Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2018

DÉRCIO DA CONCEIÇÃO MASSANGO BUCUANE

Integração de Localização Baseada em Movimento na Aplicação Móvel *EduPARK*

Integration of Motion-Based Location in the Mobile App *EduPARK*

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Professor Dr.-Ing. Joaquim João Estrela Ribeiro Silvestre Madeira, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Professor Doutor Paulo Miguel de Jesus Dias, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho à minha mãe Filomena da Graça Massango.

o júri

presidente

Professor Doutor Joaquim Arnaldo Carvalho Martins
Professor Catedrático, Universidade de Aveiro

vogais

Professor Doutor Telmo Eduardo Miranda Castelão da Silva
Professor Auxiliar, Universidade de Aveiro (Arguente Principal)

Professor Dr.-Ing. Joaquim João Estrela Ribeiro Silvestre Madeira
Professor Auxiliar, Universidade de Aveiro (Orientador)

agradecimentos

Agradecer às pessoas que tornaram possivel a realização deste trabalho é um dever moral, mas acima de tudo é meu maior desejo, por isso, aproveito este espaço para expressar o meu profundo agradecimento ao Professor Dr.-Ing. Joaquim Madeira pela disponibilidade, orientação, e partilha de conhecimentos durante a realização do trabalho. Agradeço também ao Professor Doutor Paulo Dias pelos conselhos e co-orientação do trabalho.

Especial agradecimento vai também para a Professora Doutora Lúcia Pombo e para a Doutora Margarida Marques, coordenadoras do projeto EduPARK, por me terem recebido no projeto, partilhado conhecimento e informações importantes, bem como pela atenção a mim dispensada durante a realização do trabalho.

Agradeço imenso à Novabase por ter financiado a minha formação, bem como pelo suporte contínuo que me foi dado.

Por último, mas não menos, agradeço à minha familia, aos colegas, e a todos que directa ou indirectamente contribuiram para a realização deste trabalho. A todos o meu MUITO OBRIGADO.

EduPARK é um projeto financiado por Fundos FEDER através do Programa Operacional Competitividade e Internacionalização - COMPETE 2020 e por Fundos Nacionais através da FCT - Fundação para a Ciência e a Tecnologia no âmbito do projeto POCI-01-0145-FEDER-016542.

Palavras-chave Localização de pedestres, Realidade Aumentada, Aprendizagem Móvel

Resumo

Cada vez mais, as aplicações móveis requerem soluções de localização precisa nos mais variados ambientes. Apesar de o GPS ser amplamente usado como solução para localização, pode apresentar alguns problemas de precisão em condições especiais, como mau tempo, ou espaços com várias obstruções, como parques públicos. Para estes casos, soluções alternativas ao GPS são de extrema relevância e veem sendo desenvolvidas. A presente dissertação estuda o caso do projeto EduPARK, que é uma aplicação móvel de realidade aumentada para o parque *Infante D. Pedro* em Aveiro. Devido à fraca precisão do GPS nesse parque, a implementação de funcionalidades baseadas no posionamento e de realidade aumentada sem marcadores apresenta dificuldades. São analisados sistemas relevantes existentes e é proposta uma arquitetura baseada em localização de pedestres. Em seguida é apresentada a correspondente implementação, que consiste numa solução de posicionamento usando os sensores disponiveis nos smartphones, um algoritmo de deteção de passos, um estimador de distância percorrida, um estimador de orientação e um estimador de posicionamento. Para a validação desta solução, foram implementadas funcionalidades na aplicação EduPARK para fins de teste, e realizados testes com utilizadores e testes de usabilidade. Os resultados obtidos demostram que a solução proposta pode ser uma alternativa para a localização no interior do parque *Infante D. Pedro*, viabilizando desta forma a implementação de funcionalidades baseadas no posicionamento e de realidade aumenta sem marcadores.

Keywords Pedestrian Dead Reckoning, Augmented Reality, Mobile Learning

Abstract More and more, mobile applications require precise localization solutions in a variety of environments. Although GPS is widely used as localization solution, it may present some accuracy problems in special conditions such as unfavorable weather or spaces with multiple obstructions such as public parks. For these scenarios, alternative solutions to GPS are of extreme relevance and are widely studied recently. This dissertation studies the case of *EduPARK* application, which is an augmented reality application that is implemented in the *Infante D. Pedro* park in Aveiro. Due to the poor accuracy of GPS in this park, the implementation of positioning and marker-less augmented reality functionalities presents difficulties. Existing relevant systems are analyzed, and an architecture based on pedestrian dead reckoning is proposed. The corresponding implementation is presented, which consists of a positioning solution using the sensors available in the smartphones, a step detection algorithm, a distance traveled estimator, an orientation estimator and a position estimator. For the validation of this solution, functionalities were implemented in the EduPARK application for testing purposes and usability tests performed. The results obtained show that the proposed solution can be an alternative to provide accurate positioning within the *Infante D. Pedro* park, thus enabling the implementation of functionalities of geocaching and marker-less augmented reality.

Contents

List of Figures

List of Tables

List of Acronyms

AR	Augmented reality
AoA	Angle of Arrival
CDF	Cumulative Distribution Function
GPS	Global Positioning System
HMDs	Head Mounted Devices
HVM	Human Vision Systems
IMUs	Inertial Measurement Units
OCR	Optical Character Recognition
POIs	Points of Interests
PDR	Pedestrian Dead Reckoning
QR	Quick Response
RSS	Received Signal Strength
SDK	Software Development Kit
SAR	Spatial Augmented Reality
TDoA	Time Difference of Arrival
ToA	Time of Arrival
UML	Unified Modeling Language

1. Introduction

1.1 Motivation

Satellite-based technologies such as GPS are still the predominant technologies used by mobile applications to provide localization services. However, they do not perform well in certain environments such as crowded cities, indoor settings, or even in unfavorable weather. The case under study in the present dissertation is the *EduPARK* project game-like application, an augmented reality application implemented in the *Infante D. Pedro* urban park in Aveiro. Due to poor GPS precision and accuracy within the urban park, the implementation of marker-less augmented reality and other functionalities of geocaching were inviable for the project. This justifies the quest for an alternative localization solution to provide accurate positioning within the urban park and, consequently, enable the implementation of geocaching functionalities and marker-less augmented reality. The present work proposes an alternative system to provide positioning with increased precision using only the smartphone built-in sensors.

1.2 Objectives

The main objective of this dissertation was the research and implementation of a solution that could provide greater precision in the positioning within the *Infante D. Pedro* urban park and be an effective alternative to the use of GPS. Considering this goal, the first specific objective was the study of related systems and recent technologies in order to substantiate the proposed system. The second specific objective was the study of the current state of the *EduPARK* application, and the implementation and integration of the proposed positioning system. Another specific objective of the present work was the creation of 3D models of the monuments in the urban park to be integrated in the EduPARK application. The realization of tests and usability studies to evaluate the proposed solutions were also required.

1.3 Structure

This dissertation is divided into six main chapters. The first chapter introduces the motivation and objectives behind the work. The second chapter describes the state of the art on augmented reality, mobile augmented reality, the EduPARK application, and localization systems. The third chapter presents the analysis and design of the proposed positioning solution. The fourth chapter describes the implementation details of each component of the proposed solution. The fifth chapter presents the studies done to evaluate the solution developed in the present work, starting with the tests to the system, then the usability tests with users and a discussion of results obtained. Finally, the sixth chapter presents the conclusions and future work recommendations.

2. State of the Art

This chapter presents the state of the art of the fields under study in the present dissertation. Section 2.1 describes augmented reality, its characteristics, applications, challenges, and issues. Section 2.2 describes mobile augmented reality with special attention to its applications in education and then introduces an overview of the *EduPARK* application. Finally, section 2.3 presents a review on mobile localization systems and a comparison study of relevant localization solutions.

2.1 Augmented Reality

Augmented Reality (AR) is a technology that "allows overlying virtual objects in a real-world environment in real time, producing a new experience" [1], [2]. AR systems "supplement the real world with virtual (computer-generated) objects that appear to coexist in the same space as the real world" [3], enhancing reality with additional virtual information [4]. On the Reality-Virtuality continuum by Milgram and Kishino [5], AR is one part of the general area of mixed reality where there is a continuous scale ranging between the completely virtual, a "Virtual Reality", and the completely real "Reality", where AR is "closer to the real world and augmented virtuality (AV) is closer to a purely virtual environment" [6], as shown in **Figure *1***.

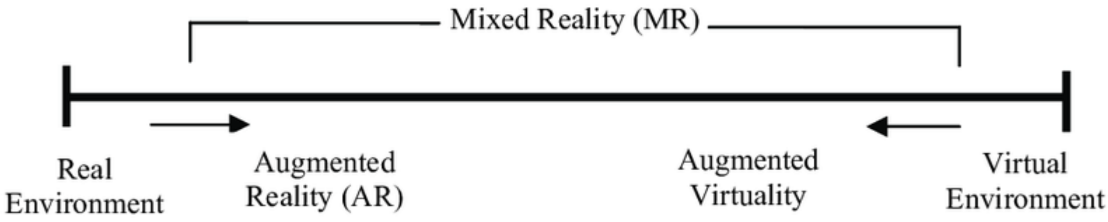

Figure 1: Milgram and Kishino´s Reality-Virtuality Continuum [5]

The previous definitions provide a helpful comprehension of the main goal of AR systems, which is to provide a new experience for the users by allowing the exploration of reality with additional virtual information, or according to Hugues et al. [6] by simplifying the user´s life by bringing virtual information to his immediate surroundings and even to any indirect view of the real-world environment, such as live-video stream. According to Azuma et al [3] and Madden [7], the main characteristics of an AR system are:

- combines real and virtual objects in a real environment;
- registers or aligns real and virtual objects with each other,
- runs interactively in 3D and in real time,
- combines real world with computer graphics,
- provides interaction with objects in real time,
- provides recognition of images or objects, and
- tracks objects in real time providing real-time context or data.

2.1.1 AR Technologies

The augmentation of a real environment can be achieved by using visual techniques and non-visual techniques [4], [6], [8].

Visual AR techniques consist of the rendering of “3D virtual objects from the same viewpoint from which the images of the real scene are being taken using tracking cameras” [6]. Visual AR is based on image registration using “different methods of computer vision mostly related to video tracking” [6]. These methods usually consists of tracking and recognition stages. The tracking stage attempts to “calculate the trajectory of an object in the image plane as it moves around a scene through features detected in a video stream” [8], this stage makes use of “feature detection, edge detection or other image processing methods to interpret the camera images” [6], through tracking, fiducial markers, optical images, or points of interest (POI) are detected. The recognition stage uses the data obtained from the tracking stage to reconstruct a real-world coordinate system [6]. Visual AR is most suitable for see-through AR systems that already have a video camera, these systems include handheld displays (mobile devices, optical see-through glasses) and head-mounted displays (video see-through glasses, holographic projector, anaglyph glasses, alternate frame sequencing, and polarization displays).

Visual AR systems can track and recognize a lot of entities by extracting features from video frames, and this process requires “software to create consistency between the elements in the image and the known 3D locations in the world” [8]. Software packages that enable feature extraction and tracking include OpenCV [9], Vuforia [10], Unity [11], and Augment [12], among others.

The main challenges in visual AR lie on the type of environment the AR device will be introduced to as well as the type of AR system [6]. In indoor or outdoor environments, places such as windows corners or wheels can be extremely difficult to match or recognize due to reflection and transparency, and objects that have irregular shape can be affected by environment conditions. To overcome these challenges recent advances reported by Hugues et al [6] propose the use of Human Vision Systems (HVM) that study how the human brain recognizes objects. If the way of recognizing things by the human brain can be modeled, computer vision and consequently visual AR will be able to overcome the challenges it is currently facing.

Non-Visual AR techniques consist of the expansion of the user experience by “providing stimulus for other senses in addition to visual augmentation” [8], these other senses can be audition, taste, touch and even smell [4].

Audio in AR has been widely used for aural augmentation or as part of the user interface [13], most applications provide solutions for visually impaired people, these solutions take advantage of optical character recognition (OCR) and computer vision techniques to detect objects and read texts [13]. A good example is LookTel [14], which is an instant recognition smartphone application for visually impaired people, it has features to read aloud things at which the user points, or even more, using GPS it can help visually impaired travelers locate orientation points on a route.

Sense, smell, and touch in AR are used in closed-space environments, where it is possible to integrate multiple sensors to enrich the user experience [13], or in specific hardware that integrate the necessary sensors to enable those senses. M5SAR [15] is a project that introduced a mobile five senses augmented reality system for museums. The system consists of a smartphone application and a gadget to be integrated with the smartphone. For touch sensation, three techniques were used: thermal touch, vibration, and air flow. The thermal touch could recreate heat and cold sensations using thermo-electric modules, the vibration was obtained with vibration motors with one on each side of the device, the air flow or wind sensation is obtained using a ventilation system with four fans, two in each side of the device. For the smell sensation, a flow of air is forced through an aromatized container, which is then inserted into the fan ventilation system of the wind feedback module, helping spread the fragrances naturally in the air. ***Figure 2*** illustrates the gadget integrated with a smartphone allowing a multiple sense AR experience to the user.

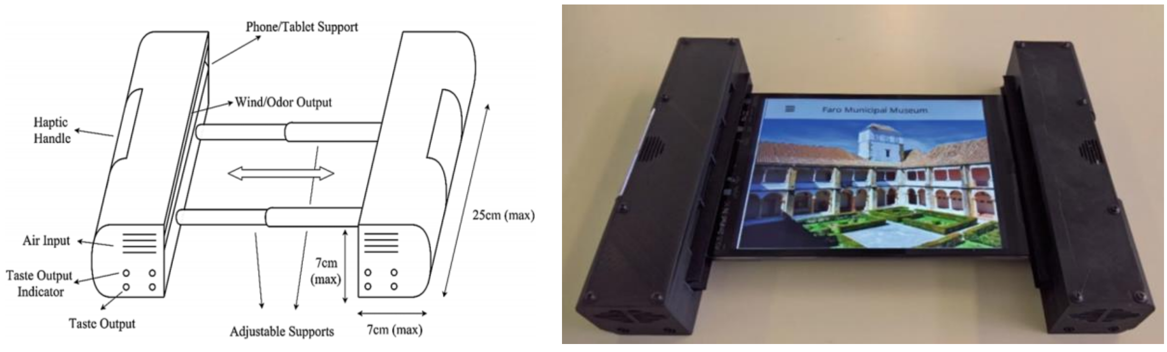

Figure 2: A portable device for the five-sense experience. Tablet or smartphone support [15].

2.1.2 AR Devices

New research has been conducted to implement AR systems for vision, audio, taste, touch and even smell, as described in 2.1.1. The hardware platforms used in the various forms of AR can be classified according to Hugues et al. [6] in displays, input devices, tracking, and computers. Displays can be of three types, head-mounted displays (HMDs), handheld displays and spatial displays. “HMD is a display device worn on the head or as part of a helmet and that places both images of the real and virtual environment over the user’s view of the world” [6] as shown in ***Figure 4***, they can either be video-see-through or optical see-through and can have a monocular or binocular display optic. Handheld displays employ small computing devices with a display that the users can hold in their hands, the most widely used handheld displays for AR are smartphones, PDAs and tablets. Spatial Augmented Reality (SAR) make use of video-projectors, optical elements, holograms, radio frequency tags, and other tracking technologies to display graphical information, an example is illustrated in ***Figure 5***. The input devices for AR are mainly composed by haptic devices [4], [6], [13], some systems utilize gloves, as shown in ***Figure 3***, other even use a wireless wristband. Tracking devices consist of digital cameras and/or other optical sensors, GPS, accelerometers, solid state compasses, wireless sensors, etc.

Figure 3: Haptic device[16]

Figure 4: HMDs[6]

Figure 5: SAR[6]

The output from those platforms comes in the form of images and sound, while the information is displayed as text, virtual objects, textures, or highlighting, ***Table 1*** characterizes the various applied categories or forms of AR systems related to its specific devices, output, and AR content provided [4], [6], [13].

Category	Description	Device	AR Information
Vision	Handheld	Mobile Devices,	Text, Virtual Objects, Highlighting, 3D text, Textures, 3D Highlighting.
	HMD	Optical see-through glasses, Video see-through glasses, Alternate frame Sequencing Displays	
	Spatial	Projector, LCD display, Autostereoscopic Display, Polarization displays	
	Wearable	Holographic projector	
Audio	Spatial	Speakers,	The direction of sound translations, Improved sound.
	HMD	Headphones	Translations, Additional sound.
	Handheld	Earphones	Improved Sound
Touch	Spatial	The haptic device, Vibrating device.	Additional motion, Haptic feedback
	Handheld	mobile device, game controller	
Smell, Taste	HMD, Handheld	Multiple sensors, Gustatory display	Fragrance, flavor

Table 1: AR systems grouped in terms of devices used and AR content [4].

2.1.3 Applications of AR

"AR has been widely used in a variety of fields for the achievement of smooth blends between the virtual and real worlds" [4]. Mekni & Lemieux [17] identified 12 well-established application domains for AR, which include the military, medical, manufacturing, entertainment, visualization, education, advertising and commercial marketing, geospatial, navigation and path planning, tourism, urban planning, and civil engineering. This section focuses on the recent advances and applications of AR in the specific domains of military, medicine, and education.

In the military, AR can be used to "display the real battlefield scene and augment it with annotation information" [17] and also for the "repairing or training of the field equipment for the soldiers" [4]. An example is the Battlefield augmented reality system (BARS) [17] developed to provide training in large-scale combat scenarios and simulating real-time enemy action. In another perspective, Canada´s Institute for Aerospace Research (NRC-IAR) developed a helicopter night vision system that uses AR to expand the operational envelope of rotorcraft and enhance pilots ability to navigate in degraded visual conditions [17]. These use cases elucidate the high relevance of AR in the planning of military interventions and training, and for preventive operations.

In the medical field AR is mostly used for the training of medical students and to help doctors during surgeries [18]. Surgical AR can allow doctors to provide guidance, help, and support with valuable information during a surgical operation, or it can support the rehearsal or discussion of the operation for which a realistic virtual version of the patient's organ is used [4]. On the other hand, since AR surgical training is both time and cost intensive, it has been applied in the form of superimposition of computer-generated virtual organs in the trainee-surgeon vision field with the use of an optical HMD such as Google Glass or the STAR 1200 XL from Vuzix [4], [19]. Another good AR based solution for the medical field is the AV400 Vein Viewing System [4], which involves the use of a handheld scanner that projects onto the skin to show the location of the patient's veins. With the use of the AV400 device, the medical practitioner is more likely to precisely find a vein during the first injection attempt. Besides the tremendous impact in the field, AR solutions have also some issues mainly related to displays and tracking. Display challenges mostly arise from the fact that the preferred type of display to use for medical applications is an HMD, as it allows the physician not only to use both hands, but it is also easier to track where the doctor is looking at to augment the right surfaces. However, it is challenging to implement HMD based solutions for medical applications [6].

In education, AR systems support user interaction, provide instant feedback, and are exciting to use, and they can potentially foster learning [20]. While the majority of existing efforts have targeted primary and high school education, college education is also another niche area of research that is under investigation [21], [22]. The complete evaluation of current applications of AR in the specific field of education is presented in section 2.2.2.

2.1.4 AR Challenges and Issues

There are a number of constraints that limit what can be done with AR applications [23], [2], [19], [24] and/or additional issues that the application developer must address to overcome those constraints.

The first limitation regards technological aspects, mainly because augmented reality systems must deal with a vast amount of information: the hardware used should be small, light, and easily portable and fast enough to display graphics [17]. But in contrast, the resources on most devices are limited, those are manifested primarily as limited memory and limited computational capability, as well as limited graphics capability, limited input and output options and especially limited screen size. Even if the system includes some type of head-based display such as glasses, they often have a limited field of view and limited resolution. Memory is a primary limitation on the amount of content that can be resident on a mobile device at any given moment. These facts bring another issue, the battery life used by such augmented reality devices, since AR features consume many resources that can decrease battery life. AR tracking needs some positioning systems such as GPS to provide accurate localization. As AR systems obtain a vast amount of information, robust software is needed to filter the information, retain useful information, discard useless data and display it in a convenient way. Tracking in unprepared environments remains a challenge but hybrid approaches are becoming the easiest way to overcome these problems: for indoor and even outdoor settings, solutions based in fingerprint localization have provided great results.

Aside from technical challenges, the user interface must also follow some guidelines such as not to overload the user with information, while also preventing the user to overly rely on the AR system such that important cues from the environment are missed [25].

2.2 *Mobile Augmented Reality*

Mobile augmented reality covers the use of smartphones or tablets to access AR content. One key advantage of mobile augmented reality, according to Craig [26], is that in “addition to being inexpensive, many people already own the necessary hardware”. Current smartphones and tablets already contain the sensors, processing, and displays necessary for mobile AR applications. Having many potential users already in possession of the required hardware is a very compelling attribute. Mobile augmented reality is especially well suited to ideas such as “ubiquitous learning” [23] in which the plan is that every person learns all the time, wherever they are, when they need to. This assumption can be related to the case studied in this dissertation, for example if someone is visiting *Infante D. Pedro* Park and wants to learn more about the history or the biodiversity of the park, he can use the mobile phone or tablet to access the application and gain additional and relevant information about the park.

2.2.1 Marker-Based *versus* Marker-Less Mobile AR

Mobile Augmented Reality can be implemented in two forms [2], [8], *Artefact-Based* or *Marker-Based* and *Geolocated or Marker Less*.

“Artefact-based AR uses physical markers or objects that are scanned by a camera and then carry out an action” [2]. A marker can be a sign or image that can be detected by a smartphone camera using image processing, pattern recognition, and computer vision techniques [8]. Markers have typically been quick response (QR) codes, as shown in ***Figure 6*** or barcodes, as shown in ***Figure 7***. However, recent technological advances have enabled the use of any kind of image defined within the AR technology, as shown in ***Figure 8***.

Figure 6: QR code

1 234567 890128 >

Figure 7: Bar codes

Figure 8: Image defined code

According to Siltanen [13], the detection of a marker by an AR system is a process that consists in finding the outlines of potential markers and deducing locations of marker´s corners in the image, which will lead to the identification of potential markers and fast rejection of obvious non-markers; then the markers are decoded using template matching or feature extraction techniques. A good marker is easily and reliably detectable under all circumstances. “Differences in luminance (brightness) are more easily detected than differences in chrominance (color) using machine vision techniques” [13]. Marker-based AR solutions are more suitable for situations such as described below, according to [4], [6], [8], [13], [27]:

- Environments that are challenging for feature tracking or geocaching: environments with large buildings or natural obstructers can difficult the accuracy of most tracking solutions employed in markerless AR, but if the user adds markers in such environments, tracking becomes possible and easier.
- Proof of Concept: Marker-based tracking might be good for a proof-of-concept type of application where the emphasis is not yet on the tracking implementation but on easily demonstrating the application concept, because marker-based systems are typically computationally cheaper to implement.
- Devices with limited computational capacity and memory: Marker-based systems need less processing power and memory compared to feature tracking. This is an important aspect in mobile augmented reality, for example with lightweight mobile devices.
- Interaction with the user: User interaction in certain types of applications is easy to implement with markers. For example, the user might move augmented objects by moving markers. Markers are tangible and, even for an inexperienced user, it is easy to understand how to move the objects.
- Markers can maintain additional information, like an ID, URL or text. This enables the system to associate data with markers and retrieve information. Marker-based AR applications have much lower costs of implementation.

Marker-Less or geolocated AR uses locational sensing, typically through Global Positioning Systems (GPS), and overlays digital information on points of interest (POIs) including physical places and map references. Users who have the appropriate equipment, typically a GPS-enabled smartphone or tablet, can view these POIs. A comparative characterization of marker-less AR and marker-based AR is presented in ***Table 2***, summarized from [8].

Marker-Based AR	**Marker-Less AR**
Uses fiducial markers; generally, fiducial marker images are black and white with a square form for easy detection.	No need for fiducial markers; uses feature tracking and positioning techniques.
Corresponding image descriptors are provided beforehand.	Does not need any pre-knowledge of a user´s environment to overlay 3D content into a scene and hold it to a fixed point in space.
Recognition library may be able to compute the pose matrix of the detected image, relative to the camera of the device.	Recognizes images that are not provided to the application beforehand.
Cheap marker detection algorithm, generally robust against lighting changes but weak if the marker is partially overlapped.	Recognition algorithm running in an application should identify patterns, colors, or other features that may exist in camera frames.

Table 2: Characterization of Marker-Based versus Marker-less AR

Marker-less solutions especially for outdoor AR tracking are mainly based on GPS to localize the camera position and inertial sensors to measure the orientation. However, GPS can face low precision and low update rate and inertial sensors can suffer from error drifting and measurement distortion [28]. Hybrid systems that are based on both GPS and inertial sensors might achieve acceptable precision and accurate tracking.

2.2.2 Mobile Augmented Reality in Education

Mobile AR applications in the specific field of education can be classified according to Antonioli et al. [29], in three categories: traditional classroom uses, outside the classroom, and special education uses.

In traditional classrooms, or indoor settings, desktop AR allows students to combine both real and computer-generated images. As is reported in the case studied in [30], where they used desktop AR that combined a screen, glasses, headphones, and a pointing device to allow students to conduct a hands-on exploration of a real object. Outside the classroom or in outdoor settings, camera phones and smartphones allow students to gather information in a variety of locations. QR codes and GPS coordinates can be used to track and guide the movement of the students [29], as presented in the case of EduPARK [1]. Special Education Uses [29] is related to the fact that AR has the potential to bring value and high-quality educational experiences to students. The study presented by Antonioli et al. reported that using augmented storybooks has led to more positive results as students were able to recall stories and had better reading comprehension. Augmented Reality is recognized as a technology that can increase student interest and motivation as well as promote self-learning [1], [18], [21], [22], [31]. Mobile augmented reality can help the understanding of more complex and abstract concepts and combined with game-based learning students may be more willing to overcome challenges and learning difficulties. In the studies [1], [18], [21], [22], [31], the main affordance of mobile augmented reality was to promote student engagement, access complementary information in different formats, such as text, sound, video or 3D models, allowing to record data and observations, e.g., with annotated photographs, answering questions and challenges related with the outdoor setting and receiving immediate formative feedback. AR allows 3D visualization of phenomena or concepts, which is not possible with traditional textbooks and the use of GPS, digital compass and gyroscope to guide students towards learning objects. The facts presented above emphasize that AR is becoming the new trend in education, gathering together outdoor and indoor learning advantages.

2.2.3 EduPARK Mobile Augmented Reality Platform

The *EduPARK* project [1] aims to contribute to the smart urban park concept by designing, implementing and evaluating the *EduPARK* game, supported by a mobile app to promote learning within the urban park *Infante D. Pedro* located in Aveiro. The final purpose of combining mobile technology with outdoor gaming strategies based on geocaching principles is to “enhance student motivation and allow learning to move beyond traditional classroom environments to natural spaces” [1]. The application is based on a question and answer game that allows users to physically explore the Park as well as the available augmented reality content: the augmented reality content is visualized through physical markers installed in the park, as shown in ***Figure 9***.

Figure 9: Physical markers for AR

The basic structure and functionalities of the application are described below and summarized from [1]:

Figure 10: Initial screens, learning guide (quiz) selection [1]

One of the initial screens of the app prompts the players to identify their team and select a learning guide, as shown in ***Figure 10***, one for First Cycle pupils (aged 9–10) and another for Third Cycle ones (aged 13–14), other learning guides are destined to tourists and undergraduate students.

Initially, the players are welcomed with a short explanation of the app's quiz-like geocaching-based game structure. A short tutorial explains how to use the camera tool to recognize the AR markers, which unlock the access to information relevant to answering questions related to that specific location. Next, the players can initiate the stages, following instructions to find a specific AR marker, using the device to recognize the prompted marker, accessing a set of multiple answer questions, and receiving adequate feedback to answers and scores, if answered correctly. The app also provides feedback through the constant display of accumulated scores and offers a sense of progress through the number of questions answered, locations visited, and caches discovered, as shown in ***Figure 11***.

Figure 11: Game Sequence [1]

To support the players progression, the app provides a number of tools: camera (to recognize AR markers and take pictures), backpack (to see the pictures taken), compass (to support the players' orientation in the park) and a map of the park (with the next location or cache to visit)[1].

Figure 12: Additional tools provided in the game-like application

According to [1], the results obtained with the implementation of the project and data collection gathered from participants (focus groups) and from observations showed that combining mobile technology with outdoor gaming activities allows learning to move beyond traditional classroom environments that pupils can explore and, simultaneously, make connections with curricular content. Furthermore, the *EduPARK* game provides collaborative, situated and authentic learning, it also offers new challenges, opens horizons and opportunities for science and education.

The main technologies used to develop the *EduPARK* game-like application are: [1]:

Unity3D: Unity is a modern cross-platform engine for creating games and applications developed by Unity technologies. The engine can be used to create games in both 2D and 3D, offers a primary scripting API in C#, as well as simulations for desktops and laptops, home consoles, smart TVs and mobile devices [11]. It is widely used as the main technology in the development of mobile learning games, such as the cases of [1], [22], [23].

Vuforia: Vuforia is an augmented reality platform and a Software Development Kit (SDK) for mobile devices developed by Qualcomm. It was used for augmented reality marker detection since it is currently the most widely adopted platform for AR technology. Vuforia Model Targets recognize objects by shape, in contrast to other existing methods that rely on detailed visual designs typically found on print media, product packaging, and many consumer goods [10];

Sketchup: Sketchup is a 3D modeling tool used to create many 3D objects to help the learning process [1].

2.3 Localization Systems

2.3.1 Fingerprint Based Outdoor Localization

Over the years, localization services applied to different contexts have been predominantly provided by satellite-based technologies such as GPS. However, since these technologies require a line of sight to a variety of satellites “they often do not perform well in crowded cities or in unfavorable weather” [32]. Another problem can be the “high power consumption, which is a serious challenge to battery-based mobile devices” [33]. To overcome the limitations of GPS based localization techniques many researchers have proposed a series of alternative solutions, including cellular-based systems [34], infrared-based systems [33], received signal strength indicator (RSSI) based systems [35], or even hybrid systems [36]. According to Du et al. [33] these methods can be classified in range-based and range-free. “Range-based methods rely on the estimated distances to achieve localization while range-free methods do not need the distance information” [33].

Range-based systems rely on relative distance generally obtained through measuring methods such as time of arrival (ToA) that “measures the distance from the unlocated devices to the anchor node through calculating the travel time of the signal” [33]; time difference of arrival (TDoA), which creates a distance indicator by “deploying the receivers at some known positions, the time of the signal arriving at each receiver is different, which can be exploited to measure the distance” [33]; propagation model, which is based in the use of received signal strength (RSS) to measure distance or basically “when a device detects available signals, it can calculate the distance between the base station and itself using the propagation model and the RSS” [33]; and angle of arrival (AoA) that can “infer the region of the anchor node through the angle at which the signal is received” [33]. The main constraint in range-based techniques is that obstacles can produce errors, because such techniques are sensitive to the surrounding environments.

Range-free methods do not rely on distance information. The most widely used range-free method is fingerprint localization [33]. “Fingerprint localization captures signatures that are matched against a set of geotagged signatures to identify a device location” [32]. It is a positioning technique that takes advantage of the presence of multiple and low-power sensors (accelerometer, gyroscope, proximity, rotation vector, etc.), cameras and even microphones on smartphones to create an alternative way of locating a device. Fingerprint based localization techniques have many advantages when compared to GPS, according to Vo et al. [32], fundamentally by saving battery life because of the low-power sensors integrated in current smartphones and by providing more accuracy. The main fingerprint types used in the literature are a visual fingerprint, motion fingerprint, signal fingerprint and hybrid fingerprint [32], [33].

“Visual fingerprint-based localization uses an image captured by the user to match against geotagged images in a database to identify the location” [32]. An example of visual fingerprint based application is Google Goggles [37], an image search application which can identify products, landmarks or paintings appearing in mobile images. By taking a photo of one landmark, Google Goggles can identify it and then localize the device [33]. Another example can be Vuforia Object Scanner [38] an “Android application that provides real-time visual feedback on the target quality, coverage, and tracking performance of the scanned objects” [32]. According to Du et al. [33], the main constraint in visual fingerprint-based localization techniques is the matching speed and battery consumption.

Motion fingerprint-based localization uses the motion data of users obtained by the built-in sensors such as accelerometer and compass, combines the readings and match them with a map of the area of interest to estimate the location of mobile devices [33]. The readings from the compass are used to estimate the orientation of the mobile devices and the readings from the accelerometer are used to detect the traveled distance [32]. These measures are made periodically and used as fingerprints and for localization.

Signal fingerprint-based localization is a technique widely used in “places where a large number of WiFi infrastructures are deployed, especially in indoor environments” [33]. According to Vo et al. [32] the basic idea is to find the location of a mobile device by comparing its signal pattern received from multiple transmitters (e.g., WiFi access points or base stations) with a pre-defined database of signal patterns. In addition, combining multiple fingerprint types can lead to more robust hybrid fingerprint-based localization systems with better performance and, most of all, to minimize the tradeoff between accuracy and power consumption of most techniques. In each technique or fingerprint type, the implementation details vary. ***Figure 13*** shows the classification of different fingerprint types, as well as the performance objectives for the systems, proposed by Vo et al. [32].

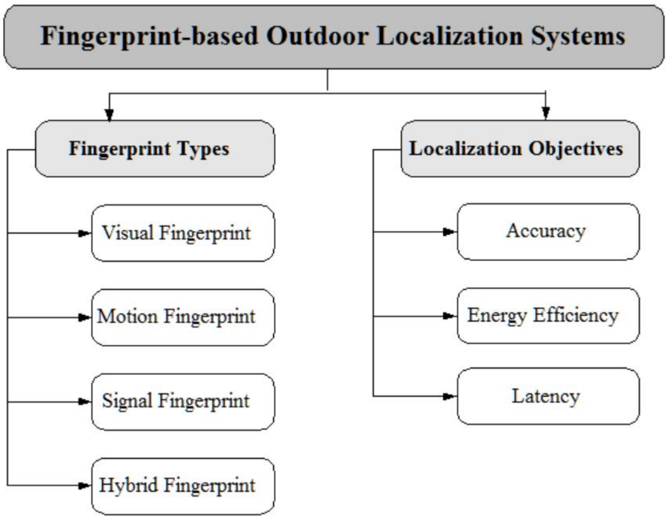

Figure 13: Different modes and performance objectives for fingerprint-based outdoor localization systems

2.3.2 Pedestrian Dead Reckoning for Outdoor Localization

The most widely used technique for generating motion fingerprints is pedestrian dead reckoning (PDR) [32], which is a localization algorithm that utilizes inertial sensors or inertial measurement units (IMU) that contains three-axis (accelerometers, gyroscopes, magnetometers, and others) data to estimate a pedestrian´s position [39]. PDR periodically records data from the IMU to estimate the travel distance and the direction of the movement of the pedestrian. The current location is estimated using the previous location and the latest motion fingerprint [32]. In that way, PDR assumes that the position of a pedestrian only changes with stepping movements, so it observes the movement of steps and integrates the inertial sensor measurements over the time to estimate the position of a pedestrian [39]. According to Wang et al. [40] and compared with other localization techniques, PDR can give an accurate position in a short period of time, faster update of the pedestrian´s position and lower power consumption. However, IMUs can generate small errors, the errors in inertial sensor measurements can be accumulated by integration. To reduce or eliminate the accumulation of errors, map-matching algorithms have been proposed [39]–[41]: "these algorithms reduce the positional errors by matching the user trajectory to the closest road on the map" [38]. The implementation of a PDR algorithm involves the following steps: travel distance estimation (which involves step detection and step length estimation) and travel direction estimation (which involves heading between each detected step estimation), [32], [39]–[41].

1. Step Detection

There are many different techniques for step detection in the literature. Wang et al. [40] characterizes the daily movement of a pedestrian using a phone and classifies the motion mode in two categories, the movement state, and the phone pose. The movement state represents the global motion of pedestrian, including Walking, Running, Upstairs and Downstairs. The phone pose represents the pose of holding or placing a phone, including Holding, Calling, Swinging and Pocket. Support vector machine (SVN), a supervised learning model, is employed to recognize the movement states of a pedestrian. Based on the results of the classification, a flexible PDR algorithm for multi-motion modes was proposed. The basic description of the proposed solution is that the phone produces a periodic motion with the steps while a pedestrian is walking, the magnitudes of accelerations in the accelerometer data can reflect the step characteristics. The acceleration magnitude presents a sinusoidal wave and the peaks represent the probable steps of the pedestrian, where a peak is a point in the signal that is preceded by a rise and followed by a slope, as shown in ***Figure 14***.

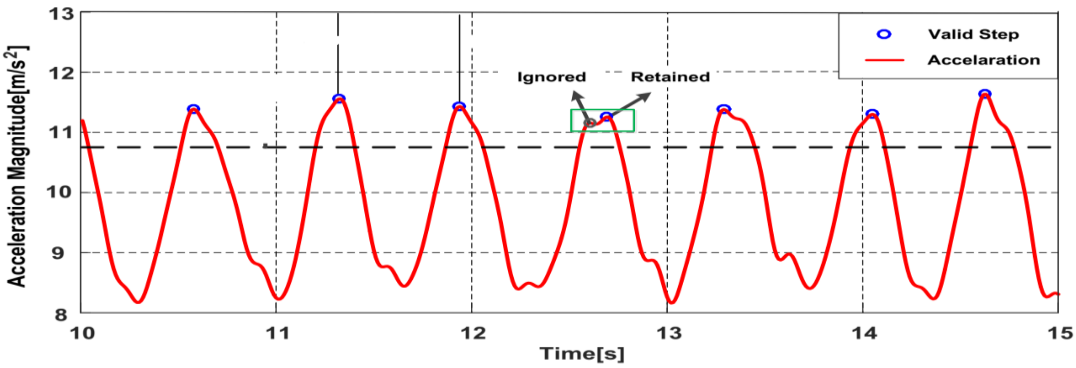

Figure 14: A sinusoidal wave of acceleration magnitudes for step detection [40]

In another perspective, Nagpal [42] carried out a study to develop a pedometer recording app based in Android sensors, employing new approaches to identify advanced information on steps, such as what type of steps is given (running, jogging or walking), the duration of each step and also step lengths. The author observed, from the plotted accelerometer data, that every type of step (walking, jogging, and running) has a unique signature pattern. These observations are described below along with the data obtained during the tests, the data represents every 10 seconds during the user´s locomotion.

For the walking signature, it was realized that most of the time, the highest peak is registered when the foot hits on the ground, the common pattern consists of a few high peaks followed by a few troughs, and finally, one possible way to identify all the individual steps is the count all the highest peaks, each of which is followed by one lowest trough, as shown in ***Figure 15***.

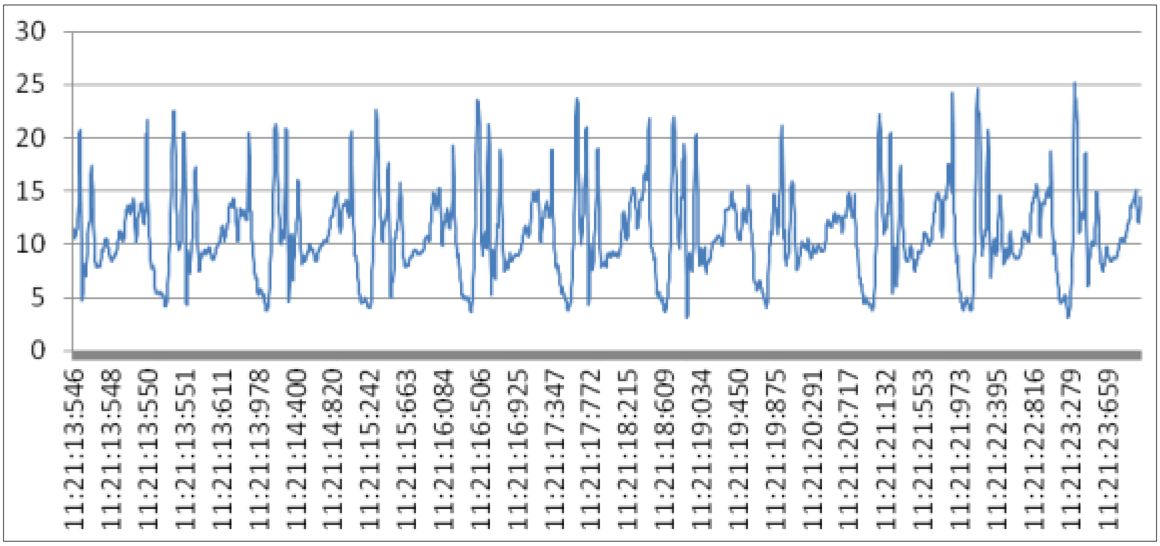

Figure 15: Signature produced by a person walking at a normal pace [42]

For the jogging signature, it was realized that the highest peak value for normal walking reaches close to 20, while the highest peak for jogging or fast walking crosses 25, in the same time duration (10 seconds) 11 highest peaks (steps) are noted. The time interval between each jogging step is shorter than the one between the normal walking steps, as shown in ***Figure 16***.

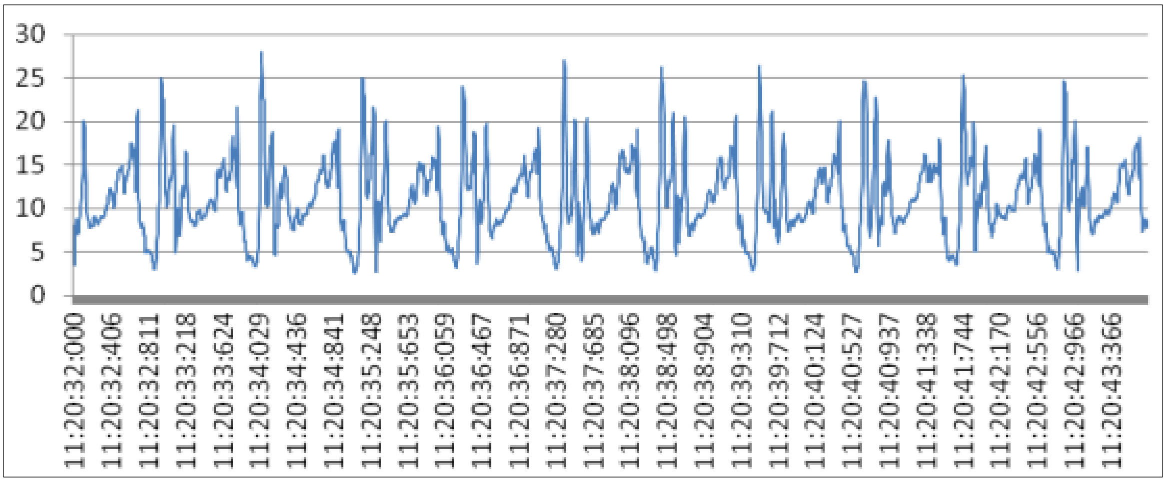

Figure 16: Signature produced by a person walking fast or jogging [42]

For the running signature, it was realized that the highest peak value for running crosses 30, while the highest peak for normal walking reaches close to 20, and the highest peak for jogging or fast walking doesn't cross 25. The time interval between running steps is shorter than the intervals between normal walking or jogging steps, as shown in ***Figure 17***.

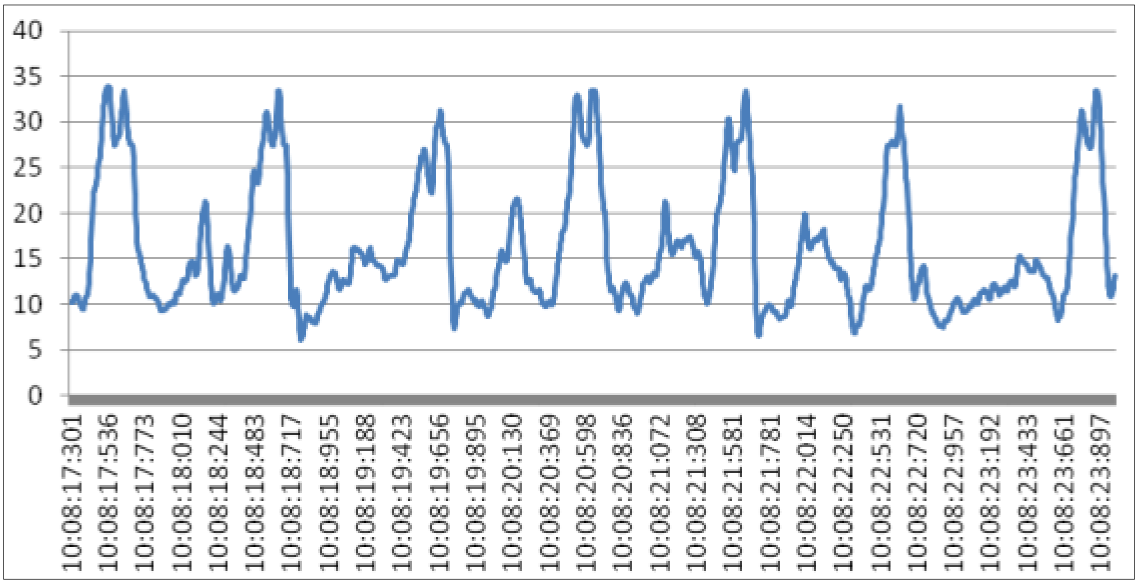

Figure 17: Signature produced by a person running [42]

By observing the accelerometer data and identifying unique signature patterns, Nagpal [42] provided a helpful approach for the correct and reliable identification of different types of steps.

Another useful approach for step detection is presented in [32], by continuously sensing and analyzing the data received from the accelerometer. Firstly, it defines two thresholds $u_n + \sigma_n$ and $u_n - \sigma_n$ to represent the levels for characterizing "up" and "down" patterns respectively. u_n represents the average of the series and σ_n the standard deviation, and an undefined state ∧. Considering acceleration magnitudes of $a_1, a_2 \dots a_n$, where a_n is the most recent data received and mapped to a bit according to the mapping shown below:

$$Q(a_n) = \begin{cases} 1 \ \ if \ a_n > u_n + \sigma_n \\ 0 \ if \ a_n > u_n - \sigma_n \\ \wedge \qquad \text{otherwise,} \end{cases}$$

This mapping yields a sequence of bits. The technique merges consecutive 1s into a single bit 1, 0s to 0, and ∧s into ∧ to form a step with a pattern of "10" or "1∧0". Whenever a step is detected, it will be reported to the map matching process for enhancing the localization accuracy.

2. Step length and Distance Estimation

"The step length varies from person to person, it should be a variable which is related to the pedestrian" [40]. According to Ju et al. [39] the step length can be modeled as a linear combination of a constant value and the step frequency, combining the walking frequency and the acceleration variance, as shown in the formula below:

$$Step\ Length = \alpha * WF + \beta * AV + \gamma$$

where WF is walking frequency, AV is a variance of the accelerometer magnitude between steps, and α, β, γ are pre-learned parameters according to the pre-calibration using a supervised method as described in the step detection section. Another approach for estimating the step length was proposed in the *AutoGait* system [43], which is a system that builds a walking profile for each user, through the gathering of GPS segments along with his steps. At the end of each segment the step frequency is averaged along with the length of each step, these values are then inserted into a regression model which, with enough samples, would outputs a linear function that represents an approximation of the linear relation between step length and step frequency. This model is consulted during step detection to obtain the length of a given step with frequency f.

3. Direction Estimation

The travel direction of a user can be measured using techniques mainly based on the orientation sensors provided by mobile phones. These sensors combine data from different IMUs such as the gyroscope, magnetometer, and accelerometer.

"Travel direction can be estimated using the angular displacement based on gyroscope readings" [32]. Gyroscope data is with respect to the Cartesian frame of reference of the phone itself. The gyroscope measures either changes in orientation or changes in rotational velocity (rate gyro). The magnetometer sensor measures the changes in the Earth's magnetic field. It provides the raw magnetic field strength in units of microtesla (*μT*). The orientation sensor is a software sensor and measures the position of the device relative to the Earth's frame of reference, by processing the raw values of the accelerometer and magnetometer sensors [42]. The combination of these sensors can be used for PDR orientation.

2.3.3 Related Systems and Comparison

Study 1: AnDReck Positioning estimation using PDR

AnDReck: Positioning Estimation using Pedestrian Dead Reckoning on Smartphones [44], is a system developed by Carlos Simões, a student at *Instituto Superior Técnico de Lisboa*, for his *MSc* dissertation. The system was designed to provide accurate positioning of a pedestrian in both indoor and outdoor conditions; the positioning technique used was pedestrian dead reckoning (PDR).

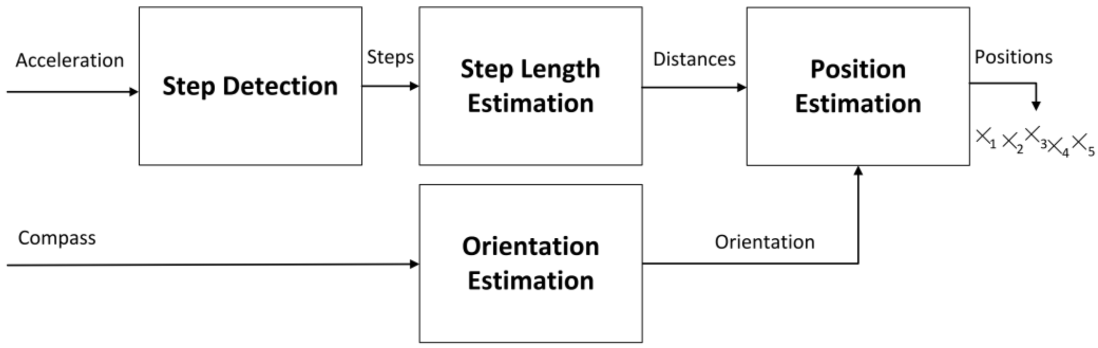

Figure 18: Study 1 System Architecture [44]

The proposed architecture is designed accounting for two different usage scenarios: Calibration and Positioning Estimation. In the Calibration scenario, step and location data are used to calibrate the step length model, while in the Positioning Estimation scenario step and orientation data along with the calibrated model are used to generate positions iteratively. The step detector receives an input signal from built-in sensors and analyses it. These samples serve as input to a modeler that estimates the step length as the samples arrive. The orientation estimator receives orientation information from built-in sensors and combine them with the step length data; this information is passed to the position estimator that produce a single position iteratively.

Study 2: PDR using Barometric Elevation and Map-Matching

This is a study developed by Broyles et al. [45] aimed at the development of a real-time, self-contained outdoor navigation application that uses only the existing sensors on a smartphone in conjunction with a preloaded digital elevation map. The proposed algorithm implements a particle filter that fuses sensor data with a stochastic pedestrian motion model to predict the user´s position, then it compares the smartphone´s barometric elevation with the digital elevation map to constrain the position estimate. ***Figure 19*** shows the block diagram of the solution.

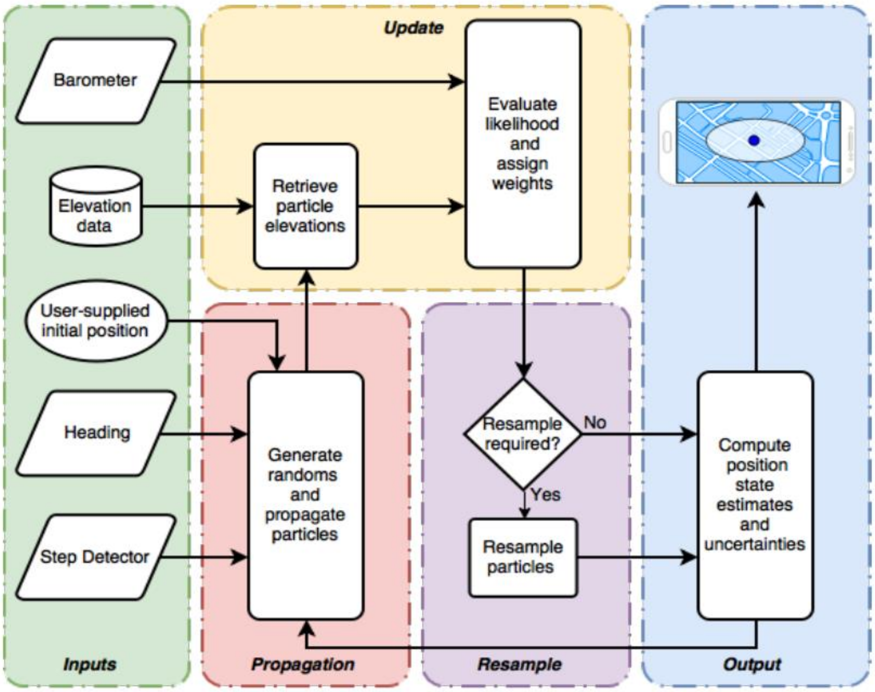

Figure 19: Study 2 Algorithm Functional Block Diagram [45]

The process begins with the initialization of the particle filter with an initial location provided. When a step is detected the particles are propagated using the current heading input and randomly generated values for step length deviations and biases. When a barometric elevation measurement is available, the elevation corresponding to each particle’s location is extracted from the digital elevation model in the update stage. A likelihood function is then evaluated which assigns particle weights based on how closely each particle’s elevation matches the realized measurement. Particles with elevations that closely match the barometric elevation are more influential in the final position.

Study 3: Hybrid Outdoor Fingerprint Localization

A hybrid outdoor localization scheme with high-position accuracy and low-power consumption is the title of a study developed by Hongwei Du et. al. [33]. The study explored WIFI fingerprinting, sensor information, and GPS statistics in order to develop a hybrid outdoor localization scheme utilizing crowdsourced WiFi signal data and built-in sensors in smartphones. The architecture of the proposed solution is presented in ***Figure 20***.

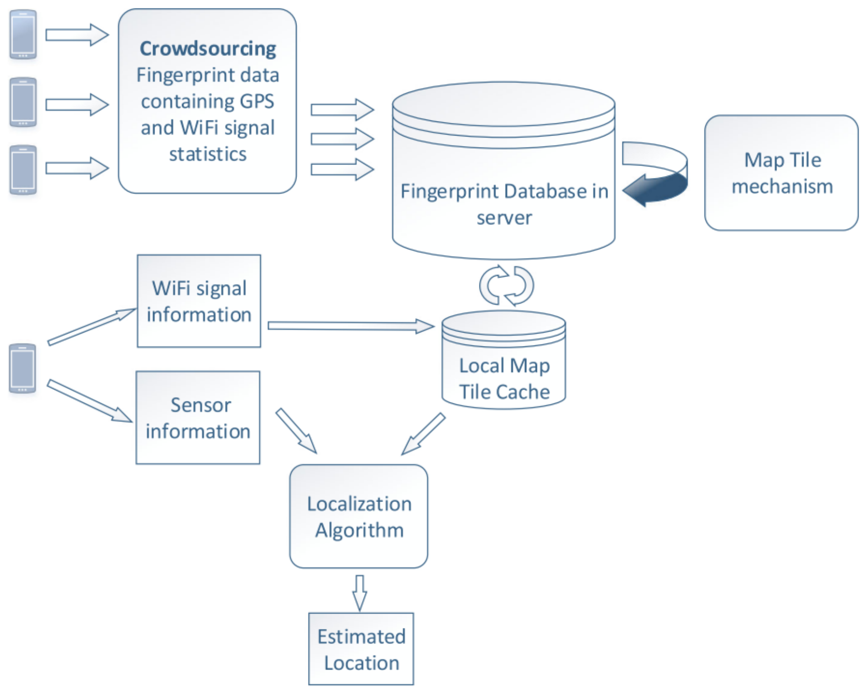

Figure 20: Study 3 System Architecture [33]

The proposed scheme consists of two phases to locate mobile devices. In the first phase, an offline WiFi fingerprint database is constructed via crowdsourcing technique. This database includes not only the WiFi fingerprint data but also the GPS statistics. In the second phase, the real-time WiFi and GPS measurements are used to match the records in the database to achieve localization. To improve the localization accuracy and matching speed, it divides the map into map tiles using a map tile cache mechanism and sensor readings to limit the matching space. The location of the fingerprint with a minimum difference will be selected as an estimation for the location of the device.

Comparison

All three systems analyzed propose alternative solutions for outdoor localization to achieve more accurate positioning in challenging environments. The study 1 proposed a solution without the use of GPS and that can be used offline; the solution is based on a core implementation of the PDR algorithm with an improved step length estimation technique, the results of the simulation scenarios were good. The study 2 introduced a new approach to improve the accuracy of PDR implementations by correcting possible precision errors of the IMUs using a map-matching technique with a digital elevation map, the solution can be used offline and presented relatively good results. The study 3 proposed a hybrid approach, combining wifi fingerprinting, GPS measurements and built-in inertial sensors, the results obtained were very satisfactory when compared to a simple GPS implementation, but it cannot be used entirely offline. The variables that will be used to evaluate the three systems are related to localization accuracy obtained in the testing experiments realized in each study. The technologies, and techniques used in each study will also be considered to derive the advantages and disadvantages of each approach. The comparison is presented below in ***Table 3***.

Study	**Technologies**	**Advantages**	**Disadvantages**
Study 1	PDR with Step length model	Offline & non-GPS based, Low power consumption, Automatic step length calibration	Weak IMUs error drifting handling,
Study 2	PDR with Map Matching, Particle filter, and elevation sensing	Non-GPS based, Robust implementation, Robust error drifting handling, Much precise than regular PDR and basic GPS.	Weak step length estimator.
Study 3	WIFI fingerprinting, PDR and GPS statistics	Big infrastructure, Highly accurate, Map-matching considered.	Online and GPS based

Table 3: Comparison table of the systems analyzed in this document.

2.4 Summary

This chapter presented the state of the art in the fields under study in the present dissertation. With the literature review on augmented reality, it was clear that this technology provides numerous advantages and it’s a trend nowadays. But in addition to having advantages, the implementation of mobile AR solutions still faces some challenges, mostly related to limited memory, limited computational capability, and power consumption, since AR applications consume resources that can affect battery life and the efficiency of the mobile devices. Another challenge is the localization accuracy, and motion fingerprint localization solutions can be an alternative in non-GPS enabled scenarios. Considering the related systems comparison realized in section 0, the most suitable motion fingerprint technique for the problem introduced in the present work is pedestrian dead reckoning (PDR), considering that it can bring more accurate positioning in offline mode and using only smartphone built-in sensors.

3. Architecture

This chapter presents the architecture of the proposed positioning solution. Section 3.1 describes the functional and non-functional requirements considering special constraints identified in the problem formulation, and section 3.2 presents an overview of the architecture and a description of the role of each of its components.

3.1 Requirements

The architecture proposed in this chapter is based on the literature review, including relevant systems and novel techniques to build a solution that is capable of outputting relative positions during pedestrian locomotion especially in outdoor conditions. The architecture defines a set of structures including software elements, relations among them and properties needed to reason about the specific solution. Considering the problems identified in the present work, the solution must comply with the following requirements:

I. Functional requirements: It must have the ability to increase the opportunities for positioning in challenging environments, in the absence of a network signal and without the use of global positioning systems. To achieve this, it should be able to accurately detect steps, estimate distance traveled, estimate pedestrian orientation and estimate the current position.

II. Non-Functional Requirements: They consist of the following quality attribute requirements: availability, interoperability, performance and functional correctness.

- Availability: “Availability refers to the ability of a system to mask or repair faults such that the cumulative service outage period does not exceed a required value over a specified time interval” [46]. In the present work, this requirement involves the study and identification of the main aspects that affect the availability of PDR based systems and implementation of fault tolerance and error recovery mechanisms. The availability tactics that will be used are “fault detection” and “fault recovery”, by detecting and handling exceptions in every component.

- Interoperability: “Interoperability is about the degree to which two or more systems can usefully exchange meaningful information via interfaces in a particular context“ [46]. As it is necessary to make readings from the device sensors and communicate these data to the Unity engine, the solution will contain an interface between the Unity engine and the Android OS. To achieve a reliable communication, the interoperability tactic that will be used is the “manage interfaces orchestrate”, which is a tactic that uses a control mechanism to coordinate, manage and sequence the invocation of particular services [46]. Using a specific interface, the Unity-based application will integrate and access the PDR solution.
- Performance: “It's about time and the software systems ability to meet timing requirements” [46], the performance tactic that will be used is “control resource demand” by managing the sampling rate. The readings made by the PDR system will be based on event listening, only executing operations when specific conditions are met or every time when a step is detected.
- Functional Correctness: This requirement involves the improvement of positioning accuracy and precision, since GPS provides an average accuracy error of approximately 15 meters without environment obstructions, the proposed solution should provide more accuracy within the urban park under study. The correctness of the positions provided by the solution will be improved using a map-matching approach.

3.2 Architecture Overview

Considering the literature review and the specific requirements identified in the previous section, PDR is an appropriate approach to provide a solution to the presented problem, as PDR systems provide accurate positioning using low power consumption inertial sensors. The proposed solution will be implemented as an Android library that can be integrated into Unity-based applications or other Android based applications. The block diagram of the system is presented below.

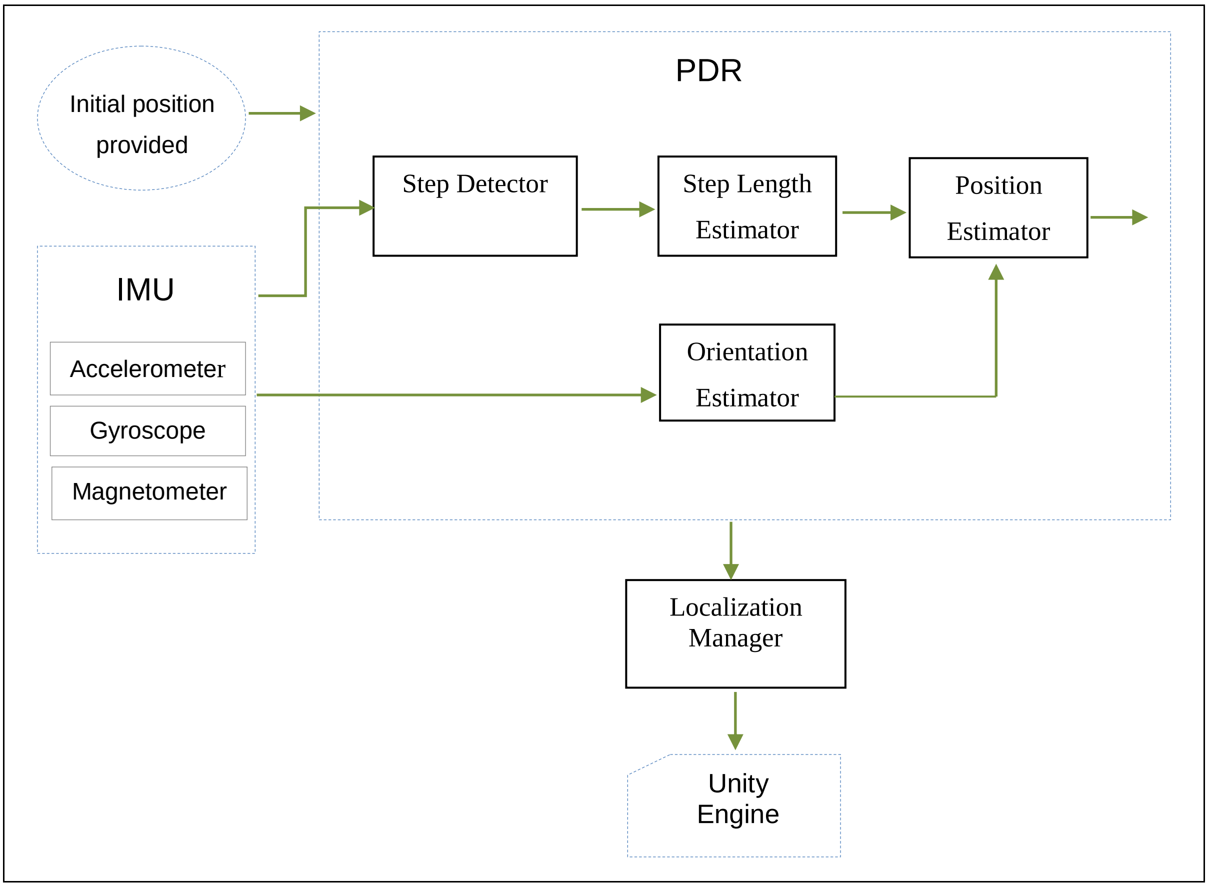

Figure 21: Block Diagram of the PDR library

The proposed architecture contains four main components, the initial position, the inertial measurement unit (IMU) sensors integrated in mobile devices (3D accelerometer, 3D gyroscope and 3D magnetometer), the pedestrian dead reckoning library, which consists in a step detector, a step length model and estimator, an orientation estimator and a position estimator, and finally a localization manager which is an interface that allows the integration of the PDR library with the Unity engine, in this case, the *EduPARK* game-like application.

The initial position is a latitude-longitude coordinate provided when the location system is launched: it serves as initial contextual position to start recording the pedestrian locomotion. In the present work, these initial positions will be based in specific points in the urban park and in positions of specific fiducial markers that initialize certain game paths.

The inertial measurement units (IMUs) consist of smartphone sensors that will be used to record motion fingerprints: the readings from the accelerometer are used to detect the traveled distance, while readings from the compass (accelerometer and magnetometer) are used to estimate the orientation. These IMUs are accessed from the PDR solution using Android motion sensors APIs. IMU sensors can suffer from drifting errors, which are accumulated acceleration errors that grow indefinitely. This happens even faster if the sensors used are low-cost, since each update accumulates even higher errors. To overcome this problem, the technique that will be used is called low-velocity updates [47], which consists of resetting the acceleration errors whenever a step is detected. Errors may still occur, but they do not accumulate as quickly.

The pedestrian dead reckoning (PDR) solution consists of two phases or scenarios: step length model calibration and positioning estimation. In the first scenario, the initial position provided, and step data obtained from the sensors are used to calibrate the step length model, while in the positioning estimation scenario, having the already calibrated step length model, step and orientation data are used to generate positions using a PDR algorithm that will be described later. This second scenario runs iteratively along the pedestrian locomotion.

The Location Manager is an intermediary script that allows the integration, instantiation, and invocation of PDR services from the Unity environment, in this way the PDR solution can be used in any Unity-based application. The component also has the map-matching implementation, which map the location provided by the PDR solution to the map of the area of interest to improve the precision of the localization solution.

An initial localization coordinate (latitude, longitude) is provided when the PDR solution is launched, then the system listens to every event generated by the IMU sensors to detect steps. For every step detected the system performs a distance estimation based in the initial position and the step length model. The distance data combined with the pedestrian orientation data (also obtained from the sensors) and using a specific algorithm will result in a relative position output which will be matched to the area of interest in a posterior and independent process of map-matching to estimate a precise location of the pedestrian. This process is initialized every time the game-like application needs localization services.

3.3 Summary

In this chapter, the architecture of the proposed location solution was presented. The need for a robust localization solution that can solve the regular constraints in the use of GPS justified the choice of pedestrian dead reckoning approach. In this context, the proposed architecture defines a set of tactics for availability, interoperability, performance and functional correctness of the solution. The proposed solution consists of a PDR library that can be integrated in any Android based application.

4. Implementation

This chapter presents the specific aspects regarding the implementation of each component of the PDR solution. Section 4.1 presents the implementation of the step detector component. Section 4.2 presents the aspects regarding the implementation of the step length estimator and the final distance estimations. Section 4.3 presents the implementation of the orientation estimator and section 4.4 the implementation of the position estimator. Section 4.5 presents the analysis class diagram of the solution, and at last, section 4.6 presents the aspects regarding the integration of the PDR solution with the EduPARK application.

4.1 Step detection

The step detection component is responsible for the identification of steps given by the pedestrian irrespective of device pose in smartphone usage environments. The technique employed to identify the steps is based on the accelerometer data, decoupling peak-valley relationships in the magnitude of acceleration. In this context, a step consists of a peak and its adjacent valley. To achieve a robust step detection this implementation introduces small adjustments to the step detection technique and considerations proposed by Nagpal [42], as described in 2.3.2.

The step detection component is based on the Android step detector and step counter motion sensors. Those are very similar software sensors used to count steps: both sensors are based on a common hardware sensor, which internally uses the accelerometer, although Android still treats them as logically separate sensors. These sensors are battery optimized and consume very low power [42] [48]. The step detector sensor triggers an event each time a step is taken by the user, which corresponds to when the foot hit the ground generating a high variation in acceleration. This sensor has very low latency in reporting the steps, which is generally within 1 and 2 seconds. The step counter sensor returns the number of steps taken by the user since the last power-on of the phone. The step detector sensor has lower accuracy and produces more false positives compared to the step counter sensor [42]. Although being more accurate, the step counter sensor has more latency, as it uses extra time after each step to remove any false positive values.

The localization solution must estimate the position of the pedestrian for every step the user gives since the system is launched. The step detector sensor is the appropriate sensor for that purpose: It has very low battery consumption and is highly optimized on the hardware level. The implementation of the step detection component consisted of the creation of the `StepDetector`, which implements the `SensorEventListener` interface so that it can receive sensor events. Firstly, the Android `SensorManager` and `Sensor` objects are initialized in the constructor using the identifier constant `Sensor.TYPE_STEP_COUNTER`, as shown below.

```
public StepDetector (SensorManager sm) {
        this.sensorManager = sm;
        this.sensor = sm.getDefaultSensor(Sensor.TYPE_STEP_DETECTOR);
}
```

The step detector sensor is registered in the `start()` method and unregistered in the `stop`() method. This means that in order to use the library, first it must call the `start`() method in order to register the sensor and after finishing unregister it with the `stop`() method. Every time a step occurs the `onSensorChanged` method is triggered, which starts processing the step length, orientation and new position estimation using the `StepDetectionListener`, which is an interface that initiates the execution of the other components. This process runs iteratively for every step detected. The complete implementation of this component can be consulted in Appendix D.

4.2 Step length and distance estimation

Step length and distance estimation is a process that aims at the precise quantification of traveled distances during pedestrian locomotion. The first approach that was considered but discarded later was proposed by Cho et al. [43], consisting in the development of a walking profile for each user in an early learning phase, recording the step frequency along a segment and inserting the values in a regression model, this model is used during step detection to obtain the length of a given step with a certain frequency. The approach implemented was proposed by Nagpal [42], consisting of the classification of each step detected as “running”, “jogging” or “walking”, this classification is based on the magnitudes of the accelerometer data, as described in section 2.3.2. The study experimental data found that walking a single step covers approximately 0.5 meters, jogging a single step covers approximately 1 meter and running a single step covers approximately 1.5 meters. Using Nagpal´s experimental data will provide a simple but concise approach for defining the lengths of each step that is given at a certain point, multiplying the distance covered by each type of step with their respective numbers to get the total distance traveled. Another affordance of the Nagpal´s approach is that it will avoid an additional operation to consult the length of a specific step in a model during step detection, in that way improving the efficiency at a cost of not estimating the length based in the specific user profile but in the averaged values provided by the experimental data.

The implementation of the step length estimation is made in the step detection phase: when a step is detected the event that is triggered performs the invocation of the respective operation in the `StepDetectionListener`, which specifies the step length based on the classification of the step; the values are based in the experimental data obtained in the Nagpal [42] tests. The complete implementation of this component can be consulted in Appendix D.

4.3 Orientation estimation

The orientation estimation is a process that aims the determination of the orientation of a device in relation to the Earth´s coordinate system to accurately determine the direction of movement of a pedestrian.

The device coordinate system is relative to the device´s screen when it is held in the default or natural orientation, this is a standard 3-axis (x, y, z) coordinate system where the x-axis is horizontal and points to the right edge of the device, the y-axis is vertical and points to the top edge of the device, and the z-axis extends up from the surface of the screen, negative z values are behind the screen, as shown in ***Figure 22***.

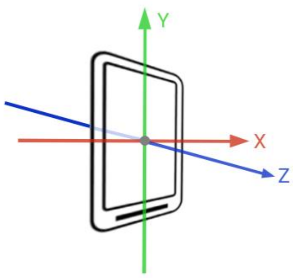

Figure 22: Device´s coordinate system

The Earth´s coordinate system is relative to the surface of the Earth itself, where the y-axis points to magnetic north, the x-axis is 90 degrees from y pointing approximately east and the z-axis extends up into space, negative z extends down into the ground, as shown in ***Figure 23***.

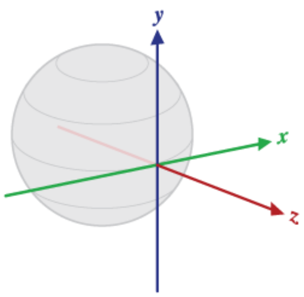

Figure 23: Earth´s coordinate system

Orientation angles describe how far the device is oriented or tilted with respect to the Earth´s coordinate system, consisting of azimuth, pitch and roll angles [42].

Azimuth represents “the degree of rotation made by the phone around the z-axis. It can also be seen as the angle between the magnetic north and the phone's y-axis” [42], its value varies from 0 to 360 degrees, 0 is magnetic north. Azimuth represents the direction (north/ south/ east/ west) the device is pointing.

Pitch represents “the degree of rotation made by the phone around the x-axis. Its value can vary from 180 to -180 degrees” [42]. Pitch represents the top-to-bottom tilt of the device, where 0 is flat.

Roll represents “the degree of rotation made by the phone around the y-axis. Its value can vary from 90 to -90 degrees” [42]. Roll represents the left-to-right tilt of the device, where 0 is flat.

The implementation of orientation estimation involves the development of a digital compass, which is a combination of acceleration and magnetic signal values. The Android system provides a set of software sensors, also called motion sensors [48], which drives its data by processing the raw values of the accelerometer, magnetometer or gyroscope IMUs, providing orientation estimations by giving the azimuth, pitch and roll angles. Early versions of Android included an explicit sensor type for orientation (`Sensor.TYPE_ORIENTATION`), which was a software-only sensor that combined data from other sensors to determine heading and tilt for the device, this sensor type was deprecated in Android API 8 due to accuracy and precision issues [42].

The orientation estimator component of the PDR solution under development, uses the Android rotation vector sensor (`Sensor.TYPE_ROTATION_VECTOR`), which is a software sensor that uses the accelerometer, gyroscope, and magnetometer if they are available. This sensor reports very stable data in adequate response time, it needs to initially orient itself and then eliminate the drift that comes with the gyroscope over time. The component registers and listens to the rotation vector sensor and using the rotation matrix from `SensorManager` gets the orientation angles (azimuth, pitch, and roll). The heading of the pedestrian consists of the azimuth angle obtained in this process. Its value is then transmitted to the position estimation component described next.

4.4 Positioning estimation

The pedestrian dead reckoning algorithm consists of a horizontal plane (2D) [39], [49]. Given an initial position of the pedestrian $(x_0,\ y_0)$; to compute the current position (x_k, y_k) at step k, given the current step length s_k and heading φ_k:

$$x_k = x_{k-1} + s_k \cos \varphi_k \tag{1}$$

$$y_k = y_{k-1} + s_k \sin \varphi_k \tag{2}$$

Where x_{k-1} and y_{k-1} are the previous position coordinates.

The PDR library receives an initial location coordinate, which consists of latitude, longitude coordinates. For every step detected, a new position is calculated using the step length estimation s_k, the heading φ_k obtained from the orientation estimation process, and the previous or initial position known. To improve the accuracy, an additional map-matching process is executed. The final algorithm can be consulted in Appendix D.

4.5 Analysis and Design

A class diagram illustrates a set of classes, interfaces, collaborations, and respective relationships, usually of dependency, generalization, and association. Its implementation aims to model the structure of the PDR solution and its simple collaborations. It is an important tool to show the internal structure of the solution, visualizing how objects collaborate internally to meet the system's functionalities.

Figure *24* presents the analysis class diagram for the PDR solution. Its elements correspond to the components described in the previous sections.

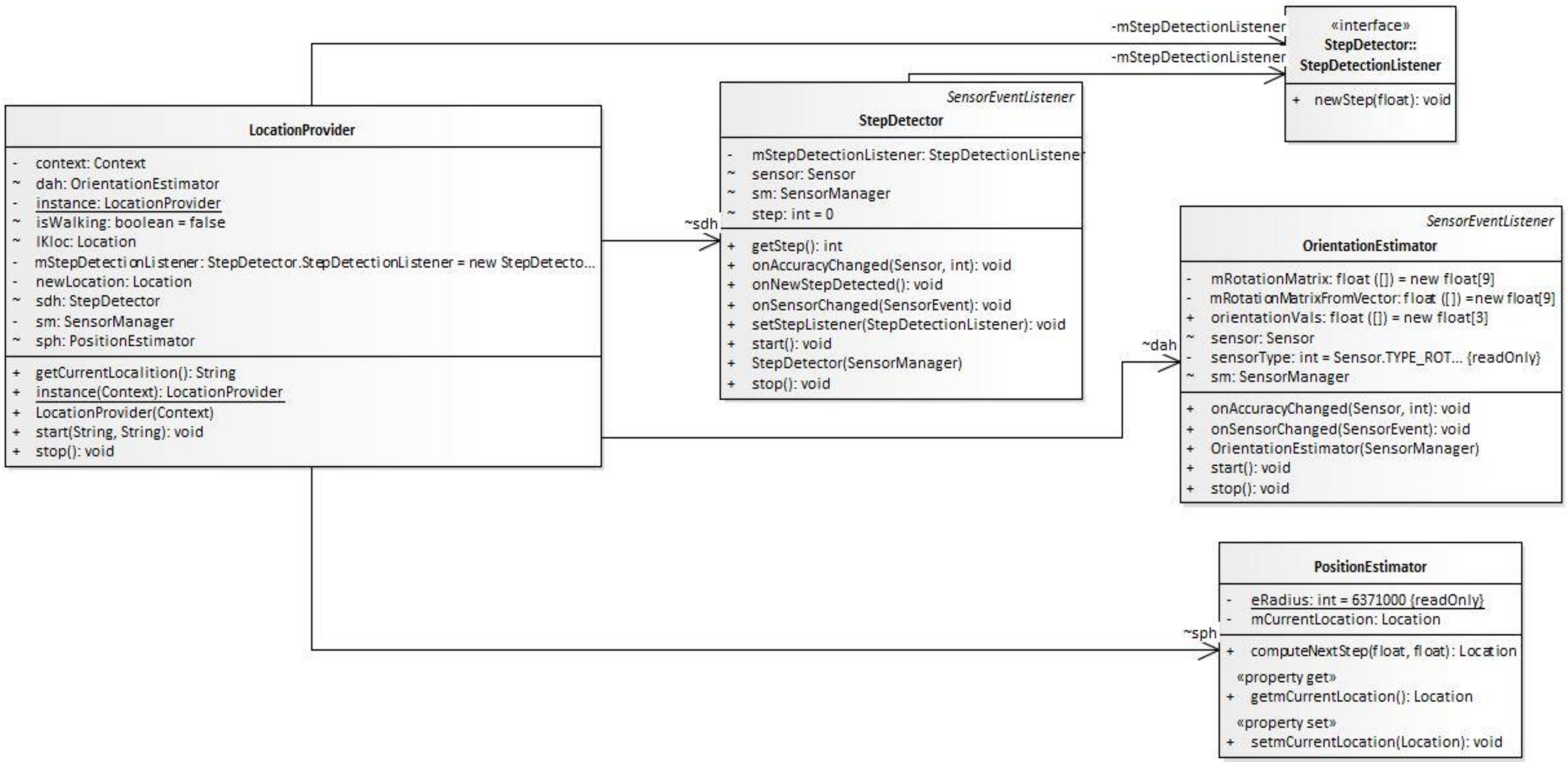

Figure 24: UML class diagram of the PDR solution.

4.6 Integration of the solution with Unity

The PDR solution is implemented as an Android library, its components are developed in Android. The localization manager that allows the integration of the PDR solution in the Unity engine is developed in the C# programming language. This structure allows a functional coordination between both environments, the EduPARK application and the Android OS.

The PDR library is integrated into the Unity-based game-like application as an Android plugin. The instantiation and invocation of services are realized through the localization manager component, which is an interface that uses the scripting API `AndroidJavaObject`, which is a Unity representation of a generic instance of `java.lang.Object`, and the `AndroidJavaClass`, a Unity representation of a generic instance of `java.lang.Class`, to instantiate and get results generated by the PDR library.

4.7 Summary

This chapter described the implementation details of each component of the pedestrian dead reckoning library developed. All the components were implemented using Android software sensors. The step detector makes use of `Sensor.TYPE_STEP_DETECTOR` sensor to record steps of the user. Step length and distance estimation are measured using the parameters proposed by Nagpal [42]. The orientation estimation is obtained using the Android sensor `Sensor.TYPE_ROTATION_VECTOR`. Finally, the final position estimation is calculated for every step detected, using the distance, orientation estimations, and an algorithm that computes the latitude and longitude coordinates that corresponds to the estimated position.

5. Evaluation

This chapter describes the experiments designed to evaluate the correctness and the performance of the proposed positioning solution and the usability of marker-less AR in the EduPARK application. Section 5.1 describes the initial experiments made to test the correctness of the proposed solution. Section 5.2 describes the usability tests realized to evaluate the functionalities implemented in the EduPARK application for testing purposes, using the proposed positioning solution.

5.1 Initial experiments

Multiple tests with different characteristics were performed to evaluate the performance of the developed positioning solution. The methodology chosen to validate the solution was to perform tests in each component of the solution separately. The solution integrates different and independent components, as described in chapter 4. Step detection, distance estimation, orientation estimation, and position estimation. Each of these components is evaluated individually with specific parameters against the real-world values.

The test field was the *Infante D. Pedro* urban park, in Aveiro. **Figure 25** illustrates the map of the park, the different colors represent different zones of the park that were mapped to organize the structure of the game quizzes. The chest icons represent fiducial markers and the pink icons represent monuments in the park. As the park contains fiducial markers that represent POIs used to enable the exploration of AR content, the test scenarios developed consisted in the definition of specific paths to drive the tests, where a path is a sequence of fiducial markers or monuments.

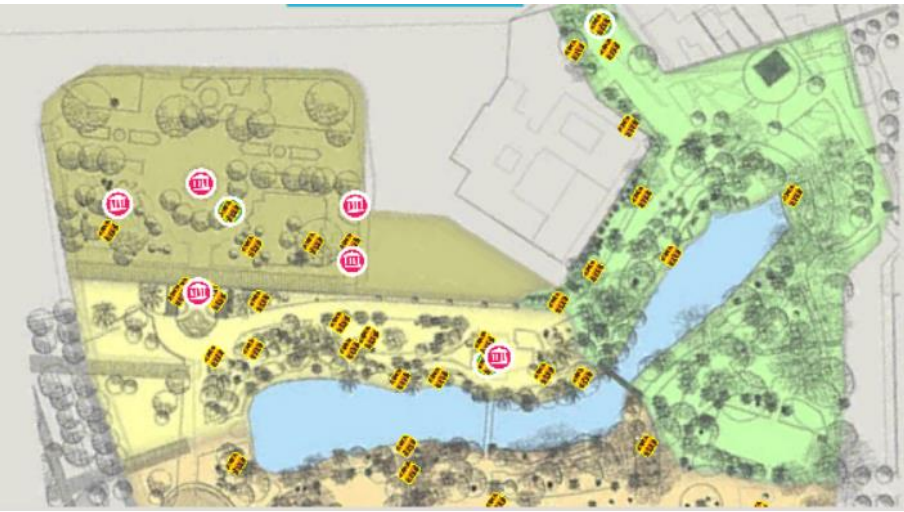

Figure 25: Map of the urban park with all points of interests illustrated

Two paths consisting of an interconnection of different POIs were created, as illustrated in ***Figure 26***.

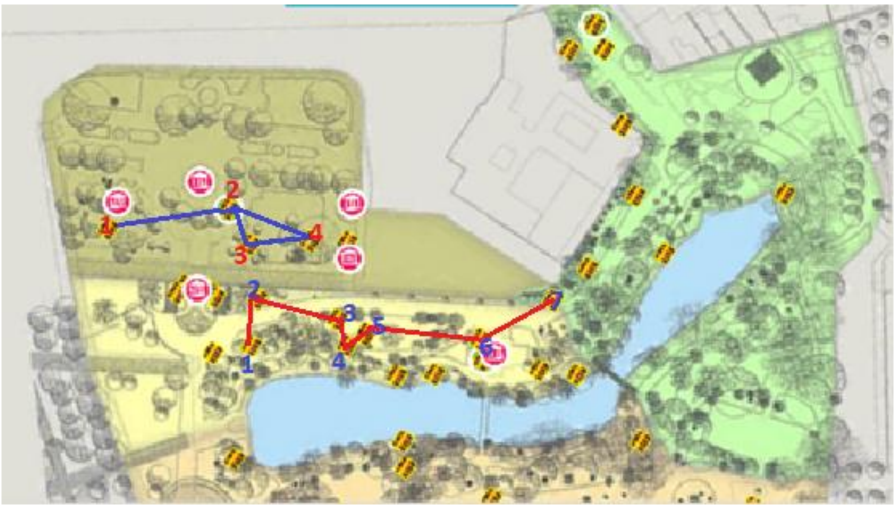

Figure 26: Paths created for tests

The first path, Path 1 (in blue), contains 4 POIs and the Path 2 (in red), contains 7 POIs. The initial process consisted of the selection of the points and the measurement of the distance from one point to another.

Since the system performs step detection, distance estimation and orientation estimation, each of these components was evaluated considering specific restrictions. To evaluate the Step detection component, steps were counted from one point to another following the illustrated sequence and considering only the walking signature. The values obtained will be confronted with the values estimated by the proposed solution, providing a useful comprehension of the accuracy and error rate of the solution.

To evaluate the step length and the estimated distance a specific approach was taken; step length is not trivial to measure since it involves individual measuring of each step. The approach consisted of performing measurements in the floor, making footprints for each step and measuring the length afterward. The distance is measured by adding the length of each individual step and also by computing the distance between the geographical coordinates of the POIs in each path; this distance is then confronted with the values estimated by the PDR solution.

As the paths created for tests group POIs that are organized in a logical sequence, the direction taken from one point to another is easily identified. The orientation evaluation is made by recording the direction estimated by the component in each individual step. The final position estimation is evaluated by calculating the distance error from the real position and the estimated position. The final position consists in the last point of each path.

The considered tests were performed with a Xiaomi Redmi Note 5A, Android version 7.

5.1.1 Step detection performance

The step detection component is evaluated on step counting accuracy to assess that no steps are being missed and the steps counted are properly measured. As each step represents a positioning event, every miscounting can affect the precision and accuracy of the system or even originate positioning errors.

The number of steps miscounted is obtained by calculating the difference between the number of steps counted manually and the ones detected by the solution. This difference value represents the step count error and is illustrated in the equation 3. The step counting accuracy ratio is calculated by dividing the step error value by the expected steps to obtain the error ratio whose complement value is the accuracy ratio. This is illustrated in equation 4.

$$StepError = abs(EstimatedStepCount - RealStepCount) \quad (3)$$

$$StepAccuracyRatio = 1 - \frac{StepError}{RealStepCount} \quad (4)$$

The results for the step detection and counting tests are presented in ***Table 4***, the values were computed using the gathered data from the system, manually counted steps, and the equations presented earlier. In the shortest path, Path 1, the step error average value was 1.25 steps with a standard deviation value of 0.5. Step counting accuracy average value was 95.27% with a standard deviation value of 1.29%. In the Path 2, the step error average value was 1.33 steps with a standard deviation value of 1.21 steps, and the step counting accuracy average value was 95.58% with standard deviation value of 3.57%.

Path	Segment	Steps Counted	Steps detected	Step Error	Accuracy
1	1 -> 2	42	40	2	95.23%
	2 -> 3	15	16	1	93.75%
	3 -> 4	22	21	1	95.45%
	4 -> 2	29	30	1	96.66%
2	1 -> 2	12	11	1	91.66%
	2 -> 3	38	36	2	94.73%
	3 -> 4	10	10	0	100%
	4 -> 5	8	8	0	100%
	5 -> 6	43	40	3	93.02%
	6 -> 7	34	32	2	94.11%

Table 4: Results of the step detection and counting tests

5.1.2 Distance estimation performance

The distance estimation tests consisted in measuring the distance between each point in the path. The step length is obtained from the model described in the implementation. As the step detection accuracy also affects the distance estimation accuracy, the distance accuracy ratio was obtained by calculating the complement of the ratio between the absolute difference between the estimated and the actual traveled distance, as illustrated in equation 5 and 6.

$$DistanceError = abs(EstimatedDistance - RealDistance) \quad (5)$$

$$DistanceAccuracyRatio = 1 - \frac{DistanceError}{RealDistance} \quad (6)$$

The results regarding the distance estimation and considering the steps detected by the system are presented in ***Table 5***. In Path 1, the average distance error value was 1.74 meters with a standard deviation of 0.79 meters. The distance accuracy ratio was 93.67% with a standard deviation value of 0.99%. In the Path 2, the average distance error value was 1.18 meters with a standard deviation of 0.60 meters. The distance accuracy ratio average value was 94.43% with a standard deviation value of 2.28%.

Path	Segment	Distance (m)	Estimated Distance (m)	Distance Error (m)	Distance Accuracy
1	1 -> 2	41.33	38.6	2.73	93.39%
	2 -> 3	14.11	14.91	0.8	94.63%
	3 -> 4	21.03	19.43	1.6	92.39%
	4 -> 2	30.49	32.34	1.85	94.27%
2	1 -> 2	16.37	15.46	0.91	94.44%
	2 -> 3	28.48	26.88	1.6	94.38%
	3 -> 4	9.43	9.13	0.30	96.81%
	4 -> 5	7.42	8.23	0.9	90.15%
	5 -> 6	41.77	39.74	2.03	95.14%
	6 -> 7	31.33	29.98	1.35	95.69%

Table 5: Results of the distance estimation tests

5.1.3 Orientation and position estimation performance

The orientation component is not evaluated individually. Orientation performance is assessed measuring the localization errors that are encountered in each point of the path, as the positioning accuracy is strongly dependent on the accuracy of the orientation estimations. The evaluation is performed in relation to the coordinates of each point defined in the two paths created for tests. In each test scenario, positions estimated using the PDR solution and positions estimated using GPS where gathered and compared.

The localization performance is evaluated in terms of localization errors, which consists in the distance between the point where the mobile/user is, and the point estimated by the localization system.

Table *6* illustrates the localization errors obtained using the PDR solution and GPS. These values represent the performance of each system during the tests.

Path	Segment	PDR Location Error (m)	Location Error GPS (m)
1	1 -> 2	2.24	8.44
	2 -> 3	3.2	5.93
	3 -> 4	3.76	11.85
2	7 -> 6	1.12	5.39
	6 -> 5	3.95	7.63
	5 -> 4	3.56	10.9
	4 -> 3	2.34	1.57
	3 -> 2	2.13	2.84
	2 -> 1	1.54	7.29

Table 6: Positioning errors registered using PDR and GPS.

Table *7* illustrates the comparison between the proposed PDR solution and the use of a standard GPS for positioning in the urban park.

Path	**System**	**Mean Error (m)**	**Standard Deviation (m)**
Path 1	PDR	3.07	0.78
	GPS	8.74	2.97
Path 2	PDR	2.44	1.12
	GPS	5.93	3.41

Table 7: Localization performance comparison.

Considering that the Cumulative Distribution Function (CDF) of the localization error gives the probability that the localization error takes on a value less than or equal to x meters. ***Figure 27*** and ***Figure 28*** illustrate the CDF of the localization error obtained in the tests realized in path 1 and path 2.

In Path 1, the PDR system registered fewer localization errors than GPS. Observing the CDF graph for Path 1 (***Figure 27***), the line that represents the performance of the PDR solution resides completely before the GPS line in all occurrences, what reveals that the PDR was more accurate than the use of GPS.

Figure 27: CDF of the localization error: performance of both the proposed PDR solution and GPS in Path 1.

In Path 2, the PDR system also registered fewer localization errors than GPS. Observing the CDF graph for Path 2 (***Figure 28***), the PDR line resides before the GPS line in all occurrences, registering less than 4 meters of cumulative localization errors, what reveals that the PDR was more accurate than GPS.

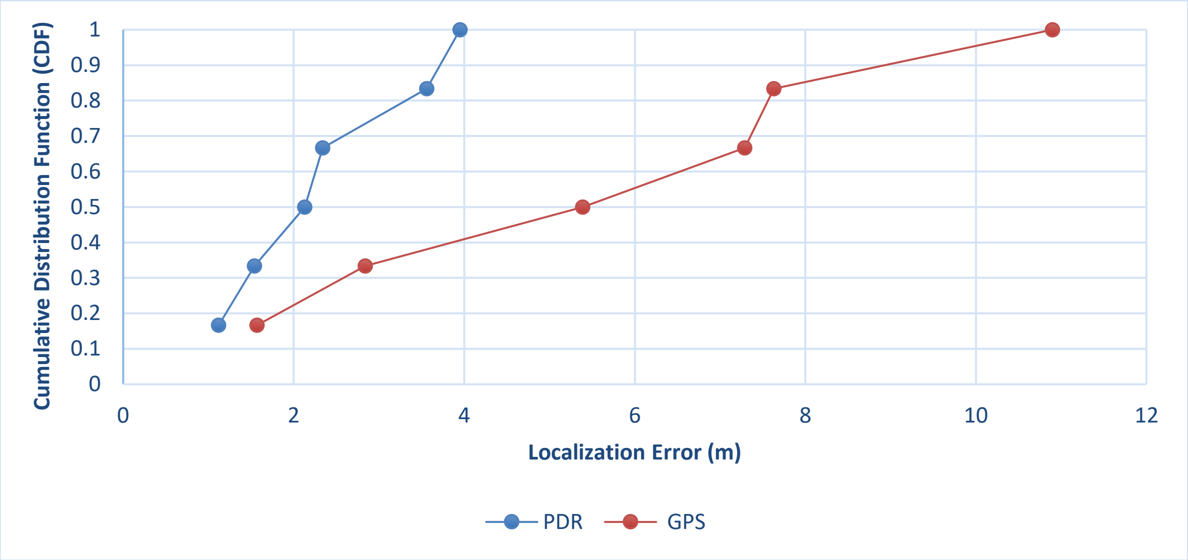

Figure 28: CDF of the localization error: performance of both the proposed PDR solution and GPS in Path 2.

5.1.4 Results and Discussion

Initial experiments were done to estimate the precision and proper functioning of the proposed positioning solution. The objective of the initial experiments was to evaluate whether the proposed positioning solution based in pedestrian dead reckoning may be a reliable alternative to Android GPS in providing accurate positioning in the urban park under study.

The experiments were realized for each component of the system and the results were compared to the real-world values and against GPS accuracy.

The step detection component registered relatively good results when considering that the average error rate was approximately 1 step in Path 1 and approximately 2 steps in Path 2. The average accuracy ratio of the component is above 95% on both paths. These results are encouraging when compared with those obtained in other studies such as [44].

The distance estimation component, which includes step length estimation, registered an average error value of 1.74 meters in Path 1, and approximately 1.5 meters in Path 2. The average accuracy ratio is above 94%.

The proposed PDR solution registered an average localization error value of 3.07 meters in Path 1 while using Android GPS resulted in an average localization error of 8.74 meters with a standard deviation of 2.97 meters, subtracting the standard deviation in the localization error average value results in 5.77 m. In this context, GPS was 2.7m less accurate than the proposed system.

In Path 2, the average localization error value was 2.44 meters while using GPS resulted in an average localization error of 5.93 meters, subtracting the standard deviation of the GPS to its localization error results in 3.49 meters, so GPS was 3.49 meters less accurate than the proposed PDR system. These results reveal that the PDR system was more accurate than GPS.

5.2 Usability studies

This section describes the experiments designed to evaluate the usability of the *EduPARK* application using marker-less augmented reality. The experiments aimed to evaluate if not using a fiducial marker in some parts of the game affects learning or the enjoyment of the users, and to prove the initial hypothesis, that marker-less augmented reality can be achieved through positioning based in pedestrian dead reckoning localization in the urban park.

The studies consisted firstly in the integration and implementation of specific functionalities in the *EduPARK* application, and the realization of tests where participants realize a set of tasks and then answer usability questionnaires.

Two questionnaires were created in this study. The first is a usability questionnaire (Appendix B) that must be filled for two tasks: a marker-based game session, and a marker-less game session. To avoid learning effects, half of the participants started with the marker-less session, the other half started with the marker-based session. The objective of this questionnaire is to assess if the participant felt a significant difference between exploring AR in the current marker-based configurations and with the proposed marker-less version, and also to evaluate if marker-less AR in the case of the EduPARK application is as enjoyable as marker-based.

The second questionnaire is the System Usability Scale (SUS) in Appendix C, used to quantify the acceptance and the usability of the application.

5.2.1 Experiment Design

The experimental functionalities implemented as test cases allow the execution of a quiz without the use of the fiducial markers, augmented reality is explored using only the geo-position of each point of interest (in this case, the positions of the quiz markers).

In the current workflow of the game, described in section 2.2.3, the user starts a new game by selection in "new game" (in red in ***Figure 29***), then chooses a quiz and follows textual and audio instructions to find the fiducial markers and points of interests (***Table 8***). The user can also use guidance tools to easily find POIs, the tools consist of a compass and a map of the park. The map of the park, shown in ***Figure 25***, has different colors for each zone of the park and interactive tooltips when clicked.

The workflow that was designed to test the PDR solution and the use of marker-less AR starts with the selection of the button “Markerless”, illustrated in green in ***Figure 29***. This option starts a new markerless game test. Afterwards, the user chooses a quiz and follows instructions to find the initial POI. The position of the initial POI is used as initial position when the PDR solution is launched. After finding the initial POI, to find other POIs the user must select the button “Obter Direções” (**Figure *32***), this option initializes the PDR solution and opens the interactive map (**Figure *33***) used to guide the user to find the next POI. Augmented reality content is triggered when the position provided by the PDR solution coincides with the position of the POI. The interactive map panel (**Figure *33***) contains the user location in the map (number 1), the user location coordinates (number 2), the distance travelled (number 3), and the number of steps taken (number 4).

Figure 29: The Home screen of the game

Figure 30: Quiz selection panel

Figure 31: Instruction panel

Table 8: Workflow with marker-less modifications

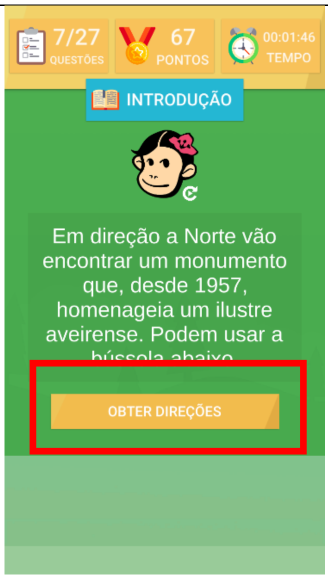

Figure 32: Button that enables the use of the interactive map to reach POIs

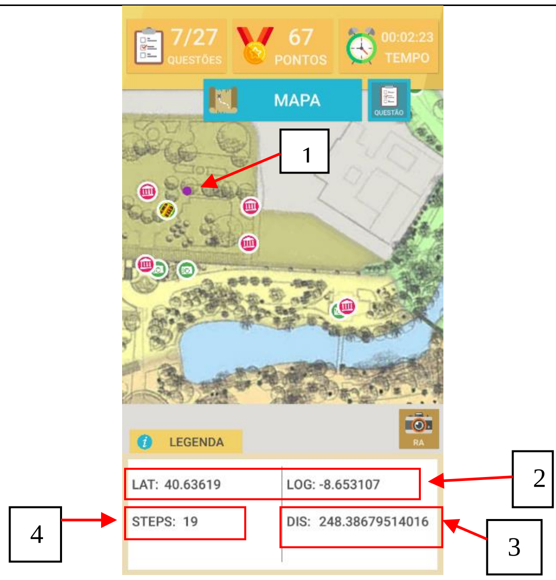

Figure 33: Interactive map with user location, geo-coordinates and number of steps given

5.2.2 Participants

A group of 10 participants was invited to take part in the study. The participants were selected from different areas of the academic community at the University of Aveiro. The group was composed of 3 women's and 7 men's, with an average age of 25 years. All the participants use a smartphone or tablet device regularly and 3 of them had never used augmented reality before.

At the beginning of the study, all the participants answered a questionnaire about demographic and other relevant information to the study (age, gender, occupation, etc.), as presented in ***Table 9***.

Each participant received a mobile device with the app and an explanation of how to use the app. The study was realized with a Xiaomi Redmi Note 5A.

Participant	Age	Gender	Student	First time using AR
1	23	Female	Yes	No
2	25	Male	Yes	Yes
3	27	Male	Yes	No
4	25	Male	Yes	Yes
5	26	Male	Yes	No
6	30	Male	Yes	No
7	29	Male	Yes	No
8	24	Female	Yes	No
9	25	Male	Yes	Yes
10	24	Female	Yes	No

Table 9: Demographic information of the study.

5.2.3 Procedure

The study was realized with one participant at a time. The experiment consisted of two phases, in the first phase the participant completes two tasks with the application and in the second phase the participant answers two questionnaires. In the first task, the participant must complete a quiz (sequence of questions in different POIs) selected by the observer. This consists in the option “new game”, illustrated in ***Figure 29***, it starts a new game session with the current marker-based characteristics. After finishing the first task the participant must answer Task 1 of the usability questionnaire (Appendix B). In the second task, the participant must complete the same quiz in a game session with marker-less configurations, this consists of the option “Markerless” illustrated in ***Figure 29***. After finishing the second task, the participant must answer Task 2 of the usability questionnaire (Appendix B).

In the final procedure of the study, the participant answers the System Usability Scale (SUS) questionnaire (Appendix C).

5.2.4 Results and Discussion

All 10 participants completed successfully the tasks and answered the questionnaires. The questionnaires were analyzed, and the results are presented below.

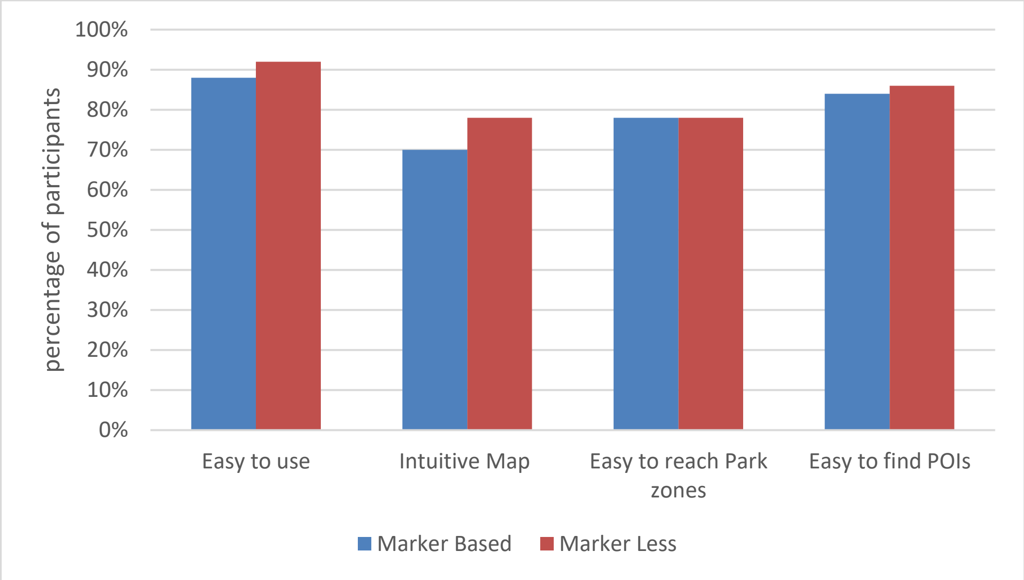

Figure 34: Results related to the usability questionnaire

Regarding the usability questionnaire that establishes a comparative relationship between the two versions of the application, 92% of the participants found the markerless version very easy to use, while only 88% found the marker-based version very easy to use. About the intuitiveness and usability of the map that serves as guide tool to reach different parts of the park, 78% of the participants found it strongly intuitive in markerless version, while only 70% found it strongly intuitive in the marker-based version. Another aspect that can be related to the map is the facility to get different zones of the park using the guided map, in this aspect the results are equilibrated. In terms of finding points of interests, the markerless version presented relatively better results, 86% found it easier in the marker-less version and 84% in the marker-based version. These results show that marker-less augmented reality can help to improve learning in the context of the EduPARK application, and also that the PDR solution, proposed in this dissertation proved to be a relatively good option to achieve good positioning within the park.

Regarding the SUS questionnaire, the results are presented below in ***Table 10***.

Participant	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**SUS Score**
p1	5	2	4	2	4	1	4	1	4	2	**82.5**
p2	4	1	5	3	4	1	3	1	5	3	**80.0**
p3	5	1	4	2	5	1	4	1	5	3	**87.5**
p4	4	1	4	2	3	2	4	2	5	3	**75.0**
p5	4	2	5	1	3	2	5	1	5	2	**85.0**
p6	4	1	4	1	4	1	4	1	3	2	**82.5**
p7	5	2	3	2	4	1	4	1	4	2	**80.0**
p8	5	2	4	2	4	1	4	1	4	3	**80.0**
p9	5	2	5	2	5	1	4	1	5	2	**90.0**
p10	5	2	5	2	4	1	4	1	5	3	**85.0**

Table 10: Results related to the SUS questionnaire

The average score obtained with all participants was 82.75, which reveals that the system has good usability. These results demonstrate that a fingerprint localization technique can be an alternative to accurate positioning in the park, and that marker-less augmented reality has a potential to foster learning and provide an enjoyable game.

6. Conclusions and Future Work

The main objective of this dissertation was the research and implementation of a solution that could provide greater precision in the positioning within the *Infante D. Pedro* park and be an effective alternative to the use of GPS. The quest for an accurate alternative to GPS was intended to bring a compact solution to allow the implementation of marker-less augmented reality functionalities to the *EduPARK* application. Several related solutions were revised in order to choose the appropriate approach for the case under study. Systems with mixed techniques to improve GPS accuracy were considered, but preference was given to solutions that could work completely offline. In this context, fingerprint-based localization approaches were analyzed, and due to its characteristics, preference was given to pedestrian dead reckoning (PDR). Different PDR systems were analyzed and compared, some of them focusing on improving different components that composed their solutions, others focusing on combining different fingerprint localization techniques. The proposed solution consists of an independent library that can be integrated into any Android based application, providing localization services in both indoor and outdoor environments using only an initial position coordinate provided and the smartphone motion sensors.

An architecture was proposed, and implementation aspects of each component were described. The interactions between these components and aspects regarding the integration of the PDR library with the EduPARK application were also described. The PDR library was developed containing the components: Step detector, distance estimator, and an orientation estimator. The values from these components are combined in the position estimator to generate a position estimation. After the development, the system was evaluated across its step detection, distance, and position estimation capabilities. Positive results were obtained for the Step Detection component, and the distance estimation component, when compared to the other similar systems revised during the literature review. Position estimation results were good when compared to GPS, showing that improvements have been achieved although the results seem to be not so positive when compared to other system results in the literature, indicating that the approach followed could be improved.

Another important requirement of this dissertation was the creation and integration of 3D models of the physical monuments in the urban park to new questions introduced in the game-like application. The models were created and are described in Appendix A.

A prototype of markerless AR was developed to evaluate if the PDR solution can solve the positioning problem and allow the implementation of markerless based functionalities. Usability tests were realized with a group of participants and the results were satisfactory. Another test made was the System Usability Scale, which provided very good results when considering that the System Usability Score average is 68 points and the obtained score was 82.75 points.

The results obtained with the present dissertation showed that pedestrian dead reckoning is a good approach to provide accurate positioning within the urban park and markerless AR can be enabled using the proposed solution.

As future work, further tests should be made to the positioning solution proposed in the present dissertation in order to confirm the results obtained and to make improvements. In particular, tests with the school children, since they are the main end users of the EduPARK application.

The present work involved the implementation of a positioning solution to allow the exploration of AR based on geo-localization within the urban park where the EduPARK application is implemented.

The positioning functionalities were implemented in test version of the EduPARK application. Considering the results obtained, the developed positioning library can now be used in the EduPARK application version made available to the public. This should allow the EduPARK team to develop new marker-less and geocaching tasks in future quizzes. Further studies can then be carried out to evaluate the usefulness of the positioning library.

Bibliography

[1] L. Pombo, M. M. Marques, V. Carlos, C. Guerra, M. Lucas, and M. J. Loureiro, “Augmented Reality and Mobile Learning in a Smart Urban Park: Pupils’ Perceptions of the EduPARK Game,” vol. 80, S. 2nd International Conference on Smart Learning Ecosystems and Regional Developments, Ed. Aveiro: Springer Science and Business Media Deutschland GmbH, 2017, pp. 90–100.

[2] E. FitzGerald, R. Ferguson, A. Adams, M. Gaved, Y. Mor, and R. Thomas, “Augmented Reality and Mobile Learning,” *Int. J. Mob. Blended Learn.*, vol. 5, no. 4, pp. 43–58, Oct. 2013.

[3] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, “Recent advances in augmented reality,” *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 34–47, 2001.

[4] S. K. Kim, S. J. Kang, Y. J. Choi, M. H. Choi, and M. Hong, “Augmented-Reality Survey: from Concept to Application,” *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 2, pp. 982–1004, Feb. 2017.

[5] P. Milgram and F. Kishino, “A TAXONOMY OF MIXED REALITY VISUAL DISPLAYS,” *IEICE Trans. Inf. Syst.*, vol. E77–D, No., p. 15, 1994.

[6] O. Hugues, P. Fuchs, and O. Nannipieri, *New Augmented Reality Taxonomy: Technologies and Features of Augmented Environment*. 2011.

[7] L. Madden, *Professional Augmented Reality Browsers for Smartphones: Programming for junaio, Layar, and Wikitude*. Chichester, West Sussex, 2011.

[8] P. Q. Brito and J. Stoyanova, “Marker versus Markerless Augmented Reality. Which Has More Impact on Users?,” *Int. J. Hum. Comput. Interact.*, vol. 34, no. 9, pp. 1–15, 2017.

[9] “OpenCV library.” [Online]. Available: https://opencv.org/. [Accessed: 28-Jul-2018].

[10] “Vuforia | Augmented Reality.” [Online]. Available: https://www.vuforia.com/. [Accessed: 28-Jul-2018].

[11] “Unity.” [Online]. Available: https://unity3d.com/pt. [Accessed: 28-Jul-2018].

[12] “Augment: Enterprise Augmented Reality Platform.” [Online]. Available: https://www.augment.com/. [Accessed: 28-Jul-2018].

[13] S. Siltanen, *Theory and applications of marker-based augmented reality*. 2012.

[14] “LookTel – Instant Recognition Apps for Persons with Low Vision or Blindness.” [Online]. Available: http://www.looktel.com/. [Accessed: 29-Jul-2018].

[15] J. D. P. Sardo *et al.*, “Portable Device for Touch, Taste and Smell Sensations in Augmented Reality Experiences,” *INCREaSE*, vol. 2, pp. 305–320, 2018.

[16] M. Anton, “Wolverine Device Generates Haptic Feedback For Objects Grasped in VR - TechTheLead - Technology for tomorrow,” 2016. [Online]. Available: https://techthelead.com/wolverine-device-generates-haptic-feedback-objects-grasped-vr/. [Accessed: 16-Aug-2018].

[17] M. Mekni and A. Lemieux, “Augmented Reality : Applications , Challenges and Future Trends,” *Appl. Comput. Sci. anywhere,* pp. 205–214, 2014.

[18] C.-H. Chen, C.-Y. Huang, and Y.-Y. Chou, “Effects of augmented reality-based multidimensional concept maps on students’ learning achievement, motivation and acceptance,” *Univers. Access Inf. Soc.*, pp. 1–12, Nov. 2017.

[19] N. Jain, P. Youngblood, M. Hasel, and S. Srivastava, “An augmented reality tool for learning spatial anatomy on mobile devices,” *Clin. Anat.*, vol. 30, no. 6, pp. 736–741, Sep. 2017.

[20] Y. Turkan, R. Radkowski, A. Karabulut-Ilgu, A. H. Behzadan, and A. Chen, “Mobile augmented reality for teaching structural analysis,” *Adv. Eng. Informatics*, vol. 34, pp. 90–100, Oct. 2017.

[21] H. Tobar-Muñoz, S. Baldiris, and R. Fabregat, “Augmented Reality Game-Based Learning: Enriching Students’ Experience During Reading Comprehension Activities,” *J. Educ. Comput. Res.*, vol. 55, no. 7, pp. 901–936, Dec. 2017.

[22] D. Atwood-Blaine and D. Huffman, “Mobile Gaming and Student Interactions in a Science Center: the Future of Gaming in Science Education,” *Int. J. Sci. Math. Educ.*, vol. 15, no. S1, pp. 45–65, May 2017.

[23] S. E. Bazarov, I. Y. Kholodilin, A. S. Nesterov, and A. V Sokhina, “Applying Augmented Reality in practical classes for engineering students,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 87, no. 3, p. 032004, Oct. 2017.

[24] G. Kurubacak and H. Altinpulluk, *Mobile Technologies and Augmented Reality in Open Education*. IGI Global, 2017.

[25] D. W. F. van Krevelen and R. Poelman, “A Survey of Augmented Reality Technologies, Applications and Limittions,” *Int. J. Virtual Real.*, vol. 9, no. 2, pp. 1–20, 2010.

[26] A. B. Craig, *Understanding augmented reality : concepts and applications*. Morgan Kaufmann, 2013.

[27] L. Pombo and M. M. Marques, “Marker-based augmented reality application for mobile learning in an urban park: Steps to make it real under the EduPARK project,” *2017 Int. Symp. Comput. Educ.*, pp. 1–5, 2017.

[28] L. Yu, S. K. Ong, and A. Y. C. Nee, “A tracking solution for mobile augmented reality based on sensor-aided marker-less tracking and panoramic mapping,” *Multimed. Tools Appl.*, vol. 75, no. 6, pp. 3199–3220, 2016.

[29] M. Antonioli, C. Blake, and K. Sparks, “Augmented reality applications in education,” *J. Technol. Stud.*, 2014.

[30] D. D. Iordache and C. Pribeanu, “A comparison of quantitative and qualitative data from a formative usability evaluation of an augmented reality learning scenario,” *Inform. Econ.*, vol. 13, pp. 67–74, 2009.

[31] A. Solano, F. Ugalde, J. Gómez, and L. Sánchez, “An Augmented Reality Application to Enhance the Children’s Engagement in an Early Development Method for Mathematics Literacy,” vol. 607, Springer Verlag, 2018, pp. 405–414.

[32] Q. D. Vo and P. De, “A survey of fingerprint-based outdoor localization,” *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 491–506, 2016.

[33] H. Du, C. Zhang, Q. Ye, W. Xu, P. L. Kibenge, and K. Yao, “A hybrid outdoor localization scheme with high-position accuracy and low-power consumption,” *Eurasip J. Wirel. Commun. Netw.*, vol. 2018, no. 1, 2018.

[34] J. Wang, J. Luo, S. J. Pan, and A. Sun, “Learning-Based Outdoor Localization Exploiting Crowd-Labeled WiFi Hotspots,” *IEEE Trans. Mob. Comput.*, vol. PP, no. c, pp. 1–1, 2018.

[35] S. Sadowski and P. Spachos, “RSSI-Based Indoor Localization with the Internet of Things,” *IEEE Access*, pp. 30149–30161, 2018.

[36] G. Chen, X. Meng, Y. Wang, Y. Zhang, P. Tian, and H. Yang, “Integrated WiFi/PDR/Smartphone Using an Unscented Kalman Filter Algorithm for 3D Indoor Localization,” *Sensors*, vol. 15, no. 9, pp. 24595–24614, 2015.

[37] Google, “Google goggles.” [Online]. Available: http://www.google.com/mobile/goggles/. [Accessed: 05-Jul-2018].

[38] “Vuforia Developer Portal |.” [Online]. Available: https://developer.vuforia.com/. [Accessed: 05-Jul-2018].

[39] H. Ju, S. Member, S. Y. Park, S. Member, and C. G. Park, “A Smartphone-based Pedestrian Dead Reckoning System with Multiple Virtual Tracking for Indoor Navigation,”

IEEE Sens. J., vol. PP, no. c, p. 1, 2018.

[40] B. Wang, X. Liu, B. Yu, R. Jia, and X. Gan, “Pedestrian Dead Reckoning Based on Motion Mode Recognition Using a Smartphone.,” *Sensors (Basel).*, vol. 18, no. 6, p. 1811, 2018.

[41] D. Trong Bui, N. Nguyen, and G. Jeong, “A Robust Step Detection Algorithm and Walking Distance Estimation Based on Daily Wrist Activity Recognition Using a Smart Band,” *Sensors*, vol. 18, no. 7, p. 2034, 2018.

[42] V. Nagpal, *Android Sensor Programming By Example*. 2016.

[43] Dae-Ki Cho, Min Mun, Uichin Lee, W. J. Kaiser, and M. Gerla, “AutoGait: A mobile platform that accurately estimates the distance walked,” *2010 IEEE Int. Conf. Pervasive Comput. Commun.*, no. 5466984, pp. 116–124, 2010.

[44] C. Simões, “AnDReck : Positioning Estimation using Pedestrian Dead Reckoning on Smartphones Thesis to obtain the Master of Science Degree in Communication Networks,” *Ist. Super. Técnico Lisboa*, no. November, 2013.

[45] D. Broyles, K. Kauffman, J. Raquet, and P. Smagowski, “Non-gnss smartphone pedestrian navigation using barometric elevation and digital map-matching,” *Sensors (Switzerland)*, vol. 18, no. 7, pp. 1–17, 2018.

[46] L. Bass, P. Clements, and R. Kazman, *Software Architecture in Practice*, Third Edit. Addison-Wesley, 2013.

[47] X. Yun, E. R. Bachmann, H. Moore, and J. Calusdian, “Self-contained Position Tracking of Human Movement Using Small Inertial/Magnetic Sensor Modules,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, vol. 3, no. April, pp. 2526–2533.

[48] “Motion sensors | Android Developers.” [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_motion. [Accessed: 30-Aug-2018].

[49] I. Klein, Y. Solaz, and G. Ohayon, “Pedestrian Dead Reckoning With Smartphone Mode Recognition,” *IEEE Sens. J.*, vol. 18, no. 18, pp. 7577–7584, 2018.

Appendix A: 3D Models of Physical Monuments

This appendix illustrates the work done regarding the development of 3D models of the physical monuments within the *Infante D. Pedro* park.

The main objective was the identification of geometric shapes that make up each of the considered monument´s global shape, for the formulation of Algebra and Analytical Geometry questions in the EduPARK application. The geometric shapes and the 3D model will enhance the experience of the user during the quiz.

Turret

The turret is a reservoir of water of the *Infante D. Pedro* park.

Figure 35 illustrates the turret and ***Figure 36*** illustrates the geometric shapes that can be derived observing its shape.

Figure 35: Turret

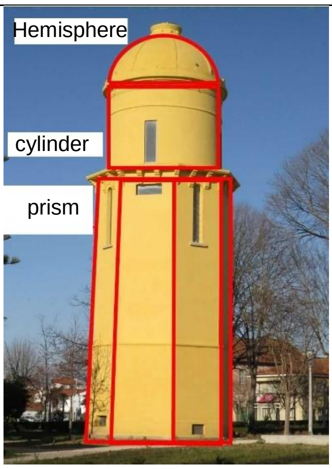

Figure 36: Geometric Shapes in the turret

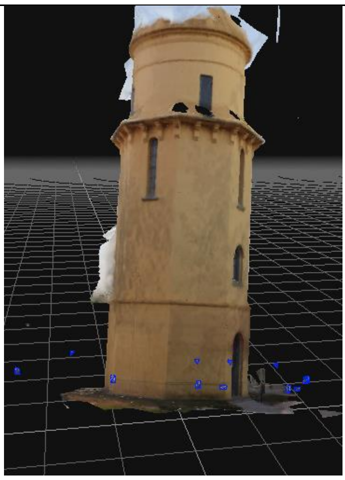

Figure 37: 3D model of the turret

The turret is designed with a cylindrical part covered by a dome in the spherical cap, It can be decomposed into an octagonal prism, a cylinder, and a hemisphere, as shown in ***Figure 36*** and in ***Table 11***.

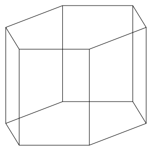	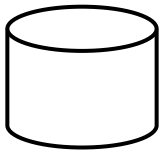	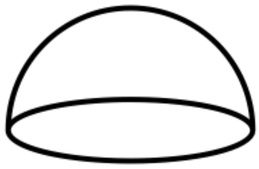

Table 11: Geometric shapes of the turret

Figure 37 illustrates the 3D model created and integrated into the game-like application.

Table *12*. presents a new question that was formulated and introduced in every quiz of the application. The new question aims at the identification of geometric shapes that can be decomposed from the format of the turret.

Question Introduction		
Head to the Water Depot or turret, which was built in 1922.		
Question	**Options**	**Correct Answers**
In what distinct geometric solids can the turret be decomposed?	○ Cylinder ○ Octagonal prism ○ Hemisphere ○ Sphere	✓ Cylinder ✓ Octagonal prism ✓ Hemisphere

Table 12: New question related to the turret.

Bandstand

The bandstand is a covered building of the urban park where several events used to be held on.

Figure 38 illustrates the bandstand, several details of its format are used in different questions of the quizzes.

Figure 39 illustrates the 3D model created, this model was integrated into specific quiz questions.

Figure 38: Bandstand

Figure 39: 3D model of the Bandstand

Table 13 presents a new question introduced and related to the bandstand. The question aims at the precise identification of the correct formula to calculate the area of the base of the bandstand.

Question	
The base of the bandstand is a circumference of 4.8 m radius, it has a side of 3.4 m and an apothem of 4.5 m. What is its area? Recall that the area of a regular polygon is half the product of the perimeter by the apothem.	
Options	**Correct Answers**
○ A = [(3,4 x 8) x 4,8] : 2 ○ A = π x $4,8^2$ ○ A = π x $4,5^2$	A = [(3,4 x 8) x 4,5] : 2

Table 13: New question related to the bandstand

Monument

A monument in honor of *Jaime Magalhães Lima* is another physical object of the urban park under study, it is formed of a polyhedron and a non-polyhedron geometric shape.

Figure 40 illustrates the image of the monument.

Figure 41 illustrates the 3D model that was created and integrated with the questions related to the monument.

Figure 40: Monument

Figure 41: 3D model of the monument

Table 14 presents a new question introduced and related to the monument. The question aims at the identification of the non-polyhedron shape in the monument.

Question	
What is the non-polyhedron that we can identify in this monument?	
Options	**Correct Answers**
○ Sphere ○ Cylinder ○ Cone ○ parallelepiped	✓ Cylinder

Table 14: New question related to the bandstand.

Appendix B: Usability Study Questionnaire

User´s Task Guide

Evaluation Session

This document presents a list of tasks to perform on the available views. This usability test is intended to evaluate the system and not the user. As such, do not feel pressured by time or the need to complete tasks successfully. If you feel difficult, you can ask for help or give up freely one or more tasks.

1. User ID: _____________
2. Age: ____
3. Gender: M ☐ F ☐
4. Education: ____
5. How often do you use smartphone/tablet devices?

   Never Regularly.

   1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
6. How often do you use game-like learning applications?

   Never Regularly.

   1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
7. How often do you use positioning systems?

   Never Regularly.

   1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
8. What is your experience with Augmented Reality applications?

Never used very experienced.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

Task 1	Start the EduPARK game-like application. Start a new game by clicking on “New Game” and select the “Visitors” Quiz. Follow the instructions and, when finished, answer the following questions. 1. The game is intuitive and easy to use. Strongly Disagree Strongly Agree 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 2. I was able to easily use the guidance tools (Compass, Map). Strongly Disagree Strongly Agree 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 3. I was able to easily reach the different zones of the park. Strongly Disagree Strongly Agree 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 4. I was able to easily find the fiducial markers and explore AR. Strongly Disagree Strongly Agree 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ **Write some improvement suggestions for the game.** ____ ____

<table>
<tr>
<td>Task 2</td>
<td>
Start the EduPARK game-like application.
Start a MARKERLESS game by clicking on “Markerless” and select the “Visitors” Quiz. Follow the instructions and, when finished, answer the following questions.
1. The game is intuitive and easy to use.
Strongly Disagree — Strongly Agree
1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
2. I was able to easily use the Map.
Strongly Disagree — Strongly Agree
1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
3. I was able to easily reach the different zones of the park.
Strongly Disagree — Strongly Agree
1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
4. I was able to easily find the points of interest and explore AR.
Strongly Disagree — Strongly Agree
1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐
Write some improvement suggestions for the markerless game.
____
____
</td>
</tr>
</table>

Appendix C: SUS Questionnaire

For each of the following statements, mark one box that best describes your reactions to the system.

1. I think that I would like to use this system frequently.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

2. I find the system unnecessarily complex.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

3. I found the system easy to use.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

4. I think that I would need assistance to be able to use this system.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

5. I found the various functions in this system were well integrated.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

6. I thought there was too much inconsistency in this system.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

7. I would imagine that most people would learn to use this system very quickly.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

8. I found this system very cumbersome/awkward to use.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

9. I felt very confident using this system.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

10. I needed to learn a lot of things before I could get going with this system.

Strongly disagree Strongly agree.

1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐

Appendix D: Implementation

This appendix presents the most relevant parts in the implementation of the PDR solution.

The source code illustrated below shows the complete implementation of the step detector using the Android sensor.

```
public class StepDetector implements SensorEventListener {
   SensorManager sm;
   Sensor sensor;
   private StepDetectionListener mStepDetectionListener;
   int step = 0;

   public StepDetector(SensorManager sm) {
      super();
      this.sm = sm;
      sensor = sm.getDefaultSensor(Sensor.TYPE_STEP_DETECTOR);
   }

   public int getStep() { return step;}

   public void start() {
      sm.registerListener(this, sensor, SensorManager.SENSOR_DELAY_NORMAL);
   }

   public void stop() { sm.unregisterListener(this); }

   @Override
   public void onAccuracyChanged(Sensor arg0, int arg1) { }
   @Override
   public void onSensorChanged(SensorEvent e) {
      if (e.sensor.getType() == Sensor.TYPE_STEP_DETECTOR) {
         onNewStepDetected();
      }
   }

   public void onNewStepDetected() {
      float distanceStep = estimateStepLength();
      step++;
      mStepDetectionListener.newStep(distanceStep);
   }

   public void setStepListener(StepDetectionListener listener) {
      mStepDetectionListener = listener;
   }
```

```
   public interface StepDetectionListener {
      public void newStep(float stepSize);
   }
}
```

Table 15: Source code of the step detector component

The source code below, shows the implementation of the method that is called every time the orientation sensor triggers an event.

```
public void onSensorChanged(SensorEvent event) {
   // Convert the rotation-vector to a 4x4 matrix.
   SensorManager.getRotationMatrixFromVector(mRotationMatrixFromVector,
         event.values);
   SensorManager.remapCoordinateSystem(mRotationMatrixFromVector,
         SensorManager.AXIS_X, SensorManager.AXIS_Z, mRotationMatrix);
   SensorManager.getOrientation(mRotationMatrix, orientationVals);

   orientationVals[0] = (float) orientationVals[0]; // azimuth axis of rotation
   orientationVals[1] = (float) orientationVals[1];
   orientationVals[2] = (float) orientationVals[2];

}
```

Table 16: Orientation data gathering (Source Code).

The source code below, shows the implementation of the method that is called to compute the next position given the orientation and step length.

```
public Location computeNextStep(float stepSize,float bearing) {
   Location newLoc = new Location(mCurrentLocation);
   float angDistance = stepSize / eRadius;
   double oldLat = mCurrentLocation.getLatitude();
   double oldLng = mCurrentLocation.getLongitude();
   double newLat = Math.asin(
Math.sin(Math.toRadians(oldLat))*Math.cos(angDistance) +
Math.cos(Math.toRadians(oldLat))*Math.sin(angDistance)*Math.cos(bearing) );
   double newLon = Math.toRadians(oldLng) +
Math.atan2(Math.sin(bearing)*Math.sin(angDistance)*Math.cos(Math.toRadians(oldLat
)),Math.cos(angDistance) - Math.sin(Math.toRadians(oldLat))*Math.sin(newLat));

   newLoc.setLatitude(Math.toDegrees(newLat));
   newLoc.setLongitude(Math.toDegrees(newLon));
   newLoc.setBearing((mCurrentLocation.getBearing()+180)% 360);
   mCurrentLocation = newLoc;
   return newLoc;
}
```

Table 17: New position calculation method (source code)