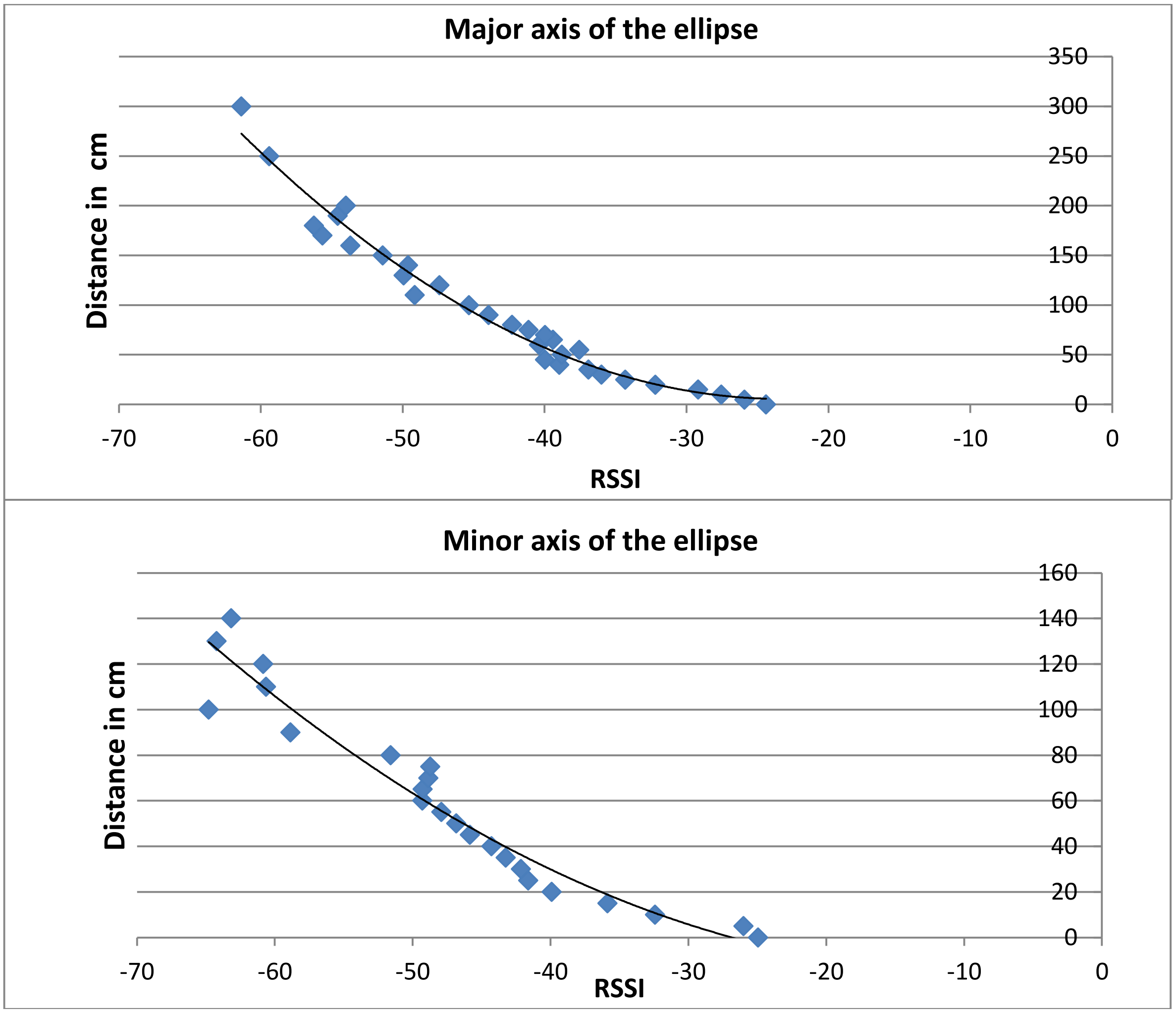


Figure 5.8: RSSI at different distances and the resulting polynomials.

5.3.2.1 Finding the intersections

In our particular context, the elliptical trilateration was implemented for the kitchen area with a set of four antennas disposed as seen on Figure 5.9. The tracked object is always seen by the four antennas, so we decided to perform trilateration for each possible pair of ellipses (a total of 6). To find their intersection points, we have to solve an equation of second or fourth degrees. This depends on which pair of antennas is involved.

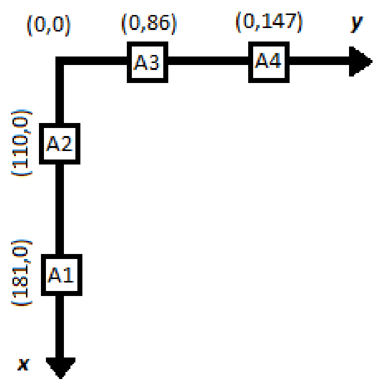


Figure 5.9: The antennas placement and their respective Cartesian coordinates

On one hand, when the two selected antennas are on the same wall (A1-A2, A3-A4), the equation of intersection is quadratic and is easy to solve. On the other hand, when we try to find the intersection points of two antennas located on opposite walls (A1-A3, A1-A4, A2-A3, A2-A4), we have to solve a quartic equation and to this end, we implemented the well-known method of Ferrari [190]. Therefore, we end up with five possible situations. For each pair of ellipses, we obtain between 0 to 4 points of intersection. Obviously, the calculation of intersection points is dependent on the configuration of the antennas in the smart environment where the trilateration is being performed. Therefore, this part of the method will change depending on how many antennas there are, and how they are placed.

5.4.2 NON-INTERSECTING PROBLEM

One problem that might happen while performing trilateration is the lack of intersection between a pair of antennas (or more). This situation could be addressed simply by progressively increasing/decreasing the size of each ellipse. The problem is that not all ellipses are equally accurate. Generally, those from the antenna receiving a stronger signal are much more reliable than the one with a weaker signal. Therefore, the rate of variation for each antenna should depend on the received signal strength. One thing that should be considered, when progressively increasing or decreasing the ellipses, is that a very small increment may be time consuming. We designed a small algorithm, called the Delta filter (Algorithm 5.1) which treats these situations and work with both the standard trilateration and the elliptical trilateration model developed in this thesis.

Algorithme 5.1: The Delta filter.

Input:	Two ellipses or two circles (Ellipse1 and Ellipse2)
Output:	One or more points of intersection

Get delta variation value for Ellipse1(V_1) and Ellipse2(V_2)

Initialize an empty table of point : P[]

Repeat till P[] is empty

  $\Delta_1 = \Delta_1 + V_1$

  $\Delta_2 = \Delta_2 + V_2$

  Compute intersection point for Ellipse1+Δ_1 and Ellipse2+Δ_2 and add them to P[]

  If P[] is empty **Then**

    Compute intersection point for Ellipse1+Δ_1 and Ellipse2+Δ_2

    Add them to P[]

  End

If P[] is empty **Then**

  Compute intersection point for Ellipse1+Δ_1 and Ellipse2-Δ_2

  Add them to P[]

End

End

Return P[]

This filter treats all possible situations. The case when the major and minor axes of the two ellipses have to be increased in order to obtain a point of intersection and also the one when an ellipse covers another ellipse (or circles, that are simply a special case of ellipses). In this case, we must reduce the shape of one ellipse and increase the shape of the other one. In brief, the delta filter is used to eliminate the situation where there is no point of intersection by modifying a pair of ellipses until they intersect and thereby create at least one point of intersection. The application of this filter results in 1 to 4 points of intersection for each pair of ellipses. The points that correspond to a complex number or those outside the eligible area are eliminated. If there is still more than one possible value, an arithmetic mean can be calculated to create a unique point. If the pair of antennas is not on the same wall, both are kept for to finally select the most promising position.

5.4.3 Selection of the final position

Because of all the uncertainty from the collection of the RSSI from each antenna to the conversion to an ellipse equation, it is quite improbable that three or more ellipses will converge in a unique intersection point. The simplest solution to this issue would again be the simple average. This is a method that should work fine. However, in our experiments, we observed that the intersection points obtained from antennas that have received stronger signal strength are more accurate. In order to improve the methodology, we developed a simple method to weight each of the potential positions. This average is performed through a filter that we called multi-point location, which returns a weight for each hypothetical point, accordingly to the following method.

The first step is to attribute a rank to each antenna so that the antenna which received the strongest signal obtains the first position and so on. Next, for each pair of antennas, we set the weight depending upon their position upon the ranking. The Table 5.3 shows how the weights are given according to the positions assigned. For example, if a point is obtained by the intersection of two ellipses that have the strongest RSSI (position 1 and 2) then this one will be attributed a weight of 1.00.

Table 5.3: The multi-point location filter matrix.

Position	**1**	**2**	**3**	**4**
1		1.00	0.80	0.40
2	1.00		0.40	0.20
3	0.80	0.40		0.00
4	0.40	0.20	0.00	

Finally, these points and their respective weights are used to compute the actual (final) position of the tracked object. This calculation is done by using the function 5.12 $f_{location}$ shown below:

(5.12)
$$f_{location}(p[\cdot]) = \frac{\sum_{i=0}^{n} p[i] * f_{weight}(i)}{\sum_{i=0}^{n} f_{weight}(i)}$$

In this function, $p[\cdot]$ represents each point (x, y) and $f_{weight}(i)$ is the weight assigned to it taken from the matrix above (Table 5.3).

5.5 IMPLEMENTATION AT THE LIARA

The localization model explained through the previous sections was implemented at the LIARA smart home exploiting the passive RFID system and the architecture described in Chapter 4. Of the eight antennas available, four antennas have been installed on the kitchen walls as presented in Figure 5.9. We selected them to test our localization system since they were in sufficient number and close enough for our purpose. The kitchen is also a logical choice because it is an area where there are multiple objects and where precision is important to achieve good recognition of ADLs. Indeed, it is a place where the resident performs a lot of complex tasks that might require assistance. We also had to select the passive tags to use for our experiments. The choice was made according to various criterions. We have chosen medium sized passive UHF tags because they were easy to integrate on objects while being strong enough for extensive usage. In addition, before installing them on our objects, we took care to select tags, which had relatively the same sensitivity [168]. With passive RFID tags, even if two tags are technically identical, sometime their sensitivity is very different. That might lead to unpredictable behaviors. Finally, we have incorporated tags to all objects of the smart home.

5.5.1 Software implementation

The new localization model was developed in Java and implemented using the Netbeans IDE. It was programmed not only for the model of this thesis, but for other models that we tested in the past or that we planned to test in the future. You can see the GUI of the software on Figure 5.10. As you can see, there are many options available that enabled us to test different pretreatment and post treatment filters and the possible combinations.

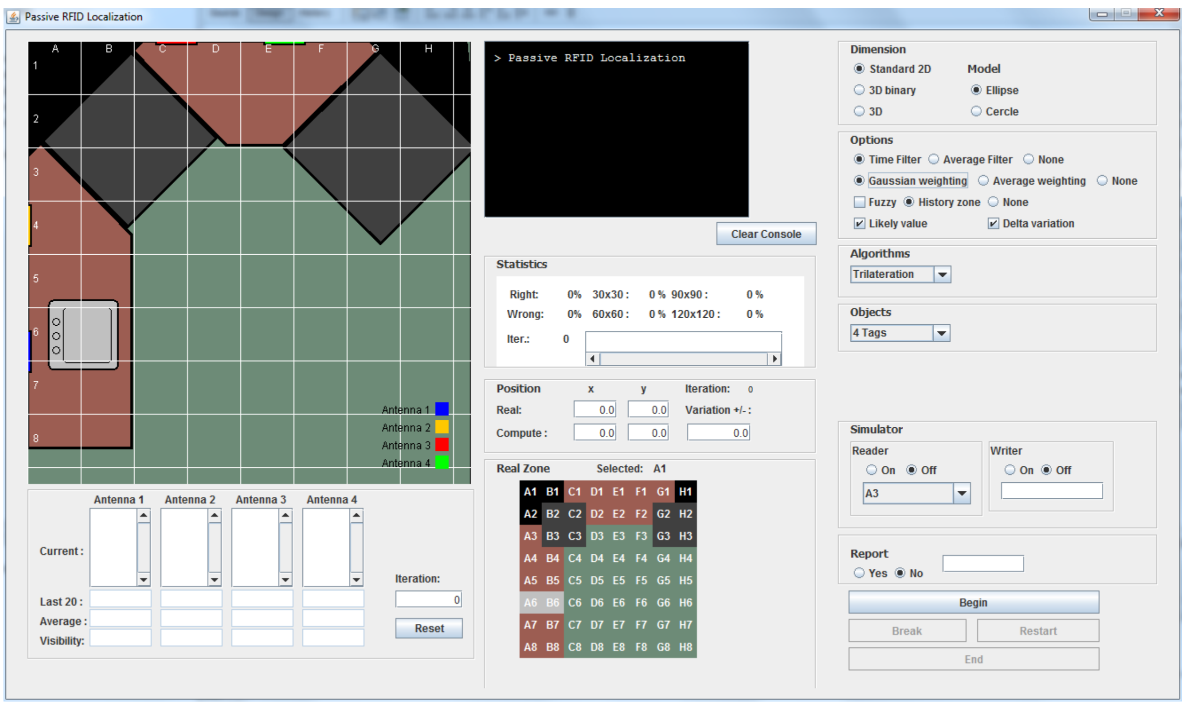


Figure 5.10: The Java implementation of the localization model.

The virtual environment was divided into logical 30cm X 30cm square zones for qualitative localization. This was indeed implemented in our previous model that exploited Mamdani’s fuzzy logic inference [189]. In that case, the fuzzy logic was exploited to express the likeliness of an object being in a qualitative zone. Three fuzzy linguistic variables (FLVs) were designed for that purpose: likeliness (what we need to infer), Euclidian distance from zone center and last appearance in the zone. For example, one of the rules of the system was: *IF near AND new THEN very_likely*. The Figure 5.11 shows two of the FLVs:

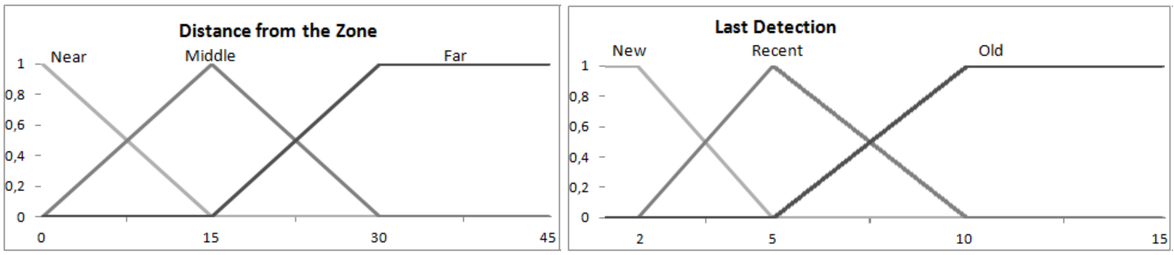


Figure 5.11: Two examples of FLVs for qualitative localization.

The software is shown in action below on the Figure 5.12. A live image from the localization of a tagged object in the smart home is shown along with the ellipses created for trilateration. On this image, the black square represents the final position that is computed in function of all the other points that were found during the previous steps. Remember that this final position is found with the methodology described through the previous section. The remaining points either are the direct crossing of the ellipses or the average points of each pair of these found intersections points. Some points also have been created per application of the delta filter. As you can see, the ellipses were fairly inaccurate during the iteration the screenshot was taken, but the position was still almost accurate due to the other enhancements. Finally, the colors of these points correspond to the weight they were given using the matrix on the Table 5.3 shown in section 5.4.3.

The red points:	[1.00-0.75[	The yellow points:	[0.50-0.25[
The orange points:	[0.75-0.50[	The white points:	[0.25-0]

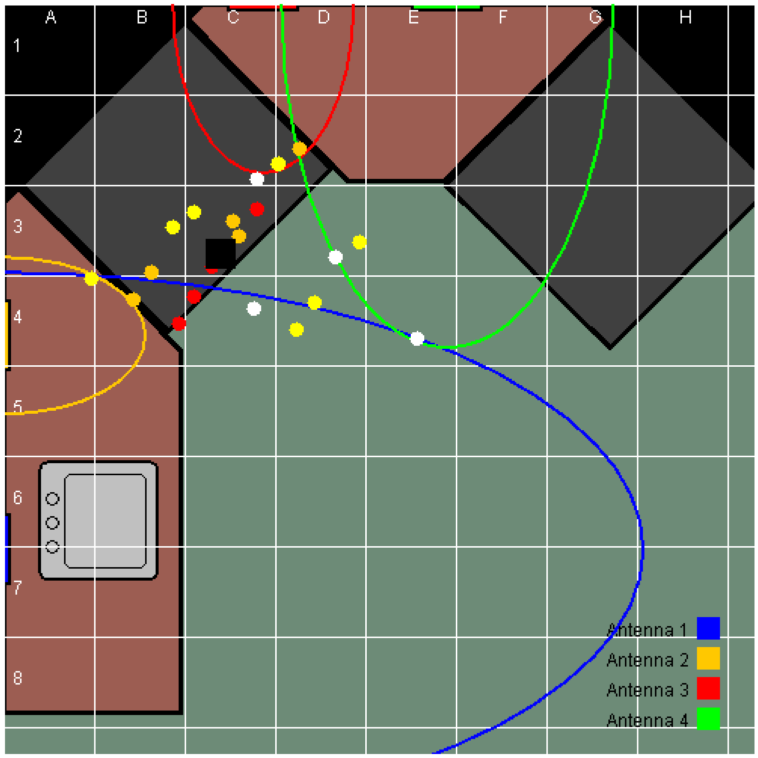


Figure 5.12: Screenshot of the localization during an iteration.

5.6 VALIDATION

In order to challenge our new tracking system and test its accuracy, we have established an experimental setup which fairly represents the reality of a smart home. We have included the entire kitchen to our experiment. This includes the two counters of 170 X 60 cm and 129 X 60 cm respectively, the oven, the sink and also whole space where there is nothing (only the floor). For the first set of experiments, we decided to use a cylindrical object made of four tags oriented in different directions.

The protocol first began by placing the object in the center of a zone, and then we recording the RSSI returned by each tag for each antennas for at least 400 iterations (80 s).

The same process was repeated for each of the available zones of the kitchen. Data's recording allows us to compare more precisely the different algorithms by eliminating the variations in the antennas' reading. However, it should be noted that the software is made to manage real-time tracking. It is only to improve the value of our results that the recordings were performed. Once all the possible scenarios are saved on the hard drive, it becomes easy to compare the efficiency of the proposed filters by using different configurations of the algorithm. For example, the Gaussian Average Weighting filter could be removed in order to verify its effectiveness.

The localization algorithm was tested on two different aspects during this period of experiments. As first measures, we computed the gaps between the real position, and the one returned by our algorithm. This gives us the approximate margin of error and with this data, we are able to compare our new model to other works although the experimental conditions and the installations are different in each case. Additionally, during this first step, we tested the contribution of the elliptical trilateration and each of our filters on our algorithm. As second measure, we tested whether our model could determine if the object has in the right zone or not. This value is relevant because identifying these zones constitutes one of the key information that we need for the recognition of ongoing activities of daily living. To verify the effectiveness of the fuzzy logic filter, we collected the success rate for different zones.

5.6.1 RESULTS AND ANALYSIS

The first series of tests exploited the recorded data as per the experimental protocol described. First localization results were obtained using trilateration with circles and no filters to give us a basic benchmark. Then, we progressively added filters and changed the parameters until the final configuration of the localization method described in this thesis. The obtained results are presented in Figure 5.13. Note that for each configuration, they are presented in terms of proximity from the center of the zone in centimeters. At the light of these results, it is clear that each proposed filter improved the performance of the algorithm. First thing to notice is the great enhancement enabled by the elliptical trilateration. It can be explained by the fact that our antennas transmit their waves such that the signal loss is greater when the object moves away laterally from it. Therefore, in the lateral areas, the location success rate is really low with circles. As evidence, on more than 1,600 iterations and with an elliptical model, we obtained an average precision of 14.12 cm and with the same settings but with a circular trilateration, the precision was reduced to 32.52 cm.

Secondly, the two pretreatment filters (the Gaussian average and the iteration filter) also improve the results. Thirdly, as shown on the Figure 5.13, the inclusion of Delta and multi-point localization filters brought one more contribution to the model. The Delta filter helps to find points that are, in some situations, crucial for a proper localization. Often, ellipses are very close one from the other, but they do not intersect. Therefore, if we do not use this filter, these points would be ignored, and the resulting position is less accurate. Furthermore, since the signal strengths and shapes of the ellipses are constantly changing, there may be an intersection point that disappears and reappears from one iteration to another. This would have the effect of changing the computed position of the object even when it does not move at all. On the other hand, the multi-point location filter assigns different weights to the points of intersection; it modulates their value according to their accuracy. After a full analysis of these results, we can conclude that each of these components is effective, but it is with their combination that we can provide a good stability coupled with a high-accuracy rate.

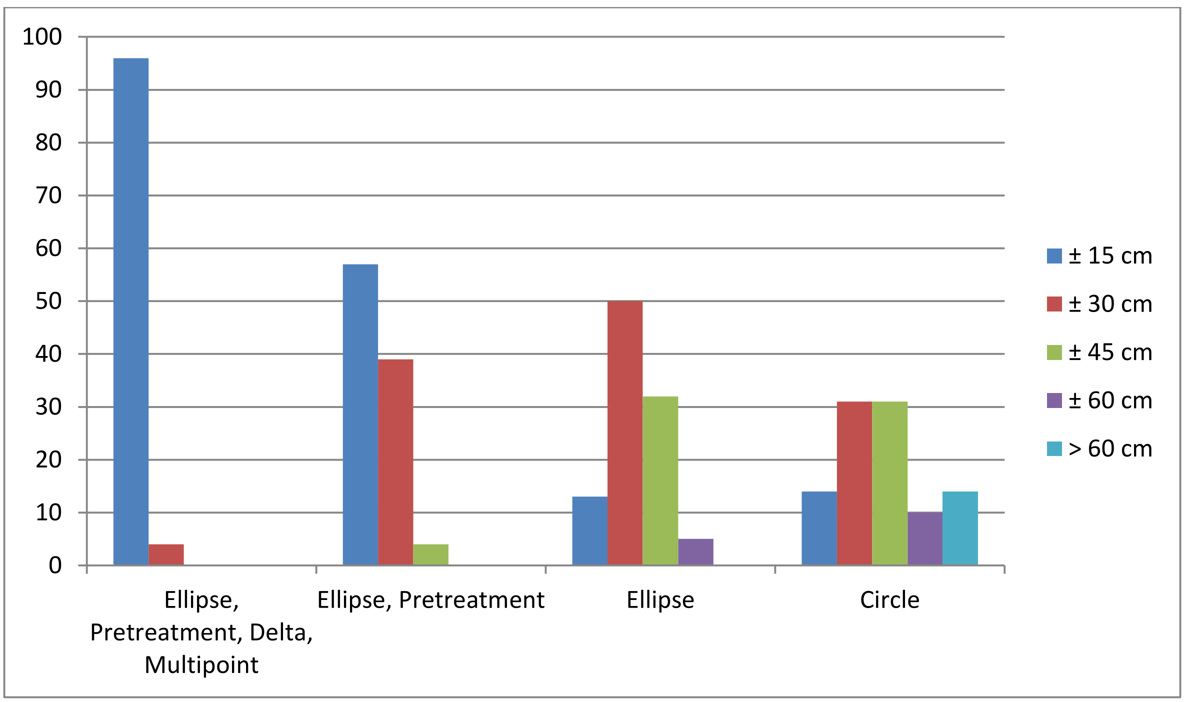


Figure 5.13: Accuracy under various configurations of the trilateration.

5.6.2 Tests with various type of objects

In addition to the tests conducted with a custom made object, we needed to validate how our localization algorithm would perform with daily life objects. To perform this new series of tests, we followed the same protocol as explained before, but with various objects such as cups, plates, utensils and others. The protocol was divided into few steps. First, we selected six representative zones of the kitchen. Then, we took each object and positioned it in the center of each of these zones. The average error in cm between the estimated position and the center of the zone was compiled for each object. We also compiled the percentage of estimations that were in the 30 X 30 cm zone, in the 60 X 60 cm zone and out of it. The Table 5.4 presents the results that were obtained in addition to some information on the objects exploited.

Table 5.4: Results of the experiments with daily life objects

Object	Tags	Diameter (cm)	Proximity from the real position in centimeter			Accuracy ±(cm)
			30x30	*60x60*	*Out of 60x60*	
Cup	4	8,9	68,6%	15,4%	16,0%	17,43
Milk	4	13,5	84,8%	15,2%	0,0%	12,43
Kettle	4	17,4	76,0%	24,0%	0,0%	13,34
Bowl	4	19,6	69,2%	30,6%	20,0%	14,01
Sugar	4	10,2	64,4%	35,6%	0,0%	15,03
Pepper / Salt	2	5,0	60,6%	19,0%	20,4%	16,47
Spoon / Fork	1	-	36,2%	27,8%	36,0%	25,08
Small plate	4	27,2	47,2%	50,8%	2,0%	17,26
Large plate	4	31,0	49,6%	50,4%	0,0%	16,32
Coffee	4	8,8	77,4%	22,6%	0,0%	13,39
Average	-	-	63,4%	29,1%	9,4%	16,08

As you can see, the results were generally good. In fact, we obtained an average error of 16.08 cm, which is slightly less accurate than with our special cylindrical object. Moreover, if we remove only a few objects of our stats, the results would be equal or better than those obtained with our special object. In overall, we were positively surprised by the results obtained with the real objects. However, for some of them (fork, spoon and plate), the results were a bit off the track. There is, however, a sound explanation to this. First, due to the shape and size of the fork/spoon, we were able to put only one tag in an inadequate orientation. Second, these utensils are made of metal and as previously mentioned, this has the effect of modifying the radio waves. Thus, the RSSI values returned by the antennas were much less accurate. Furthermore, by analyzing more deeply the results, we found that the size of the object also influenced the localization. In fact, with our plates, which have respective diameters of 27.2 and 31.0 cm, we have had some difficulty to position them correctly. In addition, the tags on plates are generally hard to fix; we encourage smart home researchers to buy special plate to allow tags to be fixed on the side. Following this series of experiments, we can conclude that our model works well even in noisy situation with various obstacles (metal, shapes, false read, etc.).

5.7 CHAPTER CONCLUSION

In this chapter, we described a localization algorithm that exploits the received signal strength indication from passive tags and four antennas that exploit multiple filters combined to trilateration. The algorithm introduces the elliptical wave propagation model in order to adapt to the antenna signal loss that is greater on the side. The goal of this algorithm is to enable the collection of the position data of the daily life object as the first step of our spatial data mining model. In particular, with this algorithm, we are able to track several objects in real-time and obtain their positions with an average error of less than fifteen centimeters. For each object in the smart home, we obtain eight RSSI data per tag that is transformed into one position. Therefore, that first step already greatly reduces the size of our data warehouse. However, there are still a lot of data collected since we have one position per object every

20ms. In our smart home, which represents a small apartment, we collect up to five millions positions per day in RFID. In the next chapter, we will see how we can transform these simple positions into high level qualitative information in order to perform spatial data mining.

In conclusion, our localization algorithm performs well and is good enough for the scope of this thesis. Nevertheless, it is far from perfect and let open many questions for the future research on passive RFID localization. One of the most important is, how to localize objects on several plans (3D localization)? Another important improvement that needs to be done is on the technology itself and the firmware of the equipment. A question also subsists over the large-scale implementation of a trilateration system with several antennas: How to avoid interferences and collisions? Passive RFID localization offers challenges for many more years of research ahead. Despite that fact, this work is a major addition to the literature [100, 102, 189] that demonstrate the power and the potential of this technology for smart homes.

CHAPTER 6

GESTURE RECOGNITION

We closed the previous chapter with a method to collect the positions of daily life objects in the smart home during the realization of the activities of daily living. From the localization algorithm presented and the smart home infrastructure of the LIARA (hardware and software), it is now possible to create our data warehouse to perform data mining. Nevertheless, the basic positions are difficult to use as they are and constitute a very large amount of data to deal with. In this chapter, we present an algorithm that was developed for this thesis, which is able to detect and recognize *atomic* and *composite* gestures. Our goal was to exploit this gesture recognition algorithm as a data preparation step. With it, we transformed the basic and redundant positions into series of qualitative movement on active objects. It therefore not only enables the exploitation of our spatial data mining but it also aggregate and significantly reduces the amount of data to process without losing expressivity.

This new model is one of the first to exploit passive RFID technology to successfully recognize gesture and it was developed so it could be used as a standalone algorithm for any other purpose. In addition to this novelty, the model introduces a very effective way of dealing with the difficult step of data segmentation. Indeed, in the literature, most models suppose that obtaining basic directions and segmenting the gesture is not a problem. Finally, our gesture recognition model performs well on computational complexity and supports variable amount of noise.

The remainder of this chapter is divided as follows. The section 6.1 introduces the basic concepts of gesture recognition. The section 6.2 gives an overview of the literature on the subject and describes the only model using passive RFID for that purpose. The section 6.3 discusses more particularly the challenges of gesture recognition with noisy data extracted from RFID localization. The section 6.4 describes the gesture recognition algorithm that was developed and exploited in during this thesis [105]. It might be noteworthy to mention that we also developed two other gesture recognition models during this thesis project that are out of the scope of our spatial data mining model [104, 106]. The section 6.5 describes the software implementation and the validation of this model. Finally, the section 6.6 concludes the chapter with an assessment of the limitations of this model and potential future work.

6.1 GESTURE RECOGNITION

A gesture is widely described and recognized as an expressive and meaningful body motion (hand, face, arms, etc.) that convey a message or more generally, embed important information of spatio-temporal nature. Gestures are ambiguous and incompletely specified, since a multitude of conceptual information can be mapped to one gesture. The usual steps to perform gesture recognition from spatio-temporal data series are the following [191]:

1. Dataset segmentation
2. Filtering of the data
3. Limitation of the directions
4. Matching with a knowledge base

In many cases, however, the segmentation step is ignored because it is assumed that the user specifies the start and the end of a gesture with a device or simply because it is assumed that it is known. For example, supposes that the gesture recognition is exploited to communicate with an application, the user could have to press a button to begin the gesture, and it could automatically end when one is recognized. That is, there is no consecutive gestures in that context and therefore no need for segmentation. However, when it is required to support gestures of varying length interleaved with small to big inactive time, the segmentation becomes the most challenging issue of the gesture recognition process. On the other hand, the filtering is a straightforward step that usually requires ad hoc methods. It consists in standardizing the data (time, format, etc.) and compensating for missing data. The step of the limitation of directions is sometime ignored too. Still, most models transform the positions into a finite set of basic directions. It is performed in order to simplify the matching step and limit the possibilities of gesture. Finally, the matching step is the one for which the sequence of directions is matched to one of the gestures in the knowledge base. The most part of the literature focuses particularly on that aspect.

6.2 LITERATURE ON GESTURE RECOGNITION

Gesture recognition is an old problem that has particularly attracted researchers on Human-Computer Interfaces (HCI) [191]. Many algorithms are used for natural and efficient design in video games, software engineering and even in smart home [192]. Among the technologies that are usually exploited, video cameras and accelerometers represent the bigger chunk of the literature [193]. But whatever the technology exploited, one can observe that most approaches are based on statistical modeling such as the Hidden Markov Machine (HMMs) [194], Kalman filtering or other particles filtering [195]. In the remainder of this section, we will review the main classical models that address gesture recognition and review their limitation for our specific needs. We will conclude with the only passive RFID based gesture recognition model that we found. This model is used in this chapter as a comparison basis to validate our approach.

6.2.1 CLASSICAL GESTURE RECOGNITION

Samaria & Young [196] have developed a gesture recognition model which exploits HMMs to extract efficiently facial expressions from a single camera. As we explained in Chapter 2, the HMM is a double stochastic process with a finite number of states, and a set of random functions associated to each state. The transition between states has a pair of probabilities (the transition and the output probabilities). The reasoning corresponds to the process of finding the HMM with the highest probability of explaining that set of observations in the same manner that it is exploited for activity recognition. It is generally required to design and train one HMM per gesture that should be recognizable [192].

Gesture recognition from particle filters based tracking is also very popular [191, 197]. For instance, Shan & al. [195] combined the technique with Mean shift to perform real time hand tracking. Their algorithm, named Mean Shift Embedded Particle Filter (MSEPF), was tested on a 12fps camera stream with a 240x180 pixels resolution. They showed that their method could robustly track a hand to recognize gestures. Particle filters are very effective in estimating the state of dynamic systems from sensors information. The key idea of these filters is to approximate the probability density distribution by a weighted sample set.

Finally, a large number of gesture recognition approaches effectively exploit Finite State Machines (FSMs). For instance, Hong & al. [198] exploited spatial clustering to learn a set of FSMs corresponding to gestures. The idea was to learn the data distributions without the temporal information at first. The clustering extracted the different states to be used for a FSM. A second phase aligned the order of those states by exploiting the temporal information. They tested their approach using four sample gestures performed in front of a video camera. They achieved a hundred percent recognition rate, but admit that with a very noisy data sample, the recognition would fail.

6.2.1.1 Limitations of those models

The main problem with the models of the literature for our precise context with passive RFID in smart home is the many hard assumptions that are made. First of all, it is often assumed that obtaining the basic directions of the movement is straightforward. It is not the case with passive RFID tracking. Secondly, it is assumed that the amount of noise is not a problem (or that there is simply no noise). Thirdly, segmentation is often not an issue within the HCI context; therefore, few models discuss let alone address this issue. Finally, they generally suppose that the user is cooperative; an intended recognition context. In that context, the user tries to make his gesture easy to recognize for the algorithm. In our case, the recognition is done unbeknownst to the user. The impact is that the gestures are potentially poorly executed and very different from one time to another.

6.2.2 Gesture recognition with RFID

Due to the inherent difficulty of localizing objects with passive RFID technology, we found only one team of researchers that tried to tackle the challenge of gesture recognition with this technology. The team of Asadzadeh & al. [199] investigated the problem with a partitioning localization technique combined to reference tags. With three antennas on a desk, they monitored an 80cm X 80cm area, which was divided into 64 equally sized square cells (10cm X 10 cm). To recognize a gesture drawn by a user, they make few assumptions on the sequence of traversed cells. First, the system is fast enough to never miss any cell of the sequence; that is, the tracked object cannot move farther than one cell away in between two readings. Second, they assume that only forward local moves are possible. For instance, the Figure 6.1 shows legal (a) and illegal moves (b-c).

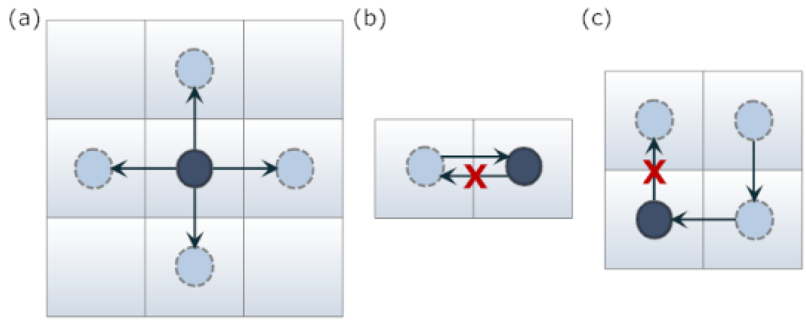


Figure 6.1: (a) legal move, (b) (c) illegal moves.

From the sequence of crossed cells, their algorithm generates a list of hypotheses by developing the possibilities into a tree structure. Next, a gesture matcher, GESREC, looks up into a dictionary and finds the gesture that best matches the sequence. Their algorithm cannot recognize two consecutive gestures (no segmentation) but works well (93% recognition) on a dictionary of twelve simple gestures. Their work showed that there is potential for gesture recognition with passive RFID. However, their assumptions made it difficult to apply their system in a smart home which requires more flexibility. For example, we cannot implement their localization system and certainly cannot reach a perfect accuracy for a localization precision under ten centimeters.

6.3 Challenges of gesture recognition

In this section, we aim at describing some important challenges of gesture recognition that we encountered during this project. But first, let us specify a little more what we mean by a gesture. A gesture is the result of daily life object manipulations or movements, which can be seen as a set of Cartesian positions comprised in a time interval, and that correspond to a recognizable pattern. That is, a gesture is a spatio-temporal series and the process of recognition corresponds to the matching of a definite number of imprecise observed positions at specific times to a known gesture. In our case, a gesture is a composition of one or many basic directions, where a direction corresponds to a general trend in the evolution of the observed Cartesian positions through the time interval within a certain precision. In our case, RFID localization suffers from a lot of imprecision, which means not only that sometime the observation can be off the real position, but also that during a certain interval, the tracked object might appear as going in a completely different direction. This constitutes the first important challenge that must be dealt with: finding the degree of acceptable variation in the observed positions before considering it is a movement. The Figure 6.2 shows an example of unlucky observations that would lead to a false interpretation.

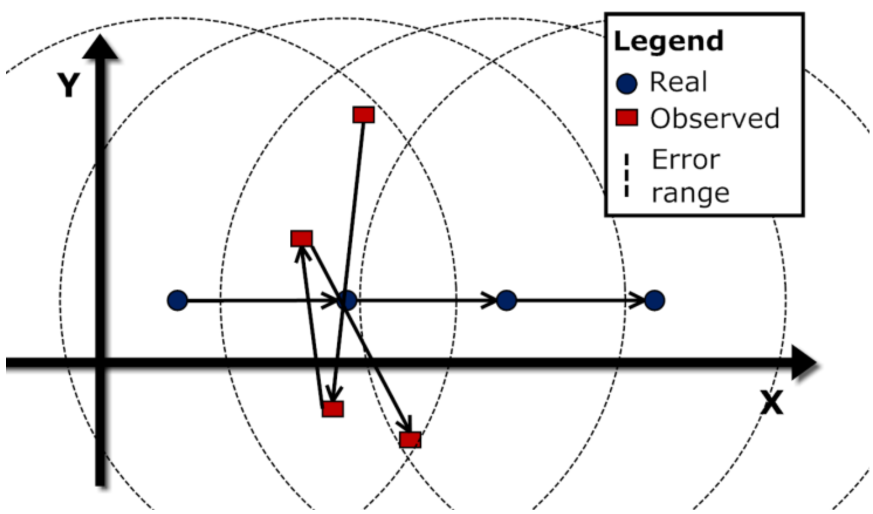


Figure 6.2: Example of unlucky observations leading to a false conclusion.

6.3.1 SEGMENTATION

The second challenge resides in the consecutive observation of many gestures, whether they are simple or complex. Indeed, during the recognition, it is hard to know exactly where a basic direction end or when a gesture end. It is even harder when there is idle time during the movement. That idle time may mark the end of a gesture and thus should be recognized too. However, it could also be insignificant and thus ignored by the algorithm. The segmentation is a tricky challenge, especially when trying to create a generalized solution that should work with different technologies or localization precision. It is also fairly dependent on the chosen granularity for the basic directions. Changing that granularity can lead to dramatically different results. The Figure 6.3 illustrates a dataset which is interpreted with three granularity values. As you can see, one results in one basic direction while the most complex results in a gesture composed of six.

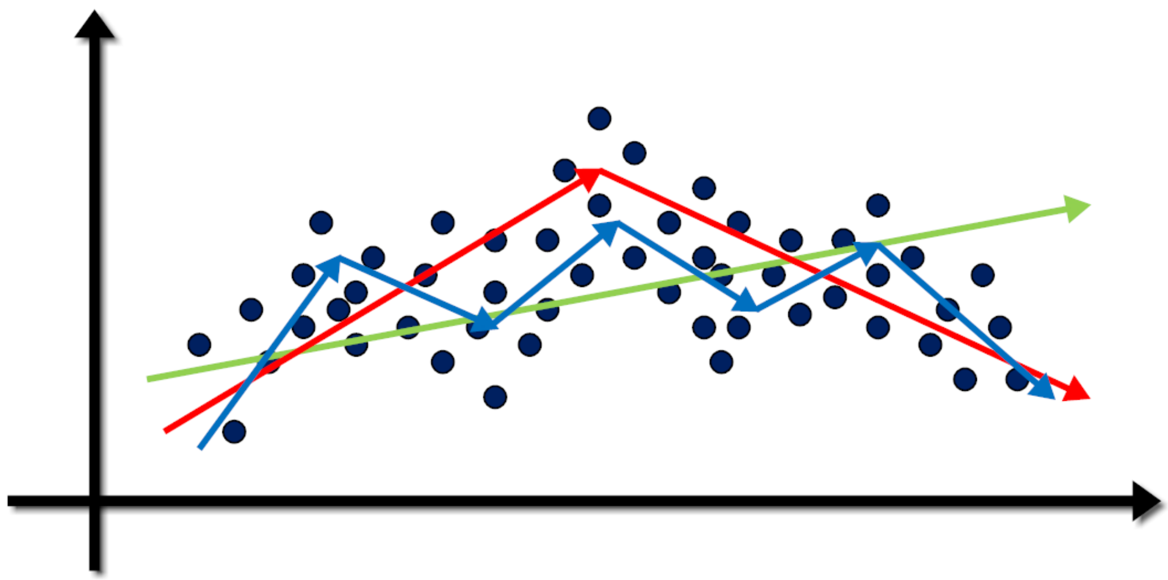


Figure 6.3: Same dataset, three granularity values, and three completely different gestures.

6.3.2 Limiting the directions

Another important aspect of gesture recognition is the reduction of the basic directions. It is very important to not have an infinite number of basic directions. Moreover, it is important to conceive it in a way that scales well. For our gesture recognition model, we decided to look on the side of qualitative spatial reasoning (QSR) to address this challenge. We selected the framework of Clementini & al. [85] in which they constructed a formal model to express distance and orientation relationships. The model enables various granularity levels which suits well to our situation. The Figure 6.4 shows three possible configurations. The first one, for example, allows to explicit two relations of distance; close (cl) and far (fr), and two relations of direction; left (l) and right (r).

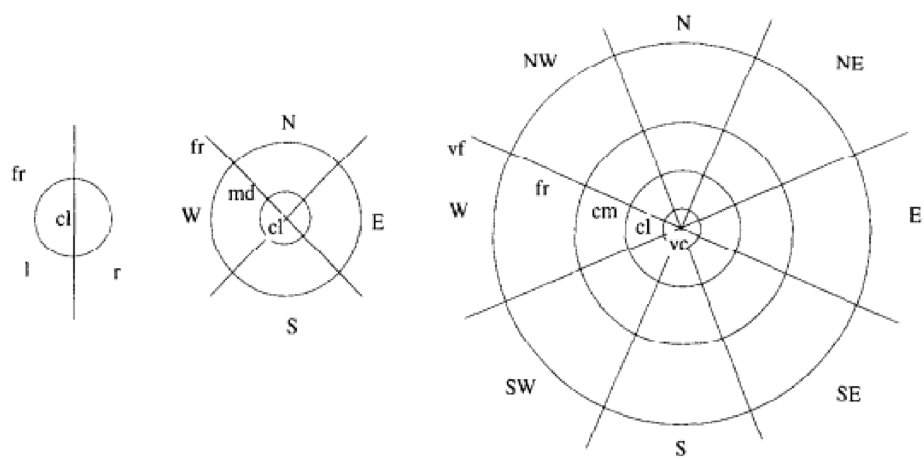


Figure 6.4: Three configuration examples for the QSR framework [85].

There are many advantages of using qualitative directions representation over quantitative. For instance, it reduces the possibility and the complexity of the recognition. However, it is mainly for future applications of the recognized gestures that we opted for a well-established reasoning framework. A QSR model enables to complete relationships information, and that could be used for many purposes in the future. For example, it could help in making prediction of the next steps/actions of a resident performing some specific tasks. As we explained in the beginning, our model was designed to work independently of the other part of the spatial data mining method presented in this thesis, and in that goal, it was designed to offer good flexibility for other potential uses.

6.4 THE NEW GESTURE RECOGNITION METHOD

In this section, we present the gesture recognition method that was developed for this thesis. The main algorithm, which identifies the basic directions, is recursive, and it works by developing the dataset of positions coordinates extracted from the localization algorithm into a tree structure and finding a way to combine the solutions. What is important to mention is that the method depends directly on the accuracy of the localization algorithm and takes it as a parameter ($\varepsilon = 14cm$). That is, a valid direction cannot be less than the average error and is probably significantly longer. This is the limitation of the granularity of our gestures. However, by taking the average localization error into account, we created an algorithm that can scale well. The algorithm can adapt to any localization algorithm provided that ε is specified. The new method also depends heavily on the reading rate of the tracking algorithm. Obviously, if the gesture is not composed of many positions, it will not be possible to recognize it. However, we have a reading rate of 1/20ms and a human gesture would never be performed faster than in approximately a second in daily life activities. The Figure 6.5 depicts the overall method that is presented through this section. We will first explain how to perform one execution of the algorithm to obtain one basic direction and then we will describe the recursion and the combination of the results. A complexity analysis is also provided at the end of the section.

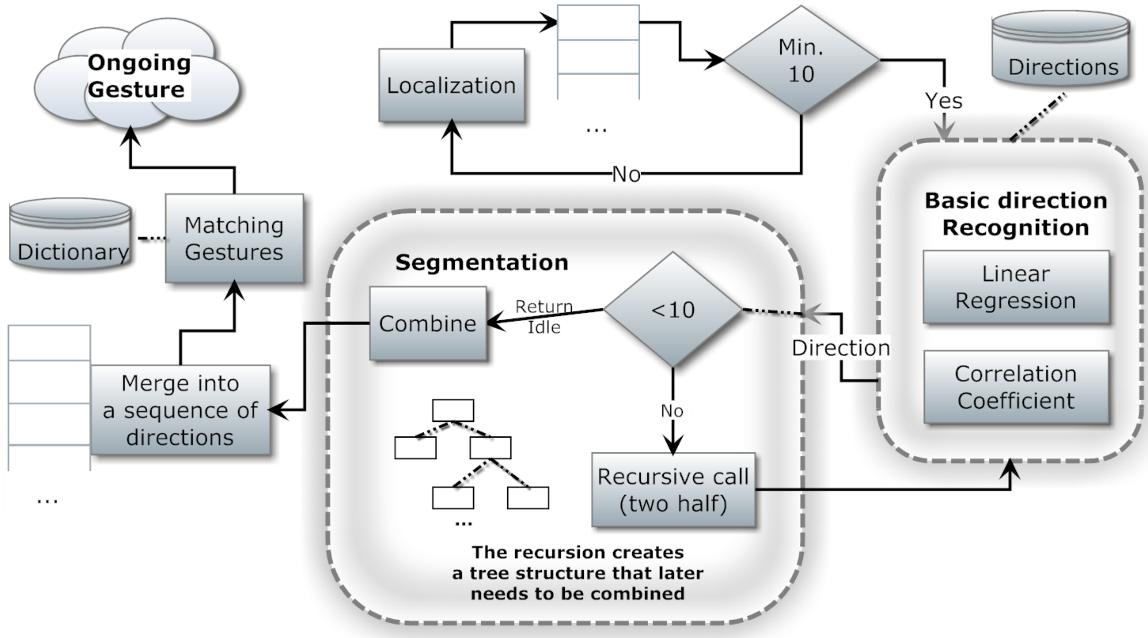


Figure 6.5: The overall gesture recognition method.

6.4.1 Basis of the algorithm

In this section, we describe a few things that are mandatory to understand the whole algorithm. First of all, the algorithm takes as input the average error ε and a list of positions $S = [p_n(x_n, y_n), p_{n+1}(x_{n+1}, y_{n+1}), \ldots, p_m(x_m, y_m)]$ where n is the iteration number of the beginning of the list and m the ending iteration number. Secondly, when the whole gesture recognition algorithm is working online, it waits for ten new positions to run. It does so because there are no significant changes under 200ms in the environment and to optimize the resource usage.

Another important point to mention is how to convert a found direction into the qualitative model of Clementini & al. [85]. A quantitative direction is a vector. In our model, the frame of reference is the origin of the vector representing the extracted direction. The number of qualitative orientation is set to eight, but again, the framework is made to be able to scale easily. These eight basic qualitative directions are $O = \{E, NE, N, NW, W, SW, S, SE\}$ that stand respectively for: East, NorthEast, North, Northwest, West, SouthWest, South, SouthEast. The distance values are specified with the average error of the localization algorithm (ε). Therefore, a gesture is composed of a list of pair (O_x, ε_y) such as $G = [(O_x, \varepsilon_y)_1, \ldots, (O_x, \varepsilon_y)_n]$. The Figure 6.6 shows an example of quantitative direction transformed through the framework where the representation would be (E, ε_4).

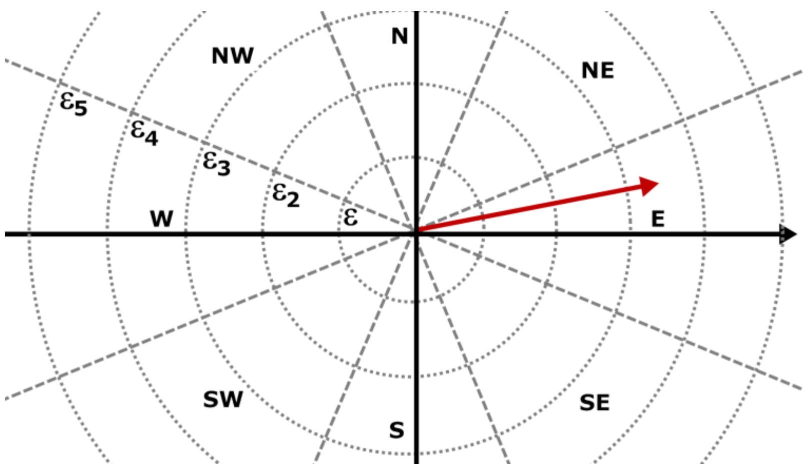


Figure 6.6: A sample vector (in red) in the QSR framework.

Additionally to the basic qualitative direction, we consider a special direction named *Idle*. In a smart environment, the tracked objects are generally not being used so we have to be able to identify this situation. It is considered that the object is *idle* whenever the quantitative direction extracted from the set of position is under ε_1 the average error. The main advantage of this method is that the algorithm can scale automatically to new smart environments and/or new localization algorithm without needing any learning or configuration. The only thing needed is the average positioning error (ε).

6.4.1.1 *Smallest enclosing circle*

We already explained that our algorithm depends on the average positioning error (ε). However, to be used as a test condition, we also require a method to find the highest distance between a pair of points in a dataset. To do so, we chose to find the smallest enclosing circle and compare its diameter to ε. There are several methods to compute the circle, but the simplest execute in $O(n^4)$ which is not desirable in our context. Therefore, we exploit the geometric approach which requires $O(n^2)$.

The method start by drawing an enclosing circle of center c (step 1). Then, the size of the circle can be reduced by finding the point a farthest from the center of circle, and drawing a new circle with the same center and passing through the point a (step 2). If the circle does not pass through two or more points, make the circle smaller by moving the center towards point a, until the circle makes contact with another point b from the set (step 3). If the circle contains an interval (point-free interval) of arc greater than half the circle's circumference on which no points lie, the circle can be made smaller (step 4). Supposes that d and e are the points at the end of that interval, reduce the circle until one of these conditions is reached:

I. The diameter is the distance $\overline{de}$

II. The circle touches another point f from the set

   a. If no such point-free arc interval exists then end
   b. Else go to (step 4) repeat the process

As you can see, the step 1 is constant and finding the point a in step 2 is done by passing all the points once: $O(n)$. The step 3 can also be done in $O(n)$. To find the point f, it is necessary to test the $n-2$ remaining points. However, every time we must verify the remaining $n-3$ are still in the enclosing circle. That last step is accomplished in $O(n^2)$.

The smallest enclosing circle diameter is used for two things in our algorithm. First, it serves to determine if a set of positions is considered as *Idle*. Second, it is used as a stop condition for the recursion.

6.4.2 PERFORMING A LINEAR REGRESSION

From a set of positions, the quantitative direction is found by performing a linear regression. This step is done provided that the smallest enclosing circle condition is respected. At this step of the process, we suppose that the set of data correspond only to one of the qualitative directions, but the segmentation will be explained later. From a set of positions, the linear regression gives a linear function of the form $y = ax + b$. The unknown constants a and b are found from S by exploiting the equation 6.1 and 6.2:

$$a = \frac{(|S|(\sum_{i=0}^{|S|} x_i y_i) - (\sum_{i=0}^{|S|} x_i)\left(\sum_{i=0}^{|S|} y_i\right))}{(n(\sum_{i=0}^{n} x_i^2) - (\sum_{i=0}^{n} x_i)^2)} \tag{6.1}$$

$$b = \frac{(\sum_{i=0}^{|S|} y_i)}{|S|} - a\frac{(\sum_{i=0}^{|S|} x_i)}{|S|} \tag{6.2}$$

Then, the direction can be inferred. To do so, the angle of the linear equation from the x-axis is calculated. It is achieved by doing the $arctan(|a|)$ of the slope. We can then use the qualitative framework as described previously.

6.4.2.1 Choosing between the two resulting directions

The linear regression does not allow us to have all the information on the qualitative direction. There are always two resulting possible directions that are usually opposites. In order to choose between them, we calculate a vector $\overrightarrow{p_n p_m}$ from the set of points $S = \{p_n(x_n, y_n), p_{n+1}(x_{n+1}, y_{n+1}), \dots, p_m(x_m, y_m)\}$ with the equation 6.3.

$$\overrightarrow{p_n p_m} = [x_m - x_n, y_m - y_n] \tag{6.3}$$

This vector, however imprecise as it may be, can then be used to take the decision. In fact, unless it is perpendicular to the found linear equation, we can always decide which directions to choose. To do so, all that is required is to compare the signs. The Figure 6.7 shows an example. As you can see, the basic direction was either East or West and the vector pointed toward NorthWest. That is, simply by using the abscissa sign we could choose.

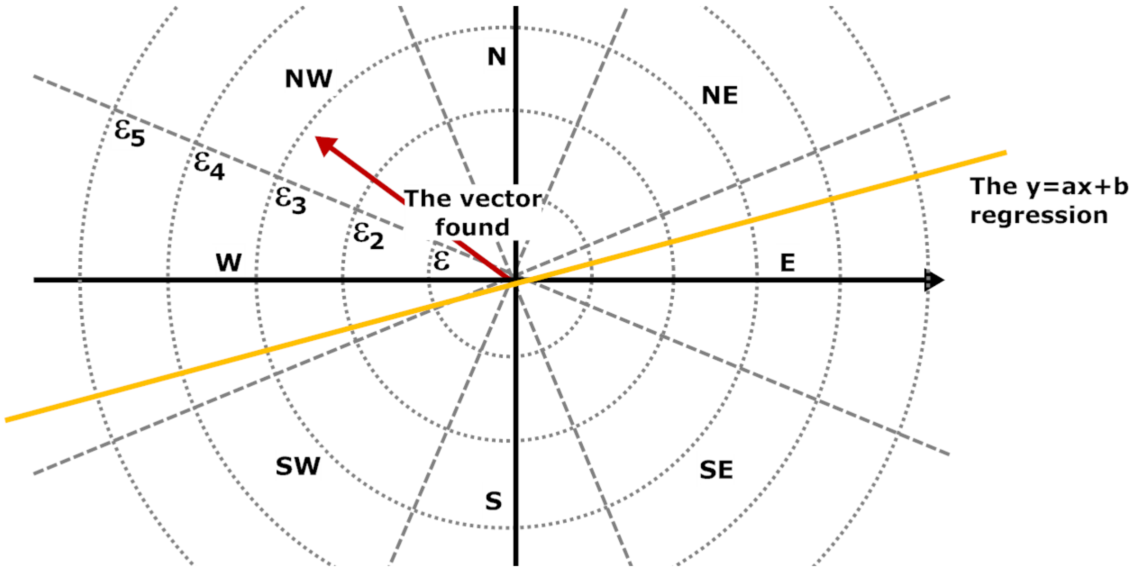


Figure 6.7 : Example of how the vector can help choosing the direction even when being very imprecise.

6.4.3 The correlation coefficient

The last piece of the puzzle in our algorithm is the correlation coefficient. This coefficient is used to evaluate to what extend the linear equation is modeled after the data. In our algorithm, it is crucial for the segmentation which is done by recursively calling the algorithm with half the data and then trying to combine the extracted basic directions. The correlation coefficient, denoted by φ is calculated using the equation 6.4:

(6.4)
$$\varphi = \frac{\left(n\left(\sum_{i=0}^{n} x_i y_i\right) - \left(\sum_{i=0}^{n} x_i\right)\left(\sum_{i=0}^{n} y_i\right)\right)}{\sqrt{n\left(\sum_{i=0}^{n} x_i^2\right) - \left(\sum_{i=0}^{n} x_i\right)^2} * \sqrt{n\left(\sum_{i=0}^{n} y_i^2\right) - \left(\sum_{i=0}^{n} y_i\right)^2}}$$

From that equation, we always obtain a value of φ comprised between -1 and 1. If the value is far from 0, the correlation is high between the data points. To be able to use it however, one last step needed to be done. We needed to learn the correct threshold of the correlation coefficient when an object was *idle* and therefore we recorded the values for several small sets of position when the object was still. We found out that on average, when the object was idle, $\varphi < 0.4$. However, assuming that an idle object is moving when it is not can be very damaging for the algorithms using the data (keeps in mind that we work on gesture recognition in the goal of exploiting the knowledge for assistive smart homes). Consequently, we used a slightly higher value (0.5) in our implementation to decide whether the object is idle over the period evaluated or moving in a certain direction.

6.4.4 ATOMIC GESTURE IDENTIFICATION

We now have seen all the necessary elements to proceed to the identification of what we call atomic gestures. The atomic gesture, in this context is a gesture composed of only one basic direction (or *idle*). The algorithm that we created is called Atomic Gesture IDentifier (AtomGID) and is shown below (Algorithm 6.1):

Algorithm 6.1: Atomic gesture identifier (AtomGID)

Input:	List of positions $L_p = [(x_n, y_n), (x_{n+1}, y_{n+1}), \dots, (x_m, y_m)]$
	Average error ε
Output:	List of atomic gesture $L_\alpha = [\dots]$
	Knowing that a gesture α is a structure $< (O_x, \varepsilon_y), \varphi >$

Compute the diameter of the smallest enclosing circle $L_p \rightarrow \delta$
If $\delta < \varepsilon$ **or** $|L_p| < 10$ **Then**
  Return $L_\alpha = (< (\,Idle, \varepsilon_0),0 >)$
End

Call AtomGID $(L_p([x_n, y_n], [x_{\frac{m}{2}-1}, y_{\frac{m}{2}-1}]), \varepsilon) \rightarrow R_g$
Call AtomGID $(L_p([x_{m/2}, y_{m/2}], [x_m, y_m]), \varepsilon) \rightarrow R_d$

Compute LinearRegression$(L_\alpha) \rightarrow \sigma$

Find QualitativeDirection $(\sigma, \overrightarrow{p_n p_m}) \to dir$
Compute CorrelationCoefficient$(L_\alpha) \to \varphi$

Call Combine$(Last(R_g), First(R_d), < dir, \varphi >) \to L_c$
Return $L_\alpha = (R_g[1, Last(R_g) - 1] + L_c + R_d[2, Last(R_d)])$

As you can see, the algorithm calls itself by dividing the set of points in two parts. The recursion, therefore, forms a binary tree and stop whenever the stop condition is triggered (less than 10 positions or smallest circle $< \varepsilon$). When that point is reached, the algorithm always returns an *idle* atomic gesture. As a consequence, all the leaves of the resulting tree are *idle* gestures.

6.4.4.1 How to merge the result

There is only one step that remains to be clarified in our algorithm. How do we combine the results of the two recursive calls? The function 6.5

$$\text{Combine}(Last(R_g), First(R_d), < dir, \varphi >) \to L_c \tag{6.5}$$

takes three parameters. It takes the found gesture at the current level of the tree, one from the left branch and one from the right branch. Remember that the left and the right result of the recursive calls are both list of atomic gesture that may contain more than one element. That is why the *Last* and the *First* of each are used. There is three possible outputs for the combine function. Either the function returns the current gesture, *idle*, or the list $L_c = [Last(R_g), First(R_d)]$. Here are the tests performed by the function:

A. If childs are identical (including *idle*) → current

B. If childs are *idle*
   a. If current $\varphi < 0.5$ → *idle*
   b. Else → current

C. If childs are different but not *idle*
   a. If the average of their $\varphi >$ current φ → two childs
   b. Else If current $\varphi < 0.5$ → *idle*
   c. Else → current

D. If one child is *idle*
   a. If non *idle* child's $\varphi >$ current φ → two childs
   b. Else If current $\varphi < 0.5$ → *idle*
   c. Else → current

6.4.4.2 Example of decision

To illustrate how the combine function works, let us look at a concrete example. The Figure 6.8 depicts a plausible tree structure that could have been created from the recursive calls:

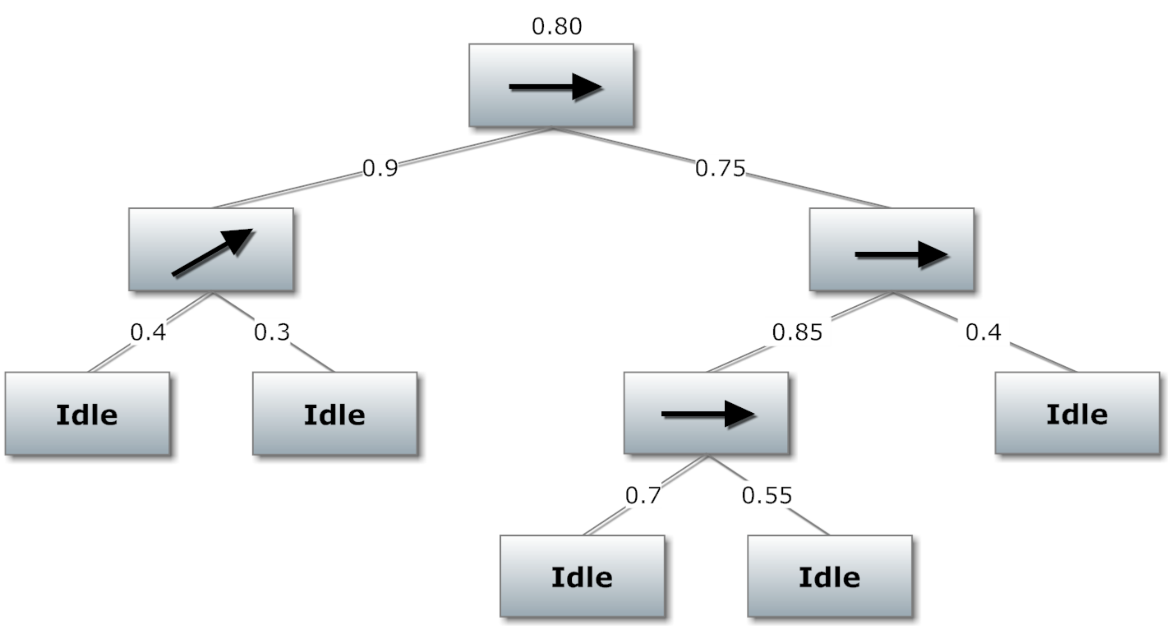


Figure 6.8: An example of tree resulting from the multiple regressions. The numbers are the correlation coefficients for each part of the dataset.

From that tree, the two pair of *idle* leaves would be merged. Then, the second half would be composed of *idle* with an East direction since the correlation coefficient of the child is higher. Then we would compare the $R_g = [(NorthEast, 0.9)]$ with the $R_d = [(East, 0.85), (idle)]$ with the root gesture $< East, 0.8 >$. The final list of gesture would be $L_\alpha = [(NorthEast, 0.9), (East, 0.85)]$ since the average of their correlation coefficient (0.875) is bigger than the one of the root (0.8).

6.4.4.3 Complexity analysis

From the early beginning of this thesis, we discussed the importance of keeping the complexity low. This algorithm might seem hungry, so we performed a complete analysis of its complexity. First of all, we need to use the Master theorem shown by the function 6.6 to evaluate the recursive calls:

(6.6)
$$T(n) = a * T\left(\frac{n}{b}\right) + f(n)$$

The parameter a is the number of recursive calls. The parameter b is the division factor. The $f(n)$ is the amount of work done the recursive call. There are four steps in our function. The regression is done in $O(n)$ since the set of positions is scanned once. The calculation of the vector is constant $O(1)$ and the correlation coefficient also requires one scan of the list of positions $O(n)$. It is the smallest enclosing circle which is the most hungry: $O(n^2)$. Therefore, we have

$$T(n) = 2 * T\left(\frac{n}{2}\right) + O(n^2)$$

The theorem says that whenever $f(n) = O(n^k)$ and that $a < b^k$ the complexity is equal to $O(f(n))$. The global complexity is $O(n^2)$. While it is satisfactory for the scope of this thesis, it would be possible to improve the performance by using a different algorithm for the smallest enclosing circle. Indeed, a complex solution in $O(n)$ exists.

6.4.5 COMPOSITE GESTURE IDENTIFICATION

The final part of our method consists in matching the list of identified atomic gesture to the gestures in the dictionary. For this part, the literature proposes a variety of methods developed through years of research [9, 12]. For this work, we preferred to keep that part simple as it is not the main challenge to gesture recognition from RFID, and because we only needed to exploit atomic gestures for our spatial data mining model. Once we are able to find basic directions and to perform segmentation, we rely on a standard method. Our gesture dictionary is a set of finite state machines (FSMs) representing each gesture. The selected ongoing gesture is the state machine that matches the sequence of atomic directions identified. Remember that the output of AtomGID is a list similar to the Figure 6.9:

…	$\leftarrow$	$\uparrow$	$\leftarrow$	$\uparrow$	…

Length	Direction	Length	Direction
ε_2	$\uparrow$	ε_4	$\leftarrow$

Figure 6.9: Example of output list of gesture from AtomGID

One thing noteworthy to mention is that the matching is not strict. If a sequence comprising of small *idle* moment does not match any gesture, they are progressively eliminated until either the sequence match or until no more remain. Contrary to most work in the literature, we do not assume that a gesture was intended. That is because in our context, the user is a normal resident or a resident with a cognitive deficit that does not purposely intend to perform a gesture with the objects he moves.

6.5 IMPLEMENTATION AND VALIDATION

We implemented the model with Netbeans IDE in the Java language in order to be able to test it. The software created was built to be able to switch easily from different gesture recognition algorithms and was used with our other models. The software is also able to select the set of positions from distinct sources. It can be connected directly to the LIARA's smart home SQLServer database, to a local MySQL database or simply open a text file containing the positions. The software's GUI is shown on Figure 6.10. The software is also able to generate simple reports on the gesture recognition (see Appendix A). In addition to being exploited for our spatial data mining model, this particular algorithm was exploited in two sets of experiments that are described in this section.

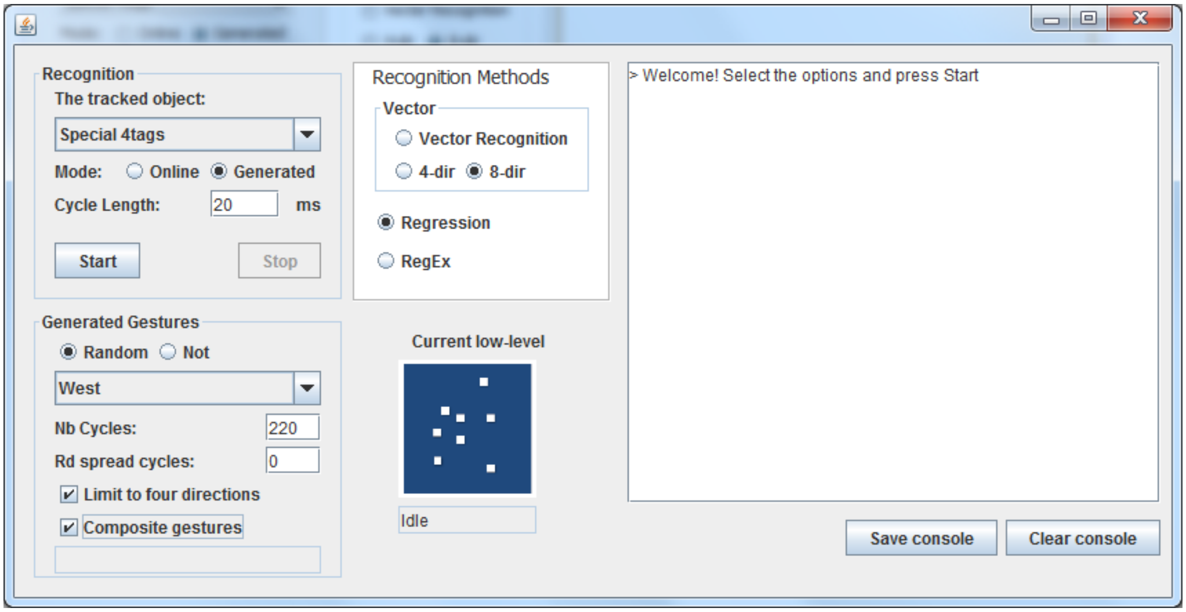


Figure 6.10: The GUI of the gesture recognition software.

6.5.1 Experiments with a simulator

As a preliminary set of experiment, we implemented a simulator that generated gestures to be recognized by our new method. We decided to do so to first have an estimation of the performance of our algorithm. Moreover, the simulator enabled us to do an extensive amount of tests in a short time interval that the complex protocol needed with human subjects would never allow us to do. The generator works simply by randomly selecting a FSM corresponding to a gesture in the dictionary and computing the next position using the parameters. These comprise the generation speed (ms), the object speed (cm/s), the gesture length (seconds) and the average positioning error. The error is used to generate noise. For example, if the object should be at (10, 0) and the error is $\pm$14cm, then the generated position would be $(-4 \leq x \leq 24, -14 \leq y \leq 14)$.

Our algorithm is able to detect gestures composed of any sequence of basic directions. However, we wanted to compare our performance with the only other RFID based gesture recognition model so we decided to reproduce their experimental setting. In their experiment, Asadzadeh et al. [199] used only four basic directions. On average, their gestures lasted 4.5 seconds at 20cm/sec. Their localization algorithm was, although unusable in our context, more accurate than our method with approximately an error of 10cm. We used that error for the simulation but reduced the length of the gestures to approximately 40cm per basic directions. Moreover, to test our segmentation, we added a random variable to the distance (-10 to +20 cm). We also added the *idle* gesture in the dictionary because in a realistic context, most of the time, objects are idle. The Figure 6.11 shows the set of gestures.

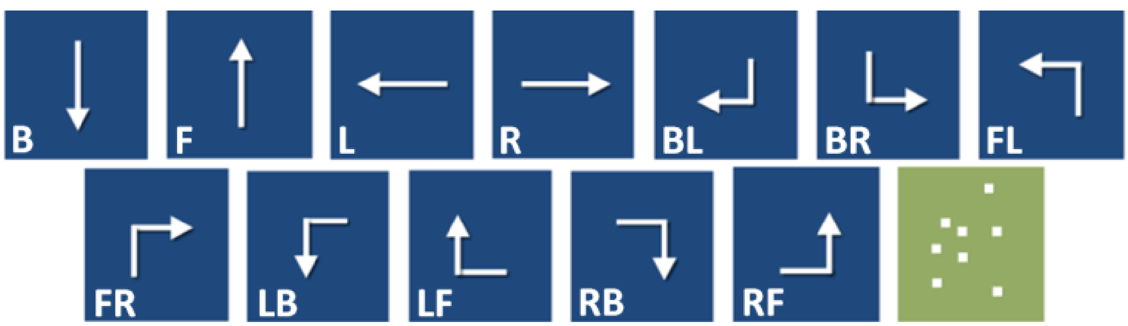


Figure 6.11: Example gestures used for the experiments. Eight are composed of two directions, four of only one. The last on the picture is *Idle*.

We let our generator work for about 2000 gestures generated randomly, and we obtained positive results (87.5% success). Table 6.1 details the results that were obtained from this set of tests. The most important thing to understand is that recognizing the directions was not difficult; most of the errors were due to the process of segmentation. It means that with the same assumption (no need for segmentation) of the team of Asadzadeh et al. [199] our new method would have performed better. The other errors are mostly misclassification between the *idle* and real directions.

Table 6.1: The results obtained from the simulation.

	True Positive	False Negative	False Positive	%
Idle	157	14	**116**	91.8%
F	149	26	8	85.1%
B	142	23	11	86%
L	145	14	13	91.2%
R	151	20	16	88.3%
LF	156	21	18	88.1%
LB	163	12	12	93.1%
BR	139	25	13	84.8%
RB	126	29	9	81.3%
FL	158	21	17	88.3%
FR	155	13	14	92.3%
BL	137	29	11	82.5%
RF	148	27	16	84.6%
	1926	**274**	**274**	**87.5%**

6.5.2 Validation with human in the smart home

Since the results obtained from the gesture generator were good, we decided to conduct a first set of experiments directly in the smart home. For that purpose, a human subject was asked to perform each gesture a total of ten times. The protocol of Asadzadeh et al. [199] was exactly reproduced. The human was using a standard cup of coffee with four RFID tags on it and the system used the position of the cup to infer the gesture. The cup was initially put on the kitchen counter for the tests. A physical guideline was put beside the cup to show the human subject the average distance and direction that should compose the gesture. Therefore, he had only to approximately move the cup following that guideline. The appropriate guideline was installed before each series of tests (for a gesture). No particular instructions were given to the human for the initial position he should begin the gesture. Sometime the human has held the cup in his hand before start and other times he just performed the gesture right away on the kitchen counter. The results of the experiments can be seen on Table 6.2.

Table 6.2: Results from the human tests.

R	L	F	B	FR	BL	FL	BR	RF	LB	LF	RB	Idle	Total
9	8	7	9	7	8	8	7	7	8	7	7	9	77%

As shown on the table, the results are slightly worse than those obtained with the generator. There are two explanations to this. First, the noise in the data obtained from the RFID system is not random. It means that often, when the data begin to be inaccurate, it moves in a distinguishable direction. That creates some issues with the segmentation. Secondly, in a realistic environment, there are unpredictable interferences that lead to recognize directions that never happened. For example, if the human is hiding one or many antennas for a certain time, this might lead to a significant modification of the estimated position and thus to identify a movement that is not real.

6.6 Chapter conclusion

In this chapter, we described AtomGID an algorithm that can extract the basic movement information from a set of noisy positions obtained from a passive RFID localization algorithm. This algorithm is flexible and scalable since it only requires to know the average error to work with a localization algorithm, and it exploits a well-established QSR framework to describe the atomic gesture. We tested this algorithm by implementing a classical gesture matching solution exploiting FSMs. We have shown that it performs better than the only other RFID based solution despite the higher difficulty of the challenge. In particular, the most important contribution of this algorithm is the ability to perform the segmentation of consecutive atomic and composite gestures.

The goal of this algorithm for this thesis was not to perform gesture recognition or at least not composite gesture. The goal was to be able to transform our dataset of positions generated from all the smart home objects into high-level spatial information. With this algorithm, we can do this easily and pass to the final step of the spatial data mining: the application of a data mining model. Remember also that we discussed that we wanted to create an aggregation solution. The RFID localization algorithm generates around five millions positions per day for about 25 objects in the smart home. The AtomGID algorithm significantly helps us in reducing the size of our data warehouse by eliminating the data which is not interesting. Supposing that only one object at the time is moving and that an atomic gesture usually requires at least 1 second, the amount of data for a day passes to only 3600 which is a 99.999% reduction. In the next chapter, we explain how to exploit the prepared data to perform clustering with an extension of the flocking algorithm.

CHAPTER 7

CLUSTERING FROM EMERGING MOVEMENT

Our journey through spatial data mining made us discover the technologies that constitute smart home, how to effectively exploit passive RFID for localization and how large dataset of noisy positions can be transformed into high level movement information. It is now near end and all that remains is to proceed to the final step of the spatial data mining methodology: the application of a data mining algorithm. To do so, we must first understand what we have come with so far. The data warehouse we collect is composed of the simple events obtained from binary sensors. We ignored the data from the other complex technology such as the ultrasonic sensors and the smart power analyzer. The data warehouse is also assumed to containing a list of movement extracted from RFID localization and gesture recognition. Another assumption is that only one object is active at the time and thus only the movements on the active object are extracted. In this chapter, we describe an extension to the well-known flocking algorithm in order to perform the task of clustering.

The remainder of this chapter is divided as follows. The section 7.1 introduces the problem that needed to be addressed by returning to the work presented at the beginning of this thesis. It also introduces the Flocking algorithm which served as a basis to design our solution. The section 7.2 describe formally the new extension to the Flocking algorithm. In particular, two new rules are described and method to reduce the complexity is explained. The section 7.3 discusses the implementation of that new model at the LIARA laboratory and describes in details two sets of experiments that were conducted to validate our complete spatial data mining model. Finally, the section 7.4 concludes this final contribution chapter.

7.1 CLUSTERING MOVING DATA

As we have seen in Chapter 3, there are three main families of data mining algorithms: the decision trees, the association rules and the clustering. Each of them possesses their advantages and inconvenient. However, the first and most important criterion that distinguishes them is if we are in a supervised or unsupervised context. Decision trees (DTs) are purely supervised method and cannot be constructed if the classes are not known in advance [16]. Association rules are generally considered as unsupervised methods, but as we have seen before, in the context of smart home, the training must be done separately for each ADLs and thus is in that broad sense very similar to supervised methods. Since in our context we work with a fully unsupervised training data set, the clustering remains the only interesting alternative.

We discussed the main clustering algorithms and gave an execution example for the well-known K-Means algorithm [109]. As we have discussed, the main problem with this algorithm is the requirement to specify the parameter *k* that defines the number of clusters to form. In our context, we cannot assume that the number of possible ADLs is known. Some other clustering algorithms, that are said hierarchical, can find this parameter for the user. It is the case for Expectation Maximisation (EM). This algorithm, which was introduced in 1977 by Dempster et al. [200], enable to find the maximum likelihood of probabilistic models' parameters. The maximum likelihood is a statistic which is used to estimate the probability distribution of sample data. With it, the algorithm is able to estimate the potential number of clusters if not specified by the user. Even so, the algorithm is computationally hungrier, and it does not always find the number of clusters. We believe that new clustering algorithms that find efficiently the clusters without having the number specified are required. In particular, we are interested to algorithms that could reflect the natural spatial aspect that was investigated into this thesis.

7.1.1 THE FLOCKING ALGORITHM

For this last step of our spatial data mining method, we explore the Flocking algorithm that was designed in 1987 by Craig Reynolds [163]. This algorithm is exploited to simulate the behaviors of animals moving as a group. The Flocking reposes on the emergence principle as we often expect from multi agent systems. The idea of emergence is that out of a multiplicity of relatively simple interactions can arise complex systems and patterns. Thus, an observer looking at such a system will think that it is complex even so the rules are simples. In the Flocking algorithm, the virtual agents, that are called *boids*, follow their internal rules without having any goal to accomplish and by seeing their environment only partially. They usually evolve in an unbounded environment and they are only able to see some boids that are near them: their neighbors. The Flocking is based on three rules: the alignment, the separation and the cohesion. These rules, which are shown on Figure 7.1, are used by the agent, every time he can reason, in order to plan its movement.

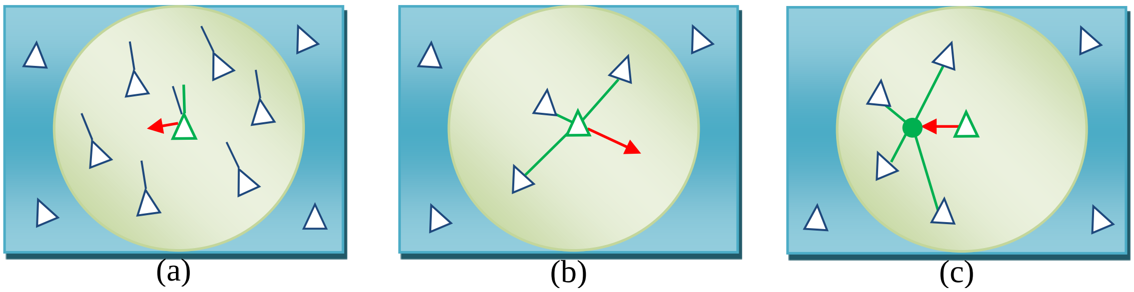


Figure 7.1: Basic rules of the Flocking (a) Alignment (b) Separation (c) Cohesion.

The alignment (a) calculates the average direction of the neighbors in order to align the group in a similar direction. The separation separates the boids from each other in order that they do not just agglutinate in a single spot. The cohesion, on the opposite, bring each boid closer to the mass center of its neighbors, so they remain a group. More details are given in the next section.

7.1.1.1 Flocking in data mining

The Flocking algorithm has been mostly used in the video game industry, in biological simulation and in animation movies [201]. Nevertheless, some authors have tried to exploit it for data mining applications. However, the basic algorithm cannot be directly used for that purpose. Indeed, the three rules that enable the boids to evolve freely ultimately make all the boids converge into a single group for most of the time. Hence, it is a necessity to define at least one new rule to the basic reasoning process of the boids. This rule should be using a discrimination criterion which will split the boids into many groups. The idea remains the same as for any clustering algorithm. A function that computes a *distance* or a *similarity* is needed to form the clusters.

In their work on document classification, Cui et al. [202] introduced that exact concept of similarity to work with the Flocking. In their case, they created a boid for each document to classify and let them evolve in a two-dimensional environment. The similarity function that they introduced is based on a words dictionary. The documents are sorted by comparing the words they contain. They demonstrated that the clustering of documents with the Flocking algorithm could help in reducing dependency toward human experts. Bellaachi & Bari [90] also worked on an extension to the Flocking algorithm in order to use it for data mining. In their work, the Flocking was exploited in order to detect outliers in cancer microarrays. The ultimate goal was to correct these outliers. The general idea was to apply the Flocking rules on an agent who was away from his group and thus removing the problem.

These first efforts have shown the potential of the Flocking algorithm for clustering. In particular, the addition simple rules enable groups to form and emerge from the apparently random movement of simple agents. In the next section, we explain how the Flocking algorithm was modified for this thesis. A first set of tests was conducted on a dataset made of binary events and was published in [107].

7.2 A FLOCKING EXTENSION FOR CLUSTERING

As we said in the introduction, the basic flocking model consists of three simple steering rules: alignment, separation and cohesion [163]. These rules are executed at each reasoning iteration by each individual agent. With these three rules, each agent gets closer to his neighbors indiscriminately adopting a herd pattern. Therefore, only one cluster is formed after several iterations. For our clustering purpose, we conceived two new rules: similarity and dissimilarity. These rules allow us to create many clusters because similar agents follow each others, and dissimilar agents tend to separate. In addition, in the basic model of flocking, if an agent finds himself alone, it stops moving. To correct this problem, we modified the rule of alignment so that the agent continues to move straightforward if it is left alone. Besides, another limitation of this algorithm is its base complexity, which is in the order of $O(n^2)$. However, by using a technique that is called cell-space partitioning we could reduce the complexity to $O(n)$ [201]. Linear complexity is desirable due to the large amount of data that must be processed. In the next subsections, we describe the five rules that we proposed, with their formalization.

7.2.1 ALIGNMENT, SEPARATION AND COHESION FORCES

We already described informally the three basic rules of the Flocking: alignment, separation and cohesion. Now we will define mathematically how each of them is calculated to produce a force used to drive the heading vector of each boid.

Alignment force attempts to keep an agent's heading aligned with its neighbors. The force is calculated by first iterating through all the neighbors and averaging their heading vectors. Considering that k is the total number of current agent's local neighbors, $\overrightarrow{Ha}$ is the agent's heading force, and $\overrightarrow{Hn}$ is a neighbor heading force, the mathematical implementation of $\overrightarrow{F_A}$, the force driven by alignment rule, is the equation 7.1:

(7.1)
$$\overrightarrow{F_A} = \frac{1}{k+1}\left(\overrightarrow{Ha} + \sum_{n}^{k}\overrightarrow{Hn}\right)$$

Separation force creates a force that steers an agent away from those in its neighborhood region. When applied to a number of agents, they will spread out, trying to maximize their distance from every other agent. Considering that k is the total number of current agent's local neighbors, $\overrightarrow{Pa}$ is the current agent's position vector, and $\overrightarrow{Pn}$ is a neighbor's position, then $\overrightarrow{F_P}$ is the force driven by separation rule defined by the mathematical formula 7.2:

(7.2)
$$\overrightarrow{F_P} = \sum_{n}^{k}\frac{\overrightarrow{Pa} - \overrightarrow{Pn}}{\left\|\overrightarrow{Pa} - \overrightarrow{Pn}\right\|^2}$$

Cohesion produces a steering force that moves an agent toward the center of mass of its neighbors. This force is used to keep a group of agents together. We calculate the average of the position vectors of the neighbors. This gives us the center of mass of the neighbors, the place the agent wants to get to, so it seeks to that position. The force $\overrightarrow{F_C}$ is the mathematical formula 7.3 where $\overrightarrow{Pa}$ is the agent's position, Ms is the agent's maximum speed (a predefined constant), $\overrightarrow{Va}$ is the agent's velocity, and , $\overrightarrow{CoM}$ is the center of mass of the boid. The $\overrightarrow{CoM}$ is determined by k the total number of current agent's local neighbors, and $\overrightarrow{Pn}$ is a neighbor's position vector (7.4).

(7.3)
$$\overrightarrow{F_C} = \left(\frac{\overrightarrow{CoM} - \overrightarrow{Pa}}{\left\|\overrightarrow{CoM}\right\|} * Ms\right) - \overrightarrow{Va}$$

(7.4)
$$\overrightarrow{CoM} = \frac{1}{k}\left(\sum_{n}^{k} \overrightarrow{Pn}\right)$$

7.2.2 DISSIMILARITY AND SIMILARITY

In addition, to the basic rules we conceived two new rules that enable to calculate a similarity force and a dissimilarity force. These forces are also computed every iteration in order to drive the heading force of each boid. Here how they work:

Dissimilarity force creates a force that steers an agent away from those in its neighborhood like the separation rule, but only between dissimilar agents. The mathematical implementation of the dissimilarity force extends the separation rule by modulating $\overrightarrow{Pa} - \overrightarrow{Pn}$ in function of $d^*(a,b)$ which represent the dissimilarity between a couple of agents a and b (7.6). The function $d^*(a,b)$ is the normalized Euclidian distance represented by the function 7.5 $distance(a,b)$ which compare agents' data in order to compute a certain distance between a and b. In the Euclidian equation, n denotes the dimension of the vector space related to the attributes of the learning data set.

(7.5)
$$distance(a,b) = \sqrt{\left(x_1^a - x_1^b\right)^2 + \cdots + \left(x_n^a - x_n^b\right)^2}$$

(7.6)
$$\overrightarrow{F_D} = \sum_{n}^{k} \frac{(\overrightarrow{Pa} - \overrightarrow{Pn}) * d^*(a,b)}{\left\|\overrightarrow{Pa} - \overrightarrow{Pn}\right\|^2}$$

Similarity force produces a steering force, likewise to the cohesion rule, but only between similar agents (7.7). However, a new center of mass ($\overrightarrow{\mathrm{CoM_S}}$) calculation, described by equation 7.8, replaces the one from the traditional cohesion rule.

(7.7)
$$\overrightarrow{F_S} = \left(\frac{\overrightarrow{CoM_S} - \overrightarrow{Pa}}{\left\|\overrightarrow{CoM_S}\right\|} * Ms\right) - \overrightarrow{Va}$$

(7.8)
$$\overrightarrow{CoM_S} = \frac{1}{k}\left(\sum_{n}^{k}\left(\left(\overrightarrow{Pa} - \overrightarrow{Pn}\right) * S(a,b) * \overrightarrow{Pn}\right)\right)$$

In this equation, k is the total number of current agent's local neighbors, Pa is the current agent's position, Pn is a neighbor's position, and $S(a,b)$ the similarity between agents a and b. The similarity the normalized Euclidian distance $d^*(a,b)$ as follows (7.9):

(7.9)
$$S(a,b) = 1 - d^*(a,b)$$

7.2.3 RESULTING FORCE

To achieve a complete Flocking behavior, the results of all rules are weighted and summed to give a steering force that will be used by the current agent for calculate his next velocity. The weights have been learned from synthetic datasets in order to maximize the efficiency and the accuracy of the clustering method. In the experiments presented later in this thesis, the values that were used are shown on Table 7.1:

Table 7.1: Weight of each force.

Force	*Similarité*	*Dissimilarité*	*Cohésion*	*Séparation*	*Alignement*
Poids	1.5	3.0	0.8	1.2	0.5

If w_X represent the predefined weight values, $\vec{F}$ our flocking force can be seen as a resulting force from the linear combination of all the other forces as defined in the next equation (7.10):

$$\vec{F} = w_S\overrightarrow{F_S} + w_D\overrightarrow{F_D} + w_C\overrightarrow{F_C} + w_P\overrightarrow{F_P} + w_A\overrightarrow{F_A} \tag{7.10}$$

It is worth mentioning that the forces are summed in this very order. It is crucial because whenever the addition of a force exceeds the maximum force that an agent can have, the remaining force will not be added. The force $\vec{F}$ can be represented by the Algorithm 7.1:

Algorithm 7.1: The Flocking final force for an agent.

Input: The agent ag

Output: $\vec{F}$ the final resulting force

Find $neighbors(ag) \rightarrow n[]_{ag}$

If $\left|n[]_{ag}\right| = 0$ **Then**

  Return $heading(ag)$

Else

  Create an ordered list $forceRules[]$

  Compute $w_S\overrightarrow{F_S} \rightarrow forceRules[]$

  Compute $w_D\overrightarrow{F_D} \rightarrow forceRules[]$

  Compute $w_C\overrightarrow{F_C} \rightarrow forceRules[]$

  Compute $w_P\overrightarrow{F_P} \rightarrow forceRules[]$

  Compute $w_A\overrightarrow{F_A} \rightarrow forceRules[]$

  For all $\vec{F}_\mathrm{i}$ in $forceRules[]$

    If $forceCanBeAdded(\vec{F}, \vec{F}_\mathrm{i})$ **Then** $\vec{F} + \vec{F}_\mathrm{i}$

    Else Return $\overrightarrow{MAX\_FORCE}$

  End

End

Return $\vec{F}$

This algorithm is dependent on the function $forceCanBeAdded(\vec{F}, \vec{F}_i)$ that verifies if it is possible to add the next force of the list. The function takes two vectors as input. The first one is the target force and the second one is the force to add. If the addition of the second force on the first does not exceed the limit, the function return true and then the addition is made. Otherwise, the function returns false and the $\overrightarrow{MAX\_FORCE}$ vector is returned. This limitation is done in order for the movement of the boids to be natural. The $\overrightarrow{MAX\_FORCE}$ vector is the maximal steering force that can be applied during an iteration on a boid heading vector. Without a limit, the boids could drastically turn randomly and thus quitting the herd movement.

7.2.3.1 Cell-space partitioning optimization

In the beginning of this section, we mentioned that it was possible to reduce the time complexity of the Flocking algorithm with a technique involving spatial partitioning. There are many methods to do that, but the general idea is always the same. Instead of testing all boid against each other in the environment, we only test boid that are in the same effective partition. The technique that we implemented is called cell-space partitioning. With this method, the 2D space is divided into a number of square cells, and each cell possess a list of the boids that are in it. When a boid enters a cell, he is added to the list and removed from the cell he was previously in. In this way, the neighbors of a boid are found simply by looking into the list of the cells in its neighborhood. The steps of the method are described below:

I. First, the boid radius is approximated with a box as shown on Figure 7.2.

II. The cells that intersect with the bounding box are tested to see if they contain boids.

III. All the entities found in step two are tested to see if they are within the neighborhood radius. If they are, they are added to the neighbor list.

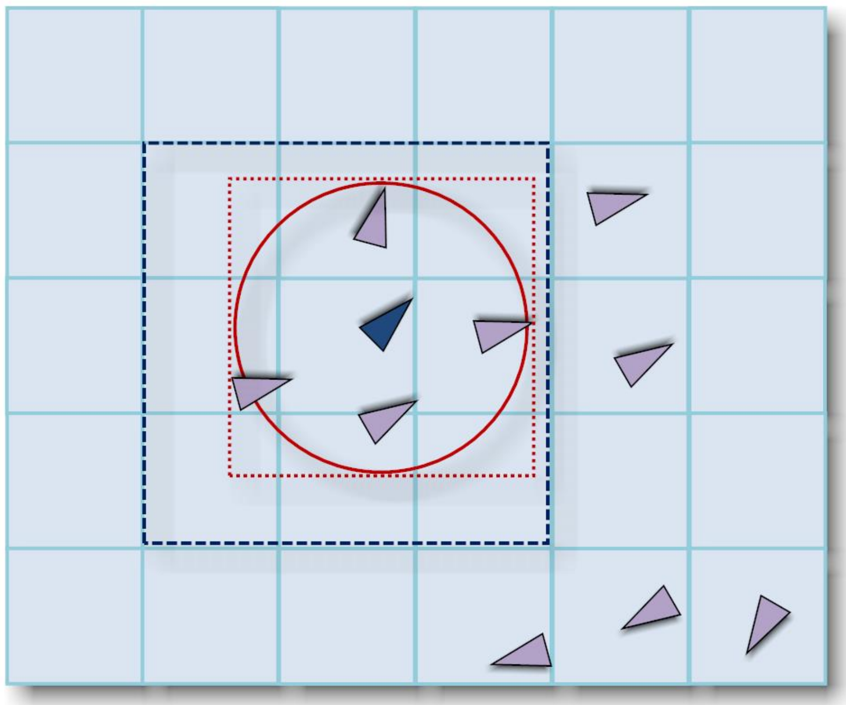


Figure 7.2: The cell-space partitioning technique.

7.3 IMPLEMENTATION AND VALIDATION

In order to test our Flocking extension for clustering, we used the Eclipse IDE with Java to create a virtual environment. This environment is 500 pixels wide per 500 pixels high and is divided into 64 logical cells where the boids can evolve freely. On the Figure 7.3 you can see the top part of the environment with some information. On the top left, you can see the weight value for each of the forces. In addition, there is the *vis:140* parameter that is used to tweak the vision of the boids. In that case, it means that each boid can see 140 pixels around him. On the top right corner, some statistics are shown purely to help the user. There is the number of iteration the boids have evolved, the number of boids (agents) and the number of agents who are parts of a cluster. The last two information are only displayed when the software is in testing mode. The first shows the average intraclass distance and the second shows the classification success in percentage.

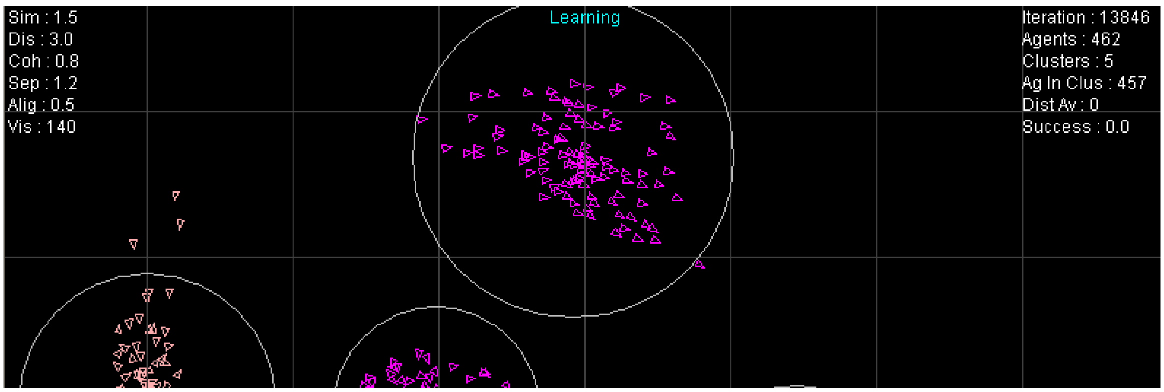


Figure 7.3: Partial vision of the environment with the parameters.

In order to valid our implementation, we designed two sets of experiments. The first one exploited a more classical approach of data mining to activity recognition. The activities chosen were high-level and the dataset was constituted of simple events from basic sensors. The reason we did so was to be able to compare our results with the literature. Indeed, with our spatial information and our fine-grained activity, it is much harder to compare the results since, in the best of our knowledge; we are the first to implement such a model. The second set of experiments aimed a validating that we could effectively exploit our movement based dataset to perform unsupervised learning of ADLs with our new Flocking algorithm. Again, notice that this contribution, the Flocking clustering model, could be exploited independently from the two first parts and is generalized for other potential applications.

7.3.1 EXPERIMENTS WITH CLASSICAL DATASET

To test the efficiency and accuracy of our new model, we have conducted extensive experiments in our smart home. A participant was asked to perform specific scenarios without precise indications. A description of the tests' dataset is given on Table 7.2.

Table 7.2: The scenario dataset.

Scenario name	*Number of events*
Cook for lunch	103
Cook for dinner	98
Go to the toilet	14
Read a book	12
Sleep in the bedroom	16

Events represent changes in the state of the different sensors of the environment. This is why simple activities like reading a book contain fewer events than tasks needing more complex operations (cooking). Indeed, fewer sensors change of states when a resident is only reading a book. Each scenario has allowed to generate a text file that contains 6 data per row, and each row symbolizes an event. The Table 7.3 contains a part of the scenario cook for lunch. The columns X and Y are the position of sensors; they are ranged between 0 and 600. The time parameter is expressed in minutes from 0h00; 710 min is 11:50 AM for example. The Appendix B describes de sensors and the fixed position that was associated to each of them.

Table 7.3: Little part of cook for lunch scenario.

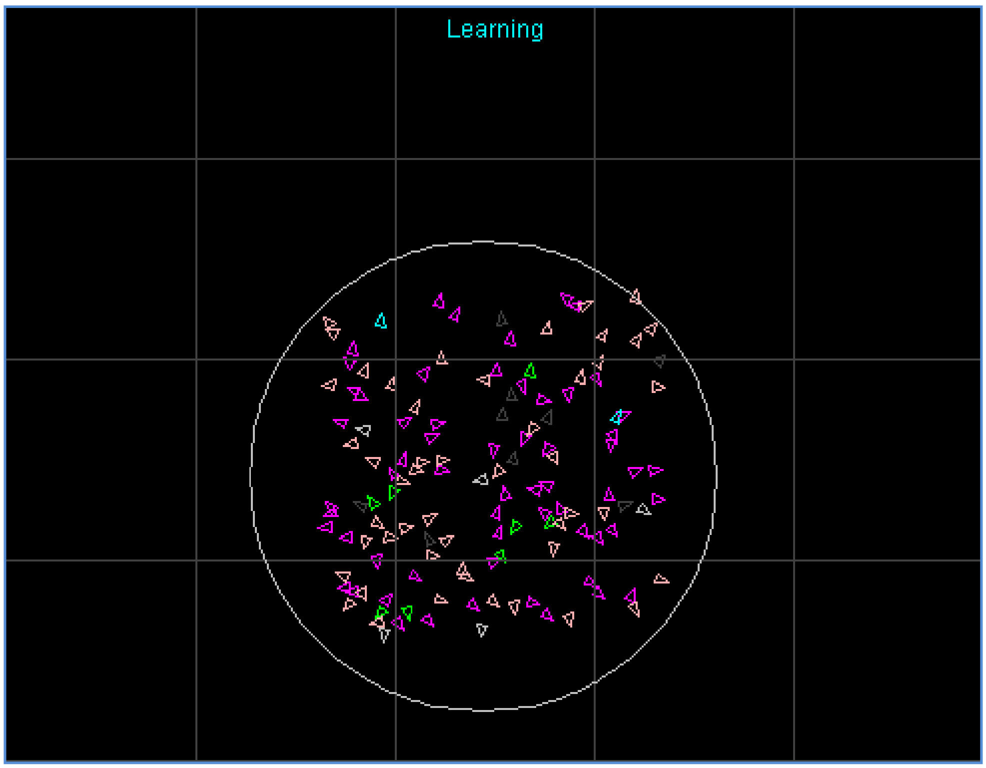
Type	*Name*	*Location*	*X*	*Y*	*Time*
Sensor	TC2	Kitchen	250	250	710
Sensor	CA5	Kitchen	300	100	710
...	...	...	...	...	...

7.3.1.1 Experimental setup

For the experimental phase, we decided to run two other clustering algorithms with our new flocking based model. Our choice fell on K-means because of its high efficacy. We also chose Expectation-Maximization (EM) which is also a clustering algorithm (among other things) that can either be used with a fixed number of clusters or that can efficiently estimate it. For the three versions, we used the same dataset and the Euclidian distance measure. We did not modify EM nor K-means. For the Flocking clustering algorithm, each data is represented as one boid. Each boid can only sense other boids located within its neighborhood distance. Higher is the sensing distance and faster the clustering result emerges (i.e. it becomes easier to find similar boids). However, at the same time, each agent needs more computational time to generate its moving direction and speed at each iteration. Likewise to others data mining techniques, we separate the learning phase and the testing phase. In the learning phase 2/3 of data are used for creating the clusters, and 1/3 for testing these clusters.

The initial distribution of the experimental dataset is shown on Figure 7.4 (a). As can be seen, at the beginning of the learning phase one unique cluster exists and contains all agents. We let the agents move freely according to the movement forces once per iteration (every 40ms). At the end of an iteration, clusters are built recursively from the agent list. If the square distance between two agents is under to their view distance, they are in the same cluster. When all agents have been assigned to a cluster, the center and the radius of the clusters are calculated relative to the agents inside. The Figure 7.4 (b) shows that five clusters are created in the learning phase after several iterations. Each of them is relative to a scenario presented in dataset. This means that our algorithm succeeds in finding the exact number of activities during the learning phase, without supervision. After several iterations, if there are no more changes in the number of clusters, the testing phase begins.

(a)



(b)

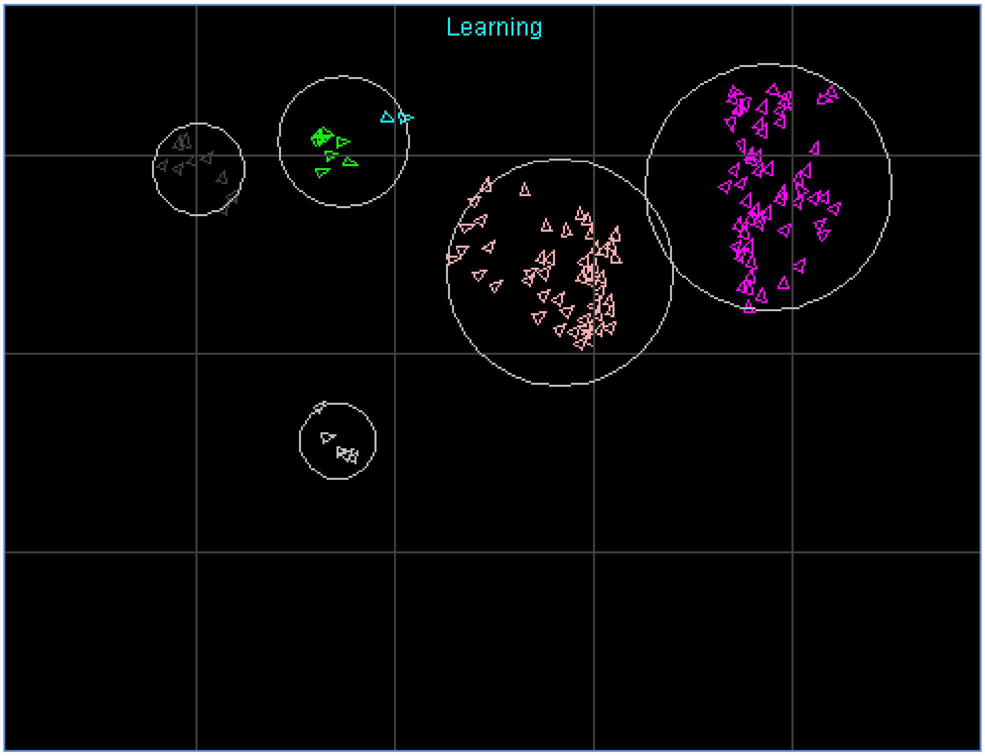


Figure 7.4: (a) Initial data distribution in the learning phase; (b) Result after 1000 iterations.

7.3.1.2 Testing phase

This phase aims to control the quality of our clusters. At the beginning of the testing phase, a representative agent is created for each cluster. It computes the average of the data of all agents in their clusters. After the creation of these representative agents, all others agents are destroyed. The 1/3 of the dataset unused replaces these destroyed agents. In this phase, clusters are built from each representative agent created at the beginning. This allows confirming if the clustering is efficient or not. The percent of success of each cluster is calculated every frame, and the average of all cluster is inferred. After 25 measures, an average of these success rates is saved to a log file. The success of a cluster is the number of agents of the same class divided by the total number of agents in the cluster.

7.3.1.3 Experimental results

We ran the algorithm 10 times with these 5 real case scenarios, and we computed the average of the results. The Table 7.4 shows a part of our results at certain iteration.

Table 7.4: The average results of Flocking clustering Algorithm on real datasets.

Iterations	*3371*	*4591*	*5753*	*6897*	*8064*	*8882*	*10227*	*13404*
Success	63.6%	71.8%	80.4%	87.2%	88.6%	89%	90%	92.5%

An iteration is the calculation of the combined forces $\vec{F}$ for all agents, and the update their position. The success rate is expressed between 0 and 1. The closer the value is to 1, the purer the cluster is. As one can see, the success rate rises rapidly at the beginning across the iterations to around eighty-seven percent and then progressively to reach the threshold of ninety-two percent after approximately thirteen thousand iterations. This means that for the five activities that the algorithm detected during the learning phase, the average purity of clusters is around ninety-two percent. To compare, the Table 7.5 presents the results obtained by using K-means and EM (Expectation Maximization) algorithms for the exact same dataset.

Table 7.5: The results of K-means and EM algorithms on real datasets

Algorithm	*Iterations*	*Success*	*Number of clusters*
K-Means	5	60.3%	5 (set at start)
EM (k set)	5	76.0%	5 (set at start)
EM (k unset)	5	62.4%	7

These algorithms are more efficient in terms of iterations number. However, they cannot achieve a high success rate partly because of the small number of data contained in our dataset. Moreover, K-means requires knowing the number of clusters at the start. Furthermore, EM cannot find the exact number of activities, and its success rate is lower if we don’t set the number of clusters. Another important thing to note is the difference in the way the iteration metric is calculated. While an iteration in classic data mining approaches represents the complete reattribution of the elements to cluster, in our flocking extension, it is only a small movement for each boids. Therefore, the difference in performance of our algorithm is not as big as it may seem. Besides, as stated in the beginning, the flocking computational complexity is linear as for the partitioning algorithms with a fixed number of clusters and as opposed to hierarchical methods, which are polynomial.

7.3.2 EXPERIMENTS WITH THE SPATIAL DATASET

The final step of this thesis project was to perform a last set of experiments on the global spatial data mining method. To do so, we exploited the localization algorithm presented in Chapter 5 and the AtomGID algorithm presented in Chapter 6 to collect a movement dataset for five fine grained activities of daily living. These activities are: preparing a bowl of cereals, preparing an instant coffee, making a burger, preparing pasta and washing hair. We chose to test this version on fine grained activities, in order for the spatial aspect to be useful. Moreover, we had to choose kitchen activities since our elliptical localization is only usable in that room with the current hardware we possess. In addition to the higher difficulty of learning and recognizing fine grained ADLs, the collection of spatial data over binary events led us to a much bigger number of agents, even if, in fact, the AtomGID algorithm reduces 99.999% the size of the data warehouse builds purely from the localization algorithm (at 1 reading per 20ms).

To conduct this set of experiments, we basically reproduced the protocol that was described through the section 7.3.1. The results are presented in the remaining of this section and thereafter an assessment of the developed Flocking algorithm will be provided.

7.3.2.1 Similarity of movement

To better understand the Flocking work on the movement of objects, it is important to explain the notion of similarity for qualitative directions. Quantitative directions are represented by a vector. This vector, when placed at the origin of a 2D Cartesian space will have an angle of a certain degree with the abscissa. This angle will be a real number between 0 and 360. If the directions to be compared are two vectors, we can easily compute the angle they make and use that scale to define the similarity. For example, if their angle is 12 degrees, they are very similar, but if it is 180 degrees, they are opposite and thus completely dissimilar. With an arbitrary number of qualitative directions, it can be a little bit harder. In this thesis, we propose generating a neighborhood graph with the weight on the arc. That neighborhood graph will allow separating the calculus from the clustering algorithm and thus, if the number of qualitative direction in a dataset is different, we only require using a different neighborhood graph. The similarity can be generated by dividing 100 by half the number of qualitative directions. The Figure 7.5 shows the graph we used:

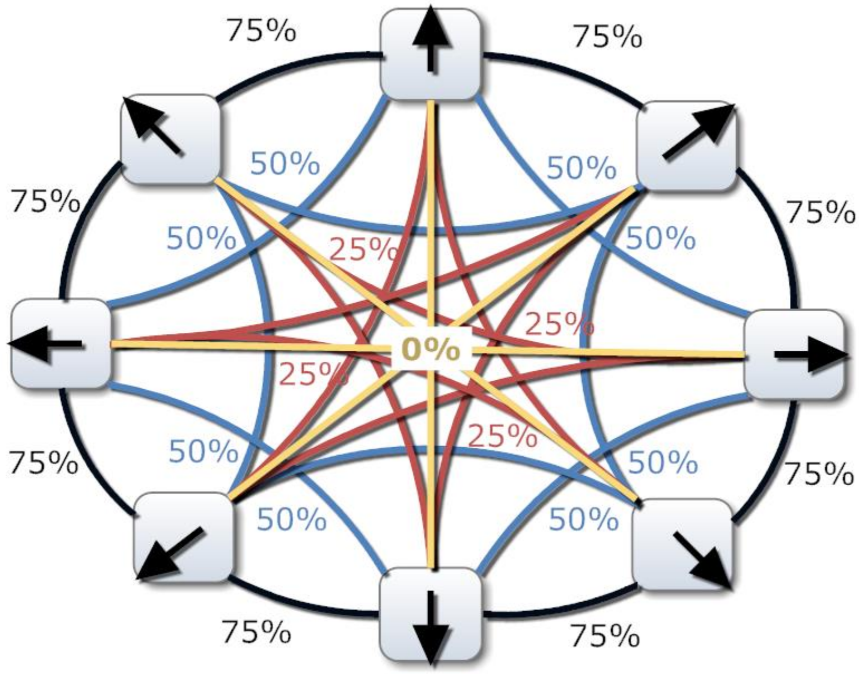
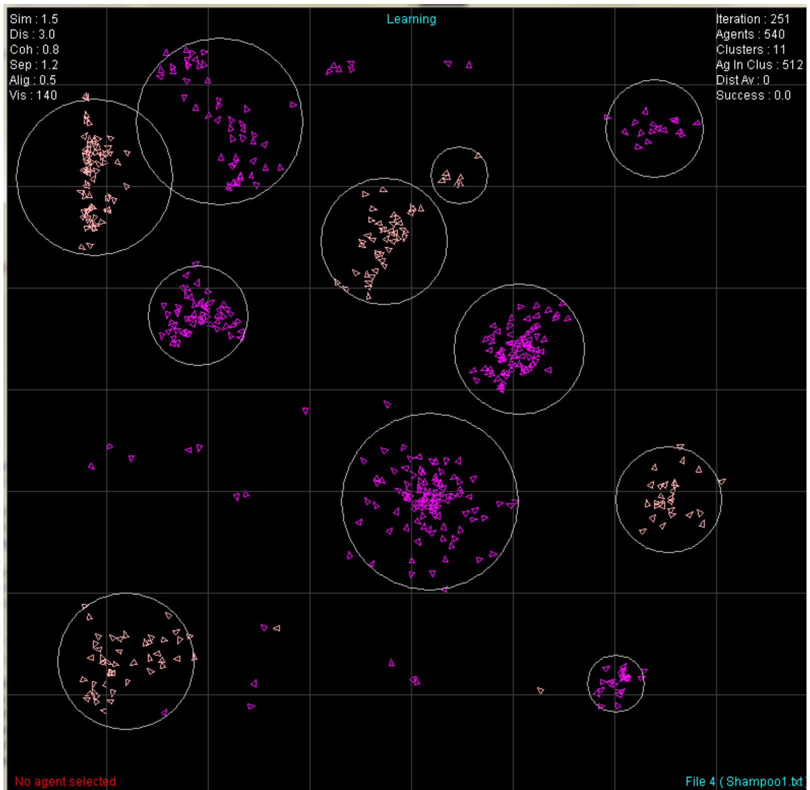


Figure 7.5: The neighborhood graph of the qualitative directions.

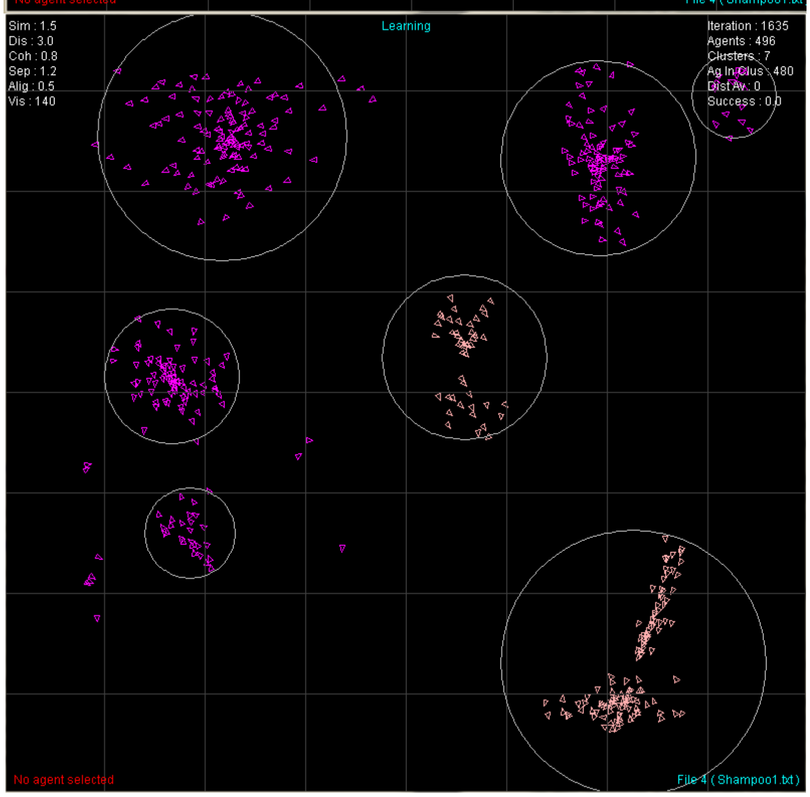
7.3.2.2 Experimental setup

For the experimental phase, again the dataset was divided 2/3 to learn and create the clusters and the other 1/3 for testing these clusters. Again at the beginning of the learning phase only one big cluster was created. However, this time, since the number of agents was much higher, they spread across the environment very fast creating eleven clusters in few hundred iterations as shown on the Figure 7.6 (a). The Figure 7.6 (b)(c) depict the rapid evolution at the iteration 1635 and 5587. As it can be seen, in less than six thousands iterations, the number of clusters went from eleven to seven and finally to five. Our algorithm was again able to find the correct number of ADLs in the dataset.

(a)


Sim : 1.5
Dis : 3.0
Coh : 0.8
Sep : 1.2
Alig : 0.5
Vis : 140
Learning
Iteration : 251
Agents : 540
Clusters : 11
Ag In Clus : 512
Dist Av : 0
Success : 0.0
No agent selected
File 4 (Shampoo1.txt)

(b)


Sim : 1.5
Dis : 3.0
Coh : 0.8
Sep : 1.2
Alig : 0.5
Vis : 140
Learning
Iteration : 1635
Agents : 496
Clusters : 7
Ag In Clus : 480
Dist Av : 0
Success : 0.0
No agent selected
File 4 (Shampoo1.txt)

(c)

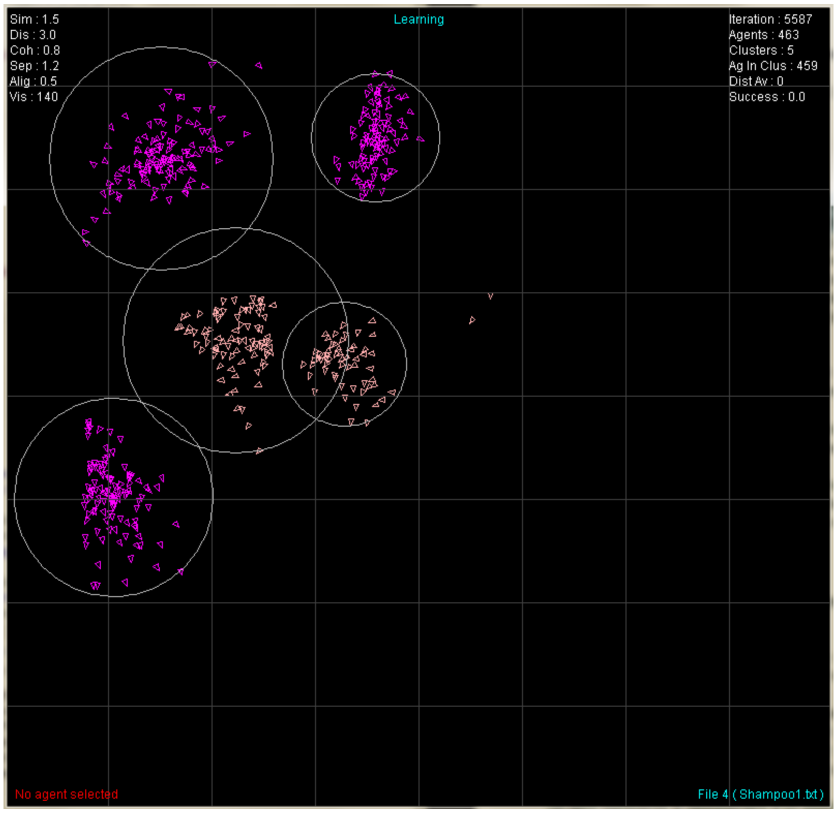


Figure 7.6: Progression of the learning phase.

Following the learning phase, we launched the algorithm in testing phase with the remaining data. The Table 7.6 shows a part of our results at certain iteration. The first thing to observe is that despite the significantly higher recognition challenge, the recognition accuracy evolved really similarly with the first set of experiments. We achieved a 86.73% of good classification at the iteration 11891 which is very good considering the amount of noise in the dataset. We let the algorithm run for more time after, but the classification did not improve very much more. Obviously, this dataset contained more useless information than the previous one which contained only the events generated by simple binary sensors such as the motion sensors or the electromagnetic contacts. In that condition, it is not surprising to see a good classification percentage a little bit more than five points lower.

Table 7.6: The results of Flocking clustering algorithm on real movement based datasets.

Iterations	*1877*	*3867*	*6244*	*6768*	*9092*	*9331*	*11891*
Success	36.8%	68.9%	80%	79.4%	85.3%	85.7%	86.7%

7.3.3 Assessment of the Flocking

Our Flocking based clustering method has proven to be effective in various kinds of dataset. It reposes on solid foundations and could be certainly exploited for unsupervised learning of ADLs in the future. The Flocking algorithm seems to perform better than the classical methods such as K-Means or Expectation Maximization as shown by the first set of experiments. The main advantages of this algorithm, in addition of being an unsupervised method are its complexity and the emergence that appears from the simple interaction between the boids. Indeed, as we have shown, with a simple method like cell-space partitioning, the Flocking complexity is linear, which is important in a context of exponentially growing data warehouse. The second one, the emergence, helps in finding automatically the number of clusters, which is generally required in clustering.

Nevertheless, the Flocking clustering algorithm possesses its own limitations. The first disadvantage is the basic time required for clusters to emerge. As we have seen, whatever the size of the dataset, around 5000-6000 iterations minimum are required to get a good clustering. That base time made it slower than K-Means or even EM when confronted to a reasonable amount of data. The second drawback of the method is with the weight associated to each computed force. Indeed, these weight values can be hard to optimize and tweak, especially if there is no supervised training data available to do so. Finally, the rendering of the boids is CPU hungry. While it is only optional to be able to visualize the boids evolving, it is very helpful for a human user doing experiments.

7.4 CHAPTER CONCLUSION

In this chapter, we have finally laid the last piece on our spatial data mining method that is able to exploit high-level qualitative movement information in order to extract in a fully unsupervised fashion the models of activities. To do so, we have developed an extension to the Flocking, an emergent behavior algorithm, in order to transform it into a clustering method. This was achieved by designing two new rules that enable agents, seemingly moving randomly, to follow the similar agent and avoid those that are dissimilar. This new algorithm was first tested with classical events based dataset extracted from high-level ADLs such as what is found in the activity recognition literature. We have shown in that case that it performs better than K-Means and Expectation Maximization while not suffering from the problem of requirement of the specification of the number of clusters to find. We also conducted a second set of experiments which this time was exploiting the complete process of spatial data mining described in this thesis. Finally, while this second set resulted in a decrease of the classification success rate, the problem was much harder and the dataset was also bigger.

PART IV

CONCLUSION AND APPENDIX

CHAPTER 8

GENERAL CONCLUSION

This thesis research project presented in the seven previous chapters has proposed original solutions to the challenges arising from the new context of computer science and technology. This context, which comes from the intersection of ubiquitous computing and mobility, is one where the data grow exponentially and are underexploited due to the lack of adequate tools. The research on assistive smart home is one of the disciplines, which would greatly benefit from a better exploitation of the data. As we have seen in Chapter 4, these environments are equipped with a large number of sensors that generate Big Data warehouse. Several research teams have begun to turn to data mining methods to tackle this problem. As discussed at the beginning, one of the major challenges of research on smart homes is the recognition of activities of daily living. Most of the current approaches suppose that a human expert can create from scratch an elaborated library containing the possible ADLs. Moreover, the literature gives very little space to the spatial aspect, which is, nevertheless, fundamental in the realization of activities.

To address these problems and to the limits of the previous work, we have proposed, through this thesis a complete spatial data mining solution. This model takes the raw input of sensors in the smart environments to shape a data warehouse and transform it in high level spatial features that enable the mining technique to learn models of activities of daily living. In summary, the general objective of this project was to explore the exploitation of a spatial aspect in conjunction with data mining technique in order to tackle the difficult challenge of extracting interesting patterns in a Big Data warehouse. This final chapter will cast a light on the overall project by reviewing the objectives, the models and the potential perspective for the future.

8.1. REALIZATION OF THE OBJECTIVES

The first phase of this project aimed at investigating in depth the context of research. In a first time, the smart home technologies and the context of research was explored. This has allowed us to better understand the issues and special features to this domain. In the second part, we explored the classical artificial intelligence approaches to the challenge of activity recognition. By doing so, we could assess the main logical [68, 94, 114, 115] and probabilistic [46, 113, 118, 120] approaches. We also reviewed model that integrated the spatial aspect [72, 125], since it was a potential direction for this project. The results of this investigation have allowed us to glimpse the strengths and limitations of conventional models of activity recognition in order to understand the interest of data mining approaches. In particular, we came to the conclusion that the main problem linking these models was the need for a human expert to create the library of plans, which is a difficult and very long task. The third part of this phase consisted in reviewing the literature on data mining. To do so, we explored the main data mining algorithms: decision tree [17, 130], association rules [135, 136] and clustering [109]. We also described briefly some models of activity recognition exploiting these algorithms [108, 131]. We concluded by surveying the advances in the field of Geographical Information System (GIS) which actively develop new algorithms to exploit various spatial aspects. In particular, we described a very interesting algorithm named Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [83].

The second phase of this project has, in the light of previous investigations, to set forward our research hypotheses to define a model of spatial data mining meeting the identified needs. We decided to focus on one spatial aspect that takes a significant place in human activity: the movement. From that point, we designed a complete data mining solution that consisted of three steps: 1-Collecting the data, 2-Preparing the data and 3-Proceeding to data mining. Once we had a general idea of the complete spatial data mining model, we designed adapted solutions for each part. At the collection step, we designed a localization algorithm able to exploit noisy Received Signal Strength Indication (RSSI) of passive RFID tags in order to approximate, the position of objects and track them in real time. At the preparation step, we turned the data warehouse of positions into high-level spatial information with an algorithm that we created for the purpose of gesture recognition and direction segmentation (AtomGID). At the step of data mining, we designed a clustering algorithm based on the Flocking in order to exploit the emergence concept with the movement based dataset.

The last phase of this project consisted in validating the model developed in a concrete and realistic experimental environment. To this end, we have exploited the cutting-edge smart home infrastructure of the LIARA laboratory which was designed to reproduce a real living environment. This infrastructure integrates a wide range of simple and complex sensing technologies that provided us with all the information required. Within this infrastructure, we conducted many experiments that thoroughly tested each composite of our new model. Among other things, this final phase of the research project allowed us to identify the strengths and weaknesses of each part and of the overall spatial data mining solution. In addition, we were able to pinpoint interesting improvement for the future that we will discuss later in this chapter.

It should be noted that in conjunction with this project, we also worked on a smart range prototype that can assist and help a user in the completion of his recipes. The smart range is equipped with four load cells for which an algorithm analyzes the signal and estimates the weight and positions of objects on the stove or in the oven. The prototype can also detect fire hazard and cut the power at any moment. We also conducted many experiments on this prototype in parallel with this research project, but since it is off topic, we will refer the reader to [93] for more information. Finally, we recently received a provisional patent on which I am a co inventor with Pr. Bruno Bouchard, Ph.D. and Pr. Abdenour Bouzouane, Ph.D for that prototype. The smart range was also well received by the scientific community as it is shown by the acceptance in the *Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence* (IAAI-14) organized by the prestigious *Association for the Advancement of Artificial Intelligence* (AAAI).

8.2. REVIEW OF THE DEVELOPED MODEL

The model of spatial data mining described in this thesis proposes several interesting innovations in relation to the scientific literature. First, it is a model completely unsupervised, that is to say, that the training data do not need to be classified in advance thanks to the expertise of a human. Very few approaches of this type have been previously proposed, and existing ones are usually very simple recognizing only grossly defined activities. For example, we previously mentioned the approach of Palmes et al. [63] which proposes to explore the web an unsupervised way to create activity models. However, roughly, their approach only allows to define an activity by a single key object posing a fundamental limit that our model does not suffer. Moreover, contrary to other clustering based algorithm, our new model does not require the user to know the number of clusters to create in advance and has a linear complexity.

The second contribution of our model is the integration of the fundamental spatial aspect for the recognition of ADLs. As we have seen in the second part of this thesis, very few models consider the spatial aspect differently than other data and thus a lot of expressivity is often lost. Yet, when human try to perform the task of activity recognition, they will mostly use the different spatial aspects that they observe to achieve it. For example, if a human observes that the actor is moving the coffee and a cup on the kitchen counter, he will instantaneously infer the correct activity. In that context, it is therefore only natural to try to better exploit spatial aspects in activity recognition. Our previous work integrated the topological relations and for this thesis, we focused on the movement aspect.

Another important contribution is the limitation of the data growth. As we have stated in the beginning, in the new context of Big Data, new solution must be found to perform the task of data mining since the classical algorithm cannot process huge amount of data. There are many avenues of solutions currently explored by the researchers and one of them is the data aggregation. Our model partially addresses this issue. Indeed, it can transform large data warehouses of positions into small dataset of high level qualitative direction. As we demonstrated, these directions can then be exploited into a clustering algorithm.

Finally, this new spatial data mining was conceived in a way that each part are fully autonomous and independent. The main advantage is that each algorithm developed can be exploited within a different applicative context. In addition, each of the developed algorithms were generalized so they could be used with different technology, precision and type of data.

8.2.1 KNOWN LIMITATIONS OF THE PROPOSED MODEL

Despite the success of the model proposed in this thesis, we believe that the research on spatial data mining, especially for smart homes, in the context of Big Data will require many more years of research. In particular, our model addresses the growth of the data with an aggregation solution, but it is often undesirable. Aggregation of data carries the risk of losing important information, and thus many researchers prefer to work on the complete Big Data warehouse. We did not have the problem of losing information because we were well aware of the properties of our dataset. Indeed, the positions are, for the most part, on inactive objects and give less information than our qualitative movement based model.

Another drawback of our model is that the performance decreased in more realistic usage context (busier environment). Even if it did not reflect on the final experiments with the flocking algorithm, both the localization and the gesture recognition algorithm performed less well than when they were tested as standalone algorithms. Particularly, the localization algorithm often had big spikes of imprecision that were probably due to human interference. The precision also decreased during activity realization if too many objects were grouped together. On the side of the gesture recognition algorithm, what we did not expected is the problems coming from spatial jumps that sometime happen for an idle object. Indeed, sometime an idle object jumps directly on another position (without transition points), and if it remains there for many iterations a gesture is inferred.

The final limitation of our model comes from the Flocking algorithm. While it provided us with good results, we expected the movement information to have a bigger impact on the recognition success. In our opinion, the Flocking model designed during this thesis project does not fully exploit the potential of the movement information. More research should be done on that aspect in the future in order to design specific algorithms that palliate to this issue.

8.3. PROSPECTS AND FUTURE WORK

Although the spatial data mining model developed possesses its drawback, we are very optimistic about the future. This thesis project has laid a foundation on an emerging field of research that should provide a lot of challenges to the community for many years to come. In this section, we discuss the future work on spatial data mining.

8.3.1 EXPERIMENTATIONS IN DIFFERENT CONTEXTS

The first short-term development outlook would be to envisage application of the new model or parts of it in a different context of use. Indeed, it would be interesting to test the complete spatial data mining process within an unknown applicative context. One of them could be the monitoring of car traffic as in the work of Liu et al. [95]. Also, it could be interesting to exploit the localization algorithm in different smart home, with different hardware configuration to see how it scales in real-life context. The same could go for the gesture recognition algorithm. We designed the algorithm so it could handle varying degree of precision and adjust automatically. The algorithm should also support a varying number of basic directions. It would then be interesting to test it within different configurations and compare the results.

8.3.2 EXPLOITATION OF OTHER SPATIAL ASPECTS

Another interesting path of improvement would be to explore the other features of the spatial aspect. For example, in our previous work, we successfully exploited the topological relationships to design a probabilistic activity recognition algorithm. It would be interesting to explore the possibility of including both the movement information and the topological relationship in order to perform spatial data mining. Other spatial aspects that we did not discuss in this thesis could also be interesting: shape, distance, orientation, etc.

8.3.3 Long term prospects

Over the long term, this type of approaches could enable the smart home assistance to be deployed on a large scale. As we said in the introduction, this technology could not only ease the life of elder and prolong autonomous care, but also address partially the challenges related to healthcare systems such as the rising cost and the professional shortage. Another interesting application of data mining method for healthcare is on the side of Business Intelligence. By learning the profile of a patient, it could be possible in the future to develop applications that could help the physicians to monitor the status of his patients over his smart phone or his PC. Such application could prove particularly useful when dealing with persons suffering from dementia. In their case, they often are unable to provide good-quality information to their physician due to their impairment. Finally, spatial data mining could find many other uses outside the context of smart home. For example, it could be a good avenue of research for *smart cities*, which is an emerging topic of research. Smart cities could not only provide the resident with useful service, but with spatial data mining, the resources such as transport vehicles or electricity could be optimized by efficiently distributing them.

8.4. Personal assessment on this research

In conclusion, I would like to use few last words to do a brief personal assessment of my initiation to the world of research. The journey made throughout this project was quite a hard and constant work. However, it was very rewarding, worthy of all these short nights for which I traded hours of sleep for acquisition new precious knowledge in the targeted area of expertise of spatial data mining in the context of smart home for activity recognition. I was able to successfully conduct this project because of its stimulating nature. As a member of a formidable multidisciplinary team, I have been lucky enough to participate in multiple projects and activities with peer from different fields. This experience allowed me to develop important new skills such as a rigorous research methodology and communication skills. This rewarding experience also allowed me to make few contributions to the scientific community in my field of research that I presented at the occasion of notorious international conferences [93, 104-107, 189] and journals [101, 102]. After such a positive introduction to research, I only look toward beginning a career as a researcher and pushing the limit of science in new territories. My last words go to all the persons that supported me, one way or another, intentionally or not, in my quest to obtain an expertise, new skills set and priceless knowledge.

APPENDIX A

GENERATED REPORT EXAMPLE

In section 6.5, we discussed the gesture recognition software that was developed in order to test our various algorithms through this research project. One interesting aspec of this software is the ability to simulate RFID localization with a random amount of noise. When provided with a dictionary of gestures, the software can test the gesture recognition algorithm by itself and generate a simple report containing the results. Here is an example of such a report for 13 possible gestures:

	Front	LeftFr	Back	BackRi	RightF	RightB	Noise	BackLe	LeftBa	FrontR	FrontL	Right	Left
Front	6	0	0	0	0	0	3	0	0	0	0	0	0
LeftFront	0	17	0	0	0	0	0	0	0	0	0	0	0
Back	0	0	5	0	0	0	0	0	0	0	0	0	0
BackRight	0	0	0	5	0	1	0	0	0	0	0	0	0
RightFront	0	0	0	0	6	0	0	0	0	0	0	0	0
RightBack	0	0	0	0	0	3	0	0	0	0	0	0	0
Noise	0	0	0	0	0	0	19	0	0	0	0	0	0
BackLeft	0	0	0	0	0	0	0	6	1	0	0	0	0
LeftBack	0	0	0	0	0	0	0	0	3	0	0	0	0
FrontRight	0	0	0	0	1	0	0	0	0	2	0	0	0
FrontLeft	0	0	0	0	0	0	0	0	0	0	7	0	0
Right	0	0	0	0	0	0	1	0	0	0	0	1	0
Left	0	0	0	0	0	0	0	0	0	0	0	0	2

	TP	FN	TN	FP
Front	6	2	80	0
LeftFront	17	0	71	0
Back	5	0	83	0
BackRight	5	1	82	0
RightFront	6	0	81	1
RightBack	3	0	84	1
Noise	19	0	66	3
BackLeft	6	1	81	0
LeftBack	3	0	84	1
FrontRight	2	1	85	0
FrontLeft	7	0	81	0
Right	1	1	86	0
Left	2	0	86	0

Overall: 93 % (82/88)

APPENDIX B

CARTESIAN POSITION OF THE MAIN FIXED SENSORS

In this appendix, we present a table describing the sensors exploited for the experiments with the Flocking and their fixed positions.

Name	*Room*	*X*	*Y*	*Description*
CA1	Kitchen	400	285	Electromagnetic contact of cabinet
CA2	Kitchen	400	265	Electromagnetic contact of cabinet
DB1	Kitchen	400	290	Flowmeter
DB2	Kitchen	400	285	Flowmeter
CB1	Kitchen	390	300	Electromagnetic contact of cabinet
CB2	Kitchen	390	200	Electromagnetic contact of cabinet
CB3	Kitchen	390	280	Electromagnetic contact of cabinet
CB4	Kitchen	390	260	Electromagnetic contact of cabinet
CA3	Kitchen	400	190	Electromagnetic contact of cabinet
CA4	Kitchen	400	170	Electromagnetic contact of cabinet
CA5	Kitchen	300	100	Electromagnetic contact of cabinet
CA6	Kitchen	250	100	Electromagnetic contact of cabinet
CB6	Kitchen	270	100	Electromagnetic contact of cabinet
LTC	Kitchen	365	130	Range hood light effector
LT1	Kitchen	365	130	Range hood light sensor
CB7	Kitchen	250	250	Electromagnetic contact of oven door
CB8	Kitchen	250	250	Electromagnetic contact of range drawer
MCR_i	Kitchen	250	250	Electrical power of the range
LTF	Kitchen	250	250	Light in the oven
TC1	Kitchen	250	250	Temperature sensor front left hub
TC2	Kitchen	250	250	Temperature sensor back left hub
TC3	Kitchen	250	250	Temperature sensor central hub

TC4	Kitchen	250	250	Temperature sensor front right hub
TC5	Kitchen	250	250	Temperature sensor back right hub
TC6	Kitchen	250	250	Temperature sensor of the oven
RL13	Kitchen	250	250	Emergency shut down
MV2	Bedroom	450	450	Motion sensor, bedroom
MV3	Bedroom	450	450	Motion sensor, bathroom
MV4	Bedroom	450	450	Motion sensor, entrance hall
CD1	Bedroom	450	450	Motion sensor, shelf 1
CD2	Bedroom	450	450	Motion sensor, shelf 2
CD3	Bedroom	450	450	Motion sensor, shelf 3
CD4	Bedroom	450	450	Motion sensor, shelf 4
TP1	Bedroom	450	450	Tactile mat, bedroom
TP2	Bathroom	50	50	Tactile mat, bathroom
CC5	Bathroom	50	50	Electromagnetic contact of cabinet
DB3	Bathroom	50	50	Flow meter, hot water of the bath
DB4	Bathroom	50	50	Flow meter, cold water of the bath
DB5	Bathroom	50	50	Flow meter, hot water of the sink
DB6	Bathroom	50	50	Flow meter, cold water of the sink
DB7	Bathroom	50	50	Flow meter, toilet
CE1	Living room	150	150	Electromagnetic contact, entrance door
MV5	Living room	150	150	Motion sensor, living room
CE2	Living room	150	150	Electromagnetic contact, door
CA10	Kitchen	250	250	Electromagnetic contact, freezer
CA11	Kitchen	250	250	Electromagnetic contact, refrigerator
LD1	Living room	150	150	Light sensor, entrance
LD2	Living room	150	150	Light sensor, table
LD3	Kitchen	250	250	Light sensor, kitchen
LD4	Bedroom	450	450	Light sensor, bedroom
LD5	Bathroom	50	50	Light sensor, bathroom
TC8	Kitchen	250	250	Temperature sensor, sink
TC9	Living room	150	150	Temperature sensor, living room
TC10	Bedroom	450	450	Temperature sensor, bedroom
TC11	Bathroom	50	50	Temperature sensor, bathroom
TC12	Bathroom	50	50	Temperature sensor, sink
TC13	Bathroom	50	50	Temperature sensor, bath

BIBLIOGRAPHY

[1] M. J. Wooldridge, *An introduction to multiagent systems*. Chichester, England: Wiley; 2nd edition (July 7, 2009), 2009.

[2] G. Weiss, *Multiagent Systems*: MIT Press, 2000.

[3] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*: Prentice Hall, 2010.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks,* vol. 54, pp. 2787-2805, 2010.

[5] M. Weiser, "The computer for the 21st century," *Scientific American,* vol. 265, pp. 66-75, 1991.

[6] C. Ramos, J. C. Augusto, and D. Shapiro, "Ambient Intelligence: the Next Step for Artificial Intelligence," *IEEE Intelligent Systems,* vol. 23, pp. 15-18, 2008.

[7] F. Frankel and R. Reid, "Big data: Distilling meaning from data," *Nature,* vol. 455, pp. 30-30, 2008.

[8] C. Lynch, "Big data: How do your data grow?," *Nature,* vol. 455, pp. 28-29, 2008.

[9] S. Chaudhuri, U. Dayal, and V. Narasayya, "An overview of business intelligence technology," *Commun. ACM,* vol. 54, pp. 88-98, 2011.

[10] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao, "Data mining techniques and applications – A decade review from 2000 to 2011," *Expert Systems with Applications,* vol. 39, pp. 11303-11311, 2012.

[11] W. B. Heinzelman, A. L. Murphy, H. S. Carvalho, and M. A. Perillo, "Middleware to support sensor network applications," *Network, IEEE,* vol. 18, pp. 6-14, 2004.

[12] C. Ordonez, "Can we analyze big data inside a DBMS?," in *Proceedings of the sixteenth international workshop on Data warehousing and OLAP*, 2013, pp. 85-92.

[13] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Commun. ACM,* vol. 51, pp. 107-113, 2008.

[14] B. Alexandru, "MongoDB vs Oracle -- Database Comparison," 2012, pp. 330-335.

[15] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, 2010, pp. 1-10.

[16] Ian H. Witten and E. Franck, "Data mining: practical machine learning tools and techniques," *Elsevier editor*, 2010.

[17] J. R. Quinlan and J. Ghosh, "Top 10 Algorithms in Data Mining," in *International Conference on Data Mining (ICDM'06)*, Hong Kong, 2006.

[18] H. B. Barlow, "Unsupervised learning," *Neural computation*, vol. 1, pp. 295-311, 1989.

[19] J. Carbonell, R. Michalski, and T. Mitchell, "An Overview of Machine Learning," in *Machine Learning*, R. Michalski, J. Carbonell, and T. Mitchell, Eds., ed: Springer Berlin Heidelberg, 1983, pp. 3-23.

[20] V. Jakkula and D. J. Cook, "Mining Sensor Data in Smart Environment for Temporal Activity Prediction," in *KDD'07*, San Jose, California, USA, 2007.

[21] J. Li, Y. Liu, W.-k. Liao, and A. Choudhary, "Parallel data mining algorithms for association rules and clustering," in *International Conference on Management of Data*, 2008.

[22] U. Nations, *World population ageing 2009*: United Nations, Dept. of Economic and Social Affairs, Population Division, 2010.

[23] J. Diamond, "A report on Alzheimer disease and current research.," *Alzheimer Society of Canada*, pp. 1-26, 2006.

[24] D. Patterson, O. Etzioni, D. Fox, and H. Kautz, "Intelligent ubiquitous computing to support Alzheimer's patients: Enabling the cognitively disabled," in *In UbiCog '02: First International Workshop on Ubiquitous Computing for Cognitive Aids*, ed, 2002.

[25] C. Nugent, M. Mulvenna, F. Moelaert, B. Bergvall-Kareborn, F. Meiland, D. Craig, R. Davies, A. Reinersmann, M. Hettinga, A.-L. Andersson, R.-M. Droes, and J. E. Bengtsson, "Home based assistive technologies for people with mild dementia," presented at the Proceedings of the 5th international conference on Smart homes and health telematics, Nara, Japan, 2007.

[26] M. Van Tassel, J. Bouchard, B. Bouchard, and A. Bouzouane, *Guidelines for Increasing Prompt Efficiency in Smart Homes According to the Resident's Profile and Task Characteristics* vol. 6719: Springer, 2011.

[27] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, and A. Mihailidis, "A decision-theoretic approach to task assistance for persons with dementia," presented at the Proceedings of the 19th international joint conference on Artificial intelligence, Edinburgh, Scotland, 2005.

[28] S. Carberry, "Techniques for Plan Recognition," *User Modeling and User-Adapted Interaction*, vol. 11, pp. 31-48, 2001.

[29] K. Z. Haigh, L. M. Kiff, J. Myers, V. Guralnik, C. W. Geib, J. Phelps, and T. Wagner, "The independent lifestyle assistant\&\#8482; (I.L.S.A.): AI lessons learned,"

presented at the Proceedings of the 16th conference on Innovative applications of artifical intelligence, San Jose, California, 2004.

[30] D. Patterson, H. Kautz, and D. Fox, "Pervasive computing in the home and community," in *Pervasive Computing in Healthcare*, ed: CRC Press, 2007, pp. 79-103.

[31] J. Hoey, P. Poupart, A. v. Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis, "Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process," *Computer Vision and Image Understanding*, vol. 114, pp. 503-519, 2010.

[32] B. Bouchard, S. Giroux, and A. Bouzouane, "A keyhole plan recognition model for Alzheimer's patients: First results," *Journal of Applied Artificial Intelligence*, vol. 21, pp. 623-658, 2007.

[33] P. Roy, B. Bouchard, A. Bouzouane, and S. Giroux, "A hybrid plan recognition model for Alzheimer's patients: Interleaved-erroneous dilemma," *Web Intelli. and Agent Sys.*, vol. 7, pp. 375-397, 2009.

[34] S. Katz, A. B. Ford, R. W. Moskowitz, B. A. Jackson, and M. W. Jaffe, "Studies of Illness in the Aged," *JAMA: The Journal of the American Medical Association*, vol. 185, pp. 914-919, September 21, 1963 1963.

[35] T. Giovannetti, D. J. Libon, L. J. Buxbaum, and M. F. Schwartz, "Naturalistic action impairments in dementia," *Neuropsychologia*, vol. 40, pp. 1220-1232, 2002.

[36] C. Baum and D. F. Edwards, *Cognitive performance in senile dementia of the Alzheimer's type: the Kitchen Task Assessment* vol. 47, 1993.

[37] M. F. Schwartz, M. Segal, T. Veramonti, M. Ferraro, and L. J. Buxbaum, *The Naturalistic Action Test: A standardised assessment for everyday action impairment* vol. 12. Hove, ROYAUME-UNI: Psychology Press, 2002.

[38] S. Giroux, T. Leblanc, A. Bouzouane, B. Bouchard, H. Pigot, and J. Bauchet, "The Praxis of Cognitive Assistance in Smart Homes," in *Behaviour Monitoring and Interpretation*. vol. 3, B. Gottfried and H. K. Aghajan, Eds., ed BMI Book: IOS Press, 2009, pp. 183-211.

[39] J. Modayil, T. Bai, and H. Kautz, "Improving the recognition of interleaved activities," presented at the Proceedings of the 10th international conference on Ubiquitous computing, Seoul, Korea, 2008.

[40] C. F. Schmidt, N. S. Sridharan, and J. L. Goodson, "The plan recognition problem: An intersection of psychology and artificial intelligence," *Artificial Intelligence*, vol. 11, pp. 45-83, 1978.

[41] Y. Wærn and R. Ramberg, "People's perception of human and computer advice," *Computers in Human Behavior*, vol. 12, pp. 17-27, 1996.

[42] M. P. Georgeff, Lansky, A. L., "Reactive Reasoning and Planning," in *AAAI-87 Proceedings*, 1987, pp. 677-682.

[43] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-Grained Activity Recognition by Aggregating Abstract Object Usage," presented at the Proceedings of the Ninth IEEE International Symposium on Wearable Computers, 2005.

[44] J. Boger, J. Hoey, P. Poupart, C. Boutilier, G. Fernie, and A. Mihailidis, *A planning system based on Markov decision processes to guide people with dementia through activities of daily living* vol. 10: IEEE, 2006.

[45] R. Goldman, C. Geib, and C. Miller, "A New Model of Plan Recognition," *Artificial Intelligence*, vol. 64, pp. 53-79, 1999.

[46] D. J. Patterson, L. A. Lin, D. Fox, and H. Kautz, "Inferring high-level behavior from low-level sensors," in *Ubicomp 2003: Ubiquitous Computing*. vol. 2864, A. K. Dey, A. Schmidt, and J. F. McCarthy, Eds., ed Berlin: Springer-Verlag Berlin, 2003, pp. 73-89.

[47] H. Kautz and J. Allen, "Generalized plan recognition," in *National Conference on Artificial Intelligence (AAAI)*, 1986, pp. 32-37.

[48] B. Bouchard, "Un modèle de reconnaissance de plan pour les personnes atteintes de la maladie d'Alzheimer basé sur la théorie des treillis et sur un modèle d'action en logique de description," Ph.D., Université de Sherbrooke, Sherbrooke, 2006.

[49] L. Chen, C. D. Nugent, and H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes," *IEEE Trans. on Knowl. and Data Eng.,* vol. 24, pp. 961-974, 2012.

[50] D. W. Albrecht, I. Zukerman, and A. E. Nicholson, "Bayesian Models for Keyhole Plan Recognition in an Adventure Game," *User Modeling and User-Adapted Interaction,* vol. 8, pp. 5-47, 1998.

[51] V. R. Jakkula and D. J. Cook, "Using temporal relations in smart home data for activity prediction," in *Proc. of the ICML Workshop on the Induction of Process Models*, 2007.

[52] K. Bouchard, B. Bouchard, and A. Bouzouane, "Qualitative Spatial Activity Recognition Using a Complete Platform Based on Passive RFID Tags: Experimentations and Results," presented at the 9th International Conference On Smart homes and health Telematics, Montreal, 2011.

[53] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities : A Survey," *IEEE Transactions on Circuits and Systems for Video Technology*, p. 16, 2008.

[54] M. C. Huebscher and J. A. McCann, "Adaptive middleware for context-aware applications in smart-homes," in *Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, 2004, pp. 111-116.

[55] C. Gouin-Vallerand, B. Abdulrazak, S. Giroux, and M. Mokhtari, "A self-configuration middleware for smart spaces," *self*, vol. 3, 2009.

[56] L. Coyle, S. Neely, G. Stevenson, M. Sullivan, S. Dobson, and P. Nixon, "Sensor Fusion-Based Middleware for Smart Homes," *International Journal of Assistive Robotics and Mechatronics (IJARM)*, vol. 8, pp. 53-60, 2007.

[57] A. Kamilaris, A. Pitsillides, and V. Trifa, "The smart home meets the web of things," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 7, pp. 145-154, 2011.

[58] V. Ricquebourg, D. Durand, L. Delahoche, D. Menga, B. Marhic, and C. Loge, "The Smart Home Concept : our immediate future," in *1ST IEEE International Conference on E-Learning in Industrial Electronics*, 2006, pp. 23-28.

[59] A. Komninos and S. Stamou, "HealthPal: an intelligent personal medical assistant for supporting the self-monitoring of healthcare in the ageing society," *Proceedings of the UbiHealth*, 2006.

[60] T. Gu, L. Wang, Z. Wu, X. Tao, and J. Lu, "A Pattern Mining Approach to Sensor-Based Human Activity Recognition," *IEEE Trans. on Knowl. and Data Eng.*, vol. 23, pp. 1359-1372, 2011.

[61] E. Kim, S. Helal, and D. Cook, "Human Activity Recognition and Pattern Discovery," *Ieee Pervasive Computing*, vol. 9, pp. 48-53, 2010.

[62] T. v. Kasteren, A. Noulas, G. Englebienne, and B. Krose, "Accurate activity recognition in a home setting," presented at the Proceedings of the 10th international conference on Ubiquitous computing, Seoul, Korea, 2008.

[63] P. Palmes, H. K. Pung, T. Gu, W. Xue, and S. Chen, "Object relevance weight pattern mining for activity recognition and segmentation," *Pervasive Mob. Comput.*, vol. 6, pp. 43-57, 2010.

[64] D. Wyatt, M. Philipose, and T. Choudhury, "Unsupervised activity recognition using automatically mined common sense," presented at the Proceedings of the 20th national conference on Artificial intelligence - Volume 1, Pittsburgh, Pennsylvania, 2005.

[65] V. R. Jakkula and D. J. Cook, "Enhancing Smart Home Algorithms Using Temporal Relations," in *Technology and Aging*. vol. 21, A. Mihailidis, J. Boger, H. Kautz, and L. Normie, Eds., ed Amsterdam: IOS Press, 2008, pp. 3-10.

[66] B. Gottfried, H. Guesgen, and S. Hübner, "Spatiotemporal Reasoning for Smart Homes," *Designing Smart Homes*, vol. 4008, pp. 16-34, 2006.

[67] J. C. Augusto and C. Nugent, "The use of temporal reasoning and management of complex events in smart homes," in *Proceedings of ECAI 2004.*, 2004, pp. 778-82.

[68] W. Wobcke, "Two logical theories of plan recognition," *Journal of Logic and Computation*, vol. 12, pp. 371-412, Jun 2002.

[69] S. Saria and S. Mahadevan, "Probabilistic plan recognition in multiagent systems," in *Proceedings of International Conference on AI and Planning Systems*, 2004.

[70] J. F. Allen and P. J. Hayes, "A common-sense theory of time," presented at the Proceedings of the 9th international joint conference on Artificial intelligence - Volume 1, Los Angeles, California, 1985.

[71] V. R. Jakkula, A. S. Crandall, and D. J. Cook, "Knowledge Discovery in Entity Based Smart Environment Resident Data Using Temporal Relation Based Data Mining," presented at the Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, 2007.

[72] J. C. Augusto, J. Liu, P. McCullagh, and H. Wang, "Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a Smart Home," *International Journal of Computational Intelligence Systems* vol. 1, pp. 361 - 378, 2008.

[73] J. C. Augusto and C. D. Nugent, "Smart homes can be smarter," in *Designing Smart Homes: Role of Artificial Intelligence*. vol. 4008, ed Berlin: Springer-Verlag Berlin, 2006, pp. 1-15.

[74] H. Laprise, J. Bouchard, B. Bouchard, and A. Bouzouane, "Creating tools and trial data sets for smart home researchers: experimenting activities of daily living with normal subjects to compare with Alzheimer's patients," in *International Conference IADIS e-Health*, Freiburg, Germany, 2010, pp. 1-8.

[75] D. Lymberopoulos, A. Bamis, and A. Savvides, "Extracting spatiotemporal human activity patterns in assisted living using a home sensor network," presented at the Proceedings of the 1st international conference on PErvasive Technologies Related to Assistive Environments, Athens, Greece, 2008.

[76] K. Bouchard, B. Bouchard, and A. Bouzouane, "Discovery of Topological Relations for Spatial Activity Recognition," in *IEEE Symposium Series on Computational Intelligence*, Singapore, 2013.

[77] K. Bouchard, B. Bouchard, and A. Bouzouane, "Spatial recognition of activities for cognitive assistance: realistic scenarios using clinical data from Alzheimer's patients," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-16, 2013/09/13 2013.

[78] M. J. Egenhofer and R. D. Franzosa, "POINT-SET TOPOLOGICAL SPATIAL RELATIONS," *International Journal of Geographical Information Systems*, vol. 5, pp. 161-174, 1991.

[79] A. G. Cohn, B. Bennett, J. Gooday, and N. M. Gotts, "Qualitative Spatial Representation and Reasoning with the Region Connection Calculus," *Geoinformatica*, vol. 1, pp. 275-316, 1997.

[80] S. Shekhar, P. Zhang, and Y. Huang, *Spatial Data Mining*: Springer US, 2010.

[81] K. Koperski and J. Han, "Discovery of Spatial Association Rules in Geographic Information Databases," presented at the Proc. of the 4th International Symposium on Advances in Spatial Databases, 1995.

[82] M. Ester, H.-P. Kriegel, and J. Sander, "Knowledge Discovery in Spatial Databases," presented at the Proceedings of the 23rd Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence, 1999.

[83] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data Mining and Knowledge Discovery,* vol. 2, pp. 169-194, 1998.

[84] N. T. Nguyen, D. Q. Phung, S. Venkatesh, and H. Bui, "Learning and Detecting Activities from Movement Trajectories Using the Hierarchical Hidden Markov Models," presented at the Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02, 2005.

[85] E. Clementini, P. D. Felice, and D. Hernández, "Qualitative representation of positional information," *Artificial Intelligence,* vol. 95, pp. 317-356, 1997.

[86] W. Navidi, W. S. Murphy Jr, and W. Hereman, "Statistical methods in surveying by trilateration," *Computational statistics & data analysis,* vol. 27, pp. 209-227, 1998.

[87] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance Second Edition*: Massachusetts: Ganga-Jamuna Press, 2006.

[88] S. Brandt, *Data Analysis: Statistical and computational methods for scientists and engineers with 273 illustrations, includes CD-Rom*: Springer, 1999.

[89] J. Toner and Y. Tu, "Flocks, herds, and schools: A quantitative theory of flocking," *Physical Review E,* vol. 58, p. 4828, 1998.

[90] Bellaachi A. and Bari A., "OFLOSCAN : A Flocking-Based Data Mining Algorithm for Detecting Outliers in Cancer Microarrays," 2010.

[91] Cui X., Gao J., and Potok T., "A flocking based algorithm for document clustering analysis," *Journal of Systems Architecture,* vol. Vol. 52, pp. 505-515, 2006.

[92] J. Lapointe, B. Bouchard, J. Bouchard, and A. Bouzouane, "Smart Homes for People with Alzheimer's Disease : Adapting Prompting Strategies to the Patient's Cognitive Profile," presented at the 5th ACM International Conference on Pervasive Technologies Related to Assistive Environments, Crete Island, Greece, 2012.

[93] B. Bouchard, K. Bouchard, and A. Bouzouane, "A Smart Range Helping Cognitively-Impaired Persons Cooking," in *Twenty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence*, Quebec, Canada, 2014.

[94] H. A. Kautz, "A formal theory of plan recognition and its implementation," in *Reasoning about plans*, ed: Morgan Kaufmann Publishers Inc., 1991, pp. 69-124.

[95] S. Liu, Y. Liu, L. M. Ni, J. Fan, and M. Li, "Towards mobility-based clustering," presented at the Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, DC, USA, 2010.

[96] P. Berkhin, "Survey of clustering data mining techniques," *Accrue Software Research Paper*, 2002.

[97] M. J. Egenhofer, *Spherical topological relations* vol. Journal on Data Semantics. Berlin, ALLEMAGNE: Springer, 2005.

[98] S. Cicerone and P. D. Felice, "Cardinal directions between spatial objects: the pairwise-consistency problem," *Inf. Sci. Inf. Comput. Sci.*, vol. 164, pp. 165-188, 2004.

[99] H. W. Guesgen, *Spatial reasoning based on Allen's temporal logic*: International Computer Science Institute, 1989.

[100] D. Fortin-Simard, K. Bouchard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Accurate Passive RFID Localization System for Smart Homes," presented at the 3th IEEE International Conference on Networked Embedded Systems for Every Application, Liverpool, UK, 2012.

[101] K. Bouchard, D. Fortin-Simard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Accurate RFID Trilateration to Learn and Recognize Spatial Activities in Smart Environment," *International Journal of Distributed Sensor Networks*, vol. 2013, p. 16, 2013.

[102] K. Bouchard, D. Fortin-Simard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Accurate Trilateration for Passive RFID Localization in Smart Homes," *International Journal of Wireless Information Networks*, vol. 21, pp. 32-47, 2014/03/01 2014.

[103] K. Bouchard, J.-S. Bilodeau, D. Fortin-Simard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Exploiting Passive RFID Technology for Indoor Localization in Smart Environments," in *Opportunistic networking, smart home, smart city, smart systems*, S. M. S. Z. Nazmul H. Siddique, Syed Faraz Hasan, Ed., ed: CRC press, Taylor & Francis, 2014, pp. 1-20.

[104] K. Bouchard, B. Bouchard, and A. Bouzouane, "Gesture Recognition in Smart Home Using Passive RFID Technology," in *7th International Conference on PErvasive Technologies Related to Assistive Environments*, Rhodes Island, Greece, 2014.

[105] K. Bouchard, B. Bouchard, and A. Bouzouane, "Regression Analysis for Gesture Recognition Using RFID Technology," in *12th International Conference On Smart homes and health Telematics*, Denver, USA, 2014.

[106] K. Bouchard, S. Gaboury, T. Moutacalli, B. Bouchard, and A. Bouzouane, "Analysis and Knowledge Discovery of Moving Objects Equipped with RFID Tags," in *The AAAI-14 Workshop on artificial intelligence applied to assistive technologies and smart environments (ATSE 2014)*, Quebec, Canada, 2014.

[107] J. Lapalu, K. Bouchard, A. Bouzouane, B. Bouchard, and S. Giroux, "Unsupervised Mining of Activities for Smart Home Prediction," *Procedia Computer Science*, vol. 19, pp. 503-510, 2013.

[108] V. Jakkula and D. J. Cook, "Learning Temporal Relations in Smart Home Data," in *Proceedings of the Second International Conference on Technology and Aging*, Canada, 2007.

[109] S. Z. Selim and M. A. Ismail, "K-means type algorithms: a generalized convergence theorem and characterization of local optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*, pp. 81–87, 1984.

[110] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," 1996.

[111] P. R. Cohen, "The pragmatics of referring and the modality of communication," *Computational Linguistics*, vol. 10, pp. 97-146, 1984.

[112] P. Roy, B. Bouchard, A. Bouzouane, and S. Giroux, "Challenging issues of ambient activity recognition for cognitive assistance," *Ambient Intelligence and Smart Environments*, pp. 1-25, 2010.

[113] N. Nguyen, H. Bui, S. Venkatesh, and G. West, "Recognising and Monitoring High-Level Behaviours in Complex Spatial Environments," in *In IEEE International Conference on Computer Vision and Pattern Recognition (CVPR*, 2003, pp. 620-625.

[114] D. Py, "Reconnaissance de plan pour l'aide à la démonstration dans un tuteur intelligent de la géométrie," Ph. D., Université de Rennes, Rennes, 1992.

[115] Nerzic, "Two methods for recognizing erroneous plans in human-machine dialogue," presented at the Proc. of National Conference on Artificial Intelligence (AAAI '96), 1996.

[116] J. F. Allen, "Maintaining knowledge about temporal intervals," *Artificial Intelligence and Language Processing*, vol. 26, pp. 832-843, 1983.

[117] J. McCarthy, "Circumscription - {A} Form of Non-Monotonic Reasoning," *Artificial Intelligence Journal*, vol. 13, pp. 27-39, 1980.

[118] E. Charniak and R. P. Goldman, "A Bayesian model of plan recognition," *Artificial Intelligence*, vol. 64, pp. 53-79, 1993.

[119] D. Wu and C. Butz, "On the complexity of probabilistic inference in singly connected bayesian networks," presented at the 10th int conf. on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Regina, Canada, 2005.

[120] M. Buettner, R. Prasad, M. Philipose, and D. Wetherall, "Recognizing daily activities with RFID-based sensors," presented at the Proceedings of the 11th international conference on Ubiquitous computing, Orlando, Florida, USA, 2009.

[121] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Readings in speech recognition*, W. Alex and L. Kai-Fu, Eds., ed: Morgan Kaufmann Publishers Inc., 1990, pp. 267-296.

[122] B. Bouchard, P. Roy, A. Bouzouane, S. Giroux, and A. Mihailidis, "An Activity Recognition Model for Alzheimer's Patients: Extension of the COACH Task Guidance System," presented at the ECAI, 2008.

[123] K. Bouchard, J.-S. Bilodeau, D. Fortin-Simard, S. Gaboury, B. Bouchard, and A. Bouzouane, "Human Activity Recognition in Smart Homes Based on Passive RFID Localization," in *7th International Conference on PErvasive Technologies Related to Assistive Environments*, Rhodes Island, Greece, 2014.

[124] J. Yang, J. Liu, J. Wang, H. Sii, and H. Wang, "Belief Rule-Base Inference Methodology Using the Evidential Reasoning Approach - RIMER," *In Proceedings of IEEE Transactions on Systems, Man, and Cybernetic,* pp. 266-285, 2006.

[125] D. E. Riedel, S. Venkatesh, and W. Liu, "Spatial Activity Recognition in a Smart Home Environment using a Chemotactic Model," presented at the International Conference on Intelligent Sensors Networks and Information Processing, Melbourne, Australia, 2005.

[126] J. Adler, "Chemotaxis in Bacteria," *Annual Review of Biochemistry,* vol. 44, pp. 341-356, 1975.

[127] R. Bourret and A. Stock, "Molecular information processing: Lessons from bacterial chemotaxis," *J. Biol. Chem.,* p. R100066200, 2002.

[128] K. Bouchard, B. Bouchard, and A. Bouzouane, "A new qualitative spatial recognition model based on Egenhofer topological approach using C4.5 algorithm: experiment and results," presented at the International Conference on Ambient Systems, Networks and Technologies, Niagara Falls, Canada, 2011.

[129] K. Bouchard, "QUALITATIVE SPATIAL REASONING FOR ACTIVITY RECOGNITION USING TOOLS OF AMBIENT INTELLIGENCE," Master, Université du Québec à Chicoutimi, 2012.

[130] J. R. Quinlan, *C4.5: programs for machine learning*: Morgan Kaufmann Publishers Inc., 1993.

[131] V. Stankovski, "application of decision trees to Smart Homes," in *Designing Smart Homes*, ed: Springer, 2006.

[132] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Pervasive Computing*, ed: Springer, 2004, pp. 1-17.

[133] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the national conference on artificial intelligence*, 2005, p. 1541.

[134] S. R. Gaddam, V. V. Phoha, and K. S. Balagani, "K-means+ id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 19, pp. 345-354, 2007.

[135] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," presented at the Proceedings of the 20th International Conference on Very Large Data Bases, 1994.

[136] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," presented at the Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, 1996.

[137] K. Gopalratnam and D. J. Cook, "Active LeZi: An Incremental Parsing Algorithm for Sequential Prediction," presented at the FLAIRS Conference, 2003.

[138] R. J. Robles and T.-h. Kim, "Applications , Systems and Methods in Smart Home Technology : A Review," *Int. Journal of Advanced Science And Technology,* vol. 15, pp. 37-48, 2010.

[139] K. Bouchard, B. Bouchard, and A. Bouzouane, "Guideline to Efficient Smart Home Design for Rapid AI Prototyping: A Case Study," in *International Conference on PErvasive Technologies Related to Assistive Environments*, Crete Island, Greece, 2012.

[140] Tim van Kasteren, Gwenn Englebienne, and B. Krose, "Activity recognition using semi-Markov models on real world smart home datasets," *Journal of Ambient Intellence and Smart Environments,* vol. 2, pp. 311-325, 2010.

[141] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 39, pp. 949-959, 2009.

[142] Y.-J. Lin, H. A. Latchman, M. Lee, and S. Katar, "A power line communication network infrastructure for the smart home," *Wireless Communications, IEEE,* vol. 9, pp. 104-111, 2002.

[143] D. J. Cook, M. Youngblood, I. Edwin O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja, "MavHome: An Agent-Based Smart Home," presented at the Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003.

[144] L. Kaila, J. Mikkonen, A.-m. Vainio, and J. Vanhala, "The eHome – a Practical Smart Home Implementation," in *Pervasive 2008,* Sydney, Australia, 2008, p. 7.

[145] H. Pigot, B. Lefebvre, J.-G. Meunier, B. Kerhervé, A. Mayers, and S. Giroux, "The role of intelligent habitats in upholding elders in residence," in *5th international conference on Simulations in Biomedicine,* 2003, pp. 497-506.

[146] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, "The gator tech smart house: A programmable pervasive space," *Computer,* vol. 38, pp. 50-60, 2005.

[147] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection," *Journal of telemedicine and telecare,* vol. 11, pp. 194-198, 2005.

[148] C. Phua, J. Biswas, A. Tolstikov, V. Foo, W. Huang, M. Jayachandran, A. P. W. Aung, P. C. Roy, H. Aloulou, M. A. Feki, S. Giroux, A. Bouzouane, and B. Bouchard, "Plan Recognition based on Sensor Produced Micro-Context for Eldercare," in *Proceedings of The First International Workshop on Context--Awareness in Smart Environments: Background, Achievements and Challenges (CASEbac 2009)*, Toyama, Japan, 2009, pp. 39-46.

[149] S. S. Intille, "Designing a Home of the Future," *IEEE Pervasive Computing,* vol. 1, pp. 76-82, 2002.

[150] M. S. Mittelman, S. H. Ferris, E. Shulman, G. Steinberg, and B. Levin, "A family intervention to delay nursing home placement of patients with Alzheimer disease : A randomized controlled trial," *Journal of American Medical Association,* vol. 276, pp. 1725-1731, 1996.

[151] T. Giannakopoulos, A. Pikrakis, and S. Theodoridis, "A novel efficient approach for audio segmentation," presented at the International Conference on Pattern Recognition, 2008.

[152] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "An adaptive framework for acoustic monitoring of potential hazards," *EURASIP Journal on Audio, Speech, and Music Processing,* vol. 2009, pp. 1-15, 2009.

[153] C. Belley, S. Gaboury, B. Bouchard, and A. Bouzouane, "Efficient and Inexpensive Method for Activity Recognition within a Smart Home based on Load Signatures of Appliances," *Journal Pervasive and Mobile Computing,* pp. pp. 1-20, 2013.

[154] D. Hauschildt and N. Kirchhof, "Improving indoor position estimation by combining active TDOA ultrasound and passive thermal infrared localization," pp. 94-99, 2011.

[155] P. Evensen and H. Meling, "Sensor virtualization with self-configuration and flexible interactions," presented at the Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems, Nara, Japan, 2009.

[156] G. E. Lancioni, M. L. La Martire, N. N. Singh, M. F. O'Reilly, J. Sigafoos, K. Pinto, and M. G. Minervini, *Persons with mild or moderate Alzheimer's disease managing daily activities via verbal instruction technology* vol. 23, 2009.

[157] A. Mihailidis, J. C. Barbenel, and G. Fernie, *The efficacy of an intelligent cognitive orthosis to facilitate handwashing by persons with moderate to severe dementia*. Hove, ROYAUME-UNI: Psychology Press, 2004.

[158] J. Rodin, *Aging and health: effects of the sense of control* vol. 233. Washington, DC, ETATS-UNIS: American Association for the Advancement of Science, 1986.

[159] J. Shirogane, T. Mori, H. Iwata, and Y. Fukazawa, "Accessibility Evaluation for GUI Software Using Source Programs," presented at the Proceedings of the 2008 conference on Knowledge-Based Software Engineering: Proceedings of the Eighth Joint Conference on Knowledge-Based Software Engineering, 2008.

[160] G. Demerism, B. K. Hensel, M. Skubic, and M. Rantz, *Senior residents' perceived need of and preferences for "smart home" sensor technologies* vol. 24. Cambridge, ROYAUME-UNI: Cambridge University Press, 2008.

[161] E. E. Stone and M. Skubic, "Evaluation of an inexpensive depth camera for passive in-home fall risk assessment," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*, 2011, pp. 71-77.

[162] J. Song, C. T. Haas, and C. H. Caldas, "A proximity-based method for locating RFID tagged objects," *Advanced Engineering Informatics,* vol. 21, pp. 367-376, 2007.

[163] Reynolds C., "Flocks, herds, and schools: a distributed behavioral model," *Computer Graphics,* vol. Vol. 21, pp. 25-34, 1987.

[164] C. Heesung and H. Kyuseo, "Combination of RFID and Vision for Mobile Robot Localization," in *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on*, 2005, pp. 75-80.

[165] B.-S. Choi and J.-J. Lee, "Mobile robot localization in indoor environment using RFID and sonar fusion system," presented at the Proceedings of the IEEE/RSJ international conference on Intelligent robots and systems, St. Louis, MO, USA, 2009.

[166] K. P. Fishkin, B. Jiang, M. Philipose, and S. Roy, "I Sense a Disturbance in the Force : Unobtrusive Detection of Interactions with RFID-tagged Objects," *Intellectual Property,* vol. 3205/2004, pp. 268-282, 2004.

[167] C. Hekimian-Williams, B. Grant, and P. Kumar, "Accurate localization of RFID tags using phase difference," *2010 IEEE International Conference on RFID IEEE RFID 2010,* pp. 89-96, 2010.

[168] K. Chawla and G. Robins, "An RFID-based object localisation framework," *International journal of radio frequency identification technology and applications* vol. 3, pp. 2-30, 2011.

[169] J. Zhou and J. Shi, *RFID localization algorithms and applications—a review* vol. 20. Heidelberg, ALLEMAGNE: Springer, 2009.

[170] A. Milella, D. Di Paola, G. Cicirelli, and T. D'Orazio, "RFID tag bearing estimation for mobile robot localization," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, 2009, pp. 1-6.

[171] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, "Implementing a Sentient Computing System," *IEEE Computer,* vol. 34, pp. 50-56, 2001.

[172] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper, "Implementing a Sentient Computing System," *Computer,* vol. 34, pp. 50-56, 2001.

[173] C. Y. Chen, J. P. Yang, G. J. Tseng, Y. H. Wu, and R. C. Hwang, "An Indoor positioning technique based on fuzzy logic," in *MultiConference of Engineers and Computer Scientists*, Hong Kong, 2010.

[174] G.-y. Jin, X.-y. Lu, and M.-S. Park, "An Indoor Localization Mechanism Using Active RFID Tag," presented at the Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06) - Volume 01, 2006.

[175] Q. FU and R. Guenther, *Active RFID Trilateration and Location Fingerprinting Based on RSSI for Pedestrian Navigation* vol. 62. Cambridge, ROYAUME-UNI: Cambridge University Press, 2009.

[176] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and Localization with RFID Tags," in *Procof the IEEE International Conference on Robotics Automation ICRA*, ed, 2004.

[177] Y. Zhang, M. G. Amin, and S. Kaushik, "Localization and tracking of passive RFID tags based on direction estimation," *International Journal of Antennas and Propagation,* vol. 2007, 2007.

[178] P. Vorst, S. Schneegans, Y. Bin, and A. Zell, "Self-Localization with RFID snapshots in densely tagged environments," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, pp. 1353-1358.

[179] D. Joho, C. Plagemann, and W. Burgard, "Modeling RFID signal strength and tag detection for localization and mapping," presented at the Proceedings of the 2009 IEEE international conference on Robotics and Automation, Kobe, Japan, 2009.

[180] Y. Lei, C. Jiannong, Z. Weiping, and T. Shaojie, "A hybrid method for achieving high accuracy and efficiency in object tracking using passive RFID," in *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*, 2012, pp. 109-115.

[181] M. Youssef, M. Mah, and A. Agrawala, "Challenges: device-free passive localization for wireless environments," presented at the Proceedings of the 13th annual ACM international conference on Mobile computing and networking, Montréal, Québec, Canada, 2007.

[182] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," *ACM Wireless Networks,* vol. 10, pp. 701-710, 2004.

[183] P. Vorst, S. Schneegans, Y. Bin, and A. Zell, "Self-Localization with RFID snapshots in densely tagged environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nice, France, 2008, pp. 1353-1358.

[184] Y. Lei, C. Jiannong, Z. Weiping, and T. Shaojie, "A hybrid method for achieving high accuracy and efficiency in object tracking using passive RFID," in *2012 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Lugano, Switzerland, 2012, pp. 109-115.

[185] K. Kim and M. Kim, "RFID-based location-sensing system for safety management," *Personal Ubiquitous Computing*, vol. 16, pp. 235-243, 2012.

[186] C. Y. Chen, J. P. Yang, G. J. Tseng, Y. H. Wu, and R. C. Hwang, "An Indoor positioning technique based on fuzzy logic," in *MultiConference of Engineers and Computer Scientists*, Hong Kong, China, 2010, pp. 854-857.

[187] J. Brusey, C. Floerkemeier, M. Fletcher, and M. Lane, "Reasoning about Uncertainty in Location Identification with RFID," *Security*, 2003.

[188] H. T. Friis, "A Note on a Simple Transmission Formula," *Proceedings of the IRE*, vol. 34, pp. 254-256, 1946.

[189] K. Bouchard, J. Lapalu, B. Bouchard, and A. Bouzouane, "Precise Passive RFID Localization for Personalized Service Delivery in Smart Home," in *Adaptable Service Delivery for Smart Environment*, Pittsburg, 2012, pp. 769-772.

[190] J. Katz, *History of Mathematics, 3rd edition*: Addison Wesley, 2008.

[191] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, pp. 311-324, 2007.

[192] T. Westeyn, H. Brashear, A. Atrash, and T. Starner, "Georgia tech gesture toolkit: supporting experiments in gesture recognition," in *Proceedings of the 5th international conference on Multimodal interfaces*, 2003, pp. 85-92.

[193] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, pp. 657-675, 2009.

[194] F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models," *Image and vision computing*, vol. 21, pp. 745-758, 2003.

[195] C. Shan, Y. Wei, T. Tan, and F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, 2004, pp. 669-674.

[196] F. Samaria and S. Young, "HMM-based architecture for face identification," *Image and vision computing*, vol. 12, pp. 537-543, 1994.

[197] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, 2002, pp. 423-428.

[198] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, 2000, pp. 410-415.

[199] P. Asadzadeh, L. Kulik, and E. Tanin, "Gesture recognition using RFID technology," *Personal and Ubiquitous Computing,* vol. 16, pp. 225-234, 2012/03/01 2012.

[200] A. P. Dempster, N. M. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological),* vol. vol. 39, pp. 1–38, 1977.

[201] Buckland M., "How to Create Autonomously Moving Game Agents," in *Programming Game AI by Example,* ed: Wordware Publishing, 2005, pp. 113-119.

[202] Xiaohui Cui, Jinzhu Gao, and Thomas E. Potok, "A flocking based algorithm for document clustering analysis," *Journal of Systems Architecture*, vol. Vol. 52, pp. 505-515, 2006.