A New AR Authoring Tool using Depth Maps for Industrial Procedures

J. Gimeno[a], P. Morillo[a], J.M. Orduña[b], M. Fernández[a]

[a]*Instituto de Robótica (IRTIC). Universidad de Valencia*
C./Catedrático José Beltrán, 2
46980 Paterna (Valencia) - Spain
Contact Address: Pedro.Morillo@uv.es
[b]*Departamento de Informática. Universidad de Valencia*
Avda. de La Universitat, S/N
46100 Burjassot (Valencia) - Spain
Contact Address: Juan.Orduna@uv.es

Abstract

Several Augmented Reality systems have been proposed for different target fields such as medical, cultural heritage, military, etc.. However, most of the current AR authoring tools are actually programming interfaces that are exclusively suitable for programmers. In this paper, we propose an AR authoring tool which provides advanced visual effect, such as occlusion or media contents. This tool allows non-programming users to develop low-cost AR applications, specially oriented to on-site assembly and maintenance/repair tasks. A new 3D edition interface is proposed, using photos and Kinect depth information to improve 3D scenes composition. In order to validate our AR authoring tool, two evaluations have been performed, to test the authoring process and the task execution using AR. The evaluation results show that overlaying 3D instructions on the actual work pieces reduces the error rate for an assembly task by more than a 75%, particularly diminishing cumulative errors common in sequential procedures. Also, the results show how the new edition interface proposed, improves the 3D authoring process making possible create more accurate AR scenarios and 70% faster.

Keywords: Augmented Reality, Authoring Tools, Non-immersive Desktop, Kinect, Occlusion

1. Introduction

Augmented Reality (AR) systems have been widely used in numerous applications such as medical procedures [11], automotive and aerospace design [27], maintenance tasks [14], or cultural applications [34]. The term Augmented Reality (AR) defines computer graphic procedures or applications where the real-world view is superimposed by computer-generated objects in real-time [2].

One of the major problems in the development of an AR application is the creation of content, due to the lack of appropriate tools for developing 3D images with depth perception. There are popular software libraries like ARToolKit [17] and ARToolKitPlus [36] that use OpenGL, VRML or OpenSceneGraph [5] to represent the 3D models on the real images in real time. However, the use of these and others computer graphics libraries requires programming skills to generate AR applications, and every AR development should be constructed from the scratch.

In order to avoid these problems, AR authoring tools were proposed a decade ago [26, 12, 33, 19]. The main advantage of AR authoring tools is that they do not rely on time and cost consuming recompilation steps, and therefore the changes and enhancements in the development of AR systems are fast and efficiently completed. Different authoring tools have been developed during last years. By order chronological order, we can cite some representative examples, like the work by MacIntyre et al. [21], that consists of a software plug-in built on top of Macromedia Adobe Director, and it allows to author AR content for this widely used multimedia development environment. Another examples are an extensible and general-purpose AR authoring platform based on XML descriptions, proposed by Ledermann and Schmalstieg [20], or STUDIERSTUBE Framework, another proposal for the prototyping of AR applications developed by Schmalstieg [28]. Even a well-known visual programming environment (ECT graphical programming tool) was modified to add support for AR input by Hampshire et al. [10]. Later, an extensible authoring tool that supports both scripting and a drag and drop interface and real time interpreted input was developed by Seichter et al. [31]. A recent work even classifies the existing AR tools depending on the use of high or low level libraries, and the programming knowledge required for using these tools [37].

Assembly, maintenance and even repair tasks are some of the direct application fields of AR tools, and a lot of proposals have been made in these industrial areas [22, 38, 9]. However, most of the proposed AR systems have been specifically developed for enhancing certain procedures in the domain of the problem. The development of augmented reality systems usually involves two key design issues: the decision of implementing a mobile or a non-mobile system [9, 29], and the choice of selecting a helmet-mounted or a hand-held display systems as AR visualization device [38]. Although several AR systems have been proposed for industrial purposes, most of them superimpose the computer-generated objects on the real view of qualified workers. This forced superposition cause the occlusion problem, which occurs in AR systems when a computer-generated object closer to the viewer obscures the view of real elements further away along the line-of-sight [4]. If the occlusion problem is not properly addressed in the development of an AR system for industrial purposes, then the developed tool does not significantly facilitate workers their actual on-the-job tasks. This fact is especially evident in the development of AR systems for assembly or repair/maintenance purposes, because of the cluttered backgrounds and the frequent occlusions in these types of industrial environments [25].

Figure 1 shows an example of the occlusion problem. Concretely, it shows a custom AR system that has been used for the on-site repair process of a CNC (Computer Numerical Control) lathe located in a machine-parts factory. The pictures included in this figure show the step when the sliding headstock is taken off by releasing a fixing stud and six protective nut caps. The left picture in this figure shows how non-occluded 3D computer-generated stud and nut caps (intentionally modeled using eye-catching black and red colors) are visualized over the headstock of the CNC lathe, showing a misleading final location of the elements. On the contrary, the right picture of the same figure shows how this augmented stud and the two nut caps have been correctly occluded by the real objects in the foreground of the scene, indicating the proper location of their positions within the back side of the CNC lathe.

In this paper, we propose an easy-to-use AR authoring tool oriented to the development of AR applications for the execution of industrial sequential procedures. The main contribution of this tool is that it allows non-programming users to develop low-cost AR applications, including occlusion capabilities, in a timely manner (by means of the use of a Kinect sensor [15, 16, 39]). A new edition interface, using real world photos as templates while 3D modeling, is introduced. These templates, denoted as scenarios, are easily created using a wizard from 2 different sources: a common photo (2D scenario) or an image captured using a Microsoft Kinect device (3D scenario).

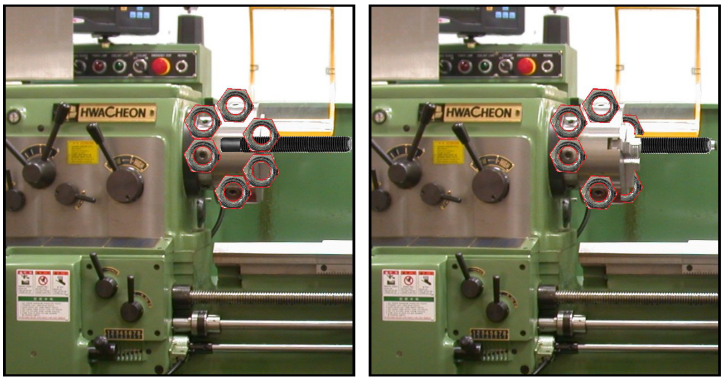


Figure 1: An example of the occlusion problem in an AR system for industrial maintenance purposes

The rest of the paper is organized as follows: Section 2 describes in detail the proposed AR authoring tool. Next, Section 3 shows different application examples of the proposed tool, and Section 4 shows the performance evaluation of AR instructions in an assembly task using the proposed tool. Finally, section 5 shows some concluding remarks and the future work to be done.

2. An overview of SUGAR

SUGAR (which stands for **S**ystem for the development of **U**nexpensive and **G**raphical **A**ugmented **R**eality application) is an open-source software platform designed to enable a rapid prototyping of low-cost AR systems. Our framework is oriented to develop complex AR software applications based on procedural simulations, which are modeled following an easy-to-use AR authoring editor. This AR editor generates an exchange file, describing the AR procedure, which can be loaded into different AR devices not requiring high computational power.

Figure 2 shows the workflow for the development of AR applications using SUGAR. The SUGAR editor allows users with non-programming skills the creation of augmented reality procedures based on steps. The main challenge when creating an Augmented Reality application is to achieve a correct overlap (registration) between the real-virtual world and the virtual information that is added on the real world. For tracking purposes, two kinds of planar markers (ARToolKitPlus or Vuforia) can be generated with a simple calibration step. The first markers consisting on square black and white fiducial ARToolKitPlus markers that the users must print and paste on the real object within the industrial environments. The second markers, based on Vuforia natural markers, are able to track planar surfaces of the real object and, thus, no additional items must be placed within the actual environment.

The software modules included in SUGAR can be classified in two groups: description of the real world, and virtual content edition. The first group includes those modules necessary for creating the scenarios where the virtual contents will be edited. Each scenario is composed of an image of the real environment, AR planar markers that are generated automatically, and a depth map (the latter one only is presented if a Kinect is available when the photo is taken). In order to create this scenario, two wizards guide the user through some easy to do steps. The first wizard, called the *PhotoKinect Wizard module*, allows taking photos storing at the same time the depth image. This depth information will be used later in order to produce correct occlusions. The other wizard, called *Locations wizard*, allows the user the creation of a scenario from either a conventional photo (2D scenario) or a photo captured with Kinect (3D scenario).

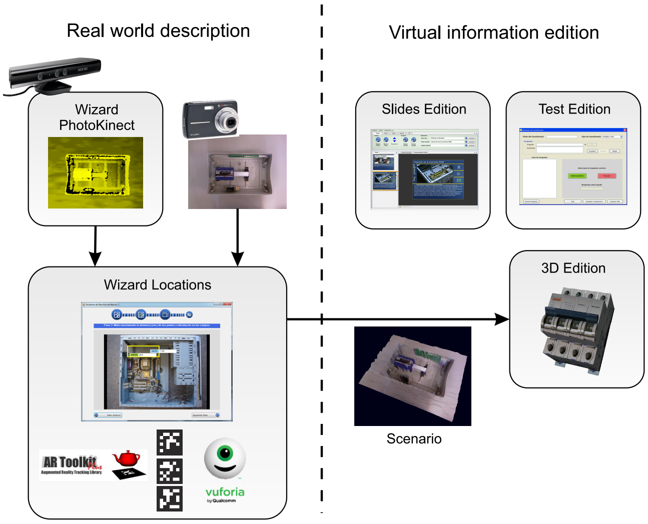


Figure 2: Proposed workflow for the development of AR applications using SUGAR

This simple wizard allows a user with neither programming, nor ARToolKitPlus/Vuforia knowledge, the creation of an AR marker with the proper size and the corresponding associated configuration file. The absence of this editor would require that the user selects the images of the markers to be used (he also locates the markers in the correct position) and he finally creates a configuration file with the description of the size, location and rotation matrix of each of the markers within the image.

After the creation of the scenarios, the other group of modules includes the procedures for defining the virtual information. The editor of SUGAR uses a structure based on steps, where each step is associated to one of the scenarios previously created. The edition of the virtual content can be split into three parts: definition of the current step of the procedure (denoted as slide), creation of the virtual content associated to each slide, and the definition of tests. The definition of the slides includes some basic office functions: creation, ordering, text edition, aspect, associated video, associated pdf files, etc. All these functions can be easily performed, in a similar way to the creation of a conventional graphic presentation.

For the creation of virtual content different kinds of 3D elements can be used. The virtual elements can be created either from basic 3D objects organized in an internal 3D library, or loading 3D models previously created using Autodesk 3D Max. The included 3D library consists of cubes, spheres, planes, cones, etc., which can be grouped to generate more complex models, and allows changing their textures, colors, or other properties. Also, we have developed an animation module based on keyframes that allows to animate the virtual objects. In the edition process, a previously created scenario is used as template to place the virtual objects. 2D scenarios provides the user with a clear reference of the position, size and orientation of all the objects located in the same plane of the marker, but it lacks depth information. The scenarios created from Kinect (3D scenarios) displays a 3D mesh so user has depth perception of the real objects. The absence of this template would force the user not only to accurately measure the location of each virtual element and its orientation on the real object according to the planar marker, but also requiring from the user a certain level of knowledge in 3D design, computational trigonometry and the use of the ARToolKitPlus or Vuforia libraries.

Finally, SUGAR has a test edition tool, named *Test Wizard*, that helps the user in the test edition process. This module can easily generate three kinds of questions: true/false questions, multiple answer questions, and selection on the image.

2.1. Software Architecture

The software architecture of SUGAR is based on a modular approach oriented to develop procedural AR systems. Basically, the SUGAR framework consists of two applications: an easy-to-use editor for AR procedures and an AR light viewer. Both applications share some software modules permitting them to reuse 3D-tracking data structure models, and visualization code.

Figure 3 shows the software architecture of the SUGAR framework for the development of AR environments. Both applications ("SUGAR Editor" and "SUGAR Viewer") share the kernel of the AR framework denoted as "Data Core". This kernel provides basic services for augmented reality facilities as camera tracking, marker handling and virtual object interaction.

Although AR editor and AR viewer share this software module, each application includes a different user's interface. Thus, the graphical user interface of the AR editor has been developed on Windows Forms and includes some components, developed on OpenSceneGraph. OpenNI software library allows accessing the depth map of the real scene using the Kinect device. Moreover, the Microsoft DirectShow API enables high-quality playback of streaming video and audio. The AR viewer corresponds to a light application, developed on C++ [3], embedding a reduced version of the OpenSceneGraph framework and including a reduced set of primitives for this 3D high-level library tool. The "SUGAR Viewer" has been ported to Symbian, IOS and Android operative systems with minimum changes within the initial source code.

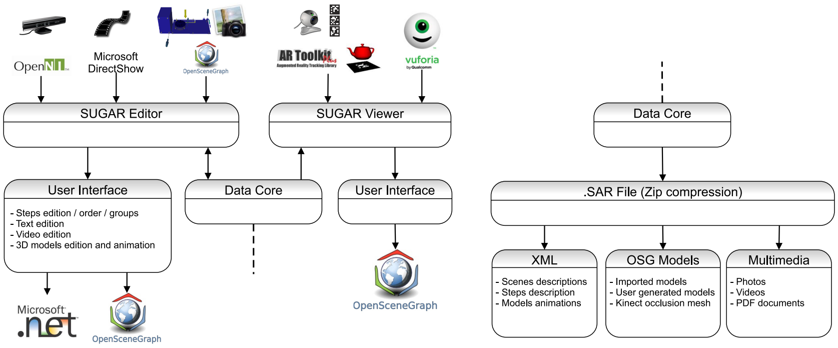


Figure 3: A modular view of the software architecture in SUGAR

The "Data Core" module also includes the definition and the basic properties of the exchange file format for the SUGAR framework, denoted as

SAR files. These files are generated by the AR editor to be imported by the SUGAR multiplatform viewers. Basically, the SAR files are zip-compressed archives containing a structured representation of AR applications, and the corresponding multimedia content, all of them defined in an included main XML file.

3. Application Examples

In order to validate our AR editor as an efficient tool for the rapid prototyping of AR applications for assembly, maintenances and repair purposes, we have tested our tool in four different application examples belonging to different industrial areas. Concretely, these application examples are the following procedures: the replacement of the cut heading within a lathe machine (metal machining area), the assembly of a computer starting from its basic components (computer and electronics manufacturing area), the repair of the admission system in a mobile lighting tower (maintenance of heavy machinery area), and the review of the spark plugs and the ignition coils on a BMW M3 E92 (420CV) engine (automobile maintenance area). We have denoted these procedures as *PROC1* to *PROC4*, respectively. Figure 1 is actually a picture of procedure PROC1. Figure 4 shows different images of the assembly of a computer (PROC2) displayed on different mobile devices. Concretely, the central image in this figure shows the execution of the AR displayer on an Apple Ipad 2. The images on the right side show the execution of the AR displayer on a Samsung Galaxy S2 (top) and Apple Iphone 4 (bottom).

Figure 5 show a snapshot of SUGAR in the edition of the AR system for the repair tasks and assembly of a mobile lighting tower (PROC3). Finally, figure 6 shows a picture of procedure PROC4.

Additionally, in this paper we study the effects of the proposed scenarios in the edition processes as described in Section 2. The proposed tool allows to create augmented scenarios either from conventional photos (2D scenario) or by using Kinect (3D scenario). We have added a third type of 3D edition process for evaluation purposes, where the user can see the maker created by the wizard but he cannot see the 2D/3D scenario used as a reference. This type of edition process is used to study the effects of the 2D or 3D references in the authoring process. As an example, Figure 7 shows the edition of a concrete step in the assembly of a computer by using each one of the three types of edition processes considered. The image on the upper right corner of

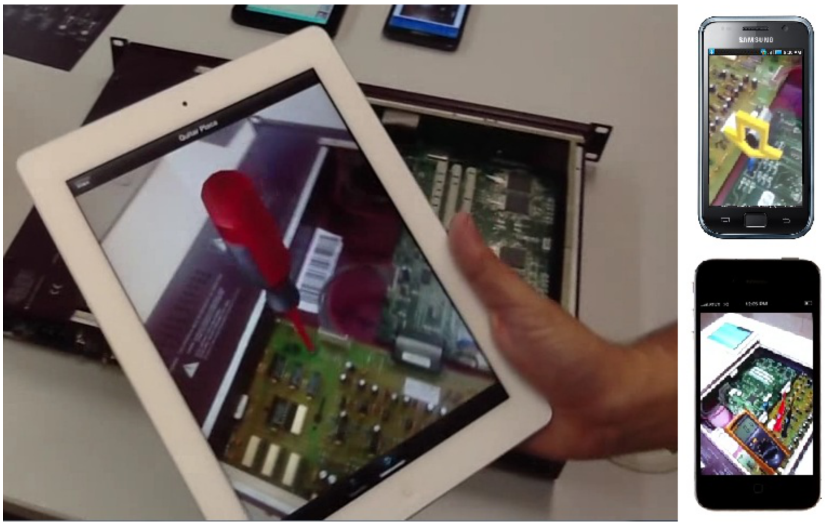
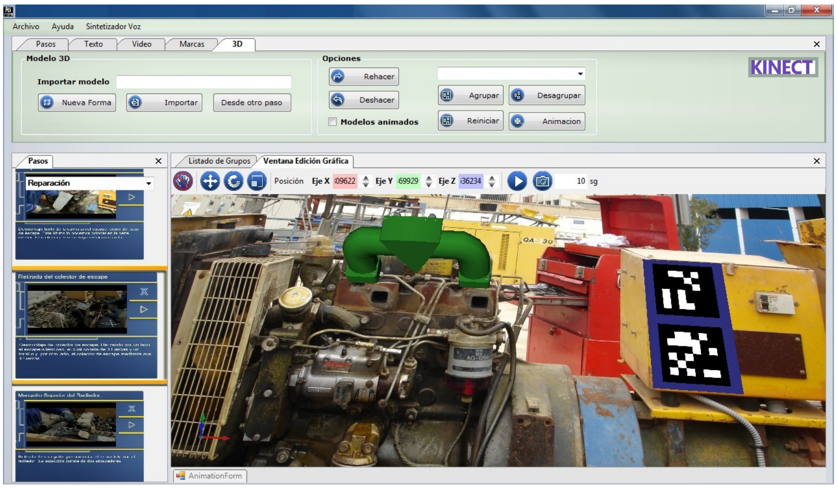


Figure 4: Examples of SUGAR snapshots for the assembly of a computer (PROC2).



Posición Eje X 09622 Eje Y 69929 Eje Z 36234 10 sg

Figure 5: Examples of SUGAR snapshots for the repair tasks in the engine of a mobile lighting tower (PROC3).

the figure corresponds to the edition from a conventional photo. The lower image corresponds to the edition using Kinect, and the image on the upper left corner corresponds to the third type edition process.

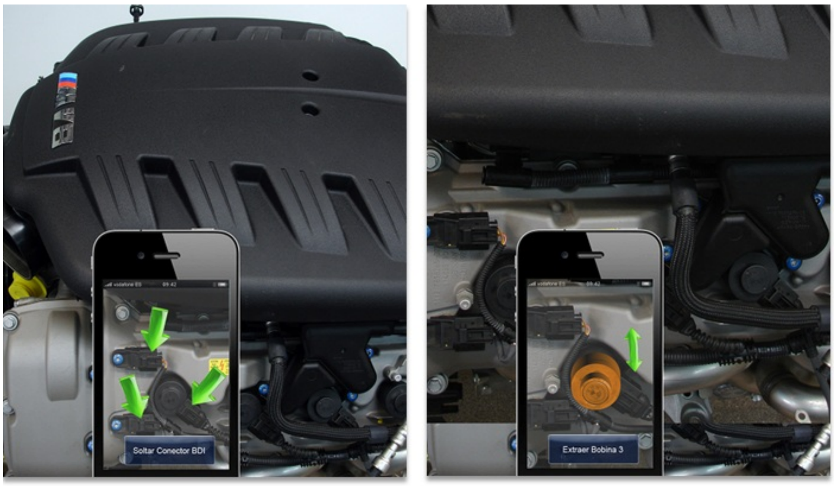


Figure 6: Examples of SUGAR snapshots for the maintenance tasks in a motor engine (PROC4).

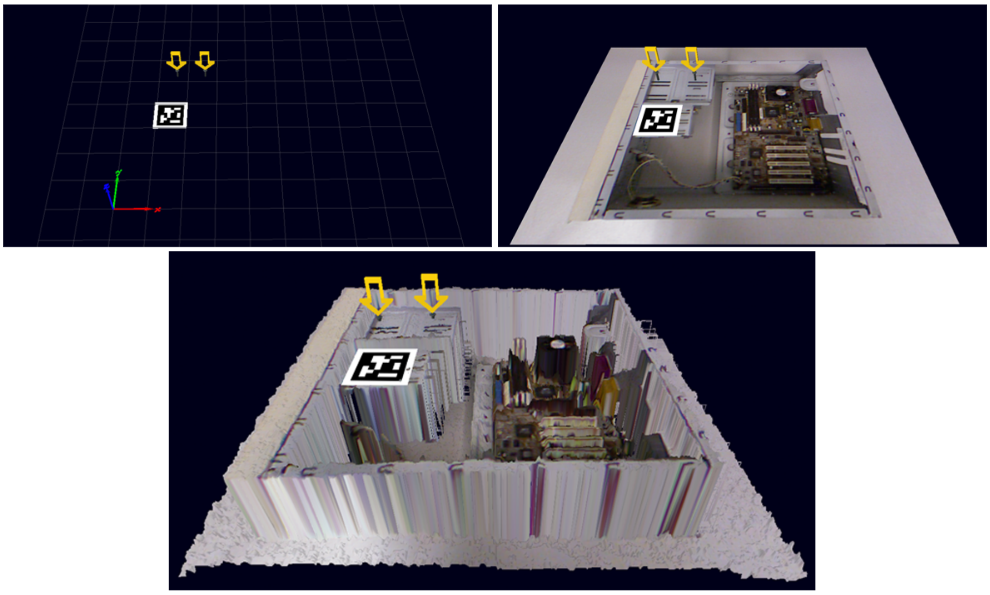


Figure 7: A snapshot of the three edition processes considered when used in the procedure oriented to the assembly of a computer (PROC2).

4. Performance Evaluation

The performance evaluation of augmented reality authoring tools results in a complex task. Although there are some works that show quantitative measures of maintenance and repair (or related) tasks [7, 13, 18, 30], there is a lack of standards that classify the quality of Augmented Reality or Virtual Reality tools according to quantitative measurements. On other hand, most of the software engineering studies [32] show a qualitative and even a fuzzy nature.

A possible way of evaluating the performance of Augmented Reality tools is a qualitative, user-centered approach [35]. According to recent studies [24], the observations and questionnaire are the basis for a qualitative analysis. The questionnaire consists of several questions where the participants in the evaluation of the AR tool can freely answer on their experience of the AR system. However, a qualitative approach does not allow neither to compare different AR tools on a fair basis, nor to evaluate their performance in comparison with traditional training or guidance tools. On the contrary, a quantitative analysis can allow a fair comparison of different tools. Some metrics like cost efficiency, development time and maintainability have been proposed for a quantitative analysis [1]. However, that work does not define concrete criteria for assigning values to the three metrics, and only the development time is accurately measured. Due to these reasons, in this work we have measured both quantitative and qualitative variables.

We propose a **quantitative** approach for the performance evaluation of the AR tool. In order to measure the complexity of the repair and maintenance tasks, we have followed the criteria proposed in [6]. In order to measure the performance provided by the AR systems prototyped with SUGAR, we have first measured the average completion time required by fifteen different users in order to completely execute each of the considered procedures. The completion time was measured following two different approaches, depending on the type of experiments described in the paper. In this sense, a manual digital chronometer (SEIKO S141) was used to measure the time required by the users to complete each of the experiments. On the other hand, we have included a timing control module in SUGAR which is automatically activated when users create and edit AR procedures. This timing control module was programmed using Multimedia Timers, a high-resolution timer API under Windows 32/64 bits, which have a one millisecond resolution.

All the users were experts technicians in their area, but a training session was performed prior to the performance evaluation in order to describe the procedures as well as to allow the users to get in contact with the AR technology. In order to avoid both skews in the results due to the different skills of different populations and the potential training with the technology, the users were randomly selected (for each procedure) to complete a given procedure using a AR system equipping a head-mounted device or a monitor display. The random selection of the AR devices ensured that neither the training session, nor the learning effect could bias the results. The evaluation sessions were performed in the real environments at a rate of one experiment per day, in order to avoid fatigue or possible learning effects. Some weeks later, after completing all the first executions of the procedures, the users also performed a very similar procedure exclusively using a printed manual provided by the manufacturer. In that second evaluation session, some random variations (including a new order of the steps in the procedure as well different location of the 3D elements in some AR steps) where included in the original procedures. In all the cases, it was the first time that the users executed the considered procedures. We have denoted as S1 the ”system” consisting of exclusively using a printed manual provided by the manufacturer. We have denoted as S2 the system consisting of a monitor display, and we have denoted as S3 the system consisting of computer assisted instructions (usually abbreviated as CAI) using a head-mounted display.

Figure 8 shows the average completion times and time quartiles (measured in minutes) required for all the procedures considered when using each system. We have represented the mean time as a blue rhombus. The ANOVA analysis confirms that the mean time values are significantly different ($p < 0.0001$). As it could be expected, the average completion times for systems S2 and S3 are much lower than the ones achieved with S1 system. Also, this table shows that the times achieved with S3 are lower than the ones achieved with S2, reaching even less than half the time required for the same procedure with system S1 (in Procedures 1, 3 and 4). These results show the significant benefits that AR systems can provide to repair and maintenance tasks.

For the **qualitative** (or subjective) evaluation of the authoring tool process, the formal hypothesis we want to prove is the following one: in order to create the steps of an industrial procedure with the proposed authoring tool, the highest level of usability is achieved if Kinect (3D scenario) is used.

In order to prove this hypothesis, we prepared a questionnaire where the main usability parameters were rated on a 7-point Likert-type scale, ranging from 1 (strongly disagree) to 7 (strongly agree), for three different versions of the editor. A Likert scale [8] is a psychometric scale based on questionnaires, and it is considered nowadays as the most widely used approach to scaling responses in survey-based research. Concretely, we selected and include in

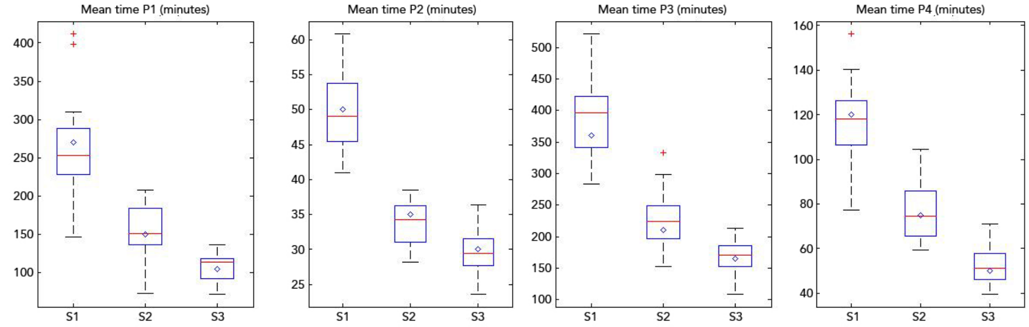


Figure 8: Completion times with different systems (Lower Quartile, Mean and Upper Quartile).

these questionnaires parameters such as learnability, efficiency, memorability, accuracy error and subjective satisfaction as the five main usability characteristic for this type of interactive application [23]. For evaluation purposes in this qualitative evaluation, we proposed the use of our tool to a group of multidisciplinary users in the field of the industrial maintenance and assembly, composed of thirty users with different level of knowledge in 3D edition processes. They had to create steps of the considered procedures using the three alternatives considered: using no references, starting from a 2D photo (2D scenario) and by using Kinect (3D scenario). In order to remove the dependency on the order of use of the different alternatives, we propose a different order to six different groups of users. As a previous step to the creation of the procedures steps, users undertook a training consisting of a 10-minutes tutorial about the use of the authoring tool and a simple edition exercise, consisting of stacking three cubes (red, green and blue) 10 cm. wide. The time and accuracy in the execution of this control task were used as a reference for evaluating the results in the creation of more complex scenes.

Figure 9 shows the evaluation results obtained by thirty users with the three types of edition processes, as well as the control task. The image on the left side shows the average value and time quartiles (measured in seconds) required for placing the elements in the augmented scenes. The image on the right side shows the average value and position error quartiles (measured in millimeters) of the virtual objects placed by the users.

Figure 9 shows that the editor with the 3D scenario provides the best results in both time and position error, while the 2D version still provides better results than the editor with no reference scenarios. Thus, the average

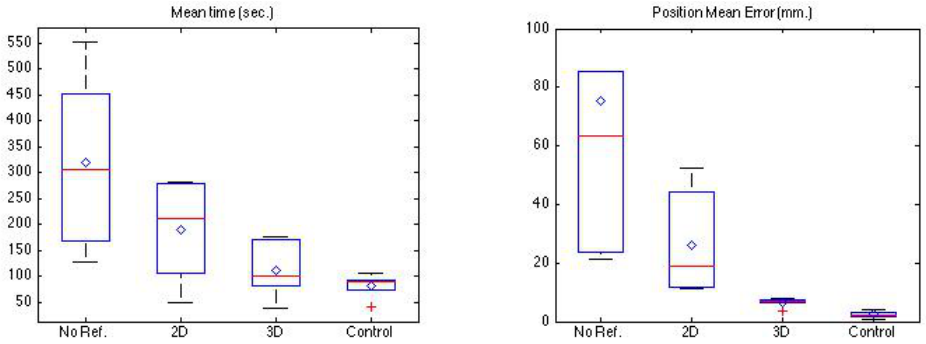


Figure 9: Mean time and Position Error (Lower Quartile, Mean and Upper Quartile).

time required for placing each element is 111.67 sec. in the 3D version, close to the time required for the control task (82.38 sec.), and around one third of the one required for placing each element in the version without reference images (318.26 sec.). Regarding the accuracy, the position error in the 2D version is 26 mm., lower than half the error in the version without reference images, that was 75 mm..However, the position error in the 3D version was 6.5 mm., one order of magnitude lower than the version without reference images and very close to the values of 2.4 mm. measured in the control task. In both metrics (images) it can be seen that the dispersion of the values obtained in the 2D version and the version without reference scenarios is much higher than the dispersion of the values obtained in the 3D version. This dispersion is due to the fact that in the former ones the user should either interpret the plane of the real scene or performing measurements in the layout plan in order to correctly placing the pieces. Therefore, his performance depends on the skill (or previous training) of the user in working with layout plans. However, the 3D version does not require from the user to make measurements in planes, since it was possible to directly place the object in the reference 3D mesh. These results show that augmented reality edition system based on 3D scenarios can provide good results quickly, regardless the previous skills of the user.

Figures 10 to 14 show a histogram of each of the usability parameters in the Likert scale (values from 1 to 7 points) [23]. These figures show different patterns for each of the systems considered. The 3D scenarios version obtains values above 15 in the ”Agree” answer for all the figures, being this answer the most selected one in all the figures. The 2D version obtains the highest values in the "Agree Somewhat" answer in efficiency, memorability, and satisfaction, and the highest values in the "Undecided" answer in accuracy and learnability. Finally, the version with no reference images obtains the highest values in the "Agree Somewhat" answer in learnability and memorability; it obtains the highest values in the "Disagree somewhat" answer in efficiency, and it obtains the highest values in the "Disagree" answer in accuracy and satisfaction. Additionally, it is the only systems that obtains "Strongly disagree" answers in some parameters.

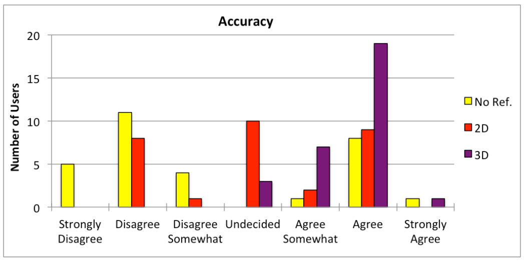


Figure 10: Accuracy histogram.

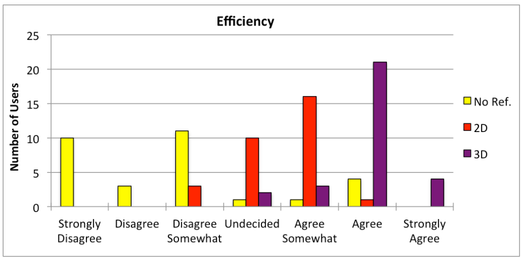


Figure 11: Efficiency histogram.

Finally, Table 1 shows the percentage of selection of the possible editor versions in seven statements presented in the questionnaire. The vast majority of users have chosen the 3-D version of the AR editor in all the questions, showing a full percentage in the case of the questions "would you use it in the future" and "general satisfaction". These results validate this version of SUGAR as the most adequate AR authoring tool for procedures where depth perception is needed.

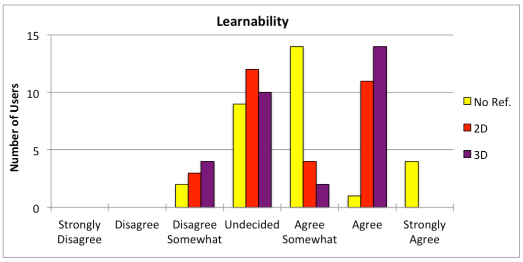


Figure 12: Learnability histogram.

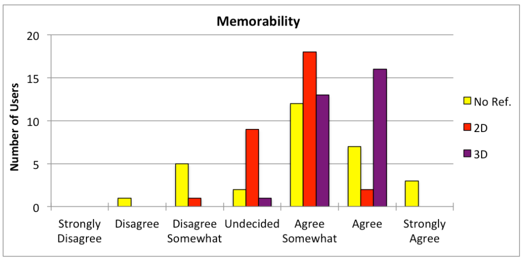


Figure 13: Memorability histogram.

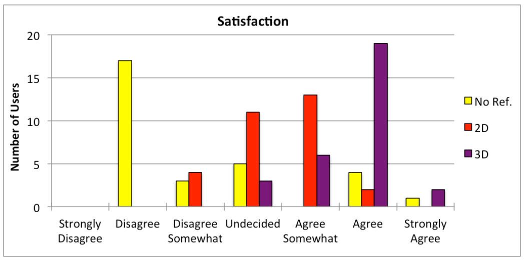


Figure 14: Satisfaction histogram.

5. Conclusions and Future Work

In this paper, we have proposed an easy-to-use AR authoring tool, which allows the easy creation of interactive augmented reality applications without any programming knowledge. This authoring tool includes an easy-to-use editor for AR procedures and an AR light viewer, which share non-proprietary exchange files describing the AR procedures. The proposed tool allows the development of AR applications with occlusion capabilities to be used in on-site industrial procedures, where a certain level of depth-perception is necessary. Unlike other recent proposals, our tool does not rely on expensive

	No Ref.	2D	3D
I would use in the future	0	0	100
Need less previous knowledge	16,6	0	83,4
More intuitive	0	6,6	93,4
More accurate	6,6	9,9	83,5
Faster	0	0	100
I've felt more comfortable	0	13,3	86,7
General Satisfaction	0	0	100

Table 1: Percentage of selection of each editor version.

or unavailable 3D models, and it uses Kinect for computing a depth map of the scene.

The performance evaluation of our AR authoring includes an assessment of the authoring process from four AR applications, belonging to different industrial areas. The performance evaluation results show that when using Kinect to create AR industrial procedures, the completion time to generate them, as well as the location error of the augmented elements significantly decrease. In these sense, non expert users in 3D AR tools have successfully designed AR procedures with high visual and functional quality, due to the ease of placing the 3D virtual objects provided by the 3D mesh representing the real scenario. This new edition mode, which combines depth visual information provided by the Kinect device with multimedia content (video, images, text, voices, etc) complement an innovative AR authoring tool oriented to the general purpose in industrial environments. Unlike other proposals in this context, this AR authoring tool allows non-expert users to create complex AR industrial procedures including high precision and elevated level of visual detail.

As a future work, we plan to replace Kinect as a data source for the creation of 3D scenarios, with mapping algorithms based on 3D natural features. Concretely, we are working on some algorithmic alternatives based on SLAM (Simultaneous Localization and Mapping) or PTAM (Parallel Tracking and Mapping) approaches, where a probabilistic filtering is used to recursively estimate a full probability density over the current camera pose and the positions of features. Thus, the application contexts will not be limited by the constraints of Kinect, in terms of data precision and detection zone size, and therefore lager industrial scenarios could be augmented more accurately.

Acknowledgements

This work has been jointly supported by the Spanish MICINN and the European Commission FEDER funds under grants Consolider-Ingenio CSD2006-00046 and TIN2009-14475-C04-04.

6. Bibliography

[1] Abawi, D. F., Luis, J., Arcos, L., Haller, M., Hartmann, W., Huhtala, K., and Trskbck, M. (2004). A mixed reality museum guide: The challenges and its realization. In *Proceedings of the 10th International Conference on Virtual Systems and Multimedia (VSMM 2004)*.

[2] Azuma, R. (1997). A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385.

[3] Blanchette, J. and Summerfield, M. (2008). *C++ GUI Programming with Qt 4*. Open Source Software Development Series. Prentice Hall.

[4] Breen, D. E., Whitaker, R. T., Rose, E., and Tuceryan, M. (1996). Interactive occlusion and automatic object placement for augmented reality. *Computer Graphics Forum*, 15(3):11–22.

[5] Burns, D. and Osfield, R. (2004). Open scene graph a: Introduction, b: Examples and applications. In *Proceedings of the IEEE Virtual Reality Conference 2004 (VR 2004)*, page 265.

[6] Campbell, D. J. (1988). Task complexity: A review and analysis. *Academy of Management Review*, 13(1):40.

[7] Curtis, D., Mizell, D., Gruenbaum, P., and Janin, A. (1999). Several devils in the details: making an ar application work in the airplane factory. In *Proceedings of the international workshop on Augmented reality: placing artificial objects in real scenes: placing artificial objects in real scenes*, pages 47–60. AK Peters, Ltd.

[8] DeVellis, R. F. (2011). *Scale Development: Theory and Applications (Applied Social Research Methods), 3rd. Ed.* SAGE Publications Inc.

[9] Friedrich, W. (2002). Arvika-augmented reality for development, production and service. In *In Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR02),*, pages 3–4.

[10] Hampshire, A., Seichter, H., Grasset, R., and Billinghurst, M. (2006). Augmented reality authoring: generic context from programmer to designer. In *In Proceedings of the Australasian Computer-Human Interaction Conference (OZCHI'06)*, pages 409–412.

[11] Hansen, C., Wieferich, J., Ritter, F., Rieder, C., and Peitgen, H.-O. (2010). Illustrative visualization of 3d planning models for augmented reality in liver surgery. *Int. J. Computer Assisted Radiology and Surgery*, 5(2):133–141.

[12] Haringer, M. and Regenbrecht, H. T. (2002). A pragmatic approach to augmented reality authoring. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR '02, pages 237–, Washington, DC, USA. IEEE Computer Society.

[13] Henderson, S. and Feiner, S. (2011). Exploring the benefits of augmented reality documentation for maintenance and repair. *Visualization and Computer Graphics, IEEE Transactions on*, 17(10):1355–1368.

[14] Henderson, S. J. and Feiner, S. (2009). Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret. In *Proc. of 8th IEEE Int.l Symp. on Mixed and Augmented Reality*, ISMAR '09, pages 135–144. IEEE Computer Society.

[15] Holmquest, L. (2013a). Starting to develop with kinect. Available at http://msdn.microsoft.com/en-us/magazine/jj159883.aspx.

[16] Holmquest, L. (2013b). Working with kinect studio. Available at http://msdn.microsoft.com/en-us/magazine/jj650892.aspx.

[17] Kato, H. and Billinghurst, M. (1999). Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of International Workshop on Augmented Reality (IWAR'99)*, pages 85–94.

[18] Kleiber, M. and Alexander, T. (2011). Evaluation of a mobile ar telemaintenance system. *Universal Access in Human-Computer Interaction. Applications and Services*, pages 253–262.

[19] Knopfle, C., Weidenhausen, J., Chauvigne, L., and Stock, I. (2005). Template based authoring for ar based service scenarios. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, pages 237 –240.

[20] Ledermann, F. and Schmalstieg, D. (2005). April: A high-level framework for creating augmented reality presentations. In *In Proceedings of the IEEE Virtual Reality Conference 2005 (VR 2005)*, pages 187–194.

[21] MacIntyre, B., Gandy, M., Dow, S., and Bolter, J. (2005). Dart: a toolkit for rapid design exploration of augmented reality experiences. *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH 2005*, 24(3).

[22] Neumann, U. and Majoros, A. (1998). Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. In *In Proceedings of the IEEE Virtual Reality Annual International Symposium (VR '98)*, pages 4–11.

[23] Nielsen, J. (1993). *Usability Engineering*. Morgan Kaufmann.

[24] Nilsson, S., Johansson, B., and Jnsson, A. (2010). *The Engineering of Mixed Reality Systems*, chapter A Holistic Approach to Design and Evaluation of Mixed Reality Systems, pages 33–55. Human-Computer Interaction Series. Springer.

[25] P. Sang-Cheol, L. Sung-Hoon, S. B.-K. S. and Seong-Whan, L. (2005). Tracking non-rigid objects using probabilistic hausdorff distance matching. *Pattern Recognition*, 38(12):2373–2384.

[26] Poupyrev, I., Tan, D., Billinghurst, M., Kato, H., Regenbrecht, H., and Tetsutani, N. (2001). Tiles: A mixed reality authoring interface. In *Proceedings of Conference on Human-Computer Interaction (INTERACT '01)*, pages 334–341.

[27] Regenbrecht, H., Baratoff, G., and Wilke, W. (2005). Augmented reality projects in the automotive and aerospace industries. *Computer Graphics and Applications, IEEE*, 25(6):48–56.

[28] Schmalstieg, D. (2005). Rapid prototyping of augmented reality applications with the studierstube framework. In *In Proceedings of the Workshop*

of Industrial Augmented Reality (IAR), IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR05).

[29] Schwald, B., Laval, B. D., Sa, T. O., and Guynemer, R. (2003). An augmented reality system for training and assistance to maintenance in the industrial context. In *In Proceedings of 11th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pages 425–432.

[30] Schwerdtfeger, B., Reif, R., Günthner, W., and Klinker, G. (2011). Pick-by-vision: there is something to pick at the end of the augmented tunnel. *Virtual reality*, 15(2):213–223.

[31] Seichter, H., Looser, J., and Billinghurst, M. (2008). Composar: An intuitive tool for authoring ar applications. In *In Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR08)*, pages 177–178.

[32] Seo, J. and OhM, S. (2007). Pvot: An interactive authoring tool for virtual reality. *International Journal of Computer Science and Network Security (IJCSNS)*, 7(4):17–26.

[33] Tang, A., Owen, C., Bioca, F., and Mou, W. (2003). Comparative effectiveness of augmented reality in object assembly. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI'03)*, pages 73–80.

[34] Tillon, A., Marchal, I., and Houlier, P. (2011). Mobile augmented reality in the museum: Can a lace-like technology take you closer to works of art? In *Mixed and Augmented Reality - Arts, Media, and Humanities (ISMAR-AMH), 2011 IEEE International Symposium On*, pages 41–47.

[35] Traskback, M., Koskinen, T., and Nieminenl, M. (2003). User-centred evaluation criteria for a mixed reality authoring application. In *Proc. of Tenth International Conference on Human-Computer Interaction (HCI)*, pages 1263–1267.

[36] Wagner, D. and Schmalstieg, D. (2007). Artoolkitplus for pose tracking on mobile devices. In *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, pages 139–146.

[37] Wang, Y., Langlotz, T., Billinghurst, M., and Bell, T. (2009). An Authoring Tool for Mobile Phone AR Environments. In *New Zealand Computer Science Research Student Conference (NZCSRSC 2009)*, volume 9, pages 1–4.

[38] Yeh, M., Merlo, J. L., Wickens, C. D., and Brandenburg, D. L. (2003). Head up versus head down: The costs of imprecision, unreliability, and visual clutter on cue effectiveness for display signaling. *Human Factors*, 45(3):390–407.

[39] Zhang, Z. (2012). Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10.