Social Shaping of Digital Publishing: Exploring the Interplay Between Culture and Technology
A.A. Baptista et al. (Eds.)
IOS Press, 2012

doi:10.3233/978-1-61499-065-9-104

"Tangible Culture" - Designing Virtual Exhibitions on Multi-Touch Devices

M. Hecher [a] R. Möstl [b] E. Eggeling [a] C. Derler [b] D. W. Fellner [a]
[a] *Fraunhofer Austria Research GmbH, Austria*
[b] *JOANNEUM RESEARCH GmbH, Austria*

Abstract.
Cultural heritage institutions such as galleries, museums and libraries increasingly use digital media to present artifacts to their audience and enable them to immerse themselves in a cultural virtual world. With the application eXhibition:editor3D museum curators and editors have a software tool at hand to interactively plan and visualize exhibitions. In this paper we present an extension to the application that enhances the workflow when designing exhibitions. By introducing multi-touch technology to the graphical user interfaces the designing phase of an exhibition is efficiently simplified, especially for non-technical users. Furthermore, multi-touch technology offers a novel way of collaborative work to be integrated into a decision making process. A flexible export system allows to store created exhibitions in various formats to display them on websites, mobile devices or custom viewers. E.g. the widespread 3D scene standard Extensible 3D (X3D) is one of the export formats and we use it to directly incorporate a realtime preview of the exhibition in the authoring process. The combination of the tangible user interfaces with the realtime preview gives curators and exhibition planers a capable tool for efficiently presenting cultural heritage in electronic media.

Keywords. multi touch, virtual exhibitions, interactive design, digital preservation, X3D

1. Introduction

In cultural heritage institutions digital content has gained increasing importance over recent years. Digitalized media like photographs, videos, audio and three-dimensional representations of exhibition objects are available to museums and recent projects like 3D-COFORM [1] gather to enhance and simplify the process of acquiring digital representations of cultural heritage assets, as well as to document and persistently store the accumulated information. Therefore, in the future even more assets will be available in digitalized and annotated form.

The availability of scanned 3D objects, photographs and videos of historical artifacts offers many advantages. Exhibits, which would be secured in an archive, can now be presented to a wide audience. Furthermore, virtual artifacts cannot only be used to replace existing objects, but also to complete fragments or to show them in a state never seen before. In digital form the content of an exhibition can be visualized with highlights and enriched with information. Combining those enriched digital artifacts with current digital media technologies and/or modern web browsers capable of displaying 3D con-

tent the cultural content reaches an audience all over the world without replacing the real museum.

One challenging problem is to represent the digitalized cultural heritage assets and bring them in a form applicable to be shown to a "virtual visitor". An interesting project in this area is the "Google Art Project" [7]. It digitalizes museums in taking panorama pictures of the exhibition areas and the exhibits. Interested users can then navigate freely within the museum and explore the exhibits in using an ordinary web browser as a viewer. The application eXhibition:editor3D we present in this paper also concentrates on the goal to allow the distribution of cultural heritage in a digital form. However, its focus is not only the presentation of an exhibition but it provides an authoring tool capable of assembling digital data as part of a virtual exhibition. eXhibition:editor3D is an existing application for authoring 3D virtual exhibitions and exhibition tours. We will not present a detailed description of the tool itself, but we describe the aspect of extending it with a gesture-based touch interface that allows a curator to utilize the advantages of multi-touch technologies to simplify the authoring process of digital exhibitions. The multi-touch paradigm also enriches the application with a new form of collaborative work when designing exhibitions, which is described later on.

The next section gives an overview of other available tools for designing digital exhibitions and shortly describes their concepts. Section 3.1 then introduces the eXhibition:editor3D application. Section 3.2 describes the extension of the tool to support gesture based user interfaces on multi-touch devices. Afterwards - in section 3.3 - the visualization module supporting the authoring process is described. Section 4 concludes the outcome of the project and gives an outlook of future work.

2. Related Work

The need for authoring tools in museums to manage the increasing amount of digitalized data gained and to be gained by projects like 3D-COFORM [1] raises the research effort in the field of computer aided planning of virtual exhibitions. Creating this kind of content is a creative process and as such there are different approaches to get the task done, depending on the specific need. Tools currently available typically provide a workflow suite for curators and editors that supports them in their daily work of managing digital exhibits, annotate the data with additional (multi-media) information and organize it in virtual exhibitions to make the cultural heritage information available for a wide audience.

The different tools have the common goal to (a) provide a workflow that allows the management and visualization of cultural heritage and (b) be usable by non-technical users which do not necessarily have experiences in the field of 3D graphics or computer programming. The scale of the applications differs regarding their feature level and coverage of the workflow process, as well as their user interface concept.

In [2] a system is presented to design virtual exhibitions from pre-existing 3D models. The tool provides a 3D interface in first-person view to navigate through the virtual space and arrange exhibits. An artificial intelligence algorithm supports the curator in defining interesting viewpoints inside the exhibition area. Finally the viewpoints are connected to form a virtual tour. For inexperienced users of 3D applications navigating through the virtual space may be complicated, especially when they are not used to this

kind of navigation. The tool encounters that issue in providing the possibility to overlay a 2D map of the virtual space to support navigation. Assistance for the placement procedure of exhibits within the scene is also provided.

Another popular approach is to utilize game-engine technologies and adapt existing frameworks to the field of cultural heritage presentation. In [11] a game engine framework forms the basis of the authoring tool and the visualization module. According to the paper the game engine basis has user and system side advantages, like the good overall performance and the ease of use for curators. From the view of a programmer the framework already provides basic functionality for the shading and the rendering of models, as well as functionality to navigate through the virtual world. In using a game-engine framework such and similar basic necessities do not have to be reimplemented and thus allow for a shorter development cycle of the product.

Besides applications with 3D user interfaces another type exists that - instead of arranging the exhibition in 3D - utilizes a 2D interface system to perform the authoring work. For instance, [15] presents a tool based on floor plans where a curator places exhibits on a flat canvas in a top view user interface. The 2D data generated by the authoring module is then converted in a 3D format and presented by an extensible visualization module.

As stated before it is not trivial for non-technical users to navigate within a three-dimensional world with traditional input devices like a mouse pointer or keyboard on a two-dimensional monitor. Similar to [15] our approach uses a 2D user interface for the authoring tool to provide a simple possibility for navigating inside the exhibition area and for arranging exhibits and virtual tours. However - in contrary to the related work presented in this section - the traditional mouse pointer input method is replaced by touch gesture input interfaces to allow for an authoring process that maps more naturally to our haptic experience of organizing and arranging objects in everyday life situations.

3. The eXhibition:editor3D Application

3.1. The Authoring Tool

The eXhibition:editor3D application is developed by the Austrian Institute of Information and Communication Technologies of JOANNEUM RESEARCH [9]. Its purpose is to create virtual 3D exhibitions from existing digitalized exhibition content, like 3D assets, photographs, videos and audio clips. The application's tool chain allows curators and editors to

- design and preview upcoming exhibitions,
- archive temporary exhibitions
- create interactive 3D presentations for multi-media terminals
- create catalogs or websites
- do advertising for their exhibitions.

Museum curators were deeply involved in shaping the look-and-feel as well as the functionality of the tool. As this cooperation has been established right from the beginning of the software development process, the application is specifically tailored to fit their needs.

To manage a virtual exhibition the editor organizes its content in virtual *rooms*, where multiple rooms can be combined together within a project to present different exhibition floors or different variants of a single exhibition. In a first step the user equips a room with the desired exhibits. The authoring of exhibits is managed by two-dimensional graphical user interfaces and concepts based on floor plans. With a room represented as a 2D floor plan the author places exhibition objects virtually on a flat canvas. The media content (e.g. images) of the exhibits is managed by so called *content repositories*. Media assets are either located in the filesystem or on a web server. Exhibits are easily created by just dragging and dropping a media asset onto the 2D canvas. The different objects are represented as icons that determine the type of exhibit, e.g. 3D object, image, video or audio clip are denoted by their own icon. The user interface previews the content represented by the icon if it is selected. This way a user can simply distinguish between the different exhibits. Moreover, a *property view* is provided that summarizes all properties of an exhibit and allows e.g. to define a name for the object or to enter a precise location or rotation value. eXhibition:editor3D uses existing digital content, the user is free to choose the tools for digitalizing the desired cultural heritage assets and to post-process them. In a second step - after the exhibit arrangement - a virtual tour through the room is planned. For this reason so called *viewpoints* are specified by the curator defining the location and view directions of a visitor to direct her or his attention to a certain aspect of the exhibition. The viewpoints are then connected to form a virtual tour. Finally the resulting exhibition and the virtual tour can be exported as a X3D or Adobe Flash SWF file for visualization. Figure 1 shows an overview of the application.

eXhibition:editor3D is based on the open-source and platform independent *Eclipse* environment. More specifically, it uses the *Eclipse Rich Client Platform* (RCP) [4]. RCP allows application developers to reuse components of the Eclipse platform for developing custom applications. Basic window management, different views on content or functionality like an undo/redo system do not have to be re-implemented, they are provided as part of the framework. Another advantage is the presence of a powerful plug-in system for extending the application. So called *extensions* can be attached to integrate additional functionality. Two custom extensions form the foundation of the modules described in the following sections. The first extends the application with gesture recognition on multi-touch devices and integrates gestures into the graphical user interfaces, the second implements a convenient realtime preview for the authoring process.

An in-depth explanation of the eXhibition:editor3D application is given in [12] and [13].

3.2. Gesture Interface

3.2.1. Introduction

In this project JOANNEUM RESEARCH cooperated with the Fraunhofer Research GmbH, Austria [6] to extend the eXhibition:editor3D application with the capability to handle touch gesture input. Fraunhofer contributed its knowledge in the field of multi-touch technologies. The traditional way of interacting with the authoring application uses the mouse pointer as the input device. This user interaction paradigm has changed with the upcoming of more touch-enabled devices in recent years. Multi-touch technology is the interactive and intuitive means of providing a new user experience and enhances the usability of a range of applications. The success of gesture based applications is im-

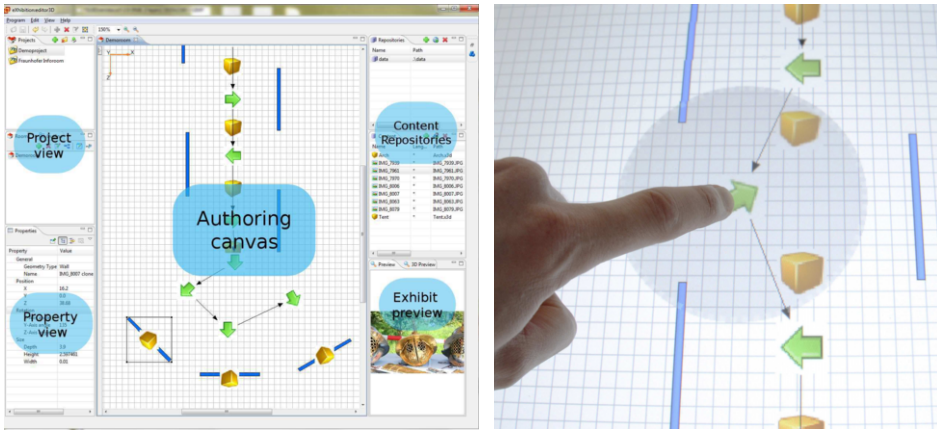

Figure 1. (left) The GUI interface of eXhibition:editor3D showing the authoring canvas, the content repositories, the exhibit preview windows and the property and project view. (right) Touching an exhibit gives visual feedback in marking the area considered for additional multi-touch input with a gray area. Nearby exhibits within this area are not selected when touching them with another finger. Instead, the touch contributes to the gesture on the currently selected exhibit.

pressively demonstrated by mobile devices, where 2D user interfaces are operated on in solely using multi-touch gesture input without mouse or keyboard input devices.

3.2.2. Supported Gestures

The multi-touch input paradigm was applied to the authoring interface of eXhibition:editor3D. Two areas are managed by gestures: The first area is controlling the transformation of an exhibition object, the second area concerns the navigation inside a virtual room. The typical workflow for selecting an item from the content repository, dragging it onto the virtual room canvas and adjusting its position and rotation with the help of mouse and keyboard is replaced by typical multi-touch gestures, allowing for an intuitive exhibition arrangement that maps naturally to the human haptic experience of positioning objects on a canvas. Gestures are provided to translate, rotate and scale an object. Translation is accomplished in selecting an object and moving it around with a single fingertip. For rotation two fingers are placed on or in the near vicinity of an object (e.g. if it is to small to place 2 fingers directly on the icon). When rotating the fingers the rotation angle is mapped on the exhibit. Scaling an object is achieved in first selecting it with a two finger tap and then dragging a third finger forth or back in the vicinity to apply a scale factor. To navigate through the virtual room two distinct gestures are implemented. The first one is a zoom gesture allowing to shrink or enlarge a section of the room. To activate the gesture two fingers are put on the room canvas without hitting an exhibit. When bringing the two fingers nearer together the view is zoomed in, dragging the fingers apart zooms out. Especially for zoomed-in views it is necessary to pan the view to explore other areas. That is accomplished with a single touch on the room canvas and dragging the view until the desired section is reached.

3.2.3. Precision Considerations

Touch input brings up a problem that is commonly named as the "fat finger" problem (a more formal explanation of the issue can be found at [5], where touch vs. mouse

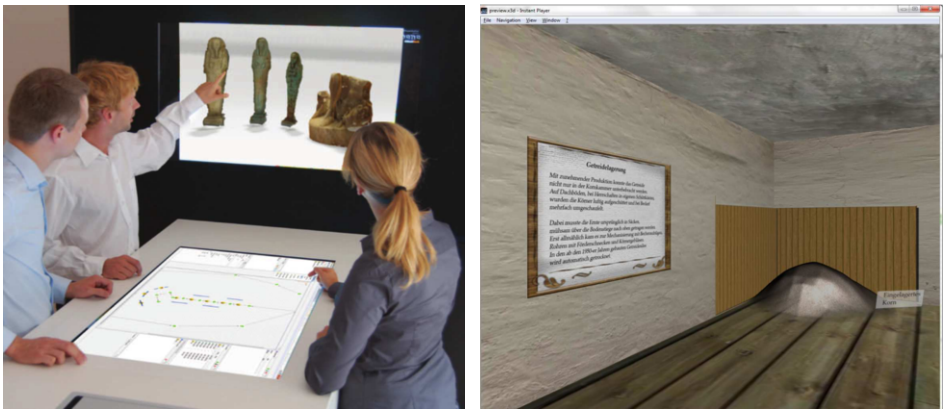

Figure 2. (left) Example setup of a monitor touch device with a separate monitor for the visualizer module with multiple persons collaboratively working on an exhibition. (right) Visualizer module previewing an exhibition using 3D elements and an image with descriptive information of the scene.

input is compared). Selecting an object that is surrounded by others can be tedious as the natural size of a finger does not provide the selection accuracy of a pixel-accurate mouse pointer. The problem gets even more complicated when a multi-touch gesture is to be applied on a relatively small object. In this case it is possible that the user wants to e.g. scale an object in putting two fingers on it and then use the third finger for the actual scaling. If the third finger is placed on a nearby object this object will be moved instead of scaling the previously selected one. As a second example, if two users are manipulating the transformation of nearby objects it is possible that one user selects an object with two fingers to apply a rotation. When the second user selects an object in vicinity with another finger to apply a translation it is possible that the third finger is interpreted as scale factor for the first object. To overcome these selection issues we introduce visual feedback for the selected object. When selecting an item the influence area accepting touches considered as part of a gesture on the object is visually highlighted as shown in Figure 1. When a second or a third finger is placed inside this area of influence the touch is always applied to the selected object. Nearby items located within the area are not considered for touch input. The visual indication of the currently selected object and its area of influence solves confusing behavior when selecting objects in a simple but effective way.

3.2.4. Collaborative Working

The multi-touch paradigm also enriches the workflow of designing exhibitions with the possibility to collaborate on a single exhibition. Around a multi-touch table setup as shown in Figure 2, curators can jointly discuss ideas respectively position and adjust exhibition artifacts in a comfortable way. The setup shows one vertical touch-monitor device integrated in a table furniture. Within the table's sideboard a second monitor is integrated. The second monitor is used to display a 3D visualization of the exhibition in real-time, a feature described in Section 2. The setup allows multiple persons to stand around the touch-device, using the eXhibition:editor3D collaboratively to directly support the decision making process. It becomes easy to try out different exhibition arrangements as

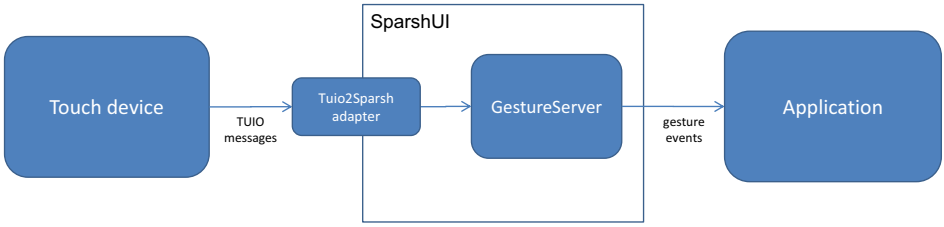

Figure 3. The TUIO input messages from the multi-touch device are converted by the SparshUI framework into gesture events and are then delivered to the touch interface of the application.

each member of the decision making group has the possibility to quickly show ideas in simply arranging the exhibits with a fingertip.

It is possible that multiple users simultaneously add, remove or transform exhibits in the virtual room. For serious exhibition planning this feature may not be very helpful. However, when using the application for educational purposes the feature can be used to provide e.g. school classes with an interesting possibility to create exhibitions and to waken the interest of pupils in packaging historical events into a playful like experience when arranging exhibits with multi-touch gestures.

3.2.5. Technical Implementation

The technical implementation of the gesture input is based on the *SparshUI* [14] library. It provides a gesture recognition framework that relieves the developer from manually transforming a sequence of unconnected low-level touch input events into corresponding gesture events. A *gesture server* calculates gestures out of a stream of touch inputs and provides them in form of events to the application. The library supports default built-in gestures like drag, rotate, scale, etc. but can be extended with user defined gestures as well. The low-level touch input is read by an adapter component that converts touch input events from different sources to the internal format of the gesture server. In this project an adapter connects the *tangible user interface objects* (TUIO) protocol [10] used by many touch devices to translate the input into the gesture server's recognized format. Adapter e.g. for the native Windows 7 touch input events are also available. The high-level gesture events are then mapped to transformation values of the exhibit objects. The objects are represented with the *Eclipse Graphical Editing Framework* (GEF) [3], a 2D based framework for creating graphical interfaces. Extending the existing eXhibition:editor3D application is achieved in utilizing *extensions*, a plugin mechanism provided by the Eclipse framework. Figure 3 shows the connection between the gesture recognition and the authoring tool.

3.3. Visualization Module

3.3.1. Realtime Preview

Two-dimensional user interfaces have the advantage to be easily handled by non-technical users, however, their disadvantage in this field of application is the lack of visual feedback on how the end result looks like. eXhibition:editor3D thus provides a 3D preview that changes and updates the 3D representation of the scene immediately. The application provides an extensible plugin system to support different output formats.

Currently the exhibition can be exported to the X3D format, as well as to Adobe Flash SWF. However, concerning user-friendliness this approach has the flaw that changes in the 2D view provide no instant visual feedback for the curator. To see the changes made on an exhibition the scene first has to be exported to the 3D format for visualization This conversion step slows down the authoring process considerably, especially when fine tuning arrangement details.

To make the workflow of (a) adjusting an exhibits transformation, (b) export the scene, (c) visualize and judge the result to then (d) iterate the process until a satisfying result is achieved the existing plugin system for visualization modules was used to extend the editor with a new module. This module is based on the X3D format and uses its dynamic capabilities to enhance the user friendliness of the authoring process in instantly previewing the current exhibition in its 3D representation. Changes are reflected in realtime by the visualizer. This instant visual feedback enhances the efficiency of the workflow and simplifies the fine tuning task considerably. From a curator's perspective the authoring tool and the visualizer component are no longer separated entities of the workflow, but form an integrated part.

The visualization module directly supports the definition of viewpoints and their connection to a virtual tour. Per drag'n'drop the curator positions viewpoints within the exhibition room. In activating a viewpoint with a double tap on its icon the visualizer changes its camera to the selected viewpoint. With translation and rotation gestures the location and viewing direction can be adjusted, the result is directly shown in the visualizer. That allows a very fast way to define interesting points of view of an exhibition and to connect these points to form a virtual tour. Besides the possibility to activate single viewpoints the application provides touchable buttons that allow to skip through the whole tour in the visualizer to give an impression of the result and to evaluate the immersion of the story telling process.

3.3.2. Technical Implementation

The visualization module is implemented on top of the *InstantReality* suite developed by Fraunhofer IGD [8]. The suite provides a X3D viewer - the *InstantPlayer* - that visualizes the 3D representation of the exhibition. The eXhibition:editor3D and the InstantPlayer are connected via the X3D scene format. When a new authoring session is started eXhibition:editor3D internally generates a X3D file representing the exhibition scene. This file is loaded by the InstantPlayer and shows the current state of the scene. The InstantPlayer supports the *External Authoring Interface* (EAI), a protocol to dynamically change the loaded 3D scene. This feature is used to instantly map changes applied in the authoring tool to the visualization. The EAI is also utilized to switch the camera in the InstantPlayer when a viewpoint is selected by the curator. The result is a preview of the scene that updates in realtime to give the curator instant visual feedback of the changes and also allows to quickly adjust viewpoint locations and viewing directions.

InstantReality is a powerful framework to visualize the exhibition scene in different environments. E.g. it is possible to directly render the result as stereoscopic image on a projection wall or in a CAVE (Cave Automatic Virtual Environment). The EAI implementation of the suite supports remote invocation over the network. Therefore the authoring tool and the visualizer do not necessarily have to run on the same host. This loose coupling allows a flexible combination of authoring and visualization devices, which can be used e.g. for interactive presentations of exhibitions where the authoring tool is

running on a multi-touch monitor device and the visualization output is presented on a stereoscopic projection wall.

4. Conclusion and Future Work

Changing the input paradigm from a mouse pointer and keyboard oriented interface to a gesture oriented touch interface extends the eXhibition:editor3D with a modern possibility for an effective planning of virtual exhibitions and museum tours. The touch interface maps the natural haptic experience of users and therefore allows an effective workflow to create exhibitions. In combination with the visualization module that is no longer a separated entity, but an integral component during the authoring process, curators can efficiently arrange and position exhibition tours. The independence of external input devices like mouse or keyboard eases collaborative work and helps to accelerate the decision making process.

An additional advantage of the multi-touch technology is its "show" factor. This factor makes the application attractive e.g. for educational purposes to awaken the interest of pupils on the topic of cultural heritage. With the innovative touch interface the young generation can be attracted to delve into the world of cultural heritage in experimenting with the presentation of exhibits, arrange their own tours and learn about the historical background during the process.

The InstantPlayer as viewer component provides a flexible tool to visualize the resulting exhibition scene on ordinary devices, as well as in complex environments like stereoscopic or CAVE setups. This flexibility allows museums to appoint the eXhibition:editor3D not only for the actual planning process of exhibitions, but also for showcases and multi-media presentations of their cultural heritage assets. With its extensible export system the application can also easily store exhibitions in formats which are suitable for different types of electronic media, e.g. to make past exhibitions accessible via a browseable internet catalogue or on mobile devices. The built-in support for the flexible and accepted exchange format X3D ensures not only interchangeability with other systems, but is also a future-proof and open format for long-term archiving of digital exhibits.

However, in some parts it becomes obvious that the application was not designed to exclusively use multi-touch input without a mouse and a keyboard device. E.g. the properties view demands some sort of keyboard input to change the different properties in entering values. Another issue is encountered in the field of collaborative work when more curators are standing around the multi-touch device to discuss the exhibition. The authoring canvas itself is independent from where it is looked at, as it uses icons and no fonts to visualize exhibits. However, for tasks like saving or loading a project, managing the content repositories, etc. it is necessary to stand on a certain side of the multi touch table to get the correct view.

Concerning future work two areas can be improved. Currently the content repositories providing the exhibits either read supported media formats directly from the file system or retrieve files from network locations. It would be beneficial to be able to access the content repository of the 3D-COFORM project. The project is currently in progress, however, supporting the upcoming infrastructure will improve the interconnectivity between cultural heritage institutions.

The second area of improvement is the authoring interface. Currently it provides an orthogonal top-view of the exhibition room. That only allows to directly translate or rotate an exhibit in one coordinate plane when using only gestures and no keyboard device. In introducing additional views all degrees of freedom would be accessible with touch gestures and the keyboard necessity would be obsolete for changing transformation values.

5. Acknowledgments

The furniture integrating the multi-touch monitor shown in Figure 2 is provided by Bene AG, Austria. The usage of the 3D content in the same Figure is granted by the Universal Museum Joanneum, Schloss Eggenberg, Graz, Austria and JOANNEUM RESEARCH.

References

[1] D. Arnold. Tools and Expertise for 3D Collection Formation. In *Proceedings of EVA 2009 Florence*, pages 94–99, 2009.

[2] Luca Chittaro, Lucio Ieronutti, Roberto Ranon, Eliana Siotto, and Domenico Visintini. A High-Level Tool for Curators of 3D Virtual Visits and its Application to a Virtual Exhibition of Renaissance Frescoes. In *VAST10: The 11th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, volume Full Papers, pages 147–154, Paris, France, 09/2010 2010. Eurographics Association, Eurographics Association.

[3] Eclipse Graphical Editing Framework (GEF). http://www.eclipse.org/gef.

[4] Eclipse Rich Client Platform (RCP). http://wiki.eclipse.org/index.php/Rich_Client_Platform.

[5] Clifton Forlines, Daniel Wigdor, Chia Shen, and Ravin Balakrishnan. Direct-Touch vs. Mouse Input for Tabletop Displays. In *PROCEEDINGS OF CHI 2007*, pages 647–656. ACM, 2007.

[6] Fraunhofer Austria Research GmbH. Austria, www.fraunhofer.at/vc.

[7] Google Art Project. http://www.googleartproject.com/.

[8] InstantReality. http://www.instantreality.org.

[9] JOANNEUM RESEARCH GmbH. Austria, www.joanneum.at.

[10] Martin Kaltenbrunner, Till Bovermann, Ross Bencina, and Enrico Costanza. TUIO: A Protocol for Table-Top Tangible User Interfaces. In *Proceedings of the 2 nd Interactive Sonification Workshop*, 2005.

[11] Victor Mateevitsi, Michael Sfakianos, George Lepouras, and Costas Vassilakis. A Game-Engine based Virtual Museum Authoring and Presentation System. In *International Conference on Digital Interactive Media in Entertainment and Arts*, pages 451–457, 2008.

[12] Moritsch, O., et al.,. Creating 3D Virtual Exhibitions. In J. Trant and D. Bearman, editors, *International Cultural Heritage Informatics Meeting (ICHIM07): Proceedings*, 2007. Published October 24, 2007 at http://www.archimuse.com/ichim07/papers/moritsch/moritsch.html.

[13] Murg S., et. al.,. Using Standardized Methods To Present Three-Dimensional Content On The Web In The Context Of Cultural Heritage. In J. Trant and D. Bearman, editors, *Museums and the Web 2008: Proceedings*, 2008. Published March 31, 2008. Consulted June 24, 2011. http://www.archimuse.com/mw2008/papers/murg/murg.html.

[14] Prasad Ramanahally, Stephen Gilbert, Thomas Niedzielski, Desirée Velázquez, and Cole Anagnost. Sparsh UI: A Multi-Touch Framework for Collaboration and Modular Gesture Recognition. *ASME Conference Proceedings*, 2009(43376):137–142, 2009.

[15] S. Vassallo, E.; Spina. VirtualXibIT: Framework for the Venue Authoring, Population and Visualisation of 3D Virtual Exhibitions. In *VAST2009: VAST-STAR, Short and Project Proceedings*, 2009.