

# Semantics of immersive web through its architectural structure and graphic primitives

Holman Bolivar<sup>1</sup>, Alexander Pacheco<sup>1</sup>, Rubén Gonzalez<sup>2</sup>

<sup>1</sup>Facultad de Ingeniería, Universidad Católica, Bogotá, Colombia

<sup>2</sup>Departamento lenguajes y sistemas informáticos, UPSAM, Madrid, Spain.

**Abstract** — Currently, practices and tools for computer-aided three-dimensional design, do not allow the semantic description of objects constructed in some cases specified notations as handling layers, or labeling of each development itself. The lack of a standard for the description of the elements represents a major drawback for using advanced three-dimensional environments such as the automation of search and construction processes that require semantic knowledge of its elements.

This project proposes the development the semantic composition from the hierarchy of three-dimensional visualization of graphics primitives used to construct three-dimensional objects, taking into account the geometric composition architecture of standard 19775-1 of the International Electrotechnical Commission of the International Organization for Standardization

For the development of semantic composition use the methodology methontology proposed by the Universidad Politécnica de Madrid, because it allows the construction of ontologies about specific domains, limiting the domain by defining classes and subclasses, relationships and the generation of instances a framework for resource description on web ontology language.

Keywords— Computer aided design, Geometry, Artificial intelligence.

---

## I. INTRODUCTION

---

To build a building first, it identifies the place where it will be built, then a study of the topography and is adapted to resist the structure then becomes the frame of the building, which is covered with either concrete, bricks or other material, to the end make the veneer of floors, walls and ceilings. In an immersive virtual environment, no restrictions, buildings can be built where the income is on the top floor and first floor or a pyramid with the point adversely to heaven, something that in reality would be impossible, because it hinders the usability of the building.

Therefore it is necessary 3D objects with simple structures, but consistent and above all reusable, and at this point where a hierarchy and semantic construction is relevant, because an object with proper semantic definition will allow the

generation of new objects, simplifying development and saving computation and memory space [15].

The hierarchical construction was developed taking into account the hierarchical model proposed by Donald Hearn [10] with base in OpenGL library, and the hierarchy of render and visualization O3D, VRML, X3D, Java3D and Xj3D [8].

The hierarchy defines a tree of object construction, which makes up the immersion environment, taking a universal reference system common, on which to make the necessary transformations to give consistency to the 3D environment.

Based on the axiom that an environment is composed of objects, you must first present the construction of an object, because of complex objects to develop is the human figure, especially the face, this will be the example to be taken for object construction, while simulating a landscape for the construction of outdoor environments and the living room of a palace for interior construction.

---

## II. GRAPHICS PRIMITIVES

---

The software used in the production of computer graphics in general have two groups, one special purpose which is designed to generate graphs or diagrams in any area of application without worrying about the procedures necessary to produce graphic images which normally has a work area and a series of menus and submenus that allow the development of a task graph specifies [11]. The other group is the general purpose provides a library of graphics functions that can be used in programming languages such as C, C++ or Java, the basic functions of a graphics library include those to specify components of the image as lines, polygons or areas, set the color, select a view of a scene and applying rotations or other transformations, some examples of general graphical programming packages are GI, OpenGL [2], VRML and Java 3D, a set of graphic functions is called application programming interface for computer graphics (CG API) [6].

For two-dimensional graphics applications, visualization is achieved through the transfer of the positions of the reference frame in world coordinates to pixel to positions in the plane of the screen, a process called pipeline [14]. The Three-Dimensional visualization operations are more complex because there are many more options for representation of the

scene, because at the time of modeling a three dimensional object, this is defined by a closed set of surfaces. And for some applications may also need to enter information about the internal structure of the object [10].

To describe an image is first necessary to establish a suitable Cartesian coordinate system, called the reference coordinate system. For example, it defines a line segment giving the position of the two end points, while a specified polygon given the set of positions of its vertices. These coordinates are stored in the description of the scene, along with the box outline, which are the values x, y, maximum and minimum for each object [9].

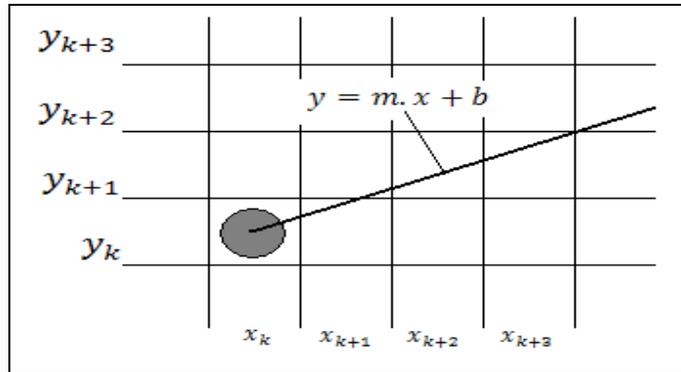


Figure 1. K-th step of the algorithm Bresenham [10].

Figure 1 illustrates the k-th step, where you draw the pixel in position  $(x_k, y_k)$  and must decide to draw pixel in column  $x_{k+1} = x_k + 1$ , where options are the positions  $(x_k + 1, y_k)$  and  $(x_k + 1, y_k + 1)$ . In order to decide, be perform positions sampling  $x_k + 1$ , labeling the vertical separation of the pixels with respect to linear mathematical way, by assigning  $y_{low}$  and  $y_{up}$ , which are calculated as follows:

$$y = m(x_k + 1) + b$$

$$y_{low} = y - y_k = m(x_k + 1) + b - y_k \quad (1)$$

$$y_{up} = (y_k + 1) - y = y_k + 1 - m(x_k + 1) - b_k$$

To determine which of the two pixels is closer to the line, can make a very efficient test based on the difference between the two separations of pixels:

$$y_{low} - y_{up} = 2m(x_k + 1) - 2y_k + 2b - 1 \quad (2)$$

Can submit an implementation of this algorithm for slopes within the range  $0 < m < 1,0$ . The pixel coordinates, are introduced at the ends of the line and the pixels are drawn starting from the leftmost [19].

In the circles is defined from any point  $(x, y)$  on the border of a circle of radius  $r$  that satisfies the equation  $f_{circ}(x, y) = 0$ . If the point is inside the circle, the function will take a negative value, while if the point is outside the circle, the value of the function is positive [19]. The relative position of any point

$(x, y)$  can be determined by checking the sign of the generating function of the circle:

$$f_{circ}(x, y) \left\{ \begin{array}{l} < 0 \text{ if } (x, y) \text{ inside the circle} \\ = 0 \text{ if } (x, y) \text{ on the circumference} \\ > 0 \text{ if } (x, y) \text{ outside the circle} \end{array} \right\} \quad (3)$$

The function (3) presents the intermediate points located in the vicinity of the circular path at every step of sampling. Thus the generating function of the circle is a decision parameter midpoint algorithm, and can determine the incremental calculations required for this function [10].

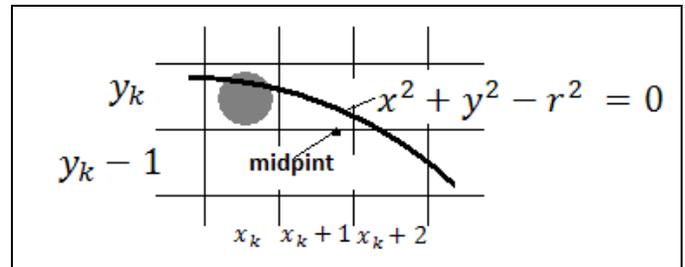


Figure 2. Midpoint between candidate pixels [10].

Figure 2 shows the midpoint between the two candidates for the position pixel sampling [1].

For the ellipse, begun of a modified circle whose radius varies from a maximum value in one direction until a minimum value in the perpendicular direction. The segments drawn inside the ellipse in both directions are called the major axis and minor axis [3].

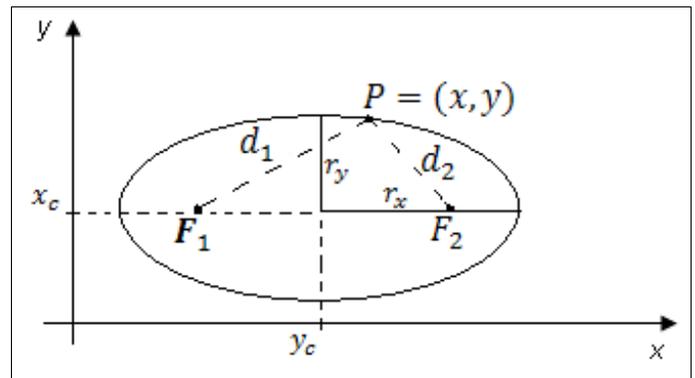


Figure 3. Generated ellipse with foci  $F_1$  and  $F_2$  [10].

The polynomials are useful in several graphics applications, including the design of object shapes, the specification of animation paths and representation of trends from a discrete set of data points. The design shapes of objects or paths of motion is usually done by first specifying a few points to define the general shape of the curve and then adjusting the selected points by a polynomial [5]. A polynomial of degree n in x is defined as:

$$y = \sum_{k=0}^n a_k x^k = a_0 + a_1 x + \dots + a_{n-1} x^{n-1} + a_n x^n \quad (4)$$

Where  $n$  is a nonnegative integer and  $a_k$  values are consistent with  $a_n \neq 0$ . One way to perform the curve fitting is to construct a cubic polynomial curve section between each pair of points specified. Each curved section is described in parametric form as follows:

$$\begin{aligned} x &= a_{x0} + a_{x1}u + a_{x2}u^2 + a_{x3}u^3 \\ y &= a_{y0} + a_{y1}u + a_{y2}u^2 + a_{y3}u^3 \end{aligned} \quad (5)$$

Quadratic surfaces are objects that are described from quadratic equations, such as spheres, ellipsoids, bulls, paraboloids and hyperboloids, these surfaces can be played by rational splines [7].

Bezier curves are line segments that are curved depending on the movement of control points. The Bezier polynomial degree is determined by the number of control points to be approximated and their relative position. You can specify the path of the curve in the vicinity of the control points using combination functions, an array of characterization, or the boundary conditions [4].

Can set the polynomial relationship generating  $n + 1$  control points, indicated as  $P_k = (x_k, y_k, z_k)$ , with  $k$  ranging from 0 to  $n$ , the points are combined to produce the position vector  $P(v)$ , which describes the trajectory of a polynomial approximation function Bézier between  $P_0$  and  $P_n$ .

$$P(v) = \sum_{k=0}^n P_k BEZ_{k,n}(v), \quad 0 \leq v \leq 1 \quad (6)$$

Where function Bezier combination  $BEZ_{k,n}(v)$  are the Bernstein polynomials [10].

$$BEZ_{k,n}(v) = C(n,k)v^k(1-v)^{n-k} \quad (7)$$

Where the parameter  $C(n,k)$  arises from the binomial coefficients:

$$C(n,k) = \frac{n!}{k!(n-k)!} \quad (8)$$

Normally a Bezier curve is a polynomial of degree less than the number of designated checkpoints, this is the case that three points, generate a parabola, four points a cubic curve and so on [20].

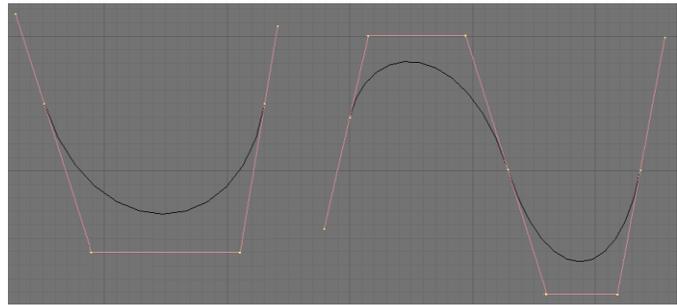


Figure 4. Bezier curves [18]

Two groups of Bezier curves can be used for the construction of the surface of an object. Vector parametric function of a Bezier surface is created with the Cartesian product of the combination of Bezier functions:

$$P(u, v) = \sum_{j=0}^m \sum_{k=0}^n P_{j,k} BEZ_{j,m}(v) BEZ_{k,n}(u) \quad (9)$$

Where  $P_{j,k}$  specifies the location of the  $(m + 1)$  by  $(n + 1)$  control points. In a Bézier surface, control points are connected by dashed lines, and lines show the curves with constant  $u$  and  $v$  constant. Each curve is drawn with constant or varying  $v$  in the range of 0 to 1, keeping a fixed value or within the unit interval. The zero-order continuity is obtained by matching the control points within the limits [16].

---

### III. WEB 3D STANDARD

---

Extensible 3D is based on the search for a specification for building interactive 3D environments, some of the technology XML (Extensible Markup Language) which emerged in 1996 incorporating a standard subset of SGML (Standard Generalized Markup Language) designed to enable the exchange of information between different platforms, hardware, software and various storage and filing systems (databases, electronic publishing, etc.) regardless of their complexity. XML 1.0 specification presents a special type for the definition of documents called DTD (Document Type Definition) The DTDs were designed for document-centric systems, and are suitable for electronic publishing applications SGML [13]

When taking the representation of 3D worlds and objects achieved with the VRML language and bring it to an orderly and rigorous specification for reading XML as it has obtained the X3D specification that enables implementation and development of 3D environments with different programming languages according to script as it uses Java libraries such as OpenGL [17].

X3D is standardized by ISO 19775, 2004 consists of two parts the first introduces and details the components and architecture bases X3D while the second delves into the interface to the scene, has been supplemented with 3 standards, the first is ISO 19776, 2005 which specify the encoding of the fields, nodes and scenarios in VRML generated from XML Schema, the second is the ISO 19777 standard in 2005 that provides a format for building interactive services with 3D

scenes from ECMAScript format which is the basis for Netscape's JavaScript and Adobe ActionScript, and the second part of this standard provides the format for building Java from SUN Microsystems, finally it complements the ISO 19774 standard named Humanoid Animation is an abstract representation for modeling three dimensional human figures, which can be created with modeling tools using motion capture data [12].

#### IV. HIERARCHICAL CONSTRUCTION MODEL

A 3D object is composed of graphics primitives, such as the coordinate system, the line segment, splines, circles, polyhedra, blobby objects among others [10]. The basic object that displays a three-dimensional cube is a cube graph and need a universal reference system, a local reference system, eight vertices and twelve line segments, such as 2D objects are contained between objects 3D, can say that there are two different levels, but each of these levels can be subdivided into other levels, such as lines, and splines, since a line can be a special type of spline.

Therefore suggest the levels of hierarchy from the graphics primitives that are used and other higher level from combinations of these primitives, thus able to model the 3D universe and have it nested, it is clear that environment consists of 3D objects, but there are exterior and interior environments and objects that are specific to each one, for example a tree is an object of external nature or a chair that is of internal nature, at this point it would conduct an investigation statistical and see how many and what type objects are exterior and interior type which in the present investigation this idea is left raised and the relationship of objects made from the experience of the authors [15].

##### 1. Level 0 Reference System

*Universal reference system: (SRU)* Since this system is built around the immersive virtual environment, is the first thing to be defined at the time of building an immersive virtual environment, this environment can have settings like a park, a palace, a forest, a street The moon, in short, many scenarios, the classification of these scenarios belong to a lower level.

*Local reference system: (SRL)* This is characteristic of each object, whether they are scenes like that are only objects, including those that still other objects are part of larger objects such as door and the wall or the wall and a picture of Van Gogh. Static objects take the frame of the dynamic object to which they belong, in order to optimize memory spending, at this point is a new problem is that once assigned a static object to a dynamic object, this will be a single object and not be separated, to separate, the only way is to remove the entire object and re-import it alone, which requires access to a library of objects

##### 2. Level 1 Surfaces

*Polyhedra based surface: (SBP):* These areas are unique in that they are open and can see only a part of such surfaces are part of a table top, doors and in further building the walls, as shown in Figure 5.



Figure 5. Object interior environment, based polyhedral

*Based on quadratic surfaces: (SBC)* These areas are closed, and normally only uses a portion of the object that generates, and can be used in the construction of domes, balloons, cups entity other.

*Spline based surface: (SBS)* Most objects are made from splines surfaces, these are irregular and not follow a precise mathematical formula but obey polynomial mathematical functions, find the equation to build the landscape can generate a set of equations that cannot be resolved

##### 3. Level 2 objects environment

*Regular environment objects (OER)* At this level, it is very difficult to classify the object because there is not a discrete and finite number of how many there are and what they are, but you can define a property from property and is that these objects are those which limit the environment, such as walls, floor and ceiling of a room.

*Irregular environment objects (OEI)* Just as objects with regular geometric structures, there will be environments with non-regular geometric structures, constructed from randomly modified meshes.

*Environment objects compounds (OEC)* Most objects are composed of a regular and other irregular, such as the facade of a house

##### 4. Level 3 Dynamic objects

Receive dynamic feature, because they are objects that can be other objects, but by themselves are objects in the environment, such as a chair in a room by itself is an object, but four or six of them along with a dining table are a might think that the dining room is a 3D object, and should be considered the table and chairs as separate objects, but the same happens with a door and a house or the eyes, nose and face, so these objects are dynamic and allow the same object to obtain more copies, for example of the same face can get two characters as presented in Figure 6.

Like the environment objects, dynamic objects can be classified into regular dynamic objects, such as a car tire in irregular dynamic objects, like the face of figure 6 and composite dynamic objects such as a hat

- *Dynamic objects regularly (SDR)*
- *Dynamic objects irregular (SDI)*
- *Dynamic objects compounds (SDC)*

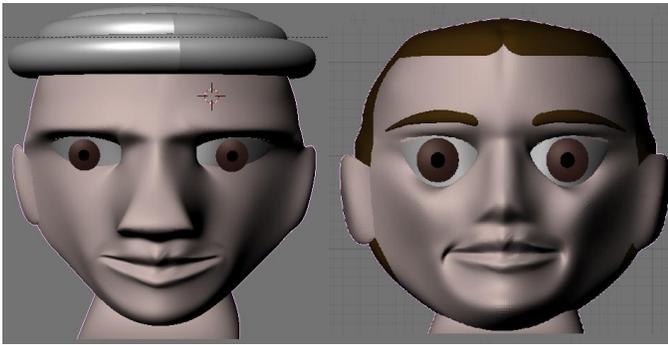


Figure 6. Dynamic objects

V. EXAMPLE OF A TREE HIERARCHY

Every 3D object needs a universal reference system and the local reference system of the object from which it generates a surface such as polyhedron, then this is deformed to build a Bézier surface, to finally build a dynamic object compound.

The path from the root of a static scene graph to the specified node is the path to static scene graph node. As a path of a scene graph is exactly one static node, there is a scene graph path for each node static on stage. All scene graph path in a scene graph completely specifies the node state information static. This information includes the location, orientation and size in the object reference. Consequently, the visual attributes of each object depend only on its scene graph path. The visualization pipeline takes advantage of this and displays the static nodes in the most efficient order. The graphical representations of a scene graph can serve as a design tool or documentation for the 3D buildings. To design an environment draws a scene graph using a set of standard symbols. After completing the design, this scene graph is the specification for the program. After completing the program, the same scene graph is a concise representation of the program. It is possible to design an illegal scene graph. As you can see in Figure 7. This scenario is illegal because it violates one of the design properties. The problem is the two objects SRL (Local Reference System) with the same subject TC (TrasformCoord) as a child. There can be only one path from the object coordinate to the static object. You might think that the structure shown in Figure 7 defines three visual objects in a virtual universe. But the scene graph defines two visual objects using the visual re (SDC) on the right side of the figure. Conceptually, each object that PBS sponsors share the SDC issue could put an image in the visual object in different locations. However, it is an illegal scene graph because the parent-child arc is not a tree. In this example, the result is that the object SBP has more of a father.

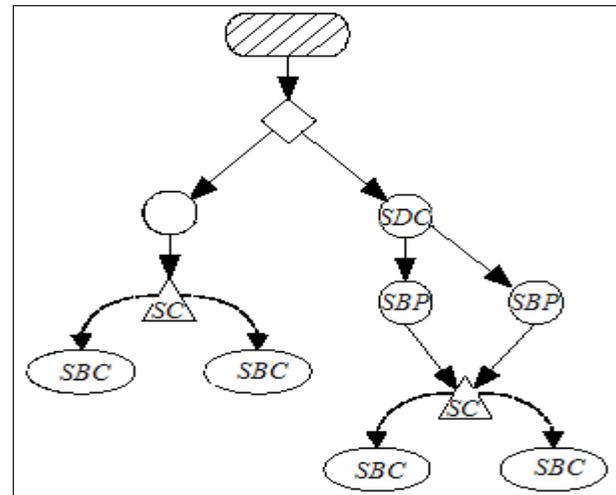


Figure 7 Illegal Construction

The explanations of the tree and the structures are correct. However, the system reports the error in terms of child-parent relationship. One result of the limitation of the tree structure is that each object SBP is limited to one parent. Figure 8, with a parent for each object PBS, shows a possible solution to this scene graph.

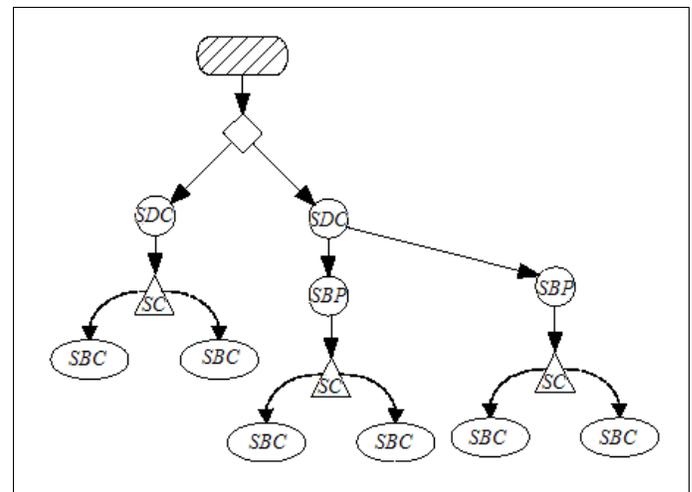


Figure 8 Correction illegal construction

Each scene graph has only one Universal Reference System (SRU). This object has a reference list of Local Systems (SRL), which provides a reference to a point in the virtual universe.

A dynamic object is the root of a sub-graph, or branch graph. There are two categories of graphics sub-scenarios: the view branch graph and the content branch graph. The content branch graph specifies the contents of the virtual universe - geometry, appearance, behavior, location, sound and lights. The view branch graph specifies the viewing parameters, such as viewing position and direction. Together, the two branches most of the work specified in order to make the display

---

REFERENCES

---

- [1] V. ANAND; "Computer graphics and geometric modeling for engineers"; John Wiley, New York, 1993, pp. 24-267, ISBN 0471599603
- [2] E. ANGEL; "Interactive Computer Graphics: A top-Down Approach Using OpenGL"; Addison Wesley, University of New Mexico, 2008, pp. 125-634, ISBN 978-03-2153-5863
- [3] I. ANGELL; "A practical introduction to computer graphics"; Paraninfo, Madrid, 1986, pp. 12-98, ISBN 8428314926
- [4] M. BERG; "Computational geometry: algorithms and applications"; Springer, Berlin, 2000, pp. 74-286, ISBN 3540656200
- [5] J. BRADBERRY; "Computer graphics environments"; Sams Publishing, Carmel, 1993, pp. 54-326, ISBN 0672303051
- [6] M. ESCRIBANO, "3D Graphics Programming", Addison Wesley, Madrid, 1994, pp. 56-87, ISBN: 9788478971565
- [7] J. FOLEY, "Computer graphics: principles and practice", Addison Wesley, Boston, Massachusetts, 1990, pp. 123-756, ISBN: 0201121107
- [8] GOOGLE CODE LAB; "O3D Technical Overview" 2009, <http://code.google.com/intl/es-CO/apis/o3d/docs/techoverview.html>
- [9] S. HARRINGTON; "Computer Graphics: A programming approach"; McGraw Hill, New York, 1987, pp. 87-258, ISBN 0070267537
- [10] D. HEARN, P. BAKER; "Computer Graphics with OpenGL"; Prentice Hall, Bloomington, 2005, pp. 84-546, ISBN 978-84-205-3980-5
- [11] J. HILERA, S. OTÓN, J. MARTÍNEZ; "Internet Virtual Reality VRML"; Universidad de Alcalá, 2000, pp. 71-181, ISBN: 84-8138-413-5.
- [12] ISO/IEC 14772-2:2004. "The Virtual Reality Modeling Language (VRML) External authoring interface (EAI)". International Web3D Consortium Electrotechnical Commission. Cambridge, Massachusetts. [http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/#ISO\\_IEC\\_14772Part2](http://www.web3d.org/x3d/specifications/vrml/ISO-IEC-14772-VRML97/#ISO_IEC_14772Part2)
- [13] E. JIMÉNEZ, M. PEREZ, E. MARTÍNEZ; "Web3D Virtual Scenarios: Simulation with VRML, Java3D and X3D" Book of proceedings of XVII International Congress INGEGRAF, Sevilla, June, 2005, INGEGRAF, ISBN: 84-923253-3-X
- [14] T. JUURSALU, J. URI, K. KOOV "Visual tags"; 3d technologies r&d. Skonnard, Estonia, 2008, <http://www.3dmlw.com/?id=10971>
- [15] T. PARISI; "Ajax3D: The Open Platform for Rich 3D Web Applications"; Media Machines, Inc., Agust, 2006: <http://www.ajax3d.org/whitepaper/>
- [16] J. SAINZ; "Computer graphics and Architecture: Drawing and Computer Aided Projects"; Nerea, Madrid, 1992, pp. 171-181, ISBN: 978-84-86763-68-8
- [17] A. SKONNARD; "The XML Files: A Quick Guide to XML Schema" MSDN Magazine <http://msdn.microsoft.com/en-us/magazine/cc302086.aspx>
- [18] K. STYRBERG; "Advanced Mesh Modelling, Blender Documentation Volume I - User Guide"; 2005, capitulo 7; <http://docs.huihoo.com/blender/user-guide-2.3/>
- [19] J. TRIAS; "Geometry for Computer Graphics"; Alfaomega, Universidad Politécnica de Cataluña 2005, pp. 172-251, ISBN: 970-150-0978-1
- [20] R. WRIGHT; "OpenGL superbible: the complete guide to opengl programming for Windows"; Waite Group Press, Corte Madera United States of America, 1996, pp. 121-285, ISBN: 978-1571690739