

A Workflow Analysis for Implementing AR-Based Maintenance Procedures

Federico Manuri¹(✉), Andrea Sanna¹, Fabrizio Lamberti¹,
Gianluca Paravati¹, and Pietro Pezzolla²

¹ Dipartimento di Automatica e Informatica, Politecnico di Torino,
C.so Duca degli Abruzzi 24, 10129 Turin, Italy
{federico.manuri, andrea.sanna, fabrizio.lamberti,
gianluca.paravati}@polito.it

² Fidia S.p.A., C.so Lombardia 11, 10099 San Mauro Torinese, Turin, Italy
p.pezzolla@fidia.it

Abstract. The widespread adoption of mobile devices is giving everyone access to augmented reality systems, possibly involving a huge number of people in AR-based apps, with a pervasive social impact that cannot be neglected. AR systems are becoming affordable to everyone and especially useful in the maintenance field. This report aims to describe in a clear and accessible way the workflow to design and develop an augmented reality (AR) application for supporting maintenance procedures. The main focus of this paper is the evaluation of markerless tracking systems, as they could provide environment-independent solutions. The tests performed on a real use case outline the robustness of 3D CAD tracking with respect to other solutions.

Keywords: Augmented Reality · Maintenance · Tracking Systems · Robustness

1 Introduction

The main purpose of an AR-based system is to allow users to “experience” the real world enriched by a set of overlapping computer-generated contents and eventually other sensory inputs, such as audio augmentation through earplugs or speakers [1]. The augmentation generated by the computer is strictly related to the user context, both in terms of view and location: the virtual objects are aligned to the real ones in a meaningful way to provide the user a better understanding of the surroundings. As augmented reality is something strictly related to the experience of the real world, an AR system should collect a wide range of information from different sensors (e.g. camera, GPS, and so on). Another peculiarity of AR systems is to be real time, so a device should provide enough computing power to interactively run an AR application.

Until some years ago, the lack of cost-efficient devices was the main barrier to a wider adoption of AR applications. Nowadays, the widespread adoption of

mobile devices has removed this limitation, as smartphones and tablets feature all the sensors and processing units needed to develop and deploy AR applications. Moreover, the technology innovations that affect mobile devices can produce new challenging products, commonly referred to as ‘wearables’, and industries are making steady progress in developing new categories of AR devices, such as the Google Glass project [2] and contact lenses from Innovega [3]. The global market for augmented reality is growing fast and the pervasive adoption of AR technologies implies an undeniable impact on today’s society. Industrial applications for maintenance, repair and manufacturing have always been an interesting domain for AR and the benefits that these solutions could offer are investigated into details in [4] and [5]. In addition, current AR applications could effectively replace the traditional maintenance solutions such as paper manuals and handbook instructions.

This manuscript aims to present a workflow for implementing markerless AR maintenance procedures, evaluating challenges, opportunities and limitations of the latest AR technologies. The main focus of this research is evaluating the robustness of different tracking systems, thus assessing the applicability of AR-based maintenance procedures with respect to variable environmental conditions.

The paper is organized as follows: Section 2 briefly outlines the state of the art of AR applications, with a focus on the maintenance procedures area. Section 3 presents a workflow diagram for implementing an AR application for maintenance, whereas evaluation of experimental data is provided in Section 4. Finally, open problems and future works are discussed in Section 5.

2 Background

Over the years, developers and researchers investigated many different domains that could profit from AR systems. Industrial, military and medical applications were the first area of research, followed soon after by commercial and entertainment apps.

In recent years AR technologies were used in many different domains, such as tourism, shopping, social networks and advertisement. SnapShop Showroom [6] allows users to capture an image of the room they wish to furnish, quickly browse through a vast catalogue of furniture from big retailers like IKEA [7] and position items in the virtual environment to see how the room may look like. Google’s Ingress [8] is a MMO strategy game where players, grouped in two factions, fight for the control of virtual territories simply walking around and looking for points of interest in the real world, such as sculptures, libraries, post offices, memorials and so on. Wikitude World Browser [9] is an AR browser that offers geographically-relevant information of the surroundings that could be valuable for the user, detailing restaurants, ATM location and many other information.

One important domain for AR is the development of applications for maintenance procedures. The use of AR systems to provide instructions to maintainers and technicians was first investigated in the early 1990s [10][11]. Since then several experiments led to prototypes and evolutions in this field using head-mounted displays to perform maintenance procedures [12]. The usefulness of

AR-based systems for training and maintenance applications was investigated in several studies, with the development of prototypes to maintain PCs [13] and other industrial contexts such as plant maintenance [14] or facility management [15]. The latest works further investigate AR-based solutions for car maintenance and suggest the use of AR applications to replace instruction manuals or handbooks.

This paper focuses on conceiving and designing a workflow for developing AR applications that could be applied to various domains, including maintenance procedures. While other researches usually point out the AR solutions that best fit with their domain or peculiar requirements, this research aims to provide a methodology to adopt when planning the development of an AR application, independently of the specific domain.

Nowadays, the market offers a wide range of AR frameworks, which differ for license (commercial, free, open source), tracking systems (marker, markerless) and target devices (computer, mobile devices) [16]. A framework based on Metaio SDK [17] is presented in this paper; Metaio offers the wider set of tracking systems among its competitors (such as Vuforia [18], Layar [19] and many others) and it provides a free SDK and well documented APIs. Nonetheless the workflow proposed in this manuscript aims to be unbounded from the AR platform of choice and easily adaptable to any AR framework. The AR market is drastically increasing and the spreading of AR applications and the pervasive adoption of AR technologies provide significant areas of research for AR developers [20]: this paper aims at filling the gap between users and AR technology, focusing on the application development and providing a clear and accessible workflow to create AR applications, and particularly maintenance apps.

3 The Workflow

Several issues and problems have to be considered when developing an AR application for maintenance, repair and assembly. First of all, an efficient tracking system has to be identified. The tracking system can be based on the object (or objects) to be tracked or it can be based on artificial features (e.g., markers), which can be added to the object itself. Moreover, the focus can be either on just one tracking system or on a hybrid approach, depending on the steps of the procedures to be performed. For example, it can be necessary to recognize an object with a complex geometry (by using 3D CAD or 3D Map tracking) and then a flat surface with several textures (by using 2D image tracking). Depending on the tracking system, a set of configuration files needs to be created. These files can consist of several 2D images, such as photos or computer-generated renderings, or 3D models of parts of the object to be tracked.

The next step involves the description of the procedure to be used for supporting the end-user during the maintenance/assembly task. Computer-generated (sometimes called virtual) hints can be: text labels, images, 3D static models, 2D and/or 3D animations, videos, audio messages.

During the last step, all the code to perform the above mentioned contents (and possibly extra features depending on the user requirements) has to be

implemented for the chosen AR technology. Figure 1 provides a high-level visualization of the workflow for developing an AR application.

3.1 Tracking Process

The tracking process is the core feature of every AR application. Among the different tracking solutions available, markerless systems do not require to add any artificial feature to the scenario and they depend only on the object to be tracked. These characteristics are fundamental when developing an application that should be reliable independently of the environment and other context conditions, such as illumination.

2D Image Tracking. This tracking system is based on providing an image to match with the current scene framed from the camera. The image could either be a photo of the object or a rendering of a 3D model of the object itself. Depending on the provided image, the environment would be part of the recognition process, but it could be excluded for better re-usability of the application. The 2D Tracking System recognizes the object when the camera frame “matches” the provided image; of course, the point of view of the camera has to be as similar as possible to the one used to gather the reference image. The recognition will go on till the tracked object is inside the area of the camera, even stretching the camera view (to some extent).

3D Map Tracking. 3D maps markerless tracking allows users to use any object as a tracking reference. This technique is based on creating a map of points by the Toolbox app. This map will be used in the developed application to match with the current viewed frame for positive recognition. The first step is to use the Toolbox app provided by Metaio to create a 3D map of the desired object or environment. The output of this process is a *.3dmap file, which is intended to be used with the Creator or the SDK.

The points created by the mapping are based on geometry and texture and they could come from both the object and the environment. All visible features in the proximity of the object are detected from the smartphone camera during the 3D map creation task. For this reason, it is better to manage the entire operation on a neutral background, avoiding any undesired interference with the surrounding environment. The map could be edited to improve its robustness, thus removing useless points and trying to focus on the object itself. As the texture information is used for the tracking, light condition and texture of the objects should be considered and could limit the object tracking.

3D CAD Tracking. 3D Markerless Tracking based on CAD data allows a precise pose localization based on a given 3D model of an object or a part of the environment; for example, a small object that should be tracked or a building that the tracking system should recognize when the camera frame “matches” the

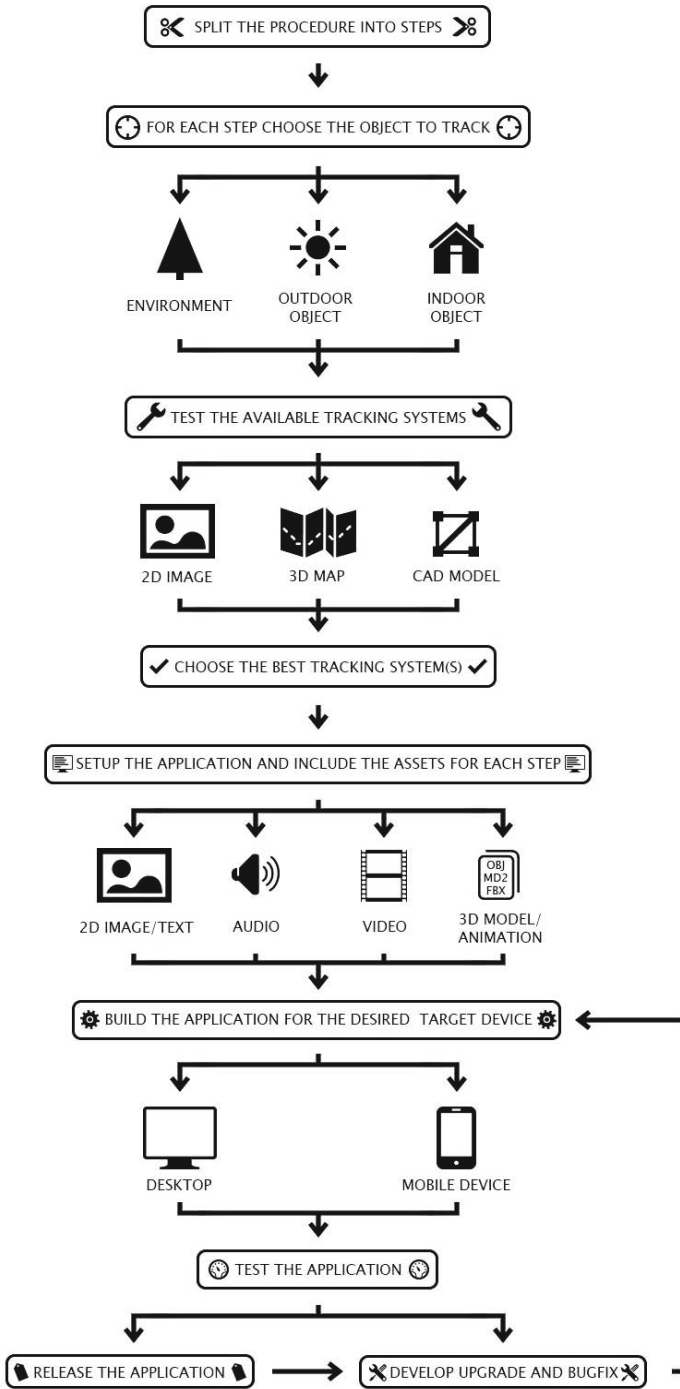


Fig. 1. The Workflow Diagram

model. The 3D model is used for an edge based pose initialization to enable an accurately scaled and localized augmentation. It detects the camera pose only once and then it switches to markerless 3D tracking. The edge based initialization process uses a separately controllable Dummy Tracker or GPS, compass and gravitational sensor information as a prior pose approximation and searches for the correct camera pose in a specifiable range around it. It requires a surface model and a line model of the environment or object to track, either getting it through the Metaio Edge Config Tool for a single view or creating it with a modeling software or automatic algorithm from the original 3D model.

3.2 Workflow Implementation: A Case Study

The following case study shows the development of an AR application for maintenance procedure on an ink-jet printer. The application has been developed to test the CAD and 2D tracking system by the Metaio framework. 3D Map tracking was avoided because it intrinsically relies on texture information and light condition, whereas the purpose of this research is to investigate solutions that are independent of the environmental conditions.

Workflow Diagram. The first step is to define a diagram that describes all the steps to be performed to develop the application, starting from the specifications. The main purpose of this application is to provide instructions for changing the two ink cartridges (black and colour) of an ink-jet printer. A 3D CAD model of the printer is necessary to implement the CAD tracking by the Metaio framework. Then the components of the printer used as assets are animated.

The next step develops the application by Metaio Creator. Finally, the application is tested to verify the reliability of the different tracking systems implemented.

Defining the Procedure Steps. The maintenance procedure for changing the ink cartridges requires the following three operations: opening the printer enclosure so that the cartridges holder is accessible; opening the cartridges holder and replace the cartridges; closing the printer enclosure to return to the initial state. Performing the procedure is possible to clearly define the steps that make up these operations:

1. opening the front panel;
2. opening the upper cover;
3. pressing the cartridges substitution button to move the cartridges holder to an accessible position;
4. opening the cartridges holder, replacing the cartridges and closing the cartridges holder;
5. pressing the cartridges substitution button to realign the cartridges holder to its default position;
6. closing the upper cover;
7. closing the front panel.

Choosing the Object to Track. The next step is to define what will be tracked to identify the different steps of the procedure. Since the printer is normally used indoor, environment and outdoor objects can be excluded as tracking. Moreover, performing the procedure pointed out that the printer configuration could be uniquely defined by the printer chassis in steps 1, 2, 3, 5, 6 and 7. Steps 4 is defined by the cartridge holder configuration and its position related to the printer chassis. For these reasons the printer chassis will be the tracking object for steps 1, 2, 3, 5, 6 and 7, while the printer chassis and the cartridge holder will define the tracking object for step 4.

Testing the Tracking Systems. At this point it is necessary to choose the most suitable tracking system. Among the three available tracking systems, the 3D map system has been avoided for the given case as it is the one that most relies on the light condition and texture of the objects. As the printer presents no texture of any kind and the light conditions could vary depending on the environment, the 3D map system could limit the object tracking and will be poorly effective. 2D Image and CAD model could equally fit for the given case study and the preliminary tests did not point out clearly which system would fit better. A procedure would usually require a lot of effort for producing the tracking object configuration for each available tracking systems and for each step of the procedure itself, so it is recommended to test only some of the most significant steps with different tracking systems at this phase of the workflow. The simpleness and shortness of the given case study allowed to develop the entire procedure with both 2D Image and CAD model tracking systems: this is the reason why all the testing results are evaluated later in the tests section. For the 2D Image tracking system, photos of the printer were taken for each steps of the procedure, while for CAD model tracking system the CAD of the printer was created, as detailed in the next section.

Modeling the Object. A 3D mesh of the printer is modeled, by using Blender [21], as similar as possible to the real printer. The model is used for: real-time tracking recognition with the 3D CAD tracking system; rendering some poses of the model to test the reliability of the 2D tracking system; creating animations that will be used as assets in the AR application.

In order to provide a better visualization of the assets in the final application, the texture “baking procedure” is used (Fig. 2). This procedure allows to: define a texture for the model, setup a lighting/shading system in Blender and perform a UV mapping of the texture that includes light parameters. This feature is not used by the 3D CAD tracking system, because it relies only on a wireframe representation (also called line object model) of the model.

When the model is ready, it is then necessary to setup poses identifying the steps of the procedure. In this example, four different poses of the printer define the procedure (Fig. 3), as steps 5, 6 and 7 are the mirror of steps 1, 2 and 3. Each pose is then exported as an .obj file, the format required by Metaio Creator. The four poses are also rendered to be used in the 2D Image tracking system.

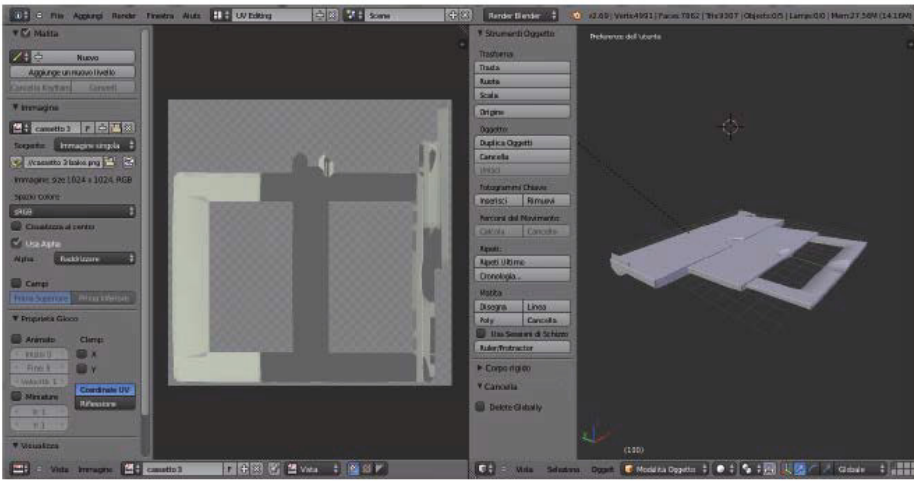


Fig. 2. The texture baking procedure: a texture is mapped by the UV mapping system to add texture, lightning and shading to the 3D model

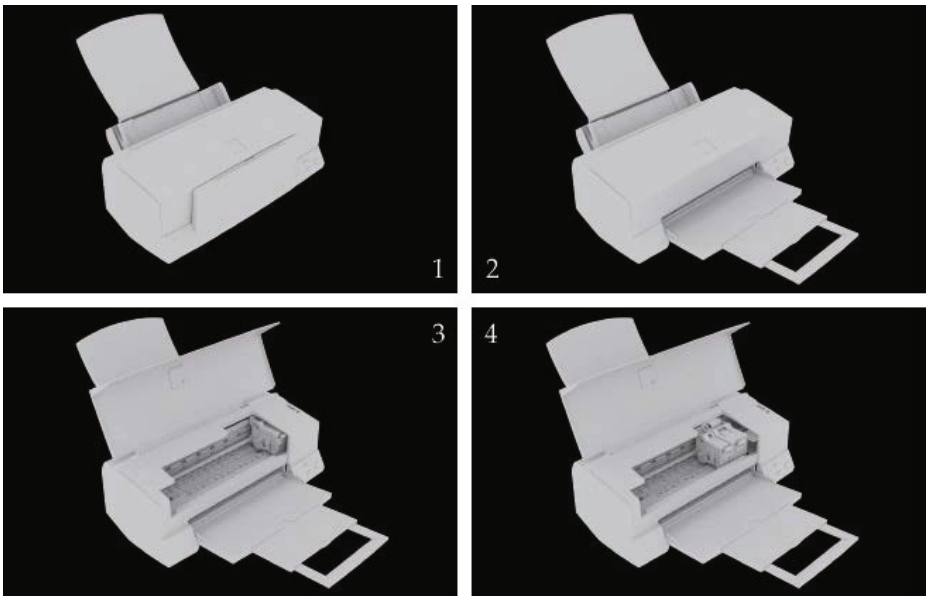


Fig. 3. The four poses defining the four different steps of the maintenance procedure

Creating the Assets. The last step to obtain all the resources for developing the application is to create the assets, a set of computer-generated hints that describe the procedure and support the end-user during the maintenance/assembly task. The Metaio SDK offers content creation guidelines in three major areas: images, movies and 3D animations. At first only a set of animations has been produced as assets for the procedure to speed up the developing process. After the first set of tests, text, audio and video instructions have been prepared as part of the “develop upgrade and bugfix” step to offer a wider set of assets for future usability tests. Each animation represents one of the task that the final user should perform for the specific maintenance procedure. The following animations are implemented in Blender: top panel opening/closing; front panel opening/closing (Fig. 4); toner case left/right translation; toner case opening/closing; toner extraction; toner insertion. When all the animations are ready, it is necessary to convert the resulting FBX files by the FBX Mesh Converter tool provided by Metaio. The output will be an MFBX package compatible with Metaio Creator.

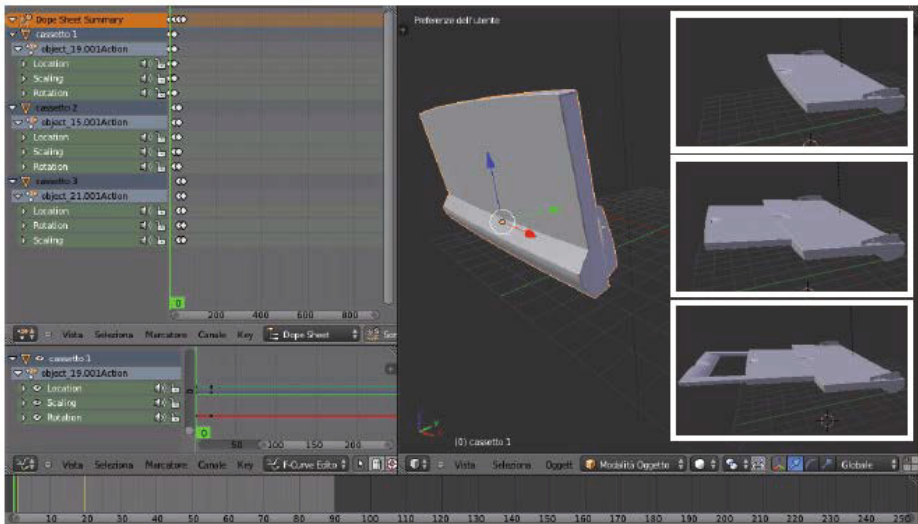


Fig. 4. The front panel animation implemented in Blender

Creating the Application with Metaio Creator. For each pose defined for the procedure, the following steps have to be performed:

1. creating a new scene;
2. importing and positioning the .obj file that represents the pose to be tracked in the scene (see Fig. 5);
3. creating a line object model through the edge tool utility (see Fig. 6);
4. tuning the parameters that define the tracking system;

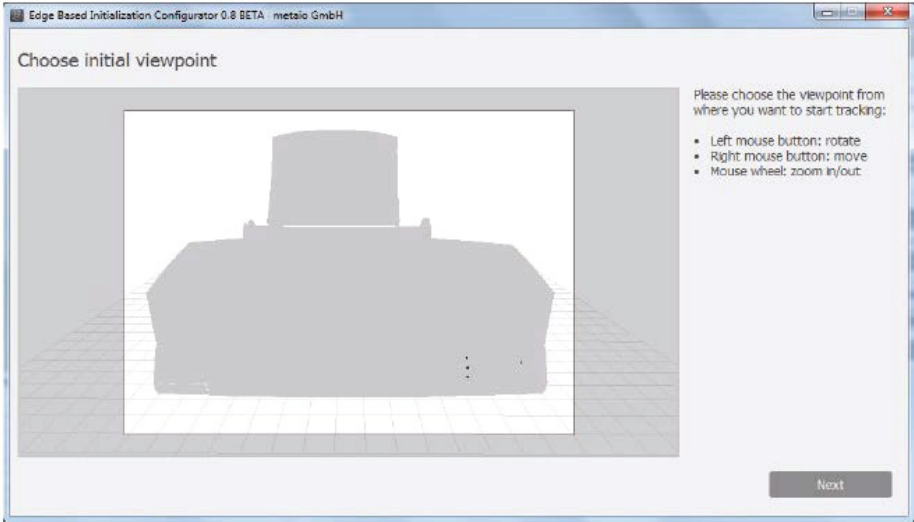


Fig. 5. An example of tracking object model

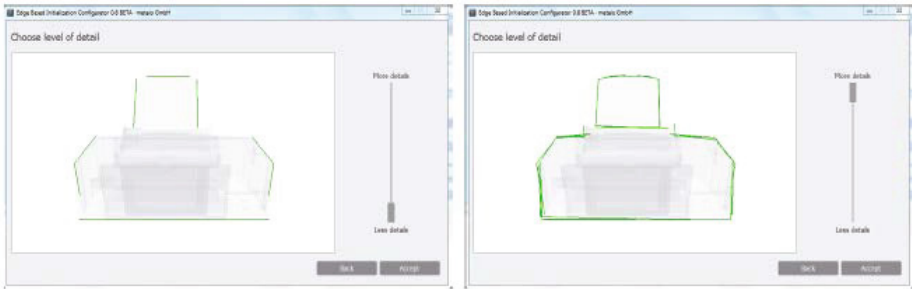


Fig. 6. An example of line model object (wireframe) representation

5. importing the assets and aligning them to the real object;
6. tuning the parameters that define the assets behavior;
7. testing the animation in the preview.

When all the scenes are ready, it is possible to export the project as an application available for the main platforms (Windows, iOS, Android). In the context of this work, the application has been tested on an Android device. To develop another version of the application that relies on the 2D Image tracking system, the previous steps are performed with the following differences: in step 2 renderings are imported as .jpeg files, instead of the .obj model, whereas step 3 is not performed as it is specific for the 3D CAD tracking system.

4 Tests

The developed application has been tested by using the real object in order to assess the efficiency of the proposed tracking systems. Tests on the real object have been repeated using 3D CAD tracking and 2D tracking. The second solution has been tested with both renderings and photos of the real object. As one of the main focus of the research is that the procedure should be repeatable independently of the environmental conditions, the photos were taken through the tracking camera and edited to remove all environmental information. As detailed before, since the last three steps of the procedure (5,6 and 7) are specular to the first three ones, only steps 1-4 were taken into account for these tests.

Four parameters have been considered in each test, in order to evaluate the performance of the two solutions: tracking object quality, recognition threshold, luminosity, alignment precision. The tracking object quality defines the quality of the image or CAD model for the tracking algorithm. Both systems have their own rules to finely craft the tracking object. Metaio Creator rates with a three star system how much the chosen tracking object is suitable. The better the rating, the easier for the system to recognize the object and avoid false positive, providing a robust solution. Recognition threshold evaluates the similarity parameter used to calibrate the matching between the tracking virtual object and the real one ranging from 0.00 to 1.00. Metaio suggests the range 0.30 - 0.70 for better results, as values below 0.30 will lead to a too inaccurate recognition and values over 0.7 could make it difficult to recognize the object. A low value means that the system is less robust: false positives or alignment errors of the assets might occur in this case. Luminosity has been evaluated to test the robustness of the systems in different situations. The tests have been performed with low light (30 lux), medium light (160 lux) and high light (300 lux), where low light represents a nearly dark room and high light represents a well illuminated office desk. The luminosity value is approximated and it is evaluated through the luminosity sensor of a Nexus 10" Tablet. 3D CAD recognition should always work, as it does not rely on the color information, such as texture or grayscale, but only on the object geometry: while the first parameter could change with different luminosity, the second should be independent of it. Alignment precision defines how precisely the system overlaps the assets when it correctly tracks the object. This parameter could only be estimated by the final user on a range of 0.00 - 1.00, where 0.00 means the object is not recognized and 1.00 means that all the assets are always perfectly aligned to the real object.

The tests were performed with the applications deployed through Metaio Creator (v.3.2.2) and a Logitech HD Webcam C310 to track the printer. Table 1 shows the results of the tests performed for the four steps of the printer maintenance procedure.

The first evaluation is that the 2D image tracking based on the renders of the model does not work properly for the proposed real object. The tests were performed changing the recognition threshold from 0.70 to 0.30, to check if a feeble value could provide some significant results for the alignment quality parameter. The main issue could be that the 3D model of the printer does not

Table 1. Results of the tests performed on the printer maintenance procedure, with different luminosity conditions and tracking configurations, for the four steps of the procedure. Legend: TOQ = Tracking Object Quality, RT = Recognition Threshold, AQ = Augmentation Quality.

STEP 1									
	LUX 40			LUX 160			LUX 300		
Tracking Configuration	TOQ	RT	AQ	TOQ	RT	AQ	TOQ	RT	AQ
3D CAD	3	0.70	0.80	3	0.70	0.80	3	0.70	0.80
2D Image with Render	2*	0.30	0.00	2*	0.30	0.00	2*	0.30	0.00
2D Image with Photos	2*	0.30	0.00	2*	0.50	0.40	2*	0.35	0.20
STEP 2									
	LUX 40			LUX 160			LUX 300		
Tracking Configuration	TOQ	RT	AQ	TOQ	RT	AQ	TOQ	RT	AQ
3D CAD	3	0.70	0.90	3	0.70	0.90	3	0.70	0.90
2D Image with Render	2*	0.30	0.00	2*	0.30	0.00	2*	0.30	0.00
2D Image with Photos	2*	0.30	0.00	2*	0.45	0.35	2*	0.30	0.00
STEP 3									
	LUX 40			LUX 160			LUX 300		
Tracking Configuration	TOQ	RT	AQ	TOQ	RT	AQ	TOQ	RT	AQ
3D CAD	3	0.70	0.90	3	0.70	0.90	3	0.70	0.90
2D Image with Render	2*	0.30	0.00	2*	0.30	0.00	2*	0.30	0.00
2D Image with Photos	2*	0.60	0.65	2*	0.85	0.90	2*	0.75	0.80
STEP 4									
	LUX 40			LUX 160			LUX 300		
Tracking Configuration	TOQ	RT	AQ	TOQ	RT	AQ	TOQ	RT	AQ
3D CAD	3	0.70	0.90	3	0.70	0.90	3	0.70	0.90
2D Image with Render	2*	0.30	0.00	2*	0.30	0.00	2*	0.30	0.00
2D Image with Photos	2*	0.60	0.65	2*	0.85	0.90	2*	0.75	0.80

*since the 2D image contains transparency, Metaio could not provide an exact evaluation of the Tracking Object Quality.

provide enough texture or visual references for the tracking algorithm to match the real object.

The 2D image tracking based on the photos provides better results but it is not robust enough. First of all, if the photo does not provide rich texture data (step 1 and 2), the recognition threshold drops significantly. As a consequence, also the alignment quality drops: alignment errors may occur in terms of deviation between the real object and the expected virtual asset position.

Moreover, the alignment might change when moving the camera from the tracking view. This problem worsen when the luminosity is changed: in step 2 the recognition fails for darker or lighter environment conditions; in step 1 it is necessary to reduce the recognition threshold significantly to obtain a minimum alignment in the lighter environment. In step 3 and 4, the printer provides better photos in terms of recognition algorithm and the alignment is possible even when the luminosity parameter changes. Finally, this solution is still feeble if the object does not provide enough texture data and it would need an algorithm

to dynamically change the recognition threshold on luminosity variation. On the other hand, this solution provides the simpler and faster way to create the tracking object, as taking the photos and doing some image editing are the only requirements. Moreover, it does not require modeling skills to produce a model of the real object, which takes more time and could be onerous.

The 3D CAD tracking is the most robust solution. Changing the luminosity of the environment does not affect the alignment quality, without the need to loosen the recognition threshold. Moreover, this system provides the best alignment quality results in each step of the procedure and it is the more suitable for the maintenance domain. On the other hand, the 3D CAD-based tracking is the most onerous system because it needs a model of the real object to properly work. If the manufacturer could not provide a 3D model of the object, it is necessary to create it with a 3D modeling software: this could be difficult depending on the complexity of the object and the skills of the user. Furthermore, 3D modeling could not be possible for a variety of reasons: the original model could not be available, the object could be too tiny or simple to provide suitable recognition features and the tracking object could be too complex to provide a 3D model similar enough to the real one.



Fig. 7. Lightening the real object with a torch

To further evaluate the robustness of these solutions, other tests were performed for the three proposed luminosity conditions: the first test consisted of casting shadows over the real object in order to change the luminosity of the



Fig. 8. Casting shadows over the real object

surface not linearly (Fig. 7). While the 3D CAD tracking was not affected by this variation, with the 2D image tracking the alignment quality would drop even more, eventually leading to not recognize the real object. The second test consisted of lightening the real object with a torch, a Maglite 3CELL D flashlight (Fig. 8): with the 2D image tracking the result was the same as the previous test; the 3D CAD tracking was sometimes affected, especially if the torch highlighted a section of the real object that defined the geometry (such as an edge section corresponding to the line object).

5 Conclusions

This paper presents a workflow to efficiently develop augmented reality markerless applications, with a special focus on the maintenance domain. The most important problems have been evaluated, investigating all the steps to design, implement and test an AR application. Finally, two markerless solutions have been compared by a real use case. Performance and robustness of the different systems have been evaluated to identify the best environment-independent tracking solution.

The problems found with the 2D image tracking based on render could originate from either the quality of the 3D model or the lack of tracking features by the real object or both of them. Moreover, image tracking based on render should be further investigated as the advantage of using renders relies on the possibility

to simulate different texture conditions, such as shadows, dirt, brightness, dust and so on.

Future works could include further research on the 3D CAD tracking system when future releases of the Metaio SDK occur, since this tracking solution is now provided in a not stable version. Moreover, 2D tracking could be improved with a system to better evaluate the quality of the images used as trackables and an algorithm could be developed to dynamically change the recognition threshold on luminosity variation.

Acknowledgments. This work is funded by the EASE-R³ project [22]: Integrated framework for a cost-effective and ease of Repair, Renovation and Re-use of machine tools within modern factory, FP7, FoF.NMP.2013-8, Grant agreement no: 608771.

References

1. Van Krevelen, D.W.F., Poelman, R.: A Survey of Augmented Reality Technologies, Applications and Limitations. *The International Journal of Virtual Reality* **9**, 1–20 (2010)
2. The Google Glass project web site. <http://www.google.com/glass/start/>
3. The Innovega web site. <http://innovega-inc.com/>
4. Henderson, S.J., Feiner, S.: Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair. *IEEE Trans. on Visualization and Computer Graphics* **17**, 1355–1368 (2011)
5. Terenzi, G., Basile, G.: Smart Maintenance: An Augmented Reality Platform for Training and Fields Operations in the Manufacturing Industry. *ARMEDIA Augmented Reality Blog* (2014). <http://arblog.inglobetechnologies.com/?p=1370>
6. The SnapShop web site. <http://www.snapshopinc.com/>
7. The Ikea web site. <http://www.ikea.com/>
8. The Ingress web site. <https://www.ingress.com/>
9. The Wikitude Web Browser site. <http://www.wikitude.com/app/>
10. Ong, S.K., Yuan, M.L., Nee, A.Y.C.: Augmented Reality Applications in Manufacturing: A Survey. *Intl. J. Production Research* **46**, 2707–2742 (2008)
11. Nee, A.Y.C., Onga, S.K., Chryssolourisb, G., Mourtzisb, D.: Augmented reality applications in design and manufacturing. *CIRP Annals - Manufacturing Technology* **61**, 657–679 (2012)
12. Feiner, S., Blair, M., Dorée, S.: Knowledge-based Augmented Reality. *Communications of the ACM* **36**, 52–62 (1993)
13. Ke, C., Kang, B., Chen, D., Li, X.: An Augmented Reality-based application for equipment maintenance. In: Tao, J., Tan, T., Picard, R.W. (eds.) *ACII 2005*. LNCS, vol. 3784, pp. 836–841. Springer, Heidelberg (2005)
14. Savioja, P., Järvinen, P., Karhela, T., Siltanen, P., Woodward, C.: Developing a Mobile, Service-Based Augmented Reality Tool for Modern Maintenance Work. In: Shumaker, R. (ed.) *HCI 2007 and ICVR 2007*. LNCS, vol. 4563, pp. 554–563. Springer, Heidelberg (2007)
15. Kahn, S., Olbrich, M., Engelke, T., Keil, J., Riess, P., Webel, S., Graf, H., Bockholt, U., Picinbono, G.: Beyond 3D “As-Built” Information Using Mobile AR Enhancing the Building Lifecycle Management. In: *12th International Conference on Cyberworlds*, pp. 29–36. IEEE Press (2012)

16. Augmented Reality SDK Comparison. <http://socialcompare.com/en/comparison/augmented-reality-sdks>
17. The Metaio web site. <http://www.metaio.com/>
18. The Augmented Reality Trends web site. <http://www.augmentedrealitytrends.com/>
19. The Vuforia web site. <https://www.vuforia.com/>
20. The Layar web site. <https://www.layar.com/>
21. The Blender web site. <http://www.blender.org/>
22. The EASE-R3 project web site. <http://www.easer3.eu/>