2011 UKSim 13th International Conference on Modelling and Simulation

Managing Industrial Simulator Visual Databases Using Geographic Information Systems

Villagrá ,M.R. , Carretero, A., Maroto, J., Muneta, M.L.
Grupo de Ingeniería Gráfica y Simulación
Polytechnic University of Madrid
Spain
mrvillagra@etsii.upm.es, acarretero@etsii.upm.es, jmaroto@etsii.upm.es, muneta@etsii.upm.es

Abstract— **Geographic Information Systems are developed to handle enormous volumes of data and are equipped with numerous functionalities intended to capture, store, edit, organise, process and analyse or represent the geographically referenced information. On the other hand, industrial simulators for driver training are real-time applications that require a virtual environment, either geospecific, geogeneric or a combination of the two, over which the simulation programs will be run. In the final instance, this environment constitutes a geographic location with its specific characteristics of geometry, appearance, functionality, topography, etc. The set of elements that enables the virtual simulation environment to be created and in which the simulator user can move, is usually called the Visual Database (VDB). The main idea behind the work being developed approaches a topic that is of major interest in the field of industrial training simulators, which is the problem of analysing, structuring and describing the virtual environments to be used in large driving simulators. This paper sets out a methodology that uses the capabilities and benefits of Geographic Information Systems for organising, optimising and managing the visual Database of the simulator and for generally enhancing the quality and performance of the simulator.**

Keywords-component; Real Time, Virtual Reality, GIS, Geographic Information System, driving simulators.

I. Introduction

Inside virtual worlds, among the requirements demanded of industrial simulators used for training drivers of land vehicles are visual realism and visual continuity. Therefore, the environment in which the simulation takes place must have a visual quality that leads to a sensation of user immersivity and will enable the user to detect, recognise and identify functional elements as well as their appearance. On the other hand, visualisation must occur without any abrupt variations being noted in the number of images per second being shown, which should never be below 30.

In order to display visual realism, the projection systems used must provide high resolution images, for example, 2048x1536 pixels. In addition, the scenes shown, in which the simulation takes place, are usually very extensive and have a large number of elements. This produces a very high work load for the display system as well as the graphic engine, which is a real-time application whose mission is to manage the database containing the geometric models, perform the calculations and generate the images that the user will see during the simulation.

For all the above reasons, the need to devote efforts to improving the efficiency of the components of the simulator is undisputable. Part of the problem approached in this paper is related to creating an efficient Visual Database. A study needs to be conducted that takes account of parameters such as the extension of the environment, its complexity or the degree of realism demanded by the specifications of the project.

On the other hand, the addition of computers to the process of generating mapping products has made it possible to convert traditional maps into something more than a static representation of the terrain. They have become essential elements for management and planning and for supports for modelling specific phenomena located on that terrain. All this has led to the development of specific graphic information management software: display units, CAD applications or computer-aided mapping and also to the simultaneous use of alphanumeric information, resulting in Geographic Information Systems (GIS), which are characterised by their ability to integrate graphic databases (DB) with alphanumeric databases and store in both of them the topological relationships among the elements represented.

Driving simulators require a virtual environment, either geospecific, geogeneric or a combination of the two, over which the simulation programs will be run. In the final instance, this environment constitutes a geographic location with its specific characteristics of geometry, appearance, functionality, topography, etc. The set of elements that enable the virtual simulation environment to be created and in which the simulator user can move, is usually called the Visual Database (VDB). It is a collection of descriptions that let the display unit represent in real time the objects making up the scene where the simulation takes place.

The traditional view held regarding the use of Geographic Information Systems is that they are basically for structuring and analysing data that contains geographic information. However, in any driving simulator development project a virtual environment needs to be available to "recreate" an actual or generic scene through which the vehicle can move. If this scene is understood as a geographic location, an associated geographic information system can be generated.

978-0-7695-4376-5/11 $26.00 © 2011 IEEE
DOI 10.1109/UKSIM.2011.27

IEEE computer society

II. Geographic Information System (GIS)

An Information System is a set of inter-related elements that are ordered according to certain rules. In a Geographic Information System, given that part of the information managed is geographic, those elements also occupy a position in space.

GIS are developed to handle enormous volumes of data and are equipped with numerous functionalities intended to capture, store, edit, organise, process and analyse or represent the geographically referenced information. They obviously allow the automation of all the processes as well as their integration with data or systems that work with different formats.

GIS are used in many sectors for their capability to process existing information and generate new information to form the basis of an important aid to decision-making. Although they may give the impression of being merely practical applications based on specific programs, they work with their own methodology and possess an important theoretical core that combines the concepts of different disciplines (topology, spatial statistics, computational geometry, etc).

The ability to generate 3D scenes in a Geographic Information System is still not virtual reality, strictly speaking [1], although large steps forward are being made towards the creation of 3D displays and symbols. Nowadays, databases are able to include geometry models that are 3D solid objects that represent features such as buildings and other objects in the scene. The capability to show 3D symbols exists to represent points, lines and polygons as well as realistic textures. Numerous 3D object libraries are available as well as translators to more usual 3D formats like OpenFlight, 3DStudio and VRML. The symbols linked to these models can be geospecific for the symbols of the 3D markers or geogeneric for the markers of a feature class. The multi-resolution global display of geographic information is possible and this allows a dynamic display of the information by working directly with the geodatabase of the GIS [2].

The constant growth of 3D GIS systems and their interaction with CAD, virtual reality (VR), and graphic computation, are having an impact on these systems and bringing them ever closer to one another. In this context, urban 3D models can participate interactively in any of the stages of the decision-making processes that are characteristic of GIS [3].

III. Virtual Reality

Virtual Reality is a set of technologies that allow the user to interact with a computer-simulated environment of a non-existent reality by means of hardware and software, the environment either being real or invented.

What is most important in Virtual Reality [4] are not the technological advances but the growing demand for these technologies and techniques in order to increase productivity, improve team communication and reduce costs.

The Visual Data Base (VDB) is a collection of descriptions on how to draw the objects making up the scene. This must be interpreted by the Visual System software. Therefore, the display unit will allow the scenes where the simulation is taking place to be graphically represented in real time.

Storing the information on the objects and the virtual world is performed in the Visual Data Base of the simulator, where there is usually another scene configuration file that is in charge of correctly assembling the environment depending on the user's position, the functional and geometric features and the visual appearance, etc. This description file stores the objects, the programs that describe the actions of these objects or the user, the lighting methods, control mechanisms and hardware support, etc.

The information about the objects [5] of the VDB stored in the simulation environment configuration file is as follows:

- Objects: The objects in a virtual world can have geometry, hierarchy, behaviour and other attributes. The characteristics of the objects have a considerable impact on an application's structure and design. In accordance with this a list of attribute-value pairs is used to describe the objects. By means of these attributes, changes can be added to the objects without the need for any changes in their data structure.
- Position, orientation and transformations: An object must be situated and oriented in the simulation environment. That is, it has location and also spatial organisation. Most objects can have these modifiable attributes to which also scaling, translation, rotation or symmetry operations can be applied.
- Hierarchy: One object can be related to another using a hierarchical structure that would contain the parental ties between siblings. Each object undergoes the transformations applied to its father, which then passes them on to its own siblings.
- Bounding volume: Every object is contained in a rectangular or spherical parallelepiped volume called a bounding volume. This aids the rapid use of the object during its rendering.
- Object geometry: Modelling shapes and their geometry is based on the concepts mentioned regarding graphic computation. Many models very profusely seek the exact geometry in respect of the real-world objects.

IV. Methodology

As previously stated, the traditional view of Geographic Information Systems is that they are basically for structuring and analysing data that contain geographic information. However, generating a realistic driving simulator requires a virtual environment that represents a real or generic scene through which the vehicle can move while that information is stored in the Visual Database. If this scene is understood as a geographic location where all the elements and all the required attributes have a spatial position, the possibilities offered by a Geographic Information System associated with the simulator for managing that Database, can be appreciated. In this way, the scene can be structured and analysed by using the advantages of the GIS for organising and optimising the visual database and for generally enhancing the performance of the simulator. Using a Geographic Information System has some major benefits:

It lets information be imported from different formats with the purpose of comparing, cleaning and unifying the starting-out information of a project. This feature avoids problems of redundant data or discrepancies since similar although different files have been worked with.

It is a very powerful geographic data analysis tool, which can be very useful when a surface needs to be optimised, for example, as it lets us analyse the visibility of the terrain areas from a point or a path.

It enables the loading of elements positioned on the surface to be optimised according to their proximity to the user's viewpoint. It also offers the option to insert elements, which, although they do not really exist on the terrain are necessary for generating the virtual environment (logical elements).

Attributes can be assigned to the geometric features. Therefore, each element of the environment is perfectly referenced and located. This feature, in turn, allows conducting an analysis of the elements through an attributes search.

It allows working with the positioning of elements and their attributes so that geoprocessing models can be generated or tools developed in various languages that in turn generate the output files required to use the graphic engine.

Some GIS allow their own data server to be generated, so that it can provide a service to clients located in field work or offices located at other sites. That is, an information server can be generated that is accessible via the Internet, that lets clients access that information either through "read only" files or with the possibility to edit them. This feature gives access to the same, exact information regardless of the geographic location of the work station. Therefore, for example, a field work client can connect to the database and edit the information directly in the field by using a mobile device or a 3G connection.

The development that will now be set out in this paper is focused on creating, analysing and optimising the Visual Database of the virtual environment that is associated with both railway and vehicle environments.

This kind of environment is characterised by its affecting a very extensive terrain surface with a very varied type of environment, which includes the natural environment that can be found in interurban areas as well as in urban areas. It contains elements in a very wide range of visible distance, thus the detail to be shown on the display is very different. There are internal and external zones, with or without functional elements, both static and dynamic, etc, which makes them highly complex environments.

A working methodology is proposed that takes advantage of the capabilities and benefits of Geographic Information Systems to organise, optimise and manage the Visual Database of the simulator and to generally enhance the quality and performance of the simulator, which will be valid and exportable to any kind of geographic working environment.

A. General flowchart of the methodology

The first step is to define the scope of the project: extension, visual quality, degree of specificity, functionality… Once the project is defined, it is necessary to obtain graphical, topographic, map and semantic information. This information will depend on values set in the functional requirements specification. For example, the extension of topographic or cartographic data will depend on the extension of the project virtual environment. In the same way, the type of information necessary will vary depending on the realism level and quality set on the functional requirements specification for this specific virtual environment. At this point, it includes obtaining information related exclusively to, for example, the underground track that is going to be represented as the track scheme, traffic lights or track circuits definition and the situation; or traffic information referred to lanes, intersections, signals or traffic lights when dealing with driving simulation placed in the urban environment. Graphic information will also be necessary, such as videos or photos of simulated zones for the element generation that will be located over the surface (buildings, trees, urban furniture…).

Once all information has been collected it is necessary to filter, purge and manage it. Most of the delivered vector files have wrong z-measures, overlapped elements and a lack of structure. Because of this, these files need to be submitted to a topological correction before their use. Most of these errors are corrected by topological rules (for point, lines or areas entities).

Once the information is purged and filtered, it is poured into a Geographic Information System. In order to do this, first of all it is necessary to create a Geo-database from which the datasets will hang where in turn, the feature classes hang. The graphic information coming from files with vector data is imported directly to the Geo-database allocating its placement according to its use. In other words, six datasets are built up: "Topography", "Cartography", "Objects", "Track" and "Simulation". All the feature classes referred to Z-points, contours, digital and terrain models must be hung inside the "Topography" dataset. In the same way, all feature classes referred to sidewalks, curbs, roads, buildings, ways, rivers… must be attached to the "Cartography" dataset. All the elements located over the surface such as trees, streetlights, urban furniture… must be placed in the "Objects" dataset, Inside the "Traffic" dataset, with every feature class being somehow related to the traffic model, for example, paths, nodes, stop lines, traffic lights, signals… Every feature class related to track description - rails (for railways), lanes (for roadways), stops, railway traffic lights, track circuits- must be in the "Track" dataset. And finally, the "Simulation" dataset must contain all the necessary feature classes defining any aspect needed for later simulation, for example, tunnel zones for a light engine, interior and exterior zones for environmental effects…

The next step, once all the graphic information has been introduced into the Geo-database, is to introduce the alphanumeric attributes associated with each element. These attributes will vary according to the feature class being studied.

In the following paragraphs, the most relevant and newest points of the methodology are discussed, such as the generation and management of the database or the spatial analyses carried out.

B. Generating the Database

Either vector, mapping or topographical information can be taken as a starting point. On occasions, terrain information is necessary. This can be given in the form of a digital terrain model (DTM), a flat text, rich point clouds [6] or as orthophotographs.

Apart from the foregoing, in order to provide greater realism to the environment where the driving will take place, information on the objects in the scene is required (trees, traffic lights, traffic signs, street furniture…)

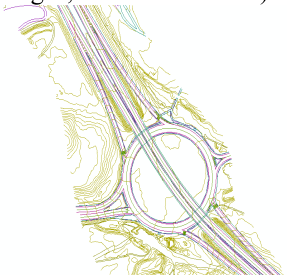


Figure 1. Vector mapping of a specific zone of Madrid

1) Structuring the GIS database

- Geodatabase

For every project carried out, a geodatabase in which all the information referred to the project will be structured (and which must also be georeferenced) must be generated.

- Dataset

Once the information about the terrain and the elements making up the environment is available, it is then imported to the geodatabase commented in the preceding section. The information contained in this geodatabase is structured into several homogeneous datasets that include the tables that they contain. They sort and restrict the data as well as how the data are related to one another, and all of this within a set geographic framework. Therefore the geodatabase will be structured into the following datasets:

a) Topography: this groups all the information existing about the terrain. This vector information can come from maps or digital terrain models and orthophotographs.

b) Mapping: this groups the vector information referred to elements built on the surface.

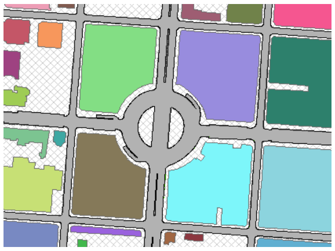


Figure 2. GIS information on pavements, roads and buildings

c) Objects: these bring together all the information on the objects, either static ones or ones with dynamic behaviour (traffic lights, street lights, etc), either functional or non-functional for simulation purposes, which are located on the surface of the scene where the training is in course. This dataset includes a wide variety of information, such as that referred to street furniture or vegetation, which is mainly static and not functional, or information referred to vertical and horizontal signposting, which may be either static or dynamic, but always functional.

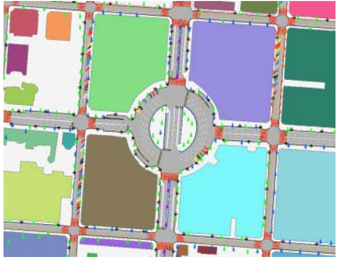


Figure 3. GIS information on environmental objects

d) Traffic: this encompasses all the information required for a full definition of the paths that vehicles can take, controlled by the driving simulator traffic model, so that the required files can then be compiled. The definition of the paths not only includes their steering geometry but also the existing topological constraints.

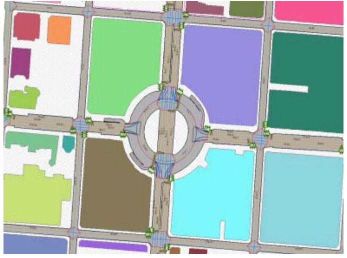


Figure 4. GIS information on paths for the traffic model

e) Simulation: this compiles all the logical information required generated for the simulator to function properly (logical features).

These datasets collect all the information referred to the correctly structured project according to its nature. The feature classes, which are the features supporting the graphic and semantic information are then hosted in each dataset.

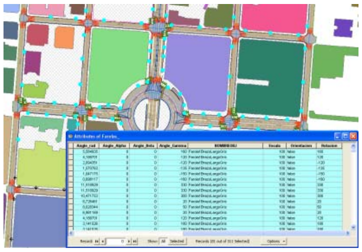
All the feature classes refer to a specific generic project. For each specific project the need to generate feature classes may arise to meet some specific requirement.

2) Entering attributes

Any technical features that are thematically related to geometry of the same nature must be stored in an orderly fashion. To do this, feature classes are used.

When a feature class is generated from zero, fields are created where the type of feature generated is shown (Point, PointZ, Line, Polygon), and its properties, such as the area and the perimeter.

In line with the above section, outside GIS, vector features are considered as purely geometric features. The geometric features need to be enriched with thematic values that provide information on each element. To achieve this, fields must be created in each feature class and the relevant information entered.



Attributes of Farolas_

Angle_rad	Angle_Alpha	Angle_Beta	Angle_Gamma	NOMBREOBJ	Escala	Orientacion	Rotacion
[illegible]	0	0	160	FarolaBrazoLargoGris	100	false	160
[illegible]	0	0	120	FarolaBrazoLargoGris	100	false	120
[illegible]	0	0	-120	FarolaBrazoLargoGris	100	false	-120
[illegible]	0	0	-135	FarolaBrazoLargoGris	100	false	-135
[illegible]	0	0	-160	FarolaBrazoLargoGris	100	false	-160
[illegible]	0	0	-160	FarolaBrazoLargoGris	100	false	-160
[illegible]	0	0	330	FarolaBrazoLargoGris	100	false	330
[illegible]	0	0	330	FarolaBrazoLargoGris	100	false	330
[illegible]	0	0	300	FarolaBrazoLargoGris	100	false	300
[illegible]	0	0	20	FarolaBrazoLargoGris	100	false	20
[illegible]	0	0	50	FarolaBrazoLargoGris	100	false	50
[illegible]	0	0	20	FarolaBrazoLargoGris	100	false	20
[illegible]	0	0	120	FarolaBrazoLargoGris	100	false	120
[illegible]	0	0	160	FarolaBrazoLargoGris	100	false	160

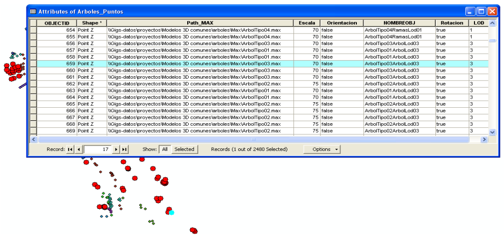
Figure 5. Semantic information on objects stored in the GIS

As an example, in order to give a main idea about the advantages provided by incorporating this information into the database, the attributes associated to the "Trees" feature class will be commented below.

When the trees are positioned, a spatial analysis will be carried out prior to their being positioned. The main idea of the spatial analysis lets the tree elements be discriminated according to track proximity and, depending how close or far away they are, a more or less elaborate model can be loaded. Once the spatial analysis has been carried out, the appropriate attributes are introduced in relation to the area they are in, in order to export this information to a text file that can be read by the virtual modeling software.

The fields that must be introduced are:

- PATH_MAX: this attribute is stored in a text field. In this field the path on the server is processed where the 3D object file is stored.
- SCALE: In this field (double type) a scale is stored with which the engine will load the object reference point. The reference scale takes the original file size.
- ORIENTATION: This is a Boolean type field in which it is specified if the object to be placed over the surface will be positioned perpendicular to the positioning face, or will always be positioned vertically.
    - Position perpendicular to the face: true
    - Vertical position: false
- NAME_OBJ: this is a text type field. In it is stored the name of the object to be loaded in the scene according to the proximity of the area which it is in. In this way, the 3D files and the database structure are optimised.
- ROTATION: In the "Trees", feature class the "Rotation" field is a Boolean type field and defines whether the object will have a random rotation or not when it is loaded over the surface.
    - Random rotation: true
    - No rotation: false
- LOD: In this text type field, for every point, a number is stored (1, 2 or 3) that indicates the proximity area in which the number is to be found.



Attributes of Arboles_Puntos

	OBJECTID	Shape *	Path_MAX	Escala	Orientacion	NOMBREOBJ	Rotacion	LOD
	654	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo04.max	70	false	ArbolTipo04RamasLod01	true	1
	655	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo04.max	70	false	ArbolTipo04RamasLod01	true	1
	656	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo03.max	70	false	ArbolTipo03ArbolLod03	true	3
	657	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo01.max	70	false	ArbolTipo01ArbolLod03	true	3
	658	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo01.max	70	false	ArbolTipo01ArbolLod03	true	3
	659	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo03.max	70	false	ArbolTipo03ArbolLod03	true	3
	660	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo03.max	70	false	ArbolTipo03ArbolLod03	true	3
	661	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo03.max	70	false	ArbolTipo03ArbolLod03	true	3
	662	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo02.max	75	false	ArbolTipo02ArbolLod03	true	3
	663	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo01.max	70	false	ArbolTipo01ArbolLod03	true	3
	664	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo01.max	70	false	ArbolTipo01ArbolLod03	true	3
	665	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo02.max	75	false	ArbolTipo02ArbolLod03	true	3
	666	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo02.max	75	false	ArbolTipo02ArbolLod03	true	3
	667	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo02.max	75	false	ArbolTipo02ArbolLod03	true	3
	668	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo02.max	75	false	ArbolTipo02ArbolLod03	true	3
	669	Point Z	\\Gigs-datos\proyectos\Modelos 3D comunes\arboles\Max\ArbolTipo02.max	75	false	ArbolTipo02ArbolLod03	true	3

Record: 17 Show: All Selected Records (1 out of 2480 Selected)

Figure 6. Detail of attributes included in the "Trees" feature class.

3) *Spatial analyses*

When all the geographic and thematic information has been entered in the zone to be studied and then virtually recreated, a Geographic Information System can carry out studies either based on the implicit geographic data in each feature class [7], or on the alphanumeric information previously entered. These studies allow analysing different models, both track and environment models, making this approach scalable to any kind of scenario, i.e. urban or country zones, with tunnels or bridges, mountain or plain environments...

By taking advantage of theses spatial analyses studies could be carried out for optimising design and VDB management in order to improve the graphic engine performance. This imposes certain rules which, although they are flexible and hardly compatible, determine the VDB structure. In order to do that, the first step to be taken is to find out any problems or difficulties that appear in the scene creation phase and that decrease graphic engine performance.

For every path and any direction, studies can be carried out on:

- Visibility: The purpose of this analysis is to study the visible areas from the viewpoint of a user of a large environment such as a city. To optimise the environment, the Visual Database is analysed to find environmental zones that are never seen from the simulator user's viewpoint. These zones can be eliminated by optimising the polygon load. For this analysis it is necessary to fix certain parameters like the azimuth or vertical angle, or the height of the viewpoint in order to simulate the driver's head movements and the distance from the ground to the driver's eyes.
- Paths analysis: The VDB can be subjected to a paths analysis, for example, for the evacuation of a city or a building (**Error! Reference source not found.**), or for calculating alternative paths for the traffic model. This analysis could use the same created network for managing the traffic model.

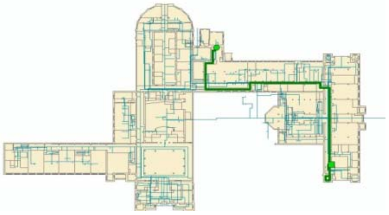


Figure 7. Route analysis for building evacuation

- Polygon load level: The purpose of this analysis is to locate any black spots that might diminish the performance of the final application. These black spots may be due to an accumulation of polygons in a zone caused by excess detail in the generation of the 3D model or to an accumulation of objects at a particular point on the display (Fig. 8). For this analysis certain map algebra functions are used for this analysis such as Focal Statistics or Reclassifying.

- Proximity to driving areas: Since we are dealing with a driving simulator, the vehicle will follow paths that are restricted to roads or tracks in the case of railway simulators. That is, there is no complete freedom to move anywhere in the environment. Good use can be made of this feature to optimise the polygon load of the environment. By generating path buffers, the information can be filtered by proximity to the zones through which the vehicle can be driven.

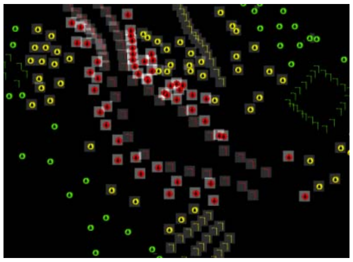


Figure 8. Mixed analyses for polygon load level and track proximity

V. Conclusions

A visual database is a collection of descriptions about how to draw the scene members, and which must be understood by the graphic engine software, so that the display will provide a real time graphic representation of the scenes in which the simulation is developed.

Based on the above conclusions, work has been done to optimise the content and the visual database structure of a virtual environment employed in a real time industrial driving simulator, taking into account that it is a methodology that is only applicable to one-dimensional environments such as urban [8] or inter urban railways, in which the driving scenarios represent the main directions from which the environments can be organised and built.

Taking advantage of Geographic Information Systems, the visual database is designed and structured by always bearing in mind the functionalities that the simulator will have. By proximity algorithms the entities are grouped (according to graphic engine requirements and element functionality) so that when they are exported to graphic engine format, the performance will be the best possible.

The ideal way of generating the environment would be by a single compressed file that would include all the files that comprise the scene, either texture or geometry, because access time to the disk would be reduced as much as possible. Every independent static geometry file stored in the general file has its own levels of detail (LOD) directly associated with it. To enter these LODs, it would be necessary to divide the environment into zones, because the database structure would be organised by taking account of the minimum number of objects because object abuse penalizes graphic engine performance [9].

The minimum number possible of texture atlases would be used for each element, differentiating the atlases for transparent textures. The main idea would be to use, the minimum number of textures for every independent element and reuse them as much as possible from one zone to another (for example, all the texture referred to repetitive elements can be joined in a single atlas in order for them to be reused by the graphic engine since they are loaded in the GPU).

Functional elements such as vehicles (either automatically driven cars or trains,), traffic lights that change their lights; pedestrians… must be independent elements and be instanced by the graphic engine. In the event of elements changing their texture, this change would be performed by the graphic engine by changing the map coordinates in the associated texture atlas. Other ways, though less efficient, are to change the entire object texture or even change the element in order of efficiency.

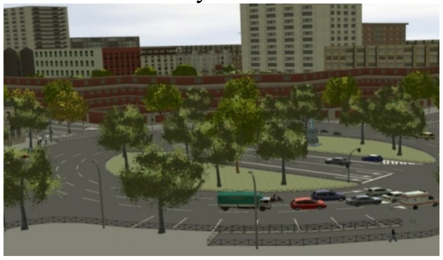


Figure 9. Example of a virtual environment generated using this methodology

References

[1] Abdul-Rahman, Sisi Zlatanova, Volver Coors,, “Innovations in 3D GeoInformationSystems,” 760 páginas. ISBN:354036997X.. 2006.

[2] Tomlinson, R “Thinking about GIS:Geographic Information System Planning for Managers,” ESRI Inc. ISBN 978-1-58948-158-9. California 2007.

[3] Huayi Wu, Zhengwei He, Jianya Gong,. “A virtual globe-based 3D visualization and interactive framework for public participation in urban planning processes.” 34 291–298. Elsevier. ISSN: 0198-9715.. 2010.

[4] Heim, M “The Metaphysics of Virtual Reality,” Oxford University Press. ISBN 0-19-508178-1 . New York, 1993

[5] Clemente, P,, “Creación de entornos virtuales con Crystal Space 3D,” Master Final Project. Madrid September 2006

[6] Stephan Nebiker , Susanne Bleisch, Martin Christen “Rich point clouds in virtual globes – A new paradigm in city modeling?,” Computers, Environment and Urban Systems. Elsevier. . ISSN: 0198-9715. 2010.

[7] Haik Lorenz, Jürgen Döllner “3D feature surface properties and their application in geovisualization.,” Computers, Environment and Urban Systems. Elsevier. ISSN: 0198-9715.2010

[8] Van Dipten, R., & Van Klaveren, E,, “The surplus value of virtual reality in urban planning,” Pags, 683-686. Second Joint Conference & Exhibition on Geographical Information,. ISBN 90 5199 2688. Barcelona 1996.

[9] E. Gobbetti, E. Bouvier “Time-critical multiresolution rendering of large complex models”. Computer-Aided Design 32 785–803. Elsevier. ISSN: 0010-4485. *2000.*