

Proposed Architecture for the Interactive Generation of Virtual Reality Environments Aimed at Improving Spatial Perception by means of Open Source Technologies

SUÁREZ QUIRÓS, J.; RUBIO GARCÍA, R.; GALLEGO SANTOS, R.; MARTÍN GONZÁLEZ, S.

Área de Expresión Gráfica en la Ingeniería

Universidad de Oviedo

Avda. Manuel Llaneza 75 343208 Gijón (Asturias)

SPAIN

quiros@uniovi.es <http://aegi.eutig.uniovi.es>

Abstract: - This work describes the design and implementation of a collaborative virtual reality environment accessible through the Web, aimed at enhancing spatial perception. The main novelty lies in the possibility it offers the teacher to design, modify and manage the virtual environment by means of a simple Web interface that allows a high degree of personalization of the trainer. Open-source tools were chosen for its implementation; these have both demonstrated their great efficacy and integration capacity and, at the same time, provide independence and the capacity to use multi-platform support.

Key-Words: - Collaborative Virtual Environments, VRML, spatial abilities, open-source technologies

1 Introduction

The ability to recreate the image of an object and manipulate it mentally has a significant practical application in fields such as mathematics, physics, architecture or engineering [10]. In the opinion of Mohler [9], this capacity, known as Spatial Perception, is the most important of all the skills required for the exercise of engineering. Although the debate on this question remains open, the skills that this capacity encompasses fit in the framework of two wide areas [11]: the recognition of a three-dimensional object from different angles (spatial relationship) and the perception of the internal structure of a given spatial configuration, by determining the spatial relations between the environment and the individual (spatial visualization).

Technological progress has led to the invasion of high graphic capacity computers in the teaching domain. They have ousted the traditional tests [12] replacing them with more effective materials such as geometrical models prepared in 2D and 3D CAD [4,7] as well as three-dimensional animations and videogames [17]. At present, Virtual Reality provides the user with an expansion of the real world previously unimaginable.

2 Motivation and Situation

The Area of Engineering Graphic Design in the Gijón E.U.I.T.I. (Industrial Engineering School) (University of Oviedo), gives a four-month course in “Computer

Graphics” in the Shared Virtual Campus [14]. This initiative was started in 2000 by the G9 Group of Universities. It is focused on the cognition of the mathematical, geometrical and physical principles that underlie the generation of graphics in a computer, and - due its cross-boundary nature - brings together a heterogeneous group of students, many of them from the industrial engineering and IT areas. After three years teaching this subject, it has been observed that a large number of students show severe cognitive problems regarding the geometrical concepts closely related to spatial perception.

2.1 Concept of Projection

Some students showed their inability to correctly identify the correlations between the source figures and their images. Furthermore, problems of identification occur between the cylindrical projections (isometric perspective, axonometries) and the conical projections (linear perspective).

2.2 The Graphic Pipeline

The set of transformations that enable the geometry defined in a three-dimensional environment from a specific viewpoint to be represented on a peripheral device are known as the *graphic pipeline*. The visualization of the scene in each of the intermediate transformations involves a level of complexity that proves insurmountable for some students.

From the algebraic point of view, the pipeline is defined as the product of a series of closely-related transformations. For the perspective transformation,

included in the pipeline, to be expressed as such, homogeneous coordinates have to be used, which also facilitates the distinction between points and vectors. The transformations undergone by each vertex P belonging to the geometrical model to be represented on the screen, are stated below.

The passage from *local coordinates* - expressed with reference to a system linked to the object - to *world coordinates*, referenced to a point in space, is achieved with the help of a *modelling matrix M*. This operation involves scaling, translations and rotations of the objects to suitably locate and position the object in the environment. Generally speaking, for a point P expressed in local coordinates, the transformation can be defined by the equation

$$P_{\text{WORLD}}=M \cdot P_{\text{LOCAL}} \quad (1)$$

The matrix *V*, or *visualization matrix*, obtains the so-called *view coordinates* (referred to the intrinsic system of coordinates of a camera) from the *world coordinates*: i.e., it changes the system of coordinates from the world reference system to the *intrinsic* system (u,v,n) linked to the camera. Therefore, the complete transformation for any point P becomes:

$$P_{\text{VIEW}}=V \cdot P_{\text{WORLD}}=V \cdot M \cdot P_{\text{LOCAL}} \quad (2)$$

The general expression for the visualization matrix, *V*, is

$$V = \begin{pmatrix} u_x & u_y & u_z & -obs \cdot u \\ v_x & v_y & v_z & -obs \cdot v \\ n_x & n_y & n_z & -obs \cdot n \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3)$$

where (u,v,n) are the axes of the intrinsic system of coordinates of a camera, expressed in a standardized form, and *obs* is the position of the camera.

The visualization associated with the camera consists of a truncated-cone volume called *frustum*, which delimits the region visible to the observer. The *perspective transformation* converts the *frustum* into a cube called *canonical visualization volume* which extends from -1 to +1 in the three directions in space, referenced in the new system of *standardized device* coordinates. The R matrix for perspective transformation corresponds to the expression

$$R = \begin{pmatrix} \frac{2N}{derecha - izquierda} & 0 & \frac{derecha + izquierda}{derecha - izquierda} & 0 \\ 0 & \frac{2N}{arriba - abajo} & \frac{arriba + abajo}{arriba - abajo} & 0 \\ 0 & 0 & \frac{-(F+N)}{F-N} & \frac{-2FN}{F-N} \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad (4)$$

where *right*, *left*, *up* and *down* are the coordinates that define the visualization plane where the image in perspective is projected (generally, the plane of the *frustum* perpendicular to the visual coordinate closest to the camera), and *N* and *F* correspond to the distances measured in the direction of the visual coordinate where we encounter the planes delimiting the frustum called *near* (*N*) and *far* (*F*). Only the objects situated between the two planes will be represented in the output device. The peculiar perspective transformation chosen greatly facilitates the task of cutting out the entities that are not completely included in the visualization volume, by using planes parallel to the coordinates at unit distances as cutting devices.

Finally, the coordinates of the vertex on the graphic device must be determined by the *screen transformation*. This means mapping the image projected on the visualization plane (perpendicular to the Z axis of the system of cut-off or boundary coordinates) on the screen. Care must be taken that the aspect relationship of the visualization plane and of the screen are identical in order to prevent deformation of the image. Assuming ($X_{\text{CUT-OFF}}, Y_{\text{CUT-OFF}}$) as the coordinates of a point inside the canonical visualization volume, and ($X_{\text{SCREEN}}, Y_{\text{SCREEN}}$) as their transform on the display device, the transformation matrix, *S*, is defined as

$$S = \begin{pmatrix} \frac{x_{MAX}}{2} & 0 & 0 & x_{MIN} + \frac{x_{MAX}}{2} \\ 0 & \frac{y_{MAX}}{2} & 0 & y_{MIN} + \frac{y_{MAX}}{2} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5)$$

Therefore, the coordinates on the display device are obtained by means of the formula

$$P_{\text{SCREEN}}=S \cdot P_{\text{CUT-OFF}} \quad (6)$$

The reasons that explain the appearance of this situation basically derive from the students' lack of a prior training in graphic language, the bi-dimensional nature of illustrations (projections mask the true 3D nature), and the lack of interactivity, which helps to

reinforce the cognitive process of learning (when different viewpoints can be chosen interactively). To sum up, it is essential to find tools that allow these needs to be met within the Web environment, where this subject is taught.

3 CVE Environments

Taking into account the possibility of improving spatial ability through training [11], the design and subsequent implementation of resources aimed at the exercise of these skills was proposed, based on the following requirements:

To develop tools with three-dimensional graphic capabilities, with a sufficient degree of immersivity to be attractive to the students, and that provide a multi-user support allowing collaborative work.

To facilitate the configuration and management of the tool by the teacher, from the academic and administrative point of view.

To guarantee the accessibility of such tools for all the students, by making them independent from any specific physical location, and to promote the availability of future updates, revisions and extensions.

To ensure that such resources are user-friendly so that the interaction with them does not constitute an obstacle.

To use open-source development environments in order to minimize costs and universalise results, taking care to ensure the independence of the resources with regard to platforms and operating systems.

The response to these concerns is found in the Lab3D project, aimed at the creation of a Collaborative Virtual Environment (CVE) [2]. This is a synthetic scene, generally three-dimensional in nature, envisaged for action and interaction, where the users can interact with each other or with other agents belonging to the virtual world, which they can access through simple graphic interfaces created for the Web or through sophisticated peripherals such as VR headpieces or datagloves. The CVEs have achieved a very high level of technical adaptation to the requirements demanded. Furthermore, there are well-founded pedagogical reasons that justify the use of CVEs as efficient tools in the teaching-learning process [8]:

A sensation of common space, as all the participants carry out their activity in the same place, whether real or fictional, with similar characteristics that do not depend on any external factor.

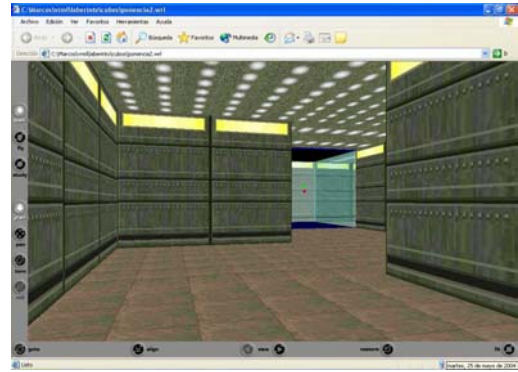


Fig. 1 The metaphor of the 3D Labyrinth

After analysing the performances offered by these environments in Lab3D and assessing the efficacy-attractiveness binomial, a spatial metaphor based on a three-dimensional labyrinth (fig. 1) was chosen. Here, the user is located at the entrance to the labyrinth and has to find its only exit. The user can move in all spatial directions allowed by the geometry of the place, using a three-dimensional representation of the labyrinth and the keyboard and/or mouse. To help the user reach the goal, two orthogonal views (plan view and elevation) of the structure are continuously displayed. These views show both the position of the user and of the labyrinth exit. In this way, the students are forced to make intensive and reiterative conversions between the three-dimensional space in front of their eyes and its bi-dimensional projection (just as the engineer does when interpreting a drawing), in order to decide which way to go when the path forks. As has been previously remarked, this process is a very effective manner of enhancing spatial abilities.

A sensation of being present, as the participants adopts virtual identities in the CVE (commonly called avatar), where they find other avatars and agents with their patterns of behaviour, with whom they can interact and develop tasks jointly. The collaborative strategy proposed in Lab3D consists of the simultaneous participation of various users in three different modes:

Competitive mode: The participants have to find the exit from the labyrinth in a limited time; the winner is the one who achieves it in the shortest time. In this case, the sensation of being present in the place contributes to the addition of a competitive factor that stimulates the solution of the activity.

Collaborative mode: A group of participants is formed, represented by individual avatars, who must unite their efforts in order to find the exit in a limited time. In this case, the sensation of being present enhances collaboration and the exchange of ideas and opinions, adding a reflexive value to the activity.

Mixed mode: Groups of participants, with in-group communication, compete with each other to solve the labyrinth problem, thus uniting the effects of the two modes described previously.

An effective form of communication, as the participants, dispersed and interacting together in the metaphor, do not only use the more conventional resources available to them for the exchange of ideas and opinions (chat, voice, videoconference). They also use non-verbal communication - so characteristic of human relationships - through gestures and expressions linked to the avatars and agents of the environment [1].

4 Technical Aspects of CVEs

CVE environments present a series of design difficulties, that derive basically from the synergy of difficulties involved in the integration in a single application of complex software developments (distributed systems, graphic applications and interactive applications), as well as a series of collateral services that have to be added to achieve proper operativity (databases, user authentication, etc) [13]. The most significant ones are detailed below, together with the solutions provided in the Lab3D design.

4.1 Graphic Driver and Display Devices

The choice of these components is a strategic decision of the first order. The graphic driver is responsible for generating the virtual scene from its geometrical design. The final display is an expensive process, in terms of computer resources, reserved till recently for high performance work-stations. Today, graphic processors give personal computers reasonable capacities, enhanced by the appearance of standards such as the OpenGL library [16].

Considering the requirements imposed on the trainer, VRML (Virtual Reality Modelling Language) [15] was used to implement the graphic functions of the CVE. This is a script language for the description of three-dimensional scenes that allows the creation of interactive models based on a tree-like structure known as grafo. The following reasons led to the choice of this language:

- * Independence from the platform and/or the operative system chosen. The VRML scene display is accomplished from the navigator itself, thanks to the inclusion of auxiliary programs known as "plug-ins" that add the necessary operativity.

- * As this is a script language, the creation and subsequent management of the scenes is simple (as they can be edited in text mode) and economical, as

the bandwidth requirements for interaction are very limited.

- * VRML is an open-source technology, not subject to the arbitrary decisions of the large multinationals in the sector.

- * It can be easily extended with the help of auxiliary languages, such as Java or PHP, which allow new node types to be created, or through the inclusion of a multi-user support.

4.2 Processor Unit

The heart of the CVE is the processor unit [13], a logical component that receives the events originated by the input devices controlled by the participants and computes the changes that these actions generate, not only on the display seen by the user that caused them, but on those of all the participants (fig. 2).

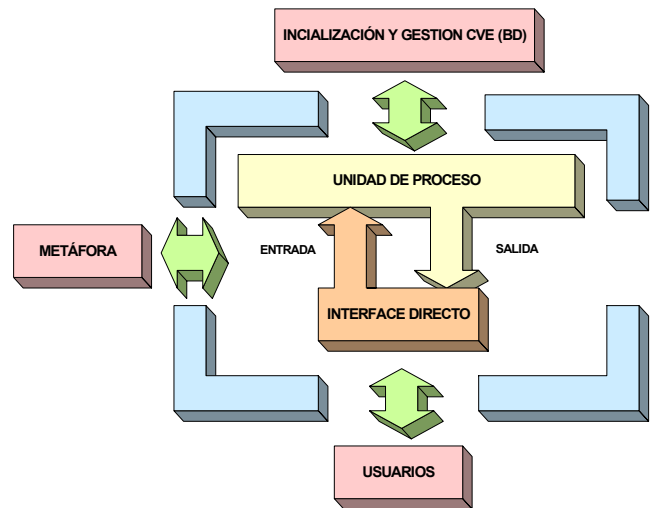


Fig 2 Functional diagram of the CVE

It simultaneously receives information about the position of the users in the metaphor, as well as about the state of agents and avatars, some of which are modelled and governed exclusively by the processor unit. Finally, it orders the graphic driver to generate an updated image of the environment and continuously manages the back-up database, which stores all this information.

In the project described, the CVE processor unit resides physically in a medium-performance computer using the Linux operative system, and has been programmed in PHP (an acronym for "PHP: Hypertext Preprocessor") [5]. This is a high-level interpreted language embedded in the HTML code and executed in the server devices; it allows the information from Web formats to be processed, and pages with dynamic content to be generated, or cookies to be sent or received. The reasons on which the use of this language is based are:

* Its excellent capacity for integration in the Web environment, which greatly facilitates its integration with the VRML motor.

* The sound infrastructure it provides for managing a large number of databases, protocols and APIs (Application Programming Interface).

* Its programming power is similar to that of other high level languages, such as C, Java or Perl, from which it takes the greater part of its syntax.

* It is an open-source product, which is currently installed in millions of servers all over the world and is supported by a vast and active community of programmers.

The operativity of PHP is developed thanks to the client-server architecture that structures Lab3D, and is described in more detail in the following sections. The VRML 2.0 [6] specification introduces the description of techniques to animate the static contents of the scene. Among them, it allows for the definition of nodes prepared for the execution of external scripts (ScriptNode), which can act on the graphic pipeline of the scene, generate events and receive notifications. This has facilitated the link between the graphic driver and the processor unit.

4.3 Communication Infrastructure

Together with the graphic driver, the architecture and characteristics of the data transmission network used for the interconnection of the participants are of crucial importance in the approach to the design of the CVE. The heterogeneity of the clients and the limitations of the communication networks impose severe constraints on the design of the CVE.

The full effectiveness of the environment is achieved when the user feels that the virtual scene resides in his own computer and that the latter responds immediately to the effect of his actions on it: i.e. when the metaphor behaviour in real-time [1]. Nevertheless, due to the fact that this is a distributed interaction that generates latencies and delays, this situation is never fully attained. Its effects are of special concern when emulating direct actions between users, such as collisions. In fact, it is the need for real-time interaction that defines the CVE processes and architecture. There are a large number of concurrent tasks and, unlike other complex systems, they almost all require immediacy to achieve a satisfactory experience.

Considering the decisions made with regard to the graphic driver and architecture of the Web-based CVE application, we preferred to use the TCP protocol (Transmission Control Protocol) over IP (popularly known as TCP/IP), one of the most widely-used today. This is a reliable system of rules that guarantees the transmission of the data thanks to

an advanced system of verification and retransmission, with a semantic system that maintains the correct order of package transmission.

The choice of the protocol and the nature of the data transmissions implicitly determine the architecture chosen for the CVE environment. The point-to-point type connection limits the use of the TCP/IP protocol with a very large number of users, as it is difficult to establish and maintain connections between each pair of users. In order to avoid this problem, the topology adopted was to have one of the computers acting as the server while the other - called client - establishes a TCP/IP connection with the server. The latter acts as a central manager of the CVE, receiving data from each client and sending them again, after appropriate processing, to the other users.

The client-server architecture is preferred whenever there is any type of task to be undertaken that affects all the participants, as it allows the CVE processor unit functions to be brought into play. In this case, the processor functions are accessed through a PHP-type programming language to carry out such tasks. Numerous functions of the spatial perception trainer (control of the time used by the participants, collaborative activity monitoring, environment initialisation interface) become possible thanks to the use of this model.

The Web Apache server (capable of supplying HTTP services) was chosen for the implementation of the CVE; it is one of the open-source applications that has had most success in the Internet field, due to the power it offers, the facility of installation and maintenance, and the technical support of a numerous community of users.

4.4 Data Management in the CVE

The difficulties involved in trying to ensure an effective sensation of reality for the participants in the CVE has already been mentioned. Singhal [13] illustrates this issue stating that "*it is impossible to share a dynamic virtual world that changes frequently and simultaneously guarantee that all the users access identical versions of it*". His statement recalls the severe constraints that quantum mechanics imposes on nature through Heisenberg's Principle of Indetermination. The delay imposed by the channels of communication in the transmission of the updates that occur in the CVE is at the origin of the loss of consistency between the users. If the scene is refreshed frequently, it is impossible to guarantee that all the clients have received the updates prior to the last modification made.

One of the techniques most frequently employed to avoid this phenomenon consists in the use of

centralized data repositories: these ensure that all the participants have a homogeneous view of the environment. For this reason, all the shared variables of the CVE must be stored so that the clients read the information regarding the state of the environment in the data repository. The existence of a client-server structure favours the choice of this solution. The simplest way to implement this proposal is to use a database. In the proposed CVE, MySQL is used to store the shared status of the environment, while the processor unit manages the connection and the enquiries from the users to the database server.

4.5 Remote Management of the CVE

Another advantage deriving from the adoption of a client-server structure is that it allows the CVE to be initialised and managed from simple Web interfaces accessible to the teacher. The centralization in the server of the data repository means that there is a single point of access to all the information stored in the CVE. Furthermore, the level of integration between the open-source technologies used (Web server, graphic driver and database) makes it possible to generate Web formats that allow enquiries to be sent to the database (environment initialisation, administrative tasks, result monitoring), and their effect to be shown immediately on the graphic visualization of the metaphor.

5 Conclusions

The new Virtual Reality technologies applied to teaching allow more efficient tools to be constructed for training in spatial perception, by taking advantage of participation in a collaborative environment and minimizing the creative effort and management demanded from the teachers. The use of open-source technologies without having recourse to sophisticated devices drastically reduces the investment required, which means that cost-effectiveness is considerably improved.

References:

[1] C. Bouras et al., Aspects of a Collaborative Learning Environments using Distributed Virtual Environments, *ED-MEDIA 2001*, pp. 173-178, Finland

[2] E. F. Churchill, D. N. Snowdon, A. J. Munro, *Collaborative Virtual Environments: Digital Places and Spaces for Interaction*. Springer Verlag, 2001

[3] S. Diehl, *Distributed Virtual Worlds*, Springer Verlag, 2001

[4] Ann Eisenberg, *An Educational Program for Paper Sculpture: a Case Study in the Design of Software to Enhance Children's Spatial Cognition*, Doctoral Thesis. University of Colorado (EE.UU.).

[5] PHP Specification (<http://www.php.net>)

[6] VRML Specification (<http://www.web3d.org/technicalinfo/specifications/>)

[7] W. E. Mack, Computer.aided Design Training and spatial visualization ability in gifted adolescents, *Journal of Technologies Studies*, 21(2), pp. 57-53, 1995.

[8] H. McLellan, Cognitive Issues in Virtual Reality, *Journal of Visual Literacy*, 18(2), pp. 175-199, 1998.

[9] James Mohler, Using Interactive Multimedia Technologies to Improve Student Understanding of Spatially-dependent Engineering concepts, *Minutes of the International Graphicon Conference on Computer Geometry and Graphics*, Nyzhny Novgorod, Rusia, 2001

[10] C. Newlin, The Total Concept of Graphics and Design in the Engineering Curriculum, *Engineering Design Graphics Journal*, 43(2), pp. 21-22, 1979.

[11] J. Pellegrino, D. Alderton, V. Shute, Understanding Spatial ability, *Educational Psychologist* 19, pp. 239-253, 1983

[12] Rebecca Robichaux, Predictors of Spatial Visualization: Structural Equations Modeling Test of Background Variables, *Journal of Integrative Psychology*, 2(3), 2002

[13] S. Singhal, M. Zyda, *Networking Virtual Environments*, Addison-Wesley Pub., 1999

[14] J. Suárez Quirós, R. P. García Díaz, P. I. Álvarez Peñín, R. Gallego Santos, Nuevas Tecnologías en la Enseñanza Virtual a través de la Red: Gráficos por Computador en el Campus Virtual de AulaNET. *Minutes of XIV International Congress of Engineering Graphics* (Santander), 2002.

[15] J. Suárez Quirós, R. P. García Díaz, Un Standard Gráfico para Internet: el Lenguaje VRML, *Minutes of IX International Congress of Engineering Graphics* (Bilbao), 1997.

[16] J. Suárez Quirós, *Gráficos por Computador*, ISBN: 84-688-1648-5 University of Oviedo, 2003

[17] E. Wiebe, Visualization of Three-dimensional Form: A Discussion of Theoretical Models of Internal Representation, *Engineering Design Graphics Journal*, 57(1), pp. 18-28, 1993.