Cloud Computing and Augmented Reality for Cultural Heritage

Pietro Vecchio[1], Francesca Mele[1], Lucio Tommaso De Paolis[1], Italo Epicoco[1(✉)],
Marco Mancini[2], and Giovanni Aloisio[1,2]

[1] Department of Engineering for Innovation, University of Salento, Lecce, Italy
pvecchio2014@gmail.com,
{francesca.mele,lucio.depaolis,italo.epicoco,
giovanni.aloisio}@unisalento.it
[2] Euro-Mediterranean Center on Climate Change, Lecce, Italy
marco.mancini@cmcc.it, giovanni.aloisio@cmcc.it

Abstract. In this paper the use of augmented reality and cloud computing technology to enrich the scenes of sites with a relevant cultural interest is proposed. The main goal is to develop a mobile application to improve the user's cultural experience during the sightseeing of a city of art through the integration of digital contents related to specific sites or historic monuments. The mobile application has been developed exploiting the Wikitude SDK software library. The huge amount of cultural digital contents (mainly represented by images) justifies the exploitation of a cloud-computing environment to obtain a multi-platform solution. In particular, we have chosen the KVM and OpenNebula as cloud platform technologies to build a private cloud environment, since they exhibit great features like openness, accessibility, simplicity and scalability. The work has been validated through test beds realized both in laboratory, using images captured from PC display, and in a real environment.

Keywords: Cloud computing · Augmented reality · Cultural heritage

1 Introduction

Nowadays many attempts for sharing and maintaining cultural contents by using cloud-computing technology have been made all over the world. ICT services and cloud computing make easier users access to cultural contents by providing application services (such as e-Learning, e-Library, e-Lab, e-Course), data services (such as learning materials, library collections and scientific publications) and infrastructure services (such as virtual storage). So, they can be used to narrowing the gap between different areas, gradually eliminate the digital divide, satisfy the people's cultural needs, improve the construction of a public cultural service system [1, 2].

The European biggest effort to digitize all of Europe's heritage and to make it publicly available for everyone is the Europeana project [3]. By combining databases from over 3,000 Europeana cultural heritage institutions and providing the access to more than 15 million cultural objects (images, texts, sounds, and videos), it gives a

unique online perspective on Europe's cultural heritage. An increasing number of projects, built using Europeana API, are contributing with technology solutions, cultural services and contents to Europeana. These projects are run by different cultural heritage institutions, and are co-funded by the European Commission's Content plus programme and the Information and Communications Technologies Policy Support Programme (ICT PSP). Among the others, CARARE is one of the most relevant project [4]. It aims at aggregating contents from the archaeology and architectural heritage making digital contents related to Europe's unique archaeological monuments and historic sites available to Europeana users. Moreover, the project provides tools and services to support its network of partners to make their digital contents interoperable with the Europeana environment and enables the access to 3D and Virtual Reality contents through Europeana cloud [5].

3D contents brought to European via the CARARE project, are complemented by 3DICONS project [6]. It enhances the contents base available to European users focusing on digital contents that include 3D models and reconstructions, enlarged models of important details and related images, texts and videos. It will also include and re-contextualize in 3D, objects belonging to a monument but presently located elsewhere, for example in museums. The project's activities also include the conversion of some existing 3D data into formats that are accessible by European users. The 3DICONS enriched Europeana with an unprecedented quantity of high-quality, 3D, well-organized and attractive information about the masterpieces of European architecture and archaeology [7].

In this context one of the most promising technology for enhancing the users experience, during the navigation through cultural heritage digital contents, is represented by the augmented reality. It can provide the user with an integration of digital contents with real images.

Due to the growing integration of powerful compute capabilities in mobile systems and smart-devices (i.e. oculus [8], google glass [9]), the augmented reality has become one of the research areas with higher demand and interest [10]. Nowadays the augmented reality is becoming even more pervasive in several applications such as in cultural heritage for improving tourism experience with a time navigation or with the integration of historic media [11, 12], in medical application to enhance and support surgical decisions [13, 14], in architectural construction, inspection and renovation [15] or to enhance the gaming experience [16].

This work describes how the augmented reality has been used in the cultural heritage context to enrich the scene of a point of interest (POI) with digital information. The implementation of such system and, in particular, the augmented reality applied in outdoor environment, requires that some challenging issues must be addressed such as the detection of a POI and its tracking when the camera moves. At the beginning, the first augmented reality platforms were implemented with some specific hardware solutions, adapted to different environments (e.g. [17–19]). With the grow of the market, mainly driven by the game consoles (i.e. Kinetic [20] and Nintendo Wii [21]), the devices which integrate cameras and sensors for motion detection, became the first augmented reality systems integrated in mobile systems. In that context, although there were some smart phones with supporting AR technology with marker-based algorithms, the solutions in an outdoor environment remained critical, due to hardware power that was still limited [22]. Many of those

mobile platforms were not really user-friendly. The *Archeoguide* platform, that proposed an augmented reality system for the exploration of cultural sites located in the archaeological site of Olympia in Greece, visualizes the missing and artifacts in the damaged areas of site of Olympia overlaying much additional information. The disadvantage of that project was the use of a heavy wearable HDM integrated helmet (Head Mounted Display) that was not quite mobile. Afterwards, the advantages deriving from the increase of computational power embedded in the new generation of mobile systems led to the development of newer and more performing techniques of augmented reality for outdoor environment. In this regard, the image processing algorithms focused to the matching and tracking of the POIs within of the frame set of observation [23–25] represent the most accurate algorithm for detecting POIs. The increase of computational performance in mobile systems and the use of cloud technologies fostered the development of frameworks devoted to easy the implementation of AR applications, such Wikitude [26, 27] and Metaio [28]. These frameworks provide easy tools to build different multimedia applications exploiting the augmented reality techniques.

The main goal of this work concerns the development of a mobile application with augmented reality capabilities, able to enhance the user's experience while she walks through the streets of a city looking at monuments, cathedrals and more in general at sites with cultural interest, having all the digital contents in his hands [29].

The next section describes the requirements analysis and the software architecture; The Sect. 3 reports the details about the implementation; follows the description of a case study. Conclusions and future work are described in the last section.

2 Architecture

The requirements analysis represents the first phase of the development process, immediately followed by the design of the software architecture. The application is meant for users with very low knowledge of computer science that would like to enjoy the digital contents related to the cultural heritage by means of a common mobile device. For this reason the application must be intuitive, easy to use and it must respond in real time to the user interaction. The following main functional requirements have been identified for the application:

- Objects detection: the user must be able to easily use the camera equipped in her own mobile device for the purpose of framing a cultural asset and getting additional related digital contents. The application must hence be able to detect objects within the scene.
- Overlapping of digital contents with the real scene: once an object has been identified, the application retrieves all the related digital contents and provides the user with such information putting the virtual content on top of the real image acquired from the camera.
- Management of huge amount of data: the application must be able to handle a huge amount of data with several formats, such as texts, images, video, sounds associated to a specific cultural asset.
- Offline mode: the application should operate also when the connectivity is reduced or absent.

The application must also meet non-functional requirements:

- Intuitive to use: the user should be able to use the application without reading any user's manual. The GUI must include intuitive icons and tips
- Real Time: the application must respond in real time to the user actions.
- Cross platform: the application should be used on different mobile platform
- Fault tolerant: the application must correctly handle faults coming from external modules i.e. connection lost or noise images from the camera.

The proposed framework can be divided into two principal subsystems. The first subsystem contains the *Mobile* components loaded into a mobile device, while the second subsystem represents the *Cloud Layer* with its support systems. The overall system architecture is shown in Fig. 1. The *Mobile* subsystem is implemented from the *Scene Identifier* and the *Rest Call-Images Collector* modules.

The *Scene Identifier* module acquires the image from the camera in real time and, by means of the algorithms of *Wikitude SDK*, obtains the recognition of a particular Point Of Interest (*POI*) inside the acquired video frame. The recognition is obtained by comparing the *POI* inside the frame with those ones loaded in the *Target Collection File* (*wtc* file) of *Wikitude*. If the comparison finds some images, then the *Scene Identifier* module invokes a *REST CALL* request to the *Data Node* subsystem, which then returns all the images related to the matching POI in the video scene. All the selected images are also placed into a logical structure, called Images Collector, which acts as a cache space for the mobile application.

As shown in Fig. 1, the collection of images used for the POI recognition is placed and, continuously updated, in the database of the Data Node. The Data Node module
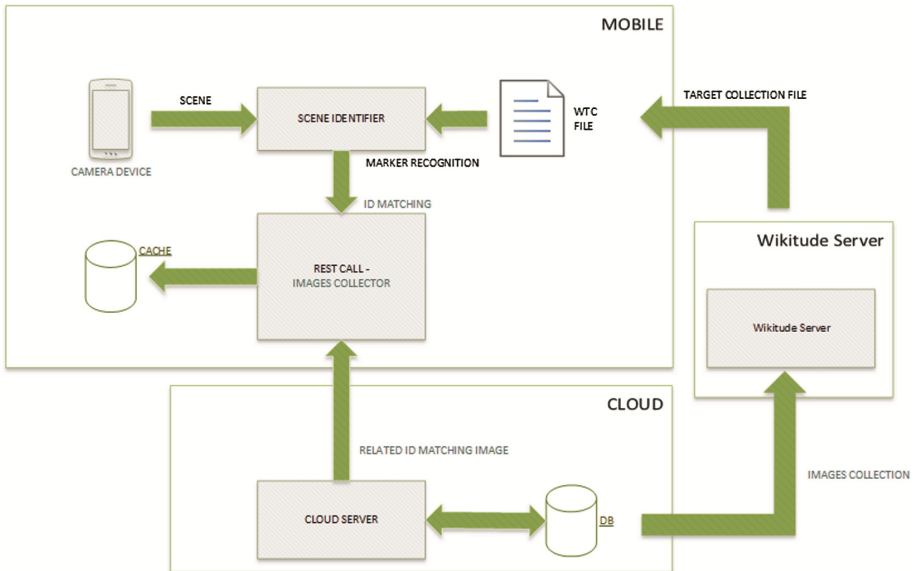
Fig. 1. Ovearall architecture of the system.

can add or remove images from the database; at the same time it interacts with the Wikitude Server to create or update the Target Collection File (wtc file) used by the Mobile subsystem. In this way, the Mobile device is able to use both the images loaded in the Data Node and the images stored locally in the mobile device for the detection of POIs. The local data cache also grants the use of the application in the offline mode.

3   Implementation

The proposed application uses the Wikitude SDK framework [10] and cloud computing technologies for obtaining an innovative, multi-platform and user-friendly augmented reality solution. Wikitude SDK ensures the integration with the main hardware platforms, i.e. Android, iOS, Smartphone, Tablet, Smart Glasses. It can integrate more cross-platform languages like JavaScript with HTML directly, to obtain a prototype portable on different mobile platforms. Wikitude implements the API calls in multi-platform environment with an SDK development toolkit in JAVA language.

All the API calls can be re-configured and customized according to the application environment. The JAVA language improves the implementation of multi-platform applications, like Android, iPhone and other smart-devices. The decision to adopt the JAVA language was also supported by the ability to integrate the java applets within web pages with dynamic content as JavaScript.

Wikitude SDK provides the tools to manage the dynamic interactions between the API calls of Augmented Reality with the HTML contents. At the same time, the use of JAVA language ensures the integration of the application modules of augmented reality with other Client/Server applications, like the cloud services and the database access.

The Wikitude framework includes a kernel for intercepting a particular POI inside the frame set of the mobile system, using particular matching and tracking algorithms. In this context, Wikitude engine continuously reads and compares in real time the captured frameset with a the model file with .wtc extension. The wtc model file contains all the information for all the POIs that the application should detect. According to the user requests and by the global position of the mobile system, the model file could be generated by the cloud or locally load into the mobile systems. When one POI is detected into the frame set, the Wikitude engine will send API calls for loading all the images from the cloud repository. A virtual slider with navigation buttons and various touch-screen interactions is created. According to the Internet services and with the user settings, the proposed application could be used offline, by using images and internet structures saved in the cache memory of the mobile system. The virtual slider is synthesized with the historical images loaded for a particular POI and it is implemented by using only the Wikitude Toolkit. The development has been realized through the Instrument for Development with Eclipse in Android environment.

For the purpose of our research we have chosen KVM (Kernel-based Virtual Machine) and OpenNebula open source cloud platform [30] as private cloud technology, since it exhibits great features like openness, simplicity and scalability. OpenNebula

assumes that physical infrastructure adopts a cluster-like architecture with a front-end, and a set of hosts where Virtual Machines (VM) will be executed. In order to meet the storage needs of Wikitude application, the Ceph software storage platform [31] is used. It is a distributed store object and file system designed to provide excellent performance, reliability and scalability. Its distributed storage solution allows the system to scale horizontally with multiple machines/heads instead of the more traditional methodologies which use centralized heads with large amounts of storage attached to them. Like most distributed storage solutions, Ceph really has 3 main components:

- *Ceph OSDs* (Object Storage Devices) i.e. the place where the blocks get stored. A Ceph OSD Daemon stores data, handles data replication, recovery, backfilling, reba-lancing, and provides some monitoring information to Ceph Monitors.
- *Ceph MDS* (Meta-Data Server) i.e. the place where the meta-data about files and blocks gets stored, especially used for looking up where the blocks are stored;
- *Ceph MON* (Monitors) i.e. the component keeping track of the state of the clustering. It is responsible for cluster management, configuration and state.

4   Case Study

In this section the cloud hardware infrastructure hosting Wikitude application is described. Our cluster configuration is composed by six nodes equipped with the following hardware specifications:

- 2x Intel Xeon Processor E5-2660 v2 (25 M Cache, 2.20 GHz)
- 256 GB RAM (16 blocks of 16 GB)
- 1x 200GB SSD IBM S3700 Enterprise
- 2x1TB HD in Raid 1 hosting the OS
- 4x3TB HD in JBOD mode (for Ceph Storage Cluster)
- 3x NIC 1 Gb/s
- 2x NIC 10 Gb/s

The cluster is configured as a private cloud with OpenNebula 4.10.2 and Ceph v0.87 Giant release (used as a datastore for Virtual Machine images). Each cluster node, running Linux Cent OS 7.0, acts both as a hypervisor node (Libvirt with QEMU/KVM) for the execution of virtual machines and as a Ceph storage node. Each node is configured with 4 Ceph OSD Daemons, one for each 3TB spinning drive; moreover on 3 of the 6 nodes, there is also the Ceph monitoring daemon. In order to augment storage perform-ance, each node SSD is used for reducing random access time and read latency and accelerating throughput; in particular each SSD is partitioned in the ratio of 1:4 to store the journal data of each Ceph OSD daemon, while the object data are stored on the spinning hard disk drives that are subject to limitations on seek time, access time, read and write times, as well as total throughput. Those physical limitations affect overall system performance–especially during recovery, and can be furthermore reduced by using a dedicated drive (2xHD in raid 1 in our case) for the operating system and

software, different from the HDs used for the object storage. Regarding the networking setup, one of the 10 Gb/s NIC is used for the accessing Ceph Storage from the VMs while the other 10 Gb/s NIC is used as an intra-cluster network during the Ceph recovery and rebalancing operations for performance and security reasons. One 1 Gb/s NIC serves to connect to Internet, another 1 Gb/s NIC is used for management, and the third 1 Gb/s NIC is used for connecting to other computing resource.

Ceph stores the client data as objects within storage pools. Using a *crush* algorithm, Ceph calculates which placement group should contain the object, and further calculates which Ceph OSD Daemon should store the placement group. The *crush* algorithm enables the Ceph Storage Cluster to scale, rebalance, and recover dynamically. Since 4.0 version, OpenNebula provides an interface for *Ceph RBDs* (RADOS Block Device), which allows registering images directly in a Ceph pool, and running VMs using that backend. In order to assure the correct use of OpenNebula with Ceph the OpenNebula worker nodes are part of a working Ceph cluster, the Ceph pool is available and the Libvirt/KVM is used as hypervisor.

For the test bed, a tablet Nexus 7 Asus with Android Kit-Kat 4.4.4 has been used. In Fig. 2 the result of the temporal/historical interaction is shown.

Fig. 2. Result shows an added images slider, on the bottom of the display, and a historical image overlapped on the real scene

In the proposed approach all of the historic images of the cathedral are displayed in the slide bar at the bottom of the screen. The user can select the historic image that will be superimposed on the real one. In the first phases, the images captured directly from the display of a PC have been used; afterwards others test beds will be conducted in real environment. When the user selects an historic image, a pop-up window is shown with the detailed information about the selection. The selected image can be seen in the contextual window and eventually in full screen (see Fig. 3).
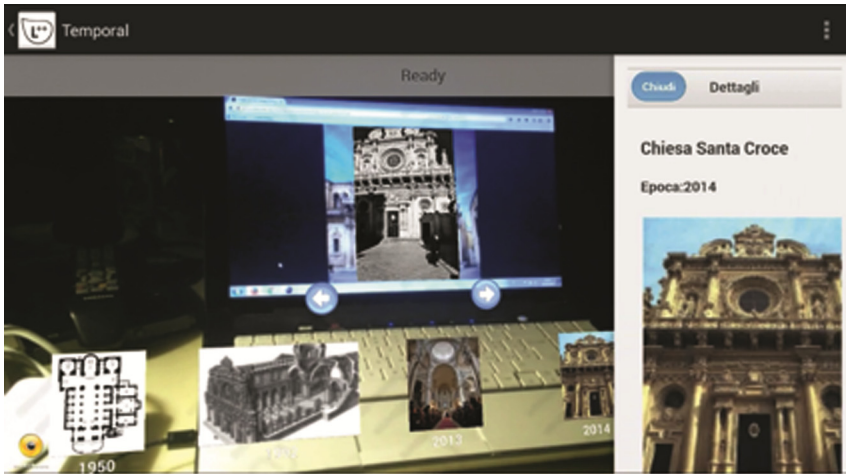
Fig. 3. The augmented reality shows an image slider and a temporal label related to the object on the right of the display

5   Conclusion

The manuscript treats an innovative system to enhance the edutainment in the cultural heritage with a cloud computing support. The proposed framework implements two principal components: *Mobile* component and the *Data Node* module. The *Data Node* module manages the *Rest Call* request calls from the mobile device and updates the images collection for detecting new *POIs*.

The *Mobile* component detects POIs inside the captured video frame and composes the augmented scene with a virtual image slider. The *Mobile* component implements the augmented services and communicates with the *Data Node* for retrieving the images to synthetize the virtual slider. The overall architecture has been implemented around the *Wikitude Sdk* engine both for the *Mobile* component and for the *Data Node*. The prototype of the *Mobile* component has been implemented through the Instrument for Development with Eclipse in Android environment; while the *Data Node* module has been performed by using Ceph software storage platform.

Future developments will be conducted by improving the interoperability with different data formats, and by introducing cross platforms technologies in order to support a wide variety of mobile devices, i.e. *iPhone* and *Window Phone*.

References

1. Li, S., Wang, D.: The study of culture sharing project based on cloud computing. In: 2013 Sixth International Conference on Business Intelligence and Financial Engineering (BIFE), pp. 62–69, Hangzhou, 14–16 November 2013
2. Nan Cenka, B.A., Hasibuan, Z.A.: Enhancing educational services using cloud technology. In: 2013 International Conference of Information and Communication Technology (ICoICT), pp. 155–160, Bandung, 20–22 March 2013
3. Europeana project website. http://www.europeana.eu/portal/
4. CARARE project website. http://www.carare.eu
5. Masci, M.E., De Santis, A., Fernie, K., Pletinckx, D.: 3D in the CARARE project: providing Europeana with 3D content for the archaeological and architectural heritage: the Pompeii case study. In: 2012 18th International Conference on Virtual Systems and Multimedia (VSMM), pp. 227–234. IEEE, Milan (2012)
6. 3D ICONS project website. http://www.3dicons-project.eu/
7. D'Andrea, A., Niccolucci, F., Bassett, S., Fernie, K.: 3D-ICONS: world heritage sites for Europeana: making complex 3D models available to everyone. In: 2012 18th International Conference on Virtual Systems and Multimedia (VSMM), pp. 517–520, Milan, 2–5 September 2012
8. LaValle, S.M., Yershova, A., Katsev, M., Antonov, M.: Head tracking for the Oculus Rift. In: International Conference on Robotics and Automation (ICRA). IEEE, 31 May 2014–7 June 2014
9. Ackerman, E.: Google gets in your face: google glass offers a slightly augmented version of reality. IEEE Spectr. **50**(1), 26–29 (2013)
10. Butchart, B.: Augmented reality for smartphones. Technical report, JISC Observator (2011)
11. Fiore, A., Mainetti, L., Manco, L., Marra, P.: Augmented reality for allowing time navigation in cultural tourism experiences: a case study. In: De Paolis, L.T., Mongelli, A. (eds.) AVR 2014. LNCS, vol. 8853, pp. 296–301. Springer, Heidelberg (2014)
12. Zoellner, M., Keil, J., Drevensek, T., Wuest, H.: Cultural heritage layers: integrating historic media in augmented reality. In: International Conference on Virtual Systems and Multimedia, VSMM 2009, vol. 15, pp. 193–196, 9–12 September 2009
13. De Paolis, L.T., Aloisio, G., Pulimeno, M.: An augmented reality application for the enhancement of surgical decisions. In: ACHI 2011: The Fourth International Conference on Advances in Computer-Human Interactions
14. Kilby, J., Gray, K., Elliott, K., Martin-Sanchez, F., Waycott, J., Dave, B.: Interface, information and interaction: an exploration of mobile augmented reality present and future. IBES The University of Melbourne
15. Webster, A., Feiner, S., MacIntyre, B., Massie, W., Krueger, T.: Augmented reality in architectural construction, inspection and renovation. In: Proceedings of the ASCE Third Congress on Computing in Civil Engineering, pp. 913–919, June 1996
16. Piekarski, W., Thomas, B.: ARQuake: the outdoor augmented reality gaming system. Commun. ACM **45**(1), 36–38 (2002)
17. Hoellerer, T., Feiner, S., Terauchi, T., Rashid, G., Hallaway, D.: Exploring mars: developing indoor and outdoor user interfaces to a mobile augmented reality system. Comput. Graph. **23**, 779–785 (1999)
18. Vlahakis, V., Karigiannis, J., Tsotros, M., Ioannidis, N., Stricker, D.: Personalized augmented reality touring of archaeological sites with wearable and mobile computers. In: Sixth International Symposium on Wearable Computers (ISWC2002), pp. 15–22 (2002)

19. Wither, J., Diverdi, S., Hllerer, T.: Using aerial photographs for improved mobile AR annotation. In: International Symposium on Mixed and Augmented Reality, pp. 159–162 (2006)
20. Zhang, Z.: Microsoft kinect sensor and its effect. Published by the IEEE Computer Society 1070–986X/12/ IEEE Multimedia at Work 2012
21. Pavlik, R.A., Vance, J.M.: A modular implementation of Wii remote head tracking for virtual reality. In: Proceedings of the ASME 2010 World Conference on Innovative Virtual Reality WINVR 2010, Ames, Iowa, USA, 12–14 May 2010
22. Hirzer, M.: Marker detection for augmented reality applications. Institute for Computer Graphics and Vision Graz University of Technology, Austria
23. Lowe, D.G.: Distinctive image features from scale-invariant key-points. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
24. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)
25. Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
26. Lechner, M.: A proposed step-by-step guide to an AR standard. Wikitude GmbH (2011)
27. Wikitude API Documentation GmbH. Homepage of Wikitude API. http://www.wikitude.com/doc/alr/index.html
28. Angermann, F., Krushwitz, M.: AR development with the Metaio product suite: demonstration of use cases in industry. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (2014)
29. Keil, J., Zollner, M., Becker, M., Wientapper, F., Engelke, T., Wuest, H.: The house of Olbrich - an augmented reality tour through architectural history. In: Proceedings of the 2011 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities, pp. 15–18. IEEE (2011)
30. Homepage of OpenNebula. http://opennebula.org
31. Homepage of Ceph. http://ceph.com