

[POSTER] Transforming your website to an augmented reality view

Dimitrios Ververidis*

Spiros Nikolopoulos

Ioannis Kompatsiaris, Senior Member, IEEE

Centre for Research and Technology Hellas (CERTH), Greece

Abstract

In this paper we present FastAR, a software component capable of transforming Joomla based websites into AR-channels compatible with the most popular augmented reality browsers (i.e. Junaio, Layar, Wikitude). FastAR exploits the consistency of the data structure across multiple sites that have been developed using the same content management system, so as to automate the transformation process of an internet website to an augmented reality channel. The proposed component abstracts all related programming tasks and significantly reduces the time required to generate and publish AR-content, making the entire process manageable by non-experts. In verifying the usefulness and effectiveness of FastAR, we conducted a survey to solicit the opinion of users who carried out the installation and transformation process.

1 Introduction

Augmented reality is currently one of the most dynamically evolving technologies for engaging users and enhancing their experience. Particularly interesting, are the applications developed for promotional and marketing purposes where the user's view is augmented in a very targeted and intriguing manner with products, services and offers. However, making data compatible with AR browsers is far from trivial since it requires a wide variety of programming skills for web, mobile and database development, as well as experience in using the Application Protocol Interfaces (APIs) offered by the existing AR browsers. This maybe easy for programmers, but it is more than difficult for content owners that ignore the exact procedures and formats for making their content AR-compatible.

On the other hand, the vast majority of enterprizes (SMEs) that sale goods or services and have a strong geo-reference aspect (e.g. the ones advertising a chain of restaurants, of hotels, of super-markets or other types of retailing activities, as well as websites that act as brokers such as TripAdvisor [1], Booking.com, etc) have a web-presence with a site that is typically empowered by one of the existing content management systems (CMS) (e.g. Joomla, Wordpress, etc). The great benefit of having a site empowered by a widely established CMS is that independently of how

the data appears in the internet browser, they are stored and organized in a certain manner, which is enforced by the default core database schema of the underlying CMS. This fact allow us to exploit the consistency of the data structure across multiple sites and build a software that is able to automate the process of turning the content of a website viewed through internet browsers into an augmented reality experience viewed through mobile AR browsers. In meeting this challenge, FastAR is able to take the content of any website developed using the SobiPro directory component [2] for Joomla and generate standardized AR content.

There are two principle choices that have to be made in rendering content as part of an augmented reality world namely, where to publish the AR content, and how to translate the data in an AR-compatible format. Concerning the former, there are three main options: a) through the free AR browsers that have been developed and marketed by commercial companies; b) using a commercial SDK or an open source toolkit for building a custom AR browser; and c) using an HTML5 based AR browser for exploiting the web browser capabilities. In FastAR, we have decided to rely on the free AR browsers offered by the major vendors in the field (i.e. Metaio [3], Layar [4] and Wikitude [5]), since it is the option that ensures maximum potential for dissemination and impact (i.e., the free AR browsers are already installed into millions of devices).

The three aforementioned companies publish content through a free AR browser using AR channels (also referred as "Layers" by Layar, or as "worlds" by Wikitude). AR channels are used to distinguish the provider of the content delivered through AR browsers. For instance, the Pizza Hut AR channel will augment the user's view only with the location of the Pizza Hut restaurants, while the Twitter AR channel will do the same only with geo-located tweets. The content of an AR channel is usually transmitted from the private server of the content provider (e.g., the enterprise website) to the server of the AR vendor and finally to the respective AR browser. In order for this to happen, the url corresponding to the API endpoint of the private server has to be registered into the server of the AR vendor. In addition, the content served by the private server should comply with the standards of the respective AR browser. However, these standards vary significantly among vendors, which makes the translation of data in AR-compatible format particularly complex.

In the typical case, each vendor offers a tool for generating content compatible with the respective AR browser. However, as will become evident in Section 2, despite their appropriately designed interfaces and easy of use, the generation of AR content using these tools is still a tedious process, since it involves typing or manually selecting data from a local repository. By focusing on websites that follow a strict data structure, FastAR succeeds in considerably automating this process for the major three AR vendors simultaneously.

The remaining of this paper is organized as follows. Section 2 reviews the related solutions and explains the added value of FastAR. Section 3 describes the technical details of

*e-mail:ververid@iti.gr. FastAR can be found at <http://arexporter.mklab.iti.gr>. The research leading to these results has received funding from the European Union 7th Framework Programme (FP7/2007-2013) under grant agreement 288815 corresponding to project Live+Gov (Reality Sensing, Mining & Augmentation for Mobile Citizen-Government Dialogue, <http://liveandgov.eu>), and EU Horizon 2020 Programme (H2020/2015-2018) under grant agreement 665066 corresponding to project DigiArt (The Internet Of Historical Things And Building New 3D Cultural Worlds, <http://digiart-project.eu/>).

FastAR, by: a) analyzing the functional scheme of Joomla and SobiPro in order to derive the core part of the database schema that remains unaltered independently of the website's nature; b) explaining the differences between the existing technological standards for AR and describing the architecture adopted to cope with their variability; and c) outlining the interconnection capabilities and usage scenario of FastAR. Section 4 provides a short overview of the activities that have been undertaken to build a community around the open source version of FastAR, before presenting the results of our survey in Section 5. Finally, conclusions are drawn in Section 6.

2 Related Solutions

In reviewing the related solutions our emphasis is on the systems that are available in the market for generating AR-compatible content. The majority of the existing platforms provide a desktop or a web based tool usually named as "Studio" or "Creator" for creating AR-compatible content. Apart from the AR companies, third party solutions also exist. In Table 1, the existing third-party tools for AR are outlined. *BuildAR* is a tool for generating content compatible with the Junaio browser, *Visar* for Layar, etc. However, as mentioned in the introduction, the functionalities supported by these tools are mostly manual which makes them inappropriate for undertaking the full transformation of a website.

It is only recently where the need to generate AR content in large scale has motivated the development of tools that aim to automate this process. More specifically, the goal of these tools is to generate an AR channel using content already stored in a database. Feedgeorge AR free plugin for Wordpress supports location and image based AR through Layar. When Feedgeorge plugin is installed in a Wordpress CMS, an AR tab appears for each post in the back-end. This tab can be used to insert the necessary AR resources such as the coordinates for triggering the post with geo-location information, or the marker image for triggering the post through vision recognition. The content is exported in Layar format. On the same direction, ConnectAR [6] is a commercial plugin for several CMSs (i.e. Typo3, Joomla, and Wordpress) that is able to export a page, an article, or a post into the Junaio AR browser. However, although Feedgeorge and ConnectAR succeed in automating part of the transformation process, some of the information that is essential to AR (e.g. the geographic coordinates of each page, article or posts), has still to be provided manually.

Motivated by the same objective but aiming at further automating the transformation process, FastAR focuses on websites that have been developed using the "Sobipro business directory" or any other Sobipro directory that already

includes longitude and latitude information. As explained in Section 3, Sobipro imposes a strict structure on the organisation of the website content, which allow us to automatically generate an AR channel based on this content.

3 FastAR

The proposed architecture is outlined in Figure 1. Joomla serves as the general platform for storing and managing the data [7]. Joomla serves also as a host for the so-called Joomla components that can be easily incorporated within a Joomla installation and offer a certain type of functionality. One such category of components that are build upon the Joomla CMS are content modules like SobiPro that are typically used for creating promotional websites advertising the services and goods offered by SMEs. These content components offer a range of tools and wizards for simplifying the creation of a promotional website but at the same time impose a rather strict structure on how the data are organized in the database tables. This structure is the critical feature exploited by FastAR so as to automate the process of generating AR content.

More specifically, FastAR is capable of undertaking the following two tasks. First, guided by the strict database structure imposed by SobiPro, FastAR is able to read the database entries (corresponding to the advertised services and goods) and generate the AR entities. These AR entities are necessary for synthesizing the AR channel. Second, FastAR interconnects with the servers of the AR vendors (i.e. Metaio, Layar and Wikitude) that are responsible for sending the data streams to the respective AR mobile browsers.

In implementing the aforementioned architecture there are two major technical challenges that need to be addressed. The first relates to the generation of fully functional AR entities out of the database entries that have been originally created to support an internet website. The second has to do with structuring and exporting these AR entities to the appropriate format so as to be compatible with the standards supported by each AR vendor.

Table 1: AR Content Management Systems review

Name	Modes	Platforms	Web page
BuildAR	LBS	Junaio	buildar.com
Dimple	LBS, IBS	Layar	dimplecms.com
Feedgeorge	LBS, IBS	Layar	feedgeorge.com
Hippo	LBS	Layar	onehippo.com
Hoppala	LBS	Junaio, Layar, Wikitude	hoppala-agency.com
Visar	LBS, IBS	Layar	visar.biz
Poistr	LBS	Layar, Junaio	poistr.com
Poiz	LBS, IBS	Layar	poiz.biz
ConnectAR	LBS, IBS	Junaio	connectar.com

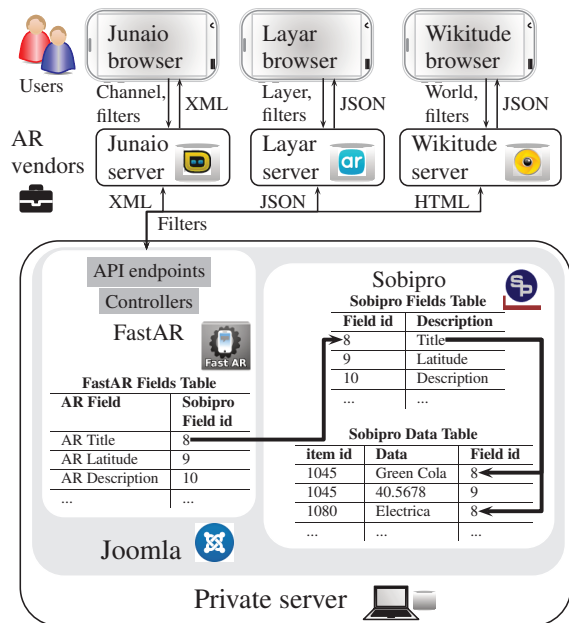


Figure 1: FastAR design to export SobiPro data to AR browsers.

3.1 Core data structure allowing AR content generation

Joomla stores html pages in a mysql database to present them in the web portal, also called front-end. Joomla is based on the model-controller-view (MCV) scheme. Briefly, the model defines the database schema, the controller defines the actions and view defines the presentation format. However, storing html pages in the database is not an efficient way of saving content when the html pages present the same kind of content, e.g. a chain of restaurants. Thus, it is typical for content modules like SobiPro to inherit the MCV scheme and make it more specific so as to simplify the process of generating the necessary html pages. SobiPro stores only the fields that describe the content, e.g. the title, the description, the image but not the entire html page. As a result, it is necessary for a website that has been developed using SobiPro to incorporate the same minimum amount of information for each of the advertised units. In addition, fields are customizable and new fields can be added in the minimum required set of information, such as longitude and latitude. FastAR associates the fields of the AR Entities with the fields of SobiPro. As shown in Figure 1 for example, by associating FastAR field “AR Title” with the SobiPro field “Title”, the titles of all SobiPro entries can be retrieved from the SobiPro “Data Table”.

On the other hand, mobile AR works by overlaying two layers. The bottom layer is the actual world layer which is visible through the device camera. The upper layer consists of the AR entities that can be computer graphics such as text, images, 2D drawings, and 3D models. The correct alignment of these two layers is achieved through proper transformations and tracking capabilities of mobile devices. Another essential element of mobile AR is the triggering mechanism that determines the conditions that must be met in order for an AR entity to appear. The triggering mechanism supported by FastAR is the proximity between the location of the mobile device and the geo-coordinates assigned to the AR entities, which can be further filtered using keywords.

Based on the above, by using the website details of the advertised unit to generate the AR entity and its geo-coordinates to define the proximity trigger, FastAR has all the information necessary to generate the AR channel. However, the content of this channel should be expressed in a format compatible with the AR vendor, which is the reason for adopting independent controllers in our architecture as described in the following section.

3.2 AR standards across vendors

As already mentioned, AR browsers work by placing a web view on top of a camera view. In this manner the interface is loaded in the web view using html, javascript, and css languages. Text or multimedia data are provided by html, javascript is used to define the human interface actions, and css defines the color style of the web view. Typical actions include calls to other applications, namely telephone, sms, e-mail, url, another AR channel, or share via social networks (through URIs).

The standard used by Metaio for the *Junaio* browser is the AR Experience Language (AREL), which is comprised of: a) XML that contains the urls of the resources; b) javascript indicating the scenario of the graphic user interface; and c) css for formatting the style of the interface. Javascript and css are loaded in a transparent web view to allow the camera preview surface to be visible. *Layar* uses javascript object notification (json) format for sending data. The json file contains both the resources (text, location coordinates,

image links, etc), as well as the actions allowed by the user during the immersion (e.g. onTap). In this case, the user interface is stylized in the channel registration portal. *Wikitude* supports three communication languages. The KML (Google Earth) is for making a simple location based channel, the ARML (OpenARML.org) which is an advanced version of KML that can be used for a location based channel with a custom user interface; and ARchitect which is the main language for allowing all features. ARchitect is formed by HTML, javascript, and CSS. FastAR uses ARchitect for Wikitude.

In order to deal with the variability of the existing AR standards, the transformation of the SobiPro data into the appropriate AR language is being performed through vendor-specific controllers. These controllers are responsible for acquiring the necessary information from the database and streaming the information to the appropriate format. Our current implementation incorporates three controllers that export the content in three different formats, namely XML for Metaio, JSON for Layar and HTML for Wikitude. In a similar fashion, a new controller can be implemented to provide support for an additional vendor. In addition, these controllers are also responsible for handling several data filtering parameters that are sent from the AR browsers to the data server, such as the visibility radius around the device; the maximum number of entities to download; keywords for filtering the entities, etc. In this way, and through these controllers, FastAR automatically generates the API End-points, i.e. the URLs that are used for registration in the portal (server) of the corresponding AR vendor, in order to stream the AR channel to the respective mobile browser.

3.3 Interconnection and usage scenario

The last functionality offered by FastAR is to enable the AR entities to interconnect with the website entities so as to facilitate the information flow. In a typical usage scenario of location-based AR the user spots the AR entity as shown in Figure 2(a) and taps on this entity to obtain more information, as shown in Figure 2(b). At this point, FastAR offers two different navigation options: a) transfer to the corresponding webpage of the advertised unit, or b) direct the user to download a native mobile application (if available) related to the advertised unit. In this way, users are prompted to visit the website or download the native application of the company.

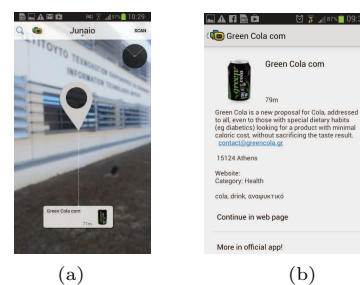


Figure 2: Views supported, (a) main view; (b) information page.

4 Open source software and community building

FastAR has been implemented as a Joomla component that is freely available as open source [8], under the Affero GPL license. The reason for wrapping FastAR as a Joomla component was to benefit from the already existing open source community. Indeed, Joomla is a widespread open source

CMS that is very often used to develop custom websites and its open architecture has resulted in the creation of a large and active community of developers, as well as a big number of extensions. Following this paradigm, we have made FastAR available as a free component through the Joomla Extension Directory [9] and also made sure to have FastAR listed as a third party tool by Layar and other AR-related blogs. In addition, we have designed our website [8] to further promote the development of a community around FastAR, by: a) providing detailed installation instructions with tutorials and videos; b) offering a quick demo, and c) offering technical support for users facing difficulties in FastAR installation and deployment.

5 System evaluation

An electronic survey was conducted across 12 evaluators (site owners, administrators and developers) that received the questionnaire through our website. The questions were selected to facilitate the qualitative evaluation of FastAR, following similar practices for evaluating AR systems [10, 11]. However, the aforementioned surveys evaluated the usefulness of AR as a visualization technology from the front-end aspect, while our goal has been to evaluate the usefulness of FastAR in rapidly generating AR content from the back-end aspect. Finally, we compare FastAR to other solutions capable of disseminating geo-referenced content such as Google Places, Foursquare and Open Street Maps.

5.1 Questionnaire results

The attractive features of FastAR is the ease of installation and configuration to expose AR content. The evaluation of these features are surveyed by the question “How much time was required to establish connection with one of the supported AR platforms?”. The answers shown in Figure 3 indicate that the majority of users achieved installation and configuration in less than 30 minutes.

The willingness of the back-end users to adopt the AR technology in their sites is surveyed with the question “Do you think AR is a useful way to advertise the content of your site?”. According to the results shown in Figure 4, the majority of the users considers AR technology a good way to advertise content with a score of 4.1 out of 5.

5.2 Comparison with alternative methods

Massively uploading multimedia content to several platforms is a non-trivial process. The enlisting of a company in *Google Maps* (GMaps) can be achieved through the *Google Places*

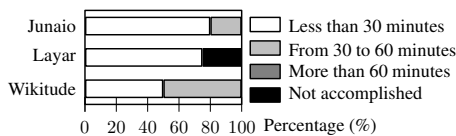


Figure 3: Surveying “How much time was required to establish connection with one of the supported AR platforms?”.



Figure 4: Results for the question “Do you think AR is a useful way to advertise the content of your site?”.

for business service. The steps of the process are a) the company owner should upload the content such as title, description, phone, latitude, longitude, website, and images; b) he or she receives a pin in the mobile phone to send it back for verification purpose; and c) after a two weeks review process, the company is enlisted. Alternatively, developers can use the *Google Places API* that allows for automatic bulk upload of several companies. However, the companies are enlisted only in the application of the certain developer, until they pass the moderation queue to be considered for Google Maps. *Open Street maps* (OSM) requires similar steps as Google Places, but there is no verification process and enlisting is almost immediate. One major drawback is that images and in general multimedia fields are not supported. OSM offers also API for adding companies massively. *Foursquare for business* is also similar with Google Places with the addition that it requires a small fee for each entry. It has also an API for retrieving but not for adding companies. The advantage of FastAR is that no additional effort is required by the company owner to advertise his business, since the content that is already hosted by the Joomla-based website is automatically exported to all major AR platforms. A brief comparison of the existing methods to advertise company content are outlined in Table 2. It is seen that FastAR is the only method that covers all three requirements, i.e. massive upload, image support and immediate availability.

Table 2: Characteristics of several platforms

Application	GMap	OSM	Foursquare	FastAR
Massive upload	Yes	Yes	No	Yes
Image support	Yes	No	Yes	Yes
Instant available	No	Yes	No	Yes

6 Conclusions and Future work

FastAR allows for making location based AR channels with as minimum effort as possible, enabling the content of web portals to become available to widespread AR browsers and drastically increase their visibility. Our focus has been placed on Sobipro business directory due to its widespread adoption on supporting business-like sites and its promotional advertising nature. However, other database schemas such as the VirtueMart and other functionalities such as image based AR will be considered in the future.

References

- [1] TripAdvisor, “Read reviews, compare prices, book your perfect trip,” www.tripadvisor.com, 2015.
- [2] S. Suski and R. Suski, “Sobipro - multi-directory component for Joomla,” <http://www.sigsiu.net>, 2014.
- [3] Metaio, “AR technology,” www.metaio.com.
- [4] Layar, “AR technology,” www.layar.com.
- [5] Wikitude, “AR technology,” www.wikitude.com.
- [6] ConnectAR, “Connect and deliver AR content,” <http://www.connectar.com/>, 2013.
- [7] Joomla, “The Joomla content management system,” <http://opensematters.org/>, 2015.
- [8] Mklab mklab.iti.gr, “FastAR - Link your Joomla to AR browsers dynamically,” <http://arexporter.mklab.iti.gr>, 2015.
- [9] OpenSourceMatters, “Joomla extension directory,” <http://extensions.joomla.org/>, 2015.
- [10] A. Dünser, R. Grasset, and M. Billinghurst, “A survey of evaluation techniques used in augmented reality studies,” in *ACM SIGGRAPH*, vol. 5, NY, 2008, pp. 1–27.
- [11] A. Dünser and M. Billinghurst, “Evaluating augmented reality systems,” in *Hdbk of Augm. Real.*, B. Furht, Ed. Springer, 2011, pp. 289–307.