

STEREOSCOPIC VISUALIZATION AND DATABASE INFORMATION RETRIEVAL

L. Sechidis, S. Sylaïou, P. Patias

The Aristotle University of Thessaloniki, Department of Cadastre Photogrammetry and Cartography Univ. Box 473, GR-54124, Thessaloniki, Greece lazikas@photo.topo.auth.gr, sylaïou@photo.topo.auth.gr, patias@auth.gr

KEYWORDS: 3D representation; visualization; stereoscopic vision; virtual reality; programming; databases

ABSTRACT

Augmented reality is one of the new applications to archaeology that gives to user the sense of “being there” and allows to observe virtually reconstructed archaeological landscapes with historical buildings. This paper presents a system that aims to representation of a virtual environment in stereo, using 3 virtual cameras. It allows user to put in scene one or more models that are ready to use and to navigate in them. In addition to this, it gives the opportunity to interact with the virtual environment in real time, to rotate the objects presented in scene and observe their detail. It also establishes links between the objects of the scene and any database and allows user to obtain additional information about the places or objects. The system can be used by researchers as a tool that will help them in developing and testing new techniques on 3D representation (e.g. multiple LOD meshes, etc) of any 3D data.

1. INTRODUCTION

Virtual Reality (VR) is not a new concept. Although VR hybrid systems existed as back as from early 90’s, unfortunately, they had low capabilities and very high cost, due to specialized hardware they needed. At that time, several companies involved to the new VR software industry, with most of their software considered VR world developing tools, while cost varied from some hundreds dollars to as much as lots of thousands. Additionally, the majority of this software composed of libraries of specific programming routines that were combined to form the substance and dynamic content of the virtual world. Nowadays, there is a great interest in VR systems; and the area of archaeology/museology is one of the rather intensive applications areas (eg. El-Hakim et.al 2003, Ogleby 2001a, 2001b). Only in the European Union many research projects in the field of cultural heritage and archaeology have been funded, as for example the SHAPE project (Hall et.al. 2001), the 3DMURALE project (Cosmas et.al. 2001), the Ename 974 project (Pletinckx et.al. 2000), the ARCHEOGUIDE project (Gleue et. al. 2001), the LIFEPLUS project (LIFEPLUS 2005), the ARCO project (ARCO 2005, Liarokapis et. al. 2004), just to name some.

In many instances there is a misconception about what a VR system really is. To make it simple, the main difference between a VR software and a more traditional 3D modelling/animation or multimedia program is that the former incorporates two key characteristics: immersion and interactivity. As Brill (Brill 1993) some 15 years ago stated “While a computer animation usually provides 3D content, it offers little or no control over this content. In contrast, a Multimedia program offers users interactive control over the program, but that control is more of a root-and-branch, menu-selection variety. VR software, on the other hand, allows users to create compelling 3D environments that also provide a degree of interactivity that goes far beyond that found in traditional multimedia programs. In VR the user “enters” the program and change segments of it at will and in real time. VR users can open and close doors, wander through preconceived landscapes, move into and out of rooms, touch various buttons and switches and generally change, rearrange and interact with their environment”.

As Brooks put it (Brooks 1999) “Four technologies are crucial for VR:

- the visual (and aural and haptic) displays that immerse the user in the virtual world and that blocks out contradictory sensory impressions from the real world;
- the graphics rendering system that generates at 20-30

fps the ever changing images;

- the tracking system that continually reports the user’s position and orientation
- the database construction and maintenance system for building and maintaining detailed and realistic models of the virtual world.”

The required hardware for a VR system, due to rapid technology progress and to dramatic fall of prices, is much more accessible now than it used to be in the past. On the contrary, on the software side the picture is different. The cost of the software is still the same, if not more expensive, since its capabilities have also increased. For example, the cost a game engine can vary from some thousands dollars to hundreds of thousands of dollars depending on the capabilities and the purpose of its usage.

Current systems to support VR software construction are of two categories: Toolkits and Authoring systems. Toolkits are programming libraries that provide a set of functions for supporting the creation of a VR application. Authoring systems are complete programs with graphical interfaces for creating worlds without requiring detailed programming. These usually include some sort of scripting language, and while simpler to use, current authoring systems do not offer all the functionalities of toolkits.

Among these categories we here present the “OpenView®” system (Sechidis et. al. 2004). “OpenView®” is not a VR Toolkit, an Authoring system nor a game engine, It is a VR presentation tool and we offer it to the scientific society as an open-source system and free-of-charge.

“OpenView®”’s purpose is to present any kind of 3D data, from simple points and lines to huge VR scenes with thousands of triangles and texture. Additionally, “OpenView®” can be used by researchers as tool that will help in developing and testing new techniques on 3D representation (e.g. multiple LOD meshes, etc) of any 3D data, since its source code is available by the authors, upon request (patias@auth.gr).

Two characteristics of “OpenView®” that will be stressed here is its stereoscopic ability as well its ability to easily connect to any type of databases.

2. VR, STEREOCOPY AND DATABASE CONNECTION

2.1 VR and Stereoscopic Vision

The most successful human-computer interface paradigm so far has been the Xerox Parc desktop metaphor, which simplified human-machine interaction by creating a palpable, concrete

illusion for users to manipulate real, physical objects positioned on a desktop. "While the desktop metaphor is interacting well with 2D worlds, it shows major drawbacks in dealing with 3D worlds, since in limits the correlation between manipulation and effect and it shows high degree of cognitive separation between users and the models they are editing" (Gobbetti et al. 1998).

The goal of the VR is to put the user in the loop of a real-time simulation, immersed in a world that can be both autonomous and responsive to its actions. The input channels of a VR environment are those with which humans emit information and interact with their environment. From all these channels, Vision is generally considered the most dominant sense, and there is evidence that human cognition is oriented around vision. Also, humans give precedence to the visual system if there are conflicting inputs from different sensory modalities. High quality visual presentation is thus critical for virtual environments. From the aspects of visual sense the one that has the most impact is the depth of perception (Gobbetti et al. 1998).

The primary human visual mechanism for perceiving depth is stereoscopic vision. Many studies have shown that there is a significant difference between monoscopic versus stereoscopic displays in terms of the subjective report of presence in the virtual world (eg. see Hendrix 1994). The subjects of the above survey have been asked, among others, the following questions, in order to check the importance of stereovision:

- "How strong was your sense of presence, "being there", in
- the virtual environment?"
- "How realistic did the virtual world appear to you?"
- "To what degree did the room and the objects in the room
- appear to have realistic depth/volume?"

The statistical results indicated that the addition of binocular disparity did significantly increase one's sense of presence in the VR environment, enhanced the sense of realism, increased the realism of the virtual world's apparent volume, and as a result, it significantly increased one's feeling of being able to

reach into the virtual world and grasp an object.

"OpenView®" supports stereoscopic vision either through polarized screens or through anaglyph viewing. VR system generally uses various formats of displays: Head-mounted displays (HMDs), CAVE-like surround projectors, panoramic projectors, workbench projectors, and desktop displays. All of them have their own merits and disadvantages in specific applications, yet none dominates. "OpenView®" can support either display system.

2.2 VR and DataBase connection

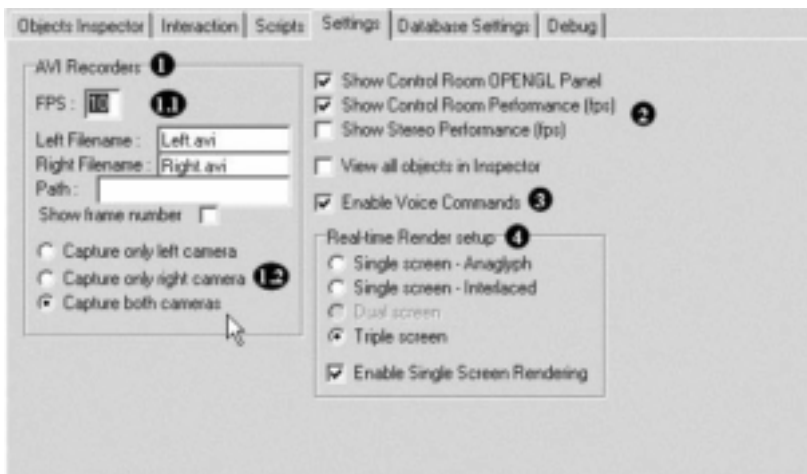
VR systems generally manage huge amounts of data. These include both vector data and raster images. In addition, it is desirable that the user gets access to non-graphic information. Although, managing the former is a big issue for VR system, and is mainly depended on the hardware technology as well as software complexity, most systems of today succeed more or less, the only issue remaining is in the use of Internet.

Managing non-graphic data is another issue, and rarely is been tackled by VR systems. The strategic decision we took during the development of "OpenView®" is to put also emphasis in this issue. The challenge was to also create a 3D presentation tools with extensive GIS capabilities. This led inevitably to a number of technical problems, compatibilities and presentation options, for a seamless and real-time connection between different databases and 3D graphic visualization tools, which are explained in detailed in the following sections.

2. "OPENVIEW®" TECHNOLOGICAL SETUP

"OpenView®" can import and handle a virtual world or any kind of 3D data, interact with viewer, render stereo-pairs and produce images for both left and right eyes in order to provide stereoscopic vision. Also, it can display information or metadata about the objects that participate in the world using any database that is published in ODBC.

Figure 1. Main Settings: Set's up 'global settings' for presentation. Usually these settings are done once.



1. Settings for the avi recorder. Avi recorder can record the scene as it is displayed in real-time into avi files.

1.1 Set's up speed (FPS), names and the path of the recorded avis.

1.2 Defines if recorder will capture left, right or both cameras.

2. Defines if information about render speed will be displayed.

3. Enable voice commands (user can 'speak' and OpenView will 'execute' the desired command).

4. Defines where and how the scene will be render. Rendering in Anaglyph or Interlaced mode is done using software render in a single screen. Rendering in Triple Screen mode is done using hardware render and is the suggested mode for systems with 2 or 3 display monitors.

"OpenView®" consists of three major windows (Fig. 1). Two of them called "left display" and "right display", are OPENGL windows and are projected to viewer. The third one called the "Control Room" is the heart of the system. Although "OpenView®" can even work on a dual-display system, it is recommended to be used with a tri-display capabilities system for better performance and interactivity.

"OpenView®" uses a typical stereoscopic presentation setup:

two polarized projectors that will project the stereo-pair on a silver-dyed screen. Viewers must wear polarized eye-glasses in order to view the projected stereo-pair. For the time being it does not support single-display setups. Additionally, one or more graphics cards with OPENGL support is needed in order to render the stereo pairs and to output the images to the projectors.

Currently, two Compaq projectors having polarized filters and

two systems with different CPU, graphics cards and RAM (both with cost less than 1500 €) are used.

4. EXPLORING OPENVIEW

4.1 Importing models and scenes – Supported formats

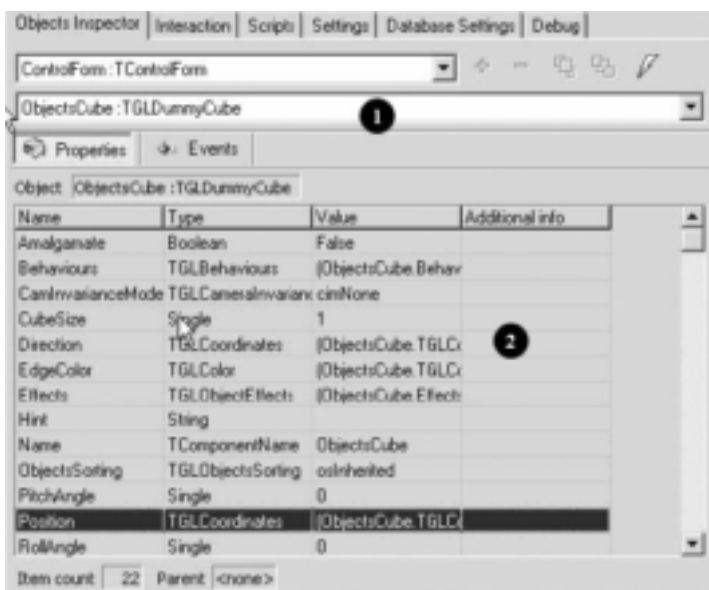
“OpenView®” is not a graphics editor, nor a VR developing tool. This means that, even if it is possible to use “OpenView®” in order to create simple objects and worlds from scratch (using scripts), it is suggested to use external, specialized tools to do this task; then all it is needed is just to import them into “OpenView®”.

Since “OpenView®” uses the GLScene Library (Lischke et. al.

2005), it supports all the formats that GLScene does. Some of them are then 3D Studio (3Ds), TIN, STL, MD2 (Quake2, animated), OBJ (WaveFront, and many others), SMD (Half-Life, skeletal animation, obtained from a decompiled MDL, f.i. with MilkShape) (Fig. 2).

Additionally, an extra implementation has been done in order to import grid data from simple ASCII files or Surfer’s DSAA format, both combined with orthoimages.

When a new object is inserted into the scene, it gets a unique name. This name is very important, since it is used by the internal scripter to identify the object. It is the parameter that is used to query the database for additional information about the object and it is displayed in objects properties Inspector.



1. Selects any object that participates in scene
2. The properties of the selected object can be viewed and changed. Changes are effected immediately on scene

Figure 2. Objects Inspector Tab: Gives access to all properties of every object that participates in scene. Additionally, can be used to set up cameras, rendering machine, sound objects etc.

4.2 Viewing and changing objects properties -Moving and rotating

Interactive applications have to model user interaction with a dynamically changing world. In order for this to be possible, it is necessary for applications to handle within a short time real-world events that are generated in an order that is not known before the simulation is run. Thus user interface software is inherently parallel, and all problems in parallel programming have to be solved (eg. synchronization, maintenance of consistency, protection of shared data) and most importantly in considerably high rates (eg. better than 20Hz), since VR applications have very stringent performance requirements in visual feedback and latency.

In “OpenView®” it is possible, in real time, to view and change almost all the properties of an object (or model). This is done using the Objects Inspector panel in the Control Room, where all objects and their properties are displayed. For example, the position, rotation vectors, scale, color or visibility of any object can be changed (Fig. 3). The number of the properties depends of the object’s type. The only thing that cannot be changed interactively is the geometry of the object, although this can also be done using scripts.

To control an interactive story or installation, interaction tools have to be available to the visitor of the virtual world. As this is open to the public and often are visited by children, young

adults and school groups, displays and their interaction tools have to be easy to understand and robust enough to survive rough treatment. This means that the interaction with the presentation or story has to be either touch less or is done with simple, robust tools. Touch-less interaction is for example possible by using cameras and hand gesture recognition. For the latter, Arcade game controllers offer a wide range of interaction tools like trackballs, button, joysticks and such. Currently “OpenView®” supports game-type controllers, whereas a touch-less, voice recognition controller is under development. Moving and rotating inside scene is achieved using the PC’s keyboard and mouse or joystick. Moving forward and backward, turning and strafing left and right, going up and down (using mouse wheel) has already implemented, while different motion behaviors (e.g. head up/down) are under construction. Motion and rotation speeds are not fixed and can be changed in real-time. Also, motion can be done using two different methods, walk or flight.

The whole scene or a specific object that participates in the without scripting. For example, it is possible to select an scene can be rotated interactively, in order to be viewed from object, to query a database about its properties or metadata and several perspectives. A later restoration of the scene to its display an image, or play a video file or a sound which is linked original position is also possible. Rotation is achieved using the mouse. Except for the rotation and properties changing, further

interaction between viewer and scene can be achieved, with or without scripting. For example, it is possible to select an object, to query a database about its properties or metadata and display an image, or play a video file or a sound which is linked to the

object.

Also, a script can be run automatically in order to automatically change any properties of the object or even to this object with another (Fig. 4).

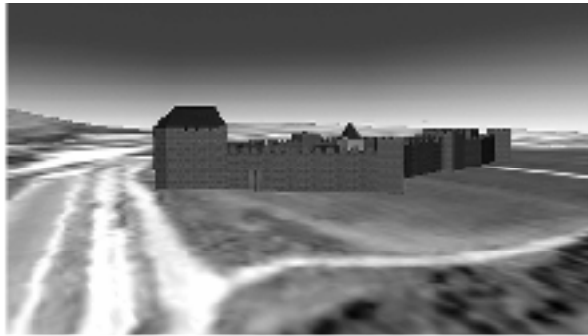
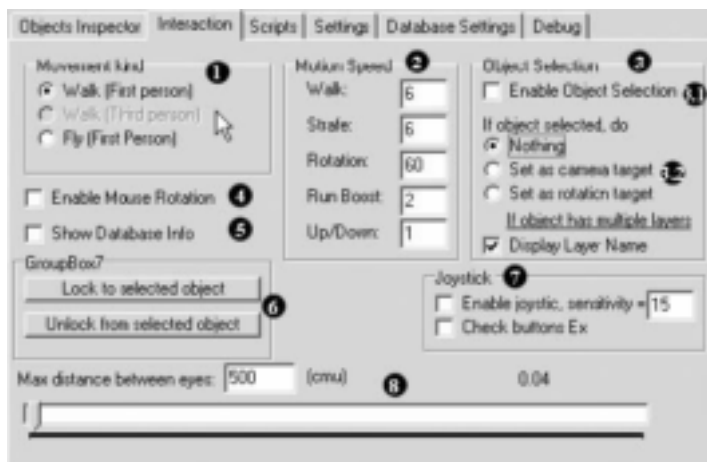


Figure 3. The position, orientation, scale, color or visibility of any object can be changed, giving thus the impression to the visitor that he flies-over or walks-in the virtual world.



1. Virtual viewer 'Walks' or 'Flies' inside the scene.
2. Settings for viewer speed (units/sec or deg/sec for rotation).
3. Object selection settings.
 - 3.1 If checked, user can select an object inside scene by clicking on it. It must be checked, in order DB info for this object to be displayed.
 - 3.2 Defines the action that will be done for the selected object.
4. Permits the rotation of the scene using the mouse
5. Enables Database Information Display for the selected object if scene is connected to a database.
6. Selects an object from Object Viewer instead from scene.
7. Enables joystick support (for motion control).
8. Defines the eyes-parallax in order to create stereoscopic vision

Figure 4. Interaction Tab: Set's up the interaction between user and scene.

4.3 Database connectivity

In order to connect "OpenView®" with a database, the database must be "published" to ODBC. "OpenView®" "talks" to database using SQL queries. This way, it is irrelevant if the database is an Oracle, an Access or any other format. "OpenView®" uses a smart interface that allows the viewer to select the proper table and a connection field, from all available

databases, tables and fields that are published to ODBC. Also, it allows the viewer to select only the wanted fields from the selected table. This connectivity can be done at any time; additionally, it can switch to another database or table during presentation. For convenience, all database settings can be stored into files and can be loaded when needed (Fig. 5). Every time database info is needed about an object, "OpenView®" creates an SQL query and passes the name of the

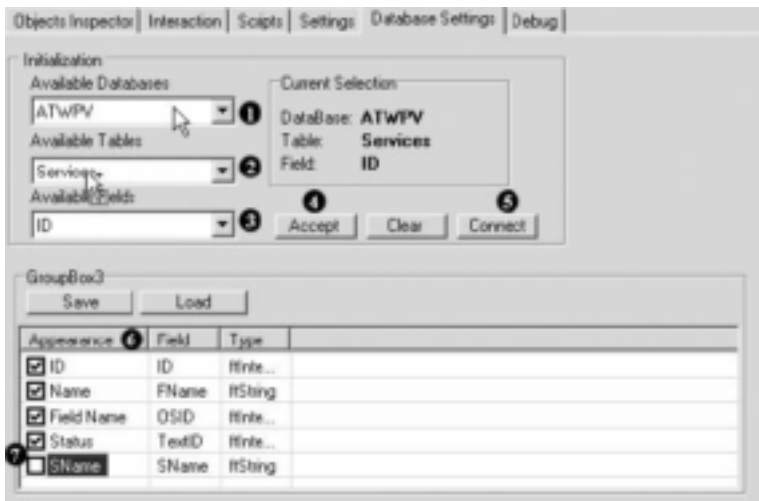
selected object as a parameter to the connection field in order to find the specific record from the selected table. This task is done in the background, so it is invisible to viewer. Then the results are displayed to both left and right windows in order to have stereoscopic view of the info.

4.4 Scripting support

Scripting support is *OpenView*'s most powerful component and is based on "Script Studio", from Tmssoftware. The script language that currently supported is "object pascal" while Visual Basic is on the way. Using scripts, the viewer is able to access every scene component, to change any of its properties, to move or rotate it, etc. Typical examples of scripts are given below:

```
Axis:=TryGLAxis.Create(ControlForm); //create the object
Axis.Name:='Axis1'; //Name for scripter and Inspector
ObjectsCube.AddChild(Axis); //put axis to ObjectsCube
Axis.Visible:=true; // Enable axis
Axis.RememberMe; // Set axis visible to scripter
Axis.Position.SetPoint(3,3,3);
Script 1: Importing a new object
```

```
For i:=0 to 5 do begin
  Axis1.Position.SetPoint(i,i,i);
  AxisCube1.PitchAngle:=i*3;
  sleep(100);
  processMessages;
end;
Script 2: Rotating and moving the object
```



1. Selects a database published in ODBC.
2. Selects a table from database.
3. Selects the field that will be used as **link** to scene objects. Since the field must match exactly with the **name** of the object in scene, it must be of 'string' or 'text' type.
4. User accepts the link and
5. Connects to database's table.
6. All available fields from selected table are displayed. User can change the name (appearance) of the fields that will be displayed in scene.
7. User can select which fields will be displayed in scene. Additionally, above settings can be saved for future usage.

Figure 5. Database settings Tab: Set's up the connection settings between OpenView and a database. OpenView can connect to any database in order to display information about scene objects.

4.5 Producing Stereo Pairs

Since human eyes are located only 6.5cm apart, on average, the geometric benefits of stereopsis are most effective at closer distances, whereas are lost for objects more distant than 30m. Other primary cues (eye convergence and accommodation) and secondary cues (perspective, parallax, size, texture, and shading) are essential for far objects and of varying importance for near ones.

In "OpenView®" the stereo pair production is done in a background process, forced to produce pairs as fast as the combination of the CPU and the graphics card allows. "OpenView®" uses three cameras when running. The first (central camera) displays its contents on the Control Room. The other two (called "left eye" and "right eye") are responsible to create the stereo pairs of images. All three cameras together are called the "Head". Rotation of the "head" rotates all three cameras at once. The position of the two cameras is not fixed but floated. Even if the two cameras lie on the same line that passes from the center of the central camera, their distance can be change. This is done because, most of the times, different scenes need different "eye distances".

There are a couple ways of setting the virtual cameras and rendering the stereo pairs: toe-in and off-axis projections (Burke 1999). The toe-in projection is easier to be implemented (just set the two virtual cameras focus at the same point) but has one major disadvantage: creates stressful stereo pairs due to vertical parallax. The off-axis projection is more difficult to be implemented since it requires a non symmetric frustum which is

not supported by all rendering packages but produces less stressful stereo pairs. By default, "OpenView®" uses the offaxis projection. But it is also possible to use the toe-in projection, too.

4.6 Creating 3D videos

"OpenView®" can be used to create 3D videos of the object or the scene. All "OpenView®"s "head" cameras can save, on demand, single images or .avi files of the rendered scene. Using this ability, creation of 3D videos is an easy task: "OpenView®" is used to produce left and right images of the scene and saves them as image or .avi files. Then an external application (3D Video Creator) imports them and produces the video.

Additionally, the position and the rotation of every camera can be saved into a file and then stored as metadata to the video stream, in order to create geo-referenced 3D videos (Sechidis et al. 2001).

5. CONCLUSIONS – FURTHER WORK

"OpenView®" is a generic tool for stereoscopic representations of 3D objects or VR scenes, enhanced by database connectivity. Performance tests have shown that it can be used for real time presentations, even with the usage of low cost CPU systems. The further growth of "OpenView®" will continue to focus to the implementation of capabilities of generic character. Thus, in

future versions we will add animation, 3D sound, internet abilities (client/server), ability to create scenes using two computers, possibility of measurements and motion that will be based on physics (ODE). For more specialized requirements, the team of "OpenView®" hopes to the help of its users. Since the source code will be available, each user is given the possibility to face and to solve each own problem. All the new solutions will be incorporated in the "OpenView®" so as to become more powerful, useful and functional. Additionally, since "OpenView®", coded in Delphi and using public domain or shareware components, depends on GLScene library, an active open-source project. Therefore any improvement on GLScene or on other components' performance will also improve "OpenView®"'s performance and abilities.

REFERENCES

- ARCO, 2005. ARCO Consortium Augmented Representation of Cultural Objects, available at <http://www.arco-web.org>, (accessed 23-3-2005).
- Brill L. M. 1993. Kicking the tires of VR software. *Computer Graphics World*, June 1993 issue: 40-43.
- Brooks F.P.Jr. 1999. What's real about Virtual Reality?. *IEEE Computer Graphics and Applications*, Nov/Dec issue, Special Report: 16-27.
- Burke P. 1999. Calculating Stereo Pairs, available at <http://astronomy.swin.edu.au/~pbourke/stereographics/stereorender/>, (accessed 23-3-2005).
- Cosmas, J., Itagaki, T., Green, D., Grabczewski, E., Van Gool, L., Zalesny, A., Vanrintel, D., Leberl, F., Grabner, M., Schindler, K., Karner, K., Gervautz, M., Hynst, S., Waelkens, M., Pollefeys, M., DeGeest, R., Sablatnig, R. & Kampel M. 2001. 3D MURALE: a multimedia system for archaeology. In *Proc. of the ACM Conference on Virtual Reality, Archaeology, and Cultural Heritage (VAST2001)*, 297-306, Athens, Greece, 28-30 November, available at <http://www.brunel.ac.uk/about/acad/sed/sedres/nmc/murale/>, (accessed 23-3-2005).
- El-Hakim, S., Gonzo, L., Picard, M., Girardi, S., Simoni, A., Paquet, E., Victor, H. & Brenner, C. 2003. Visualization of Highly Textured Surfaces. In *Proc. of 4th International Symposium on Virtual Reality Archaeology and Intelligent Cultural Heritage (VAST2003)*, 231-240, Brighton, UK, 5-7 November (on CD-ROM).
- Gleue, T. & Dähne, P. 2001. Design and Implementation of a Mobile Device for Outdoor Augmented Reality in the ARCHEOGUIDE Project. In *Proc. of the ACM Conference on Virtual Reality, Archaeology, and Cultural Heritage (VAST2001)*, 161-168, Athens, Greece, 28-30 November, available at <http://arheoguide.intranet.gr/>, (accessed 23-3-2005).
- Gobbetti, E. & Scateni, R. 1998. Virtual Reality: Past, Present and Future. In G. Riva, B. K. Wiederhold & E. Molinari (eds), *Virtual Environment in Clinical Psychology and Neuroscience*: 3-20. Amsterdam: IOS Press.
- Hall, T., Ciolfi, L., Bannon, L.J., Fraser, M., Benford, S., Bowers, J., Greenhalgh, C., Hellström, S.O., Izadi, S., Schnädelbach, H. & Flintham, M. 2001. The Visitor as Virtual Archaeologist: Using Mixed Reality Technology to Enhance Education and Social Interaction in the Museum. In *Proc. of the ACM Conference on Virtual Reality, Archaeology, and Cultural Heritage (VAST2001)*, 91-96, Athens, Greece, 28-30 November, available at <http://www.shape-dc.org/>, (accessed 23-3-2005).
- Hendrix, C. 1994. Exploratory Studies on the Sense of Presence as a Function of Visual and Auditory Display Parameters in Virtual Environment. MSc Thesis, University of Washington, available at <http://www.hitl.washington.edu/publications/HENDRIX/3.html> (accessed 23-3-2005).
- Liarokapis, F., Sylaiou, S., Basu, A., Mourkoussis, N., White, M. & Lister, P.F. 2004. An interactive visualization interface for virtual museums. In *Proc. of the 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST2004)*, 47-56, Brussels, Belgium, 6-10 December.
- LIFEPLUS, 2005. Available at <http://www.miralab.unige.ch/subpages/lifeplus/HTML/home.htm>, (accessed 23-3-2005).
- Lischke, M. & Grange, E. 2005. GLScene: OpenGL Library for Delphi, available at <http://www.glscene.org> (accessed 23-3-2005).
- Ogleby, C. 2001a. The ancient city of Ayutthaya-Explorations in virtual reality and multi-media. *International Archives of Photogrammetry and Remote Sensing*, Vol. 34, Part 5/W1: 6-11.
- Ogleby, C. 2001b. Olympia: Home of the ancient and modern Olympic Games. A VR 3D experience. *International Archives of Photogrammetry and Remote Sensing*, Vol. 34, Part 5/W1: 97-102.
- Pletinckx, D., Callebaut, D., Killebrew, A. & Silberman, N. 2000. Virtual-Reality Heritage Presentation at Ename. *IEEE Multimedia April-June 7* (2): 45-48. Available at <http://www.ename974.org/Ndl/pagina/index.html> (accessed 23-3-2005).
- ScriptStudio, 2005. Available at <http://www.tmssoftware.com/> (accessed 23-3-2005).
- Sechidis, L., Tsioukas, V. & Patias, P. 2001. Geo-referenced 3D Video as visualization and measurement tool for Cultural Heritage. *International Archives of Photogrammetry and Remote Sensing*, Vol. 34, Part 5/C7: 293-299.
- Sechidis L., Gemenetzi, D., Sylaiou, S., Patias, P. & Tsioukas, V. 2004. Openview: A Free System for Stereoscopic Representation of 3D Models or Scenes, *International Archives of Photogrammetry and Remote Sensing and Spatial Information Sciences*, Vol. 35, Part B5: 819-823.