A Mobile, AR Inside-Out Positional Tracking Algorithm, (MARIOPOT), Suitable for Modern, Affordable Cardboard-Style VR HMDs

Paul Zikas[1,2(✉)], Vasileios Bachlitzanakis[1,2], Margarita Papaefthymiou[1,2], and George Papagiannakis[1,2]

[1] Foundation for Research and Technology Hellas, Heraklion, Greece
{mpapae02,papagian}@ics.forth.gr
[2] University of Crete, Heraklion, Greece
{zikas,csd3058}@csd.uoc.gr

Abstract. Smartphone devices constitute a low-cost, mainstream and easy to use h/w for VR rendering and main component for modern, mobile VR Head-Mounted-Displays (HMDs). They support rotational tracking from on board sensors to manage orientation changes, via their Inertial Measurement Units (IMUs), but they lack positional tracking to reflect head translational movements, a key feature that modern, desktop VR HMDs nowadays provide out-of-the-box. Taking advantage of the RGB camera sensor that each modern mobile device is equipped, we describe a novel combination of inside-out AR tracking algorithms based on both marker and markerless tracking systems to provide the missing positional tracking for mobile HMDs. We employed this system as an affordable, low-cost VR visualization h/w and s/w method, for heritage professionals to employ it for VR archeological sites and Cultural Heritage related monuments interactive walk-throughs. We also compared our results with a recent holographic AR headset (Meta AR-glasses) that supports gesture recognition and interaction with the virtual objects via its RGB-D camera sensor and integrated IMU.

Keywords: Mixed reality · Positional tracking · Monument visualization · Mobile VR · Marker and markerless tracking

1 Introduction

Most of Head Mount Displays (HMDs) support the basic three Degrees Of Freedom (DOF) that includes roll, pitch and yaw. The calculation of those can be achieved by onboard Inertial Measurement Unit (IMU) sensors which determine the orientation of the camera in space. However, in order to track positional movements, we need an additional external point of reference. This is usually implemented only in desktop VR HMDs, with the use of a camera placed in the external environment (outside–in tracking) or on-top or inside the HMD (inside-out tracking).

Positional tracking is the ability to determine the absolute position of the user's HMD within a three-dimensional space. By incorporating this feature into a VR headset we

M. Ioannides et al. (Eds.): EuroMed 2016, Part I, LNCS 10058, pp. 257–268, 2016.
DOI: 10.1007/978-3-319-48496-9_21

can represent these missing three more DOF in a total of 6. Originally Most HMDs could track only head rotation, providing 3 DOF (e.g. Oculus DK1). The updated Oculus Rift DK2 and current CV1 has an outside-in system for positional tracking with an external camera placed on top of the monitor to track the position of the headset. Other commercial HMDs follow similar tracking approaches. Studies showed [14] that positional tracking reduces the motion sickness these HMDs suffer as the orientation of the virtual world is very similar to the real one. However, all current desktop-VR HMDs need to be connected by cable to a computer. In contrast, smartphones provide a mainstream and easy to use mobile VR platform but they lack positional tracking. Visual Markers seems ideal for such low cost, mobile systems [16] that need a distinct visual sing to determine a static position from camera. By placing a marker in front of our desk we are simply define a static point in the real world in order to collect information about the relative position of the marker and our camera. Although markers are easy to use it is not very convenient to attach a marker everywhere we need positional tracking features. To overcome this limitation, we can use a markerless, SLAM-based AR tracking system. The main principles remain the same as we have to scan the environment to track visual features and extract a 3D map of the surroundings. Later we will use this mapping as a 3D marker to calculate the position of the camera like before.

For the above reasons we introduce two different ways of positional tracking with both marker and markerless implementations. For the needs of this project we developed a virtual reality navigation in the Palace of Knossos. User can navigate his way through the archeological site by rotating his head and leaning back and forwards or left and right as positional tracking is supported. To the best of our knowledge this is the first time in bibliography that AR inside-out positional tracking enhance conventional IMU sensor tracking in mobile VR.

In the following sections we present MARIOPOT for both marker and markerless implementations. In more detail, in Sect. 2 we describe the previous work in the area indicating AR and VR applications which appeared useful. Afterwards (Sect. 3) we fully describe MARIOPOT for both approaches (marker and markerless) by presenting the methodology, the needed calculations and the used matrixes. The next two paragraphs describe both the marker and markerless implementation of our application by pointing out the differences in each approach. In Sect. 6 we compare our results with Meta AR-glasses using marker based positional tracking with and gesture based interaction to handle and examine the archeological site of Knossos. Finally, we present the conclusions of our work and the future work-research that can be done.

2 Previous Work

When reconstructing a digital model of an archeological site it needs to be as accurate as possible [1] to improve the user experience (UX). For this reason, the model of Knossos we used is a realistic representation of the Palace as it stands today [11]. In recent years VR approaches for educational purposes have introduced a didactic potential in the area of cultural heritage. As [3] states, there are various VR systems available for cultural heritage purposes to enhance the interest of the general public. The benefit of an immersive and interactive simulation imparts knowledge and further motivation for great interest and research in the area of cultural heritage.

Both marker and markerless tracking intensify the basic features of augmented reality applications. [2] presents an AR application with markerless tracking about a quiz based on a museum visit. Another notable approach was done in [10] where a markerless system with automatically recover from features loss, runs an AR application for the maintenance of a car's engine. A robust authoring of life-sized AR characters for mobile devices was presented in [12], where a fast pipeline based on Metaio was used to populate AR scenes. The work featured an easy to use method for rendering AR characters with a novel pipeline in under one-minute process. The tracking of the area was done beforehand with Metaio to export a 3D mapping of the scene that will be later used for the character's authoring. The augmented characters were able to perform various animations, gestures and speech with the use of SmartBody library. This SLAM tracking method allows the augmentation of any indoors or outdoors scene capable of generate a decent number of features that will export the 3D map.

The importance of an easy to use AR mobile tracking system is emphasized in [7]. Daniel Wagner et al. present an accurate 6DOF pose tracking system that tracks conventional markers to render augmented content. Since most mobile devices have a build in camera they provide a complete tracking platform able to manage marker based tracking with the least effort. There is also an interesting apposition of the performance each mobile device had after a series of tests, providing a more technical view of the research. Moreover, [13] proposed a robust tracking system to determine the position of a mobile device in space using the build-in GPS sensor. A careful study on the energy consumption of different GPS based techniques is presented in parallel with their accuracy and efficiency. Mobile devices have limited power supplies requiring such systems to tackle this issue and manage the energy consumption to the point they reduce their impact on it.

In this work we have used the open source OpenGL Geometric Application (glGA) framework [8] implemented for IOS mobile devices. More specifically, glGA framework is a shader based computer graphics framework supported by desktops and mobile devices. The architectural structure of glGA provides the basic functionalities of OpenGL applications like loading and rendering both static and rigged 3D meshes, compiling and loading shaders, load textures and play animations.

3 Our Mobile, AR Inside-Out Positional Tracking (MARIOPOT) Algorithm

For the complete 6 DOF movement we need two main components: (1) The rotation of our device to determine the orientation of HMD in space and (2) its position to compute the translation in the digital scene. In our application we integrate the basic functionalities of cardboard SDK to get the rotation values as reflected from the orientation sensors. These values appear as a 3×3 matrix that from which we constructed the final view matrix of our application. Except from the rotations we also need the camera's translation. This is the reason we used a marker to provide the translation values. The position of the camera is a vector with three elements that defines the translation of the camera in all three axes.

From the extraction of these two components we have to construct the view matrix of the application. To make the transformations work properly we should not apply the matrix directly to our object as this will cause malfunctions in the visualization process, especially when the marker is no longer in sight. We have to construct a custom LookAt matrix from the extracted values as we need to rotate the world from the point of camera and not from the position of Knossos. By doing this we eliminate an issue that positional tracking brings about when the camera is changing position. We have to take in consideration the position of the camera in every frame as we always want to rotate the scene according to this specific point.

Below there is a brief explanation of the needed calculations to export the final view matrix.

Camera's Position. The position of the mobile device we are using can be represented as a three-dimensional vector. In both cases (marker and markerless) we extract the positional vector from the computed model matrices OpenCV and Metaio provide, saving only the last column of the matrix which holds the camera's position.

For the Marker implementation we used the detection mechanism [6] proposed to get the transformation values. After we calculate the marker's position we use the positional values to generate the application's Model Matrix as seen below.

```
m_transformations = markDetect->getTransformations();
//getTransformation() implemented in [6]

for (size_t i=0; i<m_transformations.size(); i++){
  const Transformation& t = m_transformations[i];
  Matrix44 glMatrix = transformation.getMat44();
  glm::mat4 initial = glm::make_mat4(glMatrix.data);
  MarkerMatrix[3][1]=initial[3][0]/initial[3][2];
  MarkerMatrix[3][0]=initial[3][1]/initial[3][2];
  MarkerMatrix[3][2]=initial[3][2];
  //For scaling we divide by the z transformation

  }
```

The Markerless approach differentiate on the calculation of positional values made by Metaio. The code below shows the construction of the transformation matrix in each frame.

```
void MainSystem::update(){
   metaioSDK->requestCameraImage();
   metaioSDK->render();
   TrackingValues tr = metaioSDK->getTrackingValues(1);
   //If tracking quality is sufficient
   if (metaioSDKReady && (tr.quality > 0)){
       float modelMatrix2[16];
       metaioSDK->getTrackingValues(1, m2, false, true);
       float metaioMat[16];
       metaioSDK->getProjectionMatrix(metaioMat, true);
       mat4 projectionMatrix = make_mat4(metaioMat);
       modelMatrix = make_mat4(m2);
       //modelMatrix contains the transformations needed
   }

  }
```

Camera's Rotation. The next step is to calculate the orientation of our mobile device. For both implementations we used CardboardSDK to extract the desired rotational values. The SDK provides build-in functions to get the head matrix of cardboard that represents the orientation of our device.

```
const GLKMatrix4 head_from_start_matrix = [headTransform
headPoseInStartSpace];
```

From the rotational matrix we have to compute three vectors that will constitute the orientation of our camera in all three axes. In order to create a functional camera matrix, we also need to determine the eye position which describes the position of our camera in virtual space. The eye position is defined from the rotational values and the position of our camera as follows.

$$
\begin{aligned}
eye_x &= -dot(xaxis, position) \\
eye_y &= -dot(yaxis, position) \\
eye_z &= -dot(zaxis, position)
\end{aligned}
$$

Final Matrix. The final LookAt matrix is computed by combining the above vectors of rotational and positional values. Our custom LookAt matrix will be used to compute the final view matrix of our application. It is important to notice that we have to take care of the matrix order while working with matrices from different systems or SDKs as the default order for OpenGL is column-major whereas Cardboard represents matrices in row-major order (Fig. 1).

However, there is a major issue with inside-out tracking when combining rotational tracking from cardboard and positional tracking using marker of markerless implementation. When we rotate the camera of our device we have two data inputs: (1) Cardboard SDK calculates the change of mobile's orientation. (2) From the point of marker it seems the model has changed position. This not accurate as we have a conflict between these

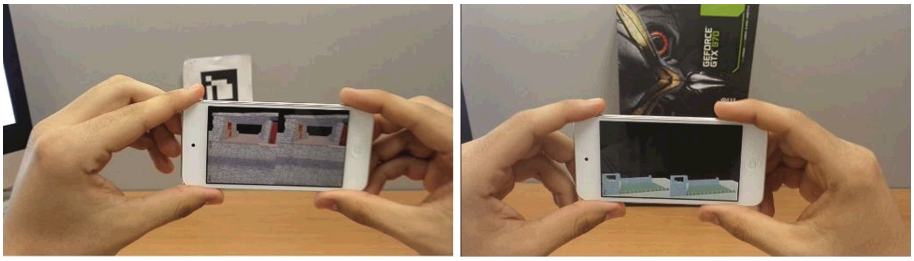


Fig. 1. Positional tracking in both marker and markerless implementation.

two data streams. Only if we use the marker for both rotational and positional tracking the visualization seems correct. To make things more clear, if we keep our device static and rotate it to the left while keeping contact with the marker, our virtual world will simultaneously rotate to the right (from Cardboard) and translate to the right (from marker) as from the camera's feedback the marker seems to move also to right. This is incorrect as we want our world to remain static but the camera will catch the marker moving to right and change the position of our virtual camera too. To reduce this effect we have to separate the two movements (rotation and translation) for both marker and markerless implementation. We provide a demonstration of this implementation in the next section (Fig. 2).

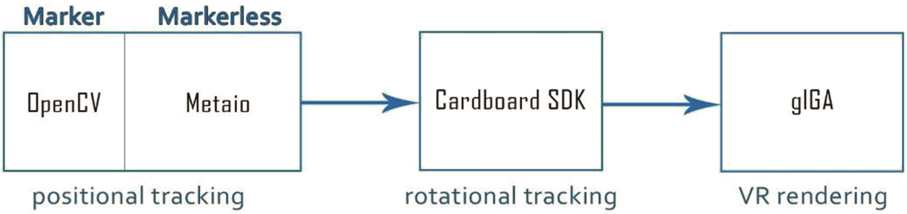


Fig. 2. The architectural diagram of our application

4 AR Positional Tracking with Markers

In order to determine the position of the camera we need to define a reference point in space. Marker based tracking is an efficient, low cost and easy to use method to achieve this result. To detect possible markers, we need to enable the camera of our mobile device and collect the data from live image streaming. Since every mobile device has an onboard camera there is no need for an additional sensor or active tracker to determine the position of the device. We used OpenCV image recognition patterns [6] to compute the position of our maker [9]. The detection of markers requires a pipeline of image processing that begins with the definition of existing candidates and finishes with the determination of the actual markers. The vector extracted from this method containing the positions in three-dimensional space will set the camera's translation in our virtual scene (Fig. 3).

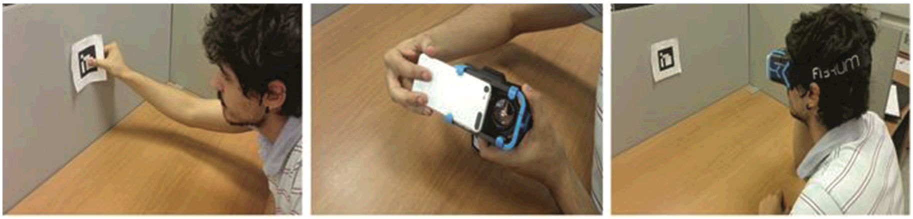


Fig. 3. Setting the marker and our HMD for marker based positional tracking

Marker based tracking has the limitation of a reference point that always need to be in camera's sight. If the marker is no longer in contact with camera the mobile device will stop calculating its position in the virtual scene as there will be no point of reference in the real world. The small field of view most mobile phones have will cause even more marker losses after an amount of rotation or translation. In our case, when the marker is out of reach of camera, the rotational tracking is applied to the last known position of the mobile device. This is achieved by saving the position we lost track with the marker and continue calculate the orientation values from the cardboard SDK. This implementation provides a smooth rotation even if the marker is not in sight and in the situation of the marker loss, the application continues to be functional with rotational tracking. When the camera tracks again the marker, we start computing again the transformation values and the positional tracking is enabled from the new position.

As mentioned before, we have to separate the positional and the rotational tracking to perform rotations without unwanted positional changes by the moving marker. To manage this issue we had to deactivate the marker when our mobile device overcomes a fixed angular threshold. By doing this the virtual scene will stop moving in a static rotation indicated incorrectly from the marker as the marker will no longer affect the position of the camera. We will activate the marker again when the mobile device reaches again lower angles and the computation of positional tracking will start again.

5 Markerless Positional Tracking

Unlike conventional tracking with markers, markerless tracking is definitely more flexible and reliable. To implement markerless AR camera tracking we employed Metaio SDK and the Toolbox. Both applications make markerless tracking an easy task by simplifying the process of capturing features and generating the final view matrix of the visualization. The procedure of markerless tracking is more complicated than placing a marker in front of your desk but it can track large areas (e.g. small rooms) providing user with more space to move around. With the markerless approach we have the opportunity to leave our desk to walk freely in a room and still receiving positional feedback from the 3D mapping we tracked (Fig. 4).

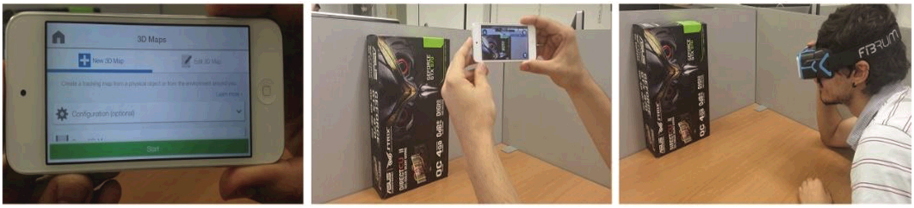


Fig. 4. The process from capturing features with Toolbox to markerless positional tracking.

To implement markerless tracking we need a 3D map of the area we are about to run the application. A 3D map is actually a file that contains the positions of district features in the environment and works the same way as a typical marker. We create this file by using Toolbox application. To make the file we run the application and scan the environment to capture visual features, the more features the better. After saving the 3D map we have to transfer it to the same folder our VR applications runs in order to detect the physical environment as a 3D marker.

Markerless implementation for positional tracking has the same methodology as marker based ones. We have to calculate again the rotational matrix form cardboard SDK and the translation matrix from Metaio SDK. The positional vector extracted from Metaio it is still a three-dimensional vector but the origin of tracking in 3D space is represented by a 3D marker (e.g. a box on top of our desk) and not by a 2D sign as marker based approach does. However, with markerless tracking we have a greater area of possible movements thus we have to scale the translation matrix accordingly to maintain the proportions of real and the virtual world translations. When the camera loses visibility of the features we maintain the last known position of the mobile phone and from there user can perform head rotations without having the ability to move to another position. After features become visible again user can start again moving around the virtual scene.

6 Comparison with Meta AR Glasses

Since we had our first results from the positional tracking we implemented this method with a different HMD that utilizes an RGB-D camera sensor. The main motivation behind this comparison is due to the fact that many forthcoming smartphones will feature such an RGB-D camera as part of their standard configuration h/w and we wanted to be the first to study their use and draw a comparison.

We reproduced our methodology in Meta AR-glasses, a holographic, see-through headset with gesture recognition. Wearing Meta AR-glasses gives user the ability to interact with holographic objects by using basic gestures [5]. Holographic headsets constitute an ideal HMD to experiment with novel AR applications [4]. Meta AR-glasses support rotational tracking from onboard sensors but they lack of positional tracking due to the absence of an external camera. However, we can utilize the embedded marker recognition mechanism to introduce positional tracking functionality (Fig. 5).

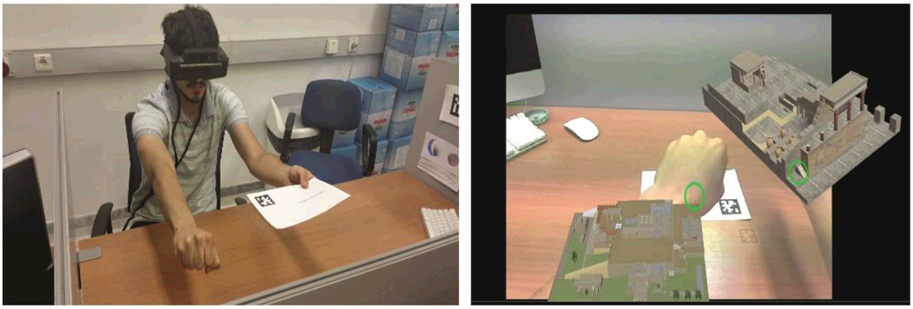


Fig. 5. Interaction using markers and hand gestures.

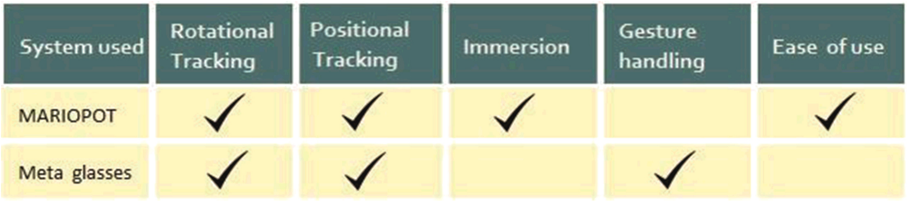
Meta use markers to attach holograms onto them but we used them in a different way. As mentioned earlier, we need to have a point of reference in the real world to manage the translations in the virtual one. To implement this idea we have to attach our virtual scene to the marker. By doing this we will enable positional tracking for Meta glasses. We need to compare applications with similar content to have more accurate results so for the Meta application we used again the model of Knossos. As indicated in [15] the two technologies have some basic differences in immersion, interaction and in the scale of the digital environment but the tracking mechanism remains the same. The positional tracking was of the same quality as it was implemented with the same principles. In general, by introducing positional tracking is Meta AR-glasses we were able to zoom in and out the palace of Knossos and see the model in more detail than previously when the only way to handle the model was with hand gestures.

While mobile VR applications have the advantage of generating an immersive experience, Meta headset with gesture recognition enhances the interaction between user and the hologram he sees. Interaction in game based applications has great importance as user is no longer an observer, he is able to handle holographic objects with native hand movements. In cultural heritage we can use this feature of Meta AR-glasses to handle and examine known monuments or important buildings from a different perspective.

In order to compare the user experience of Meta glasses and our mobile tracking system, we conducted an experiment to evaluate the use of both technologies. The participants were 7 in total (6 male, 1 female). For the mobile orientation demo participants had to navigate through the palace of Knossos by making head movements. In this way we were able to test if the navigation system was simple and easy to use. Most of the participants performed well as they managed to find their way in the ruins within less than a minute. After a while they were more confident and start to explore the archeological site in detail by zooming at point of great interest (e.g. the Prince with the lilies fresco). However, half of the participants complained about dizziness which was something expected as it takes time to get used to virtual reality. Another think that needs to be mentioned is that most of the participants had problems with the interruption of positional tracking caused by the loss of visual contact with the marker. As was shown by the process this was the most common issue we faced at the experiment.

For Meta AR glasses the participants had to handle a building of the archeological site with the marker provided. In addition to the previous experiment we enhanced the user experience with gesture based interaction to find out if there will be any significant change to the final remarks. We asked the participants to zoom in specific points of the building as well to rotate, translate and scale it with their hands. Most of the participants had a serious problem with gesture handling as they couldn't manage to grab the building correctly. Besides this, the marker based handling of the building was a task that completed successfully from all the participants. About this, there were few complains referring to the limited field of view Meta AR glasses provide as the building was cropped from being too close to the headset leading to the failure of the immersion. As a final remark, the participants were satisfied from the HMD's orientation capabilities that were enhanced with the use of positional marker based tracking.

To conclude, both HMDs perform well in the positional tracking and orientation in general. Our mobile implementation had better results considering the immersion of the archeological site as the participants were in a fully virtual environment unlike Meta glasses. Since we used conventional markers in both methods to provide positional tracking they had very similar results in accuracy and performance. An extra feature we used with Meta headset was the gesture handling they support but we faced poor results as it was difficult for most of our participants to successfully perform the correct gestures for the device to recognize (Fig. 6).



System used	Rotational Tracking	Positional Tracking	Immersion	Gesture handling	Ease of use
MARIOPOT	✓	✓	✓		✓
Meta glasses	✓	✓		✓	

Fig. 6. Comparison table of MARIOPOT and Meta AR glasses.

7 Conclusions and Future Work

In this work we presented a low cost and easy to use implementation of positional tracking for mobile device-based, cardboard-style VR HMDs that are ideal to be used as affordable visualization for cultural heritage professionals and sites. It is the first time in bibliography that such a hybrid orientation system is presented, combining sensor and camera tracking. With the contribution of positional on-top of rotational tracking in mobile VR, we were able to zoom in and out not only to change our orientation but also our translational position in a virtual world and thus appreciate the presented 3D monuments in better detail, with less motion sickness effects. The presented algorithm can improve the visualization of digitized archeological finds (e.g. pottery, frescos, and coins) instead of using conventional software methods. Since the digitization process in Cultural Heritage is used frequently, we have to develop the appropriate tools to make better use of those findings. A VR demonstration of monuments and historical buildings is without a doubt the best way of visualization as the immersion generated from this technology can really make the difference. Mobile AR, inside-out Positional tracking can extend the basic limits of VR and especially when implemented with markerless tracking in room sized areas as user will not only be an observer of the monument but he will actually walk and explore the site on his own. Such technology can be used in museum galleries and expeditions to attract the interest of general public and to inform users about the importance and the benefits of digital preservation.

In the future we aim to replace the Metaio SLAM-based markerless tracking with OpenCV. We are also planning to enhance the visualization of Knossos with gamification elements to constitute a fully interactive experience of the archeological site. Serious game industry expands rapidly over the last years introducing new ways of learning through gaming. Cultural heritage applications can definitely benefit from a more interactive environment that will augment their efficiency.

Acknowledgements. The research leading to these results has received partial funding from the European Union People Programme (FP7- PEOPLE- 2013-ITN) under grant agreement No. 608013 and from the European Union Horizon2020 Programme (H2020-SC6-CULT-COOP-9 – CSA) under grant agreement No. 727107. Special thanks also to ARdictive (www.ardictive.com) and its CEO & founder Nils Huebner for the MetaAR glasses.

References

1. Foni, A., Papagiannakis, G., Magnenat-Thalmann, N.: A taxonomy of visualization strategies for cultural heritage applications. ACM J. Comput. Cult. Heritage **3**(1), 1–21 (2010)
2. Yudiantika, A.R., Sulistyo, S., Hantono, B.S.: The development of mobile augmented reality quiz visualization methods based on markerless tracking for museum learning application. Disampaikan pada seminar "The International Forum on Strategic Technology (IFOST)", June 2015
3. Gaitatzes, A., Christopoulos, D., Roussou, M.: Reviving the past: cultural heritage meets virtual reality. In: Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage (VAST 2001), pp. 103–110. ACM, New York (2001)
4. Billinghurst, M., Clark, A., Lee, G.: A survey of augmented reality. Found. Trends Hum. Comput. Interact. **8**(1), 1–202 (2015)
5. Billinghurst, M.: Gesture based AR interaction research at the HIT Lab NZ. Presentation (2015)
6. Baggio, D.L., Emami, S., Escrivá, D.M., Ievgen, K., Mahmood, N., Saragih, J., Shilkrot, R.: Mastering OpenCV with Practical Computer Vision Projects
7. Wagner, D., Schmalstieg, D.: ARToolKitPlus for pose tracking on mobile devices. In: Grabner, M., Grabner, H. (eds.) Computer Vision Winter Workshop 2007, St. Lambrecht, Austria, 6–8 February 2007. Graz Technical University (2007)
8. Papagiannakis, G., Papanikolaou, P., Greasidou, E., Trahanias, P.: glGA: an OpenGL geometric application framework for a modern, shader-based computer graphics curriculum. In: Eurographics 2014, pp. 1–8 (2014)
9. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proceedings of the IEEE International Workshop on Augmented Reality, pp. 125–133 (1999)

10. Platonov, J., Heibel, H., Meier, P., Grollmann, B.: A mobile markerless AR system for maintenance and repair. In: Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006), pp. 105–108. IEEE Computer Society, Washington, DC (2006)
11. Kateros, S., Georgiou, S., Papaefthymiou, M., Papagiannakis, G., Tsioumas, M.: A comparison of gamified, immersive VR curation methods for enhanced presence and human-computer interaction in digital humanities. Int. J. Heritage Digit. Era **4**(2), 221–233 (2015)
12. Papaefthymiou, M., Feng, A., Shapiro, A., Papagiannakis, G.: A fast and robust pipeline for populating mobile AR scenes with gamified virtual characters. In: SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications (SA 2015). ACM, New York (2015). Article 22, 8 pages
13. Kjærgaard, M.B., Langdal, J., Godsk, T., Toftkjær, T.: EnTracked: energy-efficient robust position tracking for mobile devices. In: Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (MobiSys 2009), pp. 221–234. ACM, New York (2009)
14. Desai, P. R., Desai, P. N., Ajmera, K. D., Mehta, K.: Int. J. Eng. Trends Technol. (IJETT) **13**(4) (2014)
15. Zikas, P., Bachlitzanakis, V., Papaefthymiou, M., Kateros, S., Georgiou, S., Lydatakis, N., Papagiannakis, G.: Mixed reality serious games and gamification for smart education. In: 10th European Conference on Games Based Learning (ECGBL), October 2016
16. Zhang, X., Fronz, S., Navab, N.: Visual marker detection and decoding in AR systems: a comparative study. In: Proceedings of the 1st International Symposium on Mixed and Augmented Reality (ISMAR 2002), p. 97. IEEE Computer Society, Washington, DC (2002)