Gesture-Based Natural Interaction with Smart Environments

Diogo Matos, Gonçalo Silva, Mauro Filho, Inês Santos,
Tiago Silvestre, Nuno Vidal

Oriented by Dr.ª Ana Rocha, Prof. Samuel Silva
Supervised by Prof. José Moreira
aprocha@ua.pt, sss@ua.pt, jose.moreira@ua.pt

UA

Gesture-Based Natural Interaction with Smart Environments

DETI

UA

Diogo Matos, Gonçalo Silva, Mauro Filho, Inês Santos,

Tiago Silvestre, Nuno Vidal

12/06/2023

Abstract

The importance of Human-Computer Interaction (HCI) is undeniable. This field is widely studied, new researches are always thriving to make the human interaction with a computational system more natural and accessible. A lot of those systems are specially developed to be deployed in people's homes. And, as consequence to the success of mass-marketed technologies, like Apple Home Kit [1], Alexa from Amazon [2], or Google Home [3], which make the interaction with the house devices seamless, smart and simple. A special need for a deeper exploration on the improvement of those kinds of systems emerged. Most of them are voice-based and bring both situational and accessibility issues, for example they cannot be used in noise sensitive/loud environments or by people with communication difficulties. Hybrid systems both based on speech and gestures seem to accommodate most of the people's needs, but the use of cameras for gesture recognition is not ideal for home usage since it is very susceptible to privacy concerns (no one wants a camera monitoring us, when peacefully at home). In this report, we propose a new radar-based gesture recognition system constituted by a gesture input modality coupled with a home interaction ecosystem that is capable of controlling the house (change the state and obtain information from a selection of devices). The system follows a Multimodal Architecture and Interface (MMI), to facilitate improvements and addition of new modules. A Frequency-Modulated Continuous-Wave Radar (FMCW) radar is used to capture gestures, which will be recognized using a Machine Learning (ML) model built using a previously collected dataset built by us and Transfer Learning (TL). Following a user-centered design, we studied our system's target users by defining personas and scenarios, from which we extracted requirements. Subsequently, through the execution of a Focus Group, we delved deeper into analyzing the users' needs, providing an opportunity to fine-tune the requirements and gain a more tangible comprehension of the specific tasks, gestures, scenarios, and overall system requirements. With this deeper knowledge we developed, as part of the home interaction ecosystem output modality, a virtual home to prove the system's feasibility and to show the its complete interaction.

Keywords: gesture recognition, natural human-computer interaction, smart homes, ambient sensors, radar, "transfer learning", "user-centered development"

Report contents

Acronyms

FMCW Frequency-Modulated Continuous-Wave Radar

HCI Human-Computer Interaction

AI Artificial Intelligence

ML Machine Learning

HGR Hand Gesture Recognition

SFCW Single-Frequency Continuous-Wave Radar

CNN Convolutional Neural Networks

TL Transfer Learning

CW Continuous-Wave

IF Intermediate Frequency

RDM Range-Doppler Map

DT Doppler-Time

RT Range-Time

UCD User Centered Design

API Application Programming Interface

kNN k-Nearest Neighbors Algorithm

SVM Support Vector Machine

LSTM Long Short-Term Memory

IM Interaction Manager

FE Fusion Engine

SSE Server Sent Events

MMI Multimodal Architecture and Interface

DBSCAN Density-Based Spatial Clustering of Applications with Noise

GIM Gesture Input Modality

TLV Tag Length and Values

Chapter 1

Introduction

1.1 Context

Since the dawn of technology the concept of house is increasingly becoming smarter. Who would have though that we could interact with our house in so many ways, and the field is yet to be fully explored. We can turn on the lights without being at home, have an audio system answering our questions, regulate the heat with touching a dial and so much more. However there is still a great opportunity to improve the way we interact with our homes.

1.2 Motivation

Modern technologies that allow us such interactions rely heavily on voice, screens and camera inputs. But as the field is more explored and more accessible, the need for new approaches arises. The existing interactions raise some situational problems, for example, if we are in a loud environment and want to control the house by speech, our voice recognition systems will probably be unable function properly. What if we are at the sofa and due to our mobility issues, we are not capable of reaching the home interaction device. Combining these issues, there are situations where people with poor mobility and/or verbal communication problems, such as people suffering from Aphasia, are not capable of these luxuries. We should strive to provide an alternative in the HCI realm and this is precisely why systems like the one proposed in this work were initially created, i.e., - to facilitate people's lives.

Of course, solutions based on RGB cameras revolutionised the field, because they solve most of the problems mentioned earlier and more, but other ones emerge, such as their dependence in good lighting conditions and privacy issues. Nevertheless, some kind of hybrid system that both relies on voice and gestures seems to be the way to go in the future. Since the first type of system (voice) allows for a cheaper and simpler setup, products like Alexa from Amazon are densely commercialized. Activity and gesture recognition using RGB cameras is nothing new too, and can be seen deployed in a lot of places, with various

purposes. But there must be a reason why we usually do not see such systems inside houses. Actually, the reason is quite simple - an RGB camera is still a threat for us, regarding privacy and is more than obvious that no one wants it inside their home, especially in more sensitive divisions such as the bedroom and bathroom.

1.3 Goal

Our main goal is to introduce a new way of interaction with smart environments, mainly smart homes both by addressing the accessibility and intrusiveness issues of systems based on voice and/or gadgets(tablets, wristbands, etc) and the privacy issues of camera-based systems. Our proposal introduces a human gesture-based input system that enhances the natural interaction between individuals and smart environments, specifically within a smart home setting. This system leverages gestures as the primary input method, incorporating a Gesture Input Modality designed to discern intentional hand gestures using environmental sensors, notably radars. This modality should be designed in such a way that is able to be integrated with third-party home interaction ecosystems. As proof-of-concept, we will be implementing a prototype of the system that not only includes the demonstrator, but also the a home ecosystem simulating a virtual home.

1.4 Report Structure

This report commences with an overview of the current state-of-the-art, providing insights that support the selection of technologies and methods employed in the development of our solution. Next, we delve into Requirements Elicitation, outlining the process and outcomes of gathering the necessary requirements. Furthermore, we discuss the methodology employed in identifying key interactions and gestures deemed essential for implementation in the final version of the system.

Subsequently, we present our proposed system, encompassing the generic system architecture and detailed descriptions of each module. The report progresses from defining personas to the prototype implementation, with a significant focus on the outcomes of the Focus Group sessions, requirements, and results of the gesture recognition model.

Concluding the report, we offer a concise summary of the project, along with our reflections on its implications for the future.

Chapter 2

Background and State-of-the-art

As our living and working spaces (homes, offices, stores, cars, etc.) become smarter, various means of interacting with them have emerged. Voice (e.g., smart speakers) and text/touch (e.g., smartphones, tablets) have been common methods, but they come with limitations. Background noise can hinder voice commands, and individuals with mobility difficulties may struggle with touch-based devices.

To address these limitations, human gesture-based input systems offer a more natural and inclusive approach. By using environmental sensors like radars, these systems recognize and interpret purposeful hand gestures, allowing for effortless interaction.

Popular assistants like Alexa [2], Google Assistant [3], Siri [1], and Bixby [4] have enabled voice control in smart homes.

Nevertheless, ongoing research is exploring alternative natural interaction methods that have the potential to serve as viable replacements or complements to the aforementioned approaches, particularly in specific contexts or for certain users.

2.1 Human Gestures

Interaction with gestures implies Gesture Recognition using data gathered from sensors, for that, we first need to understand what are the body parts most commonly used as a link between the human and the computer. It is easy to forget that a lot of the communication among humans is nonverbal [5].

In Figure 2.1 from [6], we can see the body parts that can be used as a communication vehicle in HCI. This study [6] shows that more than 40% of contributions the studies use the hand. That includes a single hand, both hands, or just the fingers. Undoubtedly, our hands constitute a big part of our day-to-day interactions. So, since HCI should be as natural as possible, hand-related movements seem to be accepted as the way to go when developing new nonverbal HCI systems.

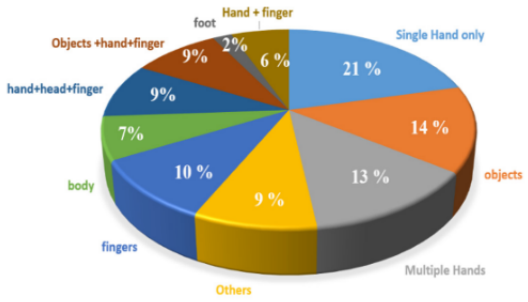


Figure 2.1: Usage of different body parts for human-computer interaction to make a human–computer interface [6]

2.2 The Trends for Gesture Recognition

Based on the study [7], which reviews contributions in the area of the vision-based hand gesture recognition field, the number of studies in HCI is increasing and people are starting to realize that gestures are an important part of the way we manifest ourselves. Solutions based on Hand Gesture Recognition (HGR) are increasing as an alternative to the more traditional ways of interaction, such as speech, or the basic computer interfaces we have nowadays. However, researchers are aware that, for HGR to be possible, there are some challenges that need to be solved during the development of this solutions. Mainly:

- How are the gestures going to be detected?
  - Using a wearable device might be a solution, but that is not ideal. We humans do not like to be carrying a lot of things.
- Are we going to use a camera?
  - That levitates towards privacy concerns.

Researchers have been exploring with these questions in mind and gravitating towards the use of radar sensors to solve these issues. Since 2009, when one of the first studies was made with the radar in this context, the number of articles regarding this subject has increased more than ten times. For example, in 2019, at least twelve article about HGR using radars were published. Even though the field is quite recent, some trends have developed. By analysing them, we can make sure not to fall in the "traps" other have fallen. On the other hand, it shows what is yet to be explored. That being said, bellow is a list of some of those trends referenced in [6] the [6] study:

- Use of FMCW or Single-Frequency Continuous-Wave Radar (SFCW) radars. Both together appears to be the chosen in more than 60% of the studies

- The most common machine learning algorithms for gesture classification are k-Nearest Neighbors Algorithm (kNN) and Support Vector Machine (SVM). With deep learning, Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) are also used.
- Usually full-hand, with no finger movement motions, are considered when selecting gestures.
- The vast majority of the studies explore single-hand movements. Almost no research was made on detecting movements with both hands.

2.3 Sensors for Gesture Recognition

Since the objective we are trying to achieve is the use of a technology that fits the scenario of a smart home and is adequate in terms of low intrusiveness, the Radar is a perfect contender to the throne. No one can feel safe in their homes if there is a third party system with a an RGB camera installed, but a Radar can make us achieve our objectives without this problem, so the user can feel safer. Also, as mentioned above, the trends all point to the direction of Radar-based systems in this scenario.

2.3.1 The Radar and the mmWave Radar

A radar is a detection system that uses radio waves to determine the distance (range), angle, and radial velocity of objects relative to the device. In this project, we are going to dig deeper into the realm of mmWave radar. The mmWave radar is a special type of radar that, like mentioned, uses electromagnetic waves. However, this type of technology uses a short wavelength in the millimeter range, which is capable of detecting movements as small as a fraction of a millimeter, with high accuracy [8], [9].

2.3.1.1 Doppler Effect

If we want to understand radars and more specifically mmWave radars, we need to understand what is the Doppler effect. The Doppler effect is used by all radar types, to detect the change in frequency of a wave in relation to an observer who is moving relative to the wave source [10]. In simpler terms, when something moves towards you, the frequency of the sound waves increases. The opposite is also true, if the object moves away from you, the frequency of the waves decreases. The radars rely on it in order to work.

2.3.1.2 Types of Radars

All radars adhere to the same functional philosophy, but there exist various types of radars. Among them, the following radars are the most commonly used in gesture recognition, according to [6]:

- Pulsed Radar
- Continuous-Wave (CW) Radar
  - Frequency-Modulated CW (FMCW) - mmWave Radar
  - Single-frequency CW (SFCW)

2.3.1.3 Pulsed Radars

Pulsed radars transmit an impulse signal with a wide frequency spectrum. Usually they have a smaller size, allowing for more portable hardware. There have been studies carried out using this kind of radar to recognize hand gestures, some even design a customized pulsed radar for the goal. Nevertheless, this type of radar is not used much in the gesture recognition context and the following radar type seems the trend when dealing with this [6].

2.3.1.4 Continuous-wave radars: SFCW and FMCW

SFCW radars sense the target based on the change in frequency of the transmitted and received pulses. With the target moving, the Doppler effect shifts the signal frequency. That is true in the case of any movement, including a hand.

FMCW radars transmit varying frequency signals. Starting with a low frequency signal and, as time passes by, the frequency is increased until a predefined threshold is reached. The signal then travels to the target and part of it is reflected back. The radar will detect the reflected signal and compare it to the original one by mixing them and processing the resulting signal [6].

2.3.1.5 Differences Between Radar and Other Technologies

The radars, they have some key advantages/disadvantages compared to other technologies such as optical cameras. Table 2.1 shows some key differences between these two technologies.

	Radar	Optical camera
Price	usually higher	usually lower
Visible Light	Cannot differentiate	Can differentiate different light sources
View hidden objects	Given that radar waves have low frequency, they can easily penetrate objects	Cannot view covered occluded objects
Useful in dark places	Yes	No
Useful in dark places	Yes	No
Speed and distance of object	Yes	No

Table 2.1: Key differences between radar and optical cameras [11].

2.3.2 Key takeaways

We see that the Radar is an adequate choice for our project because of its discreet and versatile nature. The trends point to a direction using Radar-based systems in gesture recognition solutions using primarily a FMCW radar. Based on these points, the tool to be used by our gesture input modality for gesture recognition is a FMCW radar, namely a AWR1642 by Texas Instruments since it is easily accessible and it is used in two articles that we analysed in the Related Work on Radar-Based Gesture Recognition.

2.4 Related Work on Radar-Based Gesture Recognition

In order to better understand what has been made in the field of gesture recognition using a radar sensor, wWe searched papers using Google Scholar with keywords such as "gesture recognition", "radar", "human-computer interaction", and "machine learning". The most recent(2014 onwards) and similar to our desired system were selected. Based on the articles we reviewed, we noticed that there are different ways of achieving the same goal. In this section, some of those articles are briefly presented.

2.4.1 Studies Summary

Table 2.2 summarizes the papers that were read and analyzed more carefully. From the table, we conclude that the vast majority of studies use deep learning and the systems support gesture recognition at short distances up to 2 meters. The number of gestures that most studies explored, varied between 3 and 5 gestures. The accuracy between studies cannot be directly compared since the data acquisition conditions and validation methods

vary from each other. However, they are usually above 85%. In the next sections, we are going to explain each of this contributions more carefully.

Study and year	**Radar type**	**Algorithm**	**Distance between hand and camera**	**Number of gestures**	**Accuracy** [1]
Gesture Recognition for Smart Home Applications using Portable Radar Sensors (2014) [12]	Continuous-Wave (CW)	Supervised learning (kNN)	2 m	3	95%
TS-I3D Based Hand Gesture Recognition Method With Radar Sensor (2019) [13]	FMCW	Deep Learning (One 3D neural network and 2 two CSTM networks)	Unknown but short	10	96.17%
Radar-based gesture recognition towards supporting communication in aphasia: The bedroom scenario (2022) [14]	FMCW	Transfer Learning	1 m	3	$\geq 90\%$
Time-Space Dimension Reduction of Millimeter-Wave Radar Point-Clouds for Smart-Home Hand-Gesture Recognition (2022)	FMCW	CNN	3 m	5	$\geq 87.1\%$
Cross-modal Learning of Graph Representations using Radar Point Cloud for Long-Range Gesture Recognition (2022)	FMCW	DGCNN	1-2 m	5	98.4%

Table 2.2: Synthetic comparison between the main studies referred in this chapter.

[1]Different studies used different methodologies and evaluation conditions (although all of them were conducted on a controlled environment). Therefore, higher accuracy values may not necessarily indicate better results.

2.4.2 Gesture Recognition for Smart Home Applications using Portable Radars

In article [12], the authors present a gesture recognition system using a CW portable radar (coupled with AAA batteries).

Feature extraction and classifier: The feature extraction process was possible using Principal Component Analysis (PCA). The authors also considered feature extraction based on physical attributes like the speed of the gesture and the direction of it. Furthermore, the classifier algorithm used was kNN.

Accuracy: They achieved an accuracy of 95% (10-fold cross-validation). Although they had a high accuracy, it was evaluated only on three gestures, hand pushing, hand lifting, and head shake. The distance between the user and the radar was around two meters.

2.4.3 TS-I3D Based Hand Gesture Recognition Method with Radar Sensor

In article [13], the authors highlight the limitations and challenges of existing hand gesture recognition methods and then describe their proposed TS-I3D based approach as a solution to address those limitations. Below, very briefly, we present the methods used.

Data collection and processing: A FMCW radar (the same model that we are going to use) is the source to collect hand gesture data. The range and speed of the continuous hand gesture is calculated by 2D-FFT algorithm based on the Intermediate Frequency (IF). Range-Doppler Map (RDM) is generated according to the relationship among range, speed and the frequency of IF signal. The peak interference including the step peak interference in single-frame RDM and the static peak interference caused by the standing body and the arm in multi-frame RDM are filtered. To end, Wavelet transform is applied for image enhancement of the gesture peak in RDM. Each gesture is represented by a 32-frame RDM.

Feature extraction and and classifier: A 3D neural network is designed to extract RDM features and generate Range-Time (RT) feature sequence and Doppler-Time (DT) feature sequence. Then, two more LSTM networks are employed to extract temporal gesture information from the RDM. At last, the extracted RT and DT features are concatenated and classified using a softmax layer.

Dataset and accuracy: 10 kinds of gestures were collected, 400 times each. The average recognition accuracy was 96.17% with a simple cross-validation. The dataset was taken from a single person standing still around 1 m from the radar.

2.4.4 Radar-Based Gesture Recognition Towards Supporting Communication in Aphasia: The Bedroom Scenario

In article [14], the authors applied a FMCW radar for use in bed, specifically oriented to people with communication disorders, such as Aphasia. For gesture recognition, transfer learning was used.

General architecture and layout: A FMCW captures data from the human body, sends the data to the processing unit, where they are pre-processed by removing outliers. Features are then extracted and used to recognize the gesture and the decision is sent to a smartphone. The Radar is positioned on the left side of the bed, where it roughly matches the height of the bed (0.55 m from the ground) and placed at 1 m from the bed, parallel to its longest side.

Data processing: All the various types of data acquired by the FMCW radar are processed using a sliding window of 5 s without overlap. For each window, pre-processing consists of removing outliers corresponding to unwanted reflections or noise. A detected target is considered to be an outlier if its distance to the radar is not in the [0.5, 3] m range or its absolute Doppler index is outside of the interval [1e-5, 10]. All data samples with X and Y coordinates outside of the intervals [-1.5, 1.5] m and [0, 2.25] m, respectively, are also discarded.

From the filtered data, three maps are created, one for each type of data versus elapsed time. The beginning and ending of the window where no movement is detected are discarded.

The resulting images from feature extraction are fed into a model that performs gesture recognition. This model was previously trained for three gestures, using transfer learning, relying on pre-trained deep neural network model for image classification and a given dataset.

Dataset, gesture recognition models, and evaluation: The dataset had a size of 150 images, 50 for each gesture. However, to obtain better performance and avoid overfitting, the authors used offline data augmentation, generating 5-10 new images, from each image, just by adding noise. The noise combined the Gaussian, salt, pepper, and Poisson types.

Since the aim was to run train model on a limited-capability device, with a relatively small dataset, TL was used. Each model was evaluated using an adapted 10-fold cross-validation approach, where 80% of the dataset is used for training, 10% for validation, and 10% for testing, in each cross-validation iteration. It achieved an accuracy equal or superior to 90%.

Conclusions reached: The research showed a good result on recognizing a small set of three gestures, made by a single user, in the in-bed scenario. However, with a larger dataset

corresponding to more users and gestures, should be possible to train a model to perform more broadly.

2.4.5 Time-Space Dimension Reduction of mmWave Radar Point-Clouds for Smart-Home Hand-Gesture Recognition

This study [15] tackles a big problem, and an important aspect that we have to take in consideration in our work. In a similar way to this article, we are trying to make a modality that is capable of detecting and recognizing human gestures in a Smart-Home environment. However, we come to a problem that is distance and the capacity of the Radar based system to detect human gestures at a long distance. In this study, they come to a solution to this problem.

Time spectrum for Point Clouds: To represent Human gestures, such as gestures that vary with time and are going to create multiple point clouds, they create time spectrum features to represent the multiple point clouds of a gesture. The point clouds used are detected and extracted from the original radar spectrum, and then passed to a point cloud time spectrum feature. This process filters radar sidelobes and weak multi-path inference. Because the radar is limited by low angular resolution, meaning that it has trouble with larger configured field of view, the researchers use a multi-position and multi-feature fusion learning with a multi-channel CNN type of neural network to improve the performance of spatial positions in these scenarios.

Multi-location and Multi-Dimensional Feature Learning: Because Deep learning requires high computational cost, or has poor anti-interference performance, in this study they used a lightweight version of CNN with three 2D convolution layers. In this way, the real-time gesture recognition performance is greatly improved.

Gesture Dataset: For this study, 5 gestures were selected. The Wave up, Wave down, Wave left, Wave right and, push gestures were be performed by 3 different users. To train the model, one of the training users, user A, performs each gesture 25-35 times in each position, such positions in -30º, -15º, 0º, 15º, and 30º angles from the radar and at the distances of 1 m, 2 m and 3 m. Then to test the system, trained user A, and 2 untrained users, B and C, will perform each gesture 25-40 times, at the same distances and at -20º, -15º, -10º, 0º, 10º, 15º, 20º angles of the radar. It is noted that the training was made with the user looking at the radar direction and performing the gestures with their right hand.

Accuracy: In terms of performance, they conclude that the best point-cloud time spectrum features are the XTA (X-Time-Amplitude), YTA (Y-Time-Amplitude) and ZTA (Z-Time-Amplitude) all combined. For each user A, B and C, they could on average classify their gestures with 99.7% , 91.1% and 87.1% accuracy respectively.

2.4.6 Cross-modal Learning of Graph Representations using Radar Point Cloud for Long-Range Gesture Recognition

Like the previously presented study, this study [16] is in the context of Radar point clouds and long-range gesture recognition. In this study, they use a 60-GHz FMCW Radar to create the point clouds and create a new architecture capable of detecting Human gestures from the 1-2 m range. They mention, like we also concluded, that most studies explore the Radar in the Gesture Recognition realm only in limited scenarios where the person performing the Gestures is very close to the radar.

The Architecture: Like mentioned, the researchers explored a new architecture to ensure the possibility of detecting gestures from a longer range than other colleagues tried before with the radar technology for this purpose. To that end, they came to a solution where they used normal cameras to help the algorithm in the learning process. To make this, they setup 2 cameras in a room alongside a radar. What the cameras achieve is the position of the person in the room by creating a camera point cloud, with the some key points of human physique, registering where the person is in the room and where the hands and other parts of the body are, so when the radar captures a movement, the model is capable of identifying where the movement comes from and what it is more easily. Then, a radar point cloud is generated and is fed to the learning algorithm alongside the camera point cloud.

The Learning Algorithm: Because now the input modalities are 2, a way to learn from these 2 sources is needed. For that, the researchers used a Cross learning Algorithm based on a graph neural network (DGCNN). This way, they feed both the camera information and the radar information, and with the Encoder, an element of the learning architecture, they can merge information and classify the movement.

Gesture Dataset and Performance: To test their implementation, they chose the gestures that were more relevant and referenced in other studies and gestures that are simpler. They or chose a hand swipe to the sides, a push movement where the hand goes from the body in direction to the radar, a pull movement where the hand comes from the radar in direction to the body, and finally a clockwise movement and anti-clockwise movement with the hand. Then, 5 volunteers trained the model with these gestures from the distances ranging 1 meter to 2 meters, alternating the left and right hand. Then to evaluate the model similar actions were performed by other 5 volunteers but in a different room and radar orientation setup to demonstrate the proposed architecture's generalization capability. The final benchmarks show a result of 98.4% accuracy in gesture recognition.

2.5 Conclusion and key points

Based on the State-of-the-Art and related work, we know that Human gestures are very important for communication between individuals and hand gestures are a very natural way of interpretation by other people.

We conclude that our solution of a smart-home interaction using gestures is possible. There are many studies, some of which we mentioned here, that work in this scenario using Radar technology achieving great results. However, none of them really came to a finished product, ready to be used. So, there is an important part on this project that is making sure we develop a solution in a modular way, which can have switchable parts to be adapted and ready to use in any situation.

Chapter 3

Requirements Elicitation

Following a User Centered Design (UCD), the system should suit the needs of the user. In such an early phase, it was hard to make a list of requirements following those bases, because dealing with people is always tricky and it takes time. Since time does not wait for us, and development needed to start as early as possible, we had to find a way to overcome this problem without forgetting the UCD philosophy. So, in the first stage, we put ourselves in the skin of the user and, as unbiased as possible, we tried to find out what the end users might want from our system. To do that, we created personas and a scenario for each one of them. These were the basis to extract the system's requirements.

Since the beginning, was our intention to make perform a focus group where we bring together a number of people to discuss certain key aspects of our project, such as "Is the purpose of the project viable?", "What are the 'pains' when interacting with a smart home?" and "What movements feel more natural to people when wanting to communicate a specific action?". Questions as those cannot be asked directly, answers need to be built fractionally. There is a lot to take into account when assembling this kind of meetings. After all, we are in the human atmosphere, not in the machine where everything predetermined. Since the project could not wait for this focus group to be prepared and carried out, it was executed alongside the system's development. However, this approach also means that the requirements initially may suffer changes during development.

In this chapter, we present a summary of what we studied about UCD, then a view of various elicitation methods and information about focus groups carried out by others. Finally, we describe our personas and present the results obtained from the focus group, both of which contribute to defining the requirements.

3.1 User Centered Design

In HCI, the more capable the system is, the better it suits the user's needs. The human, as it is an animal, most particularly a intelligent being, is very unique. Everyone is different

and, because of that, we struggle to communicate with each other. But fortunately, we have the capacity to adjust to our surroundings, so that we can understand what someone like us mean. All of this happens effortlessly, it is a natural thing for us. But for a computer, where all is very deterministic, that cannot be achieved with that much ease.

For a HCI solution to be successful, both the human and the computer need to "speak" the same language and expect something from the other. Humans like to predict everything. The more predictable something is, the more the persons feels comfortable interacting with it.

So, to develop this project, it is crucial to use an approach that is focused on the user. For that, we are going to follow a user-centered design UCD [17] framework. This approach is based on the understanding of a user, their demands, priorities, and experiences. When used, it is known to lead to an increased product usefulness and usability, as it delivers more satisfaction to the user.

In Figure 3.1, we can see the four basic steps of UCD. They are a representation of the philosophy behind this type of design that is based on the following pillars/"rules"[17]:

- Make it easy to determine what actions are possible at any moment;
- Make things visible, including the conceptual model of the system, the alternative actions, and the results of actions;
- Make it easy to evaluate the current state of the system;
- Follow natural mappings between intentions and the required actions, between actions and the resulting effect, and between the information that is visible and the interpretation of the system state.

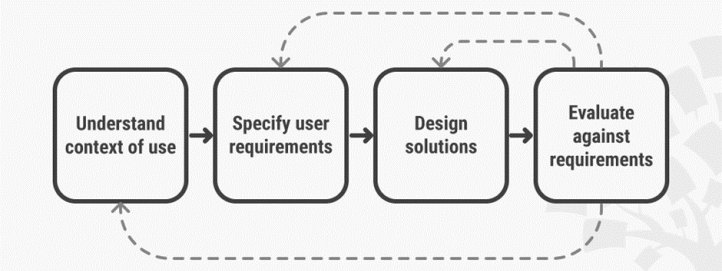


Figure 3.1: Main steps of the iterative UCD process [18]

.

3.2 Elicitation

UCD and Elicitation methods are closely related because both involve gathering information about the users of a product or service, including their needs, preferences and behavior.

This information helps us create a product that is tailored to our target audience's needs and more likely to be successful.

3.2.1 Possible Elicitation Methods

As per the definition, Elicitation or Requirement Elicitation is a process used to determine the needs of customers and users, so that systems can be built with a high probability of satisfying those needs. It is a well-know opinion that the success or failure of a system's development effort depends heavily on the quality of the requirements.

In [19], an in-depth interview with experts in elicitation analysis is presented. What is interesting is that the authors of the paper themselves conducted a elicitation study while interviewing the experts. The selected participants were purposefully selected to have the broadest possible coverage to elicitation techniques, domains and situations. In this case, individual interviews were conducted, but they could have been group discussions, if the subjects' opinions or experiences matched. We want to highlight some of the methods referenced in the paper, which were based on the opinions, experience of the interviewed, and our project needs.

- **Collaborative sessions** is when multiple stakeholders are gathered in a single room to conduct a group discussion. Experts say that it should be considered and almost always conducted.
- **Interviewing** is recommended to gather background information when working on new projects in new domains. Another use is when a group participant has a heavy or completely different opinion than the others in the group discussion. In that way, it is easier to guarantee that the discussion tends to productive topics.
- **Team building** should be especially used when the participants do not know each other and need to debate ideas together. In that way, participants can try to know and be more comfortable talking/discussing topic with other participants.
- **Ethnography** or observation/recording of users interaction should always be conducted.
- **Issues list** is a list of outstanding issues on the side, where they can be worked or delayed and new issues can be added.
- **Models** such as data flow diagrams, time-ordered sequences of events to capture system interaction, user role models, task model and others. They serve as an easy way to define and create requirements as well as simulating the purpose or architecture of the systems. However, they should not be used as final customer documentation.

- **Questionnaires** were one of the first methods that we thought when doing requirements elicitation. However, the experts consider its use very limited, mostly to concrete problems or understanding customer needs. Which means that it is not a good method for us to apply.

- **Role Playing** was mentioned as a possible use when the stakeholders are too busy or unavailable. In our case, they can be useful to define personas or user stories, which help us to stay focused on customer needs and develop easy-to-use products, from a customer perspective.

In the concluding notes, the paper referred the need to always identify users and other stakeholders first and then to find a spokesperson for each, if possible [19]. A general advice was to always ask, do not tell nor assume. The main consensus between the specialists was that the use of stories in elicitation is powerful and applicable most of the times.

3.2.2 Elicitation Study Example

In [20], the authors researched the best user-defined gestures for touch screen interaction, specially in a big screen. For that, they selected a small group of individuals and conducted interviews with each member. Their main selection method was guaranteeing that the participants had similar demographic categories, did not use touch screen devices, and did not have Computer Science or related knowledge. This way, constraints of reliability, feasibility and other are removed, it is just user intuition in play.

The authors of the paper reference a previous article's findings, which show that gestures are most commonly used for executing simple, direct, physical commands, while speech is used for a high level or abstract commands.

Choosing non-technical users improves user-centered design and human-computer interaction, but that does not mean that every gesture or feature should be implemented. That is why care must be taken to elicit applicable user behaviour.

The authors also mentioned the importance of feedback. As humans, our conversations are mostly based on context and feedback. So, a system that serves as a bridge, to sit in between a environment and the user, should always provide feedback. That way, the user can adapt or learn better ways to interact with it, if needed.

Based on our research, we think that our focus group should be composed of up to ten participants, preferably people with no technical knowledge. That way, we can better design our system to be suitable for user interaction in different kind of environments, with the most diverse people.

3.2.3 Focus Group: Article Analysis

This method was preferred over other approaches because it allows for the gathering of rich, in-depth data from a diverse group of individuals who have been carefully selected

to represent the target users. This type of data is particularly valuable when it comes to understanding the needs, wants, and expectations of potential users, as it provides insight into their attitudes, behaviors, and motivations. Additionally, the focus group setting allows for the facilitation of discussions and the exploration of ideas and opinions in a way that is more natural and less structured than other methods. This allows for the uncovering of new and unexpected information that can be used to inform the design and development of the product or service. In a project like ours, where it is essential to have a great understanding of our core audience and their thoughts, all of the reasons mentioned above are key to the successful execution of the work.

3.2.3.1 What is a Focus Group?

The authors of [21] offer an introduction to the Focus Group methodology and some of its main aspects through the analysis and review of several concrete case studies previously conducted by investigators of the methodology and researchers who chose to obtain results through its application.

The paper defines a Focus Group as "an informal discussion among selected individuals about specific topics". The handling of focus group sessions generally involve one, or more, group discussions with a collective focus upon the study's subject. This focus is generally presented by the researcher in the form of questions, films, games, or any other experience that serve as a guide to the discussion.

When it comes to selecting the people who will partake in a specific session, one of the main challenges of handling Focus Group studies (more on that later), it is recommended that the size of the group stays around 6-8 people (very rarely more than 12) and that the people involved in a certain session will be able to communicate and understand each other's points of view, therefore facilitating their discussions and allowing the researcher to observe the evolution of their opinions as the session goes on. Generally, that means choosing people who share personal aspects, such as similar experiences, age, social-economic situation, or fields of expertise, for example.

Discussions during sessions are usually recorded (or videotaped) and later transcribed into text, which will then serve as the main resource for getting conclusions out of study. This method is distinctive for its data-collection procedures, which focus on extracting interactions out of the group, instead of solely focusing on getting quantitative data. This interaction focus also distinguishes the Focus Group from other kinds of methods, like one-on-one interviews, which do not offer room for elaboration of ideas, discussions and other kind of interactions among the group members.

3.2.3.2 How Can it be Useful for Research?

The same paper [21] mentions three main ways in which Focus Groups have been used throughout history: as an adjunct to other methods as a part of a multi-method research;

as a primary research method; as a form of participatory action research.

1. *As an adjunct to other methods:* usually Focus Groups are used alongside other, commonly quantitative, research methods. The use of Focus Groups allows for conducting a research based on a sensitive understanding of the matter being considered.

2. *As a primary research method:* As a standalone research method, focus groups can be used either to find new areas of research within the study subject, or to further explore already-known areas from the participants perspectives.

3. *As a form of participatory action research:* In this case, many researchers who have used Focus Groups suggest that the methodology is particularly useful to access opinions of people who have been poorly served by the traditional lines of questioning (e.g., people who struggle to formulate thoughts and feel pressured by individual interviews or questionnaires).

3.2.3.3 When is a Focus Group Appropriate?

The focus group methodology has many advantages over traditional methodologies, one of the main ones being its flexibility and consequential wide range of use. When talking about the method's flexibility.

Besides the advantages to the methodology, Focus Groups also present a number of disadvantages, including limited reliability and validity, and various forms of biases from either the moderators or the participants. In addition, general data analysis is often proven to be inefficient for treating the outputs obtained.

There are three key points to be considered before deciding to choose this methodology for your research, and we will explore the three of them in the following sections:

1. *Purpose of research:* Focus groups are an appropriate choice of method when the objective of the research is to elicit people's understandings, opinions and views, or to explore how these are advanced, elaborated and negotiated in a social context.

2. *Type of output desired:* As previously mentioned, one of the disadvantages (and also advantages) of a Focus Group is the data it generally produces. While this kind of data can be helpful for qualitative analysis, it is usually hard to take quantitative conclusions out of it.

3. *Practical aspects:* Although Focus Groups can be considered easy and straightforward to handle, the preparation of a session entails a large amount of effort. From creating questions and training a moderator, to preparing a physical room and ensuring your participants show up to the session.

3.2.3.4 Key Features: What Can You Get from a Focus Group?

In this section, we shall highlight the three key features of Focus Groups, which make them stand out in comparison to similar elicitation methods, like one-on-one interviews. All three of them are derivative from the interactive nature of the group sessions and the data it produces.

1. *Providing access to participants' own language, concepts and concerns:* The relatively free discussion among participants of group sessions offers an opportunity to hear the language used by the demographic being subjected to this methodology when talking about the study object. This way, the researcher can become familiar with the way participants usually approach the topic at hand, and gain an insight on commonly assumed assumptions that constitute and inform participants.

2. *Encouraging the production of more fully articulated accounts:* In focus groups, people tend to disclose opinions and views on the world that they would not typically disclose in a different setting. Therefore, the insights gained by watching the discussion are usually closer to the truth and much more articulated, allowing researchers to explore the topic with a greater understanding of it after each session.

3. *Offering an opportunity to observe the process of collective sense-making:* When talking in the group session, the tendency is that people try to get to a collective sense of their individual beliefs by interacting and challenging each other's views. This interaction constitutes the primary data extracted from this methodology.

3.3 Personas and Scenarios

To prepare for the Focus Group, we took the necessary steps of establishing Personas and Scenarios. The purpose was to understand our target audience better, gather valuable insights and define requirements. We developed four personas that embody a diverse family unit, spanning across different age groups and individual needs. This approach allows us to create a system that accommodates users with varying ages and requirements, ensuring usability for all.

Each persona possesses unique motivations and needs, all of which align with the objective of facilitating a more intuitive and seamless interaction with their environment. To illustrate the potential applications of our interaction system, we devised specific scenarios for each family member. These scenarios showcase the different ways in which individuals from this family could engage with our system, taking into account their distinct characteristics and motivations.

Our aim is to build a system that can cater to a wide range of people, encompassing various demographics and individual needs. By developing four personas that represent a diverse family, we can effectively address the varying requirements of our potential users.

3.3.1 Personas

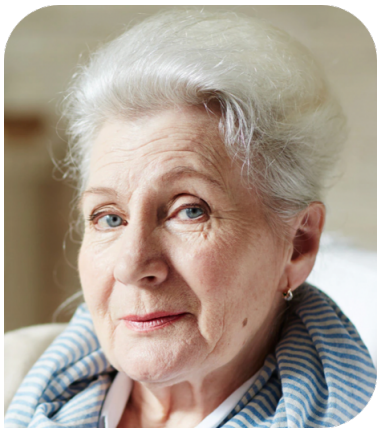
Mark, 42

- **Description:** Mark is a 42-year-old sales consultant, and also Isabela's father. When he is at home, Mark enjoys his free time by caring for his house and family. Mark is very skilled at manual work, so he is always looking for something to repair or to improve around the house. When repairing things in the house, it is not uncommon for Mark to get his hands dirty, which is not an unexpected outcome of manual work. However, when he is at home, all kinds of things can happen. Not being able to touch anything without making a mess is a huge pain, since he will either have to wash them before being able to act, or to clean everything he touches along the way.
- **Motivation:** Mark wants to be able to interact with his house without making a mess with his hands.

Isabela, 5

- **Description:** Isabela is a 5-year-old kid who loves to do all sorts of recreational activities when she is at home. She is a curious and smart kid who is on a journey of her own to discover the world. Although her explorer spirit is shining through, some tasks are still not achievable to her. Like for example reaching the light switches around the house, which makes her dependent on her father to do those simple, yet impossible (to her), tasks.
- **Motivation:** Isabela wants to act more independently from her family around her house.

Abigail, 78



- **Description:** Abigail is a 78-year-old. As a retired person, Abigail has lots of free time to enjoy at home. However, due to her age, her body does not function the way it used to anymore. She cannot walk very long distances and her eyesight is worse than it was. Tasks like standing up to open the blinds, or finding small objects around the house, can become nightmares due to her age.
- **Motivation:** Abigail wants to be able to accomplish her tasks easier without having to do too much physical effort.

John, 19

- **Description:** John is a 19-year-old student at University. He is Mark's oldest son. As a student, John is always eager to learn and is up-to-date with all the newest technological trends that come up online. As a student who is usually under pressure, John finds it important to find the time to take care of his mind and body in between his academic duties. For that reason, he likes to enjoy his nights by watching movies on his TV. The problem is that by being at home, John is susceptible to being forced out of his movie session at any minute in order to do tasks like answering the door or helping his family.
- **Motivation:** John wants to be able to accomplish his tasks without interrupting his movie sessions for too long.

3.3.2 Scenarios

3.3.2.1 Mark

Baking a cake: It's Isabela's birthday, and Mark is baking a cake to celebrate. Since he is not a very experienced chef, he is following a video tutorial that's playing on his TV in the kitchen. Mark starts playing the video and get his hands dirty with the cooking. Whenever he gets confused with the recipe, Mark needs to pause the video to think about what he needs to do next. In order to achieve that, Mark does a gesture and the video pauses. When he is ready to keep watching, he uses another gesture and the TV goes back to playing. By doing that, he is able to control the video and successfully celebrate his daughter's birthday.

3.3.2.2 Isabela

Early Morning cartoons: It's Saturday morning and Isabela is having a big bowl of cereals at the kitchen table. There is a TV in the kitchen, high up near the cabinets, in which she enjoys watching her favorite morning cartoons. Unfortunately the TV is too high for Isabela to reach and, as a 5-year-old kid, she still isn't proficient using the remote. However, that's not a problem for her at all. By using a couple of gestures, she is capable of turning the TV on and then navigate the channels until she finds her desired one. This way, Isabela can have her breakfast and enjoy the cartoons at the same time.

3.3.2.3 Abigail

Watching the sunrise: Abigail wakes up in her bedroom at 6 am, besides her husband. Since she woke up so early, she wishes to watch the sunrise from her window. However, her blinds are closed and she feels a bit weak to stand up immediately and open them. Knowing the physical challenges faced by his mom, Mark has tried to introduce devices to his mom's daily life, so she could accomplish tasks around the house with minimal effort. However, those devices were never a good fit for her needs. Smartphones were unsuccessful due to how complex they can get at times, and she recently dropped the blinds' remote and didn't manage to go down to pick it up, which makes it more of a hassle then a solution. So, while still in bed, Abigail makes a gesture and the blinds start opening, while she gets comfortable in bed again to enjoy her view.

3.3.2.4 John

Movie night: Friday is everyone's favorite day, and its no different when it comes to John. This Friday, he decided to get rid of all his work-related gear and watch a new action movie that was released some time ago. During the movie, it starts getting too cold and John wishes to turn up the heating of the room. The remote is far away, and issuing a voice command would be hard with all the noise coming from the TV. So in order to turn

up the heater, John does a gesture and accomplishes his task. Now, he can keep watching his movie in a comfortable temperature.

3.3.3 Alternative Solutions Considered for Scenarios

Table 3.1 shows an analysis of the usage of other popular existing ways of interacting with smart devices at homes, comparing with gestures, in the context of our scenarios. As we can conclude from the table, in the scenarios being considered for this analysis, all of the other interaction methods have big disadvantages, which makes gestures the best way of achieving the desired results.

3.4 Focus group

3.4.1 Hosting

Our focus group took place at IEETA on march 31st of 2023, the aim of the session was to collect comprehensive data on people's perspectives concerning gesture interaction in smart environments. The objective is to ascertain which tasks are best suited for gesture-based execution and to identify the gestures that are most natural for the proposed tasks. Furthermore, the focus group intends to observe and evaluate the participants' opinions on the usage of gestures in smart environments. By analyzing the feedback from the focus group, we hope to gain a better understanding of the effectiveness and usability of gesture-based interaction in smart environments. Ultimately, this knowledge can be utilized to design intuitive interfaces that cater to user requirements and preferences.

The focus group session included 7 participants who were recruited through personal connections. All participants were students at the University of Aveiro, ranging in age from 18 to 21 years old, selected based on their willingness to participate in the group discussion. The group was diverse in terms of gender (5 female, 2 male) and academic background, with representation from different faculties within the university (1 tourism, 1 languages and culture, 1 physics and 4 within computer sciences).

Our focus group relied on the presence of at least 3 moderators, two that were actively leading the discussion, and another one that would play the support role. The support moderator, while also being able to contribute to the discussion, played a major role in being video-taped during the sessions. Since recording videos of participants can be troublesome and make some of them unsure about their will to contribute to the research, and also taking into account that one of our objectives is to gather body gestures proposed by the participants, the support moderator would mimic the gestures showcased by the participants while being recorded, as a form of documenting our findings during the session.

The session was split in 4 sections, that added up to a total of 1 hour of duration:

- Introduction: 5 min

User scenario	**Existing physical solutions**	**Smartphones**	**Voice commands**	**Video cameras**
Early morning cartoons	TV remote is not made to be easy for a kid to browse	Isabela does not have a smartphone	She needs to keep it quiet in order to not wake her family up	For privacy reasons, having a camera film a child all the time might be an issue
Watching the sunrise	Incompatible with Abigail's physical conditions	Abigail does not have a smartphone	She needs to keep it quiet in order to not wake her husband up	The lack of light may trouble the usage of some types of cameras
Baking a cake	Require him to touch physical devices with his dirty hands	Mark cannot use his phone with his dirty hands	Mark's baking equipment is noisy and added to the video sound, could be hard to issue successful voice commands	Having a camera film your house all the time may be a privacy issue
Movie night	The remote is far away	John does not have his phone on him at that time	The TV sound may trouble issuing successful voice commands	Having a camera film your house all the time may be a privacy issue, and lack of light may trouble the usage of some types of cameras

Table 3.1: Analysis of alternative interaction methods, including the reasons why methods different from gestures for interaction with smart homes are not compatible with our scenarios.

– Introduce the objectives of the focus group, engage with the participants, and sign the informed consent.

- Warm-up Discussion: 15 min
  - Start discussing open-ended questions related to interaction with smart environments with no focus on gestures.
- Main Discussion: 35 min
  - For this stage, our goal was to find out what are the possible scenarios where people would like to see gesture interaction being applied in their daily life, and also what would be the most natural and intuitive ones to achieve such tasks. For that, participants were asked to propose objects that they could see themselves interacting with in a house, and then propose what the interaction would be (what task they would be achieving) and what would be the best gesture to accomplish that.
- Conclusion: 5 min
  - Answer any final questions the participants may have and thank them for participating.

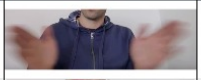
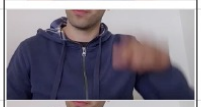
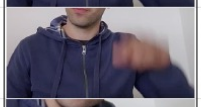
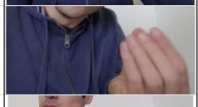
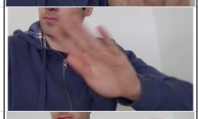
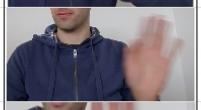
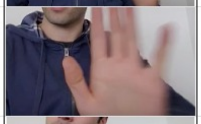
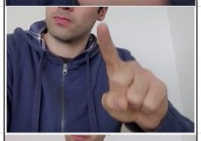
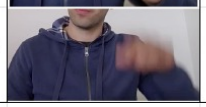
3.4.2 Conclusions

The results of this discussion are organized in the Figure 3.2, for better visualization and analysis:

As we can observe, the devices proposed for the gesture interaction that were received positively by the participants were generally devices that could belong anywhere in the house: doors, lights, TVs, Air conditioning, blinds and curtains.

It is also clear from the table that, for most devices, there was a "continuous" kind of interaction, meaning an interaction that requires real-time decisions from the user and constant action to achieve a desired result (such as picking the temperature for the shower, or opening the blinds just the right amount). However, even though those interactions were mentioned multiple times, the final perception was always negative, as the participants failed to think of intuitive and concise ways of achieving their goals. The proposed solutions were either faulty from the start, or too confusing to understand exactly how they worked, which would be crucial for finer adjustments.

We can also observe that 100% of the interactions generally approved, or received positively by the participants, were gestures that fall under the "Binary" category (interactions that only have two states that are opposite to one another), with gestures that are opposite to each other in order to take the device to the opposite state. This might be because of how simple and constantly done they are. Being actions that we make multiple times on a daily basis, makes people wanna optimize those experiences, and because they are so simple, and usually have a certain characteristic motion associated with them, it is easier to translate them into a gesture.

Device	Task	Category	Gesture	Positive
Curtains	Open/ Close	Binary		☑
Blinds	Open/ Close	Binary		☑
Blinds	Control height	Continuos input		☐
Lights	Turn on/ off	Binary		☑
Lights	Control intesity	Continuos input		☑
Doors	Open/ Close	Binary		☑
Doors	Lock/ Unlock	Binary		☐
Air conditioning	Turn on/ off	Binary		☑
Air conditioning	Control temperature	Continuos input		☐

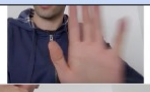
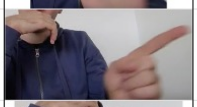
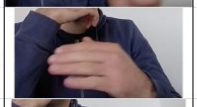
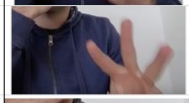
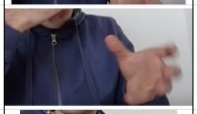
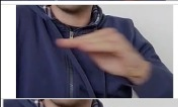
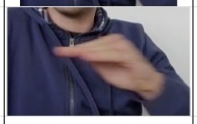
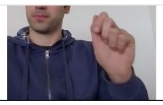
Device	Task	Category	Gesture	Positive
TV	Turn on/ off	Binary		☑
TV	Navigate menu of features	repeated commands		☐
TV	Navigate channels	repeated commands		☑
TV	Choose channel	Manual input		☐
TV	Volume	Continuos input		☑
Tap	Turn on/ off	Binary		☐
Tap	Control water temperature	Continuos input		☐
Toilet	Flush	One-time action	Automatic after leaving the place	☐
Gesture recognition system	Activate gesture recognition	Binary		☑

Figure 3.2: Focus Group Findings

On the other hand, complex interactions such as browsing features on a smart TV, or using a smart gesture-based toilet, cast doubt on the participants about how needed the change really was, which led them into questioning if the gesture interaction would even make the experience easier or just more time-consuming. In those cases, another problem that came up during the discussion was that if not all steps of an interaction can be done through gestures, you might as well not use gestures at all, leading us to the conclusion that complex interactions should probably be left out of the scope of the project for the moment. Using the TV as an example, if you will have to use the remote for accessing a specific feature at some point, turning it on with gestures is pointless in some cases.

Still talking about the said "complex interactions", we also gathered that for those scenarios, and other similar ones that were not mentioned during the Focus Group, the analogical ways of interaction should be maintained for some reasons that were brought up by the participants:

- Gesture interaction is not suitable for every scenario, since the environment can change very quickly inside one's house.
- Interacting manually with devices could be quicker and easier in some scenarios. For instance, if a person is standing right next to the light switch, it is easier to just press it rather than doing a gesture.
- Guests at one's home will most likely not know how to interact through gestures, unless they are mass-adopted
- In emergency situations the reliability of analogical interaction is preferred
- We cannot take for granted our ability to perform said gestures forever. There may be a time when we can no longer perform gestures and there will be the need to adapt to that (or in some cases, just go back to the old way)

This last point brings us to another important topic that was mentioned during the session. The participants recognized that the future is uncertain and that our ability to perform gestures may change during the course of time, and therefore, there is the need to adapt to a new reality. One way that they envisioned to do that is by having a gesture interaction system that is highly customizable and allows users to set their own gestures into it. This feature was already one of the objectives of our project, but having the participants mention it, without us even hinting at this possibility, reinforced the need and benefit of the feature.

One of the biggest controversies during the session was actually introduced very early on, and followed us until the very end of the conversation: False positives. From the beginning, participants were already worrying about how to avoid that the gestures would be identified at moments when the user was not intentionally performing a gesture with the intention of interacting with the house. The two main solutions that came up during the conversation were:

- Create an unnatural activation gesture, that would require the user to chain the activation gesture with the one that he intended doing in the first place.
- Making all gestures unnatural and uncommon, so you would hardly do it by accident or casually without intention to interact with the house.

Both options have disadvantages, the first one would make the process of performing a gesture more lengthy and less natural, while also requiring the interaction system to provide feedback (audio or visual were suggested) to the user in order to let them know that the recognition has been activated and gestures are being captured. On the other hand, the second one would make the process quicker but also sacrifice the "intuitive" part of the system completely, by making all gestures uncommon to the point of never accidentally doing them.

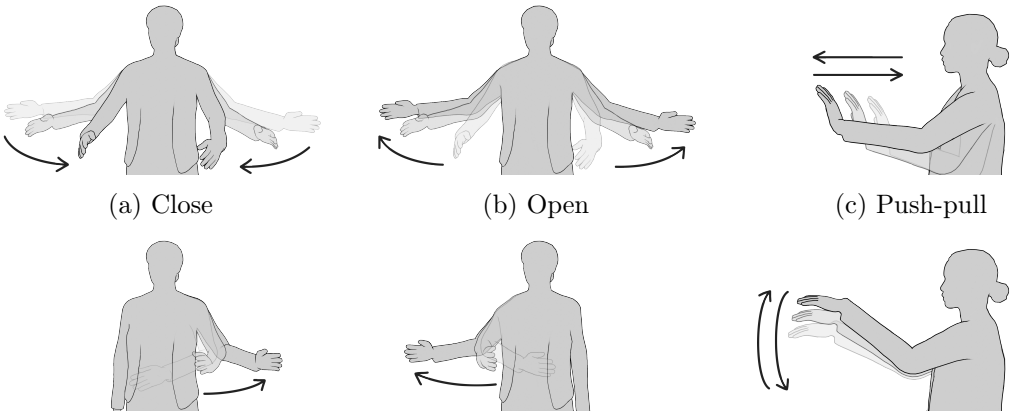
Even with the disadvantages, the first option was preferred by the participants and basically adopted as the standard solution for the problem during the stage of proposing gestures.

Lastly, another concern that arose from the participants was about how to identify which object they were interacting with. This concern came to be towards the second half of the discussion, after we had already discussed gestures for some scenarios. That happened because eventually the gestures began to mix, and the participants no longer thought it was productive to come up with different gestures for each device, but rather define gestures for actions, and then select where they would like to apply said action.

For example, we can turn on a TV, a rice cooker, a light, and even an air conditioner. For the participants, it made sense that they could use the same "turn on" gesture for all devices, just select what their target was somehow. The solution that was proposed from this was that the interaction system should recognize what device the user is facing/looking at and then use that as a selection.

3.4.3 Selected Gestures and Scenarios

After the analysis of the results gathered from the focus group, we came to the selection of the bedroom and the living room as the chosen environments for the further development of our project. That is due to the fact that all interactions with general approval from our participants could take place in those two environments (although not only). We also selected some of the well-received objects as the ones we would be interacting with, those being the television, curtains, blinds and air conditioning, as those would be a good match with the binary-type of action that was preferred by the participants. For the gestures, we also gathered some of the most commonly mentioned during the session to be a part of our final implementation of the project: swipe left, swipe right, swipe up, swipe down, open arms, close arms.



(a) Close (b) Open (c) Push-pull

(d) Swipe-left (e) Swipe-right (f) Up-down

Figure 3.3: Depiction of gestures selected from the focus group

3.5 Requirements

In order to satisfy the scenarios defined above and the pre-conditions of our project, some functional and non-functional requirements were gathered. The following tables (3.2, 3.3) provide a description of each requirement along with its purpose. Additionally, each requirement is further explained in the text below.

3.5.1 Functional requirements

Requirement	Description
FR-1	The system should enable gesture-based interaction between the user and the home, by providing a gesture input modality.
FR-2	The gesture modality should rely on an ML model that recognizes a set of gestures using data captured by sensors deployed at home.
FR-3	Gestures must be recognized in all lighting conditions.
FR-4	The system architecture must be modular.
FR-5	A prototype of the system should be implemented to demonstrate the house state and the interactions between the user and the system.
FR-6	The system should not depend on the use of wearables by the user.
FR-7	The system must inform the user (through feedback) when a gesture is detected.
FR-8 (Optional)	Allow customization of the gestures

Table 3.2: Functional requirements defined for our system.

- FR-1: Our main goal is to develop a gesture input modality. This requirement is essential because the whole system is based on the input provided by the user.
- FR-2: The detection of gestures should be based on a ML model because the context is dynamic. The dynamism happens because, for each gesture, there is an enormous amount of different inputs that can be used to identify the gesture made (caused by different environments and different users). Machine learning algorithms are well known for solving this set of problems (e.g. [22]).
- FR-3: The "Watching the sunrise" scenario from Abigail happens in a dark environment (6 AM). This scenario implies that the system must detect a gesture in a low light condition environment. To achieve this, radars can be used, since one of the advantages of the radars is the detection of objects in environments without light. This fact supports the idea of choosing the radar instead of a normal camera to detect gestures.
- FR-4: This requirement enables and facilitates future improvements on the system without changing the whole architecture. Moreover, the architecture must be modular, utilizing decoupled modules to minimize interdependence.
- FR-5: The execution of commands on physical devices within the house (turn on TV, manage gadgets,...) is out of the domain of this project. The existence of a demonstrator works as a proof-of-concept that is going to simulate the outputs

produced based on input gestures. In a real application, instead of the demonstrator, we would have real devices being controlled, so the demonstrator is only a functional requirement in the scope of our project.

- FR-6: Although the interaction relying on wearables in this case would seem natural, there are two disadvantages associated with their use. The first one is that it can cause discomfort to the users. The second one is that the use of the system can happen in unpredictable moments. This fact would force the users to use the gadget every time they want to use the system. These two disadvantages justify the idea of discarding wearables.

- FR-7: The production of feedback when a gesture is detected can improve the overall user experience by making the interaction with the system feel more responsive and natural. The feedback can be produced through audio, such as playing a specific sound for each gesture, or through visual means, such as displaying a message on a screen.

- FR-8 (Optional): Detected gestures can have a customizable output in order to satisfy the preferences of the user for different contexts

3.5.2 Non-functional requirements

Category	**Requirement**	**Description**
Availability	AR-1	Continuous availability.
Availability	AR-2	Our solution should respond to a valid gesture within 5s.
Availability	AR-3	The communication between the radar and the gesture input modality must be reliable and support the transmission speed of the radar data.
Security	SR-1	The user privacy should not be compromised.
Adaptability	AdR-1	The system shall ignore environment noise and focus on the user's movement.
Quality	QR-1	The system should not confuse one gesture with other movements.
Durability	DR-1	The whole system should be functional for a long period of time (several months).
Accessibility	AcR-1	A gesture must be detected within a range up to 3 m.

Table 3.3: Non-functional requirements defined for our system.

- AR-1: Our system is designed mainly for home usage, so it should be available 24/7, as there is no way to predict when it is going to be used. In the described Scenarios, the users never need to turn on the radar in order to detect a gesture. This fact implies that the system must be on 100% of the time.
- AR-2: For interaction to feel natural, the response time needs to be short. One second is a fair amount of time, since for a human to fully "decode" information, given by others, usually takes about 500 ms [23].
- AR-3: If the communication between the radar and the gesture input modality fails, our system suddenly stops working. Also, if the system does not support the data speed transmission of the radar, it may lose important data.
- SR-1: Since the system will be used in a home environment, it can capture private information of what is going on inside the house especially in the bedroom or bathroom. Security measures should be taken into consideration in order to make sure there is no sensitive information collected.
- AdR-1: Some kind of filter mechanism should be developed in order to minimize environment noise. This will be particularly challenging because, from the papers we analyzed above 2.4, all the studies were performed in a controlled laboratory environment.
- QR-1: Gestures wrongly identified should be avoided at all costs, because it can cause frustration to the users. If the system is undecided between a set of gestures, no detection is preferred.
- DR-1: This requirement implies that our system must be reliable enough to keep working for months. In order to keep it as reliable as possible, software and hardware tests should be considered.
- AcR-1: The system should effectively operate with the person located in any room of the house, within a maximum distance of 3 meters. This may necessitate the use of multiple radars throughout the house to ensure proper coverage and functionality.

Chapter 4

Radar-Based System for Interaction with Smart Homes

This chapter presents our proposal of a system for interacting with Smart Homes, using a radar, to aid in interacting with Smart Environments. It depicts the design and development of a prototype as a proof-of-concept. The establishment of a good system architecture/proposal is crucial to any project development. From it we can take all the atomic parts of the system and how the process flows through them.

When designing the system, we considered the following key questions:

- How to make it as modular as possible?
- How can an other output modalities interact with the gesture input modality?
- What are the steps of building a model to be used by the gesture input modality?
- What are the parts that constitute a gesture input modality?

These questions will be present and answered across this chapter.

4.1 System Overview

Based on the specifications and requirements of our project, we defined the system proposal illustrated in Figure 4.1. This architecture focuses on the idea of making every module as independent as possible, to facilitate its use in future projects and allow for simpler improvement, addition and removal of modules.

The system is composed of two main blocks, the Gesture Input Modality and the Home Interaction Ecosystem, which together implement a working system and are connected using the AM4I framework[24] Interaction Manager and Fusion Engine. The remaining block, Model Evaluation and Training, is another essential part of the system, in the sense that it is used to obtain the model used by the Gesture Input Modality.

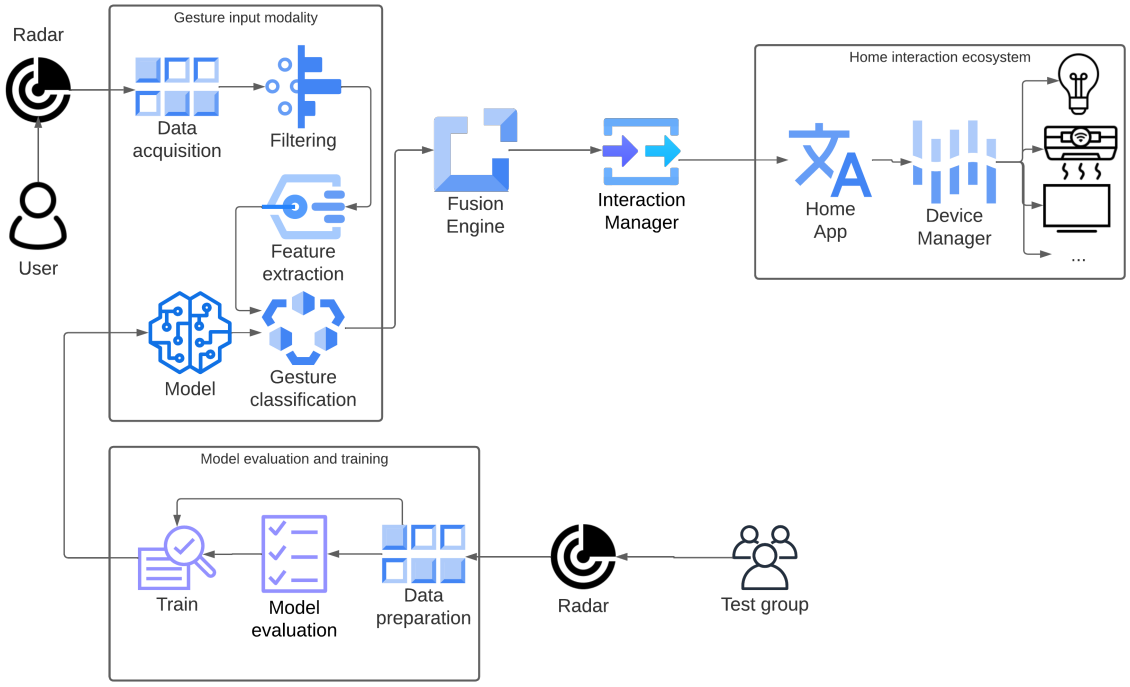


Figure 4.1: System Proposal.

4.2 Gesture Input Modality

The Gesture Input Modality (GIM) corresponds to a set of components/modules that implement our Real-Time Gesture Recognition System. As observed in Figure 4.2, our system is divided into five main components. We will briefly define their purpose, but a more comprehensive description of them is present in the following sections.

- **Data acquisition** - receives and parses the data sent by the radar (Figure 4.2)
- **Filtering** - filters the acquired data (4.2.1)
- **Feature extraction** - extracts features from the filtered data using a sliding window (an image per window) (4.2.3)
- **Gesture classification** -classifies the performed gesture using a model that recognizes which gesture was executed (if any) based on the features (4.2.4).

This components work independently and can be operated in the same or in separate devices. This is due to the fact that they communicate through a broker using the MQTT protocol. Taking advantage of the publish-subscribe ability, each component interacts with

one or two different channels, allowing us to segment communication between modules and guarantee that every component only receives messages directed at them.

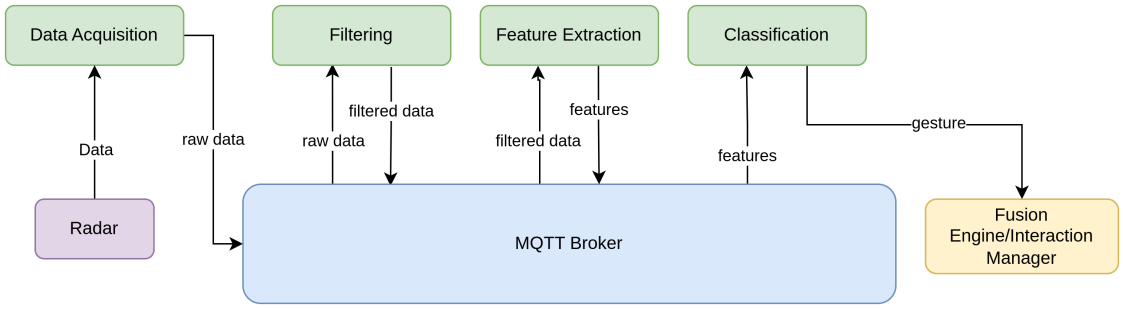


Figure 4.2: Interconnection between modules in the GIM

4.2.1 Data Acquisition

As referred previously in this report, we are using a TI mmWave FMCW radar. The radar captures and sends data in the form of packets, containing a header and the Tag Length and Values (TLV). The data is then read by this module and parsed into JSON format (as seen in Figure 4.3), which will be used by our system.


```
{
    "numObj": 2,
    "rangeIdx": [1, 255],
    "range": [0.0352734375, 8.994726562499999],
    "dopplerIdx": [0, 0],
    "doppler": [0.0, 0.0],
    "peakVal": [31544, 28959],
    "x": [0.0078125, -2.533203125],
    "y": [0.0, 7.197265625],
    "z": [0.03515625, -4.78515625]
}
```


Figure 4.3: Example of a single radar data frame corresponding to two detected targets

Operating in a three-dimensional space, the radar provides us with the following information for each detected moving target:

- X coordinate - location of the target in relation to the radar in the x-axis (horizontal)
- Y coordinate - location of the target in relation to the radar in the y-axis (distance)

- Z coordinate - location of the target in relation to the radar in the z-axis (vertical)
- Doppler - velocity
- Range - represents the radial distance between the target and the radar or objects being detected

The **Doppler** describes the rate that a target moves toward or away from the radar. A target with no range-rate reflects a frequency near the transmitter frequency and cannot be detected. A Doppler equal to zero indicates that the target is on a heading that is tangential to the radar antenna beam. The **Doppler Index**, gives us information about the movement, with the value of the index depicting the intensity of the movement and its sign indicating the direction of the target's motion (positive for approaching, negative for receding),

In order to start gathering and processing data from the radar, it is essential to configure it. The Radar is an extremely configurable device, allowing for several parameters to be fine-tuned and configured for specific use cases, such as Frame Rate, Range Resolution and others. This configuration is sent to the radar every time we start an online data acquisition, to ensure that the correct configuration is used. Before we started building all the GIM components, we had to find and test a configuration that we observed to generate good features. This configuration can be observed in Figure 4.4.

Attention: The configuration showed here worked for our use case, with the subject standing between 1 to 1.5 m away from the radar, while sitting in a chair. For different distances or situation, further exploration of the best configuration should be conducted.

Following a modular and customizable approach, we implemented the Data Acquisition module to operate based on the Frame Rate defined in the configuration file. This means that each sample sent can vary depending on the configuration used.

In Online mode, the Data Acquisition module has a parameter allowing to configure the number of seconds that data segment should amount to. Combining this parameter and the radar's frame rate, obtained through parsing the configuration file, we can determine the amount of samples/frames that each package sent or data segment should contain.

For example, for a defined frame rate of 20 fps, 2 seconds of data correspond to 40 samples/frames.

4.2.2 Data Filtering

The Filtering Component takes raw or unfiltered data from the Data Acquisition module and filters and sends them to the next module, the Feature Extraction.

There are several reasons why the radar data contains noises and seemingly random data. Sometimes simple changes in the environment, like windows, objects and people walking by (even behind walls) can introduce a lot of variability in the data. However, for the system to function effectively, gestures should be performed in a relatively motionless

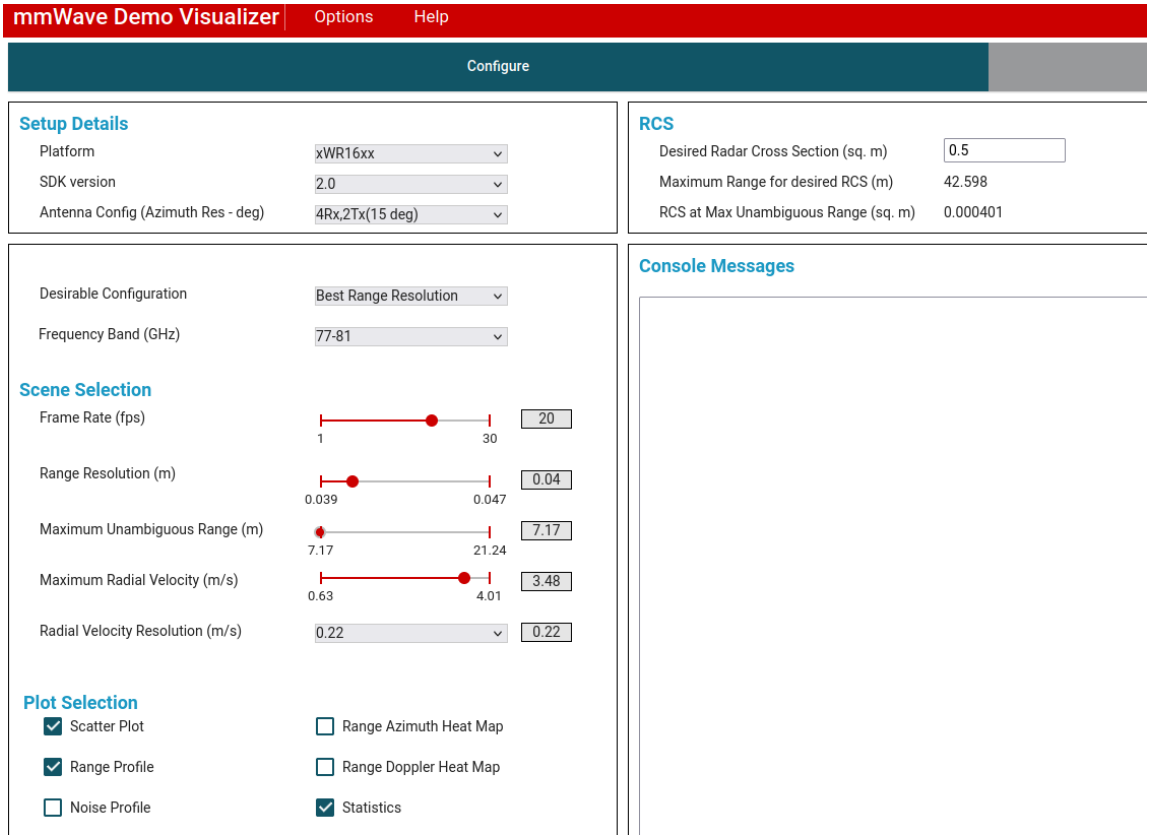
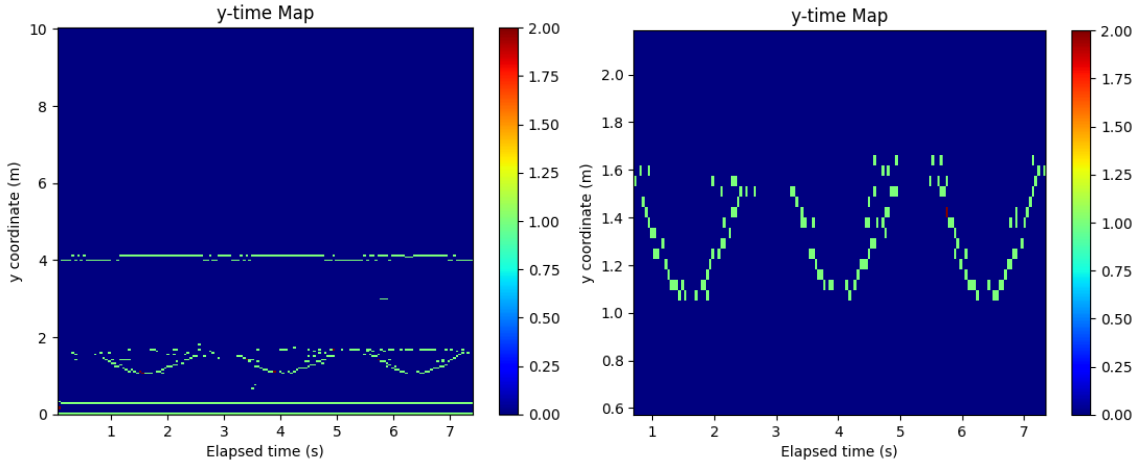


Figure 4.4: Configurable radar parameters (defined in "TI mmWave Visualizer" application)

environment. Nevertheless, easily detectable movements, whether very fast or very slow, will still be captured in the acquired data.

To filter the acquired data, first all the data with a speed close 0 m/s or with a absolute speed higher than 10 m/s is removed. Then a Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering function is applied with a minimum number of samples equal to 2 in a radius of 0.5m, in order to discard any discreet point. Since all the gestures have a relatively small variation in depth, the filtering process is finished by computing the average of all the remaining points (on all the axes) and all those >40% away from it are removed. A visual example of the difference between unfiltered and filtered data can be seen in Figure 4.5

When using the data filtering module in a online context, as established before, the Data Acquisition module sends data over a configurable amount of seconds (default is 1 second) to the filtering module. This component will not filter each data set individually.



(a) Before filtering

(b) After filtering

Figure 4.5: Heat map of the Y coordinate versus the elapsed time for the targets detected by the radar during the execution of a "push pull" gesture.

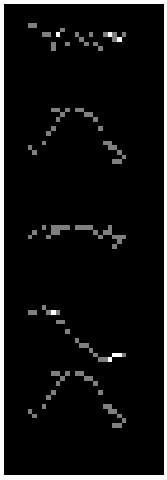
4.2.3 Feature Extraction

The Feature Extraction module receives filtered data from the Data Filtering component and using a sliding window approach, the data is then segmented with a configurable size and overlap between consecutive windows. Generates features are then parsed to gray-scale images.

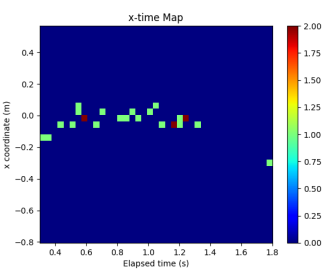
This module receives data segments every second, and these samples are combined to account for the sliding window size. By default, the sliding window size is set to two seconds, which means that two data segments are used. The number of samples in each segment depends on the defined sampling rate parsed by reading the configuration values sent together with the data.

Every feature consists of a gray-scale image formed by combining matrices composed of Heat Maps for the X, Y, and Z axes, Doppler, and Range values over time. When the concatenation occurs, all Heat Map get inverted in the resulting feature . Each pixel in the image represents a sample value, where, in the grey-scale index, white pixels indicate high values and black pixels represent low to no values.

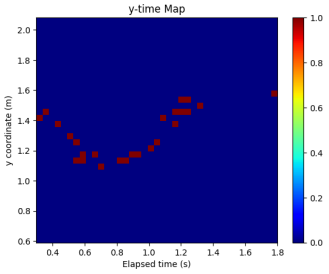
An example of this is shown in Figure 4.6, where the "push-pull" gesture was performed, with the generated Heat Maps and features.



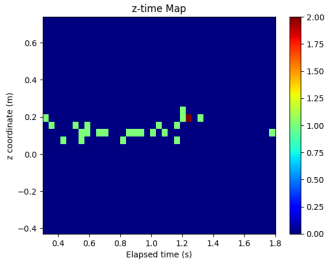
(a) Example of Push-Pull feature



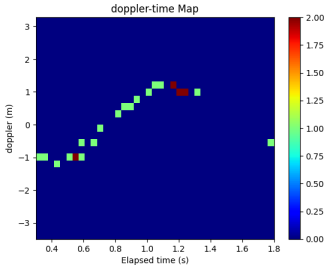
(b) X coordinate Vs. Time



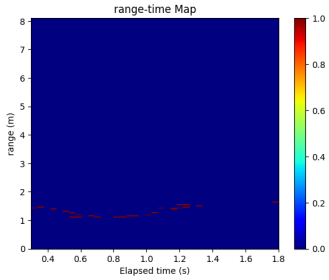
(c) Y coordinate Vs. Time



(d) Z coordinate Vs. Time



(e) Doppler Vs. Time



(f) Range Vs. Time

Figure 4.6: Combination of Heat Maps to a generated feature of the Push-Pull gesture

4.2.4 Gesture Classification

The Gesture Classification Module employs a Gesture Recognition model trained through Transfer Learning, utilizing features extracted from offline recordings of one or multiple subjects. Transfer learning involves the use of a pre-trained model, where part of the model

is re-trained using our own smaller dataset.

The Feature Extraction module provides the necessary features to the Model for determining the performed gesture. If a gesture is detected, the Gesture Classification module notifies the Interaction Manager/Fusion Engine about its occurrence.

Concerning the used gesture recognition model, it was trained relying on the MobileNetV2 pre-trained model and a dataset corresponding to six subjects, which achieves a balanced accuracy of 92% on that dataset. More details on the dataset and the approach used for evaluation are given in the next chapter (final experiment).

4.3 Interaction Manager and Fusion Engine

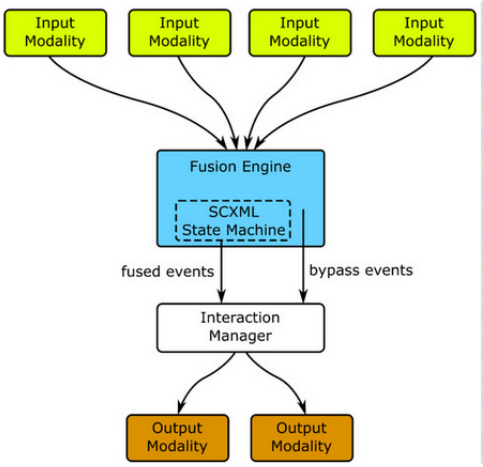


Figure 4.7: W3c MMI framework [25] multi-modal interaction architecture.

It is very important that everything is made taking into consideration that, if needed, various new modules can be assembled into the proposed architecture. Our system follows the W3C MMI framework [25]. To achieve that kind of versatility a middle point need to exist, in other words a place where everything meets, in our system the AM4I framework Fusion Engine (FE) and Interaction Manager (IM) [26] represent exactly that. Since both FE and IM were not developed by us, we will keep this section short, a full description of this elements can be found in the references.

All the communication that involves one of these two entities is implemented using Life Cycle Events as specified [24] and all the messages follow a XML like structure called EMMA[27] (take Figure 4.8 as an example).

In short, the FE receives messages from input modalities being capable to fuse various massages from different modalities. Then forwards those messages to the IM that uses a set of rules described in a SCXML file to create a simple state machine in order to finally

```
<mmi:mmi xmlns:mmi="http://www.w3.org/2008/04/mmi-arch" Version="1.0">
  <mmi:ExtensionNotification mmi:context="ctx-1" mmi:requestId="gesture-1"
  mmi:source="GESTUREIN" mmi:target="IM">
    <mmi:data>
      <emma:emma xmlns:emma="http://www.w3.org/2003/04/emma" Version="1.0">
        <emma:interpretation emma:confidence="1" emma:medium="gesture"
            emma:mode="command" emma:start="0" id="gesture-">
          <command>{"recognized": ["GESTUREIN"], "text": "left-swipe",
              "gestureMeaning": "Negative", "language": "PT"}</command>
        </emma:interpretation>
      </emma:emma>
    </mmi:data>
  </mmi:ExtensionNotification>
</mmi:mmi>
```

Figure 4.8: Message sent by the modality to the FE when a gesture named "left-swipe" is detected

notify one or more output modalities. In our scenario a message is sent to the FE by the Gesture Input Modality when it detects a gesture, then it will be forward to the IM that will finally will notify the Home Interaction Ecosystem of the detected gesture.

4.4 Home Interaction Ecosystem

Even the most complex ML modality, by itself is not useful. The generated output from the gesture classification after being distributed it needs to be interpreted and to produce a certain action. In this section we will focus on our proposal of an output modality. The main modules, as presented before (System Overview), are the Home app, the Device Manager and a set of output devices. In order to prove the feasibility of the project, instead of performing action on physical devices, we decides to build a Virtual Home that implements two scenarios and six different devices taken from the focus group (section 3.4).

4.4.1 Home app

The Home App is a simple MMI client that is always pooling the IM and when a new gesture arrives translates it into an action (a state change in the house). The translation is based on a json file that takes into account the user location and the gesture. So, it is expected that the App knows where the user is located within the house(bedroom, kitchen, ...). Before translating, a request to the **??** asking for the user current position is made. After converting the gesture into a action, the Home app sends a request to the **??** to perform the action.

4.4.2 Device Manager

An Application Programming Interface (API) was developed in order to store and retrieve the state of the house and to perform actions in the house. It was implemented using Python's Flask framework.

Request	Type	Description
/action?device=<device>&room=<room>	POST	change device state
/state	GET	retrieve full home state
/state/position	GET	retrieve user current positions
/state/device/<device_name>	GET	retrieve device state
/position/<room>	POST	set user position
/stream	GET	creates the stream with the client for gesture detection notification

Table 4.1: Home API interface[1]

A request sent to /action specifying the device and room where the device is the API will update the state of the house in the database and change the state of the device. For the propose of demonstration the state of each device is considered to be binary.

A request sent to /position specifying the room where the user is, will update that information in the database.

A request sent to /state will return the state of the home, a json organized by room and also including the current position of the user. This information is, as by now, used by the Home app in order to properly translate a recognized gesture into an action.

A request sent to /state/position will return the user current position.

A request sent to /state/device/<device_name> will return the respective device state.

Finally, a request sent to /stream will never actually return anything, so the connection will be open between client and server until the client closes it or 10 minutes pass without any communication. This request enables the use of Server Sent Events (SSE)'s[28] in order to notify the demonstrator that a certain action needs to be taken.

At this point, if the system would be deployed in a real scenario. The only change needed in order to interact with the smart devices inside the house, it would be in the /action endpoint. The manager would communicate with the physical device or some other device controller (out of the scope of this project) instead of sending a SSE to the virtual home as it is doing in the current implementation.

[1]Despite not being part of the Home API goal, one more request option is available ('/') that returns the Virtual Home home page

4.4.3 Virtual Home

In a fully deployed system in a smart home environment, the Home Interaction Ecosystem would receive input from the IM, then it would interpret it and perform a certain action upon 'real' software and/or hardware. Since we do not have the opportunity to fully deploy the system in a real physical scenario, a virtual home was built in order to show two potential scenarios, the living room and bedroom, taken from the Focus group.

(a) Living room

(b) Bedroom

Figure 4.9: Virtual home rooms

The Virtual Home is a web interface that is constantly connected to the Device Manager and waiting for SSE's asking for a certain action to be executed. When that happens, it changes the current frame/video of the room the user is in. Both 3D models representing the scenarios were built from scratch and rendered using Blender.

4.5 Tools and Technologies

The modality relies on data provided by the AWR1642 by Texas Instruments radar, following the trend of using a FMCW radar. In order to properly configure the radar and define its various parameters, we relied on the TI mmWave Visualizer application.

For the gesture recognition model, we decided to use Transfer Learning, since building a CNN model, like most related works used, takes a large amount of data. Moreover, TL can be the way to go if time is a constraint or a large processing power is not available. TL allows to use already pre-trained models and only provide a small amount of new samples. The use of a pre-trained model usually does not mean a loss in performance and simplifies the process 2.4.4.

We developed the Gesture input modality using Python as it is one of the most used programming languages for machine learning. It is very well documented, it has the simplicity of a high-level programming language but still offers access to a lot of modules, such as *numpy*, which is based on a library written in C and offers significantly performance advantages. It also provides plenty of other good frameworks for Artificial Intelligence (AI),

such as PyTorch.

For the virtual home, we developed a web app using Python and Flask, since it is a quick and lightweight solution to implementing a web application. The current frame is loaded from the image/video of the current room state and updated using JavaScript.

Chapter 5

Gesture Recognition Model Evaluation

5.1 Pipeline

In order to create a model for classification (not specific to TL), the abstract process goes as follows:

1. **Dataset preparation** - Offline data acquisition, filtering and feature extraction.
2. **Model evaluation** - One or more pre-trained models are coupled with the features resulting from the dataset preparation step and evaluated. After this phase, we should be able to know both what is the best pre-trained model from those tested and if the extracted features are appropriate.
3. **Final Model Training** - After going through the steps presented above, it is time to train the model using the whole dataset. From this training, a model will be created, which can be used by the GIM.

5.2 Experiments

For a comprehensive evaluation of gesture recognition capabilities within the Gesture Input Modality (GIM), two main sequential experiments were carried out. The Initial experiment aimed at exploring and testing the capability of the radar for gesture recognition. The second and final experiment was performed to obtain a model to use in the final functional prototype, considering the results extracted from the focus group and the Initial Experiment. The aim was to refine the model's performance and address challenges encountered during the evaluation process.

To ensure a comprehensive model evaluation, it is crucial to maintain consistency in the setup and methodology employed for constructing the dataset and evaluating the models. Therefore, all recordings utilizing the radar were conducted in a controlled environment.

The road-map to obtaining a final model to be used in our prototype involved a lot of testing, both offline and online. Several tests were performed, where different methods and configurations were explored, such as using offline dataset augmentation or not, different number of folds for cross-validation, different number of neurons in the fully connected layer (transfer learning), and different configuration regarding feature extraction. In total, around 50 models were evaluated across all tests trying to find and tweak parameters and methodologies to reach the best possible classification module.

However, for simplicity, only the most relevant tests and corresponding results for both experiments are presented in the remaining of this chapter. The data processing steps and evaluation approach used in those tests is described in the next subsection.

5.2.1 Data Processing and Evaluation Approach

In the tests presented below, the recorded radar data were processed with the same code used for filtering and feature extraction in the GIM. A sliding window of 2 seconds was used to filter the data and then extract features. The main difference is that, for evaluation, feature were extracted using an overlap of 99% between consecutive windows, to obtain as many examples as possible. The dataset resulting from offline feature extraction was not augmented nor balanced.

For evaluation, a subject-dependent solution was considered, where a model is trained for each individual based on data acquired from that person only. The model was evaluated using a variation of the the 5-fold cross-validation approach, where 60% of the dataset is used for training, 20% for validation, and 20% for testing in each iteration.

Concerning model training, using Transfer Learning, the top layers of the pre-trained models were replaced by a single fully connected layer with 256 neurons (ReLU activation function) and an output layer with a number of neurons corresponding to the number o gestures (softmax activation function). The used optimizer was ADAM (default parameters). Crossentropy was used as the loss function, and accuracy as the metric to be evaluated during training and validation. Training was stopped when the validation loss has not decreased more than 0.1 for 5 epochs. The resulting model was then evaluated on the test data of the corresponding iteration.

The used base model was MobileNetV2, since it's a lightweight model that has a good balance between speed and accuracy [29]. However, one aspect that could be later experimented is to test different base models and how that impacts gesture recognition performance.

5.2.2 Initial Experiment

The initial experiment was conducted prior to the focus group and aimed at exploring the possibility of using a radar for gesture recognition and familiarize ourselves with the model evaluation aspects of our project. To achieve that, we considered a low number of people with only a few selected gestures by us.

5.2.2.1 Participants, Setup and Protocol

This initial experiment involved a group of three subjects. They were all male and right-handed, in the 20-22 year old range. The experiment was done in IEETA, with the radar set in a table and each subject performing the gestures while sitting in a chair from a distance of 1.5 m to the radar. To begin, we deliberated on the selection of gestures for this experiment. In this case, we focused on incorporating natural arm motions, eventually choosing three specific gestures: push-pull, swipe-left, and swipe-right. The push-pull gesture involved the subject extending their arm forward and then retracting it. For the swipe-left and swipe-right motions, the participants simply waved their entire arm to one side or the other. Each subject performed 10 repetitions of each gesture in a single recording (1 recording per subject and gesture), with a interval of 2 seconds between each repetition.

5.2.2.2 Dataset

The recorded radar data was processed using the method described above, resulting in the number of examples per gesture and subject shown in Table 5.1. There is a notable difference in the number of examples of Subject 2 for the Swipe-left and Swipe-right movement, which was caused by lack of movement detection by the radar because the subject did not use a lot of depth (i.e., movement in the direction of the radar) when performing the Swipe motions. This was something that we observed after some additional tests with the radar, i.e., that the radar needs to at least have minimal movement in the Y axis (depth) to be able to capture properly a movement.

	Push-pull	**Swipe-left**	**Swipe-right**	**Total**
Subject1	300	240	288	828
Subject2	333	23	17	373
Subject3	334	158	235	727
Total	967	421	540	1928

Table 5.1: Number of examples per gesture and subject.

5.2.2.3 Results and Discussion

As can be seen in Figure 5.1, the F1 score values achieved by the model are within the range of around 90% to 100% for each gesture, when excluding outliers. Despite these promising results, when using the model trained with the whole dataset to classify gesture data online, it failed to perform, with miss classification of gestures and sometimes, even two features of the same gesture resulting in two different classified gestures.

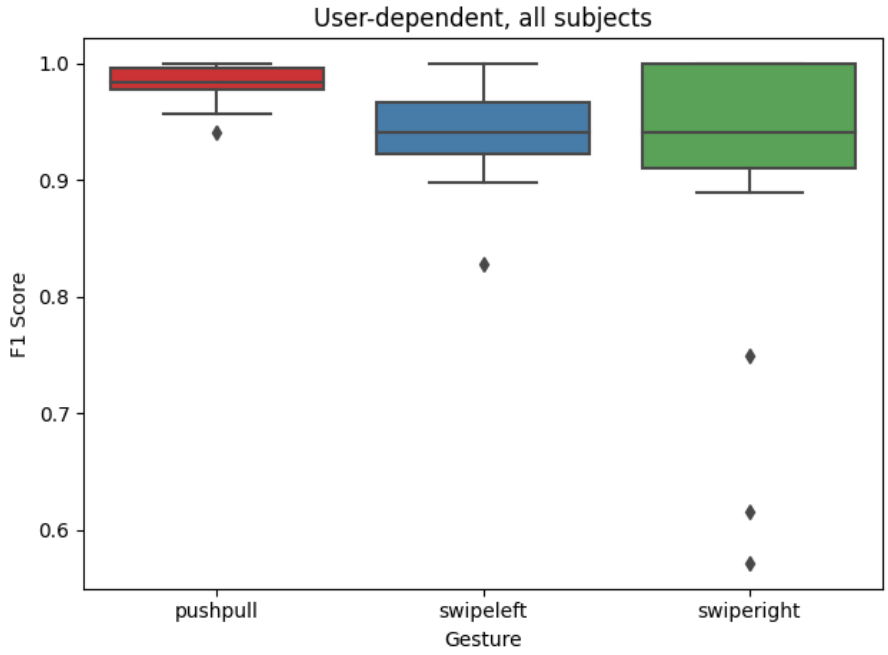


Figure 5.1: Boxplots for the F1 score achieved by the model evaluated in the initial experiment for the considered tree gestures (subject-dependent).

Upon further analysis, we discovered two main factors contributing to this problem. Firstly, we found that the radar technology proved to be highly sensitive to noise and reflections, which posed challenges in our recording environment. The presence of numerous people and multiple windows in the room exacerbated this sensitivity. Secondly, our dataset was relatively small, even worse when it comes to subject 2 like reference early, leading to overfitting of the model, despite our attempts to address it through techniques such as data augmentation, and early stopping and dropout rate adjustments.

Another factor that lead us to conclude that we need more data is depicted in Figure 5.2, where we can see that features for the same gesture, in this case Swipe-Left, can exhibit significant variations. This disparity can be attributed to factors such as noise, reflections, or variability in the way the subject executes the gesture on different occasions. To address

this issue, one possible solution is to increase the duration of the sample recordings and subsequently expand the size of our dataset. By doing so, we can capture a more comprehensive range of variations in gesture performance, ultimately enhancing the model's ability to handle such variability.

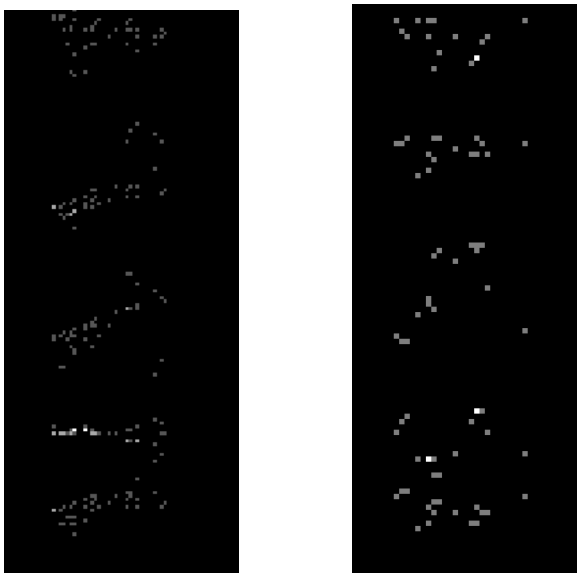


Figure 5.2: Two examples of features corresponding to the Swipe-Left gesture, performed by the same given subject.

5.2.3 Final Experiment

Following the focus group analysis, a second experiment was conducted. With new suggestions, additional gestures were incorporated. In this experiment, we included both the previously utilized gestures (Push-pull, swipe-left, and swipe-right) and newly suggested gestures by the participants of the focus group. These new gestures are the following:

- **Up-Down**: Involves raising and lowering the arm.
- **Open** and **Close** arms: Resemble the swipe left and right motions but are performed with both arms simultaneously in an open and close motion.

5.2.3.1 Participants, Setup and Protocol

This experiment was performed with 6 subjects, 5 males and 1 female, all in the 19-22 year old range and right-handed. The experiment was also done in IEETA, the radar was set in a tripod with a similar height as the table from the first experience, about 1 meter, and

each subject performed the gestures while sitting in a chair from a 1.5 m distance from the radar.

Based on the findings from the initial experiment, a decision was made to modify the acquisition approach for this phase. In the second experiment, subjects were instructed to perform 10 repetitions of each gesture in three different locations within the same room, the distance between the subject and the radar were the same in all 3 what changed was relative positioning in the room deciding to rotate the radar orientation and subject location, resulting in a total of 30 repetitions per gesture with 10 per diffent radar orientation. This adjustment was implemented to test whether increasing the number of examples and variability among them would help mitigate the overfitting issue identified in the first experiment. In addition, in each recording there is a 4 second interval between repetition, to make a more clear distinction between each repetition when processed in Feature Extraction.

5.2.3.2 Dataset

The number of examples per subject and gesture of the dataset resulting from filtering and feature extraction, over the recorded data in the second experiment, is indicted in Table 5.2.

	Push-pull	**Swipe-left**	**Swipe-right**	**Open**	**Close**	**Up-Down**
Subject1	1119	440	559	416	153	454
Subject2	842	190	43	139	0	890
Subject3	582	556	256	240	139	493
Subject4	810	153	430	83	29	83
Subject5	839	173	290	150	38	51
Subject6	990	508	686	240	21	504
Total	5182	2020	2264	1268	380	2475

Table 5.2: Number of examples per gesture and per subject corresponding to the dataset of the second experiment.

5.2.3.3 Results and Discussion

As evidenced in Figure 5.3, the F1 score median values for each gesture ranged from 60% to 100%. However, in comparison to Initial Experiment, a larger number of outliers corresponding to low F1 scores were observed. Once again, when we tested the model with data outside of the data used for evalution for gesture classification, we encountered the same issue as in Initial Experiment, signs of model overfitting, leading to inadequate performance in the online gesture classification.

Given these recurring challenges, we began questioning our methods for preventing overfitting, such as early stopping, cross-fold validation, and dropout rates. Despite training

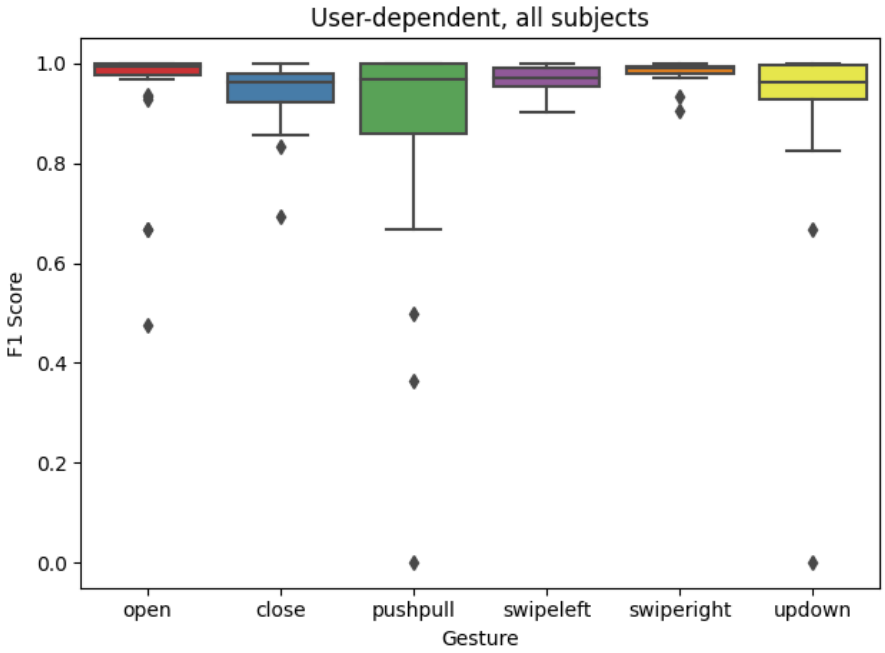


Figure 5.3: Boxplots for the F1 Score achieved by the model evaluated in the final experiment for the six considered gestures (subject-dependent).

multiple models and attempting to fine-tune these parameters, we still achieved similar results.

Consequently, we opted for a more systematic approach. For each subject-specific model (a total of 6 models), we tested the model using data from the other 5 subjects and calculated the average F1 score for each gesture. We then aggregated these results, yielding the average results outlined in Table 5.3.

			F1 score		
Close	**Open**	**Push-pull**	**Swipe-left**	**Swipe-right**	**Up-down**
15%	46%	65%	16%	10%	37%

Table 5.3: F1 Score for model performance in gesture classification with data unknown to the model, that was not part of training Dataset.

The analysis of Table 5.3 reveals several gestures with notably low F1 scores, prompting us to question whether specific issues with these gestures were impacting our overall results. Upon closer examination, we discovered that the three gestures exhibiting F1 scores in the range of 10-20% had problems with their respective data pools. Firstly, in comparison to

the others, some of the gestures, for example "Close" gesture has a very low amount of features compared to other gestures (Table 5.2). This particular gesture had significantly fewer extracted features despite the initial appearance of the raw data being satisfactory. This discrepancy suggests that the radar was unable to accurately detect these gestures, resulting in numerous gaps in the feature data. Consequently, these incomplete features failed to meet the filtering criteria and were deemed unsuitable for feeding into our model.

The visual representation in Figure 5.4 illustrates the stark contrast between the Features extracted from a Push-pull gesture (a well-performing gesture) and a Swipe-left gesture (a poorly-performing gesture). It becomes evident that the excessive noise present in certain gestures poses a significant challenge in effectively filtering the data and generating suitable inputs for the model.

(a) Push-pull (b) Swipe-left

Figure 5.4: Push-pull and Swipe-left features comparison

Even though Swipe-left and Swipe-right had not a very low number of examples, their quality was very questionable, because of improper filtering capability leaving too much gaps in those examples like shown in Figure 5.4, introducing an even bigger variability between them, such variety that we cannot support yet with such a small dataset. Consequently, this discrepancy in the amount of extracted Features or quality of the features, between well-performing and poorly performing gestures can be attributed to the difficulty in accurately filtering the data due to high levels of noise.

In order to construct a final model, we carefully considered the findings presented above.It became apparent that certain gestures had the potential to adversely affect the model's performance due to their limited sample size and/or example quality during training. To investigate this further, we evaluated a model under the same conditions, but we excluding three gestures (Swipe-left, Swipe-right, and Close arms) that had a potential impact on our model.

The results obtained for the three remaining gestures, i.e., Push-pull, Up-down, and Open arms gestures, are shown in Figure 5.5. We can observe that the same methodology yielded highly promising outcomes, even when applied to unseen data (results presented in Table 5.4), for each subject-dependent model. By ensuring an adequate sample size for each gesture and eliminating gestures that were prone to overfitting due to unreliable data, we have successfully developed a model that exhibits reasonable accuracy in classifying gestures within our Gesture Input Modality. This achievement highlights the effectiveness of our approach and instills confidence in the model's capability to accurately recognize and classify gestures.

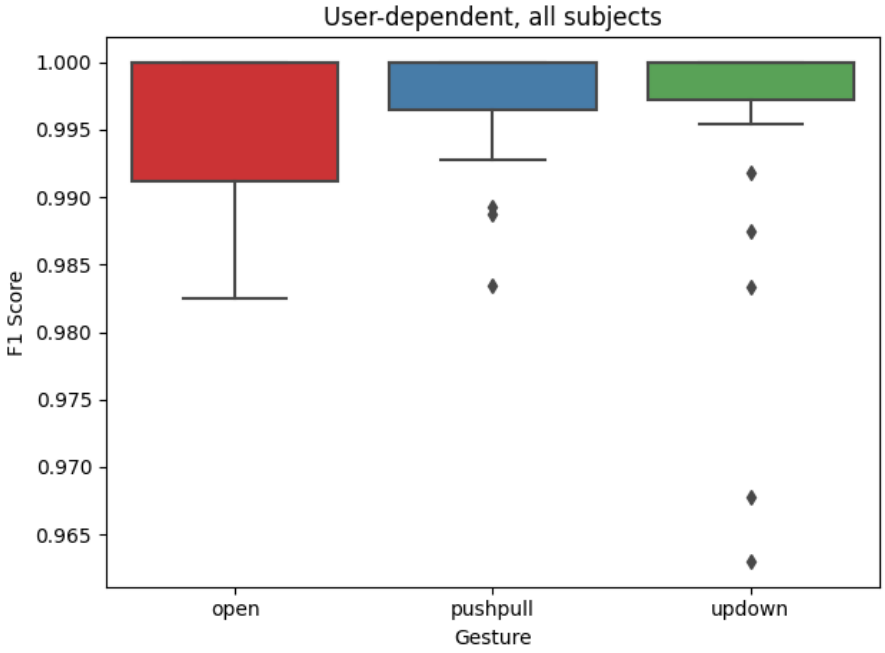


Figure 5.5: Boxplots for the F1 score achieved by the model evaluated in the final experiment for 3 selected gestures - Push-pull, Up-down, and Open (subject-dependent).

	Open-arms	Push-pull	Up-down
F1-Score	73%	81%	61%

Table 5.4: F1 Score for model performance in gesture classification with data unknown to the model, that was not part of training Dataset.

Chapter 6

Conclusion

6.1 Conclusions

The main objectives for this project were successfully accomplished, leading to the development of a system for natural interaction with smart homes. Our system lets the users interact with their home devices using hand gestures captured by a radar sensor. The integration of gestures enhance the user experience and provide an alternative and convenient method of interaction. The use of a radar sensor, more than just an innovative way to capture gestures, is less intrusive and physically more versatile (e.g., can be built behind a complete opaque layer of most materials).

Since the beginning we followed a user centered design employing techniques like the creation of personas, scenarios of usage and the conduction of a focus group. So, more than building a gesture-based system for hand gesture recognition, we deepened the study on gesture-based human-computer interaction in a home environment. We were able to achieve more concrete data on what the users wishes are and what main questions are yet to be answered. The system has a modular architecture, based on the AM4I architecture aligned with the W3C recommendations, which facilitates future system enhancement and expansion.

The radar-based system for interaction with smart homes is composed by two main blocks, the Gesture Input Modality and the Home Interaction Ecosystem. The first acquires data from the radar, processes them and classifies the gestures. The second turns the detected gestures into specific actions to take place in the house devices. The modular nature of the system was further improved by using W3C MMI facilitating future system enhancement and expansion.

To enable gesture recognition within the gesture modality, multiple models were explored using TL. Two Datasets were collected, the last one including 6 gestures performed by 6 people in 3 distinct locations, are another important outcome since it will serve for further exploration and analysis, enabling improvements in processing and feature extraction

methods. Each model underwent an evaluation and, after various adjustments and improvements in the process (e.g., adjust the filtering and training parameters), we reached a final model capable of recognising 3 gestures with an overall balanced accuracy of 92%. This level of accuracy ensures reliable gesture recognition in a moderate noisy setting movement-wise, affirming the effectiveness of radar-based gesture recognition as an input method for smart home interaction.

Finally, to show the practical application of the system, a demonstrator was developed as a proof-of-concept. This demonstrator features a virtual smart home environment complete with two home divisions, that are controlled by an application communicating with a device management API. This implementation facilitates easy configuration of the system in the future, enabling it to operate within a real home environment.

In the end, the project has established a strong basis for further research on radar-based gesture recognition for interaction with smart homes proving the feasibility and potential of this approach.

6.2 Future work

For future work, one important step would be to improve the gesture input modality, by increasing the dataset size, since having more variability is the best way to ensure a good performance of the model and prevent overfitting. This can be achieved by adding more diverse examples, including a greater number of subjects and more repetitions per subject carried out in different locations and days, and with varying angles and distances between the subject and the radar. Exploring other pre-trained deep learning models can also be beneficial, since they can potentially lead to a better gesture recognition performance.

Another improvement that can be carried out related to interaction between the user and the home is to introduce a focus modality that detects a user's focal point (where their attention is directed at). This would enable different operations to be associated with each gesture, based on the user's current focus. Additionally, investigating the potential for an activation gesture can help avoid false positives, ensuring that unintended gestures do not trigger unwanted actions.

Since Feedback is an important tool in communication or interacting with our environment, it would serve an important purpose in our system, not only in asserting that the correct gesture was executed, but also helping users perform the gestures. To help the users learn to perform the gestures, in an initial training phase, an application or website with icon/images depicting the movement and a small video executing it [30] could be used. There are several possible approaches to providing feedback for gestures. These include emitting a short sound when the gesture is executed, utilizing a voice assistant, sending a notification to the user's smartphone, or displaying a notification pop-up or alert within the current system being used.

Lastly, developing a platform that facilitates the definition of gestures and the integra-

tion of controlled devices within the user's environment can improve usability. This platform may provide interfaces that allow users to define and customize gestures according to their preferences, as well as controlling various devices.

Chapter 7

Acknowledgements

We would like to thank our project supervisors, Dr[a] Ana Rocha and Prof. Samuel Silva, as well as to Prof. José Moreira, for their valuable guidance, expertise, and continuous support. Their commitment to our project and insightful feedback played a crucial role in shaping our research and methodology.

We would also like to extend our gratitude to our fellow classmates and team members for their dedication, hard work, and collaboration. Each member brought unique skills and perspectives to the project, and our collective effort and teamwork was instrumental in overcoming challenges and achieving our goals.

Additionally, we are thankful to the faculty and staff of DETI and IEETA, for providing us with the necessary resources and facilities to conduct our experiments and research.

Without the contributions and support of all these individuals, this project would not have been possible. We are proud of our achievements and look forward to applying the knowledge and skills gained from this project on our future endeavors.

Bibliography

[1] Apple, *Siri*. [Online]. Available: https://www.apple.com/ios/siri/.

[2] Amazon, *Alexa*. [Online]. Available: https://alexa.amazon.com/.

[3] Google, *Google assistant*. [Online]. Available: https://assistant.google.com/.

[4] Samsung, *Bixby*. [Online]. Available: https://www.samsung.com/global/galaxy/what-is/bixby/.

[5] C. M. Hurley, A. E. Anker, M. G. Frank, D. Matsumoto, and H. C. Hwang, "Background factors predicting accuracy and improvement in micro expression recognition," *Motivation and Emotion*, vol. 38, no. 5, pp. 700–714, Oct. 2014, ISSN: 1573-6644. DOI: 10.1007/s11031-014-9410-9. [Online]. Available: https://doi.org/10.1007/s11031-014-9410-9.

[6] S. Ahmed, K. D. Kallu, S. Ahmed, and S. H. Cho, "Hand gestures recognition using radar sensors for human-computer-interaction: A review," *Remote Sensing*, vol. 13, no. 3, 2021, ISSN: 2072-4292. DOI: 10.3390/rs13030527. [Online]. Available: https://www.mdpi.com/2072-4292/13/3/527.

[7] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, Jan. 2015, ISSN: 1573-7462. DOI: 10.1007/s10462-012-9356-9. [Online]. Available: https://doi.org/10.1007/s10462-012-9356-9.

[8] furuno, *Radar basics*. [Online]. Available: https://www.furuno.com/en/technology/radar/basic/.

[9] *Mmwave radar*, 2020. [Online]. Available: https://www.ti.com/lit/wp/spyy005a/spyy005a.pdf?ts=1668600958027.

[10] Nasa, *Doppler effect*. [Online]. Available: https://www.grc.nasa.gov/www/k-12/airplane/doppler.html.

[11] *Radar advantages*, Oct. 2022. [Online]. Available: https://lidarradar.com/info/advantages-and-disadvantages-of-radar-systems.

[12] Q. Wan, Y. Li, C. Li, and R. Pal, "Gesture recognition for smart home applications using portable radar sensors," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 6414–6417. DOI: 10.1109/EMBC.2014.6945096.

[13] Y. Wang, S. Wang, M. Zhou, Q. Jiang, and Z. Tian, "Ts-i3d based hand gesture recognition method with radar sensor," *IEEE Access*, vol. 7, pp. 22 902–22 913, 2019. DOI: 10.1109/ACCESS.2019.2897060.

[14] L. Santana, A. P. Rocha, A. Guimarães, *et al.*, "Radar-based gesture recognition towards supporting communication in aphasia: The bedroom scenario," in *Mobile and Ubiquitous Systems: Computing, Networking and Services*, T. Hara and H. Yamaguchi, Eds., Cham: Springer International Publishing, 2022, pp. 500–506, ISBN: 978-3-030-94822-1.

[15] Z. Xia and F. Xu, "Time-space dimension reduction of millimeter-wave radar point-clouds for smart-home hand-gesture recognition," *IEEE Sensors Journal*, vol. 22, no. 5, pp. 4425–4437, 2022. DOI: 10.1109/JSEN.2022.3145844.

[16] S. Hazra, H. Feng, G. N. Kiprit, *et al.*, *Cross-modal learning of graph representations using radar point cloud for long-range gesture recognition*, 2022. DOI: 10.48550/ARXIV.2203.17066. [Online]. Available: https://arxiv.org/abs/2203.17066.

[17] C. Abras, D. Maloney-Krichmar, J. Preece, *et al.*, "User-centered design," *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, vol. 37, no. 4, pp. 445–456, 2004.

[18] I. D. Foundation, *User-centered design*, n.d. [Online]. Available: https://www.interaction-design.org/literature/topics/user-centered-design.

[19] A. Hickey and A. Davis, "Elicitation technique selection: How do experts do it?" In *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.*, 2003, pp. 169–178. DOI: 10.1109/ICRE.2003.1232748.

[20] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-defined gestures for surface computing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09, Boston, MA, USA: Association for Computing Machinery, 2009, pp. 1083–1092, ISBN: 9781605582467. DOI: 10.1145/1518701.1518866. [Online]. Available: https://doi.org/10.1145/1518701.1518866.

[21] S. Wilkinson, "Focus group methodology: A review," *International Journal of Social Research Methodology*, vol. 1:3, pp. 181–203, 2014. DOI: 10.1080/13645579.1998.10846874. [Online]. Available: https://www.tandfonline.com/doi/pdf/10.1080/13645579.1998.10846874?needAccess=true.

[22] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, *Support vector machines*, 1998. DOI: 10.1109/5254.708428.

[23] L. Isik, A. Mynick, D. Pantazis, and N. Kanwisher, "The speed of human social interaction perception," *NeuroImage*, vol. 215, p. 116 844, 2020, ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2020.116844. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811920303311.

[24] A. Deb, M. Candeleria, S. Hwang, and X. Xiao, "Multimodal Interaction Architecture," World Wide Web Consortium, W3C Recommendation 25 October 2007, 2007. [Online]. Available: https://www.w3.org/TR/mmi-arch/#LifeCycleEvents.

[25] D. Dahl, M. Johnston, S. McGlashan, and M. Froumentin, "W3C Multimodal Interaction Architecture," World Wide Web Consortium, Tech. Rep., 2007. [Online]. Available: https://www.w3.org/TR/mmi-arch/.

[26] N. Almeida, A. Teixeira, S. Silva, and M. Ketsmur, "The am4i architecture and framework for multimodal interaction and its application to smart environments," *Sensors*, vol. 19, no. 11, 2019, ISSN: 1424-8220. DOI: 10.3390/s19112587. [Online]. Available: https://www.mdpi.com/1424-8220/19/11/2587.

[27] W. W. W. Consortium, "EMMA: Extensible MultiModal Annotation Markup Language," World Wide Web Consortium, W3C Recommendation 14 October 2008, 2008. [Online]. Available: https://www.w3.org/TR/emma/.

[28] M. D. Network, *Using server-sent events*, https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events/Using_server-sent_events, [Accessed March 10, 2023].

[29] L. Zhao, L. Wang, Y. Jia, and Y. Cui, "A lightweight deep neural network with higher accuracy," en, *PLoS One*, vol. 17, no. 8, e0271225, Aug. 2022.

[30] A. Kamal, Y. Li, and E. Lank, "Teaching motion gestures via recognizer feedback," in *Proceedings of the 19th International Conference on Intelligent User Interfaces*, ser. IUI '14, Haifa, Israel: Association for Computing Machinery, 2014, pp. 73–82, ISBN: 9781450321846. DOI: 10.1145/2557500.2557521. [Online]. Available: https://doi.org/10.1145/2557500.2557521.