



SAPIENZA
UNIVERSITÀ DI ROMA

MIFTel: a Multimodal Interactive Framework based on Temporal Logic Rules

Scuola di dottorato La Sapienza

Dottorato di Ricerca in Informatica – XXXI Ciclo

Candidate

Marco Raoul Marini

ID number 1310497

Thesis Advisor

Prof. Luigi Cinque

Co-Advisor

Prof. Danilo Avola

01 2019

MIFTel: a Multimodal Interactive Framework based on Temporal Logic Rules
Ph.D. thesis. Sapienza – University of Rome

© 2019 Marco Raoul Marini. All rights reserved

This thesis has been typeset by \LaTeX and the Sapthesis class.

Version: January 28, 2019

Author's email: marini@di.uniroma1.it

Abstract

Human-computer and multimodal interaction are increasingly used in everyday life. Machines are able to get more from the surrounding world, assisting humans in different application areas. In this context, the correct processing and management of signals provided by the environments is determinant for structuring the data. Different sources and acquisition times can be exploited for improving recognition results. On the basis of these assumptions, we are proposing a multimodal system that exploits Allen's temporal logic combined with a prevision method. The main object is to correlate user's events with system's reactions. After post-elaborating coming data from different signal sources (RGB images, depth maps, sounds, proximity sensors, etc.), the system is managing the correlations between recognition/detection results and events in real-time to create an interactive environment for the user. For increasing the recognition reliability, a predictive model is also associated with the proposed method. The modularity of the system grants a full dynamic development and upgrade with custom modules. Finally, a comparison with other similar systems is shown, underlining the high flexibility and robustness of the proposed event management method.

Ringraziamenti

Questo percorso di crescita personale ha portato ad interiorizzare conoscenze notevoli in ambito della computer vision e non solo. Devo molto al Prof. Luigi Cinque per questa opportunità offertami. Inoltre, la sua umanità è stata la virtù più rara che abbia trasmesso. Contestualmente, ho avuto il piacere e l'onore di poter lavorare con un gruppo di ricerca tra i migliori d'Italia. Il Visionlab è diventato un punto di riferimento in questi 3 anni e senza Daniele, Cristiano e Marco probabilmente non avrei raggiunto i risultati di oggi. Devo molto anche a loro. Infine, devo fare un ringraziamento speciale a Danilo Avola, colui che ha fornito un supporto determinante alla ricerca dell'intero laboratorio. Una guida, un maestro ed anche un amico.

Contents

Contents	v
1 Introduction	1
2 Related works	5
2.1 Temporal Logic and Events Management	6
2.2 Gesture Recognition and Pointing	7
2.3 Person Re-identification	10
3 System architecture	12
3.1 Layout Builder	13
3.2 Rules Builder	14
3.3 Rules Actuator	16
3.3.1 Sub-modules	17
4 Probabilistic temporal logic finite state machines	31
4.1 Temporal Logic Rules Management	31
4.1.1 Event Representation and Management	32
4.1.2 Probabilistic Finite State Machines	36
4.1.3 Events Reinforcement and Final Probability Update	45
5 Experiments and results	48
5.1 Sample Scenario: Interactive Museum	48
5.2 Physical Architecture of the Framework	49
5.2.1 Layout Builder	49
5.2.2 Rules Builder	52
5.2.3 Rules Actuator	53
5.3 Sub-modules Tests	56
5.3.1 Gesture Recognition: Experiments	56
5.3.2 Gesture Recognition: Results	64
5.3.3 Pointing: Experiments	66
5.3.4 Pointing: Results	67
5.3.5 Person Reidentification: Experiments	69
5.3.6 Person Reidentification: Results	70
5.4 Entire System Test	71
5.4.1 Experiments	77
5.4.2 Results	79

Contents	vi
6 Conclusions	84
Bibliography	85
List of Figures	98
List of Tables	102
List of Algorithms	104

Chapter 1

Introduction

Actions recognition is a complex field in computer science and still presents numerous open problems. It usually exploits specific devices designed for avoiding unnatural movements. These devices can be cameras, wearable sensors, microphones and each non-constrictive capture method. In fact, the main focus of this human-computer interaction (HCI) topic is to provide a high degree of freedom to the user. However, there are some limits that should be considered. First, this kind of interaction is harder to manage than a standardized one: a gesture performed with an arm is more difficult to identify than a tap on a touch-screen. It means that a proper classification of events requires both a higher computational power and more refined techniques. Ambiguity is an often underestimated obstacle that needs to be properly managed. During the years, new systems are developed for facilitating the natural interaction. For example, according to [119], in virtual and augmented reality the natural user interfaces (NUI) are extremely important for providing the immersion effect. Moreover, in this application area, different types of devices were proposed, from the simplest to the most advanced ones. They can be grouped according to the technology of the input device involved. The haptic devices are among the oldest and most restrictive ones. They usually provide very accurate information but oblige the user to use them according to their degrees of freedom. On the contrary, wearable devices are not so constrictive in movements, but require the touch of the user, compromising the hygienic factor. Moreover, they both are usually designed for specific interaction with body parts only. The vision-based devices are some of the most advanced ones for natural interaction, even though they require numerous shrewdnesses. Occlusions, illuminations, colours, shapes and a considerable amount of other noises could affect the results. On the contrary, there are minimal bonds for the user while interacting with the machine. Multimodal systems try to find solutions for improving recognition quality. They increase input source number and type, merging data for obtaining a single decision. According to Bourguet's definition, the term "multimodal interaction" consists in "interacting with the virtual and physical environment through natural modes of communication" [17]. Then, he underlines the multi-input characteristics and how to treat them properly. There are numerous approaches that use different techniques for specific aims [52], however there are some common elements between them: a lower bound of two input sources, a fusion method and a classification method. Concerning sources, the involved

sensors are related to each sub-problem for each study case. The collected data are obtained from different sources in each capture. In particular, the system considers specific features from each raw information and should use them for fusion and classification functions. The fusion can be applied before or after a first classification or a computation on each single feature [8]. According to [65] some factors are critical for allowing a fusion of different input type:

- **Noncommensurability:** A natural outcome is that the raw measurements may be represented by different types of physical units that do not commute;
- **Different Resolutions:** A scale factor to normalize different data;
- **Incompatible Size:** Different number of samples from different sources for each instance;
- **Alignment and Registration:** Registration is the task of aligning several data sets, usually images, on the same coordinate system;
- **Noise:** Each sensor can have a noise and it can influence the merged data;
- **Balancing Information From Different Origins:** Input data can have different confidence levels, reliability or information quality;
- **Conflicting, Contradicting, or Inconsistent Data:** Avoid conflicts between different sources data that can generate misunderstandings while merging and classifying;
- **Missing Values:** Avoid lack of data in an instance.

Classification usually consists in a data normalization process [53] by which a common scale can be used for fusing and classifying instances [81]. The results are related to the accuracy at each step of the system. In fact, if a sub-module is not accurate, its errors propagate and influence the final results according to its error percentage. In Figure 1.1, a generic architecture of a multimodal system is shown. Sequence steps are horizontally disposed.

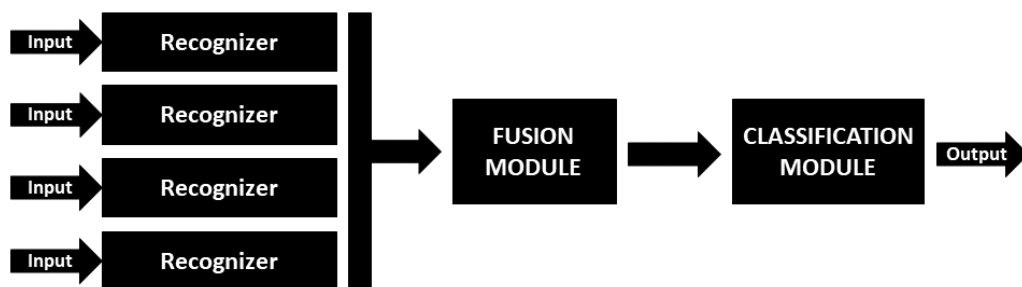


Figure 1.1. High level multimodal module architecture.

A more specific architecture is shown in Figure 1.2 according to [87]. This scheme is showing an example of steps that a dual modal (speech and gesture based) system should implement for a correct data flow management. In particular,

multimodal integration module is critical. It comprehends some focus operations such as temporal and semantic filtering, a hot topic in this field. Numerous works, described in the next chapter, present different approaches for optimizing the results of this module, adapting its algorithm according to the problem to solve and the environmental factors.

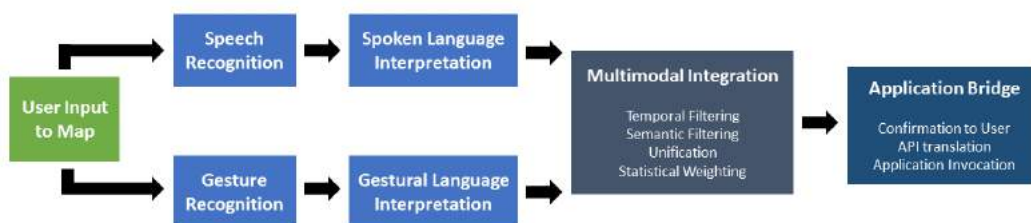


Figure 1.2. Complete logic architecture of a gesture and speech based multimodal system. The multimodal integration step is one of the most difficult and important phases to manage.

Some considerations in [86] support the effectiveness and the dynamism of multimodal systems in different scenarios. However, more than into the highlighted problems, the developer can get into other obstacles that can decrease the effectiveness of the system. As a consequence, a preliminary analysis is critical and requires a focus on sensor types, data normalization, fusion and classification methods for solving a specific problem. The context, the actors and the equipment involved in the scene must be clear to the developer and the researcher. The event recognition problem is strictly related with multimodal interaction. In fact, in the majority of the cases, multimodal systems are specifically designed for recognizing actions performed by a human. However, in biometry, multimodal systems are used for acquiring multiple information and merging them for increasing the identification accuracy [112]. Despite the different aims, data acquisition and merging techniques are usually shared between multimodal systems. Numerous examples of that are described in literature [16] [52]. Event recognition is an extremely wide topic and is linked to numerous other areas. It is used in different application contexts such as behavioral biometrics, content-based video analysis, security and surveillance, interactive applications, animation and synthesis [117]. It means that there is a wide range of variables that should be considered when deploying an event recognizer. Trajectories, gestures or poses can provide useful visive feedbacks for the system. But also speech, touch and other input methods can be associated to specific triggers. One of the most complete application area that involves multimodal system and numerous related topics is the medical rehabilitation. For its completeness, it is largely treated in the next chapter to explore the panorama of all possibilities in multimodal systems. It is also possible to associate inputs with semantic or temporal logic rules in these systems. A semantic rule is concerned with the relationship between signifiers and what they stand for [73]. One of the most important challenges in semantic connection discovery is concept disambiguation. It consists in contextualizing words according to the desired meaning. If we apply this concept to multimodal applications, a very complex and wide range of possibilities issues. In fact, the combination of different input types adds information to the

scene. If there is an incorrect management of these data, the disambiguation could be more difficult. On the contrary, classification can be more accurate analyzing merged data according to a correct policy. It means that multimodal systems could not be the solution for every natural interaction problem. In this context, the temporal logic algebra [2] could be an improving factor in complex event recognition. It is principally used in model checking and concurrent systems [24], however it can be used also in multimodal contexts because of application similarities. For example, some inputs of the same type can occur at the same time, generating concurrency. The developer could also need to log occurring events in time-space and evaluate them according to the registered sequence. Moreover, sometimes a temporal relation between events is needed for monitoring events in specific time intervals. The possible applications are numerous and various. However, we have to consider an important aspect of these systems: the final precision is always related to the precision of each sub-module that composes the entire system. This fact implies that temporal logic management is strictly related to multimodal classifier outputs. It is possible to introduce some techniques for increasing precision using probabilities. In this context, we are presenting a multimodal framework based on temporal logic rules. It allows to manage the application environment, the involved devices, the associated actions and the temporal logic rules with input and output. In particular, a custom grammar for managing rules is presented. This grammar is exploited by the user to properly configure the interactive environment according to his/her needs. Some prediction algorithms are also included in the computation for reducing ambiguities. Moreover, we present some study cases with sample sub-modules for testing the effectiveness of the system. The modules are some of the most commonly used in multimodal systems: gesture recognition, speech recognition, re-identification, trajectory recognition and quantity of motion. The involved sensors are RGB cameras, depth cameras and microphones. Output devices are speakers, micro-controllers and displays. We tested the effectiveness of the system evaluating the results obtained with two methods: comparing the performance of detectors and proposed classifier and comparing features and results with similar systems. In the first case, we performed some tests with users in two realistic scenarios and we evaluated the performances of the system according to the classification accuracy. We also collected personal judgments from the involved users for retrieving information about usability and interaction quality. The second test involves other systems. In particular, we compared the proposed framework with other similar ones, according to accuracy and features. The final results are promising. They encourage to improve the idea with future developments, discussed at the end of this document. The thesis is structured as follows. Section 2 summarizes related works in multimodal interaction and temporal logic algebra. A small section of it is also dedicated to gesture recognition and re-identification due to the fact that a sub-module for each one of them has been implemented. These two topics are some of the most discussed ones in computer science of last years. Section 3 presents the framework, its structure and the implemented sub-modules. Section 4 presents temporal logic algebra details, the used grammar and the proposed prediction method. In section 5 the experimental environments and the results are presented. We divided this section according to the previously mentioned categorization. Finally, in section 6 conclusions and future works are presented.

Chapter 2

Related works

Nowadays, multimodal systems are various and heterogeneous. Even if they could seem similar, they can be used for completely different tasks. It is almost impossible to define a common thread among all systems. Referring to [52], multimodal human computer interaction (MMHCI) systems can be categorized according to involved sensors. For example, in human-centered vision we can divide systems in large-scale body movements, hand gestures, and gaze. Also the fusion methods follow different approaches divided in early, intermediate and late. They can provide decisive changes in classification results [66]. However, the design of the system is always linked to the aim of the developer or the researcher. In fact, the architecture and the sensors are strictly related with the involved environments and the scenarios. For example, in [23] [102] ideas on how to exploit smartphones sensors for obtaining information on user's actions and classifying the performed activity are shown. In this case, the consideration of which sensors are embedded in the device and its dimensions is critical. On the contrary, in [79] a gesture recognition system based on machine learning and a multi-sensor camera (RGB and depth) are shown. In this case, the system is not portable and the device is not wearable. So, the usability context is completely different from the first example. A recent work [96] shows how a multimodal (video and audio) system can remotely provide a speech-to-text service. The relocation of elements of the system is a progress towards the distributed system concept, a technique used always more often for reducing computational costs. In fact, according to [25] a distributed application is compliant with the modularization principles, increasing the efficiency of the entire system. However, synchronization problems should be properly managed during data merge.

An important application area that is suitable for using multimodal systems is the medical one. In particular, body rehabilitation requires specific movements to be effective and multimodal systems can improve interaction (and capture) quality during exercises execution. Recently, the growing computational power of computers is supporting the use of machine learning (ML) and, specifically, deep learning approaches for solving problems. They are increasing systems' performances and also the multimodal ones. It means that an overview on rehabilitation multimodal systems using machine learning approaches can provide an enough accurate perspective on the entire panorama of possibilities. In this chapter, all the named topics are presented in detail, divided by category.

2.1 Temporal Logic and Events Management

Temporal logic algebra is usually linked to event recognition and management. Allen's logic is one of the most representative algebras for describing events in time interval-based relations [2]. In fact, this grammar allows to describe relationships between couples of events in defined temporal windows. In [14] the authors propose an improvement of this logic in two dimensional space called propositional spatio-temporal logic (PSTL), increasing the expressiveness of the language. Concerning probability, the work in [132] explores possible links between temporal logic and finite state machines. Dynamic Bayesian Networks are used for calculating probabilities among the states according to the events' temporal relationships. This kind of network is called Interval Algebra Network (IAN). Machine learning and Allen's logic are also combined in [22] using Markov Logic Networks (MLN)[100]. These networks are composed of undirected graphs and a set of potential functions. A potential function is a nonnegative real-valued function of the state of the corresponding clique (of the graph itself). This work is also connected to the gesture recognition topic, that is treated in the next chapter. In fact, temporal logic is usually applied on different contexts for solving problems not directly related to it. MLN are also used in one work of Song [113] for managing temporal logic, finite state machine and probabilities together. In [72] authors try to integrate multimodal inputs in accordance to spatial, temporal and semantic constraints. The visual state chart language is used in combination with an incremental and parallel parsing approach, allowing to manage continuous and discrete interactions. In this context, more complex systems can be involved. Frameworks allow to integrate multiple functionalities. They can be considered an upgrade of standard systems. In a recent work [28] a framework for activity recognition is presented. It uses intermediate semantic representation of concepts, always referring to Allen's algebra. For managing ambiguities, an ontology language and a probabilistic approach are introduced. During tests, researchers also evaluated the overall speed performances of the system. A synchronization was required among all components. This factor is often critical when real-time interactions are required. In this way, the delay of the system is reduced. The usability in a system based on interactive events management is extremely important. This work is one of the most similar to our own and the numerous common factors allowed us to compare some aspects at the very end of the document. In [15] a different approach is shown. Instead of Allen's logic, the authors proposed a method that is similar to a propositional logic grammar for describing events. In particular, the presented modal logic is related to multimodal scenarios, so they introduced a multi-dimensional logic. The described 2D modal logic is optimized for reducing expressive limits and improving overall performances with other methods such as modal conjunctive normal forms or continuous motion management. From this complex and wide panorama, we can denote that the majority of forms exploit Allen's temporal logic algebra for successfully managing events.

2.2 Gesture Recognition and Pointing

Gesture recognition is linked to computer vision for its intrinsic characteristics. The body of a person can provide direct and indirect information. For example, an explicit gesture is different from a pose acquired without the user's awareness. However, the captured data are more ambiguous than using other acquisition methods: haptic sensors are more accurate but more constrictive for the users. These characteristics bring researchers and developers to create more complex systems for managing information in the right way. According to [76], in Figure 2.1 the high level architecture of most common gesture recognition systems based on computer vision is shown. In the initialization phase the preliminary steps are performed, then the tracking method is applied according to the chosen policy. It can consist in single snapshots or in sequences of data. The pose estimation consists in associating a manikin or an avatar to the tracked body according to the user's movements. It could consist in stick-figures, a group of primitive geometrical forms (3D or 2D), a cloud of points or complex silhouettes. Finally, the recognition is performed to allow, for example, the pose identification.



Figure 2.1. A general structure for systems analyzing human body motion.

There are numerous approaches in literature for gesture recognition. The Hidden Markov Model (HMM) is one of the oldest a still used methods for gesture recognition. Since the first applications [129], HMM provided great results. For example, the system described in [101] is focusing on performing silhouette extraction with HMM and a discrete cosine transformed representation of the images. The statistical methods are largely used [75] and application modalities can be completely different from each others. For example [19] shows how to obtain inference from sequences and uses finite state machines for reducing ambiguity. In fact, one of the most difficult challenges in this application area is related to the management of a high number of variables. It can be reduced to a disambiguation problem. HMM can also be applied for body parts and to support the accomplishment of more specific tasks. The works of Starner [1, 115] are the first dedicated to gesture recognition for sign language recognition based on HMM. Numerous other techniques were introduced, often combining computer vision methods with machine learning principles [50]. The latter is always more often used also in this application area due to the fact that nowadays the machines are powerful enough to perform very complex calculations, as mentioned before. In fact, literature in the last years is presenting hundreds of deep learning based systems for gesture recognition [7]. The performances are better than classical machine learning based and not-machine learning based works. For example, in [91] a Convolutional Neural Network (CNN) based algorithm is presented for gesture recognition from Red Green Blue (RGB) videos. According to the results, the precision percentage denotes the incomparability with systems that do not use deep

learning. Moreover, not optimized or not combined methods are obsolete. In fact, in sign language gesture [71], pose recognition [82] and action recognition [120], also in the wild. Due to the nature of the topic, the majority of the approaches involve data sequences, using Long-Short Time Memory (LSTM) networks. The gesture recognition is such an actual topic that also industries are developing new devices, such as [84], every year. A sub-category of gesture recognition topic is dedicated to pointing management. It is still studied in recent years. In [63] the authors developed a robot-robot communication system based on a probabilistic pointing gesture recognizer. Differently from classical gesture recognition, this application area always involves the entire environment for allowing to identify the targets. The system presented in [92] is one of the most representative and recent works that comprehend both multimodal and gesture recognition. It uses hypothesis-dependent grammar to forecast possible actions that users could perform according to a scoring method. Concerning Allen's Logic in combination with gesture recognition, a work of Wan's team tries to merge these two topics with an inference engine [122]. However, the application of temporal logic based systems in combination with gesture recognition is very rare because these two topics are usually individually treated.

A field that can be representative for exploring the gesture recognition field from multiple points of view is the rehabilitation. It needs high accuracy and dynamism due to the requirements of the field. Numerous approaches have been proposed during years. In [31] a cluster of wearable sensors is used to capture body movements information of a patient while performing exercises. This kind of systems represents the new starting point for therapists and patients: the former can acquire more accurate data and the latter can increase their motivational factor. An increasing number of works is supporting human-computer and virtual/augmented reality principles for improving overall performances, expanding the cross-topic range. In fact, in recent years, virtual reality (VR) based systems are setting a new paradigm of immersive interaction. The systems supporting it can be categorized according to their involved capture devices. Vision based sensors are some of the most used, complex and versatile ones, so it is important to describe some of them according to their main characteristics. Regarding natural user interfaces (NUIs), the Kinect, a Microsoft multi-sensor composed by an RGB and depth camera and a microphone, is one of the most representative device. The authors in [108] proposed an effective rehabilitation system with an attractive game environment to increase the motivation and attention of patients. The system is composed of four exercise routines to train the patient's body balance and limb impairment. In [56], the authors developed a tele-rehabilitation system for stroke patients by using several devices. The Kinect device is used to measure the movements, while a 3D display is used to present 3D images by using binocular parallax. The acquired data is stored and supplied to the hospital by a Backend-as-a-Service cloud computing service. In [9], a system for developing serious games focused on body rehabilitation is proposed. The system acquires the patient's data by a Kinect and provides a feedback by using a 2D monitor. Authors in [109] proposed an upper body rehabilitation system based on a touchless interface. Designed specifically for hands, the patients can interact with objects in a virtual environment provided by a 3D monitor. In [104] and [114], two different full body touchless rehabilitation systems based on Kinect are proposed.

Table 2.1. Main characteristics of state-of-the-art systems

Works	Interaction type	Feedback type	Disease	Dedicated device
Gargantini et al. [41]	Head movement	3D glasses	Amblyopia	No
Munroe et al. [77]	Arm gesture	HMD	Cerebral palsy	No
Mavs et al. [68]	Head movement	HMD	Stroke	No
Knight et al. [61]	Entire body	Visual on projected images	Upper limb prosthetic training	Yes
Sen et al. [108]	Entire body	2D screen	Stroke	No
Kato et al. [56]	Entire body	3D screen	Stroke	No
Avola et al. [9]	Entire body	2D screen	Stroke	No
Shiratuddin et al. [109]	Hands	2D screen	Stroke	No
Saini et al. [104]	Entire body	2D screen	Stroke	No
Sosa et al. [114]	Entire body	2D screen	Multiple sclerosis	No
García-Martínez et al. [40]	Hands with specific controller	2D screen	Stroke	Yes
Pei et al. [89]	Entire body	2D screen	Stroke	No

In detail, in [104] a minimal customization of pre-built exercises is allowed, and the system implements a bio-feedback mechanism to evaluate the patients' performances. In [114], instead, the patients' performances are measured by the response time, velocity, and range of motion of neck, shoulder, elbow, hip and knee joints. Finally, another system that uses a Kinect device is reported in [89]. The authors proposed a system in which a 3D environment is presented to a patient by a 2D monitor. The rehabilitation training is evaluated with the Fugl-Meyer scoring method, which uses human body joints, in particular their angles, to evaluate the impairment of a patient. Concerning works that do not use NUIs, the authors in [41] proposed a system for patients suffering of amblyopia. The system is composed by a mobile application and a Google Cardboard, i.e., a low cost device able to reproduce virtual reality by means of a smartphone. In [77], the Myo armband [97] is used to perform home-based neurorehabilitation, by an augmented reality game, for children with cerebral palsy. In some contexts, the VR systems are independent from the used devices, as in [68]. In this work, a mobile VR application for rehabilitation programs of post-stroke patients is designed to work with different VR peripherals. Another interesting work is reported in [40], where a framework for designing rehabilitation exercises is described. As case study, the authors presented a game for hand rehabilitation by using a hand-grip sensor. Finally, there are works based on real advanced multimodal devices, as the CAREN system that performs upper limb prosthetic training and rehabilitation [61]. This system consists of 10 real-time motion capture cameras, with a double-belted instrumented treadmill and a 180 degree cylindrical screen projection device. This last kind of systems are very effective. Despite this, they are also extremely expensive and require wide rooms and dedicated devices. Moreover, they cannot be used in home environments for long duration rehabilitation. In Table 2.1, a summary of the main characteristics of the state-of-the-art systems is reported. Compared to the others, the proposed system is one of the most versatile in terms of motor rehabilitation (i.e., body and hands). Moreover, it is very cheap and does not require equipped rooms or dedicated devices. Finally, it is the only one that uses a deep learning algorithm to train and evaluate serious games.

Multimodal systems can also be associated with biometry [32]. A multimodal biometric system exploits more than a single biometry to identify the person. In particular, involved sensors are related to natural marks of users that provide biometric measurements. Gait, infrared thermogram, odor or hand geometry are example of personal marks that can be captured without asking to perform a specific action. It means that some marks can be captured without the user really noticing

it. Biometry is also linkable to reidentification problem, treated below.

2.3 Person Re-identification

Person re-identification is one of the open problems in computer vision. However, numerous approaches were proposed over the years for increasing the accuracy of the systems. In fact, this application area is affected by a multitude of factors that can completely fake the results: light conditions changes, occlusions, environmental changes, etc. However, the problem is related to the matching between models. According to [12], in Figure 2.2 the generic high level architecture diagram of re-identification systems is shown.

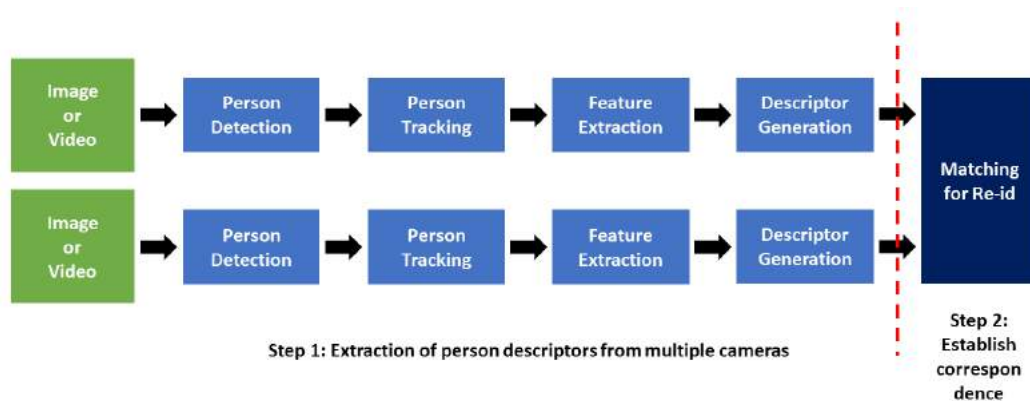


Figure 2.2. General re-identification systems diagram.

The main characteristic of this topic consists in allowing the identification of a subject among multiple devices. This process is very hard to manage due to numerous factors that alter the images, such as background, light, point of view (POV) of the camera, etc. It is usually applied in forensic and surveillance systems ???. According to that, appearance and biometrics features are the most used to perform a re-identification of a human being. One of the most relevant works proposed for re-identifying a person is [35]. It shows a method based on RGB cameras images analysis for comparing human figures according to their relevant features: chromatism, regions of colors and local motifs with entropy. However, nowadays all of the most innovative and performing techniques expect machine learning algorithms appliance [12]. For example, the ranking-based methods are used in some recent works with different learning and classification methods [94, 39]. The first work is implementing an improved version of classical support vector machine (SVM) with ranking features, the second one instead is a more complex system that exploits information from ranking correlations and contexts. In [21], the authors propose a novel approach for optimizing cross-view visual data. In particular, they designed a method for correlating common information from different sources according to features augmentation principles. The new adaptive space results more accurate than the single ones or other combination methods that are compared with it. A different approach is described in [130] where a privilege-based learning

method is described. It exploits both trained and raw data for providing a more accurate distance metrics while classifying. It is scalable and allows the multi-view of the scene. The authors in [125] tried to solve another problem related to machine learning based systems: information transferring. When a system is trained in a specific environment and the model is used into another one, the results are usually poor. They proposed a novel technique to exploit desirable common features among all RGB camera sensor systems. These features are called identitydiscriminative. It allows to directly re-identify people in an unknown environment, furthering the unsupervised re-identification methods. In [124], the authors proposed a machine learning method based on Mahalanobis distance for re-identifying people among a network of cameras. Class data separation is the first step of their method, minimizing intra-classes distances and maximizing differences between class instances. Then, a simplex is generated and a matching phase is executed to associate a probe to the nearest vertex of it through least-square regression. As shown in this chapter, the state of the art of re-identification is nowadays based on machine-learning methods and it is still in continuous evolution.

Chapter 3

System architecture

In this chapter the architecture of the framework is presented. In Figure 3.1 the high level structure of the system is shown. The scope of the system is to allow a developer to autonomously build and manage an interactive environment for multiple purposes. Concerning the latter, some sample scenarios are presented in the next chapters. Due to the complexity of the project, it is subdivided into multiple simpler tasks according to logical steps. First, the developer should provide information about the environmental structure, such as the dimensions of the involved area and the position of the objects inside it. Then, the temporal logic rules are needed. A section of the framework is specifically designed for allowing their planning. Finally, an actuator uses the information provided by the first two modules (Layout Builder and Rules Builder) for starting the interaction with the users. It is based on the parameters obtained according to the settings.

So, the framework is composed by 3 main parts, one for each specific aim:

- First step - Layout Builder
- Second step - Rules Builder
- Third step - Rules Actuator

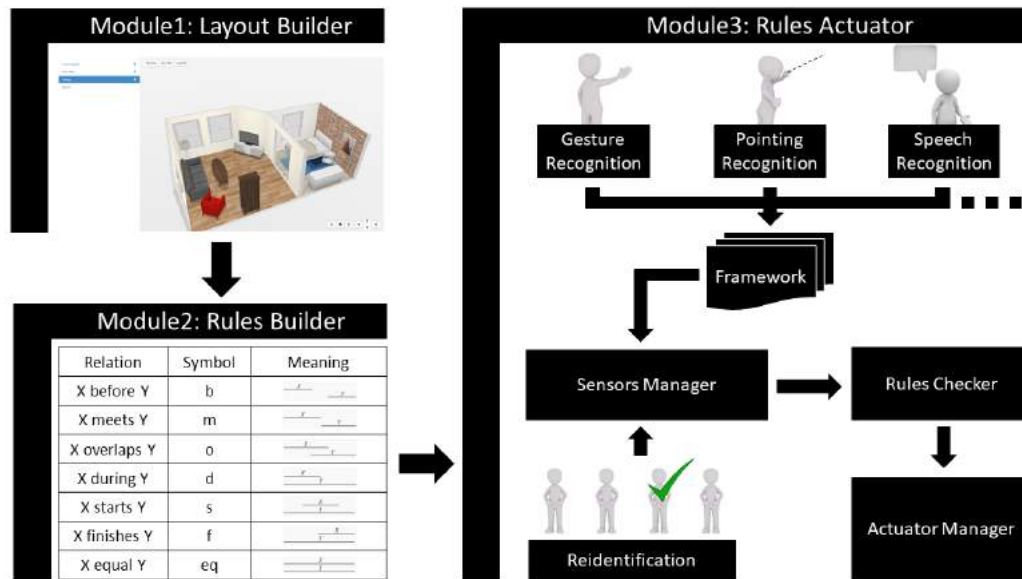


Figure 3.1. Framework architecture, overview.

This structure is specifically designed for simplifying the setup phase. In fact, the developer could not be able to set some parameters without related preparatory information. In the next sections, details of these 3 modules will be presented.

3.1 Layout Builder

The setup process starts from the planimetry of the involved scenario. In particular, dimensions and proportions are extremely important in this phase. The environmental layout is necessary for defining rules that link items and behaviors according to user's interactions. In [106], some specific functions are needed for allowing a simple customization of the environment without information loss or too high restrictions:

- **Translation:** the item in the scene can be moved on the three axes;
- **Rotation:** the item in the scene can be oriented modifying pitch, yaw and roll;
- **Resize:** the item in the scene can be resized (likely keeping aspect ratio).

However, for real implementations, some limits must be set anyway. For meeting the needs of developers and users, the following features are necessary:

- **Item descriptor:** the item inserted inside the scene should be identified with a unique ID. It avoids ambiguity;
- **Item type:** due to the fact that the system is multimodal, different sensor types should be considered. Input data and potential behaviors are related to that. In fact, the sensor type allows to categorize items and correctly treat interactions with other items or the user according to the item's possibilities. More in deep, the item type involves:

- **Input type:** it is specified only when the output type is omitted. The input type is a bound for the developer and the framework while setting up the input methods for the environment.
 - **Output type:** it is specified only when the input type is omitted. The output type is a bound for the developer and the framework while setting up the output methods for the environment.
 - **Category:** the item can be categorized according to its typology. Moreover, multiple sub-categories can be deployed for a more granular subdivision of characteristics.
 - **Added bonds:** the items can have some specific bonds related to their typology. These bonds can be completely different from each other.
- **Specific item bonds:** differently from item's type bonds, these limitations are related to the specific item. In fact, the developer could need to add some unique restrictions while building the environment layout.

The layout building phase is the first step of the data flow. The information retrieved at the end of the layout deployment are used in the next step of the framework.

3.2 Rules Builder

The rules builder is the second step of the data flow. It associates temporal operation to the items inserted in the environment. It is based on Allen's temporal logic [2] and some added functions. In particular, the rules building system is dedicated to the development of temporal logic formulas according to a specifically designed grammar and syntax. It allows to manage every kind of interaction that involves the environment and the users.

Relation	Symbol	Meaning
X before Y	b	
X meets Y	m	
X overlaps Y	o	
X during Y	d	
X starts Y	s	
X finishes Y	f	
X equal Y	eq	

Figure 3.2. Allen's temporal logic algebra applied on events (X and Y). Each formula can be negated.

In Figure 3.2 the Allen's temporal logic algebra operations are shown. The first column refers to the verbose form of the possible operations that can be performed. In the last column, a timeline in which events X and Y are correlated is shown. Each formula can be negated. This kind of problem is related to parallelism and multi-threading due to the fact that, in real cases, the events can overlap. Moreover, the multimodal approach increases the challenge. The involved parameters that the developer can use for building a complete rule are:

- **Involved items:** the items are the sources from where to capture information and produce output. It means that for each rule at least an input and an output device must be selected;
- **Involved event:** each item is linked to an input or an output event. Concerning the input, the developer can specify which kind of event he/she wants to be recognized by a specific device. The output is produced according to the characteristics and the availability of the item's types. Events can also be composed by multiple sub-events;
- **Temporal relations:** two events can be correlated together with a temporal relation, according to Allen's algebra. The developer can insert an arbitrary number of temporal relations and events in the conditional section of the formula, however, the temporal comparison is always performed between two contiguous elements and produces a boolean result. More details are provided in the following chapter;
- **Temporal interval:** Between event recognition and output generation the developer can set a temporal interval;

- **Involved persons:** Events can involve specific persons. The rule builder should allow the developer to indicate the system to produce different outputs if a well-known or unknown user is involved in a relevant event.

The rules builder offers the developer the tools for creating complex formulas using these elements. However, the scalability of the grammar allows to introduce new expressions when needed (if there are no conflicts with current grammar elements). The developer has a wide range of possibilities, enough to cover the most common human activities. The syntax is defined as following:

$$\begin{aligned}
& \text{TemporalOperation}_1 \{ \text{ItemInputID}_a (\text{input}_b, [\text{personInvolved}_c]), \\
& \text{ItemInputID}_d (\text{input}_e, [\text{personInvolved}_f]) \} \\
& \wedge \text{TemporalOperation}_2 \{ \text{ItemInputID}_g (\text{input}_h, [\text{personInvolved}_i]), \\
& \text{ItemInputID}_j (\text{input}_k, [\text{target}_l (\text{item}_m)]) \} \dots \\
& \rightarrow [\text{IntervalBeforeOutput}] \\
& \rightarrow \text{ItemOutputID}_n (\text{output}_o) [\text{IntervalBetweenOutputs}_1] \\
& \text{ItemOutputID}_p (\text{output}_q) \dots
\end{aligned} \tag{3.1}$$

where ItemInputID refers to the identifiers of the input devices inside the scene, input is the input to recognize, *personInvolved* is the possible person involved in the event, TemporalOperation is the operation that correlates the previous and the following events, IntervalBeforeOutput is the time interval before producing the output specified with *output* produced by ItemOutputID. Temporal logic rule can be linked with AND (\wedge) or OR (\vee) operators. The subscripted alphabet letters are only used for distinguish each element. The arrow separates the two parts of the formula. The first one can be considered the hypothesis and the second part the thesis. If not satisfied, the hypothesis condition is the "conditio sine qua non" the thesis, so the output, is not involved. According to this notation, the user is able to manage almost every kind of temporal event that can occur in the environment.

In other words, the proposed grammar is a formalization of sentences that the developer could express in normal language: "If the sensor X is recognizing event A and at the same time sensor X is detecting that person P is in the scene, then actuator Z produces output O."

Moreover, a logical connection between rules is provided. This function collaborates in identifying consequences among all possible rules. This operation is deeply explained in the next chapter.

3.3 Rules Actuator

This module is the core of the system. The prior inserted information are necessary for setting up the virtual environment. The collected data are finally treated according to the Rule Actuator. It consists in an infrastructure specifically designed for monitoring the status of the system and acting when necessary. In particular, it checks the completeness of the hypothesis of each rule according to the incoming recognized events. In Figure 3.3 input, the architecture of the actuator is summarized.

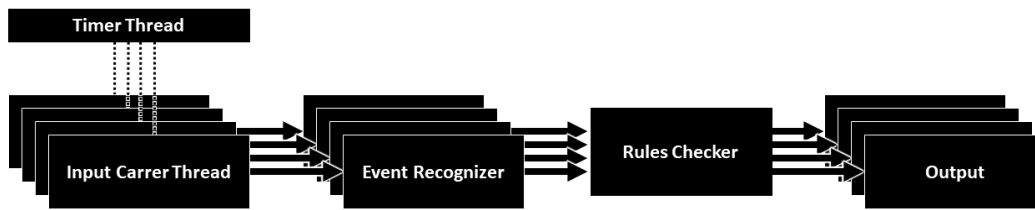


Figure 3.3. Actuator's logical data flow.

A timer synchronizes the input carrier threads and a specific window size is applied to each one of them. It implies that the system can be treated like a timed finite state machine. The finite state machines are structures of the family of graphs and belong to the automata theory [62]. These structures allow to track the status of a system and design the possible paths between a state and another. More details are explained in the next section.

3.3.1 Sub-modules

The rules actuator's scalability allows to insert and remove any number of sub-modules. These correspond to features or operations that the system can execute. The Temporal Logic Module is the only one that can not be removed because it corresponds to the "rules checker" box of the Figure 3.3.

Temporal Logic Module

According to [4], we can infer that four types of interactions can be identified in computer science application area: peer-to-peer (P2P), machine-to-peer (M2P), peer-to-machine (P2M) and machine-to-machine (M2M). Considering the aim of the proposed framework, only two of these interactions should be managed, the P2M and the P2P. The first one is compliant with the concept of event recognition. In fact, the interaction between two users can be captured by a sensor and processed for analyzing their behavior. Obviously, it requires a minimum number of two actors in the scene. On the contrary, the second and the third interactions are the most common ones. In these cases, the interaction types can be divided in two classes: when the user wants to interact with the machine, the interaction can be classified as "direct". On the contrary, when a user just performs some actions and the machine autonomously identifies them, the interaction is "undirected". Both cases should be considered while exploring the entire range of possible interactions. That being so, the temporal logic is specifically designed for managing incoming events distinctly from source types and numbers. It tracks the relevant past identified actions performed and detected. The surrounding information, like the ones related to specific persons, type of actions and time intervals support the system's status identification, according to a "finite state machine" representation. In the next chapter the temporal logic method is described in detail, underlining the link between events and probabilities. The next presented modules are samples of possible modules that can be implemented and connected to the temporal logic module.

Gesture Recognition

The gesture recognizer is one of the possible modules that can be integrated into the proposed framework. The vision-based sensor type that is considered is the depth one. In this case, the information of each frame is collected in a depth map. It is generated calculating the distances of the surfaces from the POV of the sensor. The resulting image indicates the depth levels in greyscale, with darker (or lighter) shades based on the distance. This technique is very useful when shapes and foreground elements should be detected. The analysis criteria is based on the use of the depth maps created by the depth sensor. For detecting the human figure inside the scene, body parts are derived from the depth information. The algorithm is based on [110] and is composed by three steps: training, rendering of fictitious data and applying a randomized decision forest for identifying body parts. The latter passage is crucial for allowing to take intermediate and final decisions while random questions are selected in a pool of about 2000. These questions are related to a comparison between pixel normalization, for identifying offsets. The resulting body parts are shown in Figure 3.4.

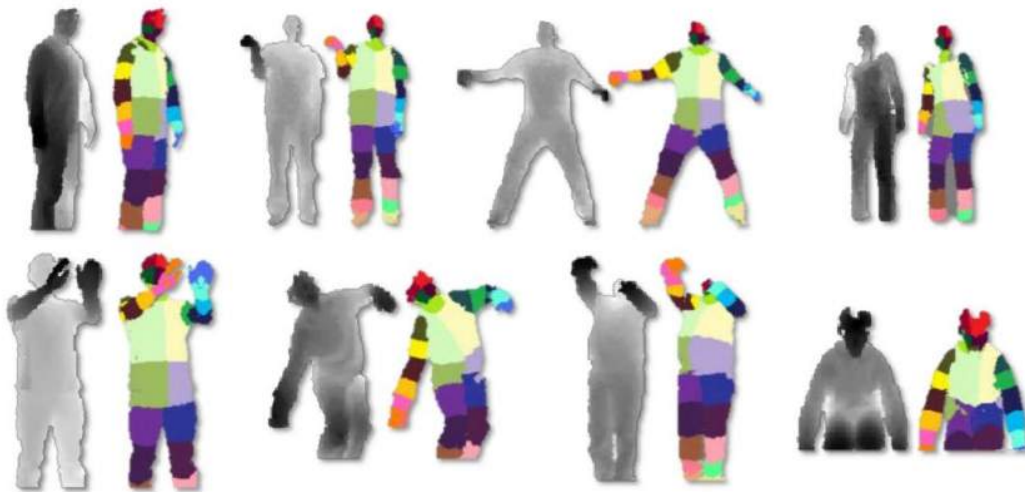


Figure 3.4. Association between depth map (images on the left) and body patches (images on the right) generated during human body detection phase. Image from [110].

The foreground shape detection allows to proceed to the next steps: the representation of intermediate body parts and their skeletal transformation. It is done using mean shift algorithm [26] with a weighted Gaussina kernel. The density estimator per body part is defined in [110] with the formula 3.2:

$$f_c(\hat{x}) \propto \sum_{i=1}^N \omega_{ic} \exp\left(-\left\|\frac{\hat{x} - \hat{x}_i}{b_c}\right\|^2\right) \quad (3.2)$$

where \hat{x} is a coordinate in 3D space, N is the number of image pixels, ω_{ic} is a pixel weighting, \hat{x}_i is the reprojection of image pixel x_i into the scene space given depth $d_I(x_i)$ and b_c is a learned pre-part bandwidth. The pixel weighting ω_{ic} is defined according to formula 3.3:

$$\omega_{ic} = P(c|I, x_i) \cdot d_I(x_i)^2 \quad (3.3)$$

where c is a body part and $P(c|I, x_i)$ is a posterior probability that can be applied over a small set of parts. The skeletal map generated in this way is defining a "stickman" with human structure which junction points are almost completely covering the real ones. It means that the generated model is accurate enough to allow the management of full body movements. The entire structure of joints and connections is show in Figure 3.5.

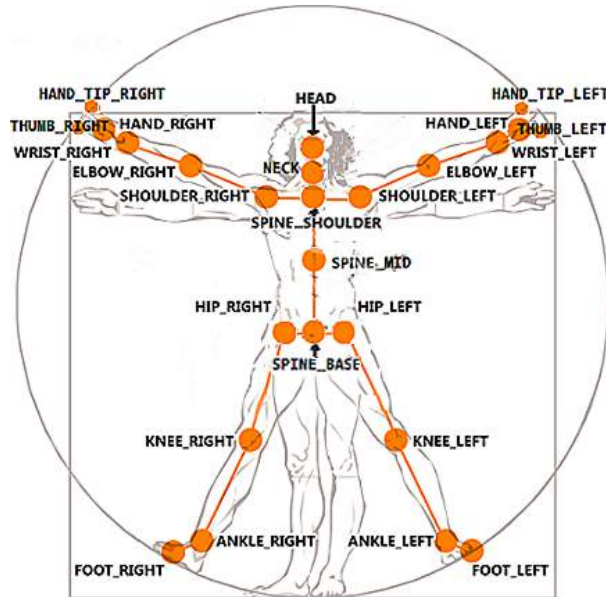


Figure 3.5. Joints map created by Kinect SDK 2.0.

Starting from this model, we developed a machine learning structure able to exploit those joints for identifying gestures and poses.

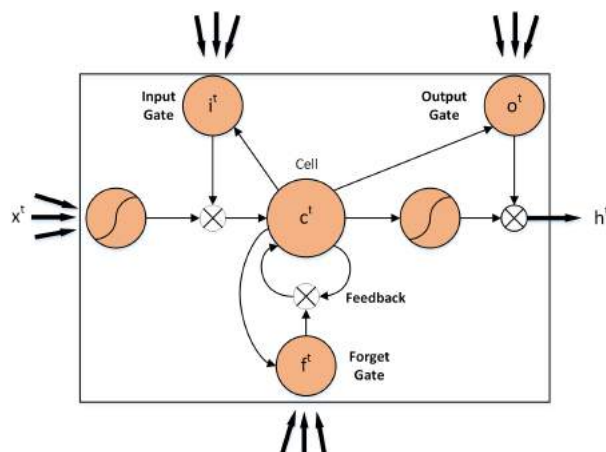


Figure 3.6. Representation of a LSTM block having one cell

The retrieved skeletal structure is the ground information for the machine learning

module. The proposed gesture recognition system is, in particular, using the deep learning principles of LSTM[47]. In general, a Recurrent Neural Network (RNN) is a neural network in which there is a feedback loop that produces a recurrent connection, thus allowing the RNN to model the contextual information of a temporal sequence [7, 131]. Considering an input sequence $x = (x^0, \dots, x^{T-1})$, the hidden states of a layer $h = (h^0, \dots, h^{T-1})$, and the output of a single RNN layer $y = (y^0, \dots, y^{T-1})$, this last can be computed as follows:

$$h^t = H(W_{xh}x^t + W_{hh}h^{t-1} + b_h) \quad (3.4)$$

$$y^t = O(W_{ho}h^t + b_o) \quad (3.5)$$

where, W_{xh} , W_{hh} , and W_{ho} are the weights of the connections among input, hidden layer, and output layer. While b_h and b_o are the bias vectors, finally, $H(\cdot)$ and $O(\cdot)$ are the activation functions of the hidden and output layers.

Even if the RNNs are able to handle long term dependencies, different works have highlighted that the training of these networks is a hard task due to the vanishing gradient problem, as reported in [13]. For this reason, in the proposed system, the LSTM networks, which are a particular kind of RNNs, are used. To overcome the RNNs limits, the LSTMs are explicitly designed to learn long-term dependencies. This is why they are usually used for handwriting recognition, speech recognition and translation and, in general, for propagating data over time [111]. In detail, a LSTM block is designed to replace the non-linear units within a traditional RNN. In Figure 3.6 a LSTM single-cell memory block is shown. Notice that it contains one self-connected memory cell, denoted as c^t , and three units called *gates*. These gates are used for storing and accessing the previous contextual information of a temporal sequence (t represents the reference to the time), they are of three types: input gate, forget gate and output gate. In Figure 3.6, they are referred as i^t , f^t , and o^t , respectively. The activations of the LSTM components are defined as follows:

$$i^t = \sigma(W_{xi}x^t + W_{hi}h^{t-1} + W_{ci}c^{t-1} + b_i) \quad (3.6)$$

$$f^t = \sigma(W_{xf}x^t + W_{hf}h^{t-1} + W_{cf}c^{t-1} + b_f) \quad (3.7)$$

$$c^t = f^t c^{t-1} + \tanh(W_{xc}x^t + W_{hc}h^{t-1} + b_c) \quad (3.8)$$

$$o^t = \sigma(W_{xo}x^t + W_{ho}h^{t-1} + W_{co}c^{t-1} + b_o) \quad (3.9)$$

$$h^t = o^t \tanh(c^t) \quad (3.10)$$

where, $\sigma(\cdot)$ is the sigmoid function, and all the $W_{..}$ are the matrices of the connection weights between two units. For the learning architecture, we obtained a deep LSTM network by stacking two LSTM layers. In this way, the features from a given time instant are processed by a single non-linear unit before contributing to the output [105]. This submodule is exploiting LSTM improved technique to classify gestures, poses and actions performed with the entire body. We proposed a performing solution according to literature of machine learning in gesture recognition [118]. It consists in the combination of RGB, depth and retrieved skeletal data. The first passage was the definition of a correct window size for the image sequence of a gesture. The topic is largely treated [10] and we can't find a standard value. However, the time range interval is between 1 and 7 seconds in the majority of the publications. It depends on contexts and gesture types. During experiments we denoted that the middle value

of 4 seconds is enough for completing some of the slowest actions (in the proposed experimental scenarios). However, we adapted it to the different tested dataset sequences dimension. The RGB and Depth data are treated with Convolutional Neural Networks (CNN) in sequences, according to the approach described in [134]. It is based on 3D-CNNs [116] that are specifically designed for preserving temporal information of input signals (images), resulting in output volumes. Following the approach described in [134], processing phase is divided into three steps: each group of frame of RGB and depth sequence is processed into a convolutional LSTM, then a spatial pyramid pooling is applied for reducing the dimensionality and finally the fully connected layer of the net learns from the generated feature maps. For improving the fusion of scores, a method shown in [36] is applied. The two networks' features are fused in a middle convolutional layer and a single net ends the computation. As previously mentioned, the obtained output is combined with the result of the skeleton analysis. The latter operation is inspired by another method, described in [67]. The spatio-temporal LSTM (ST-LSTM) structure allows to easily track joints of the skeleton during the time. We used a stacked architecture of three ST-LSTM. The features are selected according to the most relevant body parts involved in common gestures. In experimental phase we tuned and finalized them. The Table 3.1 shows the list of features, categorized per distances, angles, orientations and speeds.

Table 3.1. Skeleton features for gesture recognition.

Distance Features	Angle Features	Speed Features	Orientation Features
Distances between hands and spine base		Speeds of hands	Orientations of hands
Distances between wrists and spine base	Angles between wrists, elbows and shoulders	Speeds of wrists	Orientations of wrists
Distances between elbows and spine base	Angles between elbows, shoulders and spine shoulder	Speeds of elbows	Orientations of elbows
Distances between shoulders and spine base	Angles between shoulders, spine shoulder and spine base	Speeds of shoulders	Orientations of shoulders
Distance between neck and spine base	Angle between head, neck and spine shoulder	Speed of head	Orientation of head
Distance between head and spine base		Speed of spine shoulder	Orientation of spine shoulder

In this way, two networks were trained over three different data sources. For classifying an instance, the final step consists in a late fusion. After *softmax* function, the probability values of each class from both vectors is summed, obtaining the final score. In this way we avoided the early fusion due to possible inconsistent characteristics [83] between nets. The entire process is summarized in Figure 3.7.

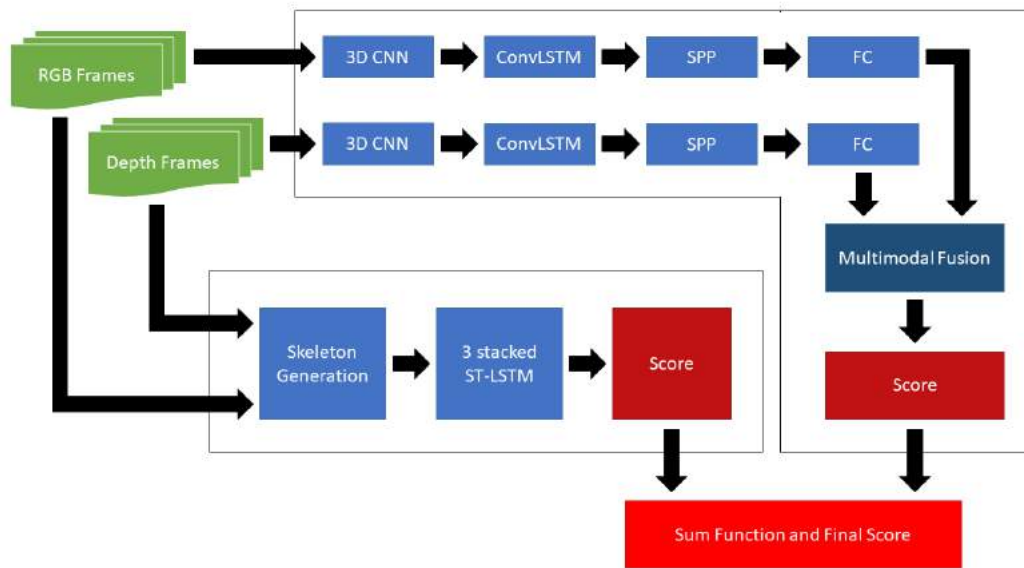


Figure 3.7. Logical architecture of the proposed gesture recognizer.

Pointing Recognition

The pointing action is one of the most used when the body is involved in a natural interaction. This gesture is commonly performed in multiple situations, so, we decided to implement the proposed pointing recognition module in our framework for testing its behavior in combination with the other modules. Since in this case there are numerous background information, the implementation of this module is based on stored environmental data. The majority of well-known methods for analyzing pointed direction [63, 27] start from different situations, usually with none or minimal information of the scene. This allows us to achieve better results with lower computational weight.

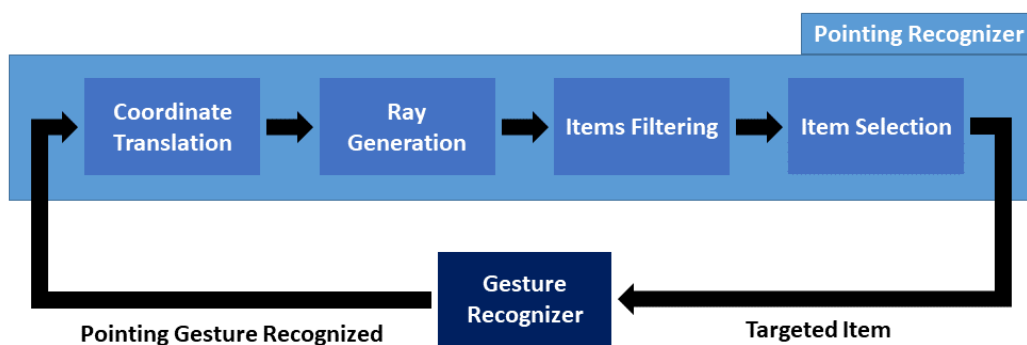


Figure 3.8. Pointing recognition module: logical architecture and its communication method with the Gesture Recognizer. When the recognizer identifies a pointing action the target analysis starts. The process provides an item as output of the module. This information is exploited by Temporal Logic Module to verify the status of the system.

The proposed method is based on the depth data provided by the depth map. In

Figure 3.8 the logical structure of the module is shown. The first step requires that the gesture recognition module identifies a pointing action performed by the actor in the scene. When this trigger is activated, the target identification starts. Background information allows to avoid point cloud processing for retrieving data of the dimensions of the objects, reducing computational cost. The same happens for the distances. Considering the fact that the camera is the referring point of the scene, some operations are needed for identifying the exact position inside the environment of a captured element. Considering the POV of a generic depth camera placed in horizontal position, the most common one, the environmental reference system is different from the one related to this POV. It means that a translation function is needed when the camera is detecting an object inside its field of view and its coordinates should be identified inside the environmental system coordinates. In Figure 3.9 an example of third person view (a) and first person view (FPV) from the POV of the camera (b) is shown. This scenario corresponds to the simplest one: the camera has the same reference system as the environmental one. However, when the camera is rotated on the Y axis (green), the reference systems become different. Due to the fact that the majority of depth cameras correctly work when they are frontally placed in respect of the human figure, we do not consider the fact that it can be rotated also on X axis (red). The rotation on Z axis (blue) has no application too.

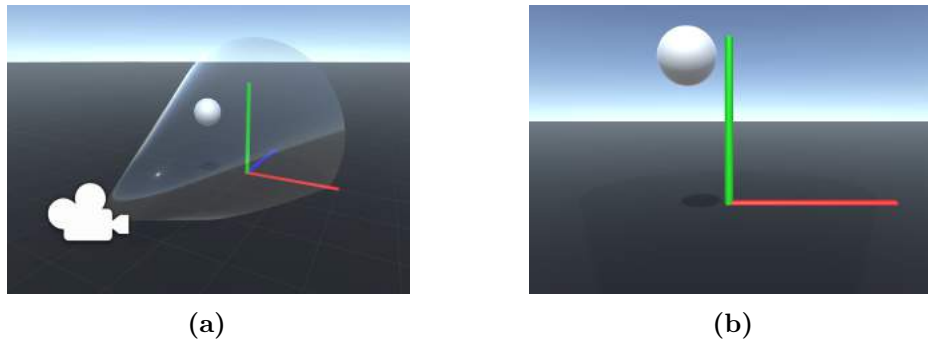


Figure 3.9. (a) Side view of sample 3D environment. The depth camera (white camera on the left) has a field of view defined by the transparent conic shape. The reference system is expressed according to the Y axis (green), X axis (red) and Z axis (blue) shapes. The sphere represents a sample object inside the field of view of the camera. (b) First person view (FPV) of the depth camera in the sample scene shown in figure a. The reference system is defined according to the X (red) and Y (green) axis of the image, width and height respectively. The Z axis (blue) is the depth one, indicating the distance of an object from the POV of the camera. Due to the fact that the environmental reference system is fixed, if the camera is rotated on Y axis, a coordinate translation method is required for calculating the real position of an object in the scene when it is identified only by the camera.

Considering the fact that the main reference system is always the environmental one, from now we will consider only 2D images from top view of the scene, because the Y axis always corresponds to the camera's reference system one. This approach helps the geometrical analysis of the proposed translation problem. If θ is the orientation of the camera (on Y axis), (K_x, K_y, K_z) is the coordinates vector of the camera and

(J_x, J_y, J_z) is the coordinates vector of a point in 3D space, the translation function can be applied according to the following formulas, one for each axis:

$$T_x := K_x + (\cos(\theta) \times J_x) + (\sin(\theta) \times J_z) \quad (3.11)$$

$$T_y := K_y + J_y \quad (3.12)$$

$$T_z := K_z + (\cos(\theta) \times J_z) + (\sin(\theta) \times J_x) \quad (3.13)$$

where (T_x, T_y, T_z) is the coordinates vector of translated position of the 3D point into the main reference system. The latter is shown in Figure 3.10a. The camera is oriented with the environment, so $\theta = 0^\circ$. A point can be easily translated with the defined formulas in the main reference system in the simplest case, as shown in Figure 3.10b and in more complex ones, as shown in Figure 3.10c.

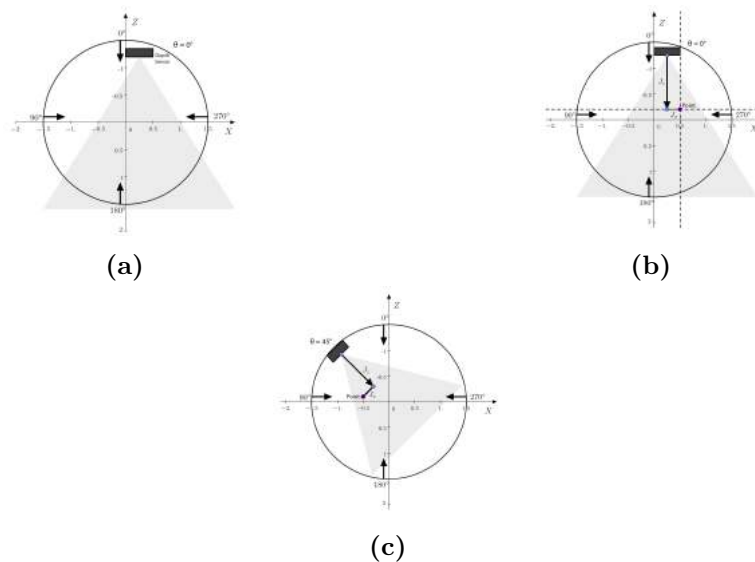


Figure 3.10. (a) Sample environment from top view. The camera is oriented according to $\theta = 0^\circ$. (b) Sample environment from top view in which a point should be translated. The camera is oriented according to $\theta = 0^\circ$. c Sample environment from top view in which a point should be translated. The camera is oriented according to $\theta = 45^\circ$.

In a common reference system the calculation of pointing direction and targeted objects is easier. However, the techniques that allow to perform this task are numerous. We decided to use a probabilistic approach. Before calculating the target, we need to cast the ray. Also in this case, it is possible to use different information for applying distinct procedures: exploiting fingers [38], the entire arm [95, 37, 27] or other naive approaches [57]. The most common and supported method, according to the considered scenario, is the second one for the following reasons:

- Physical limits: the depth sensors, like the Kinect for XBOX One used in our experiments, are not able to distinguish tiny object, such as fingers, at medium distance yet,

- General performances: it provides the best results among all approaches,
- Minimal control: it is also effective in scenes "in the wild".

In particular, the proposed method exploits elbow and wrist of the dominant arm (specified when the person is registered in the system, otherwise the right arm is assumed as dominant). When a human points an object with his/her arm, the pointed direction can be easily approximated with a ray starting from the elbow and passing through the wrist. The method is considered reliable also when the arm is flexed. For casting the ray, it is sufficient to apply a simple coordinates vector subtraction between wrist and elbow vectors. The result is a line that defines the pointed direction. The last steps consist in filtering the object according to a logical approach in order to avoid ambiguity. In this phase multiple factors are introduced to reach the goal. The first step consists in calculating the ray passing on each object centroid and the elbow of the user. This operation allows to identify distances (the distance in 3D space between the centroid and the elbow joint for each object) and angles between the ray casted by the user and the ones generated for each object. The angle is calculated after normalizing the coordinates vector between elbow and object and finally applying a scalar product between the generated and the casted ray. The entire operation can be summarized in the following steps:

- Trace the ray passing through user's elbow and wrist. It can be defined as vector \vec{D} ,
- For each object i , trace a ray passing through its centroid and the user elbow. It can be defined as vector \vec{O}_i ,
- Normalize each vector \vec{O}_i ,
- For each item i calculate the angle cosine with the following formula:

$$\cos(\alpha_i) = \vec{O}_i \times \vec{D} \quad (3.14)$$

- For each item i calculate the angle with the reverse trigonometric formula:

$$\alpha_i = \arccos(\cos(\alpha_i)) \times \left(\frac{180}{\pi}\right) \quad (3.15)$$

Each angle α is defining the distance between the pointed direction and the position of the object in the space. A threshold is needed for starting the final phase: the filter over the object. According to the experimental results and the other probability-based systems [37], the best threshold aperture angle is 30° . It means that the area of candidate objects is defined by a cone with 30° vertex angle; the vertex starts from the wrist of the user and the height of the cone is parallel to the ray casted by the user. In Figure 3.11 a sample scenario of the pointing action is shown. The points E_p and W_p are the elbow and wrist of the user, the conic area is colored in red. The candidate objects are O_1 and O_2 .

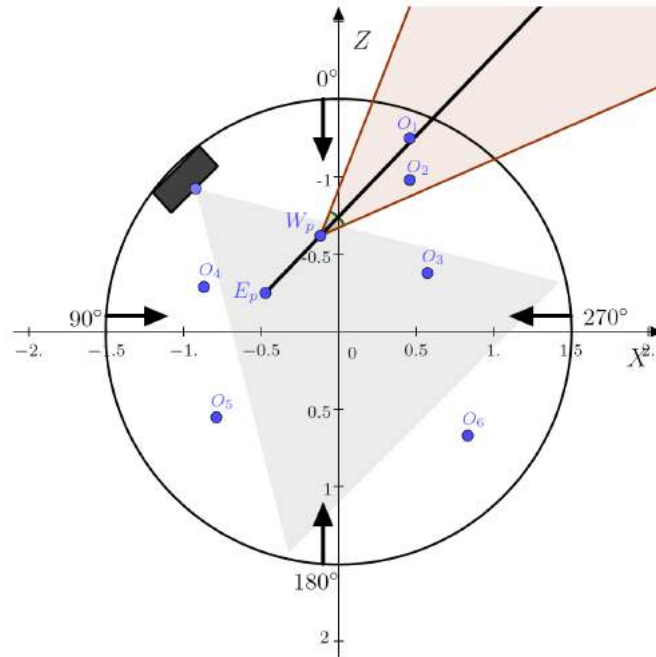


Figure 3.11. Sample scenario of pointing action from top view. The points E_p and W_p are the elbow and wrist of the user. Each element labeled with O is an object. The conic selection area is colored in red. O_1 and O_2 are the candidate objects.

The last step consists in the target selection. Three variables are involved: α , the distance between the centroid object and the perpendicular to \vec{D} and the dimensions of the object. The distance d is calculated with the following formula:

$$d_{ci} = (dis_{ci} - dim_{ci}) \times \alpha \quad (3.16)$$

where, for each candidate item ci , dis is the minimum distance between the centroid of the item and the ray \vec{D} according to [127], dim is the dimension of the item considering it as a sphere (and it corresponds to its radius). The lowest value is the nearest object to the casted ray, so the pointed one.

However, for our specific integration in the proposed framework, a percentage value of probability is required for each item. It means that a proportion is performed, according to the following formula:

$$x_{item} = 100 - \left[\left(100 \times d_{ci}^{item} \right) / \sum d_{ci} \right] \quad (3.17)$$

where x_{item} is the percentage value for the $item$ and d_{ci}^{item} is the value d_{ci} of the $item$. Then, the higher value is indicating the most probable pointed item and is selected as "target".

We have to underline the fact that the strategy of using a conic area is specifically chosen for reducing both research area dimension and computational cost.

Person Identification and Re-identification

A person can be identified and re-identified inside an environment when his body is detected. Each term is referring to a specific function: the first one is the process that allows the machine to recognize an already known user; the second operation is the identification of the same person from different sources, usually between a short time period. In our proposed methods, both these functions share a section of the approach but are not the same. In fact, the identification process is usually based on the physical features of the user, such as biometrical information. The re-identification, instead, can also involve "temporary" features, such as clothes, due to the fact that the comparison between the instances is performed in a short time period. We focused on RGB and Depth also in this module. Concerning identification method, it is limited by the application environment. In fact, in literature the most common systems are tested in controlled environments [78, 6, 121, 30], considerably decreasing the challenge level. It implies that, in our case, an effective approach in uncontrolled scenarios is needed. In a [88] an interesting re-identification system is shown. In particular, one of the operation performed is involving the skeleton of the human (the same structure described in Gesture Recognition section of this document). Some anthropometric measurements are introduced. They are considered "soft biometrics" due to their inaccuracy in comparison with other more robust approaches, such as fingerprint, face recognition or iris scan. However, differently from those listed methods, anthropometric measures can be acquired from a distant sensors. The distances are the following:

- $d1$ = distance between floor and head;
- $d2$ = ratio between torso and legs;
- $d3$ = height (distance between the highest body silhouette point and the floor plane);
- $d4$ = distance between floor and neck;
- $d5$ = distance between neck and shoulder;
- $d6$ = distance between torso centre and shoulder;
- $d7$ = distance between torso centre and hip;
- $d8$ = arm length (sum of the distances between shoulder and elbow, and between the elbow and wrist);
- $d9$ = leg length (sum of the distances between hip and knee, and between knee and ankle).

These measures can be respectively compared for identifying the similarity between the probe subject and the ones stored in dataset. The first step consists in capturing these measures in fair conditions. In fact, the uncontrolled environment allows a high freedom level for the user and, usually, it means that if not correctly managed the captures can be inaccurate. For avoiding that, we set two threshold distances

that indicate the optimal capture interval. These two values depend on the sensors characteristics. Finally, we denoted that only one capture can be not precise enough for identifying a person due to possible temporary glitches, bugs, occlusions or other noise factors. So, we propose a method for stemming the problem: the average among multiple captures. It allows to reduce the error and is largely used in biometric multimodal systems [54]. The entire process is composed by a second step, the comparison method. It is based on a simple difference and an algorithm for calculating the scores. The complete scheme of steps is shown in Algorithm 3.1:

```

Function CaptureFunction(captureFramesVector):
    | captureVector  $\leftarrow$  0;
    | foreach anthropometric measure  $d_i$  do
    | | captureVector  $\leftarrow$  calculate z-score value of  $d_i$  according to each
    | | capture in captureFramesVector;
    | end
Function findPerson(captureVector):
    | resultsVector  $\leftarrow$  0;
    | foreach person  $k$  in dataset do
    | | foreach anthropometric measure  $d_i$  in captureVector do
    | | | resultsVector[ $k$ ][ $i$ ]  $\leftarrow$  difference between  $k(d_i)$  and
    | | | captureVector( $d_i$ );
    | | end
    | end
    | foreach person  $p$  in resultsVector do
    | | similaritiesVector[ $p$ ]  $\leftarrow$  MajorityVoting(resultsVector[ $p$ ]);
    | end
    | return userID according to the highest score in similaritiesVector;
Function findPerson(vectorOfFeatures):
    | execute algorithm for each item  $i$  in vectorOfFeatures;
    | return vector with scores;

```

Algorithm 3.1: Person identification pseudo-code algorithm

where the *MajorityVoting* algorithm correspond to the one described in [103]. More in depth, the majority voting is executed on the entire list of measures and ambiguities can occur. In fact, two or more scores can have the same values. In this case, the sum of differences of each measure is performed and the lowest value is chosen. The probability of obtaining the same values is very low, however, if it happens again, the system treats the associated events as they are performed by all the retrieved persons. However, as soon as possible, the system could retries to identify the user, repeating the process from the beginning. The introduced method is partially used also in re-identification step. Always using RGB and depth data, we designed a combined method that increases the precision of the module. Concerning RGB, we refers to a work of Martinel's team [70] that exploits dense histogram features, warp functions and finally classifies with random forests. It is integrated with information retrieved by depth data, in particular, we used the same approach described for identification: the use of anthropometric measures. Also in this case, the conditions of appliance are the same, it means that some bounds

of optimal distances based on sensor performances are required. The combination decision is based on a priority-based method. Due to the fact that anthropometric measures retrieved by the proposed method are weaker than the RGB analysis for re-identifying a person, we suggest giving a different weight to each output. The first operation consists in transforming the scores obtained with majority voting procedure in percentage values. It can be done summing all the values and dividing each one of them by that sum. In this way, we obtain percentage values from both classifiers. Then, the weights are applied. Therefore, the final score is retrieved according to the following formula:

$$fs = \text{highestValue}([\text{percentageValue}(\text{similaritiesVector})] \times a', \text{randomForestPercentages} \times a'') \quad (3.18)$$

where $\text{highestValue}()$ function is providing the highest value among all the values of two vectors, comparing them by index, $\text{percentageValue}()$ is retrieving the scores in a vector in percentage (as previously described), $\text{similaritiesVector}$ and $\text{randomForestPercentages}$ are the vectors of results obtained over depth and RGB and finally the values a' and a'' are the assigned weights. In our specific case, they correspond to 80% for RGB and 20% for depth. These values are retrieved in experimental phase.

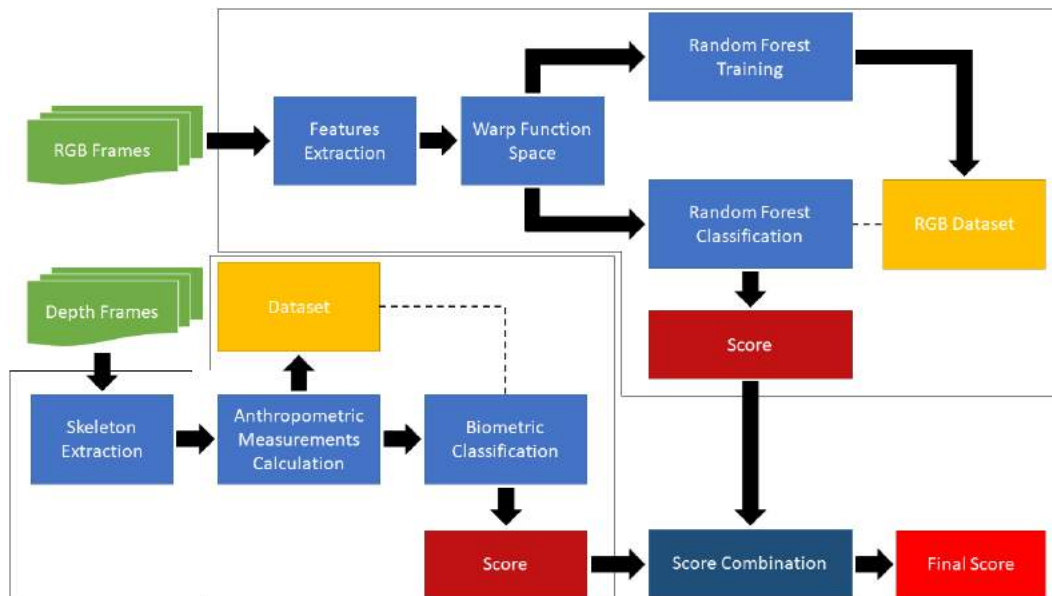


Figure 3.12. Re-identification module: logical architecture. The dataset block is used with temporary data registered during the first capture when the re-identification function is invoked. On the contrary, permanent information are stored in depth data dataset for the identification and the RGB frames are ignored.

Other modules

We provide some added modules in the framework, however they implement classical approaches. It means that they do not bring innovation to the state of the art. They

are only used for testing the completeness of the proposed framework and will be presented in experiment chapter of this document.

Chapter 4

Probabilistic temporal logic finite state machines

4.1 Temporal Logic Rules Management

The accuracy of a multimodal system is often related to each single mode, but some tunings can be applied at merge phase for improving performances. We are describing a method for enhancing disambiguation of events.

The temporal logic rules management module corresponds to the "Rules Checker" block in Figure 3.1. As previously mentioned, it is the core of the Rules Actuator and also the linker between input and output. It means that this module is the most important one of the entire framework.

In this chapter we explain the theory behind the proposed management method, underlining the defined grammar, the event correlation and the probability factors. In a common scenario, events are continuously occurring. The system should constantly listen to the sensors and keep track of the flowing time. When an event is recognized, a timestamp can be associated for defining an effective timeline. Considering single events in a monomodal environment, the interactions management is very simple. For each trigger, starting and ending time should be considered in order to chose if a condition is satisfied or not. In the same context, we can consider a multiple event case. There are two possible situations that can occur: when a single listener is involved and when multiple ones are. In the first scenario, each event can be separated or contiguous to another. The system can not recognize two overlapped events, even if the user interrupts an action for starting a new one. The second scenario instead manage the case of multiple events occurring simultaneously. It means that also overlaps are allowed. This scenario can be contextualized according to Allen's algebra. Before explaining the proposed method for managing events, we need to analyze this algebra grammar. According to Figure 3.2, 7 cases and relative negations can occur:

- **X before Y:** in this case the two events are completely independent. There are no links between them. It requires at least a minimum time range that separates the end of the first event from the beginning of the second.
- **X meet Y:** the events are contiguous. The end of the first event corresponds

to the beginning of the second. In this case, it is important to consider that the theoretical concept requires specific tunings to be correctly implemented in real cases.

- **X overlaps Y:** in this case the events are overlapping according to specific rules. The end of the first event should occur during the second event. The beginning of the second event should occur during the first event. None of the beginnings or the ends should occur contemporary to another begin or end.
- **X during Y:** the first event is occurring during the second one. It is important that the beginning of both starts at the same time and the end of the first event is occurring before the end of the second.
- **X starts Y:** the first event is occurring during the second one. In this case both end and beginning of the first event should occur during the second event.
- **X finishes Y:** the first event is occurring during the second one. The ends of both events are synchronized, but the beginnings are not.
- **X equal Y:** the first and the second event are occurring at the same time, with both beginnings and ends synchronized.

This grammar provides any possible combination between two events. There are numerous factors that should be considered when categorizing pairs of events according to this temporal logic. In fact, there could be ambiguities during both event recognition and temporal relation. The accuracy of the sensors derives from their quality and the method used for the designed aim. For example, a high quality camera could provide good quality images but if the algorithm used for completing a task is poorly designed, the overall performances could be low. The same could happen in the inverse situation.

In this theoretical example we are assuming that there are no delays among involved sensors, recognition methods and classifiers.

4.1.1 Event Representation and Management

The proposed framework is designed for managing almost every kind of event. It implies a generalization of the system for allowing full integration with modules and functionalities. The proposed logic is based on contexts and involved sensors. In fact, the only way to communicate with the system is to interact with an input device. We can assume that at least one sensor is always required when an event should be recognized. When two sensors collaborate to recognize the same event, it could be not necessary to consider the combination of single results. The proposed grammar is inspired by other multimodal spatio-temporal logic based frameworks [123, 15]. A rule is composed by two main sections: the condition and the actuation. If the condition is satisfied, the output is produced according to the actuation. The rules are designed following a specific semantic, designed for allowing the maximum degree of freedom to the user. The method is based on a dynamic vocabulary that is composed by sensors, events, persons, temporal operations, time intervals, actuators,

interactive objects and implications. Temporal correlations are a fixed number: the Allen's logic provide the entire range of possible combination between events. So, the complete grammar is composed by the following elements:

- s : is the sensor involved. It can be found only in a condition, the first part of a rule.
- a : is the actuator involved. It can be found only in an actuation, the second part of a rule.
- to : is the temporal operation. It can be found only in a condition, the first part of a rule.
- e : is the event.
- p : is the person involved. It can be found only in a condition, the first part of a rule.
- o : is an interactive object. It can assume multiple functions, like a target of a pointing action or a trigger when the user comes closer to it, etc.
- ti : is the time interval. It is an optional parameter related to temporal operations to , to the delay time between the satisfaction of a condition and the actuation of the output (the symbol \rightarrow) and, finally, to the actuators a .
- lo : is a logical operator. It can be an AND (\wedge) or an OR (\vee) operator. The AND can be correlated with a time interval.
- \rightarrow : implication that separates the condition from the actuation of a rule. It can be sided by a time interval ti .

In particular, $s \in \Sigma^{input}$ where Σ^{input} is the dictionary of available input sensors, $a \in \Sigma^{output}$ where Σ^{output} is the dictionary of output actuators and $o \in \Sigma^{object}$ where Σ^{object} is the dictionary of interactive objects. These three dictionaries are populated according to the settings of the Layout Builder. $to \in \Sigma^{Allen}$ where Σ^{Allen} is the Allen's logic dictionary, without negations, $e \in \Sigma^{events}$ where Σ^{events} is the dictionary of all possible events. The latter are divided according to sensors or actuators. Moreover Σ^{events} is populated according to possible events that the system is able to recognize. For example, it contains classes of instances for datasets of each Rules Actuator sub-module. $p \in \Sigma^{persons}$ where $\Sigma^{persons}$ includes all the persons to recognize or re-identify. In the first case, the information are permanently stored in a dataset. In the second, instead, user's data are only temporary memorized. Finally, $i \in \mathbb{N}$ due to the fact that a time interval must be not-negative. Some elements are composed by multiple attributes. In particular:

- The sensor s is composed by:
 - $inputPosition(x, y, z)$: the location in 3D space.
 - $inputOrientation(\alpha, \beta, \gamma)$: the orientation according to the three components of pitch (α), yaw (β) and roll (γ), on respective x , y and z axis.

- *inputType*: the sensor type, for allowing to dynamically classify it.
- *inputDomain*: it consists in the group of events that the sensor is able to recognize.
- The actuator a is composed by:
 - *outputPosition* (x, y, z) : the location in 3D space.
 - *outputOrientation* (α, β, γ) : the orientation according to the three components of pitch (α) , yaw (β) and roll (γ) , on respective x , y and z axis.
 - *outputType*: the actuator type, for allowing to dynamically classify it.
- The event e is composed by:
 - *timeInterval* event recognition time. It contains the starting and ending time of an event.

Some elements are not involved in the logical analysis of the proposed semantic. However, they are useful in the practical appliance of a rule in a real usage case. Concerning the semantic, it is defined according to logical dependences among all the taxonomic terms. The following bonds define them:

- The atomic elements of a rule are the sensor for the condition section and the actuator for the actuation.
- Each rule must have at least one sensor s in the condition, an implication \rightarrow and an actuator a in the actuation.
- Each rule can be linked to the others. This relation is used for the disambiguation of future recognitions.
- The sensor s must always recognize a single event e . It can be defined with the following structure: $s(e)$. The complexity of the event is related to the module designed for its recognition.
- The sensor s can involve also a person p and an object i . They can be defined according to the following structure: $s(e, p, t)$.
- A temporal operation to always involves two sensors s . It can be defined with the following structure: $to(s_1, s_2)$.
- Any temporal operation to can be associated to a time interval ti . It can be defined with the following structure: $to(s_1, s_2, ti)$.
- Only temporal operations to that are not specifying contemporaneity can be associated to time intervals ti .
- Between two temporal logic operations there is always a logical operator lo . It can be defined with the following structure: $to(s_1, s_2) \vee to(s_3, s_4)$. The logical operator \wedge can be correlated with a time interval that is expressed in

"windows". It means that it is possible to delay the check of the following events according to the needs. A wait time of two windows is defined with the following structure: \wedge_2 .




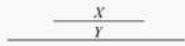
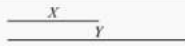
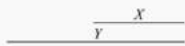
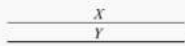
Relation name	Symbolism	Graphical representation
X before Y	X_bY	
X meets Y	X_mY	
X overlaps Y	X_oY	
X during Y	X_dY	
X starts Y	X_sY	
X finishes Y	X_fY	
X equal Y	X_eY	

Figure 4.1. Allen's temporal logic algebra and the symbolism applied in the proposed system.

The symbolism applied in the proposed system is shown in Figure 4.1. The following example shows how to manage a rule with three different sensors that should recognize three separated events. They are linked with two temporal logic operations, the first is referring to "X equal Y" and the second to "X meets Y". It is important to underline that both temporal operations can not be associated to a time interval due to the fact that a contemporaneity is required. On the contrary, if a "X before Y" is used, the time can be specified. Finally, an output is produced by a single actuator. It can be translated as:

$$to_{XeY}(s_1(e_1), s_2(e_2)) \wedge to_{XmY}(s_3(e_3)) \rightarrow a_1$$

More than two events are managed as if they are a sequence of linked pairs. The condition can be analyzed according to a sliding window of size 2. For example, in a condition composed by 3 events linked with 2 temporal logic relations we can have:

$$to_1(s_1(e_1), s_2(e_2)) \wedge to_2(s_3(e_3))$$

In this case, if we consider that the result of $to_1(s_1(e_1), s_2(e_2))$ can only be a boolean, true or false, we can say that at least a section of the condition could be satisfied. We are obtaining a partial result. In the same way, moving the windows to

the next couple of the group, we are considering $to_2(s_2(e_2), s_3(e_3))$ that will always provide a boolean result. So, the formula can be decomposed in:

$$\begin{aligned} b_1 &= to_1(s_1(e_1), s_2(e_2)) \\ b_2 &= to_2(s_2(e_2), s_3(e_3)) \end{aligned}$$

then, with a simple logical AND operation on b_1 and b_2 we obtain the complete result of the condition, that is always a boolean.

4.1.2 Probabilistic Finite State Machines

The output of each rule is generated only when a condition is satisfied. In this section of the document we explain details about the optimization and tuning of this process. As a first approach we introduce the theory of finite state machines [42]. We applied this theory to the proposed semantic for temporal logic rules development. According to literature, a deterministic finite state machine is composed by these elements:

$$(\Sigma, S, s_0, \delta, F) \tag{4.1}$$

where Σ is the input alphabet (a finite, non-empty set of symbols), S is a finite, non-empty set of states, s_0 is an initial state, an element of S , δ is the state-transition function ($\delta : S \times \Sigma \rightarrow S$) and F is the set of final states, a subset of S .

These elements can be associated to the ones that compose the temporal logic rules in the proposed method. In particular:

- Σ : is the alphabet related to the possible input that the rule can present, based on sensors and events that can be recognized.
- S : when a condition is satisfied we can assume that, according to the finite state machine theory, the related state is reached. Each event recognized by a specific sensor can be the trigger for switching state and approaching to the completion of the entire conditional section.
- s_0 : is represented by the starting point, in which no conditions are satisfied yet.
- δ : corresponds to the correlation created by the rule's conditional section itself. In fact, if the states are defined according to the conditions, an event (recognized exploiting a specific sensor) that satisfies the condition is the transition function that allows to move from a state to another one.
- F : is the completion of the conditional section. It corresponds to a single state f_0 in the proposed case.

From s_0 to the final state f_0 in F each intermediate state corresponds to a partial completion of the entire conditional state. It means that the structure of the automaton is a complete tree due to the fact that each trigger event contributes in a transition to reach the final state at any time. An example can be more explicative than a theoretical formalization. Consider this scenario: two sensors s_1 and s_2 that should recognize two events e_1 and e_2 and are correlated with a temporal operation to_1 . Then, we have other two sensors s_3 and s_4 that must recognize two events e_2 (this is the same event named before) and e_3 and are correlated according to a temporal operation to_2 . The two temporal operations are in AND. We can say that for completing the conditional section the entire following formula must be accomplished:

$$to_1(s_1(e_1), s_2(e_2)) \wedge to_2(s_3(e_2), s_4(e_3)) = TRUE \quad (4.2)$$

In this case, we can say that the states are 3, one for each event to be recognized. For correctly managing the temporal operations, a new middle state is introduced. If, for example, e_1 and e_2 are recognized but the correlation to_1 is wrong, it is not true that both events are discarded because e_2 is also involved in the second correlation to_2 . So, the conditional section can still be satisfied, depending on the value of to_1 . In fact, if to_1 is "X before Y", the event e_2 can satisfy the condition only during the following iteration. This specific case can be considered a "middle satisfaction state" due to a partial completion of the condition.

A new factor is introduced due to spatio-temporal events: the temporization. The timed automata theory [5] can provide the correct approach to manage this crucial factor. We must consider that the events are associated to the temporal window in which they occur. It means that for satisfying the conditions it could be necessary to check the involved window and the adjacent ones. Moreover, there is a validity time of the recognized event. It is correlated to the temporal operation, if one is engaged. In Table 4.1 the link between temporal windows, event validity and Allen's temporal operations is shown.

Table 4.1. Correlation between temporal windows, event validity and Allen's temporal operations. In particular the Time-To-Live (TTL) of the events is expressed in iterations, that corresponds to a temporal window. In some cases is impossible to define the sliding windows and the TTL of an event a priori, due to the fact that the temporal operation can involve multiple contiguous windows.

Temporal Logic Operation	Temporal Windows involved while recognizing the second event	First Event TTL (expressed in iterations)
XbY	Previous (based on the time interval), Current	based on the time interval
XmY	Previous, Current	2
XoY	Previous, Current	2
XdY	Current, Following	2
XsY	Previous (based on the time interval), Current	based on the time interval
XfY	Previous (based on the time interval), Current	based on the time interval
XeY	Current	1

We denote that the events' Time-To-Live (TTL) is exclusively related to the correlated temporal operation. In particular, when none of them is involved, the trigger is always executed without any other additional condition. On the contrary, when an operation in which a variable time interval (based on the set range) is required, the recognized events should be available since the end of the time set. Concerning the

interval, it is defined according to the homonym parameter plus the event duration. Using the timer for automaton we can keep an optimized history of passed events. It is extremely important in continuous event recognition due to the fact that the challenge is higher than with discrete models [3].

With the proposed method events and operations can be easily managed. However, some ambiguities can occur. The improvement we adopted is related to the probability while multiple rules are involved. We can say that each event is linked to the recognition probability $P(e)$ provided by the assigned module for the task. Considering that the events correspond to the states of the machine, we can say that the automaton can assume probabilistic functions. So, we can use Markov Chains [58] instead of classical finite state machines. They are used principally in probabilistic calculation and statistics, however can be applied to numerous contexts. Markov chains are similar to finite state automaton but also provide a probability value to the transition function.

A Markov chain is a sequence X_0, X_1, X_2, \dots or arbitrary variables that satisfy the rule of conditional independence. The latter is called Markov property and is described according to the following probability formula:

$$P(X_n = i_n | X_{n-1} = i_{n-1}) = \quad (4.3)$$

$$= P(X_n = i_n | X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}) \quad (4.4)$$

where $n \in \mathbb{Z}$ and i_0, i_1, \dots are possible states of the variables. It requires information of previous state probability for calculating the new ones. Moreover, for keeping track of this, the Markov chains exploit some matrices called transition matrices. A transition matrix P_t of a Markov chain X at a certain time t contains information about the probability of transitioning between states. In particular, an element of that matrix P_t is defined according to the following formula:

$$(P_t)_{i,j} = P(X_{t+1} = j | X_t = i) \quad (4.5)$$

where i and j are the respective row and column positions. In this way, the rows of the matrix correspond to probability vectors and the sum of their entries is always 1. With this structure it is possible to describe each probability related to the condition of a rule. We can outline the system in this way:

- Each row corresponds to a rule condition section. It means that the number of rows i is equal to the number of rules r .
- Each element corresponds to a probability that a specific rule is involved at a certain column. The columns are related to events. It means that the number of columns j is equal to the longest condition section among all the rules r .

So, we can say that the generated transition matrix can be read from left to right to obtain a parallelism with rules' events, according to the proposed grammar. Each column position is overlapped with the related event of a certain rule. Moreover, the transition probabilities are calculated on the number of shared events between rules and temporal logic operations involved. However, some shrewdnesses must be

applied.

Let us consider the following 3 rules conditional sections:

$$to_{X_m Y}(s_1(e_1), s_2(e_2)) \wedge to_{X_m Y}(s_3(e_3), s_4(e_4)) \wedge to_{X_d Y}(s_5(e_5), s_6(e_6)) \quad (4.6)$$

$$to_{X_e Y}(s_3(e_3), s_2(e_1)) \wedge to_{X_f Y}(s_3(e_3), s_2(e_2)) \quad (4.7)$$

$$to_{X_m Y}(s_1(e_1), s_4(e_4)) \wedge_1 s_5(e_5) \quad (4.8)$$

Allen's chronological ordering is based on the ending point of each event. It means that the first element of the two involved in a temporal operation could occur after the beginning of the second one. So, the initial step consists in chronologically ordering the beginning of each pair of events defined by a temporal operation. The relations which, according to that, are not respecting the chronological ordering are two: "X starts Y" and "X finishes Y". Both require the starting point of X after the one of Y. This preliminary step is necessary for creating the transition matrix and treating the conditional section of a rule according to the proposed algorithm 4.1.

Function firstTransitionMatrixUpdateEasy():

```

Initialize variables;
foreach chronological window w do
  |  $e_{score}$  = # of occurrences of each event  $e$ ;
end
foreach chronological window w do
  | foreach condition c in Rules[ ] do
  | |  $score_{cw}$  = sum the scores  $e_{score}$  of each event in the slot  $w$ 
  | | involved in  $c$ ;
  | end
end
 $totalOccurrencesScore$  = sum of each  $score_{cw}$  ;
create the transition matrix  $TM[ ][ ]$  where the # of rows =  $r$  and the #
of columns =  $w$ . Each cell contains the result of
 $P_{cw} = score_{cw} / totalOccurrencesScore$  for each condition and window ;
return  $TM[ ][ ]$  ;

```

Algorithm 4.1: Probability assignment for the first transition matrix. It is important to consider that each delay, named timer interval ti , can generate time shifts of events.

It is based on three main steps. The first one consists in counting the occurrences of all events e for each time slice w (temporal window). The occurrences are the "scores" of the events. The second step consists in associating events and rules. For each rule, we sum the score of events based on their occurrences in the rule for each temporal window. At the end of this process, the sum of all obtained scores is calculated, providing the total occurrences score. Finally, in the third step, the transition matrix is calculated. The cells are filled according to the proportion between scores obtained

for each rule and temporal window and the total occurrences score. In this way, each row represents a rule and each column a temporal window. The cells contains the probability that a rule is involved at a certain temporal window.

We can provide an example of the algorithm execution on the previously mentioned sample conditions (4.6) (4.7) (4.8). In Figure 4.2 their schematization is shown.

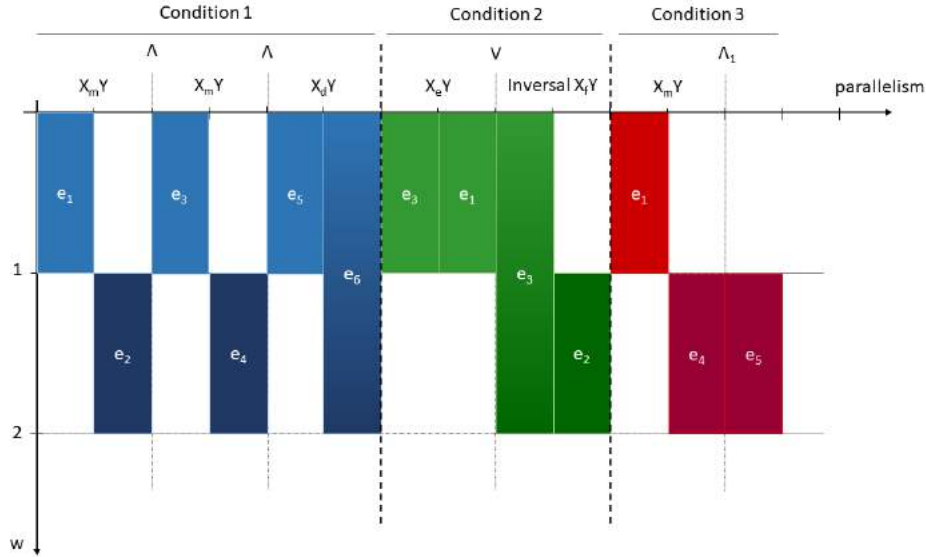


Figure 4.2. Temporal scheme of possible events that can occur in the following conditions: Condition 1 (4.6), Condition 2 (4.7) and Condition 3 (4.8). The vertical axis corresponds to the temporal intervals divided in windows (w). The horizontal axis denotes the parallelism of each potential event.

It is graphically describing how the events are managed, according to their possible occurrence. The vertical axis indicates the temporal windows w , so the "timeline" of the system. On the horizontal axis the possible events that can occur at each temporal slice are placed. As shown, all the AND and OR operations generate parallelism of the involved temporal logic relations or events, excluding the \wedge_1 in the third condition that is indicating a time interval ti shift of 1 window. The $X_f Y$ has been reversed due to the preliminary step of the algorithm.

Starting from this scheme, we can count the occurrences of each event. For an easier analysis, we consider and complete the first temporal window before proceeding with the second. We can see that the following occurrences for the first interval can be identified:

- $e_1 = 3$;
- $e_3 = 3$;
- $e_5 = 1$;
- $e_6 = 1$;
- e_2 and e_4 can't occur in the first time window;

Table 4.2. Transition matrix of conditions (4.6) (4.7) and (4.8) generated by the proposed algorithm.

	w1	w2
Condition 1	0.44	0.46
Condition 2	0.39	0.27
Condition 3	0.17	0.27

These values are summed according to the number of occurrences of the events in each condition. The scores are calculated as follows:

- $score_{c1w1} = 3 + 3 + 1 + 1 = 8$;
- $score_{c2w1} = 3 + 3 + 1 = 7$;
- $score_{c3w1} = 3$;

These values are the weights of each condition, used for the probability calculation. The sum of them is 18. This value is used for generating the proportions and consequently the probabilities as follows:

- $P_{c1w1} = 8/18 \approx 0.44$;
- $P_{c2w1} = 7/18 \approx 0.39$;
- $P_{c3w1} = 3/18 \approx 0.17$;

So, the first column of the transition matrix is composed by these values. It means that, the first condition has the highest probability to be involved during the first temporal window. Repeating the process for the second time slice, we obtain the following results:

- $P_{c1w2} = 5/11 \approx 0.45$;
- $P_{c2w2} = 3/11 \approx 0.27$;
- $P_{c3w2} = 3/11 \approx 0.27$;

The transition matrix, when no event has occurred yet, is shown in Table 4.2. The second step of the method provides a selective update of the transition matrix due to the fact that one or more events could occur. The algorithm 4.2 shows how the transition matrix is updated according to an event e_i recognized in a certain temporal window w_i .

Function nextTransitionMatrixUpdateEasy(w_i previous temporal window, $Events[]$ recognized events):

```

Initialize variables;
InvolvedEvents [] ← initialize ;
foreach chronological window  $w$  next to  $w_i$  do
  foreach condition  $c$  in Rules[] do
    foreach event  $e_i$  in Events[] do
      if the temporal operations to in  $c$  contains  $e_i$  on their first
      member then
        increment occurrence counter of the event  $e$  that is the
        second member of the temporal operation  $to$ ;
        add  $e$  to InvolvedEvents [] ;
      end
    end
  end
end
foreach event  $e$  in InvolvedEvents [] do
   $score_e$  = occurrences of  $e$  / the total occurrences ;
end
foreach condition  $c$  in Rules[] do
  if  $a$  to has an event from Events[] as first member and an event
  from InvolvedEvents[] as second member then
    weight  $w_c = w_c + score_e$ , where  $e$  is the second member of the
    temporal operation  $to$  ;
  end
end
update the value of each row  $r$  with the average value between the
current probability and the weight  $w_c$ , where  $j$  correspond to the row.
If for a row there is no modification weight, the weight is considered 0 ;
return  $TM[ ][]$  ;

```

Algorithm 4.2: Probability update for transition matrix. It is important to consider that each delay, named timer interval ti , can generate time shifts of events.

It consists in a method that increments or decrements the probability of future possible events based on history. We can summarize the process as follows:

- Among all the possible temporal operations to in each rule condition, the ones which have the recognized event e as first operator are selected;
- The occurrences of each event v as second operator of each involved to are calculated;
- A weight of each event v is calculated according to the ratio between its occurrences and the total occurrences of all events;
- The weights are associated to each rule condition summing them according to the occurrences of events v in the temporal window w_i ;
- The transition matrix is updated according to the average value between weight calculated for each condition rule and the current probability value for the

Table 4.3. Transition matrix of conditions (4.6) (4.7) and (4.8) updated by the proposed algorithm.

	w1	w2
Condition 1	0.44	0.475
Condition 2	0.39	0.135
Condition 3	0.17	0.385

temporal window w_i . If there is no weight for a certain row, the weight is considered 0 (and the average is normally calculated).

If we apply this process to the previous example, we should consider a trigger condition: let's say that the system recognizes the event e_1 in the first temporal window. From this starting state, we can look for every temporal logic operation lo that involves e_1 as first operator. There are one match in (4.6) and one in (4.8). The second operator of these temporal logic operations are respectively e_2 and e_4 and the total occurrences of events is 2. So, we can say that the weigh associated to each one of the events is 0.5 due to the fact that we consider the ratio between the single occurrences and the total ones. Finally, we can calculate how many times the temporal logic operations involve e_1 as first operator and e_2 and e_4 as second operators. We can see that rules 1 and 3 present in their conditions only 1 occurrence of the proposed cases. So, in temporal window 2, the weights of first and third row of the transition matrix are modified according to a weight of 0.5. Due to the fact that we have to proportionally modify the transition matrix, the second row will be updated with a weight of 0, because $0.5 + 0.5 + 0 = 1$. The final transition matrix, after the average between the current values on column 2 and the weights, is shown in Table 4.3.

The algorithm 4.2 can be tuned with an important additional information: the relation between rules. Let's consider that a condition is satisfied. When it happens, the transition matrix is partially or totally re-initialized, bringing the state of the system to its starting phase (for that rule). However, it could be possible that, after completing a task, a user would more probably execute a certain action instead of others. For example, someone who is asking "What is that?" pointing to an object would probably approach it during the system's answer. We can simulate a human intuition thanks to this method and the proposed associated grammar. In fact, if a logical connection is specified during the rule creation, the probabilities should be modified for taking care of it. In particular, if a condition of a rule is satisfied we could increase the probability of the other linked rules. A weight factor is required. The algorithm 4.3 is showing how this parameter is calculated and applied to each probability of the rules. It is important to notice that the links could be more than one.

Table 4.4. Transition matrix of conditions (4.6) (4.7) and (4.8) updated according to the algorithm 4.3 if (4.6) and (4.8) are linked and (4.6) is satisfied.

	w1	w2
Condition 1	0.37	0.405
Condition 2	0.33	0.133
Condition 3	0.23	0.515

Function nextRulesTransitionMatrixUpdate(*actuated rule r*, *transition matrix TM* [][]):

```

Initialize variables;
weightFactor = numberofruleslinkedtor/numberofrules ;
distributedWeight = weightFactor/numberofrulesnotlinkedtor ;
foreach rule s in Rules [ ] do
    foreach event e in condition c in s do
        if s is linked to r then
            update probability of e in TM [ ][ ] adding weightFactor
            percent value to it ;
        else
            update probability of e in TM [ ][ ] subtracting
            distributedWeight percent value to it ;
        end
    end
end
return TM [ ][ ] ;

```

Algorithm 4.3: Probability update for transition matrix after a rule condition is satisfied.

We can make an example starting from the previous one. If we set a link between (4.6) and (4.8) and the first condition is satisfied we can apply the proposed update algorithm. In this case the rules are 3 and there is only one link, so the *weightFactor* is $1/3 \approx 0.33$. The not involved rules are the first and the second, so we can say that the *distributedWeight* value is $weightFactor/2 \approx 0.16$. Then, for each event, we can calculate the new probability modifying its value with the corresponding percentage value. For example, the event e_1 in the first temporal window and the first condition (row 1, column 1) probability is updated subtracting the *distributedWeight* percentage value because the first rule is not linked with itself. So, the result is $0.44 - 16\%$ of $0.44 \approx 0.44 - 0,07 \approx 0,3$. The probability of the event e_1 of the first temporal window and the third condition (row 3, column 1) is updated summing the *weightFactor* percentage value to it because the rule is linked to the satisfied one. So, the result is $0.17 + 33\%$ of $0.17 \approx 0.17 + 0.06 \approx 0.23$. Updating all the values according to the links, we obtain the new transition matrix shown in table 4.4.

However, in the proposed example the approximation factor is decreasing the proportionality between probabilities, so the sum of each column could be not 1.

The proposed conditions management method is designed also for compensating the inaccuracy of sensors and their related modules. We can consider that some operations can be more confident than others: a gesture recognized with a camera is less precise than a proximity sensor that should detect if a certain threshold is reached or not. The proposed approach supports the predictive function of the framework. Finally we can say that this method is perfectly integrable with additional features, such as semantical correlations [29] or other techniques. Events that are not expressed in the rules conditions are not recognized at all.

4.1.3 Events Reinforcement and Final Probability Update

As mentioned above, the generated transition matrix can be used for modifying the decision of the system when some events can be mistaken. In fact, the proposed method is useful when there is a probability value related to each potential event. We are supposing that each sensor accuracy is below 100%, like in real case. It means that the system is not sure about the result of its detection and this is the reason why a recognition probability could be assigned to each event. The most representative example could be provided by a gesture recognizer: if the user performs an ambiguous movement, some elements could be associated to an event and some other to another one, causing indecision to the system. We can create a transition matrix that contains an event per row and a temporal window per column. In particular, each row will contain the occurrences of that event in the relative temporal window divided by the total number of events in the same slot. So, after the generation of this transition matrix and the detection of an incoming event in a certain temporal window, the recognition is performed and the following technique can be applied. The method is based on maintaining proportions between the transition matrix and the probability recognition after detection. The algorithm 4.4 is showing this update process.

```

Function ruleEventsProbabilityUpdate(events and associated
probabilities  $P_{events}[]$ , temporal window  $w_i$ , transition matrix  $TM_j[][]$ ):
  Initialize variables;
   $NewProbabilities_w \leftarrow$  initialization ;
   $Modifiers[] \leftarrow$  values of  $TM_j[][w_i]$  ;
  foreach probability  $p$  in  $P_{events}[]$  do
    if  $Modifiers[]$  contains event  $e$  associated to  $p$  then
       $mod \leftarrow$  probability in  $Modifiers[]$  corresponding to  $e$  ;
       $modifier_e = p + \{[(100 - p) / 100] * mod\}$  ;
      add  $modifier_e$  to  $NewProbabilities_w$  ;
    end
  return  $NewProbabilities_w[]$  ;

```

Algorithm 4.4: Potential events probability update based on transition matrix.

The main idea consists in adding a proportional amount of probability score. It is performed calculating the difference between 100 and the probability score, obtaining the remaining percentage value. Then, its percentage is calculated according to the transition matrix corresponding value. Finally the probability is updated with

the added value. In this way we can reinforce the results with a low probability rate with respect to the highest ones according to the expectations. An example could clarify this aspect. Let's consider the previous mentioned rules (4.6), (4.7) and (4.8). The relative first generated transition matrix is shown in table 4.2. If e_1 and e_3 , that can be both recognized by s_1 , could be easily mistaken due to possible similarities. Let's consider that the related module, after recognizing the occurring event, provides some percentage value of probability of matching with the events in dataset. Let's consider that the following values are provided:

- $e_1 = 83\%$
- $e_3 = 87\%$
- $e_4 = 81\%$

if we apply the algorithm 4.4 considering w_i the current temporal window, we obtain the following new values:

- $e_1 = 83\% + (37.5\% \text{ of } 17) \approx 83 + 6.37 = 89.37\%$
- $e_3 = 87\% + (37.5\% \text{ of } 13) \approx 87 + 4.87 = 91.87\%$
- $e_4 = 81\%$

In this case, due to the fact that the occurrences of e_1 and e_3 are the same, the proportion is respected and the result is unchanged. However the results could drastically change if the number of occurrences of an event is higher than the others. The event e_4 is not involved in the first temporal window so there is no modification of his probability. Then, for improving the results we can use the transition matrix related to the rules. In fact, it can provide a second reinforcement of the event based on condition's probability. The procedure is similar: for each event, if present in a condition, we calculate the difference between 100 and the current percentage value, then we calculate the percentage of it based on the value of the condition in the rules' transition matrix and finally this value is summed to the current percentage value of the event. The algorithm 4.5 corresponds to a modified version of 4.4 that involves the rules' transition matrix.

Function ruleEventsProbabilityUpdate(*temporal window* w_i , *transition matrix* $TM[[]]$, *events and associated involved probabilities* $P_{events}[]$):

```

Initialize variables;
NewProbabilitieswr ← initialization ;
Modifiers[] ← values of  $TM[[]][w_i]$  ;
foreach probability  $p$  in  $P_{events}[]$  do
    if  $TM[w_i][[]]$  contains event  $e$  associated to  $p$  then
         $mod$  ← probability in  $TM[w_i][x]$  where  $x$  corresponds to the
            index of  $e$  in  $TM[[]][[]]$  ;
         $modifier_e = p + \{[(100 - p) / 100] * mod\}$  ;
        add  $modifier_e$  to  $NewProbabilities_{wr}$  ;
    end
return  $NewProbabilities_{wr}[]$  ;

```

Algorithm 4.5: Potential events probability update based on transition matrix.

Applying the algorithm on the previous example, we can say that event e_1 in the first temporal window is involved in condition 1, 2 and 3. Then, we can find e_3 in condition 1 and 2 in the same slot. For each one of them we can apply the algorithm and we obtain the following percentage value update:

- e_1 in Condition 1 $\approx 89.37\% + (44\% \text{ of } 0.10) = 89.37\% + 4.4\% = 93.77\%$
- e_1 in Condition 2 $\approx 89.37\% + (39\% \text{ of } 0.10) = 89.37\% + 3.9\% = 93.27\%$
- e_1 in Condition 3 $\approx 89.37\% + (17\% \text{ of } 0.10) = 89.37\% + 1.7\% = 92.07\%$
- e_3 in Condition 1 $\approx 91.87\% + (44\% \text{ of } 0.08) = 91.87\% + 3.5\% = 95.37\%$
- e_3 in Condition 2 $\approx 91.87\% + (39\% \text{ of } 0.08) = 91.87\% + 3.1\% = 94.87\%$

Then we chose the highest percentage value of each event. The process is analogue for each temporal window.

Chapter 5

Experiments and results

In this section we provide experimental scenarios and results, focusing on each proposed module and the overall framework effectiveness. The entire structure is so dynamic that could be used in almost any application area, so we tested it in a rehabilitation, a security and a generic interactive environment. We provide a sample scenario in a museum that could provide a fully functional overview of the framework. Then, we describe the experimental setup and environments for each section of the framework. Finally, we performed evaluations based on subjective and objective results. The tests are mainly executed on self evaluating effectiveness of the framework, however some comparisons are also performed according to similar systems in same and different application areas.

5.1 Sample Scenario: Interactive Museum

This example provides an application scenario that clarifies the interaction between the users and the system in a real scenario. A museum director wants to improve its structure using new technologies. The proposed system can satisfy this requirement. A developer is chosen for configuring it. Then, a team of psychologists and historians trained by the developer on the possibilities of the system decides which kind of actions would be performed inside two interactive rooms of the museum, also designing the system's responses and the relations between events. When the entire operative plan is defined, the developer decides where to place sensors and actuators for providing the required result. If necessary, the involved modules are trained with sample data of specific actions, for example a series of gestures captured by a depth camera. Finally, the system is tested and eventually tuned for providing the best result. The development phase is over. The system is ready to be used with the users, that is to say the museum's visitors. Now, we analyze the scene from their point of view. A tourist enters the first room and the lights around each find are turned on. Then, a video on a monitor is played: an assistant, "Giorgia", appears and offers her help to the visitor. She also underlines which kind of actions could be performed, showing a mini-tutorial. After that, she asks him his name and the visitor answers "Paolo". Then, Paolo, moving towards the monitor, asks information of a find, pointing at it and saying "What is that?". After some instants, Giorgia starts speaking about the targeted item. Then, Paolo gets closer to it and the lights

of the other finds in the room start to lose intensity. Then, Paolo tries to touch it and Giorgia drastically stops speaking for saying "Please, don't touch anything in the room". Then, Paolo says "Sorry" and moves towards another item, while the lights of the room turn completely on again. Giorgia starts speaking about the find that Paolo is looking. Then, he says "Ok, thanks Giorgia." and uses his right arm for performing a "Stop" gesture. Giorgia stops speaking. Then Paolo goes out from the room and enters the following one. While executing this action, the lights in the first room are turned off and the ones in the second are turned on. Giorgia appears on another monitor and says "Welcome, Paolo, did you enjoy our first room?". Paolo answers "Yes" and asks "Can you please tell me where the bathroom is?". Giorgia starts explaining where to go while the lights in the room become brighter around the exit door. Paolo says "Thanks", goes out and comes back in the room while Giorgia says "Welcome back, Paolo. Would you like to take a tour of the entire room? You have seen nothing till now.". Paolo says "Yes." and the programmed tour starts, lighting the finds while Giorgia explains everything. At a certain point, Paolo stops Giorgia because she has said something related to a find in the first room and he is not sure to have noticed it. So, he goes back and Giorgia, appearing again on the screen says "Welcome back in the first room, Paolo.". Then Paolo asks "Hey, Giorgia, which finds do I have seen right now in this room?" and Giorgia lights them. So, Paolo can see that the focus item is not lighted and he asks information about it to Giorgia. Finally he says "Thanks" and goes out from both rooms while Giorgia says "Goodbye, Paolo".

5.2 Physical Architecture of the Framework

This section of the document describes the physical setup used for testing the framework. The hardware involved comprehends a CPU Intel Core i7-5930k, an Asus Rampage V Extreme motherboard (that provides 12 USB on back panel, useful when numerous sensors are needed), a GPU NVidia GTX 1070, 16 GB DDR4 RAM memory and a Samsung 850 Pro SSD of 256GB. The machine is a high-end desktop due to the requirements: multiple threads and numerous sensors involved at the same time.

5.2.1 Layout Builder

The layout builder is based on a 3D environmental editor. Three.js ¹ is a WebGL library that allows to do that. It is based on javascript language and is very versatile due to its compatibility with all the most common web browsers. This is one of the most important reasons why it was selected among the possible choices. Starting from an existing project ², the layout builder has been integrated with models generated with Blender ³ and some important functions have been introduced. In Figure 5.1 an example of 2D layout building phase is shown. In this step, the walls and the proportions are inserted.

¹<https://threejs.org/>

²<https://github.com/furnishup/blueprint3d>

³<https://www.blender.org/>

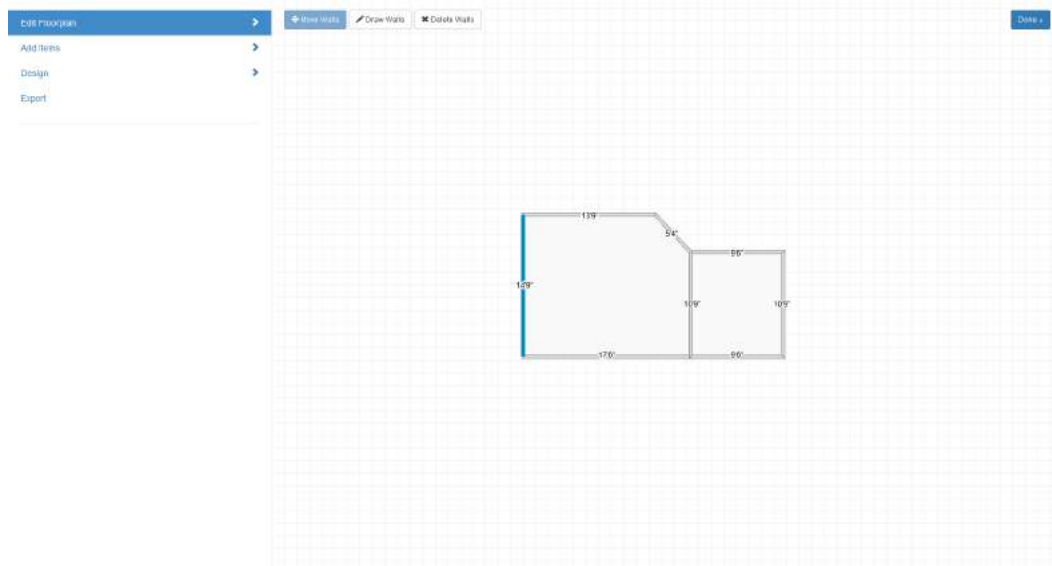


Figure 5.1. Layout Builder: planimetry designer.

The items management is shown in Figure 5.2. The environment is displayed in 3D space.

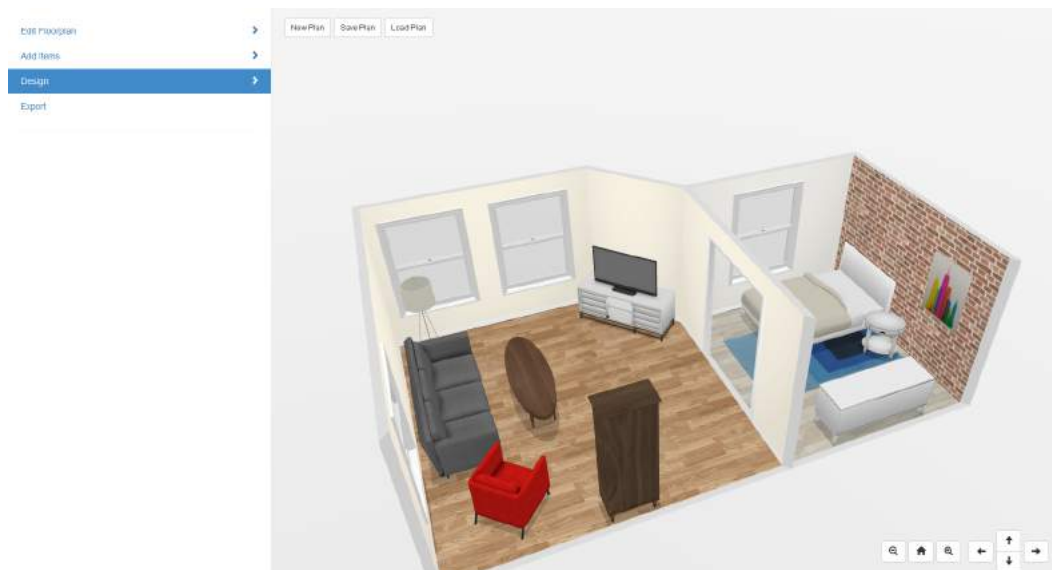


Figure 5.2. Layout Builder: overview of 3D editor for managing items in the scene.

Each item's dimension, position and orientation can be modified according to the set boundaries: collisions and ground object/wall object. The last named function limits the position of an item on the ground or on the wall, for avoiding to wrongly place it. An example of 3D editor for the layout builder is shown in Figure 5.3.

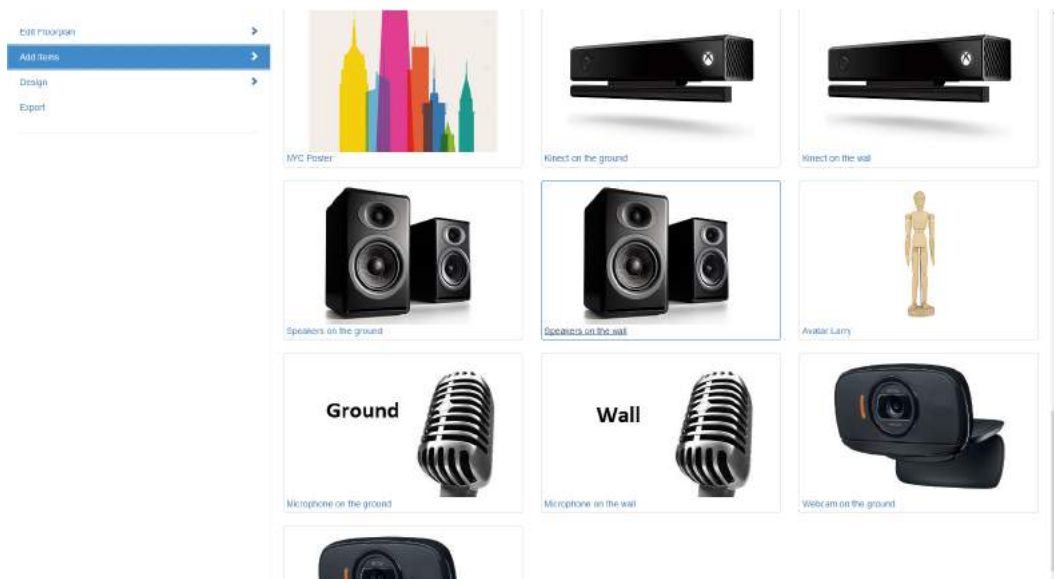


Figure 5.5. Layout Builder: addable items list. The modularity of the system allows to create new items in any moment according to the needs.

Finally, after deploying the environment, the system can export all the information. The Figure 5.6 shows the output file that can be parsed by the Rules Builder. This process promotes the modularity of the framework.

```

1 | {"location": {"x": 100, "y": 100, "z": 100}, "name": "MVC Poster", "type": "Image", "width": 100, "height": 100, "color": "#FFFFFF"},
2 | {"location": {"x": 200, "y": 200, "z": 200}, "name": "Kinect on the ground", "type": "Kinect", "width": 100, "height": 100, "color": "#000000"},
3 | {"location": {"x": 300, "y": 300, "z": 300}, "name": "Kinect on the wall", "type": "Kinect", "width": 100, "height": 100, "color": "#000000"},
4 | {"location": {"x": 400, "y": 400, "z": 400}, "name": "Speakers on the ground", "type": "Speaker", "width": 100, "height": 100, "color": "#000000"},
5 | {"location": {"x": 500, "y": 500, "z": 500}, "name": "Speakers on the wall", "type": "Speaker", "width": 100, "height": 100, "color": "#000000"},
6 | {"location": {"x": 600, "y": 600, "z": 600}, "name": "Avatar Larry", "type": "Avatar", "width": 100, "height": 100, "color": "#FFD700"},
7 | {"location": {"x": 700, "y": 700, "z": 700}, "name": "Microphone on the ground", "type": "Microphone", "width": 100, "height": 100, "color": "#000000"},
8 | {"location": {"x": 800, "y": 800, "z": 800}, "name": "Microphone on the wall", "type": "Microphone", "width": 100, "height": 100, "color": "#000000"},
9 | {"location": {"x": 900, "y": 900, "z": 900}, "name": "Webcam on the ground", "type": "Webcam", "width": 100, "height": 100, "color": "#000000"},
10 | {"location": {"x": 900, "y": 900, "z": 900}, "name": "Webcam on the wall", "type": "Webcam", "width": 100, "height": 100, "color": "#000000"}

```

Figure 5.6. Layout Builder: output data file. Exported information are related to walls and items, with all their variables.

5.2.2 Rules Builder

This second system is developed in Java. It parses the file produced by Layout Builder and shows a front view of the environment, underlining the name of each item in a unique form. It allows to avoid ambiguity. The user can define the rules following the previously mentioned syntax. There is no length limit. If the format is wrong, the system notifies the error and does not allow to proceed to save the rules file. Concerning connection between items, the system allows to select which rules should be linked to the one that the user is writing. Besides allowing to use the Allen's temporal logic form, it also provides a list of preregistered users that could be involved in specified events. Moreover, if a parametric form is introduced, such as

"person1" or "p1", the system will call the reidentification feature. If a rule presents a variable with the same name of a previously encountered one, a reidentified user is needed for that event. This operation and all the others related to the rules are performed by the Rules Actuator.

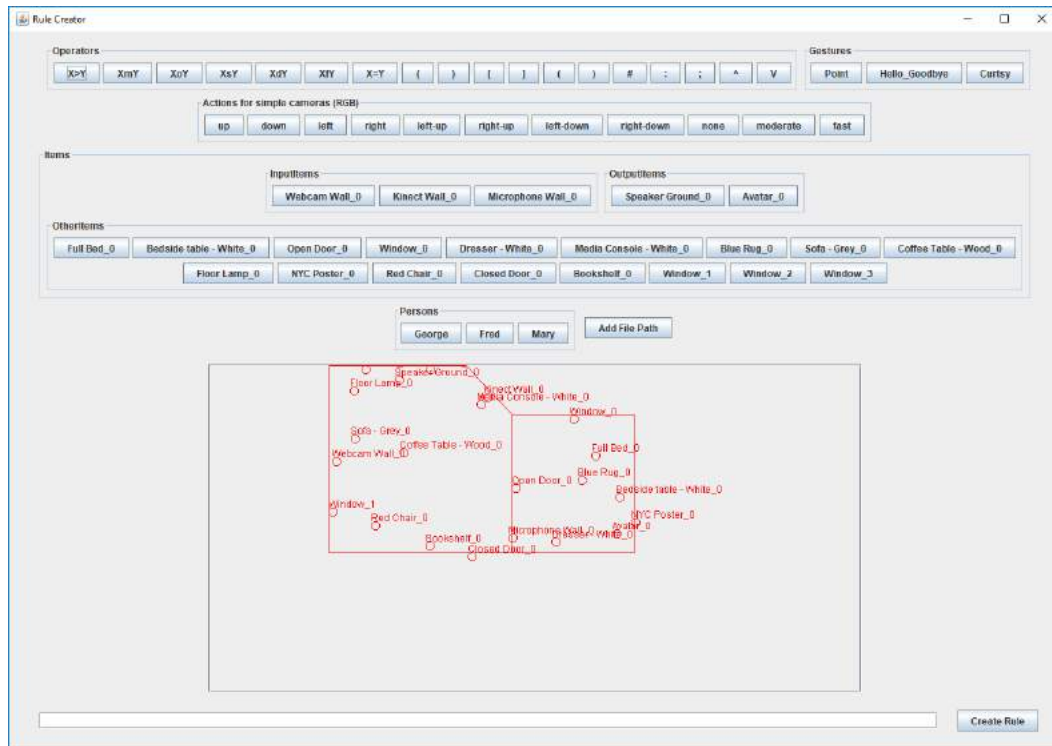


Figure 5.7. Rules Builder: GUI of the system. The user should follow the syntax proposed on the top of the image for creating rules. The buttons are dynamically generated based on the Layout Builder output and previously registered users. For using the reidentification features, the person can be parametrically specified.

In Figure 5.7 the GUI of the Rules Builder is shown. After completing to write a rule, the user can store it. At the really end of the process he/she can save the rules on a file that comprehends all the information, also the environmental one. This file is finally loaded by the Rules Actuator for starting the on-line phase.

5.2.3 Rules Actuator

The system is developed in C# due to a compatibility factor. In fact, for some submodules, we used the Microsoft Kinect for XBOX One sensor [69] and a Microsoft Windows powered computer is suggested. A WPF application is deployed starting from a powerful software named Vitruvius⁴. It allows us to easily interact with Microsoft Kinect V2 SDK. Moreover, we integrate the system with compatible libraries, such as Microsoft Cognitive Toolkit [107] (CNTK) and Aforge.net [60] for adding machine learning functions. The main computer vision library involved in

⁴<https://vitruviuskinect.com/>

camera-based functions is EMGUCV ⁵, a wrapper for OpenCV [18] library, that is fully compatible with C# programming language. In particular, we exploited version 3.2.0.2682 with CUDA. The gesture recognizer is managed with CNTK ⁶. It is a Microsoft open source library and allows to introduce some modifications to its internal functions.

For each module we used different sensor types. The involved ones are:

- **2x Microsoft Kinect V2:** used by gesture recognition, re-identification and identification modules;
- **2x MB1414 USB-ProxSonar-EZ1:** used in a simple proximity calculator module;
- **2x Webcam Microsoft: LifeCam HD 3000:** used by motion detection module;
- **2x USB Sound Tech CM-1000USB omni-directional microphones:** used by speech recognition module.

However, this system is only a container that should implements a method for checking if the rules' conditions are satisfied or not. It means that it is possible to integrate external modules just considering them like black boxes. The only needed inputs consist in percentage values of every possible event for each sensor. It allows distributed computation and also the use of networks when long distances are involved.

Added Module: Pedestrian Tracker and Motion Detector

The framework involves the previously described pointing calculation, gesture recognition, re-identification and identification modules. However, for experimental purposes, we inserted some more added modules that allow to integrate the framework with more functionalities. In particular, we used some methods that are well known in literature. We used two RGB cameras and those devices could be easily exploited for identifying pedestrian and tracking them. We combined HOG and Linear SVM [33] for detecting if human figures are present in the scene. OpenCV implements a pre-trained detector that, opportunely tuned, can provide the required functionality. Then, the barycenter of the bounding box around the pedestrian is tracked, for evaluating trajectory. The latter are simply classified in 4 directions, one for each cardinal point, where up and down are referring to the distance from the sensor. More in detail, if the trajectory of a pedestrian is going from left to right, the user is moving to the right of the scene in front of the sensor. It is analogue for the left. Up and down, instead, means that the user is going farther or closer to the sensor, because the camera's view direction is parallel to the ground. We also provide a relatively simpler function related to cameras: a motion detector. It exploits DBSCAN [34] and optical flow [49] for identifying the dimension of the elements that are moving inside the scene. The sum of the area of each cluster

⁵http://www.emgu.com/wiki/index.php/Main_Page

⁶<https://docs.microsoft.com/en-us/cognitive-toolkit/>

determines the motion intensity. For this algorithm we used a limited number of frames for each temporal window due to the computational weight.

Also in these cases each output class is associated to a percentage value according to the following scheme:

- Pedestrian trajectory: each movement of the user is expressed in quantity of translation in each of the four directions and a proportion is generated. If a user is going 250 pixels "right" and 100 pixels "up", the percentage values are calculated according to the following proportions:

$$x_{right} = (100 \times 250) / (250 + 100) \approx 71,42 \quad (5.1)$$

$$x_{up} = (100 \times 100) / (250 + 100) \approx 28,57 \quad (5.2)$$

$$x_{left} = 0 \quad (5.3)$$

$$x_{down} = 0 \quad (5.4)$$

so, left and up percentage values are respectively 71,42 % and 28,57 %. The other directions are not involved. This calculation derives from 3.17.

- Motion quantity: we empirically set three thresholds for defining zero, minimal, medium and high quantity of motion. These values can be closer or farther from the calculated value according to the sensor. We simply calculate the distances of each one of them from the detected value and proportionally distribute them in a percentage scale. For example, if the thresholds are set to 2, 4 and 6 respectively, from lower to higher motion quantity, and a score of 3,4 is detected the distances are calculated as follows:

$$d_{zero} = |0 - 3,4| = 3,4 \quad (5.5)$$

$$d_{minimal} = |2 - 3,4| = 1,4 \quad (5.6)$$

$$d_{medium} = |4 - 3,4| = 0,6 \quad (5.7)$$

$$d_{maximum} = |6 - 3,4| = 2,6 \quad (5.8)$$

and the sum of distances is $s_d = 8$. The proportions are calculated according to the following formulas:

$$x_{zero} = 100 - (100 \times d_{zero}) / (s_d) = 52,5 \quad (5.9)$$

$$x_{minimal} = 100 - (100 \times d_{minimal}) / (s_d) = 82,5 \quad (5.10)$$

$$x_{medium} = 100 - (100 \times d_{medium}) / (s_d) = 92,5 \quad (5.11)$$

$$x_{maximum} = 100 - (100 \times d_{maximum}) / (s_d) = 67,5 \quad (5.12)$$

the applied formula is the same of 3.17.

Added Module: Speech Recognition

Due to the fact that the voice is one of the most used interaction method in multi-modal system, we provide also a speech-to-text module to our framework. It is based on Microsoft Speech Platform SDK 11 ⁷ and its dictionary filtered according

⁷<https://cdn.rawgit.com/Microsoft/Cognitive-Speech-STT-Windows/master/docs/SpeechSDK/html/db803a8f-b4c1-a049-00c7-4bf3472fc8cc.htm>

to the keywords defined in the rules. With this method we reduce the errors that the system can commit. Moreover, the confidence value allows to determine the accuracy of the recognition, providing percentage score in comparison with possible alternative (similar) words.

5.3 Sub-modules Tests

Before testing the entire framework we decided to estimate the effectiveness of the proposed modules: pointing, gesture recognition and re-identification. Each module has been tested in a different context, based on suitability. Finally, we tested the entire framework in a real scenario involving all the previously listed modules.

5.3.1 Gesture Recognition: Experiments

In order to show the effectiveness of this module, we tested its reliability in a critical application area: rehabilitation. In fact, the accuracy of systems used in medicine is usually higher than the majority of other topics. Moreover, the system presented in this paper is an improvement of the work described in [9]. It requires the cooperation of three low-cost devices: a HMD supported by positional sensors, a Microsoft Kinect v2 (as Tof camera) for the body, and a Leap Motion Controller (as 3D camera) for the hands. The combination of these devices allows patients a complete immersive experience with a high fidelity in acquiring data and tracking movements. The system is developed for supporting patients and therapists at the same time. In Figure 5.8 a graphical architecture is shown, highlighting the differences between the two sides (therapist/patient).

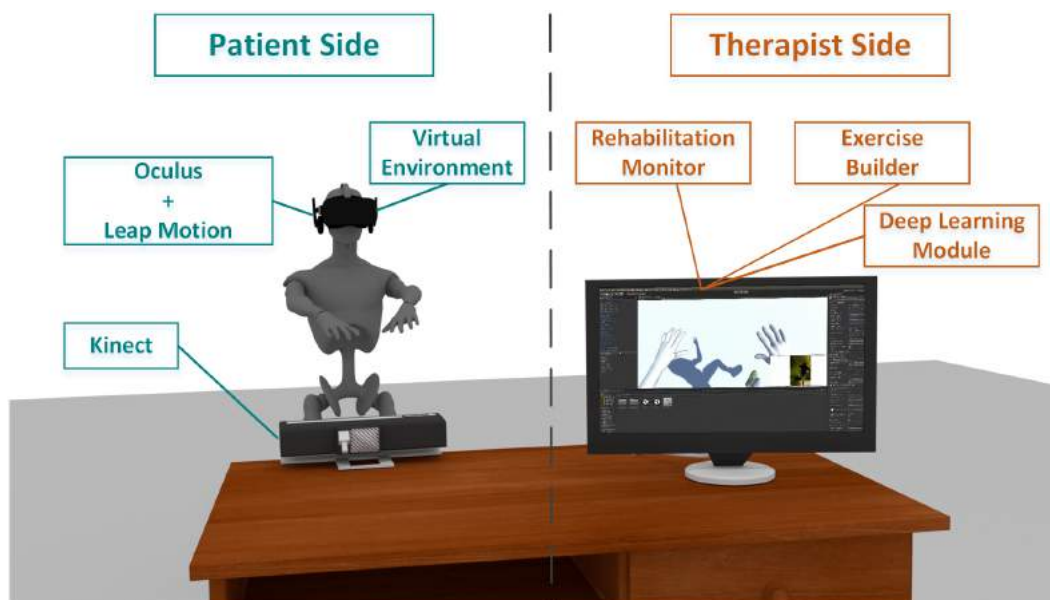


Figure 5.8. An overview of the proposed architecture. The patient side (left) highlights the used devices, while therapist side (right) points out the framework adopted to create, to monitor, and to customize the different serious games

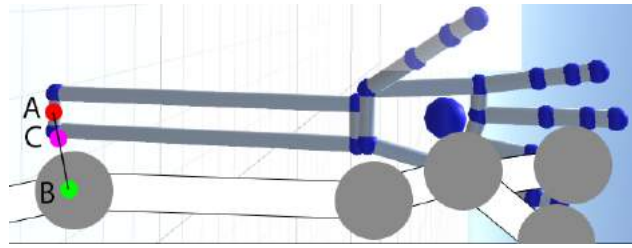


Figure 5.9. Combination of the Kinect and Leap Motion Controller models. Point A belongs to the Leap Motion Controller model, while point B belongs to the Microsoft Kinect model. The computed point C represents the best anchor point for combining the two models

As depicted in Figure 5.8, the system is composed by two logical layers: patient side and the therapist side. In the patient side, the Virtual Environment module manages input/output data from/to sensors. In detail, during the rehabilitation sessions the ToF and 3D cameras capture patients' data, i.e. hands and body, which are processed in real-time and stored for further elaborations. The type of acquired data (e.g., angles of the joints, positions of the joints) depends on the performed specific exercise. By the HMD, instead, the Virtual Environment module offers a VE with which the patients interact during the rehabilitation sessions. Like for the acquired data, the features of the interactive objects within the VE depend on the performed specific exercise. Moreover, the Virtual Environment module also provides a corresponding 3D model (i.e., an avatar) for a patient whose movements are fitted in it, thus allowing a real interactive experience.

In the therapist side, three modules are present: Rehabilitation Monitor, Exercise Builder, and Deep Learning. The first module receives data (i.e., video stream and models) from the Virtual Environment module and allows the therapist to monitor the patient's status during the execution of the rehabilitation exercise. The Exercise Builder allows the therapist to create (including training), modify, and delete the exercises within the system. For the management of the serious games, we inherited the method described in [9]. In detail, once the 3D environment is created by a skilled user, the therapist can add, modify, or delete elements of the environment and objects (including their parameters) by an eXtensible Markup Language (XML) file specifically formatted (this not a focus of the present paper, details are reported in [9]). Finally, the Deep Learning module is used in two different phases. In the first, it is used to learn the features of a new exercise added to the system. In the second, it is used to monitor and evaluate the performance of a patient during a rehabilitation session.

Combination of the models

The merging of the models provided by the Kinect and Leap Motion Controller is based on the union of selected joints. More specifically, we enable the Kinect to handle all the patient's skeleton joints except for those that concern hands and elbows. These last are taken from the Leap Motion Controller, due to its greater precision in modelling and tracking the hands. Since the elbow joints are in common between the representations provided by the two devices, they are used to merge

the two models. To know the optimal point where to anchor the model of the Leap Motion Controller on the model of the Kinect, the linear interpolation, shown below, between the two elbows points is computed:

$$\text{interp}X = x_1 + \frac{i}{n}(x_2 - x_1) \quad (5.13)$$

$$\text{interp}Y = y_1 + \frac{i}{n}(y_2 - y_1) \quad (5.14)$$

$$\text{interp}Z = z_1 + \frac{i}{n}(z_2 - z_1) \quad (5.15)$$

where, x_1, y_1, z_1 and x_2, y_2, z_2 are the coordinates of the elbow joints, n is the number of interpolated points, and i is the selected position. The more i is big, the more the point is near to the elbow point of the Kinect. By empirical tests, the following optimal configuration has been set: $n = 5$ and $i = 3$. In Figure 5.9, a graphical representation of this approach is shown.

The overall precision of the system in tracking the movements of the body and the articulation of the hands is mainly based on the precision of the used sensors. For this reason, we have adopted two of the consumer devices, i.e., Microsoft Kinect v2 and Leap Motion Controller, with the highest level of accuracy [126, 44]. In addition, since the elbow joints are in common between the models produced by the two devices, the precision of the tracking is also linked to the interpolation. In detail, the interpolation can be interpreted as a way to weight the combination of the two models. To prevent a wrong tracking of a patient, the system also implements a mechanism for the inference of the positions [133]. More specifically, if the Leap Motion Control loses the tracking of the hands, the Kinect can continue to support the movements. Likewise, if the sensors that assist the HMD fail, then the Kinect acts once again to support the movements. In this way, some levels of protection that react in case of accidental momentary faults have been implemented. Anyway, these are very infrequent events. In order to facilitate the interaction between therapists and system, we have implemented the functionalities and the interface of the system according to the most common usability criteria, i.e., the set of Nielsen's rules [80]. We have also tested the proposed system by the System Usability Scale (SUS) [20], obtaining a value of 83, which is better than the average. Summarizing, the main features that have contributed to develop a system with a good rank of usability are: a minimal interface, a simple customization of the exercises, and an adherence to the standards. To facilitate the interaction between patients and system, we have taken inspiration from the most common console games based on NUIs, taking into account the importance of achieving a clear goal in the most intuitive possible way. So, we used five patients affected by stroke or Parkinson's disease (hereinafter, PD) as case studies. The patients were distributed among three exercises commonly used in PD treatment [46, 59, 64, 90, 74]. The exercises included mixed hands and body rehabilitation tasks. The 3D environment was developed with Unity3D [85]. Notice that, all the parameters of the exercises, such as duration, number of obstacles, speed of the objects, and so on, can be modified by the therapist.

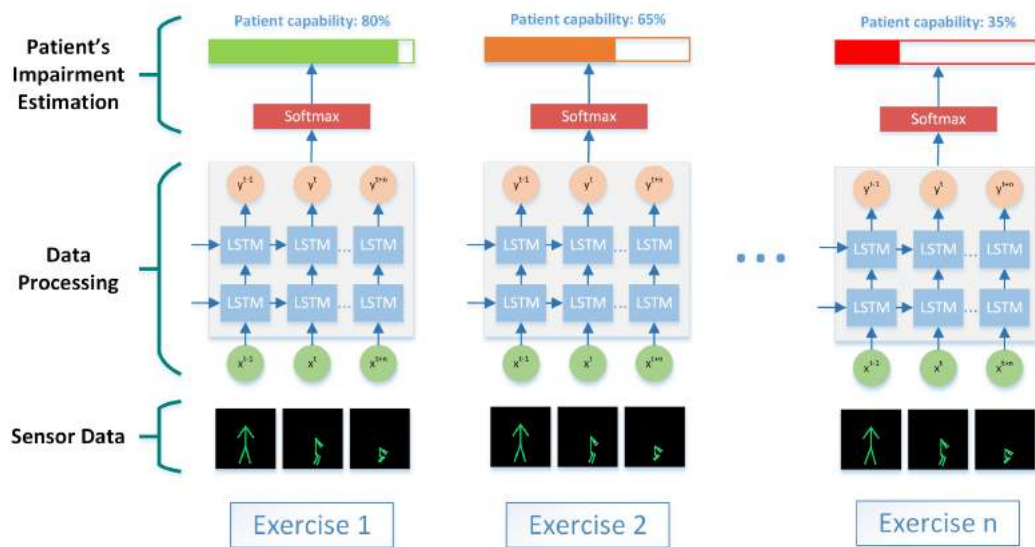


Figure 5.10. Deep learning architecture. For each serious game defined in the system, a RNN-LSTM is used to estimate the patient's performance. The data are acquired by the Kinect and Leap Motion Controller during the execution of the exercise and they are also supplied to the network (Sensor Data). Then, the latter processes the received data (Data Processing) and provides the patient's performance with respect to a set of healthy subjects (Patient's Impairment Estimation)

LSTM-RNN parameters

In this section, some considerations about the training of the network are reported. The first step to find the optimal network configuration was to try different numbers of stacked LSTM layers, maintaining constant the number of epochs, to observe how the accuracy changes. In Figure 5.11(a), the accuracy obtained with a fixed number of 800 epochs is shown. With 5 and 6 layers, we had a decrease in accuracy due to the few epochs with respect to the number of layers. This led us to a second stage, which consisted in augmenting the epochs for layer 5 and 6 up to 1600 and 2000, respectively. As shown in Figure 5.11(b), the accuracy has begun to converge to the accuracy value obtained with 4 layers. Notice that, with 4 layers we obtained the highest accuracy, but this was not a suitable solution due to the high time



Figure 5.11. LSTM-RNN accuracy with respect to the number of layers: (a) shows the accuracy obtained by using the same number of epochs for training, (b) shows the accuracy obtained by increasing the epochs for 5 and 6 layers

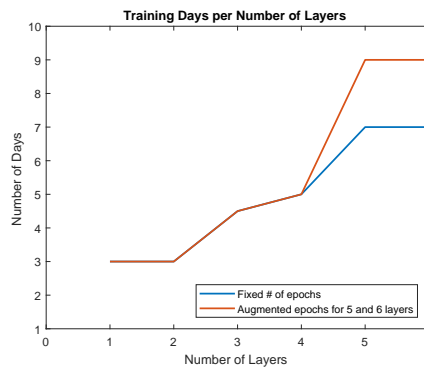


Figure 5.12. Number of days needed for training the network with respect to the number of layers. The blue line is the time needed when the number of epochs is fixed to 800, while the orange line is the time needed when the epochs are augmented to 1600 and 2000 for a 5 layers and 6 layers network, respectively

requested for the training. Between 2 and 3 layers there is a small difference in accuracy (i.e., about 0.3%), while the time needed to train a 3 layer network is higher. For these reasons, the use of 2 layers seemed a good compromise between accuracy and training time. In Figure 5.12, the number of days needed to train the network in consideration of the number of layers is reported. Concerning other training parameters, we found that a learning rate of 0.001 and a batch size of 5 has given the best results with the adopted video sequences.

Experimental protocol

All the serious games were performed by 20 healthy subjects, aged between 19 and 28 years. Their data were used to train three different LSTM-RNNs (i.e., one for each exercise). During the users' rehabilitation sessions, their data were acquired and, at the end of each session, they were given as input to the linked LSTM-RNN. The output of each obtained network was stored to enable the monitoring of the patients. In the proposed experiments, all users performed the same number of rehabilitation sessions, and each session had a duration between 45 and 60 minutes. The exercises proposed in this paper were taken from the medical literature according to the suggestions given by a set of five therapists. In order to correctly train the LSTM-RNNs, the therapists had also driven the selected healthy subjects during the execution of the exercises. In Table 5.1, the features per device used in the rehabilitation exercises are reported. To provide these features to the LSTM-RNNs, they are stored inside a vector as follows:

$$(LeftKnee_{(x,y,z)}, RightKnee_{(x,y,z)}, v) \quad (5.16)$$

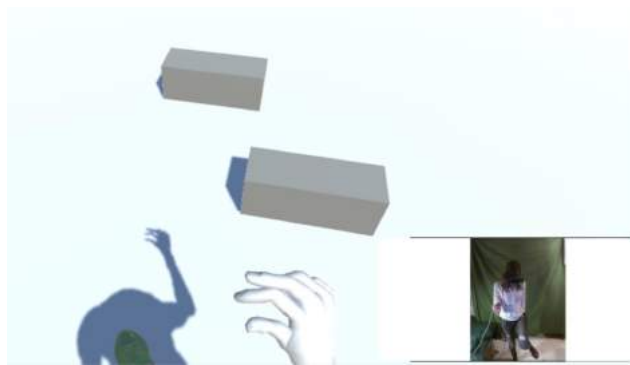
$$(LeftKnee_{AvgSpeed}, RightKnee_{AvgSpeed}, LeftFingers_{AvgSpeed}, RightFingers_{AvgSpeed}) \quad (5.17)$$

$$(\alpha_{ankle}, \beta_{ankle}, d) \quad (5.18)$$

Concerning Leap Motion Controller data, the normalization is not required since the device uses only relative values (i.e., not absolute positions in 3D space). Instead, the

Table 5.1. Features used to estimate patients' performance with respect to the performed exercise

Exercise	Used Features	
	Kinect	Leap Motion
1	Positions (x,y,z) of knees	Pinch strength, between 0.01 and 1.00
2	Average speed value of the knees	Average speed value of the fingertips
3	Angle between toe/heel and ankle, distance between hand and sphere center of mass	N/A

**Figure 5.13.** VE of the exercise 1. The bigger image shows the point of view of the user while performs the raise the knee and pinch with your fingers exercise

Kinect data were normalized by translating the skeleton of the user to the XZ plane. This is performed by using the skeleton centre of mass joint as anchor. During the experiments, at least a therapist and a software engineer were present to evaluate both the usability of the system and the effectiveness of the rehabilitation exercises.

Exercise 1: Raise the knee and pinch with your fingers

The first exercise proposed consists in raising the knee for avoiding obstacles and, at the same time, pinching with fingers of the hand opposite to the knee. Figure 5.13 shows how the exercise is seen by the patient through the HMD. The patient's 3D model is placed on a conveyor belt and some obstacles appear on the left and right side of the path. To avoid the collision, the patient has to raise the knee at

Table 5.2. Exercise 1: Average data collected from patients

Obstacle ID	null	0	1	2	3	4
Left pinch strength	0	0.82	0	0.88	0.9	0
Right pinch strength	0	0	0.96	0	0	0.98
Left knee height	-0.77	-0.29	-0.76	-0.27	-0.45	-0.73
Right knee height	-0.74	-0.78	-0.32	-0.79	-0.77	-0.28

Table 5.3. Exercise 1: Average data collected from healthy subjects

Description	Rest position	Right obstacle	Left obstacle
Left pinch strength	0	0.98	0
Right pinch strength	0	0	1.00
Left knee height	-0.76	-0.25	-0.76
Right knee height	-0.74	-0.75	-0.26

**Figure 5.14.** VE of the exercise 2. The bigger image shows the point of view of the user while performs the march and grasp exercise

the right moment. At the same time, the patient must keep the hands in front of the face and pinch with fingers of the hand on the opposite side of the raised knee (e.g., right knee left hand and vice versa). In Table 5.2 and Table 5.3, the average data collected during the exercise execution for patients and healthy subjects are shown. By having the data of the healthy subjects, the therapists can set a threshold value for knee raise and pinching action according to the Kinect and Leap Motion Controller measure units. Kinect calculates joint translation in meters, while Leap Motion Controller shows pinching strength in percentage, where 1 is the maximum pinching power and 0 is the minimum. The threshold value for pinching action was set on >0.8 and <-0.4 for knee raise.

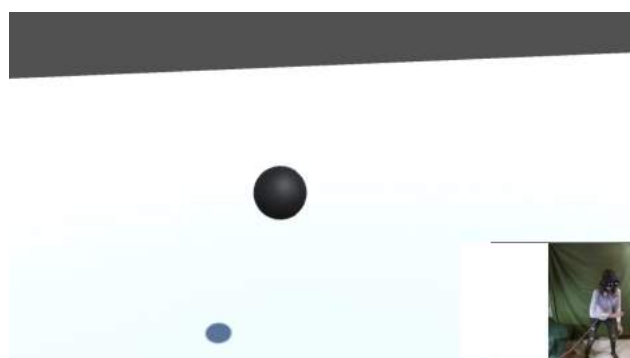
**Figure 5.15.** VE of the exercise 3. The bigger image shows the point of view of the user while performs the get up on heels or toes and cut with hands exercise

Table 5.4. Data collected from patients during the execution of the exercise 3 and their personal judgements

System Estimation	Patient's Judgement	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5
	Sphere 1	Hit\Hit	Hit\Hit	No Hit \No Hit	No Hit\No Hit	No Hit \No Hit
Sphere 2	Hit\Hit	Hit\Hit	No Hit\Hit	Hit \Hit	No Hit \No Hit	
Sphere 3	Hit\Hit	No Hit\Hit	No Hit \No Hit	Hit \Hit	No Hit \Hit	
Sphere 4	Hit\Hit	No Hit\No Hit	Hit \Hit	Hit \Hit	Hit\Hit	
Sphere 5	No Hit\No Hit	Hit\No Hit	Hit \No Hit	No Hit \Hit	Hit \Hit	
Sphere 6	No Hit\Hit	Hit\Hit	Hit \Hit	Hit \Hit	Hit \Hit	
Sphere 7	Hit\Hit	No Hit\Hit	No Hit \Hit	No Hit\Hit	No Hit \Hit	
Sphere 8	No Hit\Hit	Hit\Hit	Hit \No Hit	No Hit \Hit	No Hit\No Hit	
Sphere 9	No Hit\No Hit	No Hit\No Hit	Hit \Hit	Hit \Hit	Hit \No Hit	

Table 5.5. Comparison between the capabilities of the proposed system and similar systems at the state-of-the-art

	Realtime	Exercise Personalization	Full Body Detection	Sensor Type	Non-Haptic	Calibration	Immersion
Proposed System	Yes	Yes	Yes	Kinect v2, Oculus Rift, Leap Motion	Yes	Not needed	Full
Avola et al. [9]	Yes	Yes	No	Kinect	Yes	Not needed	No
Shiratuddin et al. [109]	Yes	No	No	Kinect (hands only)	Yes	Not needed	No
Saini et al. [104]	Yes	No	No	Kinect	Yes	Not needed	No
Sosa et al. [114]	Yes	No	No	Kinect	Yes	Needed	No
García-Martínez et al. [40]	Yes	No	No	Custom tactile controller	No	Not needed	No
Pei et al. [89]	Yes	Yes	Yes	Kinect	Yes	Not needed	No

Exercise 2: March and grasp

Figure 5.14 reports the second exercise, which consists in marching in place and grasping with the hands. Both movements must be synchronized. The VE presents a butterfly to the patient, and the more the hands are quickly closed, the more the butterfly flaps its wings rapidly. If any movement is stopped (e.g., march or grasp) the butterfly slowly falls.

Exercise 3: Get up on heels or toes and cut with hands

The third exercise consists in cutting spheres, with the hands, that appear randomly in front of the patient. The patient's fingers must be clenched and the hands opened. A cut is performed only when the hand of the 3D patient model collides with a sphere. Randomly, the system asks the patient to get on heels or toes, and when this action is performed, the system returns to the cutting part. This is an asynchronous exercise, that is to say that the focus of the patient is on a single task at time. In Figure 5.15 the execution of the exercise is shown. We asked the patients to declare if a collision between their avatars and the sphere occurs during the exercise. In Table 5.4, the comparison between what the patients stated and what the system detected is reported. Analysing the data, all the therapists (and engineers) affirmed that the system had made no mistakes, the mismatches were due to the perception of the patients, for this reason it could be treated as another parameter to monitor their progresses.

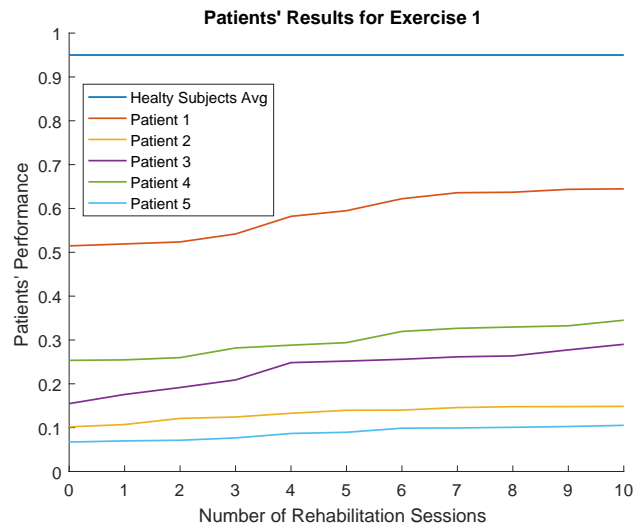


Figure 5.16. Patients' performances of the exercise 1 during rehabilitation sessions

Table 5.6. Comparisons between the average estimation proposed by the therapists and the system, respectively, for the exercise 1

Therapist Estimation	System Estimation	Patient 1	Patient 2	Patient 3	Patient 4	Patient 5
Therapist 1		60.00\58.72	10.00\13.24	20.00\23.45	30.00\29.86	15.00\8.80
Therapist 2		60.00\58.72	15.00\13.24	25.00\23.45	30.00\29.86	10.00\8.80
Therapist 3		60.00\58.72	15.00\13.24	25.00\23.45	25.00\29.86	10.00\8.80
Therapist 4		55.00\58.72	9.00\13.24	25.00\23.45	35.00\29.86	15.00\8.80
Therapist 5		55.00\58.72	20.00\13.24	20.00\23.45	25.00\29.86	5.00\8.80

5.3.2 Gesture Recognition: Results

In Table 5.5, a comparison of the capabilities of the proposed system with the current works at the state-of-the-art is reported [9, 109, 104, 40, 114, 89]. As shown, the proposed system is the only one that combines different devices to provide a fully immersive rehabilitation system, that does not require configuration and allows patients completely free movements during the rehabilitation sessions. Moreover, the system presents high levels of accuracy and reliability, thanks to the selected devices. The system also supports the creation and customization of complex serious games. Differently from other computer science fields, there are no datasets for rehabilitation. This is due to the fact that each patient is a unique case, and it is very difficult to find two patients with the same clinic history.

In Figure 5.16, Figure 5.17 and Figure 5.18 the predictions of the proposed algorithm are shown. For each exercise both the average performance of healthy subjects and the performance of each patient are reported. The performances of the healthy subjects were obtained by acquiring new data on them, as if they were patients. Despite their good physical conditions, the system detected some small impairments due to the impossibility, for the users, to perform exactly the same gestures. Depending on the freedom of movements allowed by the exercises, the system detected that a healthy subject had an impairment between 2% and 8%. So a patient who has an

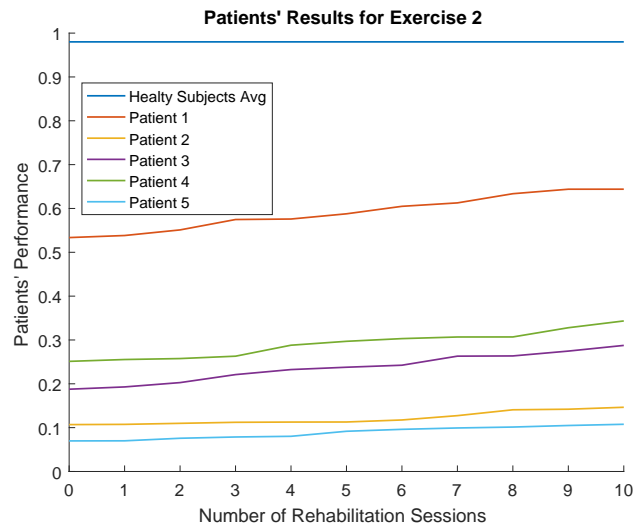


Figure 5.17. Patients' performances of the exercise 2 during rehabilitation sessions

Table 5.7. Comparisons between the average estimation proposed by the therapists and the system, respectively, for the exercise 2

Therapist Estimation		System Estimation				
		Patient 1	Patient 2	Patient 3	Patient 4	Patient 5
Therapist 1		55.00\59.09	15.00\12.15	20.00\23.69	30.00\29.09	5.00\8.87
Therapist 2		60.00\59.09	10.00\12.15	25.00\23.69	30.00\29.09	10.00\8.87
Therapist 3		60.00\59.09	10.00\12.15	20.00\23.69	35.00\29.09	5.00\8.87
Therapist 4		60.00\59.09	15.00\12.15	30.00\23.69	25.00\29.09	10.00\8.87
Therapist 5		55.00\59.09	15.00\12.15	20.00\23.69	30.00\29.09	15.00\8.87

impairment in that range, at the end of the therapy, can be considered as a healthy subject. The charts also highlight the overall conditions of the patients. The lower is the patient's starting performance, the more is the impairment. Regarding exercises 1 and 2, the extraction of the features was occurred without issues. In exercise 3, instead, some problems were found in detecting them when patients got up on heels or toes. This is due to the fact that the joints involved in the action are difficult to track, so the results of the exercise 3 were underestimated.

To verify if the estimation produced by the system was reliable, the five therapists that designed the exercises were consulted for a blind verification. In detail, we asked them how much, in their opinion, the impairment of a patient was after each session. Then, we compared their feedbacks with the results of the proposed algorithm. In

Table 5.8. Comparisons between the average estimation proposed by the therapists and the system, respectively, for the exercise 3

Therapist Estimation		System Estimation				
		Patient 1	Patient 2	Patient 3	Patient 4	Patient 5
Therapist 1		60.00\56.57	10.00\12.87	20.00\21.41	25.00\28.72	15.00\10.84
Therapist 2		55.00\56.57	15.00\12.87	15.00\21.41	25.00\28.72	10.00\10.84
Therapist 3		55.00\56.57	20.00\12.87	20.00\21.41	30.00\28.72	10.00\10.84
Therapist 4		60.00\56.57	15.00\12.87	20.00\21.41	30.00\28.72	10.00\10.84
Therapist 5		50.00\56.57	10.00\12.87	15.00\21.41	25.00\28.72	10.00\10.84

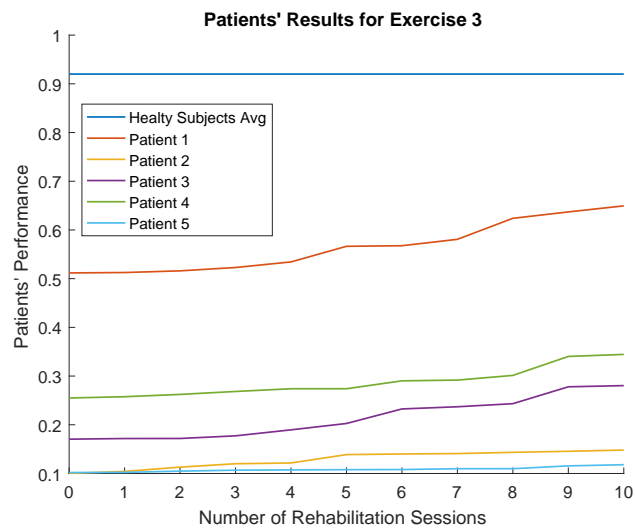


Figure 5.18. Patients' performances of the exercise 3 during rehabilitation sessions

Table 5.6, Table 5.7, and Table 5.8, the estimation values are reported. While the opinions of the therapists can be considered a qualitative measure, the results provided by the system can be considered a quantitative measure based on real data acquired from healthy subjects. If the estimations given by the therapists may vary among them, thus leading to different diagnosis, the proposed method provides a value that is closer to the patient's real conditions. The therapists strongly pointed out the benefits deriving from the proposed monitoring system. The data of the patient's rehabilitation history can be extremely important over time for creating ad-hoc exercises and more detailed reports about the healing process. Although there is already a large literature highlighting how these serious games increase patient's motivation and although similar works ensure that this kind of rehabilitation is widely accepted by the patients [55, 98, 45, 48, 93, 9, 109, 104, 40, 114, 89], we have also investigated the opinions of the patients about the system (with a detailed questionnaire). Summarizing, they considered it better than the classical approaches. In fact, patients really appreciated the immersion within the VE. They also expressed this preference according to their first experience with VR, which was something completely new for them. Moreover, they appreciated the opportunity to set targets and objectives of the games according to their capabilities. However, it is possible that, for very long sessions, they suffer of motion sickness effect, thus leading to a momentary interruption of the rehabilitation session. Anyway, judgments about this interaction technology are linked to personal feeling and predisposition.

The shown results prove that the proposed method is effective enough for using this system as gesture recognizer in the entire framework.

5.3.3 Pointing: Experiments

Due to the restrict topic and the lack of datasets, we tested the gesture recognizer with a self-testing strategy in a specifically designed environment. The aim of the



Figure 5.19. Pointing action performed by the user and recognized by the system. The pointed object is the camera itself.

proposed test consists in calculating the accuracy of the module considering different pointed directions and positions of the user in the sample environment. The designed room is based on a possible real scenario and, at the same time, a case in which the Kinect sensor is oriented in a specific direction. In Figure 5.20 the planimetry (a) and the 3D view of the room (b) with interactive objects are shown. The image is generated with the Layout Builder. The Kinect device is placed on the right-top corner, the 7.0" long one, over the TV. This location is chosen for providing the worst case, for increasing the challenge of coordinates translation function. Then, we selected some target items to be pointed. The chosen ones are the lamp, the black sofa and the red chair. The user was placed in two different positions, the first one is in front of the Kinect, in the middle of the room; the second is in front of the white door, at the maximum distance allowed for the sensor to still recognize the skeleton. After assuming the pose of pointing, the user confirms it rising the other arm and the system applies the algorithms. In Figure 5.19 the recognized pointing gesture is shown, according to the skeletization algorithm. The tests are executed repeating the pointing actions 10 times for each items and arm and with 5 different users. We collected the results as average values for each combination of users and tries. Three.js ratio metrics are calculated according to the following rule: 1 meter = 100.2917 Three.js units. This parameter keeps the right proportions between virtual and real environment.

5.3.4 Pointing: Results

The results collected after executing the tests are shown in tables 5.9 and 5.10, one for each position of the users in the room. The values underline the accuracy of

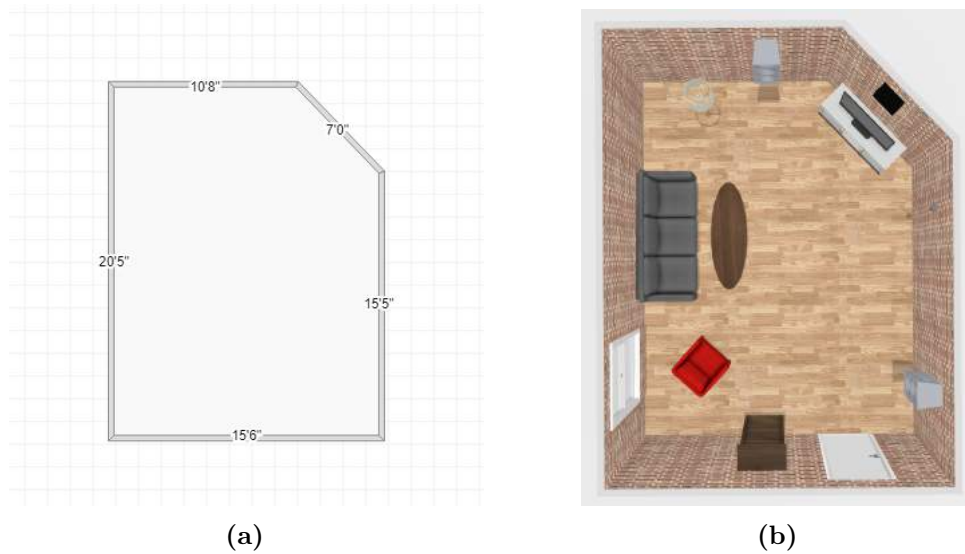


Figure 5.20. Planimetry (a) and top view (b) layout of the room used for testing the pointing recognizer. On the right wall, barely visible, there is a microphone.

the proposed method, with an average precision around 93,5% of correct identified targets. Moreover, we have to consider that there are two main noise factors. The first one is related to the sensor, in fact, the Kinect V2 is a low cost device that often causes the generation of a flickering skeletal model of the user. The second noisy factor involves occlusions. Let's consider the last column of the table 5.9. In this case, the unfair position of the arm of the user causes lack of precision. In fact, the skeleton joints are inferred due to the occlusion generated by the body of the user.

Table 5.9. Accuracy results collected during point recognizer module tests. This table is referring to the first position of the users, in the middle of the room.

	Lamp	Sofa	Chair
User 1	100%	100%	80%
User 2	100%	100%	90%
User 3	100%	90%	80%
User 4	100%	100%	70%
User 5	100%	100%	90%

Table 5.10. Accuracy results collected during point recognizer module tests. This table is referring to the second position of the users, in front of the door.

	Lamp	Sofa	Chair
User 1	100%	90%	90%
User 2	90%	100%	100%
User 3	90%	90%	90%
User 4	100%	90%	90%
User 5	100%	100%	90%

However, the results are quite satisfying, according to this preliminary test. Comparisons can not be performed due to numerous factors: low interest of the scientific community in this specific application, exiguous number of techniques in literature, lack of specific datasets, specific testing environments for each similar work.

We underline that this module is not the main focus of the work. Moreover, a not so high precision of the recognition is helpful for testing the entire framework.

5.3.5 Person Reidentification: Experiments

In order to test the reidentification module, we introduced a new environment. Due to its suitability, surveillance application area is particularly indicated for our purpose. So, we exploited a Microsoft Kinect V2.0 device for monitoring three different rooms in the same building. Then, we asked 32 persons to move into each room 12 times. Every 4 times, we changed the light conditions in each room. The result is an original dataset of 32 subjects captured by 3 different cameras in 3 distinct light conditions. It has been used for preliminary testing purposes. RGB images and skeleton data are stored. The latter consists in joints' positions for each frame. Inspired by methods described in [43], we divided the dataset according to the cameras: data associated to the first and the second is related to the train and the frames acquired by the third populated the test set. Then, we used a random frame from each sequence in test set for calculating the one-shot person reidentification accuracy.

Then we tested the system on KinectREID [11], one of the compliant dataset that are specifically designed for reidentifying persons with RGB-D cameras. We used the same approach shown in [99]. The dataset contains 483 videos of 71 persons in different rooms of the same building. Each sequence is composed by RGB, depth and skeleton information, acquired with Microsoft Kinect SDK. We randomly selected 20 sequences of different people and used all frames for training the system. The remaining sequences are used as test set. As usual in human reidentification research area, cumulative matching characteristic (CMC) is calculated for providing performances' results. Always referring to [99], we repeated experiments 10 times and calculated the average value of accuracy for obtaining the final results. Concerning the threshold distances, we skipped this passage due to the nature of this test. However, in real scenarios and using a Kinect for XBOX One, we set a minimum value to 1.3m and the maximum to 1.8m.

5.3.6 Person Reidentification: Results

We performed tests on own personal dataset before comparing the effectiveness of the system with a similar one. The CMC results, shown in Figure 5.21, denote that the rank-1 rate is around 97% and grows to 99% at rank-4.

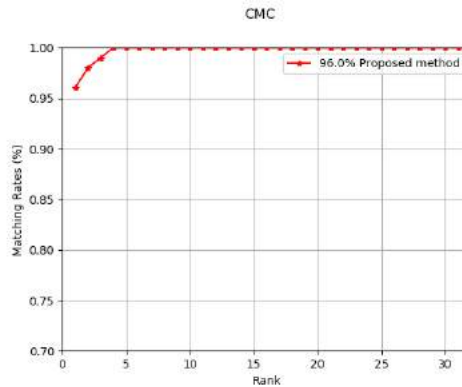


Figure 5.21. CMC of proposed method on generated dataset.

CMC tests performed on KinectREID and compared with [99] are shown in Figure 5.22. The results are similar, however the score is lower.

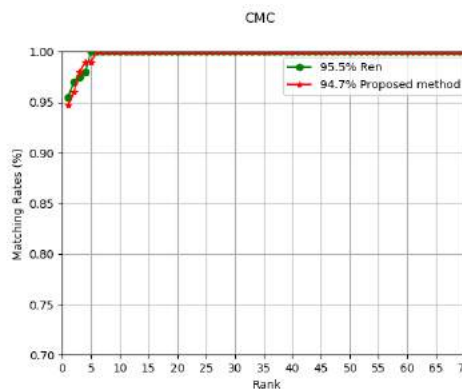


Figure 5.22. CMC of proposed method and Ren's one[99] on KinectREID dataset.

The Table 5.11 shows that the results are lower than one of the best method in literature. We have to denote that this related work is based on deep learning principles, so it requires a higher computational power.

The combination of RGB and anthropometrical measures derived from Depth is critical for obtaining these results. Single channels provide extremely low performances instead.

However, we underline that this module is not the core of the study. We wanted to implement an innovative and competitive function for testing the entire framework. We suppose that this aim has been reached. Moreover, a less precise method could be more helpful than more accurate ones. In fact, the main topic of the document is related to the effectiveness of the proposed probabilistic method for improving the precision of each involved module.

Table 5.11. Accuracy comparison between the proposed method and state of the art ones on the entire KinectREID dataset.

Method	Rank-1	Rank-5	Rank-10
SGTrP3 + Score level [51]	76.6	/	99.4
DVCov + SKL [128]	71.7	88.4	/
MMUDL [99]	76.7	87.5	96.1
Proposed Method	74.4	83.1	90.8

5.4 Entire System Test

Due to the uniqueness of the proposed framework, we designed some custom tests for rating the grade of effectiveness of the system, underlining the recognition accuracy. In our first step we designed a likely and various environment, that involves all the proposed sensors and modules. Then, we created some related rules that could produce mistakes due to ambiguous events.

We designed an intelligent domotic environment. It is composed by two rooms with the following sensors and related modules:

- Room 1:
 - Kinect V2: placed in front of the entrance of the room. It is associated to the gesture recognition and motion detection modules;
 - A microphone: it consists in a panoramic microphone placed near the Kinect device. It is associated to the speech recognition module;
 - Two proximity sensors: one for each door in the room. It is associated to a simple proximity detection module.
- Room 2:
 - Kinect V2: placed over the tv, on north wall of the room. It is associated to the gesture recognition and motion detection modules;
 - A microphone: it consists in a panoramic microphone placed near the entrance. It is associated to the speech recognition module;
 - A proximity sensors: it is placed near the entrance door. It is associated to a simple proximity detection module.

The output devices are:

- Room 1:
 - A speaker: placed in front of the entrance;
 - A tv: placed in front of the entrance;
 - A light: placed on the east of the room.
- Room 2:

- A speaker: placed on the north wall of the room;
- A tv: placed on the north of the room;
- A light: placed on the north-west of the room.

In Figures 5.23 5.24 5.25, the top and side views of the layout of the environment are shown. On the right there is the Room1 and on the left the Room2. The entrance is located in the Room1, on the south wall. The rooms are communicating through a door on the west wall of Room1 and on the east wall of Room2. In Figure 5.26 a panoramic photo of the real Room 2 is shown. It is important to underline that the added furniture is not influencing the experiment due to the fact that none of the sensors is occluded.

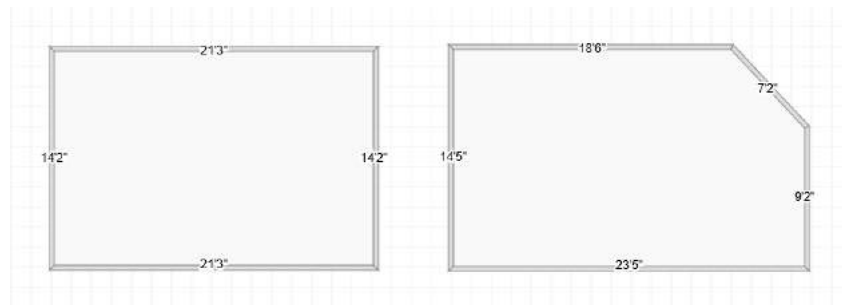


Figure 5.23. Planimetry of the testing environment. On the right, Room1, on the left, Room2. The entrance is on south wall of Room1.



Figure 5.24. Top view of the testing environment with objects. In the image the communication door is not visible due to a prospective occlusion. However, it is located on the west wall of Room1 and on the east wall of Room2.

The involved rules are logically designed for reflecting some usual actions that can be performed in the proposed environment. Semantically, the rules can be summarized in the following way:

- Enters the loft (Figure 5.27) and greets -> turn on the light and the speaker in the room 1;
- Asks for turning on the light after walking to some object -> turn on the nearest light;
- Asks information about a pointed object -> turn on the nearest light and the speaker in the involved room only. This rule is linked to the previous one



Figure 5.25. Side 3D view of the testing environment. (a) the point-of-view is watching from south-east to north-west. (b) the point-of-view is watching from north-west to south-east.



Figure 5.26. Panoramic photo of the real Room 2 environment. As shown, there is also some not relevant furniture. However, it does not influence the experiment.

due to the fact that the user could go near the pointed object after asking something about it;

- Stops music and turns off tvs -> turn off speakers and TVs in both rooms. It is most probable after some other rule completion turns on these devices;
- When moving from a room to another -> the speaker reproduces an audio with the name of the re-identified person;
- While sitting on the sofa asks for some music -> turn on the speaker in room 2;
- While sitting on the sofa asks for turning on tv -> turn on the tv in the room2;
- While sitting on the sofa asks for turning off everything -> turn off everything in room2. This rule is linked to the previous two due to the fact that the speaker or the tv could be turned off while still sitting on sofa;
- While sitting on the chair asks for some music -> turn on the speaker in room 2 **only if authorized. Persons that can perform this tasks are $p_{1,5,8,10,11}$** ;
- While sitting on the chair asks for stopping the music -> turn off the speaker in room 2. This rule is linked to the previous one;

- After moving from a room to another asks for turning on some music -> turn on the speaker in the target room;
- After moving from a room to another asks for turning on the tv -> turn on the tv in the target room;
- Greets and exit the loft -> turn everything off;



Figure 5.27. User entering in Room 2.

The authorization is linked to the identification module. The gesture recognizer has been trained for recognizing the following actions:

- Greet (Figure 5.28): rising the left or right arm and slightly moving it;
- Point (Figure 5.29): extending the right or left arm in front of the body for a few seconds;
- Sit on chair (Figure 5.30): the action is composed by the standing position, the movement and the final pose;

- Sit on sofa (Figure 5.31): the action is composed by the standing position, the movement and the final pose. It is different from the previous action due to a different movement and final pose caused by the sofa conformation;
- Stops the speaker: rising the left or the right arm and holding it in position for a few seconds;

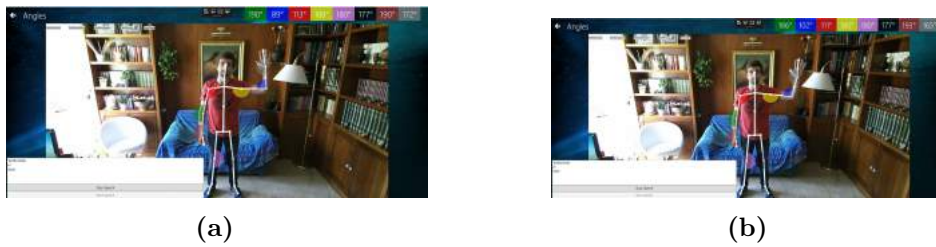


Figure 5.28. Frames from "greet" sequence.

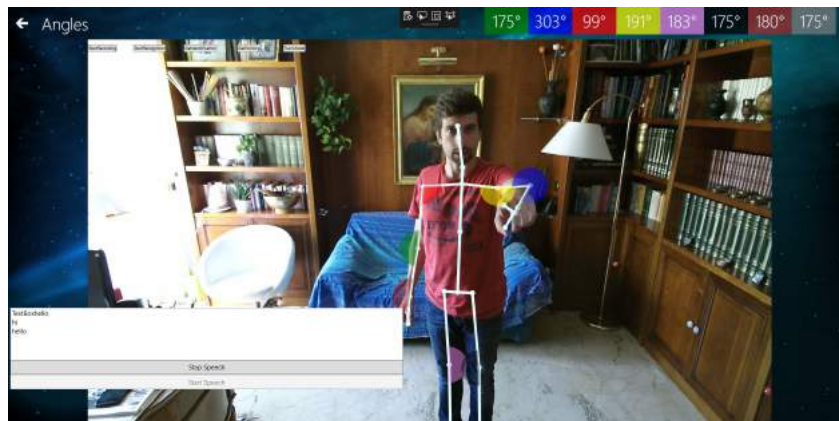


Figure 5.29. The user performing a pointing action.

The speech recognizer is providing the following dictionary:

- Greet: "hello", "goodbye", "hi" and synonyms;
- Turn on music: "turn on some music", "switch on the speaker" and similar phrases involving "turn on" or "switch on" and "speaker" or "music" keywords;
- Turn on tv: "turn on the tv", "switch on the television" and similar phrases involving "turn on" or "switch on" and "tv" or "television" keywords;
- Turn on the light: "turn on the light", "switch on the lamp" and similar phrases involving "turn on" or "switch on" and "light" or "lamp" keywords;
- What is: "what is this", "what is that" and similar questions that involve "what" or "which" keywords;
- Stop music: "stop the music", "turn off the music" and similar phrases involving keywords "stop", "switch off" or "turn off" and "music";

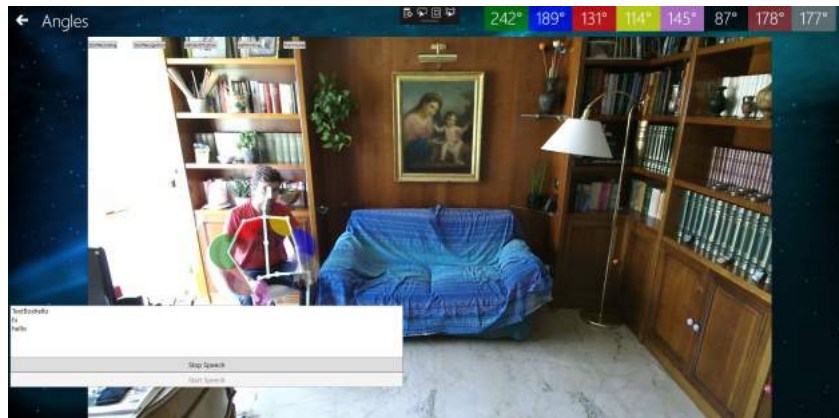


Figure 5.30. The user performing a sitting action on the chair.

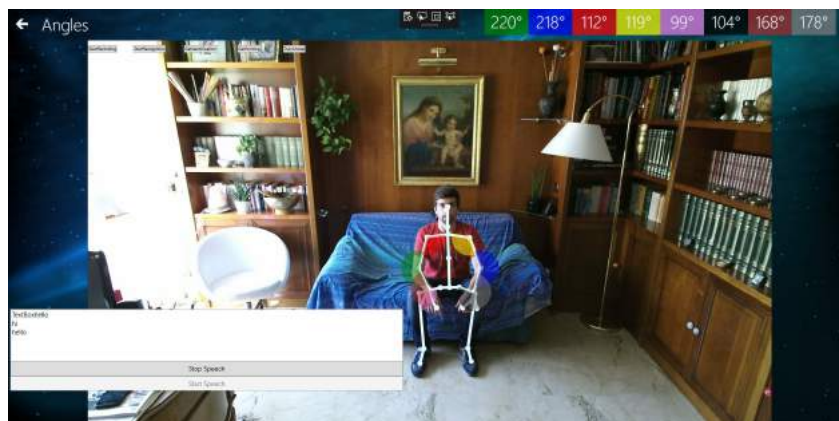


Figure 5.31. The user performing a sitting action on the sofa.

- Stop tv: "turn off the tv", "switch off the television" and similar phrases involving "turn off" or "switch off" and "tv" or "television" keywords;
- Stop: "turn off everything", "stop that" and other similar phrases involving "stop", "turn off" or "switch off" keywords and phrases that do not contain words such as "speaker" or "television" because are linked to another command;

In these cases we could denote some similarities. The sitting action. We have two different locations for allowing the user to rest, a chair and a sofa. Both are linked to separate gestures. Considering the fact that the position of the user is deliberately not tracked in this phase, the system can identify which action is executed only exploiting the performed gesture. The posture acquired by the user when sitting on both of them is slightly different due to the height and the tilt of the back. At the same time, also the speech recognizer is challenged, for example considering similarities between "turn on" and "turn off" sentences. All these cases are obviously located in the same temporal window, producing a disambiguation need.

5.4.1 Experiments

Tests of the system are performed with 13 different persons and an arbitrary number of randomic actions to execute in the rooms. The users were trained with a developer on events that the system can recognize. Then, they had between 10 to 15 minutes for testing interactions with the system and learning how the temporal windows are timed. The information about the involved persons are the following:

- Age: between 20 and 40;
- Skill in using technological devices: middle-high;
- Gender: 10 males, 3 females;
- Clothes: almost the upper body is slightly different from each other. For example, three shirts were colored with a different gradation of red;

The time provided to the users for executing the task is 2 minutes, however they could get out from the loft before the end.

The involved sensors are translated in the following variables:

- s_1 = Proximity sensor at entrance door;
- s_2 = Kinect in Room 1;
- s_3 = Microphone in Room 1;
- s_4 = Proximity sensor between Room 1 and 2;
- s_5 = Kinect in Room 2;
- s_6 = Microphone in Room 2;

The conditions are turned in rules, according to the specifications of the framework, as follows:

- **R1:** Enters the loft and greets = $to_{X_m Y}(s_1(e_1), s_2(e_2)) \vee to_{X_b Y}(s_1(e_1), s_2(e_2)) \vee to_{X_o Y}(s_1(e_1), s_2(e_2))$;
- **R2:** Asks for turning on the light after walking to some object = $to_{X_b Y}(s_2(e_3), s_3(e_4)) \vee to_{X_m Y}(s_2(e_3), s_3(e_4))$;
- **R3:** Asks information about a pointed object = $to_{X_{eq} Y}(s_2(e_5), s_3(e_6)) \vee to_{X_o Y}(s_2(e_5), s_3(e_6)) \vee to_{X_{eq} Y}(s_5(e_5), s_6(e_6)) \vee to_{X_o Y}(s_5(e_5), s_6(e_6))$;
- **R4:** Stops music and turns off tv in both rooms = $to_{X_{eq} Y}(s_2(e_7), s_3(e_8)) \vee to_{X_{eq} Y}(s_5(e_7), s_6(e_8))$. Linked to **R7**, **R8**, **R12**, **R13**, **R14** and **R15**;

- **R5:** When moving from a room to another (Room1 to Room2) = $to_{X_{eq}Y}(s_2(e_9), s_4(e_1)) \vee to_{X_mY}(s_2(e_9), s_4(e_1)) \vee // to_{X_oY}(s_2(e_9), s_4(e_1))$;
- **R6:** When moving from a room to another (Room2 to Room1) = $to_{X_{eq}Y}(s_5(e_9), s_4(e_1)) \vee to_{X_mY}(s_5(e_9), s_4(e_1)) \vee // to_{X_oY}(s_5(e_9), s_4(e_1))$;
- **R7:** While sitting on the sofa asks for some music = $to_{X_{eq}Y}(s_5(e_{10}), s_6(e_{11})) \vee to_{X_sY}(s_6(e_{11}), s_5(e_{10})) \vee to_{X_dY}(s_6(e_{11}), s_5(e_{10})) \vee to_{X_sY}(s_6(e_{11}), s_5(e_{10})) \vee to_{X_fY}(s_6(e_{11}), s_5(e_{10}))$;
- **R8:** While sitting on the sofa asks for turning on tv = $to_{X_{eq}Y}(s_5(e_{10}), s_6(e_{12})) \vee to_{X_sY}(s_6(e_{12}), s_5(e_{10})) \vee to_{X_dY}(s_6(e_{12}), s_5(e_{10})) \vee to_{X_sY}(s_6(e_{12}), s_5(e_{10})) \vee to_{X_fY}(s_6(e_{12}), s_5(e_{10}))$;
- **R9:** While sitting on the sofa asks for turning off everything = $to_{X_{eq}Y}(s_5(e_{10}), s_6(e_8)) \vee to_{X_sY}(s_6(e_8), s_5(e_{10})) \vee to_{X_dY}(s_6(e_8), s_5(e_{10})) \vee to_{X_sY}(s_6(e_8), s_5(e_{10})) \vee to_{X_fY}(s_6(e_8), s_5(e_{10}))$. Linked to **R7** and **R8**;
- **R10:** While sitting on the chair asks for some music = $to_{X_{eq}Y}(s_5(e_{13}, p_{1,5,8,10,11}), s_6(e_{11})) \vee to_{X_sY}(s_6(e_{11}), s_5(e_{13}, p_{1,5,8,10,11})) \vee to_{X_dY}(s_6(e_{11}), s_5(e_{13}, p_{1,5,8,10,11})) \vee to_{X_sY}(s_6(e_{11}), s_5(p_{1,5,8,10,11}(e_{13}))) \vee to_{X_fY}(s_6(e_{11}), s_5(e_{13}, p_{1,5,8,10,11}))$;
- **R11:** While sitting on the chair asks for stopping the music = $to_{X_{eq}Y}(s_5(e_{13}), s_6(e_8)) \vee to_{X_sY}(s_6(e_8), s_5(e_{13})) \vee to_{X_dY}(s_6(e_8), s_5(e_{13})) \vee to_{X_sY}(s_6(e_8), s_5(e_{13})) \vee to_{X_fY}(s_6(e_8), s_5(e_{13}))$. Linked to **R7**, **R12** and **R13** ;
- **R12:** After moving from a room to another asks for turning on some music (Room1 to Room2) = $to_{X_mY}(s_2(e_9), s_4(e_1)) \wedge s_6(e_{11}) \vee to_{X_{eq}Y}(s_2(e_9), s_4(e_1)) \wedge s_6(e_{11}) \vee to_{X_oY}(s_2(e_9), s_4(e_1)) \wedge s_6(e_{11})$;
- **R13:** After moving from a room to another asks for turning on some music (Room2 to Room1) = $to_{X_mY}(s_5(e_9), s_4(e_1)) \wedge s_3(e_{11}) \vee to_{X_{eq}Y}(s_5(e_9), s_4(e_1)) \wedge s_3(e_{11}) \vee to_{X_oY}(s_5(e_9), s_4(e_1)) \wedge s_3(e_{11})$;
- **R14:** After moving from a room to another asks for turning on the tv (Room1 to Room2) = $to_{X_mY}(s_2(e_9), s_4(e_1)) \wedge$

$$s_6(e_{12}) \vee to_{X_{eq}Y}(s_2(e_9), s_4(e_1)) \wedge$$

$$s_6(e_{12}) \vee to_{X_oY}(s_2(e_9), s_4(e_1)) \wedge$$

$$s_6(e_{12});$$

- **R15:** After moving from a room to another asks for turning on the tv (Room2 to Room1) = $to_{X_mY}(s_5(e_9), s_4(e_1)) \wedge$
 $s_3(e_{12}) \vee to_{X_{eq}Y}(s_5(e_9), s_4(e_1)) \wedge$
 $s_3(e_{12}) \vee to_{X_oY}(s_5(e_9), s_4(e_1)) \wedge$
 $s_3(e_{12});$
- **R16** Greets and exit the loft = $to_{X_mY}(s_2(e_2), s_1(e_1)) \vee$
 $to_{X_bY}(s_2(e_2), s_1(e_1)) \vee$
 $to_{X_oY}(s_2(e_2), s_1(e_1));$

There was no time limit for performing the task, however each run took less than 7 minutes due to the timing of temporal windows. When the desired output is not obtained, the users could decide to ignore or repeat the command from the beginning of the rule after waiting some seconds. The collaboration of users is necessary for obtaining unconcerned results.

All tests were executed on a single machine with multiple sensors. It was composed by an Intel i7 5930k, 16GB DDR4 of RAM, a Samsung 850 Pro SSD 250GB, a Nvidia GTX1070 8GB and a motherboard Asus Rampage Extreme V, the most important component due to the numerous USB connections needed. Moreover, a Corsair 850RM power supply is used for avoiding loss of energy while all sensors were on. Multiple Kinect sensors are managed as follows: one is directly used inside the main operating system, while the second is managed by a virtual machine that provides information in real-time to the main program through a "http" server-client protocol. The slight delay is avoided subtracting its calculated value to the events' timestamps. However, the user experience was less addictive.

5.4.2 Results

Results have been collected counting occurrences of events successfully identified among all users and runs. In Table 5.12 the accuracy values are shown.

It is important to underline that numerous rules have been completed thanks to the proposed increase method of probability value of involved events. The average accuracy of the entire system, according to this test, is around 94,12%. However, this result could not be enough satisfying for proving the effectiveness of the probabilistic temporal logic finite state machine method. So, we recorded the input received by each sensor and we executed again the tests disabling this module. Obtained results are shown in Table 5.13. We can denote that some rules' conditions completion has been missed due to mistakes performed during disambiguation phase. In fact, the involved ones are related to sofa's and chair's actions.

The average retrieved accuracy is around 85,01%, providing a difference of 9,10 percentage points with the proposed method. Concerning single events, we collected information about their recognition performances. The Graph 5.32 is showing the probability related to each event based on the performed experiment. We can highlight that events not involved in reinforcement of probabilistic temporal logic

Table 5.12. Results collected for each user and run. The percentage value shows the accuracy of each rule condition completion based on the times that the user tries to complete them performing the required events.

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12	User 13
R1	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)
R2	50% (2 times)	\	100% (1 time)	\	100% (2 times)	\	\	0% (1 time)	\	100% (1 time)	\	100% (1 time)	\
R3	100% (2 times)	100% (3 times)	100% (3 times)	\	50% (2 times)	100% (1 time)	100% (2 times)	100% (2 times)	100% (2 times)	\	50% (2 times)	100% (1 time)	\
R4	\	100% (1 time)	66% (3 times)	100% (1 time)	\	\	100% (1 time)	100% (2 times)	100% (1 time)	\	\	\	\
R5	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)
R6	100% (2 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (2 times)	100% (1 time)	\	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (2 time)	100% (1 time)
R7	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)	50% (2 times)	\
R8	50% (2 times)	100% (1 time)	100% (1 time)	\	100% (1 time)	\	100% (1 time)	\	0% (1 time)	\	\	100% (1 time)	\
R9	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	\	\	100% (1 time)	100% (1 time)	\	\	\
R10	\	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)	\	\	100% (1 time)
R11	\	100% (1 time)	100% (1 time)	50% (2 times)	\	100% (1 time)	\	\	100% (1 time)	100% (1 time)	\	\	\
R12	\	\	\	\	\	\	\	\	\	\	\	\	\
R13	\	0% (1 time)	\	\	\	\	100% (1 time)	\	100% (1 time)	\	100% (1 time)	\	100% (1 time)
R14	100% (1 time)	\	\	\	100% (1 time)	\	\	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)
R15	\	100% (1 time)	\	\	100% (1 time)	\	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	\
R16	100% (1 time)	100% (1 time)	\	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)

Table 5.13. Results collected for each user and run disabling the proposed probabilistic module. Some ambiguous events are mistaken and the related rules' conditions are not satisfied as well.

	User 1	User 2	User 3	User 4	User 5	User 6	User 7	User 8	User 9	User 10	User 11	User 12	User 13
R1	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)
R2	50% (2 times)	100% (1 time)	100% (1 time)	\	0% (2 times)	\	\	0% (1 time)	\	100% (1 time)	\	100% (1 time)	\
R3	100% (2 times)	100% (1 time)	66% (3 times)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (2 times)	100% (2 times)	\	50% (2 times)	\	\
R4	\	66% (3 times)	\	100% (1 time)	100% (1 time)	\	\	\	0% (1 time)	\	\	\	\
R5	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (2 times)	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)
R6	100% (2 time)	100% (1 time)	100% (1 time)	100% (1 time)	100% (2 times)	100% (1 time)	\	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (2 time)	100% (1 time)
R7	100% (1 time)	100% (1 time)	\	100% (1 time)	0% (1 time)	\	\	100% (1 time)	0% (1 time)	\	\	50% (2 times)	\
R8	50% (2 times)	100% (1 time)	100% (1 time)	\	\	\	100% (1 time)	\	0% (1 time)	\	\	0% (1 time)	\
R9	100% (1 time)	\	100% (1 time)	0% (1 time)	0% (1 time)	0% (1 time)	\	\	100% (1 time)	100% (1 time)	\	\	\
R10	\	100% (1 time)	100% (1 time)	50% (2 times)	\	100% (1 time)	\	100% (1 time)	0% (1 time)	100% (1 time)	\	\	100% (1 time)
R11	\	100% (1 time)	\	\	\	\	\	\	\	100% (1 time)	\	\	\
R12	\	\	\	\	\	\	\	\	\	\	100% (1 time)	\	\
R13	\	0% (1 time)	\	\	\	\	100% (1 time)	\	100% (1 time)	\	100% (1 time)	\	100% (1 time)
R14	100% (1 time)	\	\	\	100% (1 time)	\	\	100% (1 time)	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)
R15	\	100% (1 time)	\	\	100% (1 time)	\	\	\	100% (1 time)	0% (1 time)	100% (1 time)	100% (1 time)	\
R16	100% (1 time)	100% (1 time)	\	100% (1 time)	\	100% (1 time)	100% (1 time)	100% (1 time)	\	100% (1 time)	100% (1 time)	\	100% (1 time)

finite state machine module are acting as follows: if the base recognition rate is high (like a proximity sensor that is near 100% of precision) there are no changes, but if the accuracy is low, the results are worst then other events (with the same sensor). In fact, we can see that the event e_5 is less accurate than e_8 . However, considering for example the Kinect sensor, the differences between single gestures involved can slightly distort the results. So, we created a graph, shown in Graph 5.33, for comparing the same events. The difference between e_7 , e_8 and e_9 is noticeable. It confirms the theory that the ambiguous events are adequately treated by the proposed method.

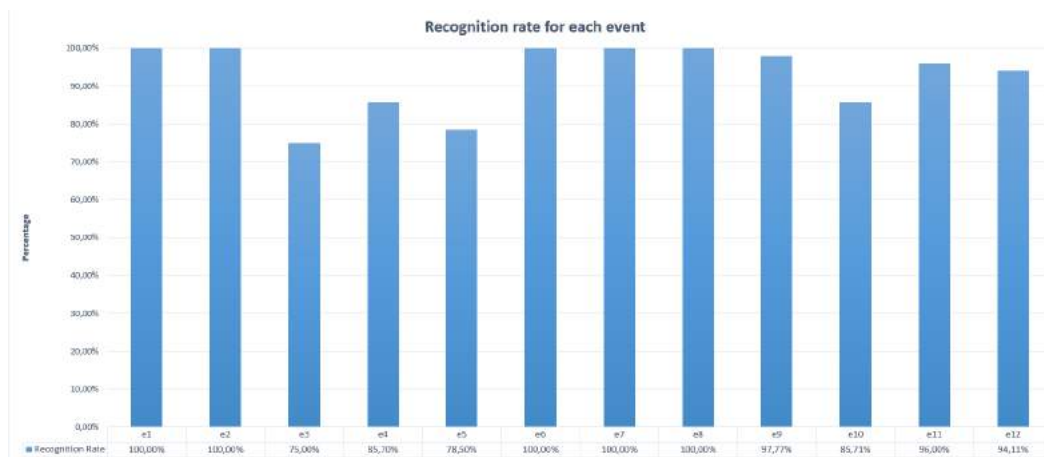


Figure 5.32. Recognition rate for each event based on the proposed experiment. We can see that the majority of errors are evident in isolated events, that are not involved in disambiguation functions.

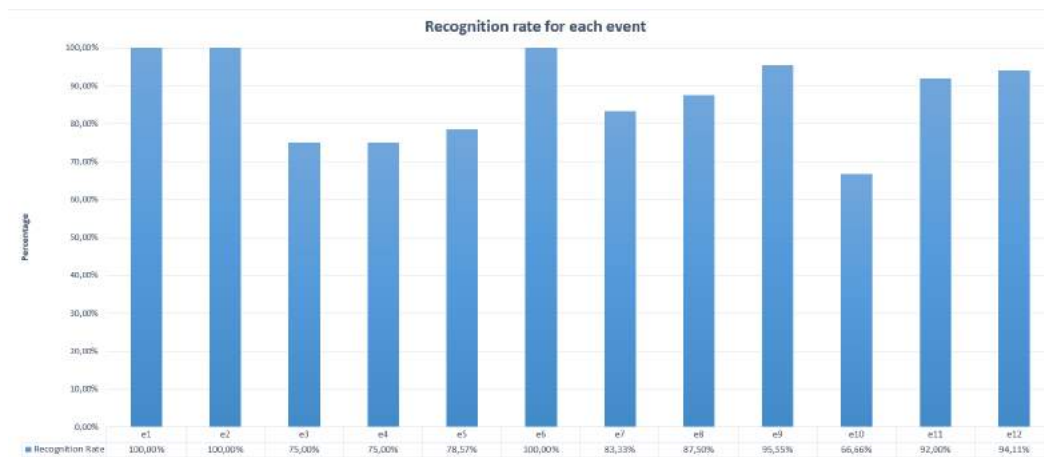


Figure 5.33. Recognition rate for each event based on the proposed experiment after disabling the probability module. Results are worst on ambiguous events.

Due to the fact that there are no dataset that can be used in the specific treated case, we decided to compare the system with other similar ones based on functionalities and used methods. In Table 5.14 these differences are shown. Most of them start from Allen's algebra grammar. Semantic relations provide high

performances and are also used in framework-like systems. However, the use of semantic connections is compatible with the proposed method. In fact, both ours and semantic relation methods calculated the probabilities according to different approaches that are not conflictual. This and all the mentioned factors create numerous divergences between all these systems and don't allow to compare them according to conventional methods.

Table 5.14. Features comparison between proposed method and the most similar works in literature.

	Base Grammar	Method	System
Bennett [15]	Spatial language	Region Connection Calculus (RCC-8) and Interval Temporal Logic	\
Zhang [132]	Allen's algebra	Interval algebra network (IAN)	Not framework like (RGB video analyzer)
Crispim-Junior [28]	Allen's algebra	Semantic fusion	Framework like
Mehlmann [72]	Allen's algebra	Semantic networks + Incremental parsing and fusion	Framework like
Song [113]	Allen's algebra	Events encoded in Markov logic	Framework like
Proposed system	Allen's algebra	Probabilistic temporal logic finite state machines	Framework like

In conclusion, we can say that the idea behind the proposed framework is promising. The accuracy difference between the results obtained with probabilistic method and without it highlights that the reinforcement is acting exactly where it is more needed. Mistakes are considerably decreased, improving disambiguation performances.

Chapter 6

Conclusions

In recent years, interactive environments have become a hot topic in numerous computer science application areas. This fact is supported by exponential hardware upgrades and advanced computation techniques. However, there are still some open problems that require improvements. In particular, when multimodal environments are involved, the accuracy of sensors could be managed with enhancing methods. They are most commonly related to fusion techniques, however there are some exceptions for special or unusual cases. According to that, in this document we proposed a framework whose core system is based on temporal logic events' boost. It is divided into 3 different sections with specific tasks: the first module (Layout Builder) allows the administrator to plan the environment, creating a virtual clone of a real one. The second module (Rules Builder) is exploiting data received from the planimetry for granting the administrator to develop some temporal logic rules. It is based on a specific grammar and semantic that includes a wide range of possibilities. The third module actuates the rules, if their conditions are satisfied. It can integrate multiple autonomous functions, such as re-identification, gesture or speech recognition. A forecasting method is provided, exploiting a probabilistic technique over state machine theory. It is based on the occurrences of each event defined in rule's condition. The method works with transition matrix-like structures and updates their internal values at each step. The obtained scores are used as weights for improving events' probability provided by single functions, decreasing the ambiguity. The system is tested in a real environment with multiple sensors and related functions. The results are promising, proving that the probability of each event is correctly improved by the proposed method. We can underline that this technique can be integrated with complementary functionalities. In fact, according to the majority of similar works in literature, the contextualization of event is largely used and seems to provide great results. It could improve the performances of the proposed method due to the fact that there are no conflicts between them. At the same time, our system can be integrated in almost every multimodal framework for events management. So, we can say that the dynamism provided by the proposed method allows multiple appliances and improvements in future tasks and works.

Bibliography

- [1] Real-time american sign language recognition from video using hidden markov models. In *Proceedings of the International Symposium on Computer Vision, ISCV '95*, pp. 265–. IEEE Computer Society, Washington, DC, USA (1995). ISBN 0-8186-7190-4. Available from: <http://dl.acm.org/citation.cfm?id=525981.849918>.
- [2] ALLEN, J. F. Maintaining knowledge about temporal intervals. In *Readings in qualitative reasoning about physical systems*, pp. 361–372. Elsevier (1990).
- [3] ALLEN, J. F. AND FERGUSON, G. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4 (1994), 531. Available from: <http://dx.doi.org/10.1093/logcom/4.5.531>, [arXiv:/oup/backfile/content_public/journal/logcom/4/5/10.1093/logcom/4.5.531/3/4-5-531.pdf](http://arxiv.org/abs/10.1093/logcom/4.5.531), doi:10.1093/logcom/4.5.531.
- [4] ALTER, S. System interaction patterns. In *2016 IEEE 18th Conference on Business Informatics (CBI)*, vol. 01, pp. 16–25 (2016). doi:10.1109/CBI.2016.11.
- [5] ALUR, R. AND DILL, D. L. A theory of timed automata. *Theoretical computer science*, 126 (1994), 183.
- [6] ANDERSSON, V. O. AND ARAUJO, R. M. Person identification using anthropometric and gait data from kinect sensor. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15*, pp. 425–431. AAAI Press (2015). ISBN 0-262-51129-0. Available from: <http://dl.acm.org/citation.cfm?id=2887007.2887067>.
- [7] ASADI-AGHBOLAGHI, M., CLAPÉS, A., BELLANTONIO, M., ESCALANTE, H. J., PONCE-LÓPEZ, V., BARÓ, X., GUYON, I., KASAEI, S., AND ESCALERA, S. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pp. 476–483 (2017). doi:10.1109/FG.2017.150.
- [8] ATREY, P. K., HOSSAIN, M. A., EL SADDIK, A., AND KANKANHALLI, M. S. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16 (2010), 345.

- [9] AVOLA, D., SPEZIALETTI, M., AND PLACIDI, G. Design of an efficient framework for fast prototyping of customized human –computer interfaces and virtual environments for rehabilitation. *Comput. Methods Programs Biomed.*, **110** (2013), 490.
- [10] BANOS, O., GALVEZ, J.-M., DAMAS, M., POMARES, H., AND ROJAS, I. Window size impact in human activity recognition. *Sensors (Basel)*, **14** (2014), 6474. Sensors-14-06474[PII]. Available from: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4029702/>, doi:10.3390/s140406474.
- [11] BARBOSA, B. I., CRISTANI, M., DEL BUE, A., BAZZANI, L., AND MURINO, V. Re-identification with rgb-d sensors. In *First International Workshop on Re-Identification* (2012).
- [12] BEDAGKAR-GALA, A. AND SHAH, S. K. A survey of approaches and trends in person re-identification. *Image and Vision Computing*, **32** (2014), 270 . Available from: <http://www.sciencedirect.com/science/article/pii/S0262885614000262>, doi:<https://doi.org/10.1016/j.imavis.2014.02.001>.
- [13] BENGIO, Y., SIMARD, P., AND FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *Trans Neur Netw*, **5** (1994), 157.
- [14] BENNETT, B., COHN, A. G., WOLTER, F., AND ZAKHARYASCHEV, M. Multi-dimensional modal logic as a framework for spatio-temporal reasoning. *Applied Intelligence*, **17** (2002), 239.
- [15] BENNETT, B., COHN, A. G., WOLTER, F., AND ZAKHARYASCHEV, M. Multi-dimensional modal logic as a framework for spatio-temporal reasoning. *Applied Intelligence*, **17** (2002), 239. Available from: <https://doi.org/10.1023/A:1020083231504>, doi:10.1023/A:1020083231504.
- [16] BOULGOURIS, N. V., PLATANIOTIS, K. N., AND MICHELI-TZANAKOU, E. *Biometrics: theory, methods, and applications*, vol. 9. John Wiley & Sons (2009).
- [17] BOURGUET, M.-L. Designing and prototyping multimodal commands. In *Interact*, vol. 3, pp. 717–720. Citeseer (2003).
- [18] BRADSKI, G. AND KAEHLER, A. Opencv. *Dr. Dobb’s journal of software tools*, **3** (2000).
- [19] BRAND, M. Shadow puppetry. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1237–1244 vol.2 (1999). doi:10.1109/ICCV.1999.790422.
- [20] BROOKE, J. Sus: A retrospective. *J. Usability Studies*, **8** (2013), 29.
- [21] CHEN, Y. C., ZHU, X., ZHENG, W. S., AND LAI, J. H. Person re-identification by camera correlation aware feature augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40** (2018), 392. doi:10.1109/TPAMI.2017.2666805.

- [22] CHENG, G., WAN, Y., BUCKLES, B. P., AND HUANG, Y. An introduction to markov logic networks and application in video activity analysis. In *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*, pp. 1–7. IEEE (2014).
- [23] CHOUDHURY, T., ET AL. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Computing*, **7** (2008).
- [24] CLARKE, E. M., EMERSON, E. A., AND SISTLA, A. P. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, **8** (1986), 244.
- [25] COHEN, P. R., JOHNSTON, M., MCGEE, D., OVIATT, S., PITTMAN, J., SMITH, I., CHEN, L., AND CLOW, J. Quickset: Multimodal interaction for distributed applications. In *Proceedings of the fifth ACM international conference on Multimedia*, pp. 31–40. ACM (1997).
- [26] COMANICIU, D. AND MEER, P. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24** (2002), 603.
- [27] COSGUN, A., TREVOR, A. J., AND CHRISTENSEN, H. I. Did you mean this object?: detecting ambiguity in pointing gesture targets. In *10th ACM/IEEE international conference on Human-Robot Interaction (HRI) workshop on Towards a Framework for Joint Action* (2015).
- [28] CRISPIM-JUNIOR, C. F., BUSO, V., AVGERINAKIS, K., MEDITSKOS, G., BRIASSOULI, A., BENOIS-PINEAU, J., KOMPATSIARIS, I. Y., AND BREMOND, F. Semantic event fusion of different visual modality concepts for activity recognition. *IEEE transactions on pattern analysis and machine intelligence*, **38** (2016), 1598.
- [29] CRISPIM-JUNIOR, C. F., BUSO, V., AVGERINAKIS, K., MEDITSKOS, G., BRIASSOULI, A., BENOIS-PINEAU, J., KOMPATSIARIS, I. Y., AND BREMOND, F. Semantic event fusion of different visual modality concepts for activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38** (2016), 1598. doi:10.1109/TPAMI.2016.2537323.
- [30] DANTCHEVA, A., VELARDO, C., D'ANGELO, A., AND DUGELAY, J.-L. Bag of soft biometrics for person identification. *Multimedia Tools and Applications*, **51** (2011), 739. Available from: <https://doi.org/10.1007/s11042-010-0635-7>, doi:10.1007/s11042-010-0635-7.
- [31] DE, D., BHARTI, P., DAS, S. K., AND CHELLAPPAN, S. Multimodal wearable sensing for fine-grained activity recognition in healthcare. *IEEE Internet Computing*, **19** (2015), 26.
- [32] DELAC, K. AND GRGIC, M. A survey of biometric recognition methods. In *Electronics in Marine, 2004. Proceedings Elmar 2004. 46th International Symposium*, pp. 184–193. IEEE (2004).

- [33] ENZWEILER, M. AND GAVRILA, D. M. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (2008), 2179.
- [34] ESTER, M., KRIEGEL, H.-P., SANDER, J., XU, X., ET AL. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, vol. 96, pp. 226–231 (1996).
- [35] FARENZENA, M., BAZZANI, L., PERINA, A., MURINO, V., AND CRISTANI, M. Person re-identification by symmetry-driven accumulation of local features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2360–2367 (2010). doi:10.1109/CVPR.2010.5539926.
- [36] FEICHTENHOFER, C., PINZ, A., AND ZISSERMAN, A. Convolutional two-stream network fusion for video action recognition. *CoRR*, abs/1604.06573 (2016). Available from: <http://arxiv.org/abs/1604.06573>, arXiv:1604.06573.
- [37] FERNÁNDEZ, A., BERGESIO, L., BERNARDOS, A. M., BESADA, J. A., AND CASAR, J. R. A kinect-based system to enable interaction by pointing in smart spaces. In *2015 IEEE Sensors Applications Symposium (SAS)*, pp. 1–6 (2015). doi:10.1109/SAS.2015.7133613.
- [38] FUKUMOTO, M., SUENAGA, Y., AND MASE, K. “finger-pointer”: Pointing interface by image processing. *Computers & Graphics*, **18** (1994), 633 . Available from: <http://www.sciencedirect.com/science/article/pii/0097849394901570>, doi:[https://doi.org/10.1016/0097-8493\(94\)90157-0](https://doi.org/10.1016/0097-8493(94)90157-0).
- [39] GARCÍA, J., MARTINEL, N., MICHELONI, C., AND GARDEL, A. Person re-identification ranking optimisation by discriminant context information analysis. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1305–1313 (2015). doi:10.1109/ICCV.2015.154.
- [40] GARCÍA-MARTÍNEZ, S., ORIHUELA-ESPINA, F., SUCAR, L. E., MORAN, A. L., AND HERNÁNDEZ-FRANCO, J. A design framework for arcade-type games for the upper-limb rehabilitation. In *International Conference on Virtual Rehabilitation (ICVR)*, pp. 235–242 (2015).
- [41] GARGANTINI, A., TERZI, F., ZAMBELLI, M., AND BONFANTI, S. A low-cost virtual reality game for amblyopia rehabilitation. In *3rd Workshop on ICTs for Improving Patients Rehabilitation Research Techniques (REHAB)*, pp. 81–84 (2015).
- [42] GILL, A. *Introduction to the theory of finite-state machines*. McGraw-Hill electronic sciences series. McGraw-Hill (1962).
- [43] GONG, S., CRISTANI, M., YAN, S., AND LOY, C. C. *Person re-identification*. Springer (2014).

- [44] GUNA, J., JAKUS, G., POGAČNIK, M., TOMAŽIČ, S., AND SODNIK, J. An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking. *Sensors*, **14** (2014), 3702.
- [45] HARRINGTON, M. C. R. Empirical evidence of priming, transfer, reinforcement, and learning in the real and virtual trillium trails. *IEEE Trans. Learn. Technol.*, **4** (2011), 175.
- [46] HIRSCH, M. AND FARLEY, B. Exercise and neuroplasticity in persons living with parkinson’s disease. *European Journal of Physical and Rehabilitation Medicine*, **45** (2009), 215.
- [47] HOCHREITER, S. AND SCHMIDHUBER, J. Long short-term memory. *Neural Comput.*, **9** (1997), 1735.
- [48] HOLZINGER, A., SCHERER, R., SEEBER, M., WAGNER, J., AND MÜLLER-PUTZ, G. Computational sensemaking on examples of knowledge discovery from neuroscience data: Towards enhancing stroke rehabilitation. In *3th International Conference on Information Technology in Bio- and Medical Informatics (ITBAM)*, pp. 166–168 (2012).
- [49] HORN, B. K. AND SCHUNCK, B. G. Determining optical flow. *Artificial intelligence*, **17** (1981), 185.
- [50] HUANG, D. Y., HU, W. C., AND CHANG, S. H. Vision-based hand gesture recognition using pca+gabor filters and svm. In *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 1–4 (2009). doi:10.1109/IIH-MSP.2009.96.
- [51] IMANI, Z. AND SOLTANIZADEH, H. Person reidentification using local pattern descriptors and anthropometric measures from videos of kinect sensor. *IEEE Sensors Journal*, **16** (2016), 6227. doi:10.1109/JSEN.2016.2579645.
- [52] JAIMES, A. AND SEBE, N. Multimodal human–computer interaction: A survey. *Computer vision and image understanding*, **108** (2007), 116.
- [53] JAIN, A., NANDAKUMAR, K., AND ROSS, A. Score normalization in multimodal biometric systems. *Pattern recognition*, **38** (2005), 2270.
- [54] JAIN, A., NANDAKUMAR, K., AND ROSS, A. Score normalization in multimodal biometric systems. *Pattern Recognition*, **38** (2005), 2270 . Available from: <http://www.sciencedirect.com/science/article/pii/S0031320305000592>, doi:<https://doi.org/10.1016/j.patcog.2005.01.012>.
- [55] JORISSEN, P., WIJNANTS, M., AND LAMOTTE, M. Dynamic interactions in physically realistic collaborative virtual environments. *IEEE Trans. Vis. Comput. Graphics*, **11** (2005), 649.

- [56] KATO, N., TANAKA, T., SUGIHARA, S., SHIMIZU, K., AND KUDO, N. Trial operation of a cloud service-based three-dimensional virtual reality tele-rehabilitation system for stroke patients. In *11th International Conference on Computer Science Education (ICCSE)*, pp. 285–290 (2016).
- [57] KEHL, R. AND GOOL, L. V. Real-time pointing gesture recognition for an immersive environment. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pp. 577–582 (2004). doi:10.1109/AFGR.2004.1301595.
- [58] KEMÉNY, J. AND SNELL, J. *Finite markov chains*. University series in undergraduate mathematics. Van Nostrand (1960).
- [59] KEUS, S. H., MUNNEKE, M., NIJKRAKE, M. J., KWAKKEL, G., AND BLOEM, B. R. Physical therapy in parkinson’s disease: Evolution and future challenges. *Mov. Disord.*, **24** (2009), 1.
- [60] KIRILLOV, A. Aforge. net framework. Retrieved September 25th from <http://www.aforgenet.com>, (2013).
- [61] KNIGHT, A., CAREY, S., AND DUBEY, R. An interim analysis of the use of virtual reality to enhance upper limb prosthetic training and rehabilitation. In *9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 1–4 (2016).
- [62] KOHAVI, Z. AND JHA, N. K. *Switching and finite automata theory*. Cambridge University Press (2009).
- [63] KONDAXAKIS, P., GULZAR, K., AND KYRKI, V. Temporal arm tracking and probabilistic pointed object selection for robot to robot interaction using deictic gestures. In *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 186–193 (2016). doi:10.1109/HUMANOIDS.2016.7803276.
- [64] KWAKKEL, G., DE GOEDE, C., AND VAN WEGEN, E. Impact of physical therapy for parkinson’s disease: A critical review of the literature. *Parkinsonism & Related Disorders*, **13** (2007), S478.
- [65] LAHAT, D., ADALI, T., AND JUTTEN, C. Multimodal data fusion: An overview of methods, challenges, and prospects. *Proceedings of the IEEE*, **103** (2015), 1449. doi:10.1109/JPR0C.2015.2460697.
- [66] LALANNE, D., NIGAY, L., ROBINSON, P., VANDERDONCKT, J., LADRY, J.-F., ET AL. Fusion engines for multimodal input: a survey. In *Proceedings of the 2009 international conference on Multimodal interfaces*, pp. 153–160. ACM (2009).
- [67] LIU, J., SHAHROUDY, A., XU, D., CHICHUNG, A. K., AND WANG, G. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2017), 1. doi:10.1109/TPAMI.2017.2771306.

- [68] LUIS, M. A. V. S., ATIENZA, R. O., AND LUIS, A. M. S. Immersive virtual reality as a supplement in the rehabilitation program of post-stroke patients. In *10th International Conference on Next Generation Mobile Applications, Security and Technologies (NGMAST)*, pp. 47–52 (2016).
- [69] LUN, R. AND ZHAO, W. A survey of applications and human motion recognition with microsoft kinect. *International Journal of Pattern Recognition and Artificial Intelligence*, **29** (2015), 1555008.
- [70] MARTINEL, N., DAS, A., MICHELONI, C., AND ROY-CHOWDHURY, A. K. Re-identification in the function space of feature warps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37** (2015), 1656. doi:10.1109/TPAMI.2014.2377748.
- [71] MASOOD, S., SRIVASTAVA, A., THUWAL, H. C., AND AHMAD, M. Real-time sign language gesture (word) recognition from video sequences using cnn and rnn. In *Intelligent Engineering Informatics* (edited by V. Bhateja, C. A. Coello Coello, S. C. Satapathy, and P. K. Pattnaik), pp. 623–632. Springer Singapore, Singapore (2018). ISBN 978-981-10-7566-7.
- [72] MEHLMANN, G. U. AND ANDRÉ, E. Modeling multimodal integration with event logic charts. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pp. 125–132. ACM (2012).
- [73] MICHALSKI, R. S. Pattern recognition as rule-guided inductive inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1980), 349.
- [74] MILJKOVIC, D., ALEKSOVSKI, D., PODPEČAN, V., LAVRAČ, N., MALLE, B., AND HOLZINGER, A. Machine learning and data mining methods for managing parkinson’s disease. In *Machine Learning for Health Informatics: State-of-the-Art and Future Challenges*, pp. 209–220 (2016).
- [75] MITRA, S. AND ACHARYA, T. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **37** (2007), 311. doi:10.1109/TSMCC.2007.893280.
- [76] MOESLUND, T. B. AND GRANUM, E. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, **81** (2001), 231 . Available from: <http://www.sciencedirect.com/science/article/pii/S107731420090897X>, doi:<https://doi.org/10.1006/cviu.2000.0897>.
- [77] MUNROE, C., MENG, Y., YANCO, H., AND BEGUM, M. Augmented reality eyeglasses for promoting home-based rehabilitation for children with cerebral palsy. In *11th ACM/IEEE International Conference on Human Robot Interaction*, pp. 565–565 (2016).
- [78] MUNSELL, B. C., TEMLYAKOV, A., QU, C., AND WANG, S. Person identification using full-body motion and anthropometric biometrics from kinect videos. In *Computer Vision – ECCV 2012. Workshops and Demonstrations*

- (edited by A. Fusiello, V. Murino, and R. Cucchiara), pp. 91–100. Springer Berlin Heidelberg, Berlin, Heidelberg (2012). ISBN 978-3-642-33885-4.
- [79] NEVEROVA, N., WOLF, C., TAYLOR, G., AND NEBOUT, F. Moddrop: Adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38** (2016), 1692. doi:10.1109/TPAMI.2015.2461544.
- [80] NIELSEN, J. AND MOLICH, R. Heuristic evaluation of user interfaces. In *SIGCHI Conference on Human Factors in Computing Systems*, pp. 249–256 (1990).
- [81] NIGAY, L. AND COUTAZ, J. A design space for multimodal systems: concurrent processing and data fusion. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pp. 172–178. ACM (1993).
- [82] NISHI, K. AND MIURA, J. Generation of human depth images with body part labels for complex human pose recognition. *Pattern Recognition*, **71** (2017), 402.
- [83] NISHIDA, N. AND NAKAYAMA, H. Multimodal gesture recognition using multi-stream recurrent neural network. In *Image and Video Technology* (edited by T. Bräunl, B. McCane, M. Rivera, and X. Yu), pp. 682–694. Springer International Publishing, Cham (2016). ISBN 978-3-319-29451-3.
- [84] NISHIHARA, H. K., HSU, S.-P., KAEHLER, A., AND JANGAARD, L. Hand-gesture recognition method (2017). US Patent 9,696,808.
- [85] OAK, J. W. AND BAE, J. H. Development of smart multiplatform game app using unity3d engine for cpr education. *International Journal of Multimedia and Ubiquitous Engineering*, **9** (2014), 263.
- [86] OVIATT, S. Ten myths of multimodal interaction. *Commun. ACM*, **42** (1999), 74. Available from: <http://doi.acm.org/10.1145/319382.319398>, doi:10.1145/319382.319398.
- [87] OVIATT, S. AND COHEN, P. Perceptual user interfaces: Multimodal interfaces that process what comes naturally. *Commun. ACM*, **43** (2000), 45. Available from: <http://doi.acm.org/10.1145/330534.330538>, doi:10.1145/330534.330538.
- [88] PALA, F., SATTÀ, R., FUMERA, G., AND ROLI, F. Multimodal person reidentification using rgb-d cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, **26** (2016), 788. doi:10.1109/TCSVT.2015.2424056.
- [89] PEI, W., XU, G., LI, M., DING, H., ZHANG, S., AND LUO, A. A motion rehabilitation self-training and evaluation system using kinect. In *13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 353–357 (2016).

- [90] PELLECCIA, M. T., GRASSO, A., BIANCARDI, L. G., SQUILLANTE, M., BONAVITA, V., AND BARONE, P. Physical therapy in parkinson's disease: an open long-term rehabilitation trial. *J. Neurol.*, **251** (2004), 595.
- [91] PIGOU, L., VAN DEN OORD, A., DIELEMAN, S., VAN HERREWEGHE, M., AND DAMBRE, J. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, **126** (2018), 430. Available from: <https://doi.org/10.1007/s11263-016-0957-7>, doi:10.1007/s11263-016-0957-7.
- [92] PITSIKALIS, V., KATSAMANIS, A., THEODORAKIS, S., AND MARAGOS, P. *Multimodal Gesture Recognition via Multiple Hypotheses Rescoring*, pp. 467–496. Springer International Publishing, Cham (2017). ISBN 978-3-319-57021-1. Available from: https://doi.org/10.1007/978-3-319-57021-1_16, doi:10.1007/978-3-319-57021-1_16.
- [93] PLACIDI, G., AVOLA, D., FERRARI, M., IACOVIELLO, D., PETRACCA, A., QUARESIMA, V., AND SPEZIALETTI, M. A low-cost real time virtual system for postural stability assessment at home. *Comput. Methods Programs Biomed.*, **117** (2014), 322.
- [94] PROSSER, B., ZHENG, W.-S., GONG, S., AND XIANG, T. Person re-identification by support vector ranking. In *Proceedings of the British Machine Vision Conference*, pp. 21.1–21.11. BMVA Press (2010). ISBN 1-901725-40-5. Doi:10.5244/C.24.21.
- [95] QUINTERO, C. P., FOMENA, R. T., SHADEMAN, A., WOLLEB, N., DICK, T., AND JAGERSAND, M. Sepo: Selecting by pointing as an intuitive human-robot command interface. In *2013 IEEE International Conference on Robotics and Automation*, pp. 1166–1171 (2013).
- [96] RAMANARAYANAN, V., SUENDERMANN-OEFT, D., LANGE, P., MUNDKOWSKY, R., IVANOV, A. V., YU, Z., QIAN, Y., AND EVANINI, K. *Assembling the Jigsaw: How Multiple Open Standards Are Synergistically Combined in the HALEF Multimodal Dialog System*, pp. 295–310. Springer International Publishing, Cham (2017). ISBN 978-3-319-42816-1. Available from: https://doi.org/10.1007/978-3-319-42816-1_13, doi:10.1007/978-3-319-42816-1_13.
- [97] RAWAT, S., VATS, S., AND KUMAR, P. Evaluating and exploring the myo armband. In *International Conference System Modeling Advancement in Research Trends (SMART)*, pp. 115–120 (2016).
- [98] REGO, P., MOREIRA, P. M., AND REIS, L. P. Serious games for rehabilitation: A survey and a classification towards a taxonomy. In *5th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6 (2010).
- [99] REN, L., LU, J., FENG, J., AND ZHOU, J. Multi-modal uniform deep learning for rgb-d person re-identification. *Pattern Recognition*, **72** (2017), 446 . Available from: <http://www.sciencedirect.com/science/article/>

- pii/S0031320317302601, doi:<https://doi.org/10.1016/j.patcog.2017.06.037>.
- [100] RICHARDSON, M. AND DOMINGOS, P. Markov logic networks. *Machine learning*, **62** (2006), 107.
 - [101] RIGOLL, G., EICKELER, S., AND MULLER, S. Person tracking in real-world scenarios using statistical methods. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pp. 342–347 (2000). doi:[10.1109/AFGR.2000.840657](https://doi.org/10.1109/AFGR.2000.840657).
 - [102] RONZHIN, A. L. AND BUDKOV, V. Y. Multimodal interaction with intelligent meeting room facilities from inside and outside. In *Smart Spaces and Next Generation Wired/Wireless Networking* (edited by S. Balandin, D. Moltchanov, and Y. Koucheryavy), pp. 77–88. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). ISBN 978-3-642-04190-7.
 - [103] RUTA, D. AND GABRYS, B. Classifier selection for majority voting. *Information fusion*, **6** (2005), 63.
 - [104] SAINI, S., RAMBLI, D. R. A., SULAIMAN, S., ZAKARIA, M. N., AND SHUKRI, S. R. M. A low-cost game framework for a home-based stroke rehabilitation system. In *International Conference on Computer Information Science (ICCIS)*, pp. 55–60 (2012).
 - [105] SAK, H., SENIOR, A. W., AND BEAUFAYS, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Interspeech*, pp. 338–342 (2014).
 - [106] SCHROEDER, W. J., LORENSEN, B., AND MARTIN, K. *The visualization toolkit: an object-oriented approach to 3D graphics*. Kitware (2004).
 - [107] SEIDE, F. AND AGARWAL, A. Cntk: Microsoft’s open-source deep-learning toolkit. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2135–2135. ACM (2016).
 - [108] SEN, S. L., XIANG, Y. B., MING, E. S. L., XIANG, K. K., FAI, Y. C., AND KHAN, Q. I. Enhancing effectiveness of virtual reality rehabilitation system: Durian runtuh. In *10th Asian Control Conference (ASCC)*, pp. 1–6 (2015).
 - [109] SHIRATUDDIN, M. F., HAJNAL, A., FARKAS, A., WONG, K. W., AND LEGRADI, G. A proposed framework for an interactive visuotactile 3d virtual environment system for visuomotor rehabilitation of stroke patients. In *International Conference on Computer Information Science (ICCIS)*, pp. 1052–1057 (2012).
 - [110] SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. Real-time human pose recognition in parts from single depth images. In *CVPR 2011*, pp. 1297–1304 (2011). doi:[10.1109/CVPR.2011.5995316](https://doi.org/10.1109/CVPR.2011.5995316).

- [111] SINGH, D., MERDIVAN, E., PSYCHOULA, I., KROPP, J., HANKE, S., GEIST, M., AND HOLZINGER, A. Human activity recognition using recurrent neural networks. In *International Cross-Domain Conference on Machine Learning and Knowledge Extraction*, pp. 267–274 (2017).
- [112] SNELICK, R., ULUDAG, U., MINK, A., INDOVINA, M., AND JAIN, A. Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE transactions on pattern analysis and machine intelligence*, **27** (2005), 450.
- [113] SONG, Y. C., KAUTZ, H., ALLEN, J., SWIFT, M., LI, Y., LUO, J., AND ZHANG, C. A markov logic framework for recognizing complex events from multimodal data. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 141–148. ACM (2013).
- [114] SOSA, G. D., SÁNCHEZ, J., AND FRANCO, H. Improved front-view tracking of human skeleton from kinect data for rehabilitation support in multiple sclerosis. In *20th Symposium on Signal Processing, Images and Computer Vision (STSIVA)*, pp. 1–7 (2015).
- [115] STARNER, T., PENTLAND, A., AND WEAVER, J. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Trans. Pattern Anal. Mach. Intell.*, **20** (1998), 1371. Available from: <https://doi.org/10.1109/34.735811>, doi:10.1109/34.735811.
- [116] TRAN, D., BOURDEV, L. D., FERGUS, R., TORRESANI, L., AND PALURI, M. C3D: generic features for video analysis. *CoRR*, abs/1412.0767 (2014). Available from: <http://arxiv.org/abs/1412.0767>, arXiv:1412.0767.
- [117] TURAGA, P., CHELLAPPA, R., SUBRAHMANIAN, V. S., AND UDREA, O. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video technology*, **18** (2008), 1473.
- [118] TURAGA, P., CHELLAPPA, R., SUBRAHMANIAN, V. S., AND UDREA, O. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, **18** (2008), 1473. doi:10.1109/TCSVT.2008.2005594.
- [119] VAN KREVELEN, D. W. F. R. AND POELMAN, R. A survey of augmented reality technologies, applications and limitations. *International Journal of Virtual Reality*, **9** (2010), 1.
- [120] VAROL, G., LAPTEV, I., AND SCHMID, C. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40** (2018), 1510. doi:10.1109/TPAMI.2017.2712608.
- [121] VELARDO, C. *Anthropometry and soft biometrics for smart monitoring*. Theses, Télécom ParisTech (2012). Available from: <https://pastel.archives-ouvertes.fr/tel-01228653>.

- [122] WAN, Y., SANTITEERAKUL, W., CHENG, G., BUCKLES, B., AND PARBERRY, I. A representation for human gesture recognition and beyond. In *Computing, Communications and Networking Technologies (ICCCNT), 2013 Fourth International Conference on*, pp. 1–6. IEEE (2013).
- [123] WAN, Y., SANTITEERAKUL, W., CHENG, G., BUCKLES, B., AND PARBERRY, I. A representation for human gesture recognition and beyond. In *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, pp. 1–6 (2013). doi:10.1109/ICCCNT.2013.6726848.
- [124] WANG, J., WANG, Z., LIANG, C., GAO, C., AND SANG, N. Equidistance constrained metric learning for person re-identification. *Pattern Recognition*, **74** (2018), 38.
- [125] WANG, J., ZHU, X., GONG, S., AND LI, W. Transferable joint attribute-identity deep learning for unsupervised person re-identification. *CoRR*, abs/1803.09786 (2018). Available from: <http://arxiv.org/abs/1803.09786>, arXiv:1803.09786.
- [126] WASENMÜLLER, O. AND STRICKER, D. Comparison of kinect v1 and v2 depth images in terms of accuracy and precision. In *Asian Conference on Computer Vision (ACCV)*, pp. 34–45 (2016).
- [127] WEISSTEIN, E. W. Point-line distance–3-dimensional. (2002).
- [128] WU, A., ZHENG, W., AND LAI, J. Robust depth-based person re-identification. *CoRR*, abs/1703.09474 (2017). Available from: <http://arxiv.org/abs/1703.09474>, arXiv:1703.09474.
- [129] YAMATO, J., OHYA, J., AND ISHII, K. Recognizing human action in time-sequential images using hidden markov model. In *Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 379–385 (1992). doi:10.1109/CVPR.1992.223161.
- [130] YANG, X., WANG, M., AND TAO, D. Person re-identification with metric learning using privileged information. *IEEE Transactions on Image Processing*, **27** (2018), 791. doi:10.1109/TIP.2017.2765836.
- [131] ZHANG, X. Y., XIE, G. S., LIU, C. L., AND BENGIO, Y. End-to-end online writer identification with recurrent neural network. *IEEE Trans. Human-Mach. Syst.*, **47** (2017), 285.
- [132] ZHANG, Y., JI, Q., AND LU, H. Event detection in complex scenes using interval temporal constraints. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 3184–3191. IEEE (2013).
- [133] ZHANG, Z. Microsoft kinect sensor and its effect. *IEEE Multimedia*, **19** (2012), 4.

-
- [134] ZHU, G., ZHANG, L., SHEN, P., AND SONG, J. Multimodal gesture recognition using 3-d convolution and convolutional lstm. *IEEE Access*, **5** (2017), 4517. doi:10.1109/ACCESS.2017.2684186.

List of Figures

1.1	High level multimodal module architecture.	2
1.2	Complete logic architecture of a gesture and speech based multimodal system. The multimodal integration step is one of the most difficult and important phases to manage.	3
2.1	A general structure for systems analyzing human body motion.	7
2.2	General re-identification systems diagram.	10
3.1	Framework architecture, overview.	13
3.2	Allen's temporal logic algebra applied on events (X and Y). Each formula can be negated.	15
3.3	Actuator's logical data flow.	17
3.4	Association between depth map (images on the left) and body patches (images on the right right) generated during human body detection phase. Image from [110].	18
3.5	Joints map created by Kinect SDK 2.0.	19
3.6	Representation of a LSTM block having one cell	19
3.7	Logical architecture of the proposed gesture recognizer.	22
3.8	Pointing recognition module: logical architecture and it's communication method with the Gesture Recognizer. When the recognizer identifies a pointing action the target analysis starts. The process provides an item as output of the module. This information is exploited by Temporal Logic Module to verify the status of the system.	22
3.9	(a) Side view of sample 3D environment. The depth camera (white camera on the left) has a field of view defined by the transparent conic shape. The reference system is expressed according to the Y axis (green), X axis (red) and Z axis (blue) shapes. The sphere represents a sample object inside the field of view of the camera. (b) First person view (FPV) of the depth camera in the sample scene shown in figure a. The reference system is defined according to the X (red) and Y (green) axis of the image, width and height respectively. The Z axis (blue) is the depth one, indicating the distance of an object from the POV of the camera. Due to the fact that the environmental reference system is fixed, if the camera is rotated on Y axis, a coordinate translation method is required for calculating the real position of an object in the scene when it is identified only by the camera.	23

3.10	(a) Sample environment from top view. The camera is oriented according to $\theta = 0^\circ$. (b) Sample environment from top view in which a point should be translated. The camera is oriented according to $\theta = 0^\circ$. c Sample environment from top view in which a point should be translated. The camera is oriented according to $\theta = 45^\circ$	24
3.11	Sample scenario of pointing action from top view. The points E_p and W_p are the elbow and wrist of the user. Each element labeled with O is an object. The conic selection area is colored in red. O_1 and O_2 are the candidate objects.	26
3.12	Re-identification module: logical architecture. The dataset block is used with temporary data registered during the first capture when the re-identification function is invoked. On the contrary, permanent information are stored in depth data dataset for the identification and the RGB frames are ignored.	29
4.1	Allen's temporal logic algebra and the symbolism applied in the proposed system.	35
4.2	Temporal scheme of possible events that can occur in the following conditions: Condition 1 (4.6), Condition 2 (4.7) and Condition 3 (4.8). The vertical axis corresponds to the temporal intervals divided in windows (w). The horizontal axis denotes the parallelism of each potential event.	40
5.1	Layout Builder: planimetry designer.	50
5.2	Layout Builder: overview of 3D editor for managing items in the scene.	50
5.3	Layout Builder: 3D editor in detail. The items can be dropped in any allowed location and can also be resized and oriented.	51
5.4	Layout Builder: JS items models (three.js json format) created with Blender's exporter 2.7.	51
5.5	Layout Builder: addable items list. The modularity of the system allows to create new items in any moment according to the needs.	52
5.6	Layout Builder: output data file. Exported information are related to walls and items, with all their variables.	52
5.7	Rules Builder: GUI of the system. The user should follow the syntax proposed on the top of the image for creating rules. The buttons are dynamically generated based on the Layout Builder output and previously registered users. For using the reidentification features, the person can be parametrically specified.	53
5.8	An overview of the proposed architecture. The patient side (left) highlights the used devices, while therapist side (right) points out the framework adopted to create, to monitor, and to customize the different serious games	56
5.9	Combination of the Kinect and Leap Motion Controller models. Point A belongs to the Leap Motion Controller model, while point B belongs to the Microsoft Kinect model. The computed point C represents the best anchor point for combining the two models	57

5.10	Deep learning architecture. For each serious game defined in the system, a RNN-LSTM is used to estimate the patient's performance. The data are acquired by the Kinect and Leap Motion Controller during the execution of the exercise and they are also supplied to the network (Sensor Data). Then, the latter processes the received data (Data Processing) and provides the patient's performance with respect to a set of healthy subjects (Patient's Impairment Estimation)	59
5.11	LSTM-RNN accuracy with respect to the number of layers: (a) shows the accuracy obtained by using the same number of epochs for training, (b) shows the accuracy obtained by increasing the epochs for 5 and 6 layers	59
5.12	Number of days needed for training the network with respect to the number of layers. The blue line is the time needed when the number of epochs is fixed to 800, while the orange line is the time needed when the epochs are augmented to 1600 and 2000 for a 5 layers and 6 layers network, respectively	60
5.13	VE of the exercise 1. The bigger image shows the point of view of the user while performs the raise the knee and pinch with your fingers exercise	61
5.14	VE of the exercise 2. The bigger image shows the point of view of the user while performs the march and grasp exercise	62
5.15	VE of the exercise 3. The bigger image shows the point of view of the user while performs the get up on heels or toes and cut with hands exercise	62
5.16	Patients' performances of the exercise 1 during rehabilitation sessions	64
5.17	Patients' performances of the exercise 2 during rehabilitation sessions	65
5.18	Patients' performances of the exercise 3 during rehabilitation sessions	66
5.19	Pointing action performed by the user and recognized by the system. The pointed object is the camera itself.	67
5.20	Planimetry (a) and top view (b) layout of the room used for testing the pointing recognizer. On the right wall, barely visible, there is a microphone.	68
5.21	CMC of proposed method on generated dataset.	70
5.22	CMC of proposed method and Ren's one[99] on KinectREID dataset.	70
5.23	Planimetry of the testing environment. On the right, Room1, on the left, Room2. The entrance is on south wall of Room1.	72
5.24	Top view of the testing environment with objects. In the image the communication door is not visible due to a prospective occlusion. However, it is located on the west wall of Room1 and on the east wall of Room2.	72
5.25	Side 3D view of the testing environment. (a) the point-of-view is watching from south-east to north-west. (b) the point-of-view is watching from north-west to south-east.	73
5.26	Panoramic photo of the real Room 2 environment. As shown, there is also some not relevant furniture. However, it does not influence the experiment.	73
5.27	User entering in Room 2.	74

5.28	Frames from "greet" sequence.	75
5.29	The user performing a pointing action.	75
5.30	The user performing a sitting action on the chair.	76
5.31	The user performing a sitting action on the sofa.	76
5.32	Recognition rate for each event based on the proposed experiment. We can see that the majority of errors are evident in isolated events, that are not involved in disambiguation functions.	82
5.33	Recognition rate for each event based on the proposed experiment after disabling the probability module. Results are worst on ambiguous events.	82

List of Tables

2.1	Main characteristics of state-of-the-art systems	9
3.1	Skeleton features for gesture recognition.	21
4.1	Correlation between temporal windows, event validity and Allen's temporal operations. In particular the Time-To-Live (TTL) of the events is expressed in iterations, that corresponds to a temporal window. In some cases is impossible to define the sliding windows and the TTL of an event a priori, due to the fact that the temporal operation can involve multiple contiguous windows.	37
4.2	Transition matrix of conditions (4.6) (4.7) and (4.8) generated by the proposed algorithm.	41
4.3	Transition matrix of conditions (4.6) (4.7) and (4.8) updated by the proposed algorithm.	43
4.4	Transition matrix of conditions (4.6) (4.7) and (4.8) updated according to the algorithm 4.3 if (4.6) and (4.8) are linked and (4.6) is satisfied.	44
5.1	Features used to estimate patients' performance with respect to the performed exercise	61
5.2	Exercise 1: Average data collected from patients	61
5.3	Exercise 1: Average data collected from healthy subjects	62
5.4	Data collected from patients during the execution of the exercise 3 and their personal judgements	63
5.5	Comparison between the capabilities of the proposed system and similar systems at the state-of-the-art	63
5.6	Comparisons between the average estimation proposed by the therapists and the system, respectively, for the exercise 1	64
5.7	Comparisons between the average estimation proposed by the therapists and the system, respectively, for the exercise 2	65
5.8	Comparisons between the average estimation proposed by the therapists and the system, respectively, for the exercise 3	65
5.9	Accuracy results collected during point recognizer module tests. This table is referring to the first position of the users, in the middle of the room.	68
5.10	Accuracy results collected during point recognizer module tests. This table is referring to the second position of the users, in front of the door.	69

5.11	Accuracy comparison between the proposed method and state of the art ones on the entire KinectREID dataset.	71
5.12	Results collected for each user and run. The percentage value shows the accuracy of each rule condition completion based on the times that the user tries to complete them performing the required events.	80
5.13	Results collected for each user and run disabling the proposed probabilistic module. Some ambiguous events are mistaken and the related rules' conditions are not satisfied as well.	81
5.14	Features comparison between proposed method and the most similar works in literature.	83

List of Algorithms

3.1	Person identification pseudo-code algorithm	28
4.1	Probability assignment for the first transition matrix. It is important to consider that each delay, named timer interval ti , can generate time shifts of events.	39
4.2	Probability update for transition matrix. It is important to consider that each delay, named timer interval ti , can generate time shifts of events.	42
4.3	Probability update for transition matrix after a rule condition is satisfied.	44
4.4	Potential events probability update based on transition matrix.	45
4.5	Potential events probability update based on transition matrix.	47