Multisensor Data Fusion
Applied to Augmented Reality

Andres Navarro -1370715
Title: Multisensor Data Fusion applied to Augmented Reality

Abstract: Multisensor data fusion deals with the combination of data generated by multiple types of sensors, in order to obtain more valuable data and perform inferences that may not be possible from a single sensor alone. Augmented reality adds information, meaning or any kind of media to a real environment by combining the real world with computer generated data. In order to overlay the real and the virtual content, the user of the AR platform must be monitored using several sensors. Multisensor data fusion is the best approach at combining these multiple sensors.

In this thesis is covered the design of a multisensor data fusion architecture for providing accurate positioning data and hand gesture interaction to an AR platform, the implementation of the different fusion processes defined in the fusion architecture and the performance of a pilot experiment within the AR platform trying to study the performance of the spatial memory in different conditions. The experiment results showed firstly the correct performance of the AR platform in terms of functionality, usability and presence; and they supported the hypothesis of a better performance of spatial memory when the user is allowed to explore an environment rather than observing it from a static position, although the hypothesis cannot be validated due to the limited number of subjects at the experiment. In addition, several guidelines were given at performing an experiment in an AR environment.
Contents

1 Introduction 1

2 Background 3
  2.1 Multisensor Data Fusion ........................................ 3
  2.2 Augmented Reality ............................................. 7
  2.3 Analysis of user behaviour .................................... 8

3 Design 9
  3.1 Hardware description .......................................... 9
    3.1.1 Ultrasound positioning system ............................ 10
    3.1.1.1 Hexamite hardware .................................. 10
    3.1.1.2 Distribution of receivers in the environment ....... 10
    3.1.2 Inertial Navigation System ............................... 11
    3.1.3 Hand gesture system .................................... 12
      3.1.3.1 MUSH ............................................. 12
      3.1.3.2 Wii Remote ....................................... 14
      3.1.3.3 MUSH-device vs Wii Remote ......................... 15
  3.2 Fusion system .................................................. 16
    3.2.1 JDL approach ............................................ 16
    3.2.2 System Architecture .................................... 17
  3.3 Conclusions .................................................... 20

4 Implementation 23
  4.1 Position location using TDOA ................................ 23
    4.1.1 Hyperbolic Equation Solving Algorithms ................. 24
    4.1.2 Chan’s method ......................................... 26
    4.1.3 TDOA algorithms implementation ......................... 28
    4.1.4 Testing ............................................... 33
  4.2 INS and ultrasound position fusion .......................... 33
    4.2.1 Ultrasound signal analysis .............................. 33
    4.2.2 INS signal analysis .................................... 34
    4.2.3 Fusion process design .................................. 39
    4.2.4 Kalman filter design ................................... 41
    4.2.5 Testing and algorithm refinement ....................... 45
4.3 Hand gesture recognition
  4.3.1 Processing of input data
  4.3.2 Acceleration signal analysis
  4.3.3 Gesture recognition
  4.3.4 Testing and redesign

4.4 Object interaction
  4.4.1 Object selection
  4.4.2 Interaction mapping

5 Experiment
  5.1 Experiment method
    5.1.1 Design
    5.1.2 AR game development
    5.1.3 Measures
    5.1.4 Participants
  5.2 Results
  5.3 Conclusions

6 Conclusions

A Code
  A.1 INS and ultrasound Kalman Filter

B Experiment instructions and questionnaires
  B.1 Experiment instruction
  B.2 Experiment questionnaires
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>MUSH-device</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>MUSH-device scheme</td>
<td>13</td>
</tr>
<tr>
<td>3.3</td>
<td>MUSH-device transfer function</td>
<td>14</td>
</tr>
<tr>
<td>3.4</td>
<td>Wii Remote</td>
<td>15</td>
</tr>
<tr>
<td>3.5</td>
<td>JDL fusion model</td>
<td>16</td>
</tr>
<tr>
<td>3.6</td>
<td>Fusion system in the JDL model</td>
<td>18</td>
</tr>
<tr>
<td>3.7</td>
<td>Carvalho et al. Data Fusion Architecture</td>
<td>19</td>
</tr>
<tr>
<td>3.8</td>
<td>Fusion architecture class diagram</td>
<td>21</td>
</tr>
<tr>
<td>4.1</td>
<td>TDOA algorithm selection</td>
<td>29</td>
</tr>
<tr>
<td>4.2</td>
<td>Ultrasound signal of horizontal axes for the target in a circular path</td>
<td>34</td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of ultrasound signal an ideal signal</td>
<td>35</td>
</tr>
<tr>
<td>4.4</td>
<td>Accumulative error of timestamping in ultrasound signal</td>
<td>35</td>
</tr>
<tr>
<td>4.5</td>
<td>Crossing by zero of ultrasound signal</td>
<td>36</td>
</tr>
<tr>
<td>4.6</td>
<td>Time between crosses by zero of ultrasound signal</td>
<td>36</td>
</tr>
<tr>
<td>4.7</td>
<td>INS acceleration data in static state</td>
<td>37</td>
</tr>
<tr>
<td>4.8</td>
<td>Ultrasound signal of the movement of a person</td>
<td>38</td>
</tr>
<tr>
<td>4.9</td>
<td>INS signal of the movement of a person</td>
<td>38</td>
</tr>
<tr>
<td>4.10</td>
<td>Indirect feedback Kalman filter</td>
<td>40</td>
</tr>
<tr>
<td>4.11</td>
<td>INS and ultrasound fusion process</td>
<td>41</td>
</tr>
<tr>
<td>4.12</td>
<td>INS and ultrasound activity diagram</td>
<td>42</td>
</tr>
<tr>
<td>4.13</td>
<td>Time difference of hand acceleration data histogram</td>
<td>47</td>
</tr>
<tr>
<td>4.14</td>
<td>Gravity effect in axis x</td>
<td>48</td>
</tr>
<tr>
<td>4.15</td>
<td>Gravity effect in axis y</td>
<td>48</td>
</tr>
<tr>
<td>4.16</td>
<td>Gravity effect in axis z</td>
<td>49</td>
</tr>
<tr>
<td>4.17</td>
<td>Rotation effect</td>
<td>50</td>
</tr>
<tr>
<td>4.18</td>
<td>Movement in x axis when gravity in y axis</td>
<td>50</td>
</tr>
<tr>
<td>4.19</td>
<td>Movement in y axis when gravity in the axis</td>
<td>51</td>
</tr>
<tr>
<td>4.20</td>
<td>Initial, middle and end time in the acceleration signal</td>
<td>54</td>
</tr>
<tr>
<td>4.21</td>
<td>Hand gesture recognition diagram</td>
<td>55</td>
</tr>
<tr>
<td>4.22</td>
<td>Hand gesture class diagram</td>
<td>56</td>
</tr>
<tr>
<td>4.23</td>
<td>Spherical coordinate system</td>
<td>58</td>
</tr>
<tr>
<td>5.1</td>
<td>Memory game classes implementation</td>
<td>63</td>
</tr>
</tbody>
</table>
5.2 Participants during the experiment .......................... 65
5.3 Statistics values of the participants .......................... 66
5.4 Analysis of questionnaires data .............................. 68
5.5 Evaluation by participants of conditions difficulty ......... 68
5.6 Looked boxes by the participants in the different conditions .... 69
5.7 Example of player movement .............................. 70
5.8 Time expended by participants to complete the game ...... 71
5.9 Time expended by participants before interacting with a box . 72
5.10 Orientation of participants before interacting with a box .... 72
## List of Tables

2.1 Fusion models summary ........................................... 5  
2.2 Techniques summary ........................................... 6  
2.3 Distributed Data Fusion advantages/disadvantages ........... 6  

3.1 Ultrasound receivers location .................................. 11  
3.2 InertiaCube3 specifications .................................... 11  
3.3 MUSH-device accelerometers specifications .................... 13  
3.4 Wiimote accelerometer specifications .......................... 15  

4.1 TDOA methods comparison ..................................... 26  
4.2 Ultrasound positioning system precision ....................... 33  
4.3 Benefits and Drawbacks of Kalman Filter Implementations ... 39  
4.4 Summary of Kalman Filter parameters ........................ 44  
4.5 Values of Kalman filter statistical parameters ................ 45  
4.6 INS and ultrasound fusion parameters refinement ............ 46  
4.7 Hand acceleration data fields .................................. 46  
4.8 Definition of fuzzy variables ................................... 53  
4.9 Interaction mapping ............................................ 58  

5.1 Descriptive statistics of the participants ...................... 66  
5.2 Cronbach's alpha for the questionnaires ...................... 67  
5.3 Paired sample T test of presence and usability questionnaires 67
Chapter 1

Introduction

In contemporary society, in our technologically supported daily lives, sensors certainly play an important role. Various applications require information about the environment and sensors are devices that transform some physical phenomena in the environment of interest into readable information. Where information from different sensors becomes available, more advanced and valuable applications can be implemented. The information obtained from these multiple different sensors must be combined to perform inferences that may not be possible from a single sensor alone. This combination must be performed by a computational system in a process called multisensor data fusion.

Augmented Reality (AR) is the combination of the real world and computer generated data. AR adds information and meaning to a real object or place by overlaying virtual content. For a realistic effect, virtual content must be correctly aligned to the real world. The alignment relies on an accurate tracking of the player position and orientation. Using multiple types of sensors is the best way of obtaining this accurate information, processing the sensed data by a multisensor data fusion process.

This thesis of the Delft University of Technology was carried out at V2 Institute of the Unstable Media, an interdisciplinary center for art and media technology\(^1\). An Augmented Reality platform for several projects is being developed at V2. One of the open tasks at the beginning of this thesis was the provision to the platform of a positioning system. Therefore, one of the aims of this project is the development of a working and accurate positioning system for the AR platform.

In multiple projects at V2_Lab, the user is observed using several sensors. An interest is present in the analysis of the user behaviour from the sensed information. Hence, a description of the analysis of the user immersed in an Augmented reality experience is desired. This analysis will be based on the interaction of the user with the AR environment by his position and his hands gestures. Therefore, a hand gesture recognition system must be developed.

\(^1\)http://www.v2.nl
To test the developed platform, a scientific experiment that makes use of the information that can be obtained by analyzing the user behaviour will be carried out. Thus, the procedure at designing, implementing and evaluating the experiment will be detailed.

Trying to cover the presented issues, the following research questions can be formulated:

- How can multisensor data fusion be applied to design an Augmented Reality platform?
- How can accurate positioning data be provided to an AR platform?
- How can hand gesture interaction be added to an AR platform?
- Perform a pilot experiment to look for procedures for carrying out scientific experiments in an AR environment?

To answer these questions, the next contents are covered in the following chapters. In the next chapter, the second one, a background about multisensor data fusion, augmented reality and analysis of user behaviour is presented. The third chapter describes the design of the fusion system from a description of the used hardware. In the fourth chapter, the implementation of the different fusion process is described. The fifth chapter covers the experiment carried out. Finally, the conclusions chapter summarizes these contents answering to the research questions listed in this introduction.
Chapter 2

Background

2.1 Multisensor Data Fusion

As part of this thesis work, a research [16] about multisensor data fusion was carried out. The study tried to cover an overview of the definitions of multisensor data fusion, the models that can be followed, the techniques that can be applied, the centralized and the decentralized approach and some examples of scenarios where multisensor data fusion can be useful.

As a summary of this research assignment, the research questions which were formulated in a first stage and answered as the final conclusions are here reproduced.

• What is multisensor data fusion? Is there an unanimous definition?

Several definitions of multisensor data fusion are given in the literature. Most of them, due to their military origin, are restrictive to a certain terminology and applications. A broader definition is needed to cover such a wide diversity of sensor fusion applications. Therefore, Wald's definition [25] is chosen:

Data fusion is a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of greater quality will depend upon the application.

As more and more applications make use of multisensor data fusion, a broader definition will be needed and discussion about it will continue.

• What models for multisensor data fusion exist in the literature? Do they have common descriptions? Do they contradict each other?

There are numerous fusion models in the literature. Some of them were described in the research assignment and summarized in Table 2.1.
A common point in most of them is the need of divide the fusion process in levels of data abstraction, while different definitions are given to those levels: direct data - feature level - high level, object - situation - impact - process, data - feature - decision, signal - feature - pattern - situation - decision.

However, more disagreement is found in the idea of a cycling processing, where the OODA loop [4] (Observe - Orientate - Decide - Act) is the predominant model. Some models propose the use of fixed steps to process the sensor fusion while others plead for a structure where all modules are interconnected.

Fusion models, as any kind of model, can be more or less explicit, while explicitness implies specification to a concrete application, which implies unavailability for other applications. Hence, there is a relationship between specification - a guiding model is given - and usability - the model can be used for many applications.

The models presented are abstract and generic, not specific for an application. To develop a real system for a concrete application, they must be used as guidelines, combining their underlying ideas for the final design.

- What techniques or methods can be used in multisensor data fusion?

There are numerous techniques for sensor fusion. Applying the data fusion at different levels of abstraction implies the use of multiple techniques. The selection of these techniques mostly depends on the application of the sensor fusion and the resources available for it.

Some techniques were described in the research assignment and classified depending on the process function and the process level that they can implement, summarized in Table 2.2.

Different viewpoints and guiding objectives for technique selection were presented. A layout or scheme for the implementation of any kind of sensor fusion application is not feasible. The design of the fusion algorithms is a lengthy task where multiple fusion techniques can be combined.

- Centralized or decentralized data fusion?

A Distributed Data Fusion system is composed of different interconnected nodes. Each node processes the local information and information from some of the other nodes, but no central node exists that processes the information of all the nodes. The decentralized fashion has some advantages and some disadvantages compared to the centralized one. They are summarized in Table 2.3.

The implementation of Distributed Data Fusion requires the use of certain fusion models that allow decentralization. Likewise, specific algorithms must be used for the decentralized fusion process.

- What scenarios can multisensor data fusion be applied to?

The list of scenarios where sensor fusion can be applied to is very long. Some examples are: image data fusion, robot navigation, ambient intelligence and augmented reality.
<table>
<thead>
<tr>
<th>Fusion model</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Different revisions</td>
</tr>
<tr>
<td></td>
<td>Last revision makes a division into these process components: Sources, Database management, HCI and these process levels:</td>
</tr>
<tr>
<td></td>
<td>0 - Sub-Object Assessment (signal estimation)</td>
</tr>
<tr>
<td></td>
<td>1 - Object Assessment (entity estimation from observations)</td>
</tr>
<tr>
<td></td>
<td>2 - Situation Assessment (entity estimation from relations)</td>
</tr>
<tr>
<td></td>
<td>3 - Impact Assessment (situations or actions effects estimation)</td>
</tr>
<tr>
<td></td>
<td>4 - Process refinement (support objects)</td>
</tr>
<tr>
<td>Dasarathy’s Functional Model [8]</td>
<td>Three levels of abstraction: data, fusion and decision.</td>
</tr>
<tr>
<td></td>
<td>Categorization of data fusion functions in terms of the type of data level at input/output.</td>
</tr>
<tr>
<td></td>
<td>Omission of feedback data flow is the major limitation.</td>
</tr>
<tr>
<td>Thomopoulos’ Fusion model [23]</td>
<td>Three levels: Signal (correlation and learning), Evidence (statistical model) and Dynamics (mathematical model).</td>
</tr>
<tr>
<td>Durrant-Whyte [9]</td>
<td>Data from all the sensors converted to a Common Representation Format (CRF) and fused.</td>
</tr>
<tr>
<td>Ominus process model [3]</td>
<td>Cyclic structure like OODA, but more fine-grained: O (Signal processing, Sensing), O (Pattern processing and Feature extraction), D (Decision making, Context processing), A (Control, Resource tasking)</td>
</tr>
<tr>
<td></td>
<td>Fusion levels: sensors, pre-processing, low-level fusion, data analysis, mixture level, variable interpretation, decision module.</td>
</tr>
<tr>
<td></td>
<td>Levels described as classes with attributes and functions.</td>
</tr>
</tbody>
</table>

Table 2.1: Fusion models summary [16]
<table>
<thead>
<tr>
<th>Technique</th>
<th>Type of method</th>
<th>JDL Level</th>
<th>Process functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman Filter</td>
<td>Estimation</td>
<td>1</td>
<td>Position and attribute estimation</td>
</tr>
<tr>
<td>Probabilistic inference</td>
<td>Inference</td>
<td>1</td>
<td>Object correlation. Identity estimation</td>
</tr>
<tr>
<td>Neural network</td>
<td>AI</td>
<td>2-3</td>
<td>Event interpretation. Intent prediction</td>
</tr>
<tr>
<td>Fuzzy logic</td>
<td>AI</td>
<td>2</td>
<td>Event interpretation</td>
</tr>
<tr>
<td>KBS</td>
<td>AI</td>
<td>2-4</td>
<td>Interpretation, prediction, management</td>
</tr>
</tbody>
</table>

Table 2.2: Techniques summary [16]

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generally, less bandwidth required</td>
<td>More bandwidth required if many sensors need data from many others</td>
</tr>
<tr>
<td>Generally, less latency between sensed event and responded action</td>
<td>Generally, less accuracy</td>
</tr>
<tr>
<td>More reliability and flexibility</td>
<td></td>
</tr>
</tbody>
</table>

Table 2.3: Distributed Data Fusion advantages/disadvantages [16]
Having answered the research questions, a final conclusion can be reached. Multisensor Data Fusion is a broad issue due to the wide range of scenarios that it can be applied to. Therefore, to find a definition, a model or an algorithm scheme that is explicit, meaning that it can be followed to implement a real system, and at the same time usable for any kind of application, is an unfeasible task. Hence, a view of the different approaches, theories and implementations in the issue of sensor fusion can be presented, intending to be useful as a collection of different ideas that should be combined in the implementation of a real fusion system.

2.2 Augmented Reality

Augmented Reality (AR) deals with the integration of virtual content in a real environment in real time. Azuma presented in [2] a definition that has been mostly accepted:

AR is a system that has the following three characteristics: (1) Combines real and virtual; (2) Interactive in real time; (3) Registered in 3-D.

The difference between Virtual Reality (VR) and Augmented Reality (AR) is that in the first one the user cannot see the real world around him, he is immersed in a completely computer-generated world. In contrast, AR allows the user to see the real world, with virtual objects superimposed upon, or composed with the real world.

van Dartel et al. presented in [24] a discussion about the differences between VR and AR for artworks and a rule for immersion:

The strongest sense of reality is created by maximising the similarity in sensory-motor patterns that exist between interacting with the real world and interacting with the artwork.

As is clear from the prior quote, for a realistic effect the virtual objects have to be correctly aligned to the real scene [13]. This alignment depends completely on accurate tracking of the viewing pose: position and orientation [26].

There are several technologies that can be used for this accurate tracking, some of them are [17]:

- Inertial Navigation System (INS): Composed of accelerometers and gyroscopes, it provides rates of acceleration and rotation.
- GPS: Satellite navigation system that can use a second receiver for a more accurate tracking (Differential GPS).
- Ultrasound tracking: Using indoor ultrasound receivers.
- Vision-based positioning: Using video data to estimate the position.
The main method to achieve an accurate tracking is the use of two or more of these technologies, fusing the positioning data. Generally, positioning data can be classified in two groups:

- **Relative data**: Output data at high frequency. However, due to their relative accumulation of data, errors also accumulate with time.

- **Absolute data**: Output data at low frequency with low error average. However, due to noise and other effects, high frequency error has a considerable level.

Hence, the method most used in tracking for AR make use (at least) of one relative data technology and one absolute data technology.

### 2.3 Analysis of user behaviour

Analysis of user behaviour is the scientific study of people behaviour. It aims to describe and recognize behaviours while subjects engage in various activities under a certain situation. It can be applied to multiple fields, as movement or gesture recognition, surveillance systems, emotion recognition or observation of cognitive processes.

Behaviour can be observed following several approaches as self report (interviews, questionnaires), direct observation by an observer and computer based observation. Last observation method has become more interesting as the computer processing has extended through so many fields of study.

When human behaviours become complex by spanning in time and space, modelling is needed for the computer behavioural recognition by inferring unobservable information about a user from the observable information. Techniques that can be used for modelling user behaviours are for example: Hidden Markov Model (and multilayer versions of the HMM), Bayesian networks and Neural networks. An interesting feature that allows the modelling of the user behaviour is the possibility to predict future user actions.

As it will be used in the project, behaviour analysis is focused on the study of human behaviour in virtual environments by self report and computer based observation. Behavioural study in virtual environments can be useful at for example, training simulators or treating psychological disorders.
Chapter 3

Design

Based on the multisensor data fusion background, a fusion system is designed to accomplish the following objectives:

- Provide accurate and high rate position data to the AR platform.
- Add hand gesture interaction to the AR platform.

This objectives must be achieved observing the next requirements:

- Make use of the available hardware
- Integrate the developed algorithms in the actual software.

The AR platform uses an stereo camera to record the image that, once the virtual content is added, is displayed in a Head Mounted 3D Display. Both devices are placed on a helmet structure that the player put on. In the same structure, the position and orientation devices are placed. The player is also carrying a laptop that is connected to all the helmet devices and process the 3D game engine and the position and orientation calculations. This laptop is connected to the network through a wireless link to have access to external devices or servers.

The fusion architecture is designed starting from the selected hardware: an ultrasound positioning system that will be combined with an inertial navigation system to obtain the accurate position data, and a hand gesture device.

Description of this hardware is followed by the system design.

3.1 Hardware description

A description of the main devices used in the progress of this project is followed. The possibilities allowed by these devices and the limitations that they have are an important issue at designing the global system and a decisive factor at implementing the fusion algorithms.
3.1.1 Ultrasound positioning system

3.1.1.1 Hexamite hardware

The ultrasound positioning system used is made by Hexamite\(^1\). Their “HX11” system consists of a (potentially large) number of receivers on one “multi-drop” RS485 serial network, and some hand-held (or body-mountable) transmitters.

The transmitters work autonomously, transmitting the ultrasound signal where a unique ID that identifies the transmitter is encoded. The receivers have no way of knowing exactly when a transmitter sends the signal. Hence, receivers refer the transmission to their internal clock, storing the transmitter ID with a Time-of-Arrival (ToA).

The server must collect all the stored events in a polling process and transform the ToA information into a Time-Difference-of-Arrival (TDoA) data.

3.1.1.2 Distribution of receivers in the environment

The ultrasound receivers must be placed to capture the signal from the transmitters. The aim of the positioning system is to provide the position of the player. Therefore, the transmitter is placed in the head of the player. Hence, receivers are placed in the ceiling.

Three factors are involved in the distribution of the receivers:

- Number of receivers
- Covered area
- Precision

Defining an area to be covered, as many receivers and more important, as the distribution of the receivers in the different 3-D axes is more spread, the precision of the position that can be determined is bigger.

Therefore, as the variation of the position of the receivers that are able to receive the signal from the transmitter in a location is bigger in one axis, the precision of the target location in this axis is bigger.

Moreover, the limitation of the distance and the angle that the transmitted signal can reach must be taken into account, in order to keep always at least four receivers in the range of the transmitted signal.

In this project, the receivers were held in the ceiling with the coordinates\(^2\) shown in Table 3.1. Because of physical constraints in the environment, the distribution of the receivers is not uniform for all the axes, existing more variation for the x axis and a poor variation for the z axis. This distribution will affect to the precision of the positioning system.

\(^1\)http://www.hexamite.com

\(^2\)In this report, when a location is referred to the absolute world coordinate system, it is based on a Cartesian coordinate system where z is the vertical axis.
### Table 3.1: Ultrasound receivers location

<table>
<thead>
<tr>
<th>x (m)</th>
<th>y (m)</th>
<th>z (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.985</td>
<td>-0.010</td>
<td>6.515</td>
</tr>
<tr>
<td>8.000</td>
<td>0.005</td>
<td>6.480</td>
</tr>
<tr>
<td>5.980</td>
<td>-0.005</td>
<td>6.490</td>
</tr>
<tr>
<td>3.985</td>
<td>0.030</td>
<td>6.510</td>
</tr>
<tr>
<td>1.970</td>
<td>0.010</td>
<td>6.475</td>
</tr>
<tr>
<td>-0.040</td>
<td>0.000</td>
<td>6.455</td>
</tr>
<tr>
<td>1.000</td>
<td>-1.715</td>
<td>6.500</td>
</tr>
<tr>
<td>3.010</td>
<td>-1.715</td>
<td>6.480</td>
</tr>
<tr>
<td>5.000</td>
<td>-1.735</td>
<td>6.530</td>
</tr>
<tr>
<td>6.990</td>
<td>-1.735</td>
<td>6.500</td>
</tr>
<tr>
<td>9.000</td>
<td>-1.740</td>
<td>6.495</td>
</tr>
<tr>
<td>10.975</td>
<td>-1.720</td>
<td>6.530</td>
</tr>
<tr>
<td>9.250</td>
<td>-0.200</td>
<td>7.300</td>
</tr>
<tr>
<td>8.340</td>
<td>0.260</td>
<td>5.250</td>
</tr>
<tr>
<td>4.885</td>
<td>-0.795</td>
<td>7.290</td>
</tr>
<tr>
<td>1.105</td>
<td>0.110</td>
<td>5.280</td>
</tr>
</tbody>
</table>

### Table 3.2: InertiaCube3 specifications

<table>
<thead>
<tr>
<th>Degrees of freedom</th>
<th>3 (Yaw, Pitch and Roll)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angular Range</td>
<td>Full 360° - All Axes</td>
</tr>
<tr>
<td>RMS Accuracy</td>
<td>1° in yaw, 0.25° in pitch &amp; roll at 25°C</td>
</tr>
<tr>
<td>RMS Angular Resolution</td>
<td>0.03°</td>
</tr>
<tr>
<td>Serial Interface Update Rate</td>
<td>180 Hz</td>
</tr>
</tbody>
</table>

#### 3.1.2 Inertial Navigation System

The Inertial Navigation System (INS) used is the InertiaCube3 by InterSense\(^3\). This device is a precision 3-DOF orientation tracking that contains three magnetometers, three accelerometers and three gyroscopes to produce an absolute orientation in 3d-space.

Libraries for different operating systems are provided that allow the user to obtain the raw data from the different sensors or to obtain the orientation and the acceleration of the target after being filtered, corrected and transformed into the desired coordinates.

Some parameters can be adjusted to modify the characteristics of the processed data.

Most important specifications of the InertiaCube3 for the purpose of this project are in Table 3.2.

---

\(^3\)http://www.isense.com
3.1.3 Hand gesture system

Two devices were used while carrying out this thesis. Initially, the MUSH device was used for the analysis and development of the gesture recognition system. After finding a problem in this device that made impossible the execution of the desired task, the Wii remote was used. In Section 4.3.4, an explanation of the problem and how was carried out the change of the device is given.

A description of both devices is followed.

3.1.3.1 MUSH

The MUSH device is a hand-held gesture-capture device designed for the “MUSH Rooms” (Multi User Sensory Hallucination Rooms) installation at V2 [21]. A picture of the device is shown in Figure 3.1.

The MUSH device contains two dual-axis accelerometers, placed orthogonally, and everything needed to convert and transmit the signal via Bluetooth. The specifications of the accelerometers are shown in Table 3.3 and a scheme of the device is shown in Figure 3.2.

Since a game controller from Logitech is used for the MUSH device, the
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensed axes</td>
<td>2</td>
</tr>
<tr>
<td>Acceleration range</td>
<td>+/- 2g</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>±0.5 %</td>
</tr>
<tr>
<td>Alignment error</td>
<td>±1 degree</td>
</tr>
<tr>
<td>Cross axis sensitivity error</td>
<td>±2 %</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>500mV/g</td>
</tr>
<tr>
<td>Noise Density (rms)</td>
<td>0.3 mg/√Hz</td>
</tr>
<tr>
<td>Typical Bandwidth</td>
<td>&gt; 100 Hz</td>
</tr>
</tbody>
</table>

Table 3.3: MUSH-device accelerometers specifications

Figure 3.2: MUSH-device scheme
device is recognized by a computer as a joystick. The data that can be obtained from the device is a value of the acceleration received in the four axes - two for each accelerometer - and the buttons pressed. How to deal with the data will depend in the application and the operating system.

There is known drawback in the device. Only values of acceleration bigger than a certain limit are considered by the game controller. Hence, a death zone is found in the transfer function between the real acceleration and the transmitted value of acceleration. The transfer function is shown in Figure 3.3.

3.1.3.2 Wii Remote

The Wii Remote, sometimes nicknamed "Wiimote", is the primary controller for Nintendo’s Wii console. A main feature of the Wii Remote is its motion sensing capability, which allows the user to interact with and manipulate items on screen via movement and pointing through the use of accelerometer and optical sensor technology. For the purpose of this project, only the accelerometer sensors are used.

The accelerometer sensors that are contained in the Wii Remote have the specifications shown in Table 3.4.

Since the release of the Wii console, people have been exploring new ways in which to use the Wii Remote. Several third-party applications are available to use the Wii Remote with a computer.

The Wii Remote communicates with the Wii via a Bluetooth wireless link,
<table>
<thead>
<tr>
<th>Sensed axes</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acceleration range</td>
<td>+/- 3.6 g</td>
</tr>
<tr>
<td>Nonlinearity</td>
<td>+/-0.3 %</td>
</tr>
<tr>
<td>Alignment error</td>
<td>+/-1 degree</td>
</tr>
<tr>
<td>Cross axis sensitivity error</td>
<td>+/-1 %</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>300mV/g</td>
</tr>
<tr>
<td>Typical Bandwidth</td>
<td>1.6 kHz</td>
</tr>
<tr>
<td>Noise Density (rms)</td>
<td>0.28 mg/√Hz</td>
</tr>
</tbody>
</table>

Table 3.4: Wiimote accelerometer specifications

![Figure 3.4: Wii Remote](image)

The Wii Remote has a number of different data reporting modes. Each of these modes combines certain Core data features with data from external peripherals. The reporting mode was adjusted in the daemon to receive the data from the buttons and the accelerometers of the Wii Remote.

In Picture 3.4 the Wii Remote and how the axes are defined are shown.

### 3.1.3.3 MUSH-device vs Wii Remote

For the purpose of this project, the most important factor of the characteristics of the hand gesture devices are the specifications of the accelerometers. Comparing the accelerometer of both devices, the accelerometer of the Wii Remote presents a better performance at errors level, sensitivity and noise, and what is
more interesting, the range of acceleration that can measure is bigger.

Moreover, no death zone is found in the transfer function of the Wii Remote, which allows to receive any level of acceleration detected by the accelerometer.

### 3.2 Fusion system

From the starting point of the selected hardware and based on the knowledge acquired in the research about multisensor data fusion, a fusion system is designed, following the objectives and requirements defined at the beginning of this chapter.

#### 3.2.1 JDL approach

As a first step, the fusion of the data provided by the different sensors is divided into different processes and layered following the JDL data fusion model [12]. A diagram of the JDL fusion model is shown in Figure 3.5.

In each level of the JDL model, different processes are carried out. A short description of the different levels and the designed fusion processes at each level is followed.

**Level 0 or source preprocessing** enables the data fusion process to concentrate on the data most pertinent to the current situation as well as reducing the data fusion processing load.

- Preprocessing of the data from the different sensors is carried out.

**Level 1 processing** combines locational, parametric, and identity information to obtain representatives of individual objects.

- Combination of data from the different ultrasound receivers to obtain a target location.
• Gravity subtraction and errors correction to obtain acceleration rates and absolute orientation of the player.

• Fusion of ultrasound position and acceleration rates to obtain an accurate position.

• Recognition of hand gesture.

**Level 2 processing** attempts to develop a contextual description of the relationship between objects and observed events.

• Fusion of recognized hand interaction with position and orientation of the player to obtain an interaction with a virtual object.

**Level 3 processing** projects the current situation into the future to draw inferences about vulnerabilities and opportunities for operation, based on a priori knowledge and predictions about the future situation.

• Analysis of player behaviour.

**Level 4 processing** is a meta-process - a process concerned with other processes.

• Decision of modifications in the fusion system.

In Figure 3.6 a diagram of the different designed processes layered in the JDL levels is shown.

### 3.2.2 System Architecture

After this first view of how to divide the system in different levels, the General Data Fusion Architecture by Carvalho et al. [6] is followed. In this architecture, shown in Figure 3.7, the data fusion is divided in three types: data oriented, variable oriented and mixture of data and variable fusion; considering data as the measurement of the environment that is generated by a sensor and variable as the result of an analysis of the data.

The possible levels of the architecture are described and the attributes and functions for the designed system, following UML (Unified Modelling Language), are presented.

• Sensors that measure physical variables and which can be individual units or multiple sets of redundant sensors.

Three types of sensors are used: ultrasound sensors, INS and hand acceleration. An identification code and the position in case it is an ultrasound sensor are the other attributes needed. The only function that they can execute is to send the data.
Pre-processing of the signal that comes from the sensor, including AD conversion, error analysis, amplification, filtering...

Data from the different sensors is filtered and synchronized.

Low-level data fusion, where data is fused before analysis.

Using the TDOA data from the different ultrasound sensors, a position of the target is estimated. The acceleration and the head pose from the INS is transformed into the world axes coordinates using the information from the different sensors of the INS. The acceleration of the hand is transformed into the world axes coordinates by obtaining the direction of the gravity at the hand gesture device.

Data analysis, where data is analyzed to obtain variables.

Two variables are obtained: the head orientation and the hand orientation.

Mixture fusion level, where data and variables are fused.

An accurate position is obtained from the fusion of the ultrasound position and the acceleration from the INS. Recognition of the hand movement from the acceleration is carried out.

High-level variable fusion, where variables obtained from data analysis are fused.
Figure 3.7: Carvalho et al. Data Fusion Architecture [6]
From the hand movement and the hand orientation a gesture is recognized. The virtual object that the player is pointing is selected from the position and the orientation of the player.

- Variable interpretation, where all the views of a variable are fused to obtain a view about the sensed environment.

An interaction that corresponds to the recognized gesture is assigned to the selected virtual object.

- Decision module, where modifications to the environment sensing, actuators or algorithms are decided, based on the information received.

A class diagram where the fusion architecture and the different attributes and functions that were designed is shown in Figure 3.8.

### 3.3 Conclusions

The objectives of providing accurate position and hand gesture interaction to the AR platform, with the requirements of using the available hardware and software, were stated.

From the characteristics of the selected hardware, a fusion system is designed. Based on the description of the theoretical fusion models, the system is designed by defining the multiple fusion process that are implemented in the different fusion levels.

Therefore, a description of how to apply multisensor data fusion to the design of an AR platform was given, and can be summarized in the following steps:

- Define the objectives and requirements of the fusion system.

- Following one or several fusion models, define different fusion processes to accomplish the objectives, observing the hardware capabilities.

- Define the fusion architecture, combining the ideas of the selected fusion models.

Different fusion process were described and are implemented in the following chapter.
Figure 3.8: Fusion architecture class diagram
Chapter 4

Implementation

Having designed the fusion architecture, the different fusion processes must be implemented.

4.1 Position location using TDOA

In order to obtain the position of a target the TDOA (Time Difference of Arrival) technique is based on estimating the difference between the arrival times of the signal from the source at multiple receivers and applying a technique called multilateration or hyperbolic positioning.

Multilateration or hyperbolic positioning is based on defining hyperbolic functions of the target position and obtaining the position by the intersection of these functions.

Working in 2-D, the time difference between two receivers defines a hyperbola - a curve consisting of two distinct and similar branches, formed by the intersection of a plane with a right circular cone - on which the target may exist. Repeating the procedure with another receiver in combination with any of the other previously used receivers, another hyperbola is defined and the intersection of the two hyperbolas results in the position location estimate of the target.

Working in 3-D the procedure is similar. With two receivers the target can be located in a hyperboloid - a quadric surface having a finite center and some of its plane sections hyperbolas. With a third receiver a new hyperboloid is defined and the intersection of both hyperboloid describes a curve where the target lies. Introducing a fourth receiver, a third measurement is available and a new hyperboloid defined. The intersection of this third hyperboloid and the previously obtained curve defines a unique point in space, the position location estimation of the target.

Therefore, the position location estimation is accomplished in two stages:

- Estimation of the TDOAs of the signal from the target between pairs of receivers.
• Use of efficient algorithms to produce an unambiguous solution to the nonlinear hyperbolic equations obtained from the TDOA measurements.

The rest of the section is focused on the different algorithms that can be used and the selection and implementation of one of them. A description of the estimation of the TDOAs is presented in [22].

4.1.1 Hyperbolic Equation Solving Algorithms

The obtained TDOA estimates are converted into range difference measurements and these measurements can be converted into nonlinear hyperbolic equations. Due to these equations are non-linear, solving them is not a trivial operation.

Firstly, the mathematical model that is used by the different algorithms is presented. The discussion will be followed for the 2-D case, the extension to the 3-D case is followed at the algorithm implementation description.

Assuming there are M receivers - indexed as \( i = 0 \ldots (M - 1) \)- arbitrarily distributed in the 2-D plane and that all the TDOAs are referred to the receiver to which the signal arrives first - indexed as \( i = 0 \)-, the squared range distance between the target and the ith receiver is given as:

\[
R_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2} \tag{4.1}
\]

where \((X_i, Y_i)\) is the known location of the ith receiver and \((x, y)\) is the target location.

The range difference between the receivers with respect to the first receiver is:

\[
R_{i,0} = c d_{i,0} = R_i - R_0 = \sqrt{(X_i - x)^2 + (Y_i - y)^2} - \sqrt{(X_0 - x)^2 + (Y_0 - y)^2} \tag{4.2}
\]

where \(c\) is the signal propagation speed, \(R_{i,0}\) is the range difference distance between the first receiver and the ith receiver, \(R_0\) is the distance between the first receiver and the target and \(d_{i,0}\) is the estimated TDOA between the first receiver and the ith receiver.

A set of non-linear hyperbolic equations is defined, whose solution gives the location of the target.

A common approach is to linearize this set of equations. For most situations, linearization of the non-linear equations does not introduce undue errors in the position location estimate. However, linearization can introduce significant errors when determining a solution in bad geometric situations. Linearization can be done by expanding the equations through the use of Taylor-series or by transforming the set of non-linear equations. Rearranging 4.2 into:

\[
R_i^2 = (R_{i,0} + R_0)^2 \tag{4.3}
\]

Equation 4.1 can be rewritten as:
\[
R_{i,0}^2 + 2R_{i,0}R_0 + R_0^2 = X_i^2 + Y_i^2 - 2X_i x - 2Y_i y + x^2 + y^2
\]  
(4.4)

Subtracting 4.1 at \(i = 0\) from 4.4, results in:

\[
R_{i,0}^2 + 2R_{i,0}R_0 = X_i^2 + Y_i^2 - 2X_{i,0} x - 2Y_{i,0} y + x^2 + y^2
\]  
(4.5)

where \(X_{i,0} = X_i - X_0\) and \(Y_{i,0} = Y_i - Y_0\). The new equations are linear with the target location \((x, y)\) and the range of the first receiver to the target \(R_0\) as the unknowns.

The estimation of the position can be reached using different algorithms.

**Friedlander’s Method**  Friedlander’s method utilizes a Least Squares (LS) and a Weighted LS (WLS) error criterion to solve for the position location estimate.

**Spherical-Intersection Method**  In the spherical-intersection (SX) method, \(R_0\) is assumed to be known and \((x, y)\) are obtained in terms of \(R_0\). To obtain \(R_0\), and consequently \((x, y)\), the least square solution of 4.1 is used. The solution is not optimal, as demonstrated in [20], because of the initial assumption that \(R_0\) is constant, reducing the degree of freedom to minimize the error vector.

**Spherical-Interpolation Method**  In the spherical-interpolation (SI) method, \((x, y)\) are firstly solved in terms of \(R_0\). The intermediate result is inserted back into 4.5 to generate equations with \(R_0\) as the only unknown. Therefore, the final result is obtained by substituting the value of \(R_0\) that minimizes the LS error to the intermediate result. Although the SI method was shown in [20] to outperform the result of the SX method, the solution is not optimal because it assumes the independence of \((x, y)\) and \(R_0\), ignoring their relationship.

**Divide-and-Conquer Method**  Divide and Conquer (DAC) method consists of dividing the TDOA measurements into groups of a size equal to the number of unknowns. Solution of the unknowns is calculated for each group and then appropriately combined to provide a final solution. This method can achieve optimum performance. However, the solution uses stochastic approximation and requires that the Fisher information be sufficiently large, that implies errors to remain small. There is also other drawback, information from receivers that cannot be grouped is not used.

**Taylor-Series Method**  The Taylor-series method linearizes the set of equations in 4.2 by Taylor-series expansion, then uses an iterative method to solve the system of linear equations. The iterative method begins with an initial guess and improves the estimate at each iteration by determining the local linear least-square (LS) solution. Although it can provide an accurate result, it is robust and it can make use of redundant measurements, it requires a good initial guess and can be computationally intensive.
Method  | Solution               | Redundant measurements | Computational load |
---      | ---                    | ---                    | ---                |
Friedlander's method | Suboptimal             |                       |                    |
SX method | Suboptimal             |                       |                    |
SI method | Suboptimal             |                       |                    |
DAC      | Optimal if Fisher      | If receivers can       |                    |
information is large | be grouped             |                       |                    |
Taylor-  | Optimal, but there is a| Yes                    | High               |
Series   | risk of converging to  |                       |                    |
method   | a local minimum if     |                       |                    |
          |  given a bad starting  |                       |                    |
          | “seed”                 |                       |                    |
Fang’s   | Closed form exact      | No                     | Low                |
method   | solution               |                       |                    |
Chan’s   | Closed form exact      | Yes                    | Low                |
method   | solution               |                       |                    |

Table 4.1: TDOA methods comparison

**Fang’s Method** Fang provides in [11] an exact solution when the number of equations equals the number of unknown target coordinates to be solved. However, redundant measurements cannot be used. This method experiences an ambiguity problem due to the inherent squaring operation, that can be solved using “a priori” information.

**Chan’s Method** Chan proposed in [7] a non-iterative solution to the hyperbolic position estimation problem which is capable of achieving optimum performance. When TDOA estimation errors are small, this method is an approximation to the maximum likelihood (ML) estimator. It can also take advantage of redundant measurements. However, it needs “a priori” information to solve an ambiguity in its calculations like the Fang’s method.

Antique presented in [1] a comparison of these methods summarized in Table 4.1.

Therefore, Chan’s method is selected to implement a TDOA positioning algorithm, for providing and optimal solution in a closed form, being able to make use of redundant measurements and having a low computational requirements. The mathematical description of the method is followed.

### 4.1.2 Chan’s method

The description of the method for the 2-D case as presented in [7] is followed.

The method makes a distinction if there is a linear relationship in the position of the receivers and if there are three receivers - minimum number of receivers needed- or more.
If the receivers are arbitrary distributed and there are three receivers, \((x, y)\) can be solved in terms of \(R_0\) from 4.5 as:

\[
\begin{bmatrix}
  x \\
  y
\end{bmatrix} = -\begin{bmatrix}
  X_{1,0} & Y_{1,0} \\
  X_{2,0} & Y_{2,0}
\end{bmatrix}^{-1} \left\{ \begin{bmatrix}
  R_{1,0} \\
  R_{2,0}
\end{bmatrix} R_0 + \frac{1}{2} \begin{bmatrix}
  R^2_{1,0} - K_1 + K_0 \\
  R^2_{2,0} - K_2 + K_0
\end{bmatrix} \right\} \quad (4.6)
\]

where \(K_i = X^2_i + Y_i^2\).

Inserting this intermediate result into 4.3 at \(i = 0\), a quadratic equation in terms of \(R_0\) is produced. Substituting the positive root back into 4.6 results in the final solution. There may exist two positive roots from the quadratic equation that can produce two different solutions, resulting in an ambiguity. This problem has to be solved using “a priori” information.

If the receivers are arbitrary distributed and there are more than three receivers - \(M\) receivers - the system is overdetermined as the number of measurements is greater than the number of unknowns. In the presence of noise, the set of equations will not meet at the same point.

Let \(z = [x \ y \ R_0]\) be the unknown vector. With noise, the error vector derived from 4.5 is \(\psi = h - G_a z^0\), where \((0)\) represents the non presence of noise and:

\[
h = \frac{1}{2} \begin{bmatrix}
  R^2_{1,0} - K_1 + K_0 \\
  R^2_{2,0} - K_2 + K_0 \\
  \vdots \\
  R^2_{M-1,0} - K_{M-1} + K_0
\end{bmatrix}
\]

\[
G_a = \begin{bmatrix}
  X_{1,0} & Y_{1,0} & R_{1,0} \\
  X_{2,0} & Y_{2,0} & R_{2,0} \\
  \vdots & \vdots & \vdots \\
  X_{M-1,0} & Y_{M-1,0} & R_{2,0}
\end{bmatrix}
\]

Expressing \(R_{i,0}\) as \(R_{i,0}^0 + cn_{i,0}\) with \(n_{i,j}\) denoting the noise component, and noting from 4.2 that \(R_i = R_{i,0}^0 + R_0^0\), \(\psi\) is found to be:

\[\psi = cBn + 0.5c^2 n \odot n\]

where \(B = \text{diag} \{R_{1,0}^0, R_{2,0}^0, ..., R_{M-1,0}^0\}\) and \(n = [n_{1,0}, ..., n_{M-1,0}]\) is the noise vector.

The covariance matrix of \(\psi\) can therefore be evaluated as:

\[
\Psi = E[\psi\psi^T] = c^2 BQB
\]

where \(Q\) is the covariance matrix of the estimated TDOA vector \(d = [d_{1,0}, ..., d_{M-1,0}]\).

The approach to solve the non-linear problem is to first assume that there is no relationship among \(x\), \(y\), and \(R_0\), solve them by LS and then obtain the final solution by imposing their known relationship. Therefore, the ML estimator of \(z\) is:
\[ z = \arg\min \left\{ (h - G_\alpha z)^T \Psi^{-1} (h - G_\alpha z) \right\} = (G_\alpha^T \Psi^{-1} G_\alpha)^{-1} G_\alpha^T \Psi^{-1} h \quad (4.7) \]

Due to \( \Psi \) is not known as \( B \) contains the true distances between the target and the receivers, further approximation is needed. When the target is far from the receivers, \( R_0^2 \) is close to \( R^0 \) so that \( B \approx R^0 I \) where \( R^0 \) designates the range and \( I \) is an identity matrix of size \( M-1 \). An approximation of the previous estimator is:

\[ z \approx (G_\alpha^T Q^{-1} G_\alpha)^{-1} G_\alpha^T Q^{-1} h \quad (4.8) \]

If the target is close, \( 4.8 \) can be used firstly to obtain an initial solution of \( B \) and then the final solution can be computed from \( 4.7 \).

The case of the receivers linearly distributed is now described. When sensors are arranged linearly the matrices involved are singular, and previous formulae require matrices to be full rank. Therefore, equation \( 4.4 \) is rewritten using the relationship between the coordinates of the receivers position, \( Y_i = aX_i + b \).

\[ -2X_{i,0} (x + ay) - 2R_{i,0}R_0 = R_{i,0}^2 - K_i + K_0 \quad (4.9) \]

This equation is linear in \( w = (x + ay) \) and \( R_0 \). A similar solution to \( 4.7 \) can be obtained:

\[ z_l = (G_l^T \Psi^{-1} G_l)^{-1} G_l^T \Psi^{-1} h \quad (4.10) \]

where

\[ z_l = \begin{bmatrix} x + ay \\ R_0 \end{bmatrix}; \quad G_l = \begin{bmatrix} X_{1,0} & R_{1,0} \\ X_{2,0} & R_{2,0} \\ \vdots & \vdots \\ X_{M-1,0} & R_{M-1,0} \end{bmatrix} \]

Since \( x + ay \) and \( R_0 \) are independent, to obtain the position estimate these values are substituted in the next equation:

\[ R_0^2 = (X_0 - x)^2 + (Y_0 - y)^2 \]

The proposed solution requires the knowledge of the TDOA covariance matrix \( Q \) which may not be known in practice. If the noise power spectral densities are similar at sensors, it can be replaced by a matrix of diagonal elements 1 and 0.5 for all other elements.

### 4.1.3 TDOA algorithms implementation

The described method for the 2-D case must be extended to a 3-D implementation. The first step is to distinguish between the different cases based on the distribution of the receivers. Two questions are involved in the distribution of the receivers:
Are all the receivers in the same plane?
Are there four receivers or more than four receivers?

In order to classify each distribution of receivers, the next process to check if the receivers are in the same plane is followed. Using the location of the first three receivers the normal of the plane that they define is calculated by the cross product of the vectors from the first receiver to the second and the third one: \( \overrightarrow{P_0P_1} \times \overrightarrow{P_0P_2} = \overrightarrow{N} \). To check if the rest of the receivers are in the same plane the product of the vector from the first receiver to each receiver with the normal of the plane must be null: \( \overrightarrow{P_0P_i} \times \overrightarrow{N} = 0 \). If any of the products is not null, the receivers are not in the same plane. Following this procedure, the normal vector that is needed in case the receivers are in the same plane is already calculated.

Hence, three cases are defined and differently implemented: Four receivers not in a plane, More than four receivers not in a plane and Receivers in a plane. The algorithm selection method is shown in Figure 4.1.

**Four receivers not in a plane**

First of all, three matrices are defined from 4.6:

\[
A = - \begin{bmatrix} X_{1.0} & Y_{1.0} & Z_{1.0} \\ X_{2.0} & Y_{2.0} & Z_{2.0} \\ X_{3.0} & Y_{3.0} & Z_{3.0} \end{bmatrix}^{-1}
\]

\[
B = \frac{1}{2} \begin{bmatrix} R_{1.0}^2 - K_1 + K_0 \\ R_{2.0}^2 - K_2 + K_0 \\ R_{3.0}^2 - K_3 + K_0 \end{bmatrix}
\]
\[ R = \begin{bmatrix} R_{1,0} \\ R_{2,0} \\ R_{3,0} \end{bmatrix} \]

In order to obtain the quadratic solution of \( R_0 \), the next steps are followed:

\[ E = A \times R \]
\[ F = A \times B \]
\[ a = 1 - E' E \]
\[ b = 2 (X_0 E - F' E) \]
\[ c = 2X_0 F - F' F - K_0 \]
\[ R_0 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \]

The correct root of \( R_0 \) is selected, in this case the positive one, and the target location is estimated by:

\[ \begin{bmatrix} x \\ y \\ z \end{bmatrix} = A \times (R R_0 + B) \]

More than four receivers not in a plane

The next matrices are defined as in 4.8:

\[ G_a = \begin{bmatrix} X_{1,0} & Y_{1,0} & Z_{1,0} & R_{1,0} \\ X_{2,0} & Y_{2,0} & Z_{2,0} & R_{2,0} \\ \vdots & \vdots & \vdots & \vdots \\ X_{M-1,0} & Y_{M-1,0} & Z_{M-1,0} & R_{2,0} \end{bmatrix} \]
\[ h = \frac{1}{2} \begin{bmatrix} R_{1,0}^2 - K_1 + K_0 \\ R_{2,0}^2 - K_2 + K_0 \\ \vdots \\ R_{M-1,0}^2 - K_{M-1} + K_0 \end{bmatrix} \]

where in the 3-D case, \( K_i = X_i^2 + Y_i^2 + Z_i^2 \).

As explained before, the TDOA covariance matrix is defined as:

\[ Q = \begin{bmatrix} 1 & 0.5 & 0.5 & \ldots & 0.5 \\ 0.5 & 1 & 0.5 & \ldots & 0.5 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ 0.5 & 0.5 & 0.5 & \ldots & 1 \end{bmatrix} \]
A first estimate is obtained,

$$\begin{bmatrix} x \\ y \\ z \\ R_0 \end{bmatrix} = (G_a^T Q^{-1} G_a)^{-1} G_a^T Q^{-1} h$$

and using the calculated $R_0$, a second estimate can be generated:

$$\begin{bmatrix} x \\ y \\ z \\ R_0 \end{bmatrix} = (G_a^T Y^{-1} G_a)^{-1} G_a^T Y^{-1} h$$

where

$$Y = c^2 BQB$$

being $R_i = R_{i,0} + R_0$ and $c$ the propagation velocity of the signal.

**Receivers in a plane**

When all the receivers are located in a plane, their coordinates are related by the next equation:

$$a X_i + b Y_i + c Z_i + d = 0$$

As explained before, equation 4.4 must be rewritten including this relationship. Hence, new variables are defined to make the equation linear.

Three pairs of variables can be defined when the next conditions are satisfied:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Variable $v$</th>
<th>Variable $w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \neq 0$</td>
<td>$v = y - \frac{b}{a} x$</td>
<td>$w = z - \frac{c}{a} x$</td>
</tr>
<tr>
<td>$b \neq 0$</td>
<td>$v = x - \frac{a}{b} y$</td>
<td>$w = z - \frac{c}{b} y$</td>
</tr>
<tr>
<td>$c \neq 0$</td>
<td>$v = x - \frac{a}{c} z$</td>
<td>$w = y - \frac{b}{c} z$</td>
</tr>
</tbody>
</table>

Furthermore, in order to maintain the linearity of the previous variables, the parameter $d$ of the plane equation must be null. Therefore, the location of the receivers is transformed into a new coordinate system where the origin is the closest receiver to the target, the receiver with index equal to zero. This transformation is carried out by subtracting the position of the closest receiver to all the positions of the receivers.

Using the previously defined variables equation 4.10 can be evaluated. Depending on the selected variables $v$ and $w$, matrix $G_i$ is defined as in the next table:
Once $v$, $w$ and $R_0$ are obtained, $[x\ y\ z]$ are estimated from the definitions of $v$ and $w$ and the relationship:

$$R_0^2 = (X_0 - x)^2 + (Y_0 - y)^2 + (Z_0 - z)^2$$

A simplification can be applied when the plane defined by the receivers is one of the planes defined by the coordinate system axes - when the receivers plane is the plane XY, XZ or YZ. In these cases, two of the three parameters - $a$, $b$ and $c$ - of the plane equation are equal to zero, and the linear variables $v$ and $w$ are simplified.

Due to the location of the ultrasound receivers in the ceiling, only plane XY will be used in the following explanation.

Hence, when the defined plane is the plane XY, parameters $a$ and $b$ are equal to zero and the linear variables are defined as: $v = x$ and $w = y$. Matrix $G_i$ is defined as in the condition $c \neq 0$. Therefore, evaluation of equation 4.10 generates directly an estimate of $x$, $y$ and $R_0$, and the remaining unknown coordinate can be obtained from:

$$z = \sqrt{-\left( K_0 - 2X_0v - 2Y_0w + v^2 + w^2 - R_0^2 \right)}$$

The computation when the simplification can be applied is simpler. This simplification can be applied when the defined plane is the plane XY. A solution to make this simplification always available is to transform the coordinates system, so that the plane defined by the receivers correspond to the XY plane. This transformation can be done by rotating the location of the receivers by the next rotation matrix:

$$\begin{bmatrix}
X_{1,0} & C_x & N_x \\
Y_{1,0} & C_y & N_y \\
Z_{1,0} & C_z & N_z
\end{bmatrix}$$

where $N$ is the normal vector of the plane and $C$ is the cross-product between $N$ and $[X_{1,0} \ Y_{1,0} \ Z_{1,0}]$.

The estimated location after rotating the receivers position is obtained in the transformed coordinates system. To convert it to the original coordinates system, the position must be rotated by the inverse rotation matrix and corrected by adding the position of the closest receiver.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Matrix $G_i$</th>
</tr>
</thead>
</table>
| $a \neq 0$ | $\begin{bmatrix}
Y_{1,0} & Z_{1,0} & R_{1,0} \\
\vdots & \vdots & \vdots \\
Y_{M-1,0} & Z_{M-1,0} & R_{M-1,0}
\end{bmatrix}$ |
| $b \neq 0$ | $\begin{bmatrix}
X_{1,0} & Z_{1,0} & R_{1,0} \\
\vdots & \vdots & \vdots \\
X_{M-1,0} & Z_{M-1,0} & R_{M-1,0}
\end{bmatrix}$ |
| $c \neq 0$ | $\begin{bmatrix}
X_{1,0} & Y_{1,0} & R_{1,0} \\
\vdots & \vdots & \vdots \\
X_{M-1,0} & Y_{M-1,0} & R_{M-1,0}
\end{bmatrix}$ |
Table 4.2: Ultrasound positioning system precision

<table>
<thead>
<tr>
<th>Axis</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>4 mm</td>
</tr>
<tr>
<td>y</td>
<td>9 mm</td>
</tr>
<tr>
<td>z</td>
<td>26 mm</td>
</tr>
</tbody>
</table>

These algorithms were implemented in Python to use them in the real system. The complete code and an extended description of the hardware and other issues can be found in [22].

4.1.4 Testing

The algorithm was tested in a simulated environment and for every condition (receivers in a plane or not, more or less than four receivers) and the error between the simulated position and the estimated position was always less than $10^{-10}m$.

As it was mentioned in Section 3.1.1.2, the error in one axis becomes lower as the distance covered by the receivers in that axis is bigger. That entails the importance of the receivers distribution in the space.

Likewise, the algorithm is tested in the real system with the receivers configuration described in 3.1.1.2. As it was expected, the precision in the different axes is not constant. In Table 4.2, the precision for the different axes is shown.

Occasional and random errors of high magnitude are also found, originated by the data polling process.

4.2 INS and ultrasound position fusion

4.2.1 Ultrasound signal analysis

In order to characterize the signal from the ultrasound system, a target that follows a known path at a constant speed is set up in the testing space. For practical reasons, a circle is chosen as the path which the target follows.

The obtained position in the horizontal axes is shown in Figure 4.2. As it was known in the implementation of the ultrasound system, random and instantaneous wrong positions are obtained. In order to discriminate the wrong samples, velocity is derived from the position and when the velocity of the sample exceeds a fixed limit, the sample is ignored.

Comparing the obtained signal with an ideal generated signal - the component in one axis of a movement drawing a circle is a sine- as shown in Figure 4.3, a variable delay in the ultrasound signal can be observed. A first hypothesis to explain this result is that the delay between the generation of the ultrasound signal in the transmitter and the reception of the position in the AR server is not constant. To solve this problem, the implementation of the ultrasound server is
modified to support the timestamping of the delay between the position samples. Hence, the delay value between the consecutive samples is added to the position data. In Figure 4.4, the accumulative error of the previous timestamps compared to the improved timestamps is shown.

As a final characterization of the timestamps quality, the signal crosses by zero are obtained, as shown in Figure 4.5, and the time between the crosses is compared, as shown in Figure 4.6. As it is clear from the Figure 4.6, the time is not constant, although it cannot be assumed that this variation is originated by the timestamps because it cannot be affirmed that the target is keeping a constant velocity.

4.2.2 INS signal analysis

A first step characterizing the signal received from the Inertial Navigation System is the study in a static state. In an ideal device, the acceleration of an static object will remain null. As it is shown in Figure 4.7, two kinds of errors are present in the signal:

- Low frequency bias
- High frequency noise

A pre-filtering process of the INS signal becomes clearly needed with two different techniques that solve the previous errors:
Figure 4.3: Comparison of ultrasound signal and ideal signal.

Figure 4.4: Accumulative error of timestamping in ultrasound signal.
Figure 4.5: Crossing by zero of ultrasound signal

Figure 4.6: Time between crosses by zero of ultrasound signal
Figure 4.7: INS acceleration data in static state

- An unbiassing process
- Low-pass filtering

In order to determine the parameters of both techniques a study of the acceleration of a person in a normal movement is required. With this purpose, acceleration is derived from the position data obtained from the ultrasound system. In Figure 4.8 the movement of a person in the testing space and the frequency spectrum of the acceleration derived from this movement is shown. As it becomes clear from the figure, the movement of a person is composed by very low frequency components. Comparing this spectrum with the spectrum of the acceleration obtained from the INS, shown in Figure 4.9, it can be stated that the high frequency components of the signal from the INS do not correspond to the movement of the person. Hence, cut-off frequency can be determined at a low frequency. Another important aspect in the design of a filter is the group delay. This parameter define the transit time of the signal through the filter. Due to the proposed objective of having a fast response of the movement of the player, group delay should be minimized. Therefore, the order of the filter should remain low.

It can also be derived from this study that the acceleration of a person has a short duration in time. Hence, the bias problem can be solved by subtracting to the signal the mean of the last few seconds of signal.
Figure 4.8: Ultrasound signal of the movement of a person

Figure 4.9: INS signal of the movement of a person
The filter would have to maintain an explicit accurate awareness of the motion as well as attempt to suppress noisy and erroneous data.

Filter sampling rate must be at least twice the highest signal frequency, producing high computation load.

If the filter fails, the entire navigation system fails.

If the filter fails, the unaltered INS information is still available.

Errors in the inertial system must remain of small magnitude.

If the filter fails, the unaltered INS information is still available.

Table 4.3: Benefits and Drawbacks of Kalman Filter Implementations [16]

### 4.2.3 Fusion process design

The aim of the fusion of the INS signal and the ultrasound signal is to obtain a position with better characteristics than using any of the previous signals by itself.

As it was presented in Section 2.2, one of the most used techniques in positioning systems for Augmented Reality is to combine a relative and an absolute positioning data.

The positioning data obtained from the ultrasound system is an absolute data and the positioning data obtained from the INS is a relative data.

Most common technique at combining both types of positioning data is the Kalman filter.

The Kalman filter (R. E. Kalman [14]) is a recursive solution to the discrete linear filtering problem. At combining data from multiple sensors, the Kalman filter weights the different data most heavily in the circumstances where they each perform best, thus providing more accurate and stable estimates than a system based on any one sensor alone.

Two methods can be applied in the implementation of the Kalman filter (Brock and Schmidt [5]). In the direct method, the desired parameters are estimated by the filter, while in the indirect method, the errors of the parameters are estimated. Benefits and drawbacks of both implementations are presented in Table 4.3.

Hence, the direct implementation is restricted to alignment, calibrations and
bias determination in laboratory testing. Using the indirect implementation the INS itself follows the high frequency motion, while the dynamics upon which the filter is based is the set of inertial system error propagation equations.

Beside the state space differences, feedforward and feedback mechanization can be implemented. In the indirect feedforward version the errors estimated by the filter are subtracted from the inertial data to obtain an optimal estimate position and velocity. In the indirect feedback the estimated errors are fed back into the INS to correct it, not allowing inertial errors to grow.

Therefore, indirect feedback implementation is chosen. A schematic of the filter is shown in Figure 4.10.

As it was mentioned during the analysis of the INS signal and the ultrasound positioning signal, some errors of these signals must be corrected. These corrections, are carried out at a pre-filtering process.

At the pre-filtering of the INS signal, a low-pass filter and an unbiassing process are applied. At the pre-filtering of the ultrasound positioning signal, a velocity discrimination is applied.

Another important issue at combining both signals is timing. INS signal does not contain any timing information. Therefore, local timestamp is added to INS signal. As it was explained in the analysis of the ultrasound positioning data, the timestamps of the signal are not synchronized with the local time of the AR server. Inter-samples delays are combined with the local time and an initial delay to obtain a corrected timing.

In order to avoid sudden and fast movements of the estimated position, generated by high frequency errors, a final smoothing process is added. This process consists in a low-pass filter.

The complete fusion process diagram is shown in Figure 4.11.

Since the data from the INS and the ultrasound system have different sampling frequency, both signals are not processed simultaneously. The activity diagram, Figure 4.12, shows how the system must be implemented.
The first step in designing the Kalman filter is to define the system model. Due to the indirect implementation of the filter was selected, the errors in the position and velocity indicated by the INS are among the estimated variables, and each measurement presented to the filter is the difference between INS and the ultrasound data.

Many approaches in the literature define the state model based on the errors of the gyroscopes and magnetometers of the INS. As it was explained in Section 3.1.2, the obtained acceleration from the INS is defined in an absolute world coordinate system and the errors of the gyroscopes and magnetometers have been already corrected in a previous stage. The implementation of a Kalman filter based on the corrected output of a previous Kalman filter in a cascade configuration, requires that the external filter have access to the covariance matrix of the internal filter. Since it is not possible to access to the internal values of the internal filter implemented in the INS libraries, the solution obtained by the proposed Kalman filter is theoretically suboptimal.

Therefore, the model of the error in the position obtained from the INS is only dependent of the accelerometers errors. Hence, the state model can be defined as:

\[
\mathbf{x} = \begin{bmatrix} \delta_r \\ \delta_v \\ \beta \end{bmatrix}
\]

where \( \delta_r \) is the error in the position, \( \delta_v \) is the error in the velocity and \( \beta \) is the bias error in the acceleration.

The measurement \( z \) is defined as:
Figure 4.12: INS and ultrasound activity diagram
The process is controlled by the linear difference equation:

\[ x(t) = F(t)x(t) + v(t) \]

with the measurement \( z \) that is related by:

\[ z(t) = H(t)x(t) + w(t) \]

where \( v \) and \( w \) are the process and measurement noise.

In order to determine the state transition model \( F \), \( \dot{x} \) is defined as:

\[
\dot{x} = \begin{bmatrix}
\delta_r \\
\delta_v \\
\beta
\end{bmatrix}
= \begin{bmatrix}
\delta_v \\
\beta
\end{bmatrix}
\]

Therefore

\[
F = \begin{bmatrix}
0 & 1 & 0 \\
0 & 1 & 0 \\
0 & 0 & -1/\tau_\beta
\end{bmatrix}
\]

For the observation model \( H \),

\[
z = [z_r] = \begin{bmatrix}
\delta_r + w_r \\
\delta_v + w_v
\end{bmatrix}
\]

Therefore

\[
H = \begin{bmatrix}
1 & 1 & 0
\end{bmatrix}
\]

Hence,

\[
v = \begin{bmatrix}
0 \\
0 \\
\nu_\beta
\end{bmatrix};
\quad w = \begin{bmatrix}
w_r \\
w_v
\end{bmatrix}
\]

And their corresponding covariances:

\[
Q = \begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 0 \\
0 & 0 & 2\sigma_\beta^2/\tau_\beta
\end{bmatrix}
\]

\[
R = \begin{bmatrix}
\sigma_r^2 \\
\sigma_v^2
\end{bmatrix}
\]

where \( \sigma_\beta^2 \) is the variance of the accelerometer bias and \( \sigma_r^2 \) is the variance of the ultrasound measurement.
### Table 4.4: Summary of Kalman Filter parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>State model</strong></td>
<td>( x = \begin{bmatrix} \delta_r \ \delta_v \ \beta \end{bmatrix} )</td>
</tr>
<tr>
<td><strong>Measurement model</strong></td>
<td>( z = \begin{bmatrix} z_r \ z_v \end{bmatrix} )</td>
</tr>
<tr>
<td><strong>Transition model</strong></td>
<td>( A = \begin{bmatrix} 1 &amp; \Delta t &amp; 0 \ 0 &amp; 1 &amp; \Delta t \ 0 &amp; 0 &amp; 1 - \Delta t/\tau_\beta \end{bmatrix} )</td>
</tr>
<tr>
<td><strong>Observation model</strong></td>
<td>( H = \begin{bmatrix} 1 &amp; 1 &amp; 0 \end{bmatrix} )</td>
</tr>
<tr>
<td><strong>Covariance of process noise of state model</strong></td>
<td>( Q = \begin{bmatrix} 0 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 0 \ 0 &amp; 0 &amp; 2\sigma^2/\tau_\beta \end{bmatrix} )</td>
</tr>
<tr>
<td><strong>Covariance of process noise of measurement model</strong></td>
<td>( R = \begin{bmatrix} \sigma^2_r \ \sigma^2_v \end{bmatrix} )</td>
</tr>
</tbody>
</table>

In discrete time

\[
x_k = Ax_{k-1} + v
\]

\[
z_k = Hx_k + w
\]

where

\[
A = e^{F(t)\Delta t} \approx I + \Delta t F
\]

So,

\[
A = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 - \Delta t/\tau_\beta \end{bmatrix}
\]

In Table 4.4 a summary of the definition of the parameters of the Kalman filter is shown.

Once the filter model is defined an statistical study is carried to determine the parameters: \( \sigma^2_\beta, \sigma^2_r, \sigma^2_v \) and \( \tau_\beta \). This study entails the characterization of the noise of the ultrasound and the INS signals. Assuming the noise is independent, white and with normal probability distribution, we can assume that the characteristics of the signal in the static state of the target are the characteristics of the noise. For the estimation of the decorrelation time of the accelerometer bias the evaluation of the equation is applied to obtain the parameter. The results are shown in Table 4.5.


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variance of accelerometer bias</td>
<td>$\sigma^2_\beta = 0.0012$</td>
</tr>
<tr>
<td>Variance of ultrasound position measurement</td>
<td>$\sigma^2_r = 0.00016$</td>
</tr>
<tr>
<td>Variance of ultrasound velocity measurement</td>
<td>$\sigma^2_v = 0.00025$</td>
</tr>
<tr>
<td>Decorrelation time of $\beta$</td>
<td>$\tau_\beta = 0.55$</td>
</tr>
</tbody>
</table>

Table 4.5: Values of Kalman filter statistical parameters

### 4.2.5 Testing and algorithm refinement

As a first step, the algorithm is implemented\(^1\) in Octave - a GNU program for performing numerical computations which is mostly compatible with MATLAB - and successfully tested with recorded data.

Therefore, it is integrated in the AR platform using the programming language Scheme - a multi-paradigm programming language, which is one of the two main dialects of Lisp. The main part of the implementation of the Kalman filter that can be reused for any Kalman filter implementation is reproduced in Appendix A.1.

The performance of the positioning system is tested subjectively in an AR environment. An interface was designed so the next parameters of the algorithms can be adjusted:

- INS pre-filter: Type of filter (FIR or IIR), order and cutoff window - the cutoff frequency is defined as the window times pi.

- Ultrasound positioning system pre-filter: Velocity limit for the discrimination. Due to the clear less precision in the z axis, axes x and y are processed differently than axis z.

- Smoothing: Making the same distinction for axis z, the type, order and cutoff window for the filter are adjusted.

Relationship between parameters and visual effects

Obtained values after the refinement are shown in Table 4.6.

### 4.3 Hand gesture recognition

#### 4.3.1 Processing of input data

As it was commented in Section 3.1.3.1, the MUSH device is recognized as a joystick device. SDL is used to access to the data. SDL (Simple DirectMedia Layer) is a cross-platform multimedia library designed to provide low level access to audio, keyboard, mouse, joystick, 3D hardware via OpenGL, and 2D video framebuffer, supporting several operating systems. Working with a joystick device, SDL deals with the data as events. Hence, only when new data is

\(^1\)The implemented code is available at: http://svn.v2.nl/andres/octave/ins_hex_fusion

45
Table 4.6: INS and ultrasound fusion parameters refinement

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Filter type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>INS pre-filter</td>
<td>Filter type</td>
<td>FIR</td>
</tr>
<tr>
<td>Order</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Cutoff window</td>
<td></td>
<td>0.012</td>
</tr>
<tr>
<td>Ultrasound pre-filter</td>
<td>Velocity limit axes x,y</td>
<td>3 m/s</td>
</tr>
<tr>
<td>Velocity limit axis z</td>
<td></td>
<td>1 m/s</td>
</tr>
<tr>
<td>Smoothing</td>
<td>Filter type</td>
<td>IIR (Butterworth)</td>
</tr>
<tr>
<td>Axes x, y</td>
<td>Order</td>
<td>1</td>
</tr>
<tr>
<td>Cutoff window</td>
<td></td>
<td>0.036</td>
</tr>
<tr>
<td>Axes z</td>
<td>Order</td>
<td>2</td>
</tr>
<tr>
<td>Cutoff window</td>
<td></td>
<td>0.012</td>
</tr>
</tbody>
</table>

Table 4.7: Hand acceleration data fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Absolute timestamp in milliseconds since the device is initialized</td>
</tr>
<tr>
<td>Device ID</td>
<td>Due to the use of only two devices - for each hand - the value can be 0 or 1</td>
</tr>
<tr>
<td>Axis ID</td>
<td>From 0 to 3: Four acceleration axes (x,y1,z,y2) From 4 to 5: Buttons</td>
</tr>
<tr>
<td>Value</td>
<td>Value between $-2^{15}$ and $2^{15}$</td>
</tr>
</tbody>
</table>

generated - a change in the acceleration is occurred - a new event is created.
Therefore, the data obtained from the MUSH device, through SDL, is composed of different events with the next fields: Device ID, Axis ID and Value.

The processing of the data is time dependant. The same data generated in a short or in a long time is not produced by the same movement. Hence, time information is needed and a timestamp is added.

A description of the resulted fields is shown in Table 4.7.

In order to carry out a temporal signal processing - for example, a filter will be used to detect acceleration from gravity - the data must be resampled to obtain a signal with a uniform sampling.

Two rules are followed in the resampling process:

- Multiple values of acceleration of the same axis, coming from different events occurred between two consecutive timestamps are merged as the mean of the values.

- If a zero value is received, previous values received since the last timestamp are discarded.

Analyzing the difference in time between consecutive events, the resampling frequency can be defined. In Figure 4.13, an histogram of the time difference is
shown. As it becomes clear from the figure, the time difference chosen for the
resampling, for being the most obtained value, is 16ms.

Therefore, the new data generated has the next fields: Timestamp and Values of acceleration of the different axes.

4.3.2 Acceleration signal analysis

As a first step in the analysis of the acceleration signal, a study of the gravity
effect in a static state is carried out.

In order to observer the effect of gravity in one axis, gravity effect is limited
to one axis by pointing this axis down to ground. In Figures 4.14, 4.15 and 4.16
the effect for the different axes is shown.

It becomes clear from the figures that the gravity can be identified as a
constant value in the signal and that this constant value is not the same for
every axis. Repeating the study several times it is also found that the gravity
value in one axis is not the same, as shown in figure.

Therefore, gravity can be extracted from the signal with a low pass filter
and a following normalization of the gravity is needed, so the dimension of the
gravity remains constant.

A study of the response of the signal at a change of gravity, a rotation, is
carried. The effect of a rotation in axis z, starting with axis x pointing down is
shown in Figure 4.17. This implies that the gravity extracted has to represent
Figure 4.14: Gravity effect in axis x

Figure 4.15: Gravity effect in axis y
Figure 4.16: Gravity effect in axis z

The change of the acceleration signal produced by a rotation.

A second step in the analysis is to study the effect of movement in the acceleration received.

The study is carried out with simple and single movements in the different axes, varying the orientation. In Figure 4.18, consecutive and opposite movements in axis x when gravity is present in axis y are shown. It can be observed that this movement is reflected in an axis that is not been affected by gravity by an acceleration peak in the direction of the movement - the start of the movement - and a opposite peak - the end of the movement. On the other hand, the movement on a perpendicular axis is the reflected in the acceleration of the axis affected by gravity as a distortion characterized by multiple peaks with different sign.

In Figure 4.19, a movement in axis y when gravity is present in the same axis is shown. It can be observed that the movement is reflected in the acceleration by a peak and a opposite peak, similar to the effect occurred to an axis with no gravity effect. No effect is observed in the other axes.

Hence, some conclusions can be extracted from this study:

- Gravity must be extracted from acceleration signal by a low pass filter.
  
The extracted gravity has to reflect rotations but not movements.

- Extracted gravity must be normalized to keep gravity dimension constant.
Figure 4.17: Rotation effect

Figure 4.18: Movement in x axis when gravity in y axis
4.3.3 Gesture recognition

Based on the conclusions derived from the signal analysis the technique selected to recognize the movement of the hands is Fuzzy logic. Fuzzy logic is based on fuzzy set theory, where elements have degrees of membership to the different sets, differing from classical set theory, where elements belong or do not belong to a certain set.

The principal objective of fuzzy logic is the formalization of humans capability to reason and make decisions in an environment of uncertainty, imprecision, incompleteness of information, and partiality of knowledge, truth and class membership. Hence, Fuzzy logic is a powerful tool for defining systems through a natural use of language.

In order to extract information from the data, Fuzzy Inference can be constructed following the next steps:

1. Fuzzification: Mapping from numerical values to the membership functions of the fuzzy variables.
2. Rule evaluation: The rules defined are evaluated using the fuzzy logic.

3. Aggregation: The results of the rules are aggregated so that they are mapped to the output variables.

4. Defuzzification: The final step is the mapping from fuzzy output variables to numerical values.

Therefore, data from every axis is mapped to the next fuzzy variables:

- Is gravity present?
- What is the gravity sign?
- Does the acceleration exceed a threshold level?
- What is the acceleration sign?

Information about the dynamics of signal is needed. Not only the information about the actual state of the signal determines the characteristics of the signal, information about how the signal is changing must be available.

In order to do not work with different temporal sets of the previous fuzzy variables, the next dynamic variables are added:

- Does the presence of gravity change?
- Does the sign of gravity change?
- Does the fact that the acceleration exceeds a threshold level change?
- Does the sign of the acceleration change?

These dynamic variables can be seen as the comparison of the first fuzzy variables at the actual time between the previous time.

The fuzzy variables are defined as the axis state and named as in the Table 4.8.

Based on the signal analysis, the next types of detected movements can be defined:

- No movement
- A peak followed by an opposite peak in a gravity-present axis.
- Multiple peaks of different sign in a gravity-present axis
- A peak followed by an opposite peak in a non gravity-present axis.
- Change of presence of gravity or change in gravity sign.
Hence, detection of peaks and changes in the gravity properties must be monitored. Therefore, the start of a movement is defined with the detection of acceleration exceeding the threshold or a change in the gravity variables. These can be translated into a detection of a true value in the fuzzy variables (th change), (g change) and (g sign change). Once a true value is detected in (g change) or (g sign change) a rotation movement can be stated. On the other hand, when a peak is detected, more information about the following state of the signal is needed.

Hence, three temporal points are defined:

- Initial time
- Middle time
- End time

In Figure 4.20, an example of the desired temporal points in the three kinds of acceleration patterns (movement in axis with no gravity, effect in a perpendicular axis with gravity and movement in an axis with gravity) is shown.

Using the data contained in the fuzzy variables at the different temporal points (initial, middle and end time), the movement in each axis is analyzed and classified by rule evaluation in the next types:

- No movement
- Rotation
- Movement in the axis
- Movement in a perpendicular axis

Once the movement is analyzed for every axis, the information of all the axis is fused in a gesture. Only if the information of all the axes is consistent, none of the movements contradicts other movement, a gesture is detected.
The gesture information is obtained in local coordinates. To present the information in world coordinates, the orientation information obtained from the gravity orientation is used. Hence, the gesture direction is rotated.

When using both hands, information from the two hands can be combined. For the example that will be described in the next chapter, the combination of both hands is defined by comparing if the movement of the hands is equal or opposite. Hence, if the movement of both hands is in the same direction, a movement is transferred to the interacted object. Otherwise, if the movement of the hands is in opposite directions, an enlargement or a reduction is transferred to the object.

In Figure 4.21, a diagram of the hand gesture recognition process that sums up the presented ideas is shown.

The analysis of the acceleration signal and the first attempts of the recognition system were implemented\(^2\) in Octave. To integrate the recognition system in the AR platform, the algorithm was implemented\(^3\) in Scheme using the division into classes showed in Figure 4.22.

### 4.3.4 Testing and redesign

While testing the system, a malfunction was detected in the MUSH device. When the maximum value of acceleration is exceeded, the game controller ad-

\(^2\)Implemented code available at: http://svn.v2.nl/andres/octave/hand_gesture
\(^3\)Implemented code available at: http://svn.v2.nl/andres/vge
Figure 4.21: Hand gesture recognition diagram
justs the scale of the acceleration the device is able to measure. In this way, a higher value of acceleration - which the accelerometer is unable to measure - corresponds to the limit value of the output of the transfer function. Hence, the transfer function is modified by expanding the acceleration input. This expansion also affects to the death zone, commented in Section 3.1.3.1, increasing its detrimental effect.

When this effect occurs, only movements with very high acceleration can be detected, becoming the interaction very stressful for the user.

Since the modification of the game controller behaviour involves changes in the hardware of the device, the device is replaced by the Wii Remote, presented in Section 3.1.3.2, to overcome the problem.

Only changes in receiving the data from the device and in constants values for gravity and peaks detection were needed. This allows to state that the hand gesture recognition process is not dependant in the device which is used to acquire the acceleration data.

4.4 Object interaction

The last algorithms deal with the selection of the object that the player is pointing and the assignment of an action to the object based on the recognized gesture of the player.
4.4.1 Object selection

In order to select the object that the player is pointing by looking at this object, to rules are defined:

- The object must be visible for the player
- The pointed object is the closest object

Position of the player, orientation of the player and position of the objects are all defined in the system in world coordinates. For an easy discrimination of the object visibility, the location of the objects is transformed from the world coordinate system to a local coordinates at the player. This coordinate system is defined with the origin in the player position and a rotation that matches the orientation of the player. Also spherical coordinates, shown in Figure 4.23, are used instead of Cartesian, so a conversion is needed.

From Cartesian coordinates \([x, y, z]\), spherical coordinates \([r, \theta, \varphi]\) are defined as:

\[
\begin{align*}
  r &= \sqrt{x^2 + y^2 + z^2} \\
  \theta &= \arccos \left( \frac{z}{\sqrt{x^2 + y^2 + z^2}} \right) \\
  \varphi &= \arctan \left( \frac{y}{x} \right)
\end{align*}
\]

Using spherical coordinates, the application of the discrimination rules is straightforward:

- A visible object is one that has \(\theta\) and \(\varphi\) coordinates between a defined limits.
- The closest object is the one that has a smallest \(r\) coordinate.

4.4.2 Interaction mapping

As it will become clear in the next chapter, interaction with virtual objects is desired to allow the player move and rotate them.

Hand gesture recognition process detects clear movements of the hand of the player. No movement is detected if the player is not making a gesture. Therefore, movements detected by the gesture recognition system can be directly assigned to the selected object.

On the other hand, rotation of the virtual object can be mapped to the orientation of the hand. However, normal behaviour of a person - hands are no kept in a fixed orientation - will be continuously interacting with objects. Hence, the player must be able to chose between interact or not with an object by rotate the hand. This selection is done by pressing or not a button in the Wii Remote.
**Figure 4.23**: Spherical coordinate system

<table>
<thead>
<tr>
<th>Gesture</th>
<th>Interaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single hand movement</td>
<td>Translation of the object in the direction of the hand movement</td>
</tr>
<tr>
<td>Hand rotation while pressing button</td>
<td>Rotation of the object</td>
</tr>
</tbody>
</table>

**Table 4.9**: Interaction mapping
In Table 4.9, a summary of the interaction mapping is shown.
An important issue that must be taken into account is the coordinate system where the gestures are defined. Movement and orientation of the hand are defined in the local coordinate system of the gesture device. A conversion into the world coordinate system must be done. Since orientation of the device is only related to the gravity, no absolute orientation of the hand can be obtained. Therefore, hand is assumed to be in front of the player and pointing ahead.
Chapter 5

Experiment

As a final step in the thesis, an experiment is proposed with the idea of testing the performance of the AR system and describing the procedure for analyzing the behaviour of the users of the AR system.

5.1 Experiment method

5.1.1 Design

**Hypothesis**  In an AR environment, the player performs better spatial memory if he can walk around in the environment and even better, if he can decide the location of the objects that he will have to remember.

**Experiment description**  A game is proposed where the player should find the box containing the correct picture. The player can interact with the boxes to move and rotate them. Rotation of boxes is needed in order to see the contained picture. The game is divided in a first stage where the player learns which picture is contained in each box, and a second stage where the player should find the box containing the requested picture.

**Conditions**  In the learning stage, where the player should learn where are the objects that he will have to find in the second stage, three different learning procedures are proposed:

1. The player remains static in a central space and boxes positions are shown.

2. The player has to explore the environment to learn where are the boxes.

3. The player places the boxes where he decides.
Variables Several variables will be taken into account to prove the hypothesis:

- Pictures that the player should match: different sets of pictures will be used, and pictures are chosen to be as less related as possible.
- 3D or VR experience: a previous test of the player experience will be done.
- Presence: A test after the game will be done
- Usability: A test after the game will be done
- Conditions order: All the combinations of the three conditions will be used and user will get use to the environment before the experiment.
- Participants characteristics: It will be tried to find a group of participants as heterogeneous as possible. Therefore, not only people interested in AR will participate. In order to find participants not interested in AR a reward will be offered.
- Get used to the interaction: To avoid the effect of learning how to interact with the objects from the first to the last game, an introduction stage is added where the player gets used to the interaction.
- To motivate the player to try to collect the boxes by remembering them and to do not check the boxes indiscriminately until finding the correct, a score is given at the end of the game. This score is reduced as time, number of looked boxes and wrong boxes collected increase.

5.1.2 AR game development

The described AR game is implemented using the V2 Game Engine (VGE), a working title of the real time graphics environment that is being developed at V2_Lab. A game engine is the core component of a video game and provides all the underlying technologies - like rendering, sound, animation, scene management... - that simplifies the development of the game by reusing the game engine for multiple projects. For the 3-d modelling, the free application Blender can be used and an exporting tool has been developed to convert the 3-d models to the game engine.

The memory game is implemented following an object oriented programming in the language Scheme. The next classes are defined:

- Memory-box: An object of this class is created for each of the boxes containing the pictures. In the object, all the attributes of the box are contained. Functions for move, rotate and change the visibility of the boxes are defined in this class.

1 Implemented code available at: http://svn.v2.nl/andres/vge
2 http://trac.v2.nl/wiki/vge
Memory-interaction: This class contains all needed for controlling the camera and the interaction of the player. Furthermore, regions are defined where if the player or certain objects are placed, an action must be executed.

Memory-game: An object of this class controls the different stages of the game, modifying attributes of objects of the previous classes.

In Figure 5.1, a diagram of the different classes is shown.

Three more classes are added to the implementation to cover all the experiment requirements, out of the game implementation:

- Memory-log: All desired values from sensed data to game states are logged in unique files per player and game.

- Memory-analysis: Sensed data of the participants is analyzed to obtain measurements of the player behaviour, as it will be explained later.

- Memory-control: The internal values of the game, an overview of the boxes location and the viewpoint of the player are displayed in an external screen to control the game performance.

5.1.3 Measures

Subjective measures The participants are requested to fill different questionnaires during the experiment.

Before the game is explained to the participant, the player must fill a questionnaire about personal information (age, gender and educational level) and computer, 3D games and VR experience.

After each game - each player completes the game in the three conditions - the participant is requested to fill the following questionnaires:

- Igroup Presence Questionnaire (IPQ): The questionnaire developed by Schubert et al. [18] to measure the subject’s sense of presence.
• Usability Questionnaire: A questionnaire aimed at measuring the subject’s subjective evaluation of the interaction technique. The questionnaire is based on the presented by Schuemie [19].

Once the player has complete the three games, he is requested to fill a subjective questionnaire evaluating the difficulty of the three conditions and to give a general opinion of the experiment.

All the questionnaires are shown in Appendix B.2.

**Objective measures** Participants behaviour is being monitored. An analysis of the behaviour is carried out based on the following points:

• Attempts:
  - Looked objects before finding the correct
  - Attempts of collecting objects with wrong pictures.

• Movement:
  - Straightness of movement before interacting with a box
  - For each interacted box and for the correct box

• Time:
  - Between interactions
  - From a new picture is requested until it is brought to the collecting spot
  - Game length

• Orientation
  - Time percent looking at the object from previous interaction until interacting with it.

• Quality of boxes location distribution at third condition

5.1.4 Participants

To find a wide diversity of participants, an advertisement was placed on the campus of Delft University of Technology and e-mailed to several mailing lists. Due to not many answers were received and the short limitation of time while the experiment could be carried out, participants were selected from colleagues at V2 and Delft University of Technology.

Instructions given to the participant to perform the experiment are given in Appendix B.1.

In Figure 5.2, participants while doing the experiment are shown.
Figure 5.2: Participants during the experiment
Table 5.1: Descriptive statistics of the participants

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th>Max</th>
<th>Mean</th>
<th>SD</th>
<th>Max score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>24</td>
<td>35</td>
<td>28.29</td>
<td>4.39</td>
<td></td>
</tr>
<tr>
<td>Computer experience</td>
<td>2.14</td>
<td>4.86</td>
<td>3.41</td>
<td>1.11</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure 5.3: Statistics values of the participants

5.2 Results

The work presented by Schuemie [19] is considering at analyzing statistically the collected data.

Participants

Because of the reasons that were explained before, only 8 participants were recruited. One participant could not finish the first game and is removed from the experiment data. Two participants could not finish the last game because a malfunction of the system and because of sickness, so some data is missing.

The statistics of the participants are shown in Table 5.1 and in Figure 5.3.

Subjective measures

To validate the results of the questionnaires, a reliability analysis must be done. Cronbach’s alpha is used. Therefore, to consider a questionnaire reliable, the alpha obtained must be greater than 0.80.

As it can be seen in the results shown in Table 5.2, presence and usability questionnaires are reliable.

Data from the presence and usability questionnaires are analyzed in base of the game order and the different conditions, shown in Figure 5.4. To check the influence of the game order and the conditions in the feeling of presence and the evaluation of usability, the T test is performed using PSPP, a free software for statistical analysis. Since the T-test can only compare two conditions, it cannot
be assured that in some condition the performance of certain characteristic is
the best or the worst. Hence, it can be used as a clue at finding a conclusion.
An ANOVA analysis or a post-hoc test like the Tukey’s HSD test would be a
better approach, but it couldn’t be achieved to perform.

One of the subjects was removed from the usability data for having given very
high values, which might happen because an intention to please the examiner.
Results are shown in Table 5.3.

Based on the results previously presented, the following observations can be
stated:

- Feeling of presence does not differ for the different conditions or the order
  they were performed.

- Usability has a significant increase while using the AR system.

- Usability differs for the different conditions. Participants evaluated usability
er when they found the game easier, confusing ease at using the system with ease at performing the game. Therefore, it also can be
extracted that participants found the second condition as the easiest.

From the questionnaire evaluating the difficulty of the game in the three
conditions, the following conclusions can be extracted:

- All the players considered they performed better the easiest game and
  worse the most difficult.

- First and third conditions were considered the most difficult, while second
  condition was the easiest. As it is shown in Figure 5.5.

- Interaction with the objects and the method of remembering the boxes
  were the most answered reasons at explaining the game difficulty.
Figure 5.4: Analysis of questionnaires data

Figure 5.5: Evaluation by participants of conditions difficulty
Figure 5.6: Looked boxes by the participants in the different conditions

Objective measures

- Attempts: Cases where the participant collected a wrong box where infrequently, so this measure is discarded. On the other hand, interesting results, shown in Figure 5.6, can be observed at the number of boxes the player looked before collecting the correct one at the different conditions. Despite performing a T test, the significance was close to confirm a difference between condition 1 and condition 2 - t(5)=−1.759, p=0.139 - and between condition 2 and condition 3 - t(5)=−1.410, p=0.218 - it can be seen that in condition 2 participants looked less boxes.

- Movement: Movement of the player is monitored and can be displayed in plots as the ones shown in Figure 5.7. In that figure the path the player covered before the first interaction and before collecting the correct box is shown.
  The study of the characteristics of the players movement is focused on the time interval before the first interaction, because it is believed that it is in this interval when only the spatial memory affects to the movement behaviour.
  No relationship is found between the speed and the straightness of the movement in that interval, and the different conditions.

- Time: In Figure 5.8 the total time the participants needed to complete the game by conditions is shown. It can be observed that participants
Figure 5.7: Example of player movement
needed less time to complete the game in condition 2, despite the results of the T test: t(4)=-1.159, p=0.311 for conditions 1 and 2; t(4)=-1.072, p=0.344 for conditions 2 and 3.

The time expended by the participants from the request of a new picture until the interaction of a box is shown in Figure 5.9. It is not clear that participants expended less time in conditions 2 and 3 comparing to condition 1. Also the T test didn’t support this observation.

- Orientation: In Figure 5.10, the time percent the participants were looking at the box they interacted with, is shown. It can be observed that participants kept orientation more aimed at the box the interacted with in condition 2, what can indicate that participants remember better the box location. The T test was close to support the observation with: t(24)=−1.501, p=0.132 for conditions 1 and 2; t(24)=−1.095, p=0.285 for conditions 2 and 3.

- Quality of boxes location distribution at third condition: A value of the distribution quality was calculated by the variance of the distribution of the boxes in horizontal degrees from the viewpoint of the player. No correlation was found between this value and the performance at the third condition. Better estimator of the distribution quality, including the distribution in vertical degrees, could find a correlation between how the participants placed the boxes and their performance at collecting them.
Figure 5.9: Time expended by participants before interacting with a box

Figure 5.10: Orientation of participants before interacting with a box
5.3 Conclusions

As it was initially stated, the purpose of the experiment was to test the performance of the system and to describe the procedure at carrying out an experiment about the behaviour of a user.

However, some valid results can be extracted from the analysis of the obtained data at the experiment.

Despite the limited number of participants, the questionnaires about presence and usability, which were proved to be reliable, show high values for all the subjects. Hence, the feeling of presence in the AR environment was acceptable - a global mean of 4.53 in a scale from 1 to 7 - and the usability of the system was good - a global mean of 5.49 in a scale from 1 to 7.

Based on these results and that there was no malfunction of the system that couldn’t be solved, it can be concluded that the performance of the system was correct, in terms of functionality, usability and immersion.

Centering in validating the experiment hypothesis, all the results showed that participants performed the game better at the second condition comparing to the performance at the first condition. This is supported by the data of attempts, time, orientation and questionnaires. Results for the third condition weren’t expected. The idea that bad performance at the third condition is due to a careless boxes distribution is still unconfirmed.

Hence, despite the results confirmed the first part of the hypothesis - In an AR environment, the player performs better spatial memory if he can walk around in the environment - hypothesis cannot be validated due to the limited number of subjects.

Focusing in the procedure of the experiment in an AR environment, a pilot experiment was described as an example of an empirical study. Similar steps than performing a Human Computer Interaction experiment must be followed, but specific considerations must be taken into account.

Several steps to follow at the different parts of the experiment, detailing the regardful parts for an AR environment, are followed:

- Design
  - Describe clearly the experiment, including an hypothesis and the different conditions.
  - Consider in advance the multiple variables that can affect to the performance of the experiment.

Since an Augmented Reality environment is composed by real and virtual content, an analysis of the feeling of presence in this mixed environment is required. Unlike VR, limited studies can be found about presence in AR. Thus, methods applied to measure presence in VR are used in AR. When applying this methods, the differences between VR and AR must be taken into consideration. For example, using presence questionnaires, terms like real and virtual world, that are widely used, can be very confusing in AR.
• Participants must be a group as heterogeneous as possible.
One of the main factors that can influence in a subject performance at an AR experiment is the experience of the subject in any kind of related interface or technology, as it can be computers, videogames and VR systems.

• Measures
  – Make sure of logging all interesting data that can be recorded at the experiment.

Several sensors are needed for the system operation. In AR, at least player position and orientation are monitored and can be recorded. This information by itself can be valueless, due to it is not related to the dynamics of the virtual environment.
  – Design in advance the analysis of the experiment data. Logged data may depend on the analysis of the information and the analysis process can be done during the experiment, reducing the amount of logged data by recording only high level information.

In an AR experiment data is generated by the user behaviour. This behaviour is composed by specific events, like interacting with a virtual object, and continuous actions, like looking and moving. These continuous actions generate a wide flow of data, which if the design of the analysis is previously done, can be processed during its generation, reducing the amount of data and not losing related information that can be needed.

• AR game development
  – Develop a control application to allow an observer to monitor the performance of the experiment.

Though every aspect of the subject experience in an AR environment cannot be perceived by an observer, the observer must be able to monitor subject viewpoint and virtual content variation, and depending on the application, to modify parameters of the virtual environment. Communication throughput must be taken into account at transferring for example video data of the viewpoint of the player.

• Document the design, the procedure and the experiment performance in different kinds of media, not only for an analysis purpose but for description.

When trying to describe an AR environment, no information can be more clear than the view of the player immersed in it.

• Test the experiment with subjects before performing the experiment. Events and conditions may occur that were not predicted at the design.
Since in an AR environment a player has freedom of movement and actuation, multiple situations where the player places, poses or interacts in an unexpected way occur. Therefore, tests with subjects must be performed that may entail modifications or even a complete redesign of the experiments, repeating cyclically the procedure.
Conclusions

In this chapter the overall conclusions are presented. First, the research questions formulated in the introduction are answered:

- How can multisensor data fusion be applied to design an Augmented Reality platform?

Any Augmented Reality platform needs to monitor the user. This monitoring can be "narrow" - tracking only for example position and orientation of the player - or "wide" - tracking position and orientation of multiple body parts, speech, pressure, heart rate - and it can be redundant or not. In any case, sensors are the devices needed for this task. When the information provided by multiple types of sensors is desired to be combined, for obtaining a higher level information that makes use of different data or an information with better quality, Multisensor data fusion should be applied.

To design an specific AR platform, once the objectives and requirements are defined, the multiple ideas that are present in the different multisensor fusion models and techniques can be combined to accomplish the defined objectives.

In this project, two fusion models were considered at designing the fusion architecture and several techniques were applied to implement the different fusion processes defined in the fusion architecture.

- How can accurate positioning data be provided to an AR platform?

Positioning data can be relative, being generated at high frequency but with low frequency errors (errors also accumulate with time), or absolute, being generated at low frequency with low error average but with high frequency errors.

The best approach at providing positioning data to an AR platform is to combine both types of location data to obtain a position at high frequency with low errors.

Multiple technologies can be used to obtain both kinds of positioning data. In this project, inertial and ultrasound, relative and absolute respectively systems were selected.
To fuse both types of sensor data Kalman filter is the most used technique. Therefore, an indirect Kalman filter was implemented, describing the steps at modelling the system.

When modelling the system, error in the estimated position must be defined as the combination of all the error sources of the system. In this project, only errors generated by the accelerometers were considered. As a future work, a new model including gyroscopes and magnetometers errors of the Inertial Navigation System, substituting the actual filter implemented at the device libraries, is proposed. Since no information is provided about the characteristics of the devices that conform the INS, this improvement can be a very hard task.

- How can hand gesture interaction be added to an AR platform?

The starting point at providing hand gesture interaction is to define the desired gestures that will be recognized. In this project, single movements of the hand were selected as the desired gestures to be recognized. Hence, a hand gesture recognition system must be firstly implemented. In this way, the approach followed in this project consists on obtain data of the hand gesture, analyze the obtained information and implement a recognition system to detect the desired gestures.

Thus, acceleration was selected to describe the hand gesture. The acceleration data obtained from the accelerometers placed in the hand held device was analyzed to find the characteristics of the signal that could make the desired gestures recognizable. Acceleration data is is relative, noisy and events of interest only generates a few samples that contain information. Therefore, the gesture recognition system was implemented based on fuzzy logic.

Using two different hand gesture devices that provide the acceleration data (while no malfunction occurred), the gesture recognition systems detected the desired interactions. Therefore, it can be entailed that the implemented hand gesture recognition system can be used with any device that provides the acceleration data of the hand movement.

Once the gesture data is available, it must be mapped with an interaction with the AR environment.

- Perform a pilot experiment to look for procedures for carrying out scientific experiments in an AR environment?

A pilot experiment was successfully performed. As a test of the system, the experiment results showed that the performance of the designed AR system is correct in terms of functionality, usability and presence.

Despite the limited number of subjects at the experiment, results supported the hypothesis that spatial memory is better performed if the player can walk around and explore the environment than if the player can only observe the environment from a static position.

In addition, several guidelines were given at performing an experiment in an AR environment.
In conclusion, Augmented reality is a recent field and multiple aspects are still unstudied. Compared to VR, which is still a field where a lot of study can be done, limited approaches can be found of presence, interaction or human behaviour in AR. In many cases, procedures designed for VR are the only available solution, entailing different problems.

Having answered the research questions, a final remark is given. Multisensor data fusion can be applied in many applications. Few scenarios can be found that bring as many possibilities of using different types of sensors as Augmented Reality does. Augmented reality is still an experimental field where research remains. Limited procedures and no standards can be found at the implementation of an AR platform or at performing experiments with it. At researching all these unclear aspects of Augmented Reality, Multisensor data fusion is a good approach to take into consideration.
Bibliography


Appendix A

Code

The complete implemented code at this project is available at http://svn.v2.nl/andres

A.1 INS and ultrasound Kalman Filter

The main cycle of the Kalman Filter is implemented in Scheme as followed:

```
(define (kalman-filter)
  ;; Prediction for state
  (set! x (matrix-mul A x))
  (set! P (matrix-add (matrix-mul
                      (matrix-mul A P)
                      (matrix-transpose A)) Q))
  ;; Compute Kalman Gain factor
  (let ((PHt (matrix-mul P (matrix-transpose H)))
        (HPHt (matrix-mul (matrix-mul H P)
                         (matrix-transpose H))))
    (set! K (matrix-mul PHt (matrix-inverse
                            (matrix-add HPHt R))))))

  ;; correction based on observation
  (let ((zHx (matrix-sub z (matrix-mul H x)))
        (KHP (matrix-mul K (matrix-mul H P)))
        (set! x (matrix-add x (matrix-mul K zHx)))
        (set! P (matrix-sub P KHP)))
```

85
Appendix B

Experiment instructions and questionnaires

B.1 Experiment instruction

The game

The aim of the game is to find the box that contains the correct picture. Hence, a picture will be shown in a central spot and you should bring the box containing the showed picture. Each box contains a picture and the picture cannot be seen unless you interact with the box.

Interaction description

In order to interact with a box, you should be close to it and looking at it directly. Meeting these conditions, two types of interaction are allowed:

Movement: You can move a box by moving the remote control in the direction desired. The gesture should reproduce the desired movement of the box instead of trying to hit the box.

Rotation: You can rotate the box by keeping pressed the button situated in the bottom of the remote control and rotating the remote control. Once you stop pressing the button, the box will return to the initial orientation.

At any time, you must remain in the area where the boxes are placed. If you situate out of this area, the location system won’t work and you will see that boxes move away. If this case, you must return to the center. If a box is placed at the edge of the area a good strategic to move it to the center is to try to bounce it at the limit of the area.

The game is divided in three stages: Introduction, Boxes Learning and Collecting.
Introduction

In this stage you should get used to the VR helmet and learn how to interact with the boxes. Therefore, a box is placed in the space to interact with, and after certain time a requested picture will appear in the central spot. Several virtual objects will appear during the game:

- **Boxes**: Characterized by being a green and yellow cube with an open face, through which the contained picture can be seen. Pictures: Characterized by being a cube with the picture covering all the faces.

- **Message cubes**: Placed in the central spot they indicate the different stages of the game.

To finish the introduction stage you must bring the box to the central spot.

Boxes Learning

**Condition 1**

In this stage you should learn where the pictures are located.

Though all this stage, you must remain close to the central spot.

A picture will appear in the central spot. You can take as much time as you need to observe the picture. Once you want to know the position of this picture, you have to move the picture and it will fly to the exact position. You must pay attention to the position of the box and you mustn’t move from the central spot.

Once the box arrives to its position, the picture will transform into a box containing the picture. The next pictures will appear consequently.

**Condition 2**

In this stage you should learn where the pictures are located.

The boxes are placed in their initial position. You should walk around in the space and rotate the boxes to be able to see the picture which is contained in each box. It is only possible to rotate the boxes, not to move them.

You must pay attention to the position of the pictures, since you will have to remember it in the next stage.

After having rotated all the boxes and being sure that all the location of the pictures can be remembered, you can situate yourself in the central spot to proceed to the next level.

**Condition 3**

In this stage you should learn where the pictures are located.

A picture will appear in the central spot. You can move it and place it where you decide. You should place it in a location that you will be able to remember and not very close to the central spot. You will know if the picture
is far enough from the central spot, because a new picture will appear in the central spot, making a characteristic sound.

Once the new picture is moved, the previous one will transform into a box containing the picture. Then, you won’t be allowed to move or rotate the previous box.

Once you have placed all the pictures, keeping in mind that you will have to remember their positions, you can situate in the central spot to continue to the next stage.

Collecting

In this stage the pictures will be requested by appearing in the central spot. A requested picture will be recognized by rotating.

When a picture appears in the central spot you should bring the box containing the picture by moving it to the central spot. This must be done:

- Using as less time as possible
- Interacting with less boxes as possible

Once the correct box is brought to the central spot and no other box is close to the central spot, the next picture will appear. The process will be repeated until all the boxes are collected.

A final score will be obtained based on:

- Required time to complete the collecting
- Number of boxes brought to the central spot before the correct one
- Number of boxes rotated for each requested picture

Less point punishment will be applied for rotating boxes than for bringing an incorrect one to the central spot.

If you bring an incorrect box to the central spot, the requesting picture will stop rotating and it will make an error sound.

For a perfect score, you should try to bring the correct box without looking at which picture is contained in each box, that is, by remembering where is the picture.
B.2 Experiment questionnaires

Questionnaire 1: Personal information and computer/VR experience

<table>
<thead>
<tr>
<th>Questionnaire 1</th>
<th>Player ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any information you will provide during the experiment will be treated confidentially and will not be linked to your name but to a number. Your movements and actions during the game will be recorded for the purpose of the the experiment.</td>
<td></td>
</tr>
<tr>
<td>Year of birth</td>
<td></td>
</tr>
<tr>
<td>Gender</td>
<td></td>
</tr>
<tr>
<td>Educational level</td>
<td></td>
</tr>
<tr>
<td>How do you rate your overall computer skills?</td>
<td>Very bad</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>How often do you use a computer?</td>
<td>Never</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>How often do you play 3D games?</td>
<td>Never</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>How often do you use 3D programs (excluding games)?</td>
<td>Never</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>Have you ever used a VR-helmet before?</td>
<td>Never</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>Have you ever used CAVE (system with glasses and projection onto walls)?</td>
<td>Never</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
<tr>
<td>Have you ever used the Wii remote control?</td>
<td>Never</td>
</tr>
<tr>
<td></td>
<td>1 2 3 4 5 6 7</td>
</tr>
</tbody>
</table>
Questionnaire 2: Presence

Please, fill the following questionnaire based on your experience during the game. The term virtual world or computer generated world refers to the merge of the real world and the virtual content. Real environment refers to the real world out of the game space.

In the computer generated world I had a sense of "being there" Not at all Very much
1 2 3 4 5 6 7

Somehow I felt that the virtual world surrounded me. Fully disagree Fully agree
1 2 3 4 5 6 7

I felt like I was just perceiving pictures. Fully disagree Fully agree
1 2 3 4 5 6 7

I did not feel present in the virtual space. Did not feel Felt present
1 2 3 4 5 6 7

I had a sense of acting in the virtual space, rather than operating something from outside. Fully disagree Fully agree
1 2 3 4 5 6 7

I felt present in the virtual space. Fully disagree Fully agree
1 2 3 4 5 6 7

How aware were you of the real world surrounding while navigating in the virtual world? (i.e. sounds, room temperature, other people, etc.)? Extremely aware Moderately aware Not aware at all
1 2 3 4 5 6 7

I was not aware of my real environment. Fully disagree Fully agree
1 2 3 4 5 6 7

I still paid attention to the real environment. Fully disagree Fully agree
1 2 3 4 5 6 7

I was completely captivated by the virtual world. Fully disagree Fully agree
1 2 3 4 5 6 7
<table>
<thead>
<tr>
<th>Question</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>How real did the virtual world seem to you?</td>
<td>Completely real, Not real at all</td>
</tr>
<tr>
<td>How much did your experience in the virtual environment seem consistent with your real world experience?</td>
<td>Not consistent, Moderately consistent, Very consistent</td>
</tr>
<tr>
<td>How real did the virtual world seem to you?</td>
<td>About as real as an imagined world, Indistinguishable from the real world</td>
</tr>
<tr>
<td>The virtual world seemed more realistic than the real world.</td>
<td>Fully disagree, Fully agree</td>
</tr>
</tbody>
</table>
Questionnaire 3: Usability

Please, fill the following questionnaire based on your experience during the game.

I have quickly learned how to interact with the virtual content. Completely disagree 1 2 3 4 5 6 7

I could do all things that I wanted to do. Completely disagree 1 2 3 4 5 6 7

It was immediately clear what I could do and what I couldn’t do with the virtual content. Completely disagree 1 2 3 4 5 6 7

I sometimes lost my orientation. Completely disagree 1 2 3 4 5 6 7

It was easy for me to look around with the VR helmet. Completely disagree 1 2 3 4 5 6 7

It took some time before I completely understood how to interact with the virtual content. Completely disagree 1 2 3 4 5 6 7

I found the system easy to use. Completely disagree 1 2 3 4 5 6 7

It was easy for me to move the boxes in the direction I wanted. Completely disagree 1 2 3 4 5 6 7

It was easy to rotate the boxes to see the contained picture. Completely disagree 1 2 3 4 5 6 7

Did you (subjectively) like playing with the system? Completely disagree 1 2 3 4 5 6 7
Questionnaire 4: Conditions difficulty

Please, fill the following questionnaire based on your experience during the three games.

- Please, sort the different games by how well you performed them.
  - First game
  - Second game
  - Third game

- What reason can explain why you were better in the game chosen in the previous question:
  - I was more used to the interaction
  - It was easier to remember pictures’ location
  - I was less tired
  - Other:

- What reason can explain why you were worse in the game chosen in the first question:
  - I was less immersed
  - I wasn’t used to interact with the boxes
  - It was very difficult to remember the pictures’ location
  - I was tired or bored
  - Other:

- Please, rate the different games from difficult to easy at remembering the location of pictures
  - First game
  - Second game
  - Third game