Handbook of Research on Technological Developments for Cultural Heritage and eTourism Applications

João M. F. Rodrigues
University of Algarve, Portugal

Célia M. Q. Ramos
University of Algarve, Portugal

Pedro J. S. Cardoso
University of Algarve, Portugal

Cláudia Henriques
University of Algarve, Portugal

A volume in the Advances in Hospitality, Tourism, and the Services Industry (AHTSI) Book Series

IGI Global
DISSEMINATOR OF KNOWLEDGE

Published in the United States of America by
IGI Global
Business Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Names: Rodrigues, Joao M. F., 1971-
Title: Handbook of research on technological developments for cultural heritage and eTourism applications / Joao M. F. Rodrigues, Celia M. Q. Ramos, Pedro J. S. Cardoso, and Claudia Henriques, Editors.
Description: Hershey PA : Business Science Reference, [2018]
Identifiers: LCCN 2017012413| ISBN 9781522529279 (hardcover) | ISBN 9781522529286 (ebook)
Subjects: LCSH: Tourism--Computer network resources. | Culture and tourism--Computer network resources. | Heritage tourism--Computer network resources.
Classification: LCC G156.5.I5 .T43 2018 | DDC 910.285--dc23 LC record available at https://lccn.loc.gov/2017012413

This book is published in the IGI Global book series Advances in Hospitality, Tourism, and the Services Industry (AHTSI) (ISSN: 2475-6547; eISSN: 2475-6555)

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 5

An Initial Framework to Develop a Mobile Five Human Senses Augmented Reality System for Museums

João M. F. Rodrigues
University of Algarve, Portugal

Pedro J. S. Cardoso
University of Algarve, Portugal

Joana Lessa
University of Algarve, Portugal

João André Pereira
University of Algarve, Portugal

João D. P. Sardo
University of Algarve, Portugal

Marco de Freitas
University of Algarve, Portugal

Jorge Semião
University of Algarve, Portugal

Jânio Monteiro
University of Algarve, Portugal & INESC-ID, Portugal

Célia M. Q. Ramos
University of Algarve, Portugal

Roberto Lam
University of Algarve, Portugal

Eduardo Esteves
University of Algarve, Portugal & CCMAR Centre of Marine Sciences, Portugal

Mauro Figueiredo
University of Algarve, Portugal

Alexandra R. Gonçalves
University of Algarve, Portugal

Miguel Gomes
SPIC – Creative Solutions, Portugal

Paulo Bica
SPIC – Creative Solutions, Portugal

DOI: 10.4018/978-1-5225-2927-9.ch005

ABSTRACT

The Mobile Five Senses Augmented Reality System for Museums (M5SAR) project aims to development an Augmented Reality mobile system for museums. Museums are amazing places, where it is important to sensorial augment as much as possible the visits, permitting to see, ear, touch, smell, and taste all the interesting objects there exist. Also fundamental is that visitors with different profiles (child, adult, expert, etc.) can have different experiences when visiting the same object. In the M5SAR system, the visitor uses its smartphone to select the object(s) to explore, and the user interface adapts on-the-fly to the object(s) and user's profile. Simultaneously, when integrated, a paired hardware device allows the extension of the augmented reality system to the human five senses, complementing the visual and auditory information about the objects. This chapter presents the initial framework to develop a five senses mobile adaptive museum system.

INTRODUCTION

The Mobile Five Senses Augmented Reality System for Museums (M5SAR) project aims at the development of an enhanced Augmented Reality (AR) system, to be a guide in cultural, historical and museum events, complementing or replacing the traditional orientation given by tour guides, directional signs, or maps. The system consists of a (i) smartphone/phablet application (APP) and a (ii) hardware device (referenced as HDevice) to be integrated with smartphones/phablets, in order to explore the five human senses: sight, hearing, touch, smell, and taste. Both components, (i) and (ii), can work in stand-alone or integrated fashion.

Nowadays, many personal and context-aware tourism and cultural experiences are constructed based on mobile APPs (Jung, Chung, & Leue, 2015), including the ones that use AR. Those APPs are increasing in number due to the popularity of built-in cameras, global positioning systems, and with the massive availability of Internet connections. On the other hand, most of the present User Interfaces (UI) still traditionally follow a one-size-fits-all model, typically ignoring the needs, abilities, and preferences of individual users. However, past research pointed out that visualization performance could be improved by adapting some of its aspects to the individual user (Steichen, Conati & Carenini, 2014). Conati, Carenini, Toker, and Lallé (2015) state that intelligent adaptive user interfaces (AUI) and/or visualizations, that can adapt on-the-fly to the specific needs and abilities of each individual user, are a long-term research goal. This is due to two main difficulties: (a) the extraction of information about the users' needs and abilities and (b) the implementation of UIs that can adapt/change "themselves" on-the-fly.

Reinecke and Bernstein (2013) refer that a modular UI that allows a flexible composition of various interface elements increases the number of variations of the interface element to the power of the number of adaptable elements. Thus, instead of designing each interface from scratch, a modular user interface approach is possibly a better solution, once it allows achieving many more versions with less design effort. Therefore, the requirement changes to the creation of different designs for all those parts of the interface that are subject to user or cultural preferences. Equal importance should be given to the UI adaptation to users with different impairments. Unfortunately, because of the great variety of existing impairments, it is expectable that manually and, probably, modular designing interfaces, for each one of those impairments, is impractical and not scalable (Gajos, Wobbrock, & Weld, 2008; Rodrigues, Lessa, Gregório, Ramos, & Cardoso, 2016). Nevertheless, the modular and/or adaptive generation of

UIs offers the promise of providing personalized interfaces on-the-fly, although this does not mean that the user will be satisfied with his/her personalized APP. According to Zhao, Lu, Zhang, and Chau (2012), the psychological process behind satisfaction is highly complex and requires a differentiation between transaction-specific satisfaction and cumulative satisfaction. Transaction-specific satisfaction is the judgment of an experienced service encounter at a specific point in time, whereas cumulative satisfaction is the result of “the overall evaluation of all services encountered over time”. Augmented reality exacerbates some of the challenges mentioned before because multiple types of augmentation are possible at once, and proactive APPs run the risk of overwhelming users.

Having the above in mind, at least three main challenges arise in the (i) APP implementation, i.e., in the AUI design and implementation: (a) How to harvest the necessary information about the preferences and skills of each user; (b) From the acquired information/data, how to give “intelligence” to the UI implementation so it will adapt on-the-fly to the user’s specificities; (c) How to develop the AUI, even a modular UI, without being necessary to develop a huge amount of different (sub-)modules and, at the same time, still optimize the user’s experience and the main principles of interaction design.

The traditional AR systems only give sensorial feedbacks to two senses, sight, and hearing. Unlike those, multi-sensorial media focuses on providing immersive communications and enhancing user quality of experience (Yuan, Bi, Muntean & Ghinea, 2015). As mentioned before, the goal of M5SAR is to extend traditional AR to the five senses. With this in mind, this chapter also explores very briefly the implementation of a (ii) hardware device to be integrated with smartphones, allowing the user to feel touch, smell, and taste, therefore expanding the typical experience to all senses. Some existing systems already make use of these concepts, like the Smelling Screen (Matsukura, Yoneda & Ishida, 2013) and the Digital Taste Interface (Ranasinghe, Cheok, Fernando, Nii & Ponnampalam, 2011) which, however, are not portable. The HDevice consists of a compact portable device, adaptable to all smartphones, phablets, and tablets, between 5’’ and 10’’. The HDevice has its own rechargeable battery and communicates via Bluetooth with the mobile application, from which it receives commands to reproduce different sensation contents at appropriate times, though a series of interactive haptic and multi-sensorial media interfaces, like thermoelectric modules, fans, vibration motors, electronic vaporizers, and fragrance dispersers.

This chapter focus on the initial M5SAR system’s framework, namely the APP development with an AUI for a museum application, integrated with gamification concepts (not detailed in this document) and marker-based AR (Cheng, & Tsai, 2013), as well as the hardware device to be integrated with the smartphones to extend AR to the five senses. The main contributions are: (a) the framework for the adaptive on-the-fly card-based UI construction (Adobe, 2016), where the development of the cards has a modular architecture, including the integration with marked-based AR. (b) The framework of the initial (mobile) hardware device to extend the AR to the five senses. In this sense, the chapter is organized as follows. The background and the state-of-the-art are presented in the next section. The following sections introduce the framework to develop the integrated mobile museum system, future research directions, and conclusions.

BACKGROUND

Alvarez-Cortes, Zayas, Uresti, and Zarate (2009) define Intelligent User Interfaces (IUI) as a sub-field of Human-Computer Interaction (HCI), with the goal of improving HCI by the use of new technologies and interaction devices, including the use of Artificial Intelligence (AI) techniques that allow adaptive or intelligent behavior. Akiki, Bandara, and Yu (2014) presented a study about adaptive model-driven UI development systems, where the focus was AUI for mobile application applied to museums, e.g., MNEMOSYNE (Karaman et al., 2016). Schuller (2015) mentioned that what differences future IUIs is the promise to lend them "emotional intelligence": interfaces that "know" and can react appropriately to the satisfaction or anger of their users. The information is thereby increasingly accessed from multiple modalities (Morency, Mihalcea, & Doshi, 2011), in affect recognition and sentiment analysis, thanks to the availability of increasingly large and realistic resources (Schuller, 2015), including deep learning and long-short-term memory architectures (Metallinou et al., 2012), and weakly supervised learning methods. In fact, in the best of all worlds, the system would have sufficient knowledge about a user's culture before he/she first accesses the interface, because the first impression counts (Lindgaard, Fernandes, Dudek, & Brown, 2006). Reinecke and Bernstein (2013) argue that, to appeal to users in expanding markets, a more comprehensive personalization of interfaces to the cultural background is needed. Those authors identify ideas on how to obtain user information in order to subsequently adapt the UI to certain aspects, e.g., by using questionnaires in an initial registration process, through performance tests (Gajos et al. 2008), or by observing the user's interaction (Kralisch, Eisend, & Berendt, 2005). Gajos and Weld (2004) proposed an automatic system for generating UI, a solution based on treating interface adaptation as an optimization problem.

Akiki et al. (2014) present an overview of adaptive model-driven UI development systems, presenting a set of criteria to evaluate the strengths and shortcomings of these systems. In order to approach accessible HCI-designs for mobile devices, the principles of universal design are important to be followed (CEUD, 2016), as well as the guidelines for supporting accessibility (Gnome, 2016). Mobile APPs should move towards completely personalized experiences. Usually, these experiences are built from the aggregation of many individual pieces of content. One way this content could, and is now being presented, is in the form of cards (Babich, 2016). The card-based interaction model is not new and is now spread pretty widely in most of the recent APPs. Experiences are also many times related to gamification which, in this chapter's sense, is the strategy of interaction between people and businesses (museums), based on offering incentives to encourage public engagement with brands in a playful manner. In practice, focused on the recommendation, dissemination, and evaluation, or to attract new customers to the brand, "companies" offer rewards to participants who perform predetermined tasks. The idea involves setting tasks that are in accordance with the company's (in this case museum) objectives, the creation of rules and the application of monitoring systems. The rewards for user interactions can range from virtual incentives such as medals (or "badges") to physical gifts (Negruşa, Toader, Sofică, Tutunea, & Rus, 2015).

This chapter focus also in the implementation of AR. Azuma et al. (2001) defined an AR system as "…supplements the real world with virtual (computer-generated) objects that appear to coexist in the same space as the real world". However, AR can also be defined (depending on the application) as a system that adds information to the environment in order to facilitate HCI in a human environment. The application presented is an AR marker-based, often also called image-based (Cheng et al., 2013). AR-based markers allow adding preset signals (e.g., paintings, statues) easily detectable in the environment and use computer vision techniques to sense them. There are many commercial AR toolkits (SDK) such as Kudan (2016), Catchoom (2016), Wikitude (2016), or Vuforia (2016) and AR content management systems, e.g., Catchoom (2016) or Layar (2016), including open source SDKs, from which probably the most know is the Artoolkit (2016). Each of the above solutions has pros and cons. Some are quite expensive, others consume too much memory (it is important to stress that, the present application will have many markers, at least one for each museum piece), others take too much time to load on the mobile device, etc. Here, the Artoolkit (2016) is briefly compared with an image marker detector developed by the authors, which is based on the ORB binary descriptor (Rublee, Rabaud, Konolige, & Bradski, 2011). A descriptor evaluation can be found in Figat, Kornuta, and Kasprzak (2014).

In terms of museum APPs, almost every known museum has its own, corresponding to a huge amount and variety of applications. For instance, the Wall Street Journal presents an interesting article with their envisage of the best APPs for visiting museums (TWSJ, 2016). The same kind of articles are presented by Information Week, with 10 (fantastic) APPs for museum visits (InformationWeek, 2016), and by The Balance, with some of the fine art museum APPs (Balance, 2016). The use of AR in museums is also not new, including the implementation of head-worn displays (Vainstein, Kuflik, & Lanir, 2016). Vainstein, Kuflik, and Lanir (2016) also discussed the necessary requirements and additional features for useable AR systems in museums. Other AR solutions, nonexclusive for museums, are the Srbija 1914 / Augmented Reality Exhibition at the Historical Museum of Serbia (HMS, 2016), the Invisible Museum from Qualcomm (Qualcomm, 2016), the interactive IPAD Museum catalog from the University of Virginia Art Museum (VAM, 2016), the interactive devices at the Cleveland Museum which includes playing games and social interaction, with face and posture recognition (MWCM, 2016), and the Science Museum - atmosphere gallery (SM, 2016). It is clear to conclude that the development of a museum APP is nothing new; nevertheless, this document will present a framework to develop an APP with different characteristics, which includes AUI and five senses AR.

The second component of the system is a (mobile) multi-sensorial display, which together with the APP, will be able to reproduce different contents for the five human senses. Psychology generally accepts that information will be better perceived if more human senses are used (Kovács, Rozinaj, Murray, Sulema & Rybárová, 2015). Still, most existing multimedia systems in use today focus only on two senses (sight and hearing), being one reason the fact that producing compelling sensations for the other senses is complex. Even though, there has been a series of attempts in the past, but almost none of them seem to be picked up for production.

One technique, which although its limitations is somehow an exception, is the sense of touch through haptic feedback with vibration motors, used for instance in videogames and cellphones (Brewster, Chohan & Brown, 2007). Another interesting technique is the usage of air as a way to stimulate the sense of touch, as the AIREAL solution introduced by Disney Research. Their device uses compressed air to hit the user with an air ring, called a vortex, generating tactile sensations on the skin (Sodhi, Poupyrev, Glisson & Israr, 2013). With the dissemination of Peltier devices, several studies and experiments were conducted on thermoreception. A haptic transmission system for telexistence was developed using the TELESAR V, which was able to detect pressure, vibrations, and temperature on its finger (Kurogi et al., 2013). However, from the point of view of efficiency and energy consumption, the usage of Peltier devices on mobile applications is problematic. Other solutions still exist, such as the virtual reality haptic technology proposed by AxonVR (2016).

The sense of smell reproduction also presents some options, like a flow of air through a scent filter or recipient, vaporizing an aromatized solution, pressurized scented cans, heated or evaporative diffusers, or ultrasonic scent atomization (Yanagida, 2012). One particularly interesting prototype was a wearable olfactory display, which used micro pumps to force a flow of air through an odor filter, releasing it on the user's face (Yamada, Yokoyama, Tanikawa, Hirota & Hirose, 2006). Another peculiar prototype is the Smelling Screen, which combines a display with four fans, rotating at low speed so the user will not perceive the flow of air (Matsukura et al., 2013). For other solutions see e.g. FeelReal (2016) multisensory VR mask. Finally, there is the sense of taste, perhaps the least explored of all senses and probably the hardest to digitally stimulate. In this matter, Ranasinghe et al. (2011) presented a research where two silver electrodes were attached to the user's tongue and pulses applied with different electric properties (like current, frequency, and voltage) seem to produce sour, bitter and salty sensations. Further research demonstrated that combining other influential factors, like temperature, enabled the system to produce a wider variety of results (Ranasinghe, Nakatsu, Nii & Gopalakrishnakone, 2012). Again, other solutions exist; see e.g. the work from Narumi, Kajinami, Nishizaka, Tanikawa, and Hirose (2011).

All the applications that could be related to tourism and cultural activities should obey a minimum of "Accessible Tourism" (EWB, 2016). Tourism experience combines the actions of tourists with a set of memories and emotions related to the visited places, as something "pleasant, engaging, and memorable" (Oh, Fiore, & Jeoung, 2007), and able to satisfy a "wide range of personal needs ranging from pleasure to a search for meaning" (Li, 2000). The consumers/users want to be involved and absorbed into the experience that they are enjoying while having access to simple products (Pine, & Gilmore, 1999). In the above context, the implementation of the present system with its mobile APP, to be successful, depends on the consumers' (museum users) adoption. In this sense, APPs or devices should be validated by a Technology Acceptance Model – TAM (Marangunić & Granić, 2015). The Unified Theory of Acceptance and Use of Technology (UTAUT) is an extension of the TAM (Venkatesh, Morris, Davis, & Davis, 2003), with new improvements in the organizational context and used to investigate the user's intentions when facing a new technology. The UTAUT2 model (Venkatesh, Thong, & Xu, 2012) extends the UTAUT model, by integrating new constructs and relationships to be tailored to the consumer's use context. The UTAUT2 model is adequate, inside the context of the M5SAR project, to study and validate the acceptance of this new technology. Having this in mind, specific surveys will be developed for each segment or group of users, each one with specific characteristics.

FRAMEWORK TO DEVELOP A MOBILE MUSEUM SYSTEM

As previously mentioned, the developed system interfaces with the user through the mobile application installed on the user's smartphone/tablet. This mobile application shows general museum information (e.g., museum's history, map, stores, and collections) and specific information of each museum object. Minimizing as much as possible the network traffic, the mobile application communicates with a "server" each time it needs more contents which, for instance, occurs when the user enters a different section of the museum. The system also integrates a device, HDdevice, that it is lent or rent at the museum's front desk. This device is compatible with different smartphones and tablets.

The mobile application is multiplatform, multilingual and has a modular structure (Reinecke, & Bernstein, 2013) based on UI cards. The development of the UI cards should adapt to the user, by changing the view and the contents or the way they appear. Although the system here presented can adapt itself to more groups of users, at present, four types were considered: child, adult, adult-expert, and family (1+ adult(s) with 1+ child(ren)). How to harvest the necessary information about each user preferences and skills, in order to adapt the UI to the user's specifications, is out of the scope of this chapter.

Other major considerations were contemplated in the APP: (a) deploy the AR module only when the mobile device is in the (almost) vertical position; (b) use of beacons (Estimote, 2016) to detect the approximated user's position in the museum and, in consequence, download any extra contents, if necessary; (c) use gamification principles, i.e., assign "rewards" to the user depending on his/her actions; (d) convert all textual contents to sound contents (audio contents should not disturb other museum users, neither should "silence" the room's ambient sounds); (e) use of adaptive UI and adaptive ("intelligent") navigation, e.g., different (suggestions) paths will be given depending on the objects already seen by the user, as well as its profile; (f) use adaptive luminosity, i.e., the UI luminosity should adapt to the luminosity of the environment and/or piece; and (g) hot/help key available, which appears "spontaneously" each time a user does not know what to do.

This section will detail the most relevant components of the system, as the authors will not go specific over all the elements of the structure (please check the M5SAR project site, for more details). Nevertheless, the four main topics of the framework will be addressed: (a) design consideration, (b) adaptive card implementation, (c) AR and (d) AR device implementation (HDevice). The presented system is being tested in the *Faro Municipal Museum* (located in Faro, Algarve, Portugal). In the future, the user will install a generic museum application in its mobile device, where several museums can be browsed by selecting and downloading the desired contents. Finally, it is also important to stress that the system is being implemented using the Unity development platform (Unity, 2016).

Application Design Considerations

To design the user experience (UX), several objectives were established: the importance to adapt the experience to users' needs, objectives and pace, promoting a pleasurable and ludic experience to engage him/her in the visit; to enrich the visit with an emotional value, which enhances the experience; to educate, giving *in-situ* piece related detailed and clear information; to assist the visitor, with museum map orientation, information on facilities and special needs assistance.

The experience in the museums can be challenging, troubling and overpowering for some visitors, since they have to deal with new information and new situations (Falk, & Dierking, 2013). Pleasurable and ludic aspects help overcome it and therefore are integrated in UX design (Saffer, 2006). The Interaction Design (IxD) approach to emotion here present allows the visitor a subjectivity emotional experience, integrating his/her personal perception and emotional values (Boehner, DePaula, Dourish, & Sengers, 2007), instead of pre-establishing a list of emotions that users should meet. This also meets the purpose of a personalized museum experience. The IxD options outline the APP as: (i) non-obtrusive – layers of information in a window paradigm frame the visitor's surroundings and allow him/her to "navigate" through the exhibition with a non-visually imposing and non-absorbent UI, focusing on the exhibition and not on the UI itself, thus applying the concepts of transparency (Cooper et al., 2014) and flow (Saffer, 2006) to create continuity of experience. (ii) Informative – giving detailed information about collections and pieces, suggesting visit circuits and allowing the visitor to tailor his/her own itinerary, depending on his/her preferences. (iii) Support providing – giving map orientation, guidance, assistance and supporting special needs. (iv) Adaptive – adjust its content to the type of visitor (child, adult, …) and adjust UI visual changes according to each one, taking into account preferences and accessibility issues like sight and physical impairments. (v) Rich, emotional experience provider – fostering and promoting a rich museum experience, enhancing the traditional museum visit also with emotionally driven content created with the convergence of media and languages, in a multisensory designed experience result.

The UI employs the premise previous established in UX, to be user adaptive and convey the perception of easy to use, effective, useful, clever and visually pleasurable. Therefore, the UI adapts to different types of visitors by adjusting visualization of content and layout/graphic adjustments (e.g., size, opacity, contrast, movement). A modular UI approach organizes visual space into chunks of information easily adjusted to the user's needs in number and size. The navigation applies Natural User Interfaces (NUI) strategies, using natural interactions through multi-touch, gestures, movements and also speech (Blake, 2013); and clear contextualized visual support to orientate navigation, comprising users' needs, pace, and digital literacy levels.

The museums are places filled with a visually rich environment: fill with color, forms, textures, etc. The UI layout should not impose itself to the user to avoid competing, for the visitor full attention, with the visual richness of the exhibition pieces. Visual consistency, clarity, and hierarchy applied to UI layout promote order/functionality and pleasure in perception. It responds to a set of attributes that convey museum context and the user experience designed: freshness, elegance, trustworthy, clarity, ludic. So, the graphic language is shaped by: (i) a reduced palette of soft predominantly light and moderated contrast colors in icons and text (suggests freshness, elegance, and clarity; for senior user, contrast is augmented); (ii) reduced number of elements on screen (less elements for senior user and children), organized in a clear structure, promoting a sense of simplicity, associated with a rational/formal attitude (suggests trustworthy) (Blake, 2013); (iii) non-serif typography (suggests clarity, trustworthy) mainly vertical (elegance; for the senior user, semi-bold or bold options considered); (iv) iconic representation use in buttons, easier to identify and understand (minimal and slick shapes for adults; rounded forms for children).

One of the major aspects that contribute to the functionality and aesthetic value is typography. Type legibility and readability, related to light and color contrast are essential to a functional (Bringhurst, 1997) as well as aesthetic UI design, adequate to the visitor segment. Sans-serif fonts promote legibility and readability particularly in small screens (Cooper et al, 2014). Moreover, letter size, weight, line length, leading, number of lines per column are adjusted to visitor type (for seniors and children size is bigger, weight is heavier, contrast is more intense and consequently fewer words per line of text).

When in museum exhibition space and piece identification mode, the UI gives visual continuity of the user's physical surroundings, framing it and enhancing it with AR information presented with a transparent background (applying flow and transparency concepts): when the user points to each major piece, hotspots are visualized on screen; they give access to tagged detailed piece related information in different media (text about author, piece, context; images, audio and video related information; suggested pieces and map location, among other; contents adjusted to visitor type) and to gaming options (quizzes and activities are piece related and adjusted to each visitor type).

Besides the graphic aspects, the APP UI also gives access to piece related audio information, tagged to major pieces. This promotes the emotional experience creating emotional ambiences with high emotional expressivity (Douek, 2013) during the museum visit. The sounds connected to each piece also explore the aspect of spatialization of sound (Begault, 2000), allowing the user to relate spatially to the piece by feeling closer or farther to a particular sound, while being closer or farther to the piece it related to. In big pieces, different parts of the piece would have different sounds, and they would mix into a sound ambience, which would gradually change depending on visitor's position.

Adaptive Card Implementation

In general, a mobile APP has several *views,* many with equal structure and layouts only changing the displayed contents. On the other hand, in a fully adaptive APP there could appear a *view* with a different structure and different contents for each user, leading to an AUI. Regardless of whether or not it is a fully AUI, the layout and structure of a *view* can be used in several "views", e.g., when looking at the products details in a shopping APP, the structure of the *views* are identical only changing the contents to match the product descriptions. Moreover, different users can be grouped such that each group uses the same layout, the same structure, and the same contents.

Given these, if each *view* is built from scratch then there will be repeated information in the application regarding the *view's* structure. Besides, building a *view* from scratch each time it is needed would be extremely inefficient in terms of CPU usage. If a structure of a *view* is to be used in different moments of the execution, why not keep it in memory and simply access it and change the contents according to the *view*'s requirements (creating a modular structure)? Keeping this in mind, the adaptive cards were split into: (i) structure and (ii) contents. The application will create the structure of a *view* and keep it in memory, where it will be used as a *template*. When the display of a *view* is requested, the APP will access this *template* and place the *view*'s contents.

To assemble the (i) cards' structure, a *Template Engine* was designed based on a tree data structure, where each tree's node can be classified as a *content format*, a *cell*, or a *template*. The basic node is the *content format*, forming *cells* when grouped with others. In turn, *cells* can be grouped again into a different *cell* or into a *template*. Therefore, every structure is made up of one or more *cells* and zero or more *templates*. Simpler *templates* can be made and then used/grouped in more complex *templates*, making the engine more efficient. Figure 1 bottom shows a wireframe of a menu *view,* and in the top the wireframe of a *template*; see also (Rodrigues et al., 2017).

Let us further explore these concepts by addressing the example in Figure 1, showing the structure and layout for a *menu template*. The image shows a wireframe of a *view,* where every dashed box represents a node and the number in the top right corner its identifier. As seen, a tree data structure was chosen to store the *view*'s layout since it easily represents our "parent – child" relationship between the (a) *content formats*, (b) *cells*, and (c) *templates*, which are the tree's nodes, as explained next. The *content format* (a) is the most basic node structure and configures the format of the content which can be a text, an image, a button, amongst others. Each content type has its own properties. The example shows four content types: a *button* in node 2, an *image* in node 3, a *spacer* in node 5 and a *text content format* in node 4. Concerning this last one, its properties include the font that should be used for the text (e.g., Arial, Times New Roman, etc.), the text color, the font size, etc. Also note that the *button's content format* contains only the "clicked" behavior (e.g., opening a new *view*) and needs a child to present the user with visual information. This child node can be of any of the mentioned categories.

The *cell* (b), expressed as a (blue) dashed box in node 1, is a node whose sole purpose is to organize its children nodes into a single line that can be oriented horizontally or vertically. A *cell* child can be of any of the three mentioned categories. A *cell* does not relay any new information to the user, being instead used to arrange nodes. In our example, Figure 1 top row, there is a single cell (node 1), that orders nodes 2, 4 and 5 into their respective positions. As a final consideration, node 5 is a *content format,* more specifically a *spacer,* and it can be seen as an invisible image used to give the proper spacing to the *cell* components. Common in both *content formats* and in *cells* are the dimension properties that establishes the size and responsiveness behavior for each node.

A *template* (c) node is a particular node that represents another preexisting template, e.g., it can represent the template menu described above. Often, different structures have common parts between them, making the *template* nodes especially helpful in these situations. This is another point where the template engine can decrease the system resource usage. Instead of creating that common part every single time, it is only created once and its contents changed (here contents represent the actual contents, like the "go back" image in node 3 and the "MUSEUM NAME" text in node 4).

In a tree data structure, there are two terms that must be highlighted: the *root* is the origin node from which all the other nodes must have some sort of relationship; and the *leaf* (or terminal) nodes which are the nodes that have no children (SmartClasses, 2016). When constructing the tree structure of a view two rules can be defined. Firstly, a *leaf* node cannot be a *cell*. As stated above, a *cell*'s goal "is to organize its children nodes". If a *cell* is placed at a *leaf* node then it does not have any child, presenting unnecessary data that needs to be transmitted and treated. Secondly, the root node cannot be a *content format* because a tree structure should represent an organized and well-defined structure of a set of contents. In addition, the root node cannot be a *template* node, if it were then it would be a copy of the referenced template. Therefore, the *root* node can only be a *cell* node. Again, this makes sense, since the sole purpose of a *cell* is to layout its children nodes in an organized manner.

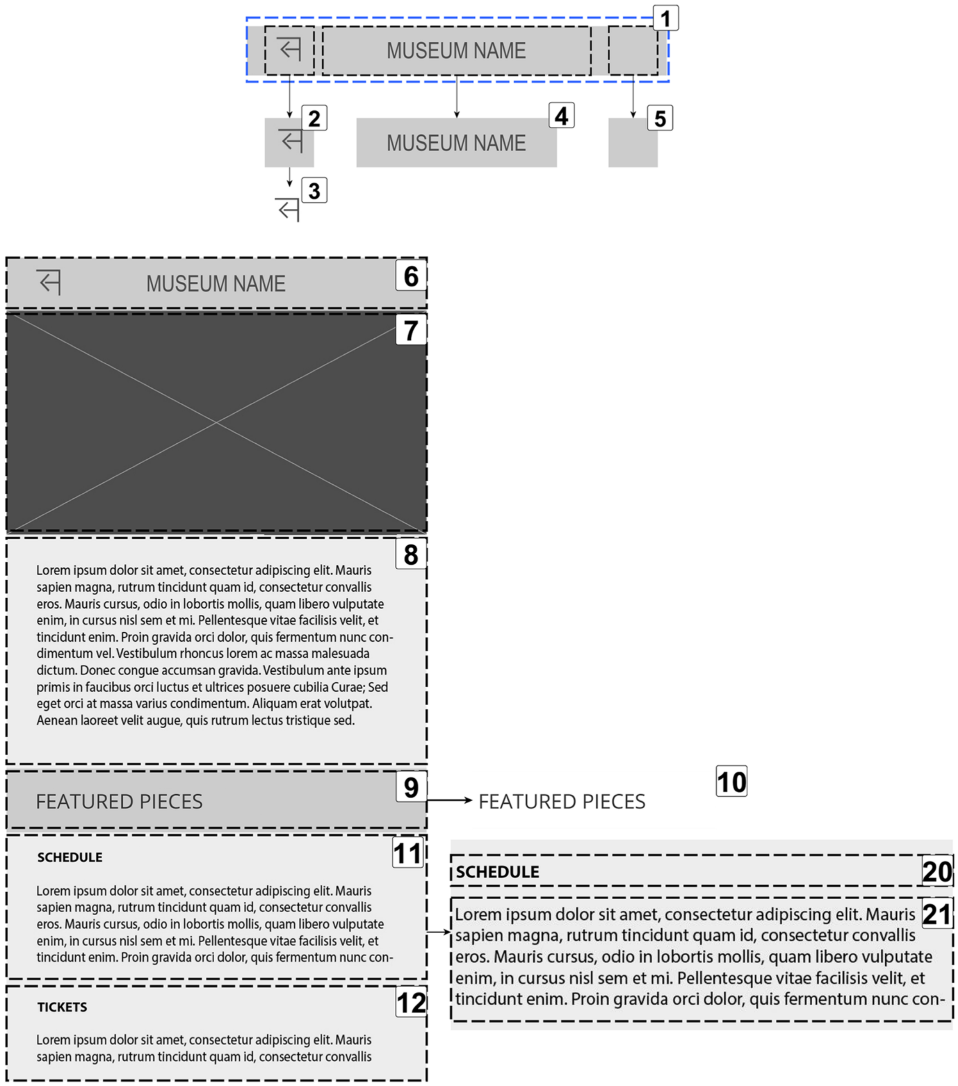
The *Template Engine* uses a depth-first approach when building the templates. In this approach, the engine performs in a recursive manner, starting at the root node and doing the following steps: (1) create the node and define its attributes; (2) for each child process node (executing step 1); and (3) state the parent-child relationships between the nodes.

As mentioned, Figure 1 bottom left shows a template of a *view* that exhibits a *museum information sheet*. In this example, the engine starts in the *root* node by defining that it is a *cell* which is divided vertically. It then goes to the first child (node 6) which, in this case, is the menu *template* detailed in Figure 1 top row (the children are represented by the dashed line). Since this node is a *template,* the engine requests it from memory if it is already instantiated. Assuming it is not, then it proceeds to assemble the menu template. To assemble the menu *template*, the engine starts by creating a horizontal *cell* (a *cell* that is divided horizontally) and defining its size. Next, it goes to the first child, the *button's content format* at node 2 and sets its action (to return to the previous view). Then it creates the *image's content format* (node 3), which comprises the aspect ratio and crop function. After that, it establishes that node 3 is a child of node 2 and node 2 a child of node 1. This finishes the work on the first child of node 1. Afterwards, the engine moves to the second child (node 4), generates the *text's content format*, assigns the correct font formatting and parents it to node 1. Next, node 4 is processed, it is a *spacer* which only contains the dimensions that it should occupy. As expected, the node 4 is set as a child of node 1. This menu template can now be used at any time by any other template that requires it. Following this, the engine now sets this entire (menu) template as a child of the root element for the *museum information sheet* and moves on to the second child (node 7).

The engine work from here on is very similar. Node 7 is a simple *image content format*, node 8 is a *text content format*, node 9 is a *button* that contains a *text* (node 10). The next two nodes (nodes 11 and 12) are a *title* and a *description template*. This template is composed of a vertical *cell* (not indicated in the image) that includes a couple of *text content format,* where the first (node 20) has bold font weight (that can be used as a title), and the second (node 21) has a regular weight (for the body of the text).

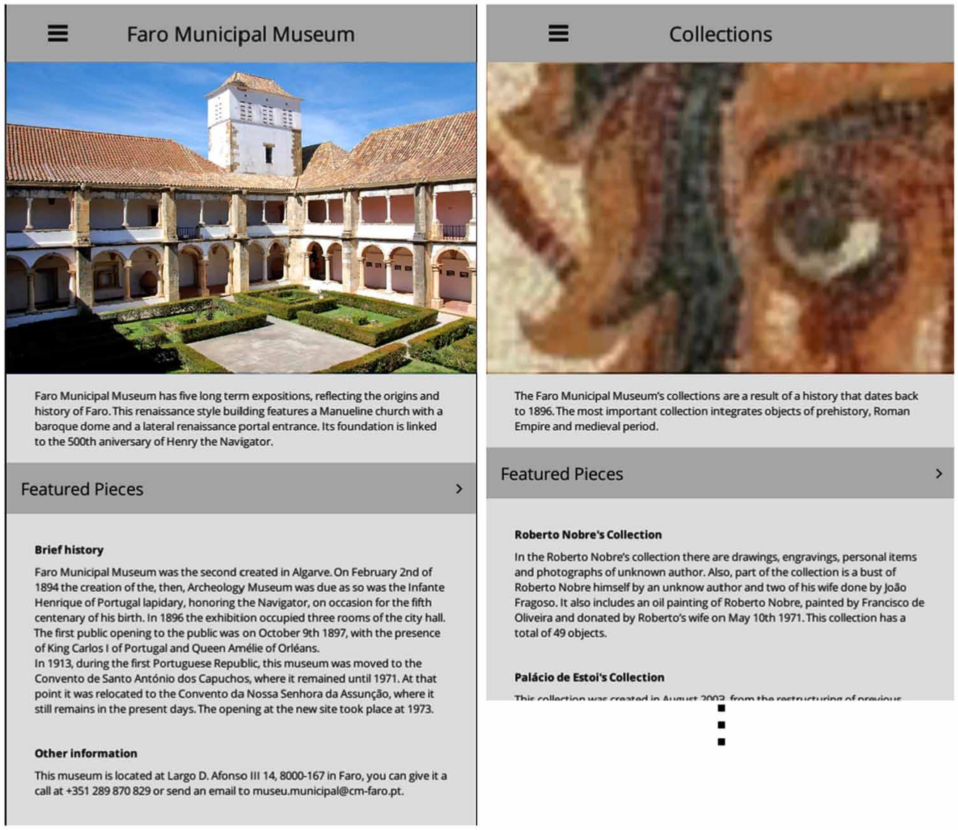
Concerning that the *Contents Assembly* (ii) of a *view* should adapt to the different users and different languages, the authors decided to divide the contents in 3 distinct categories: (a) *static contents*, contents that are independent of both the user and the language (e.g., icons); (b) *localized contents*, that are only based on the language (e.g., the "loading" fixed text); (c) *fully adaptive*, contents that change according to the user type (child, expert, …) and the language.

Figure 1. Template for a menu (top) and a museum information sheet (bottom)



To get the contents for a *view*, the application accesses a database (see next section) which returns the required contents. If contents for two or more *views* are requested, then it is likely that there will be some duplicated information. For example, in Figure 2 both *views* have in common (besides others) a "menu" icon (the three horizontal bars in the top left corner) and a "featured pieces" text. Thus, with memory and CPU optimization in mind, an algorithm was developed to avoid downloading duplicate contents from the server, preserving also network bandwidth. The algorithm produces two separate lists: (a) the contents themselves and (b) a lookup table (LUT) to identify which content belongs to which node in the template (recall that every template is constructed from a tree of nodes).

Figure 2. Two views using the information sheet template shown before



In an application where everything is built from scratch, the structure (template) and contents of a view are closely linked together. With the authors' approach, the contents are not directly coupled to their respective structure but are instead saved in separate tables and linked to their correct location through the lookup table.

Adaptive Card Database

The database (DB) plays a fundamental role in the system, by storing the contents and the harvested user's information. The later information is employed to meet the desired specifications of the user's card-layouts (structure/template) and card-contents. The card-layout implementation in the DB follows the same tree architecture that was presented before and it is subdivided in three major layers: (a) components, (b) formats, and (c) structure.

Basic properties like colors, fonts, outlines, backgrounds, and shadows are defined in the components layer (a). The type of content used in child nodes is indicated by the formats layer (b), through previously created sets of component properties, whether it is a text, an image or a button. Specific properties can also be overridden here if necessary. Finally, the parent-child relationships of the tree architecture are defined in the structure layer (c), node by node, where each node has a unique identifier. Orientation, spacing and other layout and cell information, are also stored here. All these specifications can be aggregated to form templates, so they may be reused when designing a new view, simplifying the process. Regarding the contents for a *view*, they are saved separately from the structure and stored in two lists: (a) the contents (themselves) list and a lookup table to bond them to the template. (b) The content categories, defined by the authors (static contents, localized contents and fully adaptive), are each kept on separate tables, one per category. Concerning the linkage between the contents and the structure, there is a lookup table for each view that associates every node of the used template to an entry in one of the contents table.

Before the App can request a specific *view*, it first needs the template structure and its contents, retrieved in a JSON document format. To support the retrieval of the JSON documents, three individual scripts are running on the server side: (1) the structure JSON generation script, (2) the structure JSON retrieval script, and (3) the content JSON retrieval script. When a new structure template is created, the structure JSON generation script (1) will receive the *template* index, connect to the DB and navigate from table to table, node by node, building up the tree architecture in a JSON format. The script then stores the converted file in the DB with a timestamp, so that if the App detects a newer version for that template, it can simply request the new JSON document. This script runs prior to the App's execution, when a new template has been added to the DB, or when an update was made to an existing template. After the structure template is converted to JSON and stored, it can be requested at any time. For this purpose, the server is running the structure JSON retrieval script (2). This script also determines if that particular template requires other pre-existing templates and includes them in the final assembled JSON to be sent, guaranteeing that every property needed by that template is included, in an organized fashion, inside a single file.

After building the templates, the application needs to fetch the contents to complete the view. To this end, a content JSON retrieval script (3) was developed that returns a (i) set of contents and an (ii) association list. The authors decided to keep this separation to avoid having to process duplicate contents on the mobile device. For each view, the script generates both lists (the set of contents (i) and the association list (ii)) joining the LUT (what contents the view needs) with the contents tables (the contents themselves) resulting in a set of contents (i). The association list (ii) is a list of contents indexes, from list (i), sorted by appearance order of that content in the template. Finally, after carrying out this process on all views, all the resulting lists are merged together, removing duplicate entries on the contents set (i) and updating the association list (ii) to match the final result set. It is important to stress, that this script approach was selected because Unity does not contain an appropriate multi-platform method for accessing a database. This way the application communicates directly with the server, through the retrieval scripts, downloading the necessary information via JSON formats.

Augmented Reality Implementation

In the present application, there is at least one card *view* for each object (paintings, statues, …) in the museum, which appears in the mobile screen when the object is present, i.e., if the object is on the field-of-view of the camera. In this section, the authors focus is on object detection, recognition and tracking, to trigger the card *view* of the object and/or the AR-card associated with the object. It is important to stress that two groups of cards can appear: the ones that occupy the complete screen of the mobile device, and the AR-cards that floats in the mobile screen at the same time we see the scene/object (usually non-overlaying the object).

From the various solutions to use AR (see the Background section), the one chosen for this APP was the usually denominated as "marker-based", which in a simplified way are digital pictures of the original object that work as *templates* – markers. In the present work, objects are photographed in Full HD and stored in the system's database. Resorting to Computer Vision algorithms, the frames captured by the mobile camera and markers are compared. When an object is recognized, the corresponding identifier is triggered and the object's position on the mobile screen is returned. Here, the focus is on a solution with three goals: (a) decrease the time required to download the markers onto the mobile device, (b) perform the marker recognition process in the mobile device, instead of using a server, and (c) try to be as less consuming as possible in terms of power and memory when performing the recognition. To achieve these goals, two options were considered: (A) an implementation developed by the authors, and (B) an already existing multi-platform open source library - ARToolKit (Artoolkit, 2016). The last, is still in active development and with an active community, providing a Unity package.

The authors' marker recognition (object recognition) implementation requires a reliable and fast descriptor since it is projected to run on a mobile device. For that reason, the authors' implementation (A) uses the ORB descriptor (Rublee et al., 2011) for object recognition, as it is a good alternative to other more known descriptors, such as BRIEF, SIFT or SURF (Figat et al., 2014). Although the detailed explanation of the algorithm is available in (Rodrigues et al., 2017), a brief explanation is presented here. It is important to stress that this algorithm is still in development, changes can occur in the near future.

A *Patch* is defined as a parcel of the initial image, of size $N \times M$ pixels (px); e.g. each single square in Figure 3 top, 2nd column. The algorithm works in the following way (system's server side): (i) On the *template* image. (i.1) Compute the ORB descriptor; (i.2) Separate the *template* image in patches, and obtain patches containing keypoints and corresponding descriptors (Rublee et al., 2011); (i.3) Order patches by keypoint/descriptor relevance, and select the K most meaningful extracted patches - those are to be used as *marker patches* ($K \leq 5$; ordered based on the number of keypoints for each patch); (i.4) repeat steps (i.1) to (i.3), using the *template* image resized to 1/2 and to a 1/4 of the original size (resulting in 3 scales; this allows farther and shorter validation distances when targeting the mobile camera onto an object). (i.5) Assemble and order the patches from all different scales, totalizing the number of marker patches per template, r.

(ii) When performing object recognition and tracking (system's client/mobile side): (ii.1) Apply the ORB descriptor to each frame acquired from the mobile camera (Figure 3 top, 3rd column); (ii.2) Choose the *most relevant marker patch* from the 3 scales grouping to test the frame (see step i.5); (ii.3) Select the best classified image template; (ii.4) Using every patch from the correspondent template test the object recognition performing "template-frame" matching; (ii.5) If ratio validation threshold is met then the object is recognized (Figure 3 top, 4th column); (ii.6) Flag if an object is found. Beyond this point the object is only tracked (no recognition tests are performed). (ii.7) Perform object tracking based only on a valid marker patch from the selected *template* (a match between the frame and a single valid marker patch occurs, these steps will continue being performed until the object disappears from the field-of-view of the camera for more than 1 second); (ii.8) If tracking time threshold is reached the recognition process restarts (steps ii.1 to ii.6).

On the other hand, as of May 2016, (B) ARToolKit (2016) started using FREAK as a descriptor for object recognition (Alahi, Ortiz, & Vandergheynst, 2012), and the SURF operator for object tracking (Figat et al., 2014). To create an image recognition application, ARToolKit uses two separate procedures. First, the images (marker templates) must be processed and the FREAK keypoints extracted (it is important to stress that at the time this chapter was written, there are some inconsistencies in ARtoolKit

Figure 3. Top, authors object recognition: left to right: marker template (low res.), original image divided in patches, 3 most relevant patches from the full-size image, the 1/2 and from 1/4 size image, and object matched. Bottom, ARToolKit object detection. Left to right, FREAK descriptor in 2 scales (half and full), and object recognition (all images presented were resized)

information about which descriptor is used for each procedure). This is an automated process done by a desktop application. For each image scale, it extracts two sets of features, one for tracking and another for recognition. The number of keypoints calculated depends on the image size (in a test image, it calculates around 3.000), however the keypoints kept varies according to the extraction level selected which ranges from a level 0 (in the same image, saved 13 keypoints) to a level 4 (saved 2.754 keypoints). ARToolKit requires that at least 4 keypoints are visible at all time to allow for six degrees of freedom (6DoF), i.e., to allow the marker projection to be invariant to rotations, translations and scale operations. The number of scales results from the dpi (dots per inch) range selected. The ARToolKit feature set generation script defines the number of scales as a result from the difference between the minimum and maximum dpi selected, multiplied by a constant. For example, when reducing the original image from 72 dpi (with 1000×2000 px) to 36 dpi, the resolution will be 1/2 (500×1000 px). In the mentioned example, choosing a minimum dpi value of 36 and a maximum of 72, the generation program returns four different scales (72, 57, 45 and 36). The extracted data is saved into a file and imported to the application. In turn, the application loads the markers onto memory and starts the recognition procedure.

In the present state of development and experimenting, two tests were considered: (a) object recognition at different distances, from 1 to 5 m (meters), and the (b) time necessary to load the templates into mobile memory, to recognize and track the objects. In both approaches (authors and ARToolKit) the tests were performed with the same datasets.

In the first case (a), an extra 1/3 of distractors (templates that should not be detected) were added. Both approaches identified correctly almost all targeted objects at 1m distance with good illumination, substantially decreasing the recognition rate at 5m distance. Another set of tests were also done with different lighting on the objects, so the authors could understand the effects of artificial and natural light exposure. As expected, these results were much worse than the previous. Images still would be detectable up to 1 meter for both solutions, while for farther distances, up to 5 meters, they would either not be detectable or very unstable. In this test, no substantial difference was found between the authors and ARToolKit approaches. The second case (b) tested the load time to memory for 5, 10, 20 and 50 templates and their respective times to recognize and track the objects. Each object-template recognition and tracking was averaged with 50 samples. Regarding the loading of the templates to memory, the authors approach was in average 2 to 4 times faster than ARToolKit. For recognition, it was not possible to obtain the times for ARToolKit. Last but not least, ARToolKit tracking was 20 times faster than the authors' approach (ARToolKit did not work with 50 templates), nevertheless both approaches work in real time (see also section Future Research Direction).

In conclusion, the authors approach shows promising results when compared to ARToolKit, since it can do the recognition and tracking in real time, and is still faster than ARToolKit when loading templates to memory, howbeit this last factor still needs great improvements.

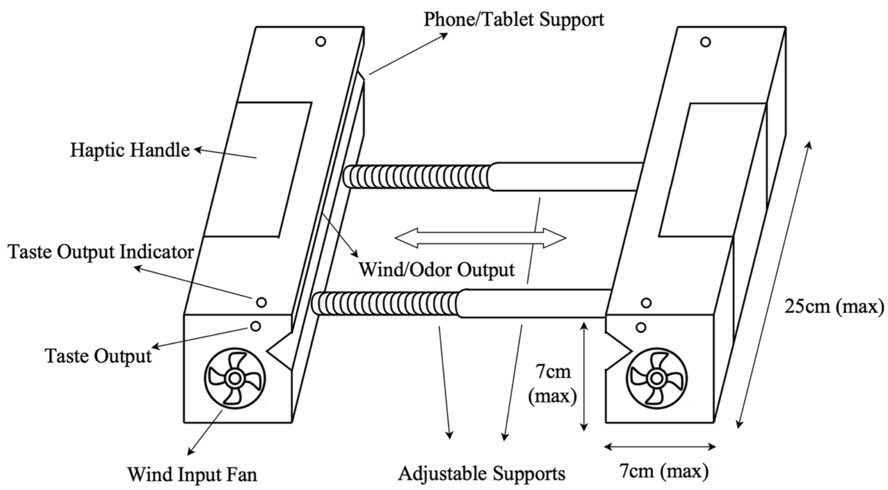
Augmented Reality Device Implementation

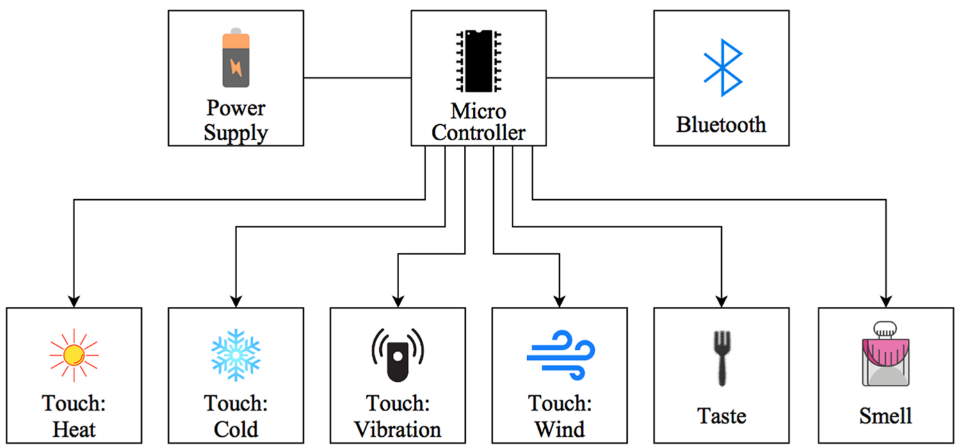
The second visible component of the system is the augmented reality device (HDevice), which extends the users' experiences to the five senses. The first challenge was to integrate the visitors' smart devices (tablets/phablets/smartphones) in a compact new device that allows the user to have these five sense experiences. Hence, the device must be portable but flexible enough to adapt itself to the different sizes of the user's smart device. Moreover, the second challenge was to integrate, in a portable and small device, all the hardware needed to allow the five senses experiences. Figure 4 top row shows a sketch of the proposed portable device. The system will consist of two similar hardware parts, placed on each side of the user's device where the mobile application is running, connected in the bottom through adjustable supports.

The device is powered with a rechargeable battery, which gives the module the ability to keep the system running during the museum visiting (the minimum time requirement considered is a couple of hours). As depicted in Figure 4 bottom row, a microcontroller is the core unit of the device. The microcontroller receives instructions from the mobile application and acts accordingly, controlling the remaining hardware of the portable device to grant the five-sense experience to the user. The communication between the device and the mobile application is possible through a wireless connection, using a Bluetooth interface. The communication with the remaining modules, denoted here as physical interfaces, will be wired. These physical interfaces are responsible for creating the stimuli that reproduce three senses: touch, taste, and smell. The remaining senses, sight and hearing (but also touch, although only vibrations), are reproduced in the user's mobile device.

Three techniques are being used for the touch sense: hot/cold touch, vibration and wind. The heat and cold is generated using thermoelectric modules, such as Peltier modules, able to reproduce a certain temperature range, safe for the user, yet still sufficiently noticeable to convey the intended sensation. The heat/cold will be conducted through the metallic handles of the HDevice, thus transmitting the heat/cold sensation to user's hands. The vibration is obtained with two vibration motors, one on each side of the device. This is the same common solution used in every smartphone device, to vibrate during a call, and used in the haptic feedback in gaming. By having two independent motors, it also enables the use on transmitting directions (left, right) to the user and help to navigate in the museum. The wind sensa-

Figure 4. Top, a schematic of the portable device – Hdevice; bottom, block diagram of basic functionalities of the HDevice





tion is obtained using a ventilation system with four fans, two in each side of the device. As proposed in (Matsukura et al., 2013), the cooperative work of these four fans can cause a sense of wind or a breeze directed to the user's face and, by varying the speed of each one individually, can generate airflow in different directions to the user.

Regarding the taste sense, the solution was to incorporate an electronic vaporizer to give the user the sensation of savoring some food and tasting different flavors. The device is being developed with four electronic vaporizers, two on each side of the HDevice, thus providing four different flavors. In order to guarantee the hygiene of the system, each visitor should receive with the HDevice a disposable tube, that will serve to connect to the outlet of the flavor and at the other end will have a nozzle to savor flavors. For the smell sensation, the first prototype of the device reuses the electronic vaporizer and available fans to help spread the fragrances in the air. This is not a unique solution, as other possibilities were also considered: air circulation per flavored container, fragrance spraying, pressurized spray, and ultrasonic atomization. However, it is important to keep the device with small dimensions, and this option provides a fair smell sensation with a small area and power overhead.

The first prototype of the HDevice is in developing/testing and further details and results will be published in the future (for more details see the M5SAR's website).

FUTURE RESEARCH DIRECTIONS

This chapter presented an initial version of the framework for a mobile prototype that can extent AR to the five senses. This is now an emerging subject, see e.g. AxonAR (2016), and a large amount of systems/solutions are expected to appear in the future. Developing solutions for the touch sense are expected to be easier than other, like the smell. Likewise, extending the AR to the taste will be for certain a huge challenge, even more if the intention is to create mobile devices that can simultaneously augment all five senses.

Different, but also very challenging subject is the full adaptive ("intelligent") UI. This subject has been researched for a few years. Nevertheless, as far as the authors know, still there is not any real adaptive APP in the market. As mentioned in the Introduction section, to harvest the necessary information about each user's preferences and skills is difficult, but with recent Big Data methodologies this starts to be realizable. Then, there is the question on how to give "intelligence" to the UI in order for it to adapt on-the-fly to the user's changes, i.e., how to develop an APP that can "satisfy" the user (Zhao et al., 2012). A final question is how to develop the adaptive UI. Although we present an initial framework to create this adaptive UI others exist, e.g. (Gajos et al., 2008). Finally, the communication between the server and the mobile, in terms of UI card building and AR templates, are being adapted from JSON to FlatBuffers (FlatBuffers, 2017). Initial tests show that this change drastically reduces times related to the marker, UI preparation.

A future emerging trend will be for sure the extension of Augmented Reality to the five senses, and the challenge of creating mobile AR five senses devices that can be applied between other areas to the "Cultural Heritage and eTourism Applications".

CONCLUSION

This chapter presented an initial framework for the development of an architecture capable of producing an adaptive UI (for a museum). The focus was on the creation process of a card-based UI, where the development of the cards has a modular architecture. In addition, a patch-based marker architecture for mobile object recognition with application to AR was also discussed. Finally, it was also presented an initial architecture for a device (HDevice) that can be connected to the user's mobile device to extent the AR to the five senses.

Despite all the modules of the system being yet in an initial stage of development, both AUI and AR marker present satisfactory results. A first prototype of HDevice is now being tested. All the modules of the system were being integrated as the chapter was written. It is important to stress, that more detailed figures and information cannot be shown in the present, once this system is being developed with a commercial company.

For future developments, the authors will focus on how to harvest the necessary information about each user preferences and skills and, from the acquired information/data, how to give "intelligence" to the UI to adapt on-the-fly to the user's changes. In the case of the mobile object recognition system, it is intended to increase the number of real time recognition of different objects on-the-fly, without any necessary connection to a server. Finally, the integration of all elements of the prototype to start real test scenarios in a museum with real persons and apply the UTAUT2 model to validate the technology will be carried out.

ACKNOWLEDGMENT

This work was supported by the Portuguese Foundation for Science and Technology (FCT), project LARSyS (UID/EEA/50009/2013), CEFAGE (PEst-C/EGE/UI4007/2013), CIAC, CIAUD and project M5SAR I&DT nr. 3322 financed by CRESC ALGARVE2020, PORTUGAL2020 and FEDER. We also thank Faro Municipal Museum and the project leader SPIC - Creative Solutions (www.spic.pt).

REFERENCES

Adobe. (2016). *XD Essentials: Card-based user interfaces*. Retrieved: Nov. 16, 2016 from https://goo.gl/gg8qUM

Akiki, P. A., Bandara, A. K., & Yu, Y. (2014). Adaptive model-driven user interface development systems. *ACM Computing Surveys*, *47*(1), 9. doi:10.1145/2597999

Alahi, A., Ortiz, R., & Vandergheynst, P. (2012). *Freak: Fast retina keypoint*. In *IEEE Conference on Computer Vision and Pattern Recognition*, 212 (pp. 510-517). IEEE.

Alvarez-Cortes, V., Zayas, B. E., Uresti, J. A. R., & Zarate, V. H. (2009). *Current Challenges and Applications for Adaptive User Interfaces*. INTECH Open Access Publisher. doi:10.5772/7745

Artoolkit. (2016). *ARtoolKit, The world's most widely used tracking library for augmented reality*. Retrieved: Nov. 16, 2016 from http://artoolkit.org/

Axon, V. R. (2016). *AxonVR.* Retrieved: Dec. 6, 2016 from http://axonvr.com

Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., & MacIntyre, B. (2001). Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, *21*(6), 34–47. doi:10.1109/38.963459

Babich, N. (2016). *Designing Card-Based User Interfaces, Smashing Magazine*. Retrieved: Nov. 16, 2016 from https://goo.gl/AM46gT

Balance. (2016). *The balance: Fine Art Museum Apps for Smart Phones*. Retrieved: Nov. 16, 2016 from https://goo.gl/WEsQRN

Begault, D. R. (2000). *3-D sound for Virtual reality and Multimedia.* NASA Center for AeroSpace Information. NASA/TM – 2000-209606.

Blake, J. (2013). *Natural user interface in. Net*. Greenwich, CT: Manning.

Boehner, K., DePaula, R., Dourish, P., & Sengers, P. (2007). How emotion is made and measured. *International Journal of Human-Computer Studies*, *65*(4), 275–291. doi:10.1016/j.ijhcs.2006.11.016

Brewster, S., Chohan, F., & Brown, L. (2007). Tactile feedback for mobile interactions. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 159-162). San Jose, CA: ACM. doi:10.1145/1240624.1240649

Bringhurst, R. (1997). *The Elements of Typographic Style*. Hartley & Marks.

Catchoom. (2016). Retrieved: Nov. 16, 2016 from http://catchoom.com/

CEUD. (2016). Retrieved: Nov. 16, 2016 from http://goo.gl/fUHPj6

Cheng, K. H., & Tsai, C. C. (2013). Affordances of augmented reality in science learning: Suggestions for future research. *Journal of Science Education and Technology*, *22*(4), 449–462. doi:10.1007/s10956-012-9405-9

Conati, C., Carenini, G., Toker, D., & Lallé, S. (2015). *Towards User-Adaptive Information Visualization*. AAAI.

Cooper, A., Reimann, R., Cronin, D., & Noessel, C. (2014). *About the face: The essentials of interaction design*. Indianapolis: Wiley.

Douek, J. (2013). Music and emotion: A composers perspective. *Frontiers in Systems Neuroscience*, *7*, 82. doi:10.3389/fnsys.2013.00082 PMID:24348344

Estimote. (2016). *Estimote: Create magical experiences in the physical world*. Retrieved: Nov. 16, 2016 from http://estimote.com/

EWB. (2016). *Europe Without Barriers*. Retrieved: Nov. 16, 2016 from http://goo.gl/sqlJrE

Falk, J. H., & Dierking, L. D. (2013). *The Museum Experience Revisited*. Walnut Creek: Left Coast Press.

FeelReal. (2016). Retrieved: Dec. 6, 2016 from http://feelreal.com/

Figat, J., Kornuta, T., & Kasprzak, W. (2014). Performance evaluation of binary descriptors of local features. In *Proceedings International Conference on Computer Vision and Graphics* (pp. 187-194). Springer International Publishing. doi:10.1007/978-3-319-11331-9_23

FlatBuffers. (2017). Retrieved: Jan. 6, 2017 from https://google.github.io/flatbuffers/

Gajos, K., & Weld, D. S. (2004). SUPPLE: automatically generating user interfaces. In *Proceedings of the 9th international conference on Intelligent user interfaces* (pp. 93-100). ACM.

Gajos, K. Z., Wobbrock, J. O., & Weld, D. S. (2008). Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 1257-1266). ACM. doi:10.1145/1357054.1357250

Gnome. (2016). *User Interface Guidelines for Supporting Accessibility*. Retrieved: Nov. 16, 2016 from https://goo.gl/xGAH6f

HMS. (2016). *Srbija 1914 / Augmented Reality Exhibition at Historical Museum of Serbia, Belgrade*. Retrieved: Nov. 16, 2016 from https://vimeo.com/126699550

InformationWeek. (2016). *InformationWeek:10 Fantastic iPhone, Android Apps For Museum Visits*. Retrieved: Nov. 16, 2016 from https://goo.gl/rYnhm2

Jung, T., Chung, N., & Leue, M. C. (2015). The determinants of recommendations to use augmented reality technologies: The case of a Korean theme park. *Tourism Management*, *49*, 75–86. doi:10.1016/j.tourman.2015.02.013

Karaman, S., Bagdanov, A. D., Landucci, L., DAmico, G., Ferracani, A., Pezzatini, D., & Del Bimbo, A. (2016). Personalized multimedia content delivery on an interactive table by passive observation of museum visitors. *Multimedia Tools and Applications*, *75*(7), 3787–3811. doi:10.1007/s11042-014-2192-y

Kovács, P. T., Rozinaj, G., Murray, N., Sulema, Y., & Rybárová, R. (2015). *Application of immersive technologies for education: state of the art*. Paper presented at the 2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL), Thessaloniki, Greece. doi:10.1109/IMCTL.2015.7359604

Kralisch, A., Eisend, M., & Berendt, B. (2005). Impact of culture on website navigation behaviour. *Proceeding of the 11th International Conference on Human-Computer Interaction*.

Kudan. (2016). *Kudan Computer Vision*. Retrieved: Nov. 16, 2016 from https://www.kudan.eu/

Kurogi, T., Nakayama, M., Sato, K., Kamuro, S., Fernando, C. L., Furukawa, M., . . . Tachi, S. (2013). Haptic transmission system to recognize differences in surface textures of objects for telexistence. In 2013 IEEE Virtual Reality (VR) (pp. 137-138). IEEE. doi:10.1109/VR.2013.6549400

Layar. (2016). Retrieved: Nov. 16, 2016 from https://www.layar.com/

Li, Y. (2000). Geographical consciousness and tourism experience. *Annals of Tourism Research*, *27*(4), 863–883. doi:10.1016/S0160-7383(99)00112-7

Lindgaard, G., Fernandes, G., Dudek, C., & Brown, J. (2006). Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & Information Technology*, *25*(2), 115–126. doi:10.1080/01449290500330448

Marangunić, N., & Granić, A. (2015). Technology acceptance model: A literature review from 1986 to 2013. *Universal Access in the Information Society*, *14*(1), 81–95. doi:10.1007/s10209-014-0348-1

Matsukura, H., Yoneda, T., & Ishida, H. (2013). Smelling screen: Development and evaluation of an olfactory display system for presenting a virtual odor source. *IEEE Transactions on Visualization and Computer Graphics*, *19*(4), 606–615. doi:10.1109/TVCG.2013.40 PMID:23428445

Metallinou, A., Wollmer, M., Katsamanis, A., Eyben, F., Schuller, B., & Narayanan, S. (2012). Context-sensitive learning for enhanced audiovisual emotion classification. *IEEE Transactions on Affective Computing*, *3*(2), 184–198. doi:10.1109/T-AFFC.2011.40

Morency, L. P., Mihalcea, R., & Doshi, P. (2011). Towards multimodal sentiment analysis: Harvesting opinions from the web. In *Proceedings of the 13th International Conference on Multimodal Interfaces* (pp. 169-176). ACM. doi:10.1145/2070481.2070509

MWCM. (2016). *Multitouch wall Cleveland Museum*. Retrieved: Nov. 16, 2016 from https://goo.gl/kI2oCh

Narumi, T., Kajinami, T., Nishizaka, S., Tanikawa, T., & Hirose, M. (2011). Pseudo-gustatory display system based on cross-modal integration of vision, olfaction and gustation. In *Virtual Reality Conference (VR)*, 2011 *IEEE* (pp. 127-130). IEEE. doi:10.1109/VR.2011.5759450

Negruşa, A. L., Toader, V., Sofică, A., Tutunea, M. F., & Rus, R. V. (2015). Exploring gamification techniques and applications for sustainable tourism. *Sustainability*, *7*(8), 11160–11189. doi:10.3390/su70811160

Oh, H., Fiore, A. M., & Jeoung, M. (2007). Measuring experience economy concepts: Tourism applications. *Journal of Travel Research*, *46*(2), 119–132. doi:10.1177/0047287507304039

Pine, J., & Gilmore, J. (1999). *The experience economy*. Boston: Harvard Business School Press.

Qualcomm. (2016). *Invisible Museum*. Retrieved: Nov. 16, 2016 from https://goo.gl/aS0NKh

Ranasinghe, N., Cheok, A. D., Fernando, O. N. N., Nii, H., & Ponnampalam, G. (2011). Electronic taste stimulation. In *Proceedings of the 13th international conference on Ubiquitous computing* (pp. 561-562). ACM.

Ranasinghe, N., Nakatsu, R., Nii, H., & Gopalakrishnakone, P. (2012). Tongue mounted interface for digitally actuating the sense of taste. In *Wearable Computers (ISWC), 2012 16th International Symposium on* (pp. 80-87). IEEE. doi:10.1109/ISWC.2012.16

Reinecke, K., & Bernstein, A. (2013). Knowing What a User Likes: A Design Science Approach to Interfaces that Automatically Adapt to Culture. *Management Information Systems Quarterly*, *37*(2), 427–453.

Rodrigues, J. M. F., Lessa, J., Gregório, M., Ramos, C., & Cardoso, P. J. S. (2016). *An Initial Framework for a Museum Application for Senior Citizens. In Proceedings Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion* (pp. 414–421). ACM. doi:10.1145/3019943.3020002

Rodrigues, J. M. F., Pereira, J. A. R., Sardo, J. D. P., Freitas, M. A. G., Cardoso, P. J. S., Gomes, M., & Bica, P. (2017) Adaptive Card Design UI Implementation for an Augmented Reality Museum Application. *11th Int. Conf. on Universal Access in Human-Computer Interaction, integrated in the 19th Int. Conf. on Human-Computer Interaction*.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the International Conference on Computer Vision* (pp. 2564-2571). IEEE. doi:10.1109/ICCV.2011.6126544

Saffer, D. (2006). *Designing for Interaction: Creating Innovative Applications and Devices*. Berkley, CA: New Riders.

Schuller, B. W. (2015). Modelling User Affect and Sentiment in Intelligent User Interfaces: A Tutorial Overview. In *Proceedings of the 20th International Conference on Intelligent User Interfaces* (pp. 443-446). ACM. doi:10.1145/2678025.2716265

SM. (2016). *Science Museum - atmosphere gallery*. Retrieved: Nov. 16, 2016 from https://vimeo.com/20789653

SmartClasses. (2016). *Smart Classes: Data Structures*. Retrieved: Nov. 16, 2016 from http://btechsmartclass.com/DS/U3_T1.html

Sodhi, R., Poupyrev, I., Glisson, M., & Israr, A. (2013). AIREAL: Interactive tactile experiences in free air. *ACM Transactions on Graphics*, *32*(4), 134. doi:10.1145/2461912.2462007

Steichen, B., Conati, C., & Carenini, G. (2014). Inferring Visualization Task Properties, User Performance, and User Cognitive Abilities from Eye Gaze Data. *ACM Transactions on Interactive Intelligent Systems*, *4*(2), 11. doi:10.1145/2633043

TWSJ. (2016). *The Wall Street Journal: Best Apps for Visiting Museums*. Retrieved: Nov. 16, 2016 from https://goo.gl/cPTyP9

Unity. (2016). *Unity 3D*. Retrieved: Nov. 16, 2016 from https://unity3d.com/pt

Vainstein, N., Kuflik, T., & Lanir, J. (2016). Towards Using Mobile, Head-Worn Displays in Cultural Heritage: User Requirements and a Research Agenda. In *Proceedings of the 21st International Conference on Intelligent User Interfaces* (pp. 327-331). ACM. doi:10.1145/2856767.2856802

VAM. (2016). *Interactive IPAD Museum Catalog*. Retrieved: Nov. 16, 2016 from https://vimeo.com/31821923

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *Management Information Systems Quarterly*, *25*(3), 425–478.

Venkatesh, V., Thong, J. Y., & Xu, X. (2012). Consumer acceptance and use of information technology: Extending the unified theory of acceptance and use of technology. *Management Information Systems Quarterly*, *36*(1), 157–178.

Vuforia. (2016). *Vuforia*. Retrieved: Nov. 16, 2016 from https://www.vuforia.com/

Wikitude. (2016). *Wikitude, see more*. Retrieved: Nov. 16, 2016 from http://www.wikitude.com/

Yamada, T., Yokoyama, S., Tanikawa, T., Hirota, K., & Hirose, M. (2006). Wearable olfactory display: Using odor in outdoor environment. In *IEEE Virtual Reality Conference (VR 2006)* (pp. 199-206). IEEE. doi:10.1109/VR.2006.147

Yanagida, Y. (2012). A survey of olfactory displays: Making and delivering scents. In Sensors, 2012 IEEE (pp. 1-4). IEEE.

Yuan, Z., Bi, T., Muntean, G. M., & Ghinea, G. (2015). Perceived synchronization of multisemedia services. *IEEE Transactions on Multimedia*, *17*(7), 957–966. doi:10.1109/TMM.2015.2431915

Zhao, L., Lu, Y., Zhang, L., & Chau, P. Y. (2012). Assessing the effects of service quality and justice on customer satisfaction and the continuance intention of mobile value-added services: An empirical test of a multidimensional model. *Decision Support Systems*, *52*(3), 645–656. doi:10.1016/j.dss.2011.10.022

KEY TERMS AND DEFINITIONS

Accessible Tourism: The set of services and facilities that can allow people with specific needs the enjoyment of the holiday and leisure without obstacles and difficulties.

Adaptive User Interfaces: A user interface that adapts to each user, i.e., changes its layout and contents according with the needs of each user.

Augmented Reality: A system that supplements the real world with virtual (computer-generated) objects that appear to coexist in the same space as the real world.

Computer Vision: An interdisciplinary field that deals with how computers can gain high-level understanding from digital images or videos.

Human-Computer Interaction: Researches the design and use of computer technology, focused on the interfaces between humans and computers. Researchers both observe the ways in which humans interact with computers and design technologies that let humans interact with computers in novel ways.

Interaction Design: The practice of designing interactive digital products, environments, systems, and service.

Multi-Sensorial Display: An output device that conveys information for multiple sensory systems.