Interaction, Collaboration and Authoring in Augmented Reality Environments

Claudio Kirner
Federal University of Ouro Preto
ckirner@gmail.com

Rafael Santin
Federal University of Jequitinhonha and Mucury Valeys
rasantin@gmail.com

Abstract

This paper describes the design of an augmented reality authoring system for end-users (ARAS-EU), focusing interaction, collaboration and authoring in augmented reality applications, showing the strategies adopted in each case. Interaction was emphasized in all levels of the system, resulting in the presentation of taxonomy of selection, manipulation and release techniques applied to augmented reality environments. We discuss the implementation of the system and an application related to the assembly process of a virtual helicopter in an augmented reality environment. We also considered aspects of the system structure and its support to work as a collaborative application based on distributed system.

1. Introduction

Augmented reality can be defined, in a general way, as an enrichment of the real world with virtual objects, using technological devices. According to Azuma et al. [3] [4], an augmented reality system must have three characteristics: combines real and virtual, is interactive in real time and is registered in 3D. Augmented reality can also be defined as the overlapping of virtual objects with the physical environment, being shown to the user in real time, using the real environment interface adjusted for visualization and interaction with real and virtual objects [10]. The virtual elements superimposed to the real world go beyond simple objects and annotations. They can involve visible and invisible points in space, virtual objects with behavior, visual and audible cues, etc.

The interactions, on the other hand, continue to be simple and intuitive, once they are executed by the user with his hands actuating at the object positions. These tangible operations, such as touch, grab, move and release are used to manipulate real and virtual objects in physical space. Besides, the system also allows the exploration of new user interactions with virtual objects, resulting in: changing features (color, light, shape and size), replication, animation, vanishing, appearing, information extraction, etc. Hence, the user interaction with objects (real and virtual) placed in the augmented reality environment assumes new functionalities and improves the power of applications.

However, the applications need to be authored before using. In many cases, the authoring and using are based on different environments, requiring users with programming skills and/or knowledge of specific tools. In other cases, the system allows intuitive and tangible user actions to develop only simple applications [7].

One way to solve this problem is to use an augmented reality system whose applications can be authored and used by end-users. In the case of more complex applications, the main authoring can be executed by an expert user who prepare the environment, so that the complementary authoring can be carried out by end-users using configuration procedures. Adopting this strategy, this paper describes the design of an augmented reality authoring system for end-users (ARAS-EU), containing simple and complex virtual objects, including behavior. The user interactions in the augmented reality environment are described and classified by a proposed taxonomy. This paper also describes the implementation of an application, showing some interactions being executed, including cues and their impacts on the environment behavior.

2. Related work

The development of augmented reality applications usually requires expert users with programming skills and/or experience on specific tools. Toolkits, frameworks and authoring tools have been used to develop augmented reality applications, involving programming, visual tools and the proper augmented reality environment as authoring tool [7].

ARToolKit [1] is a toolkit containing a library whose authoring process is based on a program calling modules.

Studierstube [16] is a framework that supports complex configuration of resources and programming to develop augmented reality applications.

AMIRE (Authoring Mixed Reality) [8] is a framework that uses visual programming and components to build augmented reality applications.

DART (Designers Augmented Reality Toolkit) [13] is an authoring tool based on Macromedia Director, and works as a multimedia authoring tool, involving pre-built elements.

TILES [15] is a mixed reality authoring interface that uses interaction techniques for easy and spatial composition and allows seamless two-handed 3D interaction with both virtual and physical objects.

The system iaTAR (immersive authoring Tangible Augmented Reality) [11] is an augmented reality authoring tool based on components, behaviors and intuitive interaction that allows developers to create augmented reality scenes from the application. The user does not need to change modes between authoring and viewing the application.

OSGAR [12] is a framework for rapid application development of mixed reality environment, using ARToolKit and OpenSceneGraph. It works with external bindings, providing the possibility to use scripting languages to obtain behaviors.

In the context of these researches, focusing on programming, configuration and tangible actions, we developed an augmented reality authoring system for end-users (ARAS-EU) based on different levels of application configuration, keyboard actions and tangible operations. ARAS-EU allows transitional authoring/viewing and works with different parts of the mixed environment, points and sets of virtual objects, involving behaviors, sounds and paths. Besides, the system was structured and implemented to be an augmented reality collaborative environment for remote users.

3. The augmented reality authoring system for end-users (ARAS-EU)

An augmented reality environment can be much more than the augmentation of the real world with simple virtual objects placed into it. The mixed world can have: interactive objects, which change in certain situations; intelligent objects, which assume behavior dependent on the context; visible or invisible objects, which vanish or appear under certain circumstances, etc.

Moreover, the augmented reality environment can be modified by creation, changing and deletion of elements such as spatial positions, virtual objects, behavior, visual and audible cues, etc.

This type of augmented reality environment, built by an augmented reality authoring system for end-users (ARAS-EU), is represented in Figure 1, which also shows the data structure used to implement the environment with its points and virtual objects. The data structure is organized in a hierarchical way, so that a root (Reference) contains points and virtual objects associated with them (Figure 2). The environment can support many REFs, extending the manipulation of a specific REF to its respective virtual elements.

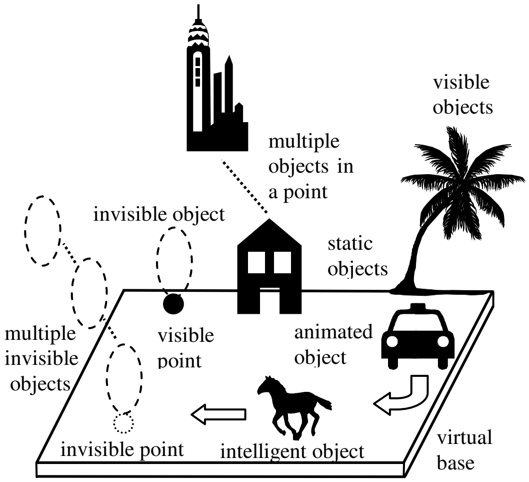

Figure 1. Representation of an augmented reality environment.

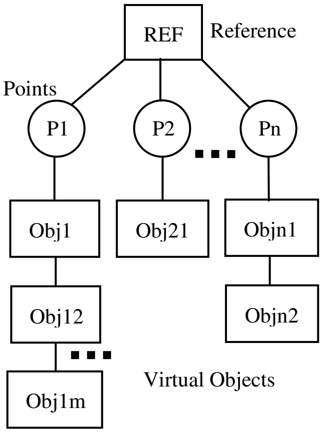

Figure 2. Data Structures of an augmented reality environment.

The life cycle of an augmented reality environment is based on three system phases: creation, utilization and finalization, according to Figure 3.

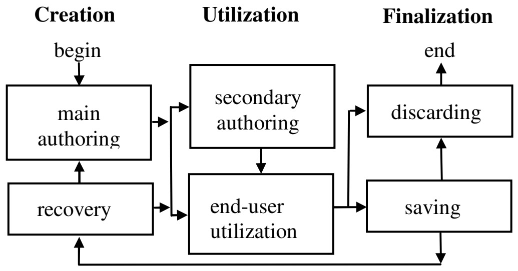

Figure 3. Life cycle of an augmented reality environment built by ARAS-EU.

3.1. Creation phase

In the creation phase, the user can start a project, executing the main authoring task required for the building of the augmented reality environment. To do that, the user adopts a virtual base associated with a main reference known by the system (a predefined real object or mark). After that, the points (spatial positions) where the virtual objects will be placed are positioned through a visual placement process creating their own local references. Another alternative is opening the point file in the system to insert the point positions by hand, using a text editor.

The user can associate virtual objects with points, using a visual process or editing the points file, letting the virtual objects visible or invisible. Moreover, the user can pick up virtual objects from a catalog, putting them on the virtual base and creating automatically the respective points.

In each point, a virtual object with its sound or a list of virtual objects and sounds can be associated with it.

These virtual objects and sounds can be selected for visualization in the authoring phase or in the utilization phase. If the virtual object is animated, it will use the point position as a reference for the animation. If the virtual object is intelligent, it will have an associated program that will examine the environment context, including the other virtual objects, to assume its behavior in each instant, during the utilization phase.

If the user wants to stop the authoring phase to continue it later, he can go to the finalization phase to store the system state in a file. Later, it will be possible to continue the authoring phase, recovering the state of an unfinished authoring.

3.2. Utilization phase

In the utilization phase, the user manipulates the augmented reality environment, using his hands. The idea in this phase is to minimize the dependency of the user from any device to manipulate the environment, so that this activity can be as intuitive as possible.

However, aiming to give more power to the system, it is possible to reconfigure the environment during the utilization phase through the execution of a secondary authoring. In this situation, the user can: change the visibility of points and virtual objects; exchange the current virtual object by another from the list associated with the point; make copies of virtual objects; delete points and virtual objects; etc. These actions require less command in comparison with the other ones used in the main authoring.

After the environment configuration, the user can utilize his hands or some auxiliary device to manipulate virtual objects. Proximity, touching or some action on virtual objects or points of the augmented reality environment begin the manipulation, followed by reactions and behaviors from the system. Movement and other actions continue the manipulation, which ends when the point or virtual object is released.

At any time, the user can execute a new configuration to change the environment and its behavior, making it unforeseen or personalized.

3.3. Finalization phase

After the utilization of the application, the user can discard it or save the final state of the system to be continued later. In this case, the system will ask the user to enter a filename or will use a default filename. This saving process can occur even when the application is not finished.

3.4. Collaborative support

ARAS-EU was structured to work as a distributed system, allowing remote collaborative applications. In this way, a shared data structure (remote reference) was conceived to remain connected and upgraded on all collaborative computers in a network. All collaborative users can see and manipulate points and virtual objects associated with the remote reference (Figure 4), which is represented by an element of the environment. The other references, which are not shared, work as local references and are private to local users. The users can interchange points and virtual objects between remote and local references.

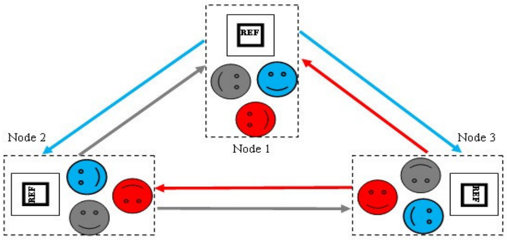

Figure 4. Remote users with a remote REF marker showing the shared virtual objects.

Figure 5 shows the visualization of a disassembled helicopter from points of view of two nodes. The green small plate (at the left) represents a remote reference, which allows sharing of the environment (large plate with the helicopter parts) and collaboration actions moving parts to assembly the helicopter by the two remote participants.

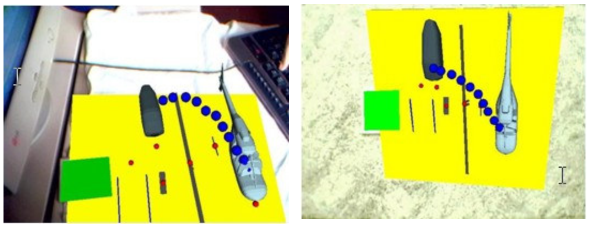

Figure 5. User's view from two nodes.

Other interactive actions were developed to support the collaborative work, such as lock and unlock. These actions are useful to keep unchanged certain points and virtual objects, assuring that only the owner can manipulate them.

4. Interaction

The user interaction with virtual objects in augmented reality environments can be classified by many ways depending on devices [14], visualization mode [10], etc. In this paper, the analysis of interactions is carried out by the point of view of the visualization mode.

4.1. Types of interaction

There are two types of user interaction based on the visualization mode: direct interaction and indirect interaction.

4.1.1. Interaction with direct visualization

The interaction with direct visualization occurs when the user sees directly the object that he is manipulating through a device coupled with him and aligned with his eyes.

This happens with the use of see-through HMD (optical or by video). This case is based on interaction in first-person and more intuitive tangible actions [9].

4.1.2. Interaction with indirect visualization

The interaction with indirect visualization takes effect when the user sees the augmented reality environment on a visualization device not coupled with him and not aligned with his eyes. This happens with the use of video camera to capture images, which are shown in monitor, projection or other device not coupled with the user. This case constitutes the visualization in third-person. It implements less intuitive tangible actions.

4.2. Techniques of selection, manipulation and release in augmented reality environment

Techniques of selection, manipulation and release in virtual reality environments were explored by Bowman et al. [5] [6], who proposed taxonomy of these interaction techniques.

However, the virtual reality environment offers different conditions of interaction, since the user is taken from the real world, being placed in virtual world to perform his actions. In this virtual environment, the user has freedom and power to interact with virtual objects and with the whole environment – he can be as big or small as he wants, as fast or slow as he wants, as strong or weak as he wants.

On the other hand, in the augmented reality environment, the restrictions of the user and physical space impose limits to distance, shape, speed, etc.

Thus, considering the characteristics of the ARAS-EU, the taxonomy of interactions in virtual reality, presented by Bowman et al. [5] [6], was adapted for augmented reality environments and is discussed in the following.

The reference scene is an augmented reality environment with points and virtual objects, complemented with feedback, cues and other actions (presence/absence of points and virtual objects, for example).

4.2.1. Selection

The selection is useful to choose a point or virtual object in the augmented reality environment. To do that, a point or virtual object must be firstly indicated. After that, the user issues the selection command, receiving a feedback.

The indication of a point in the augmented reality environment is executed using another point coupled with the user hand. When a point touches the other, a visual and/or audible feedback occurs and the user can command the selection. The indication of a virtual object is executed in a similar way, using the point associated with it.

To make this task easier, points and virtual objects are represented by invisible spheres, whose radius can be adjusted by the user, giving a type of resolution to the process. To select virtual objects placed close to others, the user must decrease the selection radius to obtain more precise actions. For virtual objects spread in the augmented reality environment, the user can increase the selection radius to become fast.

The selection command can be issued by time, gesture, keyboard, voice or haptic action.

The time command occurs when the user perceives the selection feedback, keeping this situation during some time until the selection is done. If the user changes the position before that time, the selection is cancelled.

The gesture command can be activated by moving the point coupled to the user's hand in a way that is recognized by the system.

The keyboard command is issued by pressing a specific button on the keyboard when the selection is activated.

The voice command occurs when the user informs the specific command to the system after the selection.

The haptic command is issued by pressing or depressing some sensor after the selection.

4.2.2. Manipulation

Manipulation is an action executed on a previously selected point or virtual object in the augmented reality environment. Besides the conventional moving operations, there are also other possible operations on a point or virtual object including presence/absence, changing characteristics, motion, behavior, cues and feedback.

- Presence/Absence

This is a type of manipulation that allows the control of the existence and visibility of a point or virtual object in the augmented reality environment.

It is important to mention that the point is a data structure, which contains its spatial position, visibility parameters and further information. That further information should contain pointers or pointer lists to virtual objects, sounds, behaviors and auxiliary information.

The point, when visible, is represented by a small colored sphere. The point manipulation affects all data structure while the virtual object manipulation only affects its respective contents.

Points or virtual objects can be inserted, copied, exchanged, activated or deactivated, turned visible or invisible, inverted or deleted in the augmented reality environment, as described in the following.

Insert: it is an action that allows the creation of a new point or the association of a virtual object with a previously selected point in the augmented reality environment.

Copy: this is an action that replicates a point with its content or only the content, depending on the specific action, allowing its movement to another place.

Exchange: this is an action on virtual objects that can be executed only on existing points with associated objects. This action promotes an advance on the pointer list, exchanges the current virtual object by the next one from the list and also activates its sound and behavior.

Activate: it is an action that enables a point or one of its associated virtual objects, turning it into one element of the augmented reality environment. All points and virtual objects are active by default during their creation, but they can be deactivated in other moments.

Deactivate: this is an action that disables a point or one of its associated virtual objects, giving to the user the impression that it does not exist.

Visible: it is an action that makes a point or a virtual object capable to be seen by the user.

Invisible: this is an action that disables the visualization of a point or virtual object although it remains in the environment.

Invert: this is an action that shows all invisible points and virtual objects and hides all visible ones. Two inversions return the environment to the former state.

Delete: this is an action that erases a point or associated virtual objects (one or all).

All actions, unless the "invert", require a previous selection of a point or a virtual object, returning some type of feedback to the user.

- Changing Characteristics

Changing characteristics of points and its associated virtual objects involves appearance and geometry parameters, such as color, light, shininess, transparency, shape and size. This action on points and its associated virtual objects changes its parameters and can be executed on one or several elements, depending on the commands. This can be used in complex operations on points and virtual objects, providing conditions to the creation of powerful and attractive applications.

Moreover, this mechanism can be used to implement user feedback and visual cues, since a point or a virtual object representation can have its appearance and/or geometry changed to warn the user.

- Motion

In the augmented reality environment, points are placed (translated) in relation to a base reference, keeping the reference orientation. Virtual objects can be translated and rotated in relation to the respective point reference.

An object movement consists in translating and/or rotating it relative to the point reference. Particularly, the base movement (translation and/or rotation) takes all associated points with it. Therefore, the movement of points and virtual objects allows the reorganization of the augmented reality environment while the movement of the base allows the visualization of the augmented reality environment by another point of view.

Besides the continuum movement, the system also allows discrete movement between two points, using attraction and repulsion. Thus, certain virtual objects can be attracted to a determined point when they are released near it. In this case, the final placement of the virtual object, involving translation and rotation, is adjusted to a previously defined position. Other points can have repulsion to certain virtual objects, so that these objects can not have the final position near the points. Even if the user try to do this action the objects will be repelled.

The attraction/repulsion grade depends on the sphere radius, which can be increased or decreased by a command. This characteristic is important for precise and fast movements, since the final positioning is automatically adjusted by the system. This function is useful in training activities, mainly in initial phases, when the trainee does not have good skills yet.

- Behavior

Points and virtual objects can present behaviors, which allow them to react in a simple or complex way [11], as a result of user interactions with them or with other elements of the augmented reality environment and even as a result of a system state changing.

A simple example of behavior in that environment is shown in an assembly of a virtual helicopter, which is presented next, as an application.

The parts of a virtual helicopter are assembled by the user, following visual and audible cues and receiving feedback. The behavior of all parts (except the body of the helicopter) is changing color when selected. Each part returns to its original color when it reaches its final position. The behavior of the body part is testing the proximity of any part. When one of them reaches the neighborhood of the body part, the respective final position starts blinking and stops when it is filled with the part. Besides, this behavior tests if all parts are placed and exchanges the assembled helicopter by an animated one.

The behavior is a small program or configuration associated with points and virtual objects, which can be activated or deactivated by the user. In the creation phase, the behaviors are deactivated by default. When the behavior is activated, it tests the system parameters and imposes specific behaviors in each case, interacting with the user, other virtual objects and system state.

The behavior capacity of points and virtual objects allows the development of complex and intelligent augmented reality environments.

- Visual and Audible Cues

Graphical, textual and audible cues in augmented reality environment are useful to inform the user what to do, indicating situations or system states, aiming to improve the performance of the interactions.

Graphical cues are blinking spheres or visual paths, indicating an object to be selected or a destination to be reached.

Textual cues are visible information, indicating system parameters, such as resolution of attraction/repulsion of the points, resolution of the selection tool, active commands, etc.

Audible cues are sounds, indicating certain situations, or even voice records describing points to help the user to interact with them. These cues are important for blind-oriented applications.

- Feedback

The feedback from interactions can be visual, audible, or haptic. The actions should be indicated by operations that make static or animated changes on the appearance and/or geometry of the points and virtual objects representations, play sounds, activate text, activate haptic actuators, etc. These functions must have the capability to be enabled or disabled by the user.

4.2.3. Release

The release of a selected point or a virtual object is carried out by a finalization action, meaning the end of the interaction. This action should be executed by gesture, keyboard, voice command or haptic action, in a way similar to the selection.

In the case of points, it would be interesting if the system could implement repulsion to prevent the overlapping of points. In the case of virtual objects, the release can occur close to the attraction area of a point, resulting in the adjustment of the placement to the final position. If the repulsion is activated, the released virtual object will be placed outside the repulsion area around the point, even if the user has tried to place it inside the repulsion area.

4.3. Taxonomy of selection, manipulation and release techniques for augmented reality environments

Based on the subject discussed in the former subsection, we organized the taxonomy of selection, manipulation and release techniques for augmented reality environments (see Figures 6 and 7).

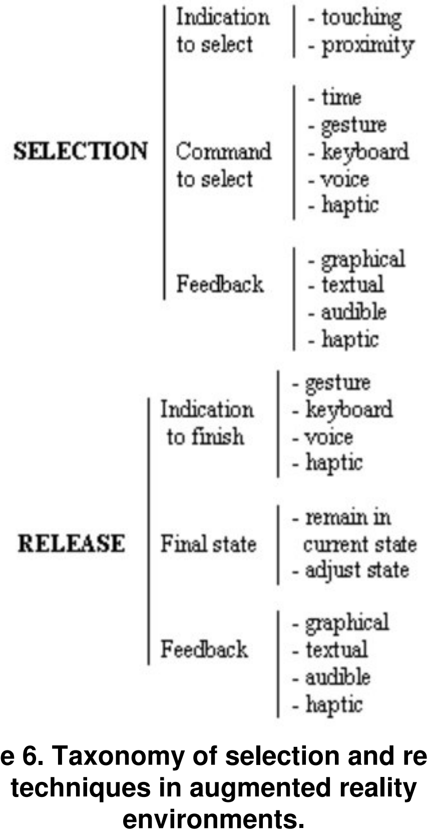

Figure 6. Taxonomy of selection and release techniques in augmented reality environments.

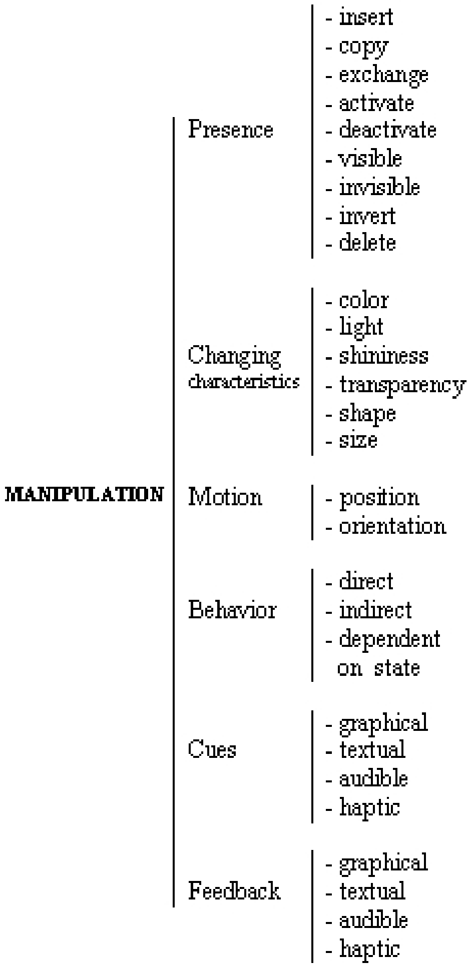

Figure 7. Taxonomy of manipulation techniques in augmented reality environments.

This taxonomy of interaction in augmented reality environments is an adaptation of a similar one prepared for virtual reality environments, presented by Bowman et al. [5] [6].

In this new taxonomy, the selection is simple, because augmented reality environments use direct and tangible actions [9].

The manipulation is more complex, once the augmented reality environment considered here has more related functions working on the combination of virtual objects with the real world.

The release is more powerful, because instead of dealing just with location, the augmented reality environment works with the state of the point or virtual object, which is a more complex data structure.

Besides, the taxonomy diagram can be expanded with one more level, showing implementation details of each action/characteristic mentioned in the third level.

5. Implementation

To validate the ideas and the concepts used in the ARAS-EU, the implementation considered the system and some application cases.

5.1. Implementation of an ARAS-EU version

An augmented reality authoring system for end-users was implemented using a toolkit, although pure programming or a combination of them could be used too.

Following the first approach, the system was implemented with ARToolKit [1] adapted and complemented with programming to present the necessary functionalities.

ARToolkit is an interesting option, because it is based on markers (cards), which indicate positions and contain identifiers. A marker can be used as a selector, point indicator, object indicator, or function indicator and its movement can be interpreted as gestures, such as occlusion, inclinations and so on.

Figure 8 shows a simplified authoring environment with a Reference Marker (REF) associated with two visible points and several virtual objects. It also shows the Function Markers containing each one a collision point to select and execute actions on a specific point or virtual object.

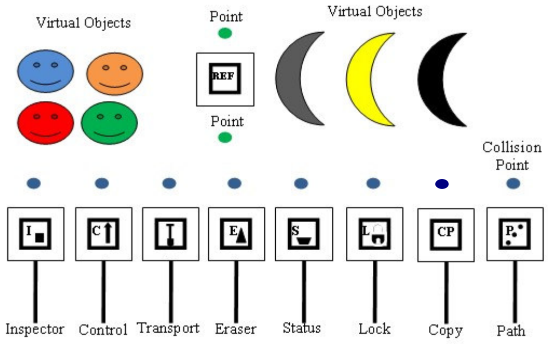

Figure 8. Simplifyed authoring environment with reference and functions markers.

In the implementation of this ARAS-EU version, the marker functions were combined with keyboard functions, giving more flexibility to the system, particularly to main authoring. Voice and haptic commands were not used.

Main authoring uses many functions based on keyboard, because it depends on technical and complex commands. Secondary authoring and the utilization phase are based on a few markers, allowing the exploration of tangible actions by non-expert users.

5.2. Implementation of an application case

To illustrate the utilization of the ARAS-EU, we developed an application of an assembly of a virtual helicopter. Firstly, it was disassembled, but the original positions of its parts were saved.

5.2.1. Authoring

In the main authoring, each part of the helicopter was placed on a virtual base (see Figure 9). Each part received a virtual point, in front of it, aiming to facilitate its selection and manipulation. Each part also received a behavior involving graphical cues and feedback, such as changing color and visible path indicating the destination.

The helicopter body received a different behavior consisting of blinking parts, attractive property and completion test. This behavior must work in the following way. When a selected part enters in the attraction area of the body part, its final position starts blinking. As soon as that part is released, it goes to the final position and the blinking stops. A visual path can also be created to help the end-user to carry on the operation. When all parts are assembled, the helicopter is exchanged by an animated one.

In this phase, the following resources were used: a catalog with parts of the virtual helicopter; a marker; keyboard functions to work in conjunction with the marker to create a point as well as copy, move and release it. Besides, special functions were used individually to insert behavior and to finish the authoring.

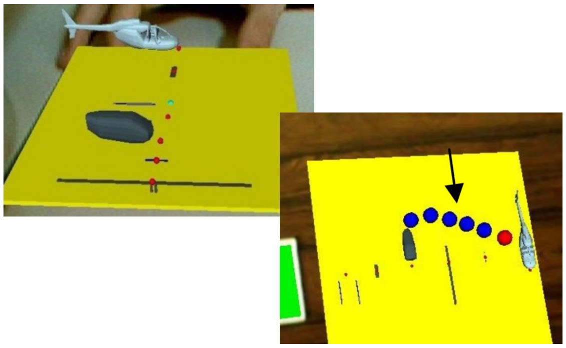

Figure 9. Assembly resources of a helicopter with graphical cues and feedback.

5.2.2. Utilization

In the utilization phase, only one marker was used to interact with the application. That marker has a small colored sphere coupled in front of it, which works as reference to select and manipulate the helicopter parts (see Figure 10).

The selection indication is carried out by the movement of the marker sphere toward the part sphere. When both spheres become close, the corresponding part changes its color (see Figure 11). After keeping both spheres in this situation for a few seconds, the selection is turned on and the part is coupled to the marker. Then, the marker with the selected part can be moved near the final position in the helicopter body (see Figure 12), starting the blinking part in that position. It is also possible to see a graphical cue showing a path from the origin to the destination for each part (see Figure 9).

When the part is released by marker occlusion in its attraction area close to final position, that part is adjusted to the final position and the feedback and cue are turned off (see Figure 13). As soon as the last part is assembled on the helicopter body, the helicopter is exchanged by an animated one.

Figures 10 to 13 show the sequence of the helicopter receiving its cockpit. A video of the helicopter assembly in an augmented reality environment can be seen on the Internet [2].

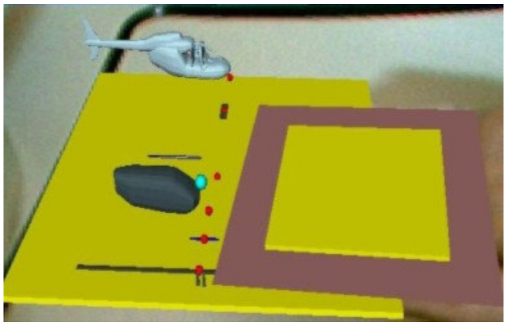

Figure 10. Selection of the cockpit.

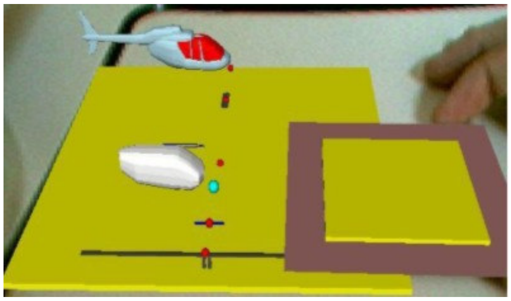

Figure 11. Helicopter showing the place to put the cockpit.

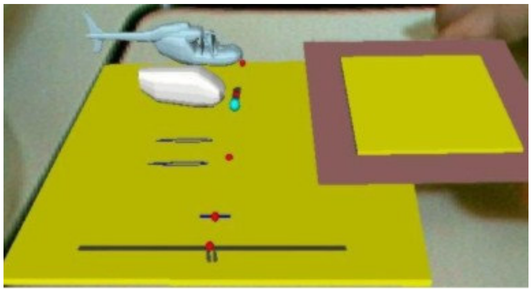

Figure 12. Helicopter receiving its cockpit.

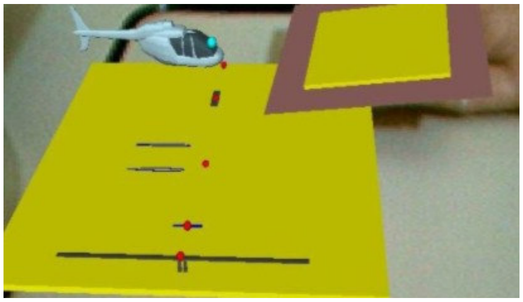

Figure 13. Helicopter with its cockpit.

5.2.3. Finalization

At any time of the authoring or utilization, the application state can be saved to be recovered later. In this case, there is a keyboard function that activates the saving process, asking the user for the filename or using a default filename. If the user is not interested to save the application state, he can discard it and turn the system off.

It is possible to use markers instead of keyboard functions, leaving the user independent of the keyboard.

6. Conclusion

In this paper, we presented the design of an augmented reality authoring system for end-users (ARAS-EU), discussing its structure, implementation and utilization, emphasizing the interaction, collaboration and authoring aspects. This augmented reality system allows the development of ready to use and easy to modify applications, based mainly in parameters configuration through edition or operations issued by keyboard functions and/or tangible actions using markers.

Taxonomy of selection, manipulation and release techniques for augmented reality environments was presented and discussed.

We also presented and discussed an application related to the assembly of a virtual helicopter in the ARAS-EU environment.

The use of points and virtual objects joined with markers and keyboard functions allowed the development of innovative applications by end-users, exploring the reconfiguration during the utilization phase.

Besides, behavior, cues and feedback have allowed the development of intelligent virtual objects and environments, pointing to researches in the hyperreality area.

The complexity of using the system during its phases decreases from the main authoring to the utilization. In the main authoring, we use many keyboard functions and markers while in the utilization we tried to use a few markers without keyboard functions. This strategy is suitable for tangible operations to be executed by end-user.

The project ARAS-EU also works as a collaborative augmented reality system, exploring the concept of local and shared environments. The user can work in private local spaces and in a shared space visible and workable by all or a group of users.

ARAS-EU is being improved with the development of different visual and audible cues to be placed in a library. These cues will be used to indicate actions that will advice the sequence of work making easy the tasks execution during learning phase.

The system can be used to develop different applications, mainly in educational area, using standalone or collaborative approaches. Authoring in two levels can be explored in many ways. One way to use the system is based on the teacher actuating in the first level, preparing the augmented reality environment, and students participating in the second level, modifying the environment or navigating on it. Another way to use the system is exploring the augmented reality environment with a marker interacting with points and virtual objects.

The system is already in test by undergraduate and graduate students, who are developing augmented reality applications for running in standalone and collaborative environments.

References

1. ARToolKit, http://www.hitl.washington.edu/artoolkit
2. AR video, http://www.ckirner.com/filmes/paginas/pag-video-8.htm
3. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent Advances in Augmented Reality. IEEE computer Graphics and Applications, 21(6), 34--47 (2001)
4. Azuma, R.: A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments, 6(4), 355--385 (1997)
5. Bowman, D.A., Kruijff, E., Laviola Jr, J.L., Poupyrev, I.: 3D User Interfaces: Theory and Practice. Adison-Wesley (2005)
6. Bowman, D.A., Johnson, D.B., Hodges, L.F.: Testbed Evaluation of Virtual Environment Interaction Techniques. Presence: Teleoperators and Virtual Environments, 10(1), pp. 75--95 (2001)
7. Broll, W., Lindt, I., Ohlenburg, J., Herbst, I., Wittkamper, M., Novotny, T.: An Infrastructure for Realizing Custom-Tailored Augmented Reality User Interfaces. IEEE Transactions on Visualization and Computer Graphics, 11(6), 722--733 (2005)
8. Grimm, P.; Haller, M.; Paelke, V.; Reinhold, S.; Reimann, C.; Zauner, R.: AMIRE – Authoring Mixed Reality. In: First IEEE International Workshop on Augmented Reality Toolkit, Darmstadt, Germany, pp. 87--88 (2002)
9. Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K., Tashibana, K.: Virtual Object Manipulation on a Table-Top AR Environment. In: International Symposium on Augmented Reality (ISAR 2000), Munich, Germany, pp. 111--110 (2000)
10. Kirner, C., Kirner, T.G.: Virtual Reality and Augmented Reality Applied to Simulation Visualization. In: Sheikh, A.E., Ajeeli, A.A., Abu-Taieh, E.M.O. (eds.) Simulation and Modeling: Current Technologies and Applications, IGI Publishing, pp. 391--419 (2008)
11. Lee, G.A., Nelles, C., Billinghurst, M., Kim, G.J.: Immersive Authoring of Tangible Augmented Reality Applications. In: Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004), Arlington, VA, USA, pp. 172--181 (2004)
16. Looser, J.; Grasset, R.; Seichter, H.; Billinghurst, M.: OSGART - A Pragmatic Approach to MR. In: International Symposium of Mixed and Augmented Reality (ISMAR 2006), Santa Barbara, CA, USA (2006)
12. MacIntyre, B., Gandy, M., Dow, S., Bolter, J. D.: DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences. In: ACM User Interface Software and Technology (UIST04), Santa Fe, NM, USA, pp. 197--206 (2004)
13. Milgram, P., Kishino, F.: A Taxonomy of Mixed Reality Visual Displays. IEICE Transactions on Information Systems, E77-D(12), 1321--1329 (1994)
14. Poupyrev, I.,Tan, D.S., Billinghurst, M., Kato, H., Regenbrecht, H., Tetsutani.N.: Tiles: A Mixed Reality Authoring Interface. In: INTERACT 2001 Conference on Human Computer Interaction, Tokyo, Japan, pp. 334--341 (2001)
15. Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavari, Z., Encarnacao, L. M., Gervautz, M., Purgathofer, W.: The Studierstube Augmented Reality Project. Presence - Teleoperators and Virtual Environments, 11(1), 33--54 (2002)