Virtual Art Gallery

Stanislav Hrk
hrk@decef.elf.stuba.sk

Department of Computer Science and Engineering
Faculty of Electrical Engineering and Information Technology
Bratislava / Slovak Republic

Abstract

Introduction of the third dimension into the distributed environment of the Internet has opened numerous new applications in various fields of human interest. The particularly interesting application is in the field of fine arts, where it provides exciting new possibilities to present the artwork to broad audience.

This paper provides an overview of techniques and one approach to development of interactive 3D galleries accessible from the web. It includes presentation of an application for designing virtual galleries and their presentation on the WWW based on the VRML (virtual reality modeling language).

Keywords: virtual reality, virtual gallery, VRML, WWW, geometry-based rendering, image-based rendering, picture panoramas.

1. Introduction

The problem of designing virtual galleries is relatively new area in broad scope of software development. It has become topical problem in relation to dynamical development of computer multimedia capabilities and mainly with expansion of the Internet.

The Internet as a global communication medium offers unique opportunity for addressing considerably broader audience than classical galleries, which are limited not only by capacity, but also by temporal and spatial constraints of real world. Furthermore, Internet presentations can take various forms, shaped by author's inspiration during the creation process and specific means provided by technologies used.

In general, there are a few mayor approaches to creation of virtual galleries. Chronologically the first and still the most widespread are two-dimensional galleries. This kind of galleries are usually authored in HTML, with optional functionality enhancements like database connectivity, search capabilities etc provided by some scripting language or Java. The main problem of these galleries lies in their two-dimensional nature. It is the fact, that subjective feeling that they evoke in a spectator is very often closer to reviewing a catalogue of given artwork, than visiting a real gallery.

A step forward to developing more realistic virtual galleries was introduction of the 3D web contents, which can be used to immerse the spectator into three-dimensional ambient of the gallery and significantly enrich overall experience of visiting such gallery. Modelling an interesting 3D scene is, however, far more complex task than creating a simple 2D presentation, which is perhaps the main reason why 3D web contents still is not as popular as 2D.

Traditional approach to three-dimensional computer graphics is based on synthesizing images from geometric models. Several 3D modelling languages were developed for use over the Internet. Perhaps the best known is VRML, a language for modelling 3D interactive worlds and objects on the WWW. Alternatives to VRML are languages developed by Flatland (3DML) [9] or Superscape (SVR) [10]. Superscape has recently come up with a new technology for 3D modelling for the WWW based on a subdivision of surfaces, which produces high quality models that are substantially smaller in size than models generated by other technologies. Specific information is not available as it is kept as a company secret.

Lately, an alternative modeling paradigm was introduced, known as Image-based rendering. Image-based rendering (IBR) describes a set of techniques that allow three-dimensional graphical interaction with objects and scenes whose original specification began as images or photographs. In an IBR pipeline, processing is applied to a set of input photographs creating an intermediate data structure. Later, this data structure is used to create new images of the scene or object [6].

One of image-base techniques quite popular nowadays uses single panoramic image for scene description. Panoramic image is mapped onto the inside of some geometric primitive, like a cube, cylinder or a sphere. The viewer is placed in a centre of chosen shape, and is allowed to modify the field of view and rotate around that single point. Main disadvantage of this technique is that the viewer is fixed to a single point. Moving to another point of view requires a new panoramic picture taken at desired position. At the present, there are commercial systems available that use this technique, like Apple's QuickTime VR [11], IPIX [12] or iMove [13], and also some free systems like PTViewer [14]. These systems enhance panoramic worlds with hotspots that provide links to other files or panoramas, embedded images, movie textures or object movies. Some galleries using picture panoramas already exist, for example at the official website of the Louvre Museum [15].

Another one of image-based rendering methods is morphing. Image morphing is used to provide smooth transition between two images. An image-based technique called plenoptic modelling is described in [5]. This technique uses both panoramic images and morphing to render a view of a scene from arbitrary point. The problem with this system is that in order to synthesise a usable interpolation between two panoramic images they must be relatively close. To enable free movement in a scene, a large number of panoramic images need to be taken, resulting in huge memory requirements. This makes this approach hardly usable on the WWW at this time, but in future some system like this might be deployed for the web.

The techniques mentioned above, both geometry-based and image-based, can be used as a base approach for creation of virtual galleries. In further text we explore one approach using VRML as the language for describing the virtual gallery and objects within. VRML was chosen because it provides flexible interface, it is well documented and commonly known.

2. Acquiring the artwork

One of interesting problems is acquiring some sort of digital representation of the artwork that is to be exhibited. Image-based representation has its advantages here over geometry-based, because recovering the models is reduced to taking 2D images, even for three-dimensional objects represented by object movies.

Capturing the geometrical representation of a 3D object is far more complex. One way is to model them by hand, but that is very time-consuming process and can hardly reach the complexity of a real object. Other possibility is to use one of object reconstruction techniques developed in the field of computer vision. A system for reconstruction of 3D object from pictures taken from multiple views developed at our faculty is described in [8]. This system provides output in VRML format. Quality of acquired objects is acceptable for use on the WWW. Some of models generated by this system were used in sample galleries (see figure 6).

3. Virtual Gallery Editor

Virtual Gallery Editor is a tool for development of three-dimensional virtual galleries. It was designed to simplify, organize and speed up the process of creating virtual galleries, setting up exhibitions, and their presentation in a form publishable on the World Wide Web. Other possible use is to build a prototype of a real exhibition before it is deployed. This tool brings new quality: distant, interactive planning of the exhibition layout.

3.1 Motivation

One could argue that it is possible to use existing powerful generic 3D modelling software to develop more realistic galleries than with a tool like VGE. On the other hand, generic software is built for use in broad range of applications, which brings substantial functionality overhead when focusing on a single application, like virtual galleries. This reflects on high price of generic software, a lot of time and effort required for learning to use it to produce quality output. This can prove to be very discouraging for most users.

The goal of this project is to provide lightweight area-specific tool that offers features that generic software does not focus to. The main advantages of VGE over generic modelling software products are:

- Fast and intuitive design of the gallery layout
- Structuring of presented artwork and related information into logical entities – exhibitions
- Reuse of galleries for multiple exhibitions. Exhibited artwork can be changed with a simple push of a button
- Automated generation of HTML pages about particular exhibited artwork (painting) from information provided in the exhibition
- Exporting the gallery data in an internet-ready form

3.2 Decomposing the gallery

In order to start building a gallery, a set basic objects that are going to be used must be identified, together with relationships among them. This was done by process of abstraction from objects in real galleries, with considerations to specific requirements put before virtual galleries and possibility of modeling that objects using resources provided by VRML. At the end, the structure of data model shown on figure 1 was adopted.

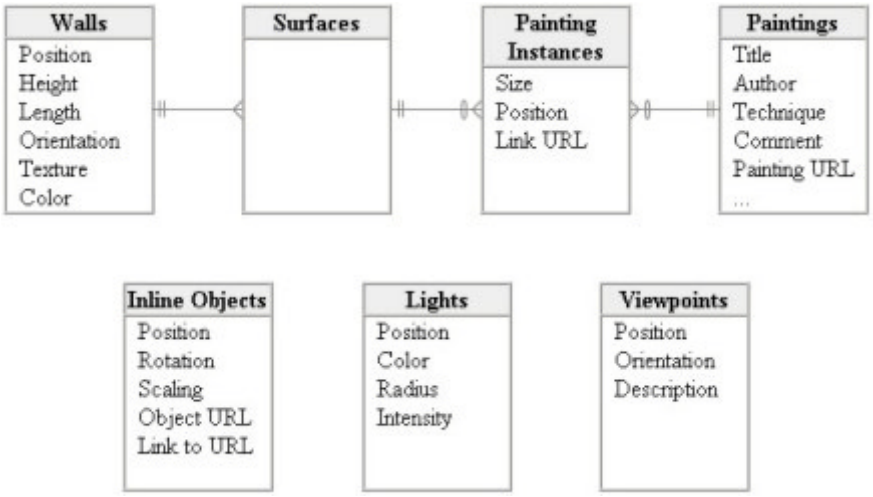

Figure 1: Data model

Key objects in the gallery are, naturally, the works of art. At the time, only exhibitions of 2D pieces of art like paintings, prints and photographs are fully supported, but there are means for presenting other types of artwork, as for instance sculptures, audio or video records, that will be discussed later. To simplify the data model, all 2D pieces of art in VGE are referred as paintings. In case of paintings, there are two data entities related to them: paintings and painting instances. The paintings entity contains various information about paintings that are part of current exhibition. What information is stored can be seen from the data model. The painting instances entity stores data about specific instance of that painting in the gallery. It is possible that a single painting occurs more than once in the gallery. The painting instances entity is linked to surfaces entity, which links a painting instance to the surface of the wall where it is hanged. Linking painting instance to a wall surface rather than specifying absolute co-ordinates of the painting was necessary to ensure that when moving a wall, painting instances attached to it would move correspondingly.

Other key objects are the walls, which shape the gallery and provide surfaces for hanging pictures. Walls are represented by entity of the same name. As seen from the data model, walls can be textured or have specified colour. The wall provides two surfaces for hanging paintings, which are represented by the surfaces entity. Surfaces is an intermediate entity between painting instances and the walls entity, and have no own attributes.

Viewpoints are used set a pre-defined view of the gallery. A set of viewpoints can be used to create a tour of the gallery. Currently, only one static tour can be implemented this way, but it is planned to implement a mechanism to allow a visitor to interactively create own gallery excursions by selecting from predefined viewpoints. This can be done before entering gallery, or dynamically during the tour using VRML EAI [3]. Viewpoints are represented by the viewpoints entity, which has attributes like position, orientation and description.

Placing lights in a gallery is more or less self-explanatory. Choosing the right illumination can greatly affect the overall appearance of a gallery.

As previously mentioned, it is also possible to exhibit 3D artefacts using VGE. This is using objects represented by data entity called inline objects. These objects provide an interface for inserting arbitrary VRML contents into the gallery. This way we can insert various static or kinetic sculptures, or even complete installations (complex scenes with 2D and 3D artefacts that cover entire exhibition hall).

3.3 Setting up an exhibition

When setting up an exhibition, the author chooses the artwork that is to be presented and enlists it into exhibition. Currently, the only type of artwork supported by exhibitions in VGE is 2D artwork.

There are two ways of setting up an exhibition in VGE. The one way to do it is to manually input data into application. This is done via interface called exhibition editor implemented as a dialog box. It allows the author to provide additional information about the piece of art presented (refer to the paintings entity in figure 1). The only required information is URL for the 2D image of the piece of art, but it is recommended to fill all fields as information provided here can be used to automatically generate HTML page for that piece of art.

The other way is to load text file containing description of exhibited artwork in predefined format. It can be written manually, or exported from some gallery created in VGE. This allows data entered for one exhibition to be reused in another. Other scenario of using this feature is in case, when authors send their work, for instance by e-mail, to curator of the gallery. Along with image files, a description file can be included to help including paintings into the gallery.

3.4 Editing the floor plan

Basic view of the gallery is through its floor plan. VGE provides a 2D interface to edit the floor plan, as can be seen from figure 2. This approach was chosen because it provides clear overview and allows precise positioning of objects in the gallery. This view implements placing, deleting, moving, resizing and setting properties specific to each type of objects. Setting properties is implemented through object-specific dialog boxes (Wall Properties box in figure 2) invoked by clicking the right button on selected object.

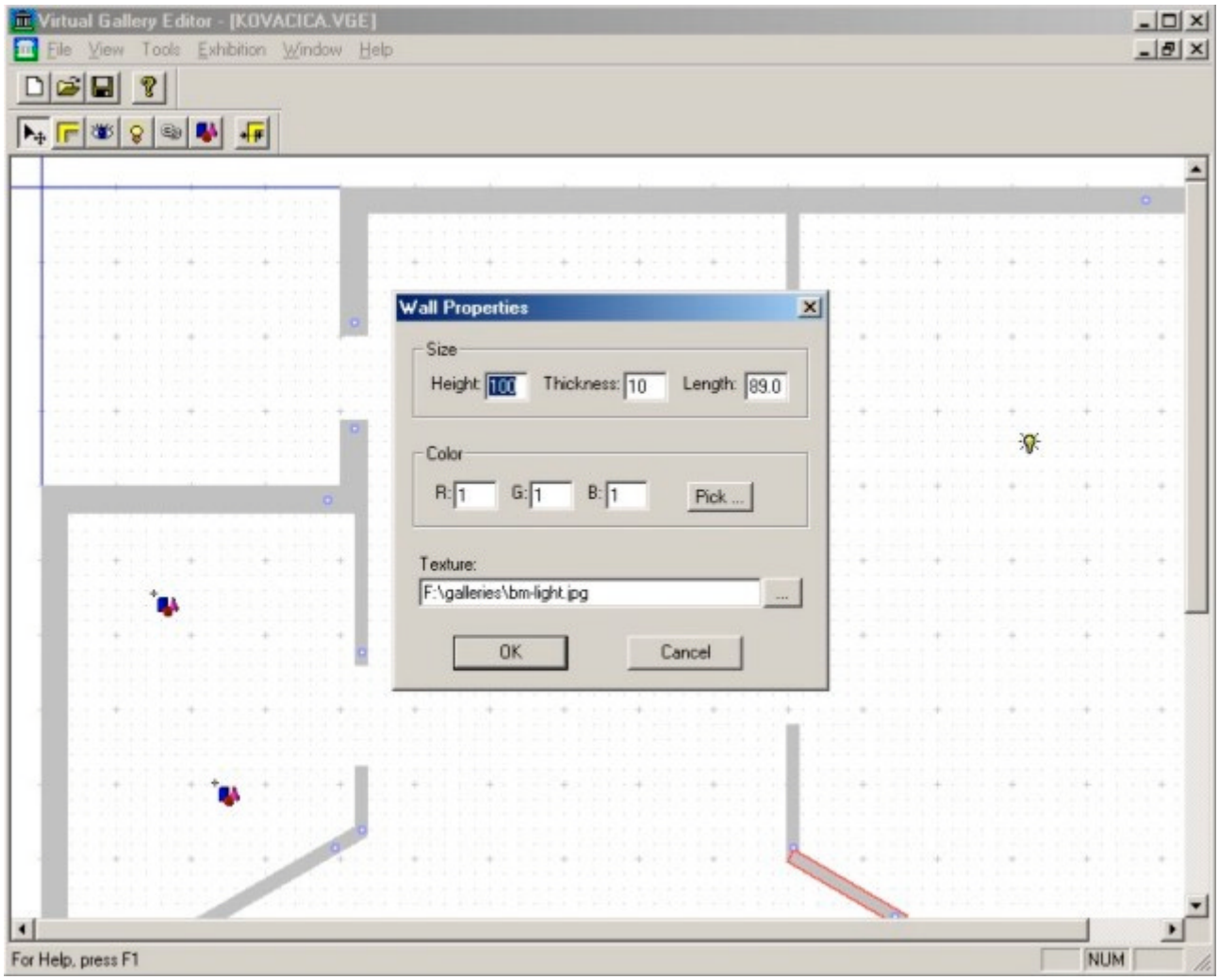

Figure 2: The floor plan

3.5 Placing the exhibited artwork

Placing paintings (in general, 2D pieces of art) in VGE is somewhat different than placing three-dimensional artwork. Since paintings are bound to walls, in order to place a painting there must be at least one wall present in the gallery. To modify the layout of selected wall, it is required to double-click on the wall, which invokes two new windows. The first represents the layout of that wall's surfaces, along with paintings placed on selected surface. The second window contains the list of paintings placed on selected surface and some additional information and options related to

selected painting instance. This window also enables to switch between surfaces of the wall by clicking on the tabs at the top of the window. Figure 3 illustrates these windows.

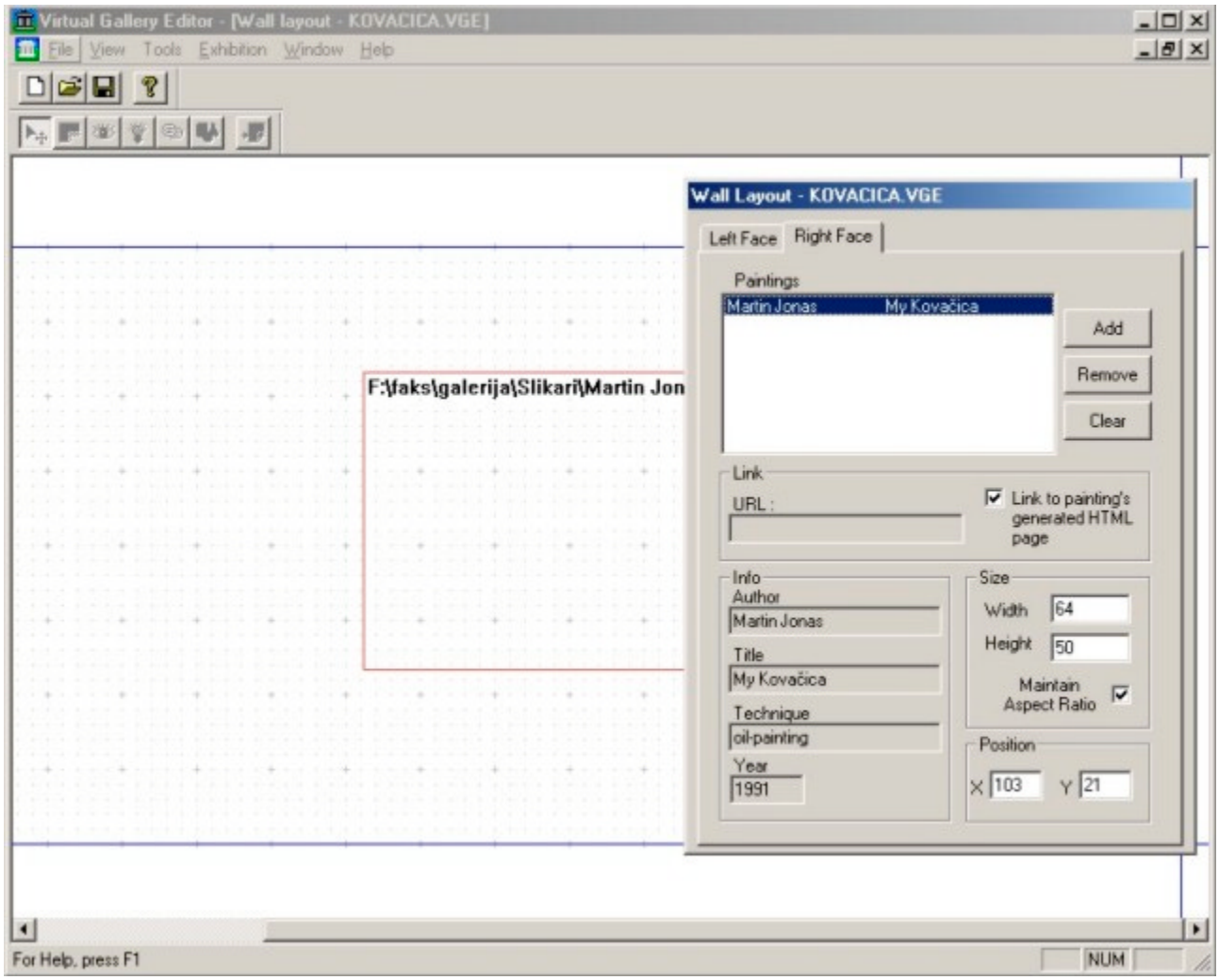

Figure 3: Wall layout view

At the wall layout window, user can add, delete, resize or move painting instances. When adding a painting instance, a dialog appears which allows choosing from paintings entered into exhibition, and the manner in which it is to be placed on the wall.

Placing sculptures is similar to placing any other inline object into gallery. Note that because of different orientation and size of VRML objects, it is sometimes necessary to rotate and scale the object to fit the gallery, which can be done at inline object's properties dialog.

3.6 The output

When finished designing the gallery, its data can be saved along with the exhibition data in an internal format, or it can be exported in VRML. When exporting to VRML, a directory structure is created containing files related to the gallery. Figure 4 illustrates the directory structure.

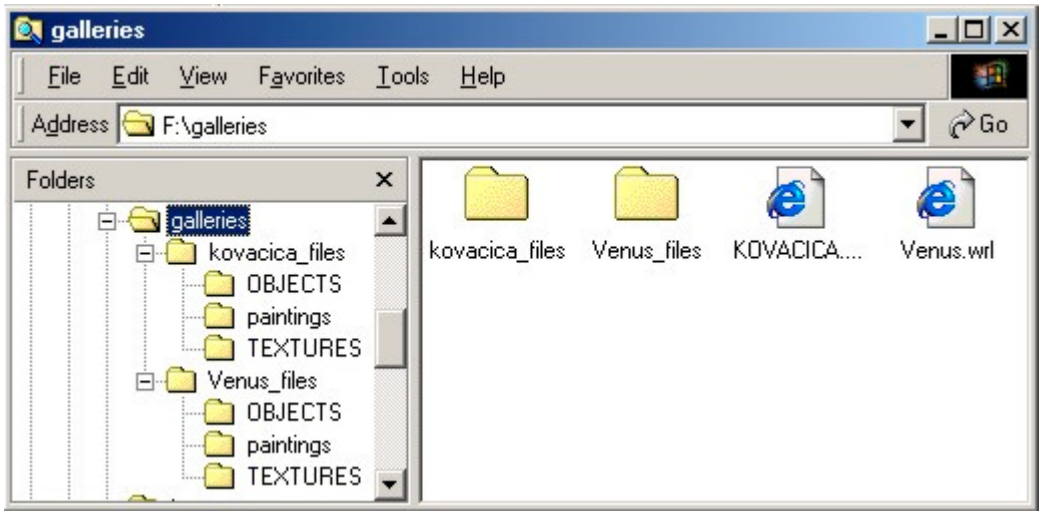

Figure 4: Directory structure of exported gallery

At the top, there is a single uncompressed VRML file entitled gallery_name.wrl file (where gallery_name is the name under which is the gallery exported) that describes the gallery itself, and a directory gallery_name_files that contains files related to the gallery.

The textures subdirectory contains textures used for walls. The paintings subdirectory contains images of paintings and possibly generated HTML pages with information about paintings for those paintings, for which it was selected. Not all paintings in the exhibition are copied into this subdirectory, but only those used (exhibited) in the gallery. The objects subdirectory contains VRML files of inline objects used in the gallery. VRML files linked to the gallery using inline objects are not parsed for links to other resources. This means that other related files, as for instance textures, sounds, video clips or other objects will not be copied and have to be copied to this subdirectory manually.

Only the files whose URL points to local disks are copied and their URL are modified to correspond to directory structure of the gallery. URL of files that cannot be found on a local disk are left intact (because it is presumed that it points somewhere on the WWW).

3.7 Implementation and testing

The Virtual Gallery Editor was implemented in C++ using MS Visual C++ 6.0 and MFC library on MS Windows operating system. This product was developed using object-oriented principles. A separate class that encapsulates the data and provides external interface for manipulation with object was created for each type of objects in the gallery.

Gallery data exported from VGE is in standard VRML 2.0 utf8 format. It was tested on the MS Windows platform, using combinations of Internet Explorer 5.5 and Netscape Navigator 4.71 web browsers with Cosmo Player 2.1.1, Cortona VRML Client 2.0 and Blaxxun Contact 4.4 VRML browsers. The results were varying from one combination to another, mainly because of differences in implementations of VRML browsers. Testing was performed on a Pentium II 350 MHz machine equipped with 3D Labs Permedia 2 graphic card with 8 MB of RAM, with a combination of Internet Explorer and Blaxxun Contact. Sample gallery data included approximately 1600 polygons and 25 textured boxes. Data size was 630 KB, from which textures occupied about 450 KB. Using Direct3D driver interactive framerates were achieved without difficulty on all resolutions up to 1280x1024 / 16 bpp. Higher resolutions were not available with only 8MB of graphic RAM.

3.8 Examples

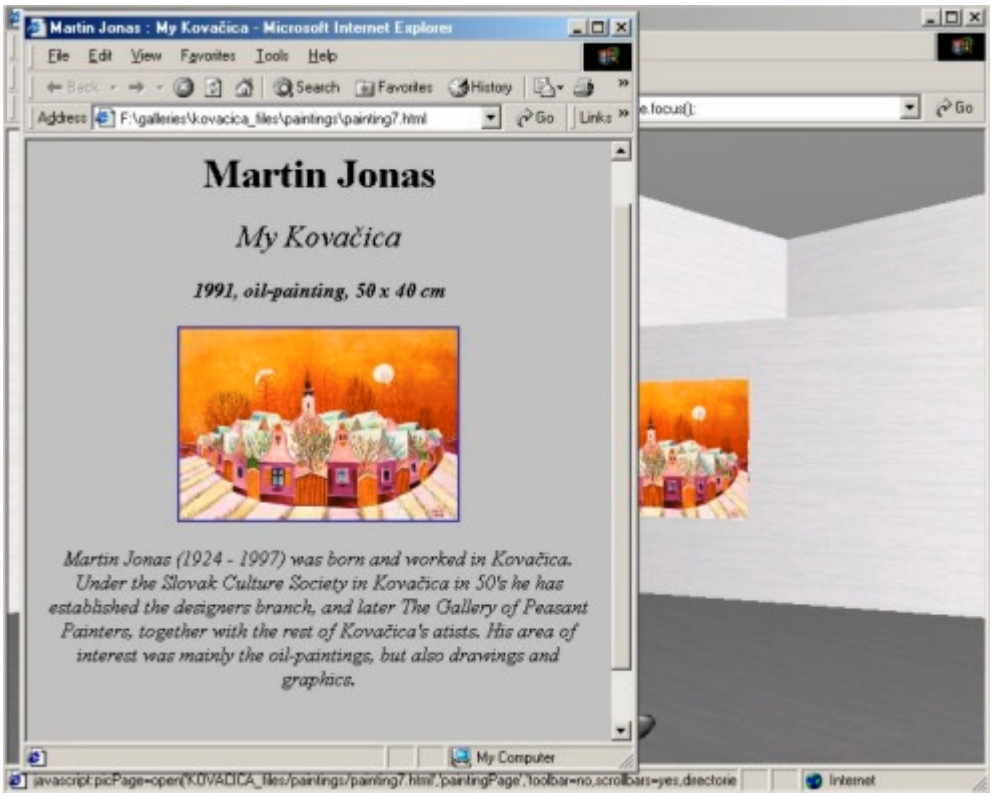

Figure 5: A shot from exhibition of Kovacica Naive Art

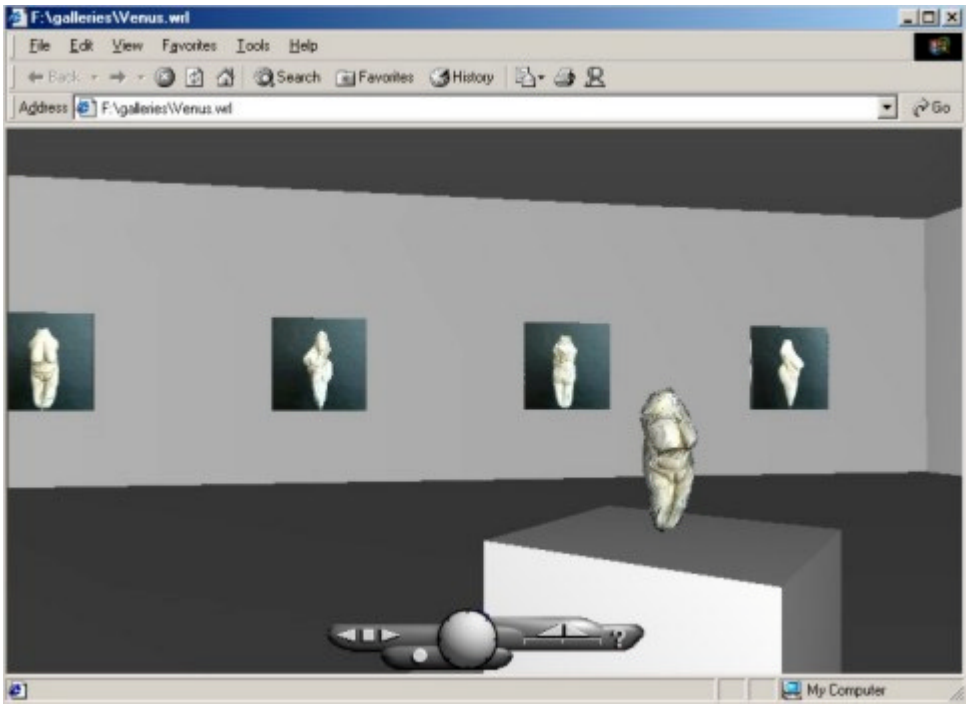

Figure 6: A shot of the statuette of Venus displayed on a stand in a virtual gallery

4. Conclusions and future work

Three-dimensional virtual galleries are interesting new application of virtual reality on the Internet. Some of these galleries can be found on the WWW, but mostly this potential is left unused. One reason for this can be the risk exhibitor undertakes that comes from diversity of tools and modeling languages for 3D contents on the WWW. Many of these languages are developed by companies in ambition to force it as a standard. This did not happen yet and the situation with 3D contents on the web still remains uncertain, as new technologies and languages are developed constantly.

This paper described one approach for construction and presentation of virtual galleries using VRML. The Virtual Gallery Editor tool is functional and can be used to produce interesting galleries deployable on the WWW. One possible drawback can be the size of complex galleries, which arises from the need of storing large amount of data presented at the same time. Reducing the quality of presented data, as for instance the resolution of images or video, in cases where the highest quality is not required can eliminate this drawback. In time, with progress in technology this problem will be gradually eliminated.

Future evolution of this project might go in two directions. The first is to enhance the capabilities of existing Virtual Gallery Editor. The list of possible improvements includes implementation of database connectivity, improving the user interface, adding exhibition support for other types of 2D and 3D artwork like sculptures, video and audio records, implementation of more basic object types directly in the gallery editor, opening the galleries for multi-user presence, ...

Other possible direction in which this project might evolve is implementation of a tool for authoring virtual galleries and exhibitions using picture panoramas, and a viewer for this kind of galleries. This solution has advantages over 3D because it provides very realistic views of the gallery interior captured from the real world, with option of adding additional contents to the scene. The gallery can include paintings, video, sounds, or sculptures using object movies, or even a virtual humanoid guide through the gallery that responds to user actions.

Possibilities remain open.

5. Acknowledgements

At this point I would like to express my gratitude to associate professor Martin Sperka for his guidance through this project. I would also like to thank Tomas Gerhat for his initial work on virtual galleries, and Dioniz Vadkerty for VRML reconstruction of the statuette of Venus used in one of the examples.

6. References

[1] The Virtual Reality Modeling Language: International Standard ISO/IEC 14772-1:1997, ISO, 14.10.1999.

http://www.vrml.org/Specifications/VRML97/index.html

[2] The Virtual Reality Modeling Language: International Standard ISO/IEC 14772:200x, ISO, 4.4.2000.

http://www.web3D.org/TaskGroups/x3d/specification

[3] VRML External Authoring Interface Specification, 15.3.2001.

http://www.vrml.org/WorkingGroups/vrml-eai/Specification/

[4] Lipkin, D.: VRML Informative Annex: Recommended Practices for SQL Database Access,

Oracle Corporation, 18.12.1998.

http://www.web3d.org/Recommended/vrml-sql/

[5] McMillan, L.: An Image-Based Approach to Three-Dimensional Computer Graphics, Chapel Hill, 1997.

[6] McMillan, L., Gortler, S.: Image-Based Rendering: A New Interface Between Computer Vision and Computer Graphics, *SIGGRAPH Computer Graphics Newsletter,* Vol.33 No.4, November 1999

http://www.siggraph.org/publications/newsletter//v33n4/contributions/mcmillan.html

[7] Gerhat, T.: Model of a virtual gallery in 3D, Final Project, Faculty of Electrical Engineering and Information Technology, Bratislava, 1999.

[8] Vadkerty, D.: Reconstruction of 3D Objects From Multiple Views, Master Thesis, Faculty of Electrical Engineering and Information Technology, Bratislava, 2000.

[9] Flatland Online, Inc., 10.3.2001.

http://www.flatland.com/index.html

[10] Superscape, 10.3.2001.

http://www.superscape.com

[11] Apple QuickTime, 15.3.2001.

http://www.apple.com/quicktime

[12] IPIX, 15.3.2001.

http://www.ipix.com

[13] iMove, 15.3.2001.

http://www.smoothmove.com

[14] PTViewer, 15.3.2001.

http://www.fh-furtwangen.de/~dersch

[15] The Official Website of the Louvre Museum, 15.3.2001.

http://www.louvre.fr